ClockIDS: A Real-Time Vehicle Intrusion Detection System Based on Clock Skew

Yilin Zhao, *Graduate Student Member, IEEE*, Yijie Xun, *Member, IEEE*, and Jiajia Liu, *Senior Member, IEEE*

***Abstract*—Although intelligent connected vehicles (ICVs) can better assist drivers and improve their driving experience, they have huge network security problems and are frequently attacked. This is because the vehicle network is connected to the Internet, which expands the attack surface of ICV, and attackers have more ways to launch attacks. In recent years, many security experts fight against attackers and propose various types of vehicle intrusion detection systems (IDSs) to protect the controller area network (CAN). However, with the continuous enhancement of attack means, especially the appearance of the masquerade attack, most IDSs are no longer applicable. In this article, we design a new fingerprint-based vehicle IDS to protect the CAN, called ClockIDS. It establishes a unique fingerprint for each electronic control unit (ECU) based on clock skew. On this basis, ClockIDS realizes the functions of intrusion detection and attack source identification by utilizing the empirical rule and dynamic time warping. It neither occupies the bandwidth of CAN bus nor needs to modify the CAN protocol. Our experiments on two real vehicles show that ClockIDS can establish a unique fingerprint for ECU without being affected by the size of message period, and can detect three types of attack with a detection accuracy of 98.63%. In addition, this system can identify the attack source, and the average recognition accuracy is 96.77%. Furthermore, ClockIDS has high real-time performance, and the average time cost of each detection is only 1.99 ms.**

***Index Terms*—Clock skew, controller area network (CAN), electronic control unit (ECU), intelligent connected vehicle (ICV), intrusion detection system (IDS).**

I. Introduction

INTELLIGENT connected vehicles (ICVs) have entered the era of digitalization and intelligence. According to research [1]–[3], they will account for 86% of the global automobile market in 2025. This is because ICVs, as a mobile platform with computing resources, have been added many functions to better assist drivers. For example, Vehicle-to-vehicle [4] communication can use information interaction between vehicles to share road conditions and reduce traffic accidents. The vehicle navigation function can display real-time road conditions, select better routes, and provide conditions for aid in emergencies. The auto-driving system frees the driver's hands and can deal with complex road conditions accurately, which considerably improves the safety of cars.

Although ICVs have a bright future, the security problems are extremely serious [5]–[7]. As the most widely used in-vehicle network, controller area network (CAN) is often attacked. In 2015, Miller and Valasek [8] utilized system vulnerability to remotely change the gateway system of Jeep, thereby controlling the braking system and engine. In 2017, the Keen Security Laboratory of Tencent realized full control of Tesla Model S with the latest firmware through the Internet [9]. In 2021, the Keen lab intruded into the T-box chip of Mercedes Benz and successfully sent arbitrary data frame to the CAN bus. According to a report by Upsteam [10], the number of automotive cybersecurity incidents increased by 605% from 2016 to January 2020.

Automotive security is related to the property and life safety of users, so it has attracted the widespread attention of researchers who put forward a variety of defensive measures to protect in-vehicle networks. At present, there are two main defense measures. One is designing new protocols that can provide security mechanisms, such as message authentication code (MAC) [11]–[14]. However, the development of this method in automobiles is limited by the bandwidth of CAN bus. Each message of CAN bus can only carry 8-bytes payload, which makes the majority of classical cryptographic algorithms infeasible. The intrusion detection system (IDS) [15], as another defense method, can deal with this problem effectively. It neither changes the existing firmware in vehicles nor affects the communication between electronic control units (ECUs).

A. Related Works

In the existing works, IDSs can be divided into traffic based and fingerprint based. The traffic-based IDS is usually implemented by using identifier (ID) period [16], ID sequence [17], information entropy [18], [19], message contents [20], [21], or remote frame [22]. Although the above IDSs provide defense for vehicles, attackers can imitate the frequency/time interval of ID to bypass traffic-based IDSs. The fingerprint-based IDS

Manuscript received 30 September 2021; revised 14 January 2022; accepted 6 February 2022. Date of publication 15 February 2022; date of current version 24 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62001393; in part by the Natural Science Basic Research Program of Shaanxi under Grant 2020JC-15; in part by the Fundamental Research Funds for the Central Universities under Grant D5000210817; in part by the Xi'an Unmanned System Security and Intelligent Communications ISTC Center; and in part by the Special Funds for Central Universities Construction of World-Class Universities (Disciplines) and Special Development Guidance under Grant 0639021GH0201024. *(Corresponding author: Jiajia Liu.)*

The authors are with the National Engineering Laboratory for Integrated Aero–Space–Ground–Ocean Big Data Application Technology, School of Cybersecurity, Northwestern Polytechnical University, Xi'an 710072, Shaanxi, China (e-mail: liujiajia@nwpu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2022.3151377

TABLE I
COMPARISON OF THE FINGERPRINTING APPROACHES

	Viden [23]	VoltageIDS [24]	Scission [25]	CIDS [27]	ClockIDS
Physical information	voltage	voltage	voltage	clock skew	clock skew
Fingerprint object	ECU	ECU	ECU	ID	ECU
Computing resource	low	high	high	low	low
Intrusion detection	-	√	√	√	√
Attack source identification	√	√	√	-	√

is an interesting scheme, which is fingerprinting ECUs by exploiting differences in electrical characteristics of different ECUs, such as voltage and clock skew. Such schemes can be used not only for intrusion detection but also for attack source identification.

Table I shows a comparison of existing methods based on fingerprint schemes. Cho and Shin [23] used voltage profiles to establish fingerprints for ECUs and designed Viden, a system to identify attack source. Choi *et al.* [24] designed VoltageIDS, which combines the time-domain and frequency-domain characteristics of voltage signals with machine learning. Kneib and Huth [25] utilized physical characteristics from analog values of CAN frames to assess whether it was sent by the legitimate ECU, and the average accuracy is 99.85%. However, it is challenging for voltage-based IDS to detect intrusion with real-time performance in practice restricted by ECU computing resources [26]. A novel clock skew-based scheme appeared in [27], and Cho and Shin designed a clock skew-based IDS, called CIDS, which calculates the clock skew from the arrival time of the ID and uses it as an ECU fingerprint. However, in the experiment, we found that CIDS can create a fingerprint for each ID, but cannot create a unique fingerprint for ECU. This is because the fingerprint established by CIDS varies not only with ECU but also with the size of ID period. In other words, the same ECU has different fingerprints for different period IDs.

B. Main Contributions

In this article, we present ClockIDS, a real-time IDS for CAN bus based on clock skew. It does not occupy the bus bandwidth and computing resources. Specifically, ClockIDS connects to the CAN bus as a monitoring unit and establishes a unique fingerprint for each ECU by recording the content and arrival time of data frames on the CAN bus. When new data frames arrive, it detects intrusions by using the empirical rule (ER) algorithm, and then finds the ECU who launched the attack by using the dynamic time warping (DTW) algorithm. The experimental results show that ClockIDS has good performance in intrusion detection and attack source identification. Our contributions are summarized as follows.

1) We design a method to establish a unique fingerprint for each ECU, which is not limited by the length of ID period. Even if each ECU can send IDs of different periods, its fingerprint is unique.

Start	Arbitration		Control			Data Field	CRC Field		ACK Field		End
SOF	Identifier (ID)	RTR	IDE	r0	DLC	Data	CRC Sequence	CRC	ACK	ACK Slot	EOF
1	11	1	1	1	4	0-64	15	1	1	1	7

SOF: Start of Frame
IDE: Identifer Extension Bit
r0: Reseved bit0
CRC: Cyclic Redundancy Check
RTR: Remote Transmission Request
DLC: Data Length Code
ACK: Acknowledgement Character
EOF: End of Frame

Fig. 1. Standard data frame of CAN 2.0. The identifier represents data type and priority of messages. The RTR indicates the type of frame.

2) We develop an IDS based on ECU clock skew, called ClockIDS. It can detect the spoofing attack, bus-off attack, and masquerade attack, and the detection accuracy is more than 98%. In addition, it can also identify the attack source of spoofing attack and masquerade attack with a recognition accuracy of more than 92%.
3) ClockIDS has a good real-time performance. It takes only 1.99 ms to identify an intrusion and finds the attack source ECU. Therefore, it can quickly inform system or driver to take measures to deal with attack and record the intrusion data for later inspection.
4) ClockIDS does not require any changes to the CAN bus and is not bandwidth or resource intensive. It acts as a monitoring unit that can be accessed directly from the second onboard diagnostics (OBD-II) port.
5) Our system has strong robustness and practical value. We have carried out experiments on two vehicles, and the experimental results show that the performances are outstanding on the two vehicles. Therefore, our system can be directly applied to vehicles.

The remainder of this article is organized as follows. Section II contains the necessary background knowledge. The description of the system model is detailed in Section III. Section IV describes the details of ClockIDS, which is evaluated and analyzed in Section V. In Section VI, we discuss the limitations of ClockIDS and the future work. Finally, we conclude this article in Section VII.

II. BACKGROUND

A. CAN

CAN is a distributed control network proposed by Bosch in Germany. Because of simple wiring and strong real-time performance, CAN is widely applied to in-vehicle network. Next, we will introduce the data link layer, physical layer, and security problems of CAN.

1) Data Link Layer: The structure of standard data frame is shown in Fig. 1. The start of frame (SOF) includes a dominant bit, and it is used to synchronize the clock in ECU. The arbitration field contains ID and remote transmission request (RTR). The ID is used to prioritize the message. The smaller the value, the higher the priority. It is worth noting that each ECU can send and receive multiple IDs with different periods. RTR indicates the type of frame. If the value is logical 0, the frame is data frame. Otherwise, the frame is remote frame.

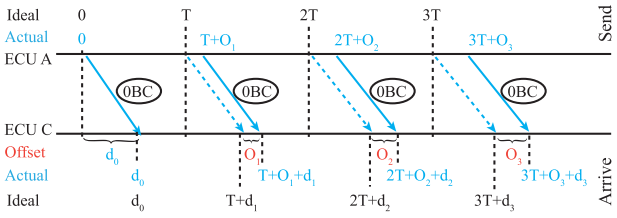


Fig. 2. Analysis of clock behavior in CAN bus. T is the ideal transmission period, d is the transmission delay, and O is the difference between the ideal arrival time and the actual arrival time.

The length of data field is 0–8 bytes and is controlled by data length code (DLC).

In communication, the sender ECU transmits data frames with IDs to CAN bus. Other ECUs listen to the CAN bus and selectively receive messages based on the value of ID. It is noted that the CAN bus only allows one message to be transmitted at a time. If multiple messages occupy the bus at the same time, the message with high priority will be allowed to send, and the message with low priority will be stopped until the bus is idle.

2) Physical Layer: In the CAN bus, all ECUs are connected by twisted pair. Each ECU contains a transceiver and a controller. The function of the transceiver is to realize the conversion between logical signal and physical signal. The controller is responsible for encapsulating data into frames and transmitting them to the transceiver. It is remarkable that the time of sending messages is determined by the quartz crystal clock of the controller [28]. In other words, the controller sends data messages to the bus only when the clock of controller reaches the specified time.

3) Security Problems: There are three main security problems in CAN bus. First, it has no data encryption. This makes it easy for attackers to analyze data frames. Second, CAN Bus without message authentication. The attackers can easily cheat ECUs by controlling external devices or any ECU on the CAN bus. Third, the access control of CAN bus is feeble. Attackers can easily access the bus through physical or remote interfaces.

B. Clock Inconsistency

Different clocks have deviations in timing due to subtle differences in hardware, such as material type and manufacturing process. This theory has been proved in [29] and [30]. Next, we describe the phenomenon of clock inconsistency in the CAN bus, and define the terms "period," "offset," and "skew" based on the description in [27].

As shown in Fig. 2, ECU A sends messages at intervals T in the ideal state, which means that the clock of ECU reports the real time at any time. $d_{k(k=1,2,3,\ldots)}$ is transmission delay on the bus, so the arrival time of message is $kT + d_k$ in ideal. However, ECU A does not strictly take T as period, and the sending time has a certain deviation (O). Therefore, the actual sending time of the k message is $kT + O_k$ and its actual arrival time is $kT + d_k + O_k$.

Period: The time interval (T) for ECU to send messages and it follows the true clock.

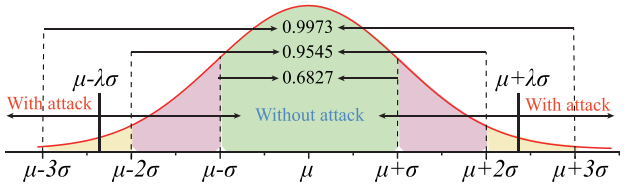


Fig. 3. ER of normal distribution data. μ is the mean of a data set that conforms to the normal distribution, and σ is the standard deviation of this group of data.

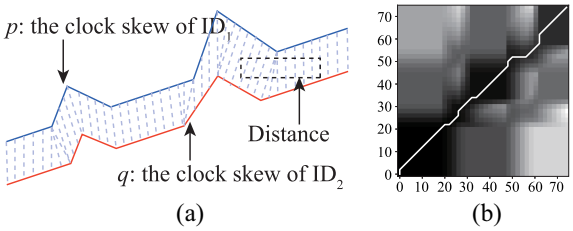


Fig. 4. Principle of DTW. (a) shows the case of matching two similar curves. (b) shows the shortest matching path found using the DTW algorithm.

Offset: The difference of clock (O_k) between ideal and actual arrival time.

Skew: The change speed of clock offset, it can be defined as $(O_k - O_{k-1})/(T + d_k - d_{k-1})$.

C. ER Algorithm

ER is a statistical method, which is often used in IDS. It expresses the objective law of the relationship between probability and distribution. As shown in Fig. 3, if a set of data (S) conforms to normal distribution, it has the following rules:

$$P(\mu - \sigma \le x \le \mu + \sigma) \approx 0.6827$$
$$P(\mu - 2\sigma \le x \le \mu + 2\sigma) \approx 0.9545$$
$$P(\mu - 3\sigma \le x \le \mu + 3\sigma) \approx 0.9973$$

where μ and σ are the mean and standard deviation of S, respectively. It can be seen that most of the data falls in the interval of $(\mu - 3\sigma, \mu + 3\sigma)$, and with the increase of interval, P also rises. When ER is used in IDS, the system needs to set a threshold λ and establishes an interval of $(\mu - \lambda\sigma, \mu + \lambda\sigma)$. If new data frames fall in the interval, they are not attacked. Otherwise, it is under attack.

D. DTW Algorithm

DTW [31] is a classical time-series matching algorithm. It is often used in speech recognition, gesture recognition, and information retrieval [32]–[34]. Fig. 4 illustrates the principle of DTW. As shown in Fig. 4(a), this method matches similar points and calculates the shortest Euler distance to discriminate similarity. The DTW algorithm first constructs an $a \times b$ matrix according to the length of the sequence p and q. The distance $d(p_i, q_i)$ between any two points p_i and q_i in the sequence is calculated and filled in the matrix. Next, DTW uses a dynamic programming (DP) algorithm to find a path with the minimum distance through the matrix, as shown in Fig. 4(b). The smaller the distance of this path, the more matching the two sequences

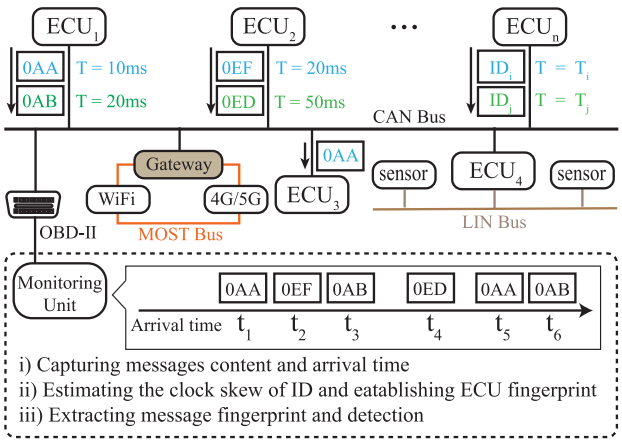


Fig. 5. ICV internal network structure and the working principle of ClockIDS. Each ECU sends many IDs (such as 0AA, 0AB, 0EF, . . .) messages with different T to CAN bus. ClockIDS runs by monitoring unit, which can record the content and arrival time of messages.

are. We use this method to identify the source of ID messages. If ID_1 and ID_2 are extremely similar, they come from the same ECU.

III. System Model

In this section, we first present the current in-vehicle network structure. Then, we explain the working principle and the application method of ClockIDS. Finally, we define the adversary model, which represents the attacker's abilities.

A. In-Vehicle Network Structure

To make ClockIDS more practical, we build it on an in-vehicle network structure of a real ICV. As shown in Fig. 5, the specific structure settings are elaborated as follows.

1) There are multiple ECUs on CAN bus, and each ECU can form a subnet with sensors via local interconnect network (LIN). For example, in Fig. 5, ECU_x ($x = 1, 2, 3, \ldots, n$) is connected to CAN bus, and ECU_4 connects two sensors through LIN bus.
2) Each ECU can send multiple IDs to CAN bus, and the periods (T) of different IDs may be different. For instance, in Fig. 5, ECU_1 sends ID 0AA messages with $T = 10$ ms and ID 0AB messages with $T = 20$ ms. This is to coordinate ID messages sending time and reduce CAN bus arbitration.
3) In the vehicle network, there are many open ports, such as OBD-II directly connected to CAN bus, 4G/5G, and WiFi on media-oriented systems transport (MOST) bus. These ports can increase the functions of vehicles and bring diversified experiences to users. However, due to the vulnerability of the system, they also provide many attack surfaces for attackers.

B. Principle of ClockIDS

Our IDS can be deployed as an in-vehicle ECU or as an additional device to access the CAN bus via OBD-II. Fig. 5 illustrates the working principle of ClockIDS. First, ClockIDS captures messages content (such as 0AA, 0EF, 0AB, . . .,) and records arrival time (t_m, $m = 1, 2, 3, \ldots$). Then, it estimates the clock skews of each ECU to build a fingerprint library (D) according to t_m. This is due to the fact that each ECU has a unique clock skew. Finally, ClockIDS uses the fingerprint library to achieve intrusion detection and attack source identification. When some new messages arrive, our system will extract the fingerprint (κ^{id}) of the messages and compare κ^{id} with the ECU fingerprint (μ_{id}) in D. If κ_{id} matches μ_{id}, the messages are normal. Otherwise, they are intrusion messages, and ClockIDS outputs attack source ECU.

C. Adversary Model

As mentioned in Section II, CAN bus has many security problems, which can be easily exploited and attacked by attackers. On the one hand, attackers can use system vulnerabilities to control internal ECU, and inject data frames into the CAN bus to make other ECUs perform wrong behavior [35]–[38]. On the other hand, the attacker can access external interfaces to invade CAN bus [39], such as OBD-II. Because the external interfaces are basically in vehicles, it is difficult for the attacker to access them without being noticed by the driver. Therefore, this article does not consider intrusion from external interfaces.

In our work, we consider the majority attacks against the CAN bus initiated by internal ECU that are fully controlled by the attacker. Next, we introduce the three attack types as the adversary model.

1) Spoofing Attack: In this attack, the attacker first needs to exploit the vulnerabilities of vehicle system and fully control an in-vehicle ECU through the remote port. Next, the attacker will inject a large number of forged messages into the CAN bus through the controlled ECU to deceive other ECUs and make them perform wrong operations. As shown in Fig. 6(a), the attacker injects a large number of ID 316 messages initially from ECU_A into the bus at a higher frequency through the controlled ECU_B. This makes the number of illegal 316 messages on the bus far exceed the number of normal 316 messages. Therefore, the ECU receiving ID 316 messages will perform more operations caused by illegal 316 messages.

2) Bus-Off Attack: In order to launch this attack, the attacker needs to control an in-vehicle ECU, and interfere with other ECUs sending messages through the controlled ECU, thus causing failures in transmission. This attack takes advantage of the bus arbitration mechanism, i.e., when an ECU sends a message, it will stop sending if there is an interfering high signal on the bus. If it fails multiple times, the ECU will go offline. For example, as shown in Fig. 6(b), ECU_A initially sends ID 316 message. After being attacked, ECU_B causes the ECU_A to send 316 messages to fail by sending signal interference. Sometime later, the ID 316 message does not appear on the bus.

3) Masquerade Attack: This is a more powerful attack combined with spoofing and bus-off attack. It can pose a threat without changing the data flow and ID sequence in the CAN bus. First, the attacker needs to make an attack on an internal ECU and fully control it. Second, the attacker needs to

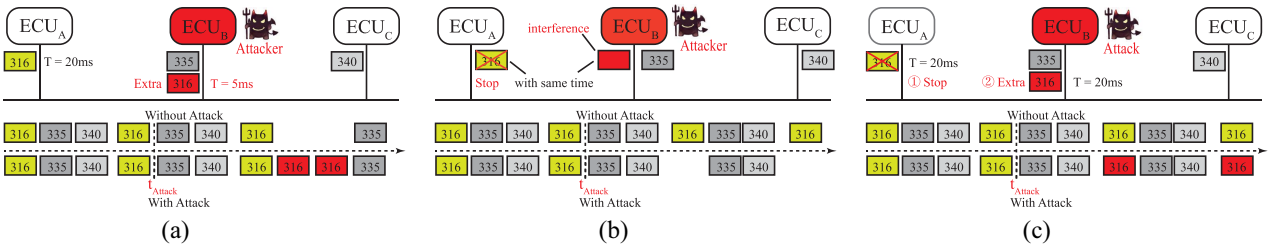


Fig. 6. Adversary model. There are three different attack types to launch an attack on ID 316 that comes from ECU_A. (a) Spoofing attack. An attacker injects a large number of ID 316 messages to trick the receiving ECU. (b) Bus-off attack. The attacker uses arbitration mechanism to make ID 316 messages no longer be sent by ECU_A. (c) Masquerade attack. The attacker uses ECU_B to send ID 316 messages instead of ECU_A.

prevent another ECU from sending ID messages through bus-off attack. Finally, the controlled ECU sends the ID messages with the same frequency. For example, as shown in Fig. 6(c), the attacker first completely controls ECU_B. Next, the attacker launches a bus-off attack on ECU_A by using ECU_B, thereby stopping it from sending ID 316 messages. At the same time, the attacker injects illegal ID 316 messages into the bus at a constant T through ECU_B. It can be observed that the data flow and ID sequence on the CAN bus does not change after being attacked, which makes the traditional methods based on period and information entropy unable to detect this attack.

IV. Real-Time Intrusion Detection System Based on Clock Skew

In this section, we present the design details of ClockIDS, which mainly includes three steps: 1) the establishment of fingerprint database; 2) intrusion detection; and 3) attack source identification.

A. Establishment of Clock Skew Fingerprint Database

As described in Section II, the clocks of different ECUs are inconsistent. Therefore, it is theoretically feasible to use clock skew to establish a unique fingerprint for ECU. Many previous works used clock skew to establish fingerprints for physical devices in [29], [30], and [40]. However, these methods are not suitable for in-vehicle network, because they need to modify the CAN protocol, such as embedding arrival time stamp in packet header. In order to solve this problem, Cho and Shin [27] proposed an evaluation method of clock skew for ECU based on the periodicity of messages. However, in our experiments, we found that this method is to establish fingerprint for ID instead of establishing fingerprint for ECU. This makes it difficult to find the source of attack.

To build IDS based on clock skew, it is very important to estimate the clock skew of each ECU. Our method is improved on the basis of [27].

Step 1: ClockIDS monitors the arrival time of each ID and estimates the period (T^{id}) of each ID by calculating the average arrival time interval of the ID. Whenever new messages arrive, the system updates the ID period. It is worth noting that if the ID message does not arrive at the time it should arrive, the system will assign it a later time.

Step 2: ClockIDS converts IDs with different periods to IDs with the same period ($T^{\text{id}}_{\text{same}}$) by calculating the least common multiple of the different ID periods. Next, the system will filter out a subset (L^{id}) from many messages, where each ID takes $T^{\text{id}}_{\text{same}}$ as the period.

Step 3: According to L^{id}, ClockIDS calculates the clock offset (O) of ECU by calculating the difference between the estimated arrival time and the actual arrival time.

Step 4: ClockIDS uses the recursive least square (RLS) algorithm [27] to calculate clock skew. It updates the clock skew in real time to reduce the errors caused by environmental changes.

By this method, we get that there is only one clock skew for each ECU and it can be used as a unique fingerprint of each ECU.

B. Intrusion Detection

As the ECU sends messages with a clock skew fingerprint, we use this characteristic to achieve the function of intrusion detection and attack source identification in ClockIDS, which are implemented by using ER and DTW. The application of DTW will be introduced in the next section. Then, we discuss the use of ER algorithm to detect intrusion on the CAN bus.

Algorithm 1 shows the pseudocode of ER in ClockIDS. It consists of four steps.

Step 1: *Fingerprinting all ECUs:*

1) For each ID, ClockIDS obtains multiple segments (n) continuous messages (line 3).
2) ClockIDS calculates the average clock skew of each segment to obtain a set of data ($Skew_{\text{id}}$) (line 4).
3) The system calculates the mean value (μ_i) and standard deviation (σ_{id}) of $Skew_{\text{id}}$ for each ID (lines 6 and 7). These initial parameters are used for intrusion detection.

Step 2: *Message Fingerprint Extraction:*

1) When new messages arrive, the system determines whether the id is in the ID list. If not, those id messages are an attack signal; otherwise, proceed to the next (lines 11 and 12).
2) ClockIDS uses RLS to calculate the clock skew of id and adds it to a sliding window. At the same time, the earliest element entering the sliding window is deleted (line 15).

Algorithm 1 Intrusion Detection With ER

1: **Initialize:** $\mu[N]$, $\sigma[N]$, n, λ
2: **for** $k^{th}step < n$ **do** /*Step 1*/
3: $L_k^{id} \leftarrow$ the arrival time of id messages.
4: $Skew_{id}.append(RLS(L_k^{id}))$
5: **end for**
6: $\mu_{id} \leftarrow average(Skew_{id})$
7: $\sigma_{id} \leftarrow std(Skew_{id})$
8:
9: **for** $k^{th}step > n$ **do**
10: **for** id appears **do** /*Step 2*/
11: **if** id not exist in $IDlist$ **then**
12: Those messages are abnormal.
13: **else**
14: $L_k^{id} \leftarrow$ the arrival time of id messages.
15: $K_k^{id} \leftarrow RLS(L_k^{id})$
16: $\kappa_k^{id} \leftarrow mean(K_k^{id})$
17: **if** $|\kappa_k^{id} - \mu_{id}|/\sigma_{id} < \lambda_{id}$ **then** /*Step 3*/
18: /* Those massages are normal. */
19: /* Update μ_{id} and σ_{id}. */ /*Step 4*/
20: $Skew_{id}.pop(1)$
21: $Skew_{id}.append(K_k^{id}))$
22: $\mu_{id} \leftarrow average(Skew_{id})$
23: $\sigma_{id} \leftarrow std(Skew_{id})$
24: **else**
25: Messages source ECU identification.
26: **end if**
27: **end if**
28: **end for**
29: **end for**

Algorithm 2 Attack Source Identification With DTW

1: **function** DTW(p, q)
2: $M \leftarrow$ matrix with the size of a ÃŮ b
3: **for** $i < a$ **do**
4: **for** $j < b$ **do**
5: $M[i][j] \leftarrow$ Euler distance between p_i and q_j.
6: **end for**
7: **end for**
8: $path \leftarrow$ shortest path through the matrix by using DP.
9: $d_{sum} \leftarrow 0$.
10: **for** $point(i, j)$ in $path$ **do**
11: $d_{sum} \leftarrow d_{sum} + d(i, j)$.
12: **end for**
13: **return** d_{sum}
14: **end function**
15: /* Step 3 */
16: **for** i in $IDList$ **do**
17: /* x is the ID from ECU A under attack.*/
18: /* i is the ID from any ECU.*/
19: /* Step 1 */
20: $L^x, L^i \leftarrow$ the part of arrival time with same period.
21: $K^x, K^i \leftarrow RLS(L^x), RLS(L^i)$
22: $similarity[i] \leftarrow DTW(K^x, K^i)$ /* Step 2 */
23: **if** $similarity[i]$ is min **then** /* Step 4 */
24: $id \leftarrow i$
25: **end if**
26: **end for**
27: $AttackSource \leftarrow$ the source ECU of id.
28: **return** $AttackSource$

3) The system uses clock offsets in the sliding window to calculate the average clock skew (κ_k^{id}) (line 16).

Step 3: *Fingerprint Comparison:*

1) If $|\kappa_k^{\mathrm{id}} - \mu_{\mathrm{id}}|/\sigma_{\mathrm{id}} < \lambda_{\mathrm{id}}$, those id messages are normal, λ_{id} is the threshold that we set; otherwise, proceed to the next (lines 17 and 18)
2) Calculate min $(\kappa_k^{\mathrm{id}} - \mu_{\mathrm{id}})/\sigma_{\mathrm{id}}$ and find the ECU corresponding to the minimum value. If this ECU is the original one, those messages are normal; otherwise, we will identify the source of the messages (lines 24 and 25).

Step 4: *Fingerprint Update:*

1) Delete the first element in the list $Skew_{\mathrm{id}}$ and fill κ_k^{id} in the list (lines 20 and 21).
2) Calculate the mean and standard deviation of skew again to update μ_i and σ_{id} (lines 22 and 23).

Notes: When selecting λ of each ECU, the range shall be reduced as much as possible to avoid the intersection of reasonable intervals of different ECUs. If the intersection part is generated, it may reduce the detection ability of ClockIDS to intrusion messages. For the part outside the intersection, ClockIDS will use DTW algorithm to detect.

C. Attack Source Identification

The identification of attackers is a part of icing on the cake for IDS, and it make the system more targeted to deal with attackers. As mentioned in the previous section, the clock skew caused by different IDs from the same ECU has a similar trend. We can judge whether the clock skews of the two IDs are similar to determine the source of ECU. We use the DTW algorithm to evaluate the similarity of two ID clock skews. Through this method, we can not only improve the ability of intrusion detection but also identify the ECU of message source.

Algorithm 2 shows the process of using DTW to identify attackers.

Step 1: *Fingerprint Extraction:*

1) The system extracts the part of arrival times for ID x and i with the same interval, and gets two arrival time series L^x and L^i, where ID x is from compromised ECU and ID i is from any ECU (line 19).
2) L^x and L^i are input into the RLS algorithm, respectively, to obtain the instantaneous clock skew sequences (K^x and K^i) of ID^x and ID^i (line 20).

Step 2: *Fingerprint Matching:*
Input K^x and K^i into the DTW algorithm and return the similarity ($similarity[i]$) between them (line 21).

1) ClockIDS first establishes a matrix (M) with the size of $a \times b$ based on the length of the two clock skews p and q (line 2).
2) Next, the system calculates the Euler distance $d(p_i, q_j)$ from each point on p to any point on q, and fills in M (lines 3–7).
3) Then, DP is used to find the shortest path through M (line 8).
4) Finally, we add the values $d(i, j)$ of each point on the path to obtain the total distance (d_{sum}) and return it (lines 9–14). The calculation method of d_{sum} is

$$d_{\text{sum}}(i,j) = \min\{d_{\text{sum}}(i-1,j), d_{\text{sum}}(i-1,j) \\ d_{\text{sum}}(i-1,j-1)\} + M(i,j). \quad (1)$$

Step 3: *Output Similarity Table:* Repeating the above steps to obtain similarity table (*similarity*), which represents the possibility that ID x comes from every ECUs (lines 16–21).

Step 4 *Output Attack Source:* Locate the ID i corresponding to the minimum value in *similarity*. The ECU sending the ID i is the attack source (lines 22, 23, 26, and 27).

V. Experimental Result

In this section, we first introduce our experimental environment. Next, we research the feasibility of using clock skew as fingerprint for ECU. Then, we study the detection of ClockIDS against different types of attacks. Finally, we study the performance of ClockIDS in intrusion detection, attack source identification, and running time.

A. Experimental Setup

Experimental Equipment and Data Frame Collection: Our experimental equipment mainly includes two BYD dashboards, two vehicles, and a data collection tool CANalyst-II. The dashboard is a necessary equipment in every vehicle and contains an ECU. It can truly reflect the ID message sent by ECU in vehicles. Based on this, we chose two BYD dashboards to test and they are named $\text{ECU}_1^{\text{BYD}}$ and $\text{ECU}_2^{\text{BYD}}$, respectively, as shown in Fig. 7(a). In addition, to study whether our method can be applied to real vehicles, we conduct experiments on two real vehicles, Luxgen U5, and Buick Regal, as shown in Fig. 7(c) and (d). In order to obtain the data of CAN bus in the vehicles, we use CANalyst-II to collect data from the OBD-II port, as shown in Fig. 7(b).

Hardware and Software Environments: ClockIDS is developed with Python 3.6 and pycharm community. In terms of hardware, we use a computer with AMD R5 1600X CPU, 8-GB RAM to realize the system.

Construction of Attack Data Set: We directly modified the collected packets to simulate the data traffic on the CAN bus under three different attacks.

1) *Spoofing Attack:* When launching an attack, the attacker injects the message much more frequently than the sending ID of the normal ECU. We choose to modify the more frequent id_2 to the attacked id_1. In other words, we modify the ID field of id_2 messages from ECU_B to id_1 from ECU_A on the collected data sets. After the modification, id_1 is both from the normal ECU A and from the ECU B controlled by the attacker, and the ECU B injects id_1 more frequently. This approach simulates the traffic in the CAN bus under a spoofing attack.
2) *Bus-off Attack:* As described in Section III, when a bus off attack is launched on an ID, the ID will disappear from the CAN bus next time. We construct the attack data sets by removing all id_1 messages sent by ECU_A from the collected data sets. This simulates the scenario of launching a bus-off attack on id_1 of ECU_A via ECU_B.
3) *Masquerade Attack:* Since this attack does not change the ID sequence in the traffic, we use the arrival time of id_2 sent by ECU_B to replace the arrival time of id_1 from ECU_A. Specifically, we first delete all id_1 messages, and then chose to modify the ID field of id_2 message, which has the same frequency as id_1, to id_1. After modification, id_1 originally from ECU_A now comes from ECU_B.

Size of Data Set: For the spoofing attack and the masquerade attack, we launch attacks on the same ID using five different ECUs in two vehicles, thus constructing ten attack data sets, respectively. For the bus-off attack, we launch the attack on five different IDs in two vehicles separately, and obtain ten attack data sets. Every attack data set contains 5000 segments of normal data traffic and attack data traffic, respectively.

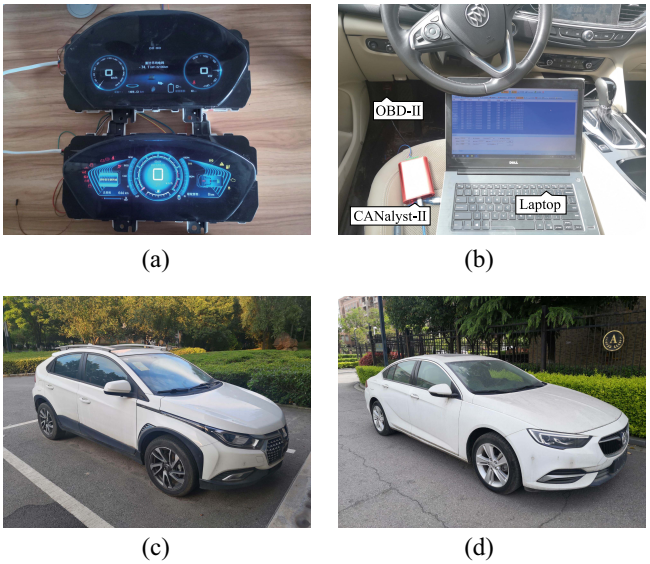


Fig. 7. Experimental vehicles and collection environment. Vehicle devices: two BYD dashboards, Luxgen U5, and Buick Regal. Collection devices: CANalyst-II and laptop. (a) BYD dashboards. (b) Collection environment. (c) Luxgen U5. (d) Buick Regal.

B. Establishment of Clock Skew Fingerprint

To investigate whether ClockIDS can create a unique fingerprint for ECU, we conduct an experiment on two BYD dashboards, Luxgen U5, and Buick Regal.

First, we select one ECU per vehicle, and each ECU can send IDs with different periods, where $\text{ECU}_1^{\text{BYD}}$ and $\text{ECU}_2^{\text{BYD}}$

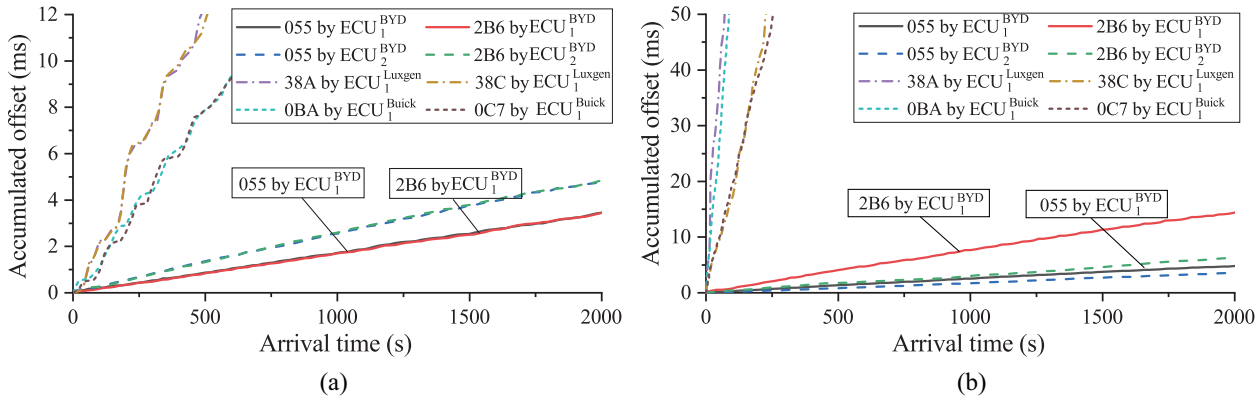


Fig. 8. Accumulated offsets obtained from the evaluation of ClockIDS and CIDS [27] on different automotive ECUs. $T_{055} = 1000.0$ ms, $T_{2B6} = 500.0$ ms, $T_{38A} = 10.0$ ms, $T_{38C} = 20.0$ ms, $T_{0BA} = 25.0$ ms, and $T_{0C7} = 12.5$ ms. (a) ClockIDS. (b) CIDS [27].

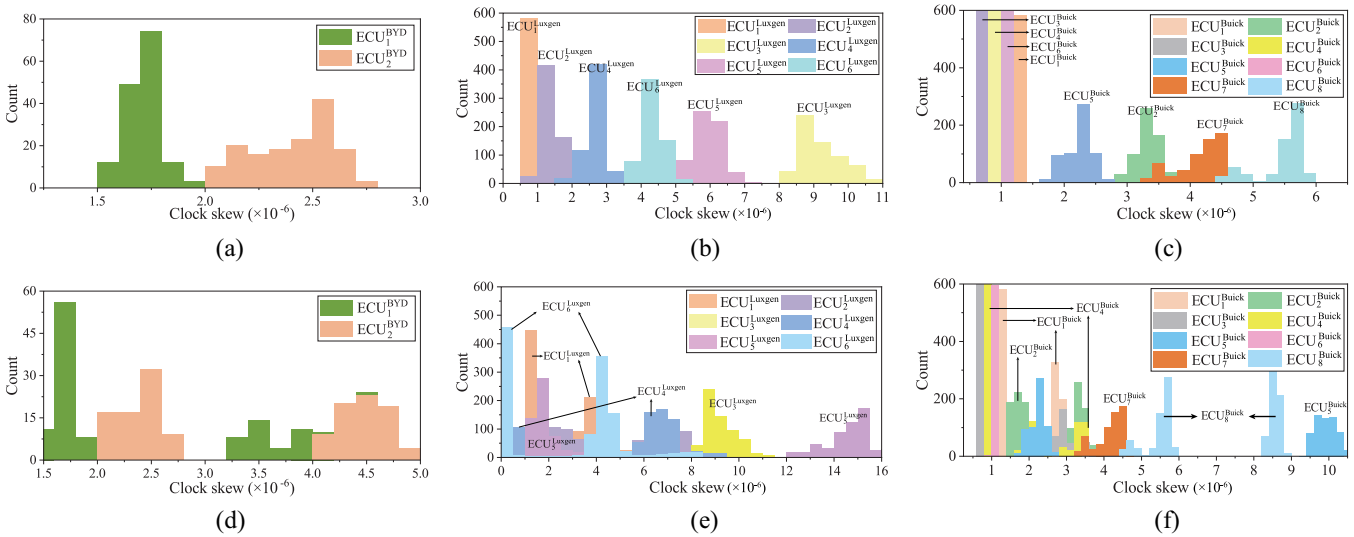


Fig. 9. Distribution of different ECU clock skew in vehicles. (a)–(c) are the result of ClockIDS, and (d)–(f) are the result of CIDS [27]. (a) and (d) are experimental results obtained on the dashboard, (b) and (e) are results on Luxgen U5, and (e) and (f) are results on Buick Regal.

can send ID 055 with $T = 1000$ ms and ID 2B6 with $T = 500$ ms, $\text{ECU}_1^{\text{Luxgen}}$ can send ID 38A with $T = 10$ ms and ID 38C with $T = 20$ ms, and $\text{ECU}_1^{\text{Buick}}$ can send ID 0BA with $T = 25$ ms and ID 0C7 with $T = 12.5$ ms. Then, we use ClockIDS and CIDS [27] to evaluate the clock skew of different IDs, respectively.

Fig. 8 shows the accumulated offsets obtained from the evaluation of ClockIDS and CIDS on different automotive ECUs, respectively. It can be seen from Fig. 8(a) that the clock skew of the same ECU obtained by ClockIDS is similar, even if the periods of IDs are not the same. Fig. 8(b) shows the result of CIDS, and it can be clearly seen that CIDS can be affected by T. In other words, the fingerprint of the same ECU can change with different ID periods. For example, $\text{ECU}_1^{\text{BYD}}$ can send ID 055 messages with $T = 1000.0$ ms and ID 2B6 messages with $T = 500.0$ ms, the clock skews obtained by CIDS for this ECU are $\text{skew}_{055} = 1.75 \times 10^{-6}$ and $\text{skew}_{2B6} = 3.52 \times 10^{-6}$, while the result obtained by ClockIDS is $\text{skew}_{055} = \text{skew}_{2B6} = 1.75 \times 10^{-6}$. The same situation also appears on $\text{ECU}_2^{\text{BYD}}$, $\text{ECU}_1^{\text{Luxgen}}$, and $\text{ECU}_1^{\text{Buick}}$. In addition, we can also see that the clock skew obtained by using ClockIDS is different for different ECUs, such as clock skew of $\text{ECU}_1^{\text{BYD}}$ is 1.75×10^{-6} and $\text{ECU}_2^{\text{BYD}}$ is 2.52×10^{-6}. It is worth noting that the clock skew of $\text{ECU}_1^{\text{Luxgen}}$ and $\text{ECU}_1^{\text{Buick}}$ is not very smooth, but it is stable in the overall skew trend, and the instantaneous clock skew of different IDs of the same ECU is the same, which also makes the clock skew of ECU more unique. Therefore, unlike CIDS, which creates fingerprints for IDs, ClockIDS can create fingerprints for ECUs by calculating clock skew.

Next, to investigate whether ClockIDS can create fingerprints for ECUs under actual operating CAN bus conditions, we investigate the clock skew distribution of different ECUs in the same vehicle environment. In this part, we conduct experiments in BYD, Luxgen U5, and Buick Regal, respectively. In BYD, we use two dashboards as a simple CAN bus environment. In Luxgen and Buick, we find six and eight ECUs, respectively, which can send ID messages with different T. Fig. 9 shows the clock skew distribution of different ECUs in vehicles. On the one hand, Fig. 9(a)–(c) shows the results of ClockIDS. It can be seen that the clock skew distribution of each ECU is stable around one value, whether on

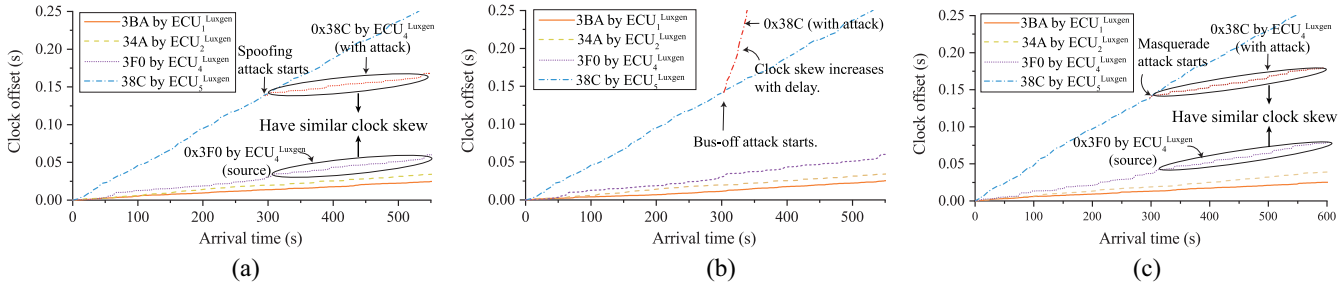


Fig. 10. Accumulated offsets of different IDs under different attacks. In Luxgen U5, ECU_5^{Luxgen} is initially set to send ID $0 \times 38C$ and ECU_4^{Luxgen} is set to send ID $0 \times 3F0$. We use compromised ECU_4^{Luxgen} to launch spoofing attack, bus-off attack, and masquerade attack on ID 38C, respectively. (a) Spoofing attack. (b) Bus-off attack. (c) Masquerade attack.

BYD, Luxgen, or Buick. For example, in the BYD dashboards, the clock skew of the ECU_1^{BYD} is only about 1.75×10^{-6}, and the clock skew of the ECU_2^{BYD} is about 2.5×10^{-6}. Fig. 9(b) and (c) shows that the clock skew of all ECUs in Luxgen and Buick also has a similar situation. It is worth noting that there are a few intersection intervals between different ECUs. For example, in Fig. 9(b), some clock skews of ECU_2^{Luxgen} are the same as those of ECU_1^{Luxgen}. The detection rate of ER algorithm used in this part will decrease, so we add DTW to detect the legitimacy of intersection interval. On the other hand, the CIDS results are shown in Fig. 9(d)–(f), and it is evident that multiple clock skews occur in the same ECU, which is related to ID period. For example, the clock skew of ECU_6^{Luxgen} contains both 0.5×10^{-6} and 4.1×10^{-6} parts, as shown in Fig. 9(e). In contrast, the clock skew of ECU_6^{Luxgen} obtained by ClockIDS is only distributed around 4.2×10^{-6}, as shown in Fig. 9(b). The same case also occurs for ECU_1^{Luxgen}, ECU_2^{Luxgen}, ECU_4^{Luxgen}, and ECU_5^{Luxgen} in Luxgen, and ECU_1^{Buick}, ECU_2^{Buick}, ECU_3^{Buick}, ECU_4^{Buick}, and ECU_8^{Buick} in Buick. In addition, there are some ECUs that are uniquely fingerprinted by using CIDS, such as ECU_3^{Luxgen} and ECU_7^{Buick}, because these ECUs send different IDs with the same T. In conclusion, ClockIDS can create a unique fingerprint for each ECU using clock skew.

C. Detection of Multiple Type Attacks

The above experiments prove that clock skew can be used as the unique fingerprint of ECU. In order to study the feasibility of ClockIDS in detecting multiple types of attack, we make the following experiments in Luxgen U5. Specifically, by modifying the packets collected from the CAN bus, we obtain attack packets of spoofing attack, bus-off attack, and masquerade attack that are launched using ECU_4^{Luxgen} against ID 38C from ECU_5^{Luxgen}, respectively. Next, we use ClockIDS to detect unmodified packets and attack packets, respectively, and compare the clock skew of ECU in them.

Fig. 10 shows the changes of ID clock skew under three different attacks. As can be seen from Fig. 10(a), $skew_{38C}$ hardly changes without being attacked, and the clock offset increases steadily with time. However, when the spoofing attack begins, $skew_{38C}$ changes significantly, and the newly generated $skew_{38C}$ is very similar to $skew_{3F0}$ sent by the attack source ECU_4^{Luxgen}. According to this change, we can judge that ID 38C has been attacked, and find the attack source ECU_4^{Luxgen} according to the matching clock skew. The same situation occurs in the masquerade attack, as shown in Fig. 10(c). This is because the spoofing attack is different from the masquerade attack only in the sending period of attack messages, and ClockIDS will unify the period of attack messages and the period used to establish fingerprint by finding the least common multiple. Therefore, the $skew_{38C}$ carried by the attack messages is the same as $skew_{3F0}$ of the attack source ECU_4^{Luxgen}. In addition, it can be seen from Fig. 10(b) that $skew_{38C}$ also changes significantly under the bus-off attack. Because the 38C messages do not appear in the bus under bus-off attack, and ClockIDS gives 38C messages some late arrival time compared with the expected arrival time. This will make a big change in $skew_{38C}$. Different from the spoofing attack and masquerade attack, $skew_{38C}$ under bus-off attack cannot be matched with other ECUs. In other words, we cannot find the compromised ECU that launched the bus-off attack. Combined with the analysis of the bus off attack model shown in Fig. 6(b), this is because the attacker only interrupts ECU_5^{Luxgen} to send ID 38C by sending a interference signal through ECU_4^{Luxgen}, and does not send ID 38C messages. As a result, ClockIDS cannot receive the message sent by the attacker and extract the fingerprint, so it cannot identify the source of bus-off attacks. All in all, ClockIDS can detect the above three different types of attacks and find the source ECU of the spoofing attack and the masquerade attack, by creating a unique fingerprint for each ECU.

D. Performance of Intrusion Detection

In this section, we study the intrusion detection performance of ClockIDS, and conduct experiments using various types of attack data sets in two vehicles. In the Luxgen, we launch the spoofing attack, bus-off attack, and masquerade attack on ID 316, 34F, 38C, 39A, and 3BA in turn and combine them into three data sets with different attack types. Similarly, in the Buick, we launch attacks on ID 0AA, 0C9, 1ED, 1F4, and 1A1 and construct three attack data sets. Our evaluation indicators include accuracy, precision, false positives rate (FPR), and false negative rate (FNR). The accuracy rate refers to the proportion of correct results predicted by ClockIDS. The precision rate indicates the percentage of samples identified as intrusion that are indeed intrusion. The FPR indicates the

TABLE II
INTRUSION DETECTION PERFORMANCE OF CLOCKIDS IN LUXGEN U5 AND BUICK REGAL

Evaluation index		Spoofing	Bus-off	Masquerade
Accuracy (%) $= \frac{T_P+T_N}{T_P+T_N+F_P+F_N}$	Luxgen	98.63	99.61	99.21
	Buick	99.15	99.84	99.17
Precision (%) $= \frac{T_P}{T_P+F_P}$	Luxgen	99.72	99.89	100.00
	Buick	99.23	100.00	99.69
FPR (%) $= \frac{F_P}{T_N+F_P}$	Luxgen	0.27	0.11	0.00
	Buick	0.77	0.00	0.31
FNR (%) $= \frac{F_N}{T_P+F_N}$	Luxgen	2.47	0.68	1.59
	Buick	0.94	0.32	1.35

percentage of samples that are predicted to be intrusion but actually normal in the total predicted intrusion. The FNR can be interpreted as the percentage of all intrusion classes that are predicted to be normal. Among them, T_P indicates that the normal message is correctly detected, T_N indicates that the intrusion message is correctly detected, F_P indicates that the intrusion message is incorrectly detected as a normal message, and F_N indicates that the normal message is incorrectly detected as an intrusion.

Table II shows the detection performance of ClockIDS for three types of attacks. For the spoofing attack, the accuracy of ClockIDS can reach 99.15%, which means it has a good ability to identify normal messages and intrusion messages. The precision of ClockIDS is 99.72% and 99.23% in two vehicles, respectively, which indicates ClockIDS has good performance on intrusion messages detection. The FPR of ClockIDS is 0.27% in Luxgen and 0.77% in Buick. This indicates that the system has a low probability of recognizing a normal message as an intrusion message. ClockIDS has a very low FNR in both vehicles, which is 2.47% and 0.94%, respectively, indicating that it is difficult to have undetected attack messages. For the bus-off attack, the accuracy can reach 99.84%, the precision can be above 99.89%, the FPR not over 0.11%, and FNR is only 0.68%. For the masquerade attack, ClockIDS has a 99.17% accuracy, 99.69% precision, 0.31% FPR, and 1.35% FNR. It is obvious that ClockIDS has a high detection rate for the three types of attacks. However, it still has errors. This is mainly because the internal clock can be affected by the physical environment, resulting in the change of ECU clock skew. The detection accuracy of ClockIDS for bus-off is generally higher than that of spoofing attacks and masquerade attacks. This is because the clock offset caused by the nonarrival of the message has a great impact on the estimated clock skew, so this attack is easier to detect. At the same time, it can be observed that the detection accuracy of spoofing attacks is not exactly the same as that of masquerade. This is because when an attacker uses the same ECU to send IDs messages with different T, the clock skew will produce small fluctuations due to the existence of clock jitter, so the detection performance will have small deviations. Besides, there are differences in the performance of ClockIDS in different vehicles, but this difference is very small, and its maximum is no more than 0.5%. The reason for this is that the physical environment in which the ECU is located varies from vehicle to vehicle, and the variation in the physical environment is not unique. At the same time, the threshold (λ_{id}) we set for each ECU is different when we use the ER algorithm. Therefore, the performance of ClockIDS is can vary a little in different vehicles.

In summary, for the above three attacks, ClockIDS has good performance in intrusion detection. It has a high detection accuracy and basically does not affect the normal function of vehicles.

TABLE III
CORRESPONDING RELATIONSHIP BETWEEN ECU AND ID IN CAN-BUS

Vehicle	ECU	ID	Vehicle	ECU	ID
Luxgen	ECU_1^L	3BA, 3C5	Buick	ECU_1^B	0AA, 0BE
	ECU_2^L	34A, 34F		ECU_2^B	0C9, 0F9
	ECU_3^L	316, 329		ECU_3^B	1ED, 1EF
	ECU_4^L	39A, 39E		ECU_4^B	1F4, 1F5
	ECU_5^L	38C, 38F		ECU_5^B	1A1, 1A3

TABLE IV
RECOGNITION ACCURACY OF THE INTRUSION SIGNAL IN LUXGEN U5

ECU	ECU_1^L	ECU_2^L	ECU_3^L	ECU_4^L	ECU_5^L
0x316 by ECU_1^L	**99.22**	0	0	0.53	0.25
0x316 by ECU_2^L	0	**94.66**	0.99	4.35	0
0x316 by ECU_3^L	0	0	**100**	0	0
0x316 by ECU_4^L	0	7.02	2.35	**90.63**	0
0x316 by ECU_5^L	0	0	1.48	0	**98.52**

E. Performance of Attack Source Identification

The important premise of using the DTW algorithm to find attack source is to determine which IDs come from the same ECU. As the developer document of the automotive is confidential, it is difficult to obtain the relationship between ECU and ID in vehicles. We use a method in [41], which calculates and compares the average and deviation of ID clock drift. If IDs appear at the same values, they are from the same ECU, otherwise, they are from different ECUs. Table III shows the corresponding relationship between ECU and ID in two vehicles obtained by using the method in [41]. We take this as the verification standard of identification.

Tables IV and V, respectively, show the recognition rate of ClockIDS for different attack sources in two vehicles. It can be seen that our method performs very well in identifying the attack source. In Luxgen, the recognition accuracies of ECU_3^L and ECU_5^L are above 98%, and those of ECU_2^L and ECU_4^L are 94.66% and 90.63%, respectively. In Buick, the recognition accuracies of ECU_2^B and ECU_3^B are above 97%, and ECU_5^B is 92.35%. It is worth noting that ECU_2^L and ECU_4^L are misclassified to each other in a high proportion. The same situation also occurs in ECU_3^B and ECU_4^B. Based on the analysis in

TABLE V
RECOGNITION ACCURACY OF THE INTRUSION SIGNAL IN BUICK REGAL

ECU	ECU_1^B	ECU_2^B	ECU_3^B	ECU_4^B	ECU_5^B
0x0AA by ECU_1^B	**98.43**	1.49	0	0	0.08
0x0AA by ECU_2^B	0.04	**97.54**	0.43	0.47	1.52
0x0AA by ECU_3^B	0.02	0.24	**97.62**	2.12	0
0x0AA by ECU_4^B	0	0.06	1.23	**98.71**	0
0x0AA by ECU_5^B	0	5.64	1.18	0.83	**92.35**

Fig. 9, the two ECUs misclassified with each other usually contain similar clock skews, and even intersect. Because the clock skews of the two are very similar, there is a phenomenon of misclassification. In particular, 5.64% of ECU_5^B was mistakenly identified as ECU_2^B, but ECU_2^B was not mistakenly identified as ECU_5^B. The reason may be that when sending attack messages by ECU_5^B, some messages failed to be sent due to bus arbitration, resulting in that ClockIDS could not accurately extract the fingerprint carried by the attack message. From these data in Tables IV and V, we can see that our method works for each ECU, and the average recognition accuracy on both vehicles is over 96%. Consequently, regardless of the ECU through which the attacker attacks, our method can identify the source of the attack.

F. Running-Time Performance

Since automotive IDSs require high real-time performance, we investigate whether the runtime of ClockIDS can meet the requirement. We count the recognition time of 20 000 times in Buick and Luxgen, respectively, including total running time, average running time, etc.

Table VI shows the running time performance of different algorithms from ClockIDS in two vehicles. First, it can be seen that the total time of detecting 20 000 data in Luxgen and Buick vehicles using the ER algorithm is 6.01 and 6.64 s, respectively, and the average time cost is 0.30 and 0.33 ms, respectively. Second, the average time consumption of using the DTW algorithm to find attack sources is 1.89 and 1.90 ms, respectively. Third, it can be seen from Table VI that the average running time of using ER and DTW algorithms at the same time is 1.98 and 2.01 ms, respectively, in Luxgen and Buick. It can be seen that the time of using only ER algorithm is significantly less than that of other algorithms. Therefore, ClockIDS first uses ER algorithm to detect intrusion and enhances the detection speed, and then use DTW to supplement detection and identification of attack sources.

Besides, as shown in Table VI, the standard deviations of the ER algorithm in the two vehicles are 0.46 and 0.47, respectively. Compared with the average running time, the fluctuation is obvious, but its confidence intervals are (0.00, 1.19) and (0.00, 1.24), respectively. So, it will not cause particularly large overhead for attack detection. The standard deviation of the DTW algorithm on the two vehicles is 0.52. Compared with the average running time of 1.89–1.90, their fluctuation is not obvious. The confidence intervals of DTW are (0.87, 2.91) and (0.88, 2.91) on the two vehicles, respectively. Therefore, it will not take long to find the attack source after the attack. In addition, the standard deviation of ER + DTW is 0.51, whether in Luxgen or Buick. The confidence intervals were (0.97, 2.98) and (1.00, 3.01). From these data, it can be seen that the cost time for ClockIDS to detect intrusion and identify attack source will hardly exceed 3.01 ms. Therefore, ClockIDS only needs a short time to find the attack source ECU.

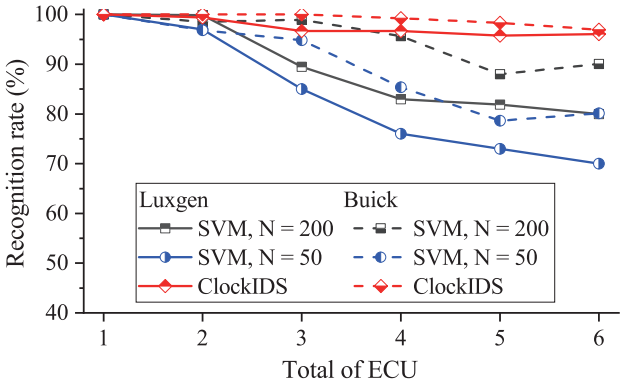


Fig. 11. Relationship between total number of ECUs and recognition rate of attack source in different works. *N* is the number of data frames required for fingerprint extraction in work [42].

G. Algorithm Comparison

1) Comparison of Different IDSs: To study the advantages of ClockIDS in attack source identification, we also compare it with previous work [42]. Since the total number of ECUs in each vehicle varies, as well as the frequency with which attackers may launch attacks, we compared with the work [42] on the attack ratio and the total number of ECUs.

First, we studied the influence of ECU number on recognition rate in the CAN bus with different ECU numbers, and compared ClockIDS with support vector machine (SVM) [42]. Fig. 11 shows our results, where *N* is the number of data frames required by SVM for each detection. On the one hand, it can be seen that the recognition rate of ClockIDS is higher than that of SVM regardless of the number of ECUs, and SVM needs to reach 200 for *N* to get good results. On the other hand, with the increase of the total number of ECUs, the recognition rate of ClockIDS does not change much, while SVM changes greatly. This is because with the increase of the number of ECUs, the probability of similar clock skew of different ECUs increases, resulting in the decline of recognition rate. For example, when the number of ECUs is 5, the effect of SVM decreases significantly in Buick, because the clock skew of the newly added ECU is very similar to that of one ECU in CAN bus. On the contrary, the change of ClockIDS is not obvious. Therefore, ClockIDS has stronger ECU recognition performance than SVM.

Next, we study the relationship between the sending period of attack messages on the recognition rate of ClockIDS and compare it with SVM. Fig. 12 shows the result. It can be seen from the figure that ClockIDS has good attack source recognition performance under different attack frequencies, with an

TABLE VI
PERFORMANCE OF RUNNING TIME OF CLOCKIDS IN TWO VEHICLES. CONFIDENCE COEFFICIENT $1-\alpha=0.95$

Algorithm	Number	Luxgen				Buick			
		Total runnging time (s)	Average runnging time (ms)	Standard deviation (ms)	Confidence interval (ms)	Total runnging time (s)	Average runnging time (ms)	Standard deviation (ms)	Confidence interval (ms)
ER	20000	6.01	0.30	0.46	(0.00, 1.19)	6.64	0.33	0.47	(0.00, 1.24)
DTW	20000	37.83	1.89	0.52	(0.87, 2.91)	37.93	1.90	0.52	(0.88, 2.91)
ER+DTW	20000	39.63	1.98	0.51	(0.97, 2.98)	40.14	2.01	0.51	(1.00, 3.01)

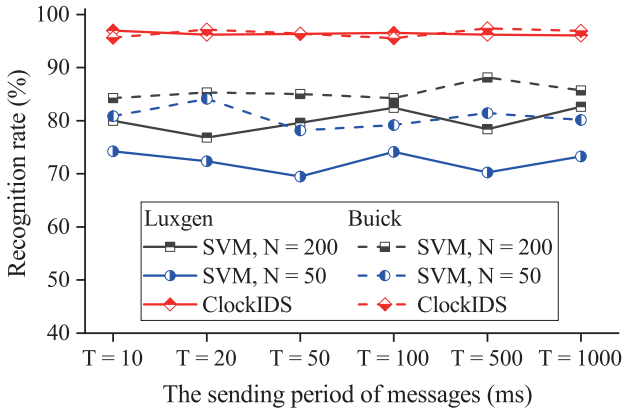


Fig. 12. Relationship between the sending period of messages and recognition rate of attack source in different works. T is the period of attack data frames. N is the number of data frames required for fingerprint extraction in work [42].

TABLE VII
RECOGNITION RATE AND AVERAGE RUNNING TIME OF DIFFERENT METHODS

Algorithm	Total numbers	Identification rate (%)	Average running time (ms)
DP	20000	96.77	2.10
Dijkstra	20000	91.82	0.81

average recognition rate of more than 96% and little fluctuation. In addition, SVM only achieves good results when $N = 200$. The average recognition rates on Luxgen and Buick are 79.97% and 85.45%, respectively. All in all, ClockIDS is not affected by the attack ratio, and the attack source recognition rate is higher than SVM.

2) Shortest Path Algorithm of DTW: To improve the performance of the DTW algorithm, we also try two methods to find the shortest path, including the DP algorithm and Dijkstra algorithm. Table VII shows the comparison of the recognition rate and average running time of the two algorithms. It can be seen that the recognition rate of ClockIDS using the DP algorithm is 96.77%, which is about 5% higher than that of ClockIDS using the Dijkstra algorithm. This is because the Dijkstra algorithm cannot traverse all points in the matrix, so it can only find the local optimal solution. In addition, the average running time of the DP algorithm is 2.10 ms, which is higher than 0.81 ms of the Dijkstra algorithm. In order to pursue a higher recognition rate, we choose to use the DP algorithm to find the shortest path.

VI. DISCUSSION

In this section, we discuss the limitations of ClockIDS and our future research.

A. Limitations of ClockIDS

1) Physical Port Protection: In this article, we only consider the scenario of launching an attack through an internal ECU, and the attacker can also directly access an additional ECU from the physical port to launch attacks. Our method can also be extended to the detection of external attacks. On the one hand, we can establish the distribution range of all normal ECUs in the vehicle. When the fingerprint of the new message is not within the distribution range, it is determined as an external ECU intrusion. On the other hand, we can also set a threshold when using the DTW algorithm. If the similarity does not reach the threshold, it can also be determined as external ECU intrusion.

2) Parameter Selection: ClockIDS needs to manually analyze and select the parameters of each ECU. If the parameters are not selected properly, the overlap of skew distribution between different ECUs may become a potential threat. Specifically, if the normal range of clock skew given to ECU_A is too large, resulting in an intersection with ECU_B, the attacker may imperceptibly modify the fingerprint of ECU A by injecting the ID of ECU_A through ECU_B. In addition, when replacing the vehicle platform, we need to readjust the parameters of ClockIDS, which is more suitable for manufacturers to integrate in the vehicle or upgrade the firmware of existing vehicles.

3) Aperiodic Problem: We use periodic ID to establish a unique fingerprint for the corresponding ECU. However, there are aperiodic IDs on the CAN bus. Our method cannot evaluate the clock skew of aperiodic ID. Besides, the intrusion detection of ClockIDS is not affected by the size of period, but it is not perfect in identifying the attack source. Although ClockIDS can identify the attack sources of different injection periodic attacks, it cannot identify the attack sources of aperiodic injection attacks.

B. Future Research

1) Aperiodic Fingerprint: The biggest problem of clock skew-based IDS is that it cannot detect aperiodic IDs. To solve this problem, voltage-based methods [23]–[25], [43], [44] are proposed in the community, but these works also have their

own shortcomings. For example, it is difficult to update fingerprints on devices with less resources. Therefore, the integration of clock skew-based and voltage-based IDS will be a more interesting direction in the future.

2) Identification Attack Source of Bus-Off: Using clock skew fingerprint to find the compromised ECU of bus-off attack is also an important and promising problem. In case of bus-off attacks, although the ID message does not appear on the CAN bus, an interference signal will appear. If we can accurately collect the transmission time of the interference signal, we may still extract the fingerprint and trace the source of the attack.

3) Intrusion Prevention System: In addition, building an intrusion prevention system is also a very important work. It can deal with the attack after detecting the intrusion, so as to make attacks invalid or slow down the attack. This may use the arbitration mechanism of CAN bus. When IDS detects an intrusion message, it needs to send an interference signal to make the attack message fail.

VII. Conclusion

In this article, we have designed a real-time IDS, called ClockIDS, using the clock skew of ECU. It can be directly connected to the CAN bus as a monitoring unit without modifying the existing bus. The experimental results on two real cars show that ClockIDS has a good detection rate for many types of attacks, such as spoofing, bus-off, and masquerade attacks. Meanwhile, it can also trace the source of attack for periodic injection attacks. In addition, ClockIDS has high real-time performance to detect intrusion messages. ClockIDS can be directly accessed and operated from the interface in the vehicle as an external device. Therefore, ClockIDS can significantly improve vehicle security without changing the existing vehicle network environment.

References

[1] B. Mao, F. Tang, Y. Kawamoto, and N. Kato, "Optimizing computation offloading in satellite-UAV-served 6G IoT: A deep learning approach," *IEEE Netw.*, vol. 35, no. 4, pp. 102–108, Jul./Aug. 2021.

[2] "Upstream Security's 2021 Global Automotive Cybersecurity Report." 2020. [Online]. Available: https://upstream.auto/2021Report/

[3] B. Mao, F. Tang, Z. M. Fadlullah, and N. Kato, "An intelligent route computation approach based on real-time deep learning strategy for software defined communication systems," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1554–1565, Jul.–Sep. 2021.

[4] Y. Cui, L. Du, H. Wang, D. Wu, and R. Wang, "Reinforcement learning for joint optimization of communication and computation in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13062–13072, Dec. 2021.

[5] Q. Yang, S. Fu, H. Wang, and H. Fang, "Machine-learning-enabled cooperative perception for connected autonomous vehicles: Challenges and opportunities," *IEEE Netw.*, vol. 35, no. 3, pp. 96–101, May/Jun. 2021.

[6] M. Albanese, E. Battista, and S. Jajodia, "Deceiving attackers by creating a virtual attack surface," in *Cyber Deception*. Cham, Switzerland: Springer Int., 2016, pp. 167–199.

[7] M. Albanese, E. Battista, S. Jajodia, and V. Casola, "Manipulating the attacker's view of a system's attack surface," in *Proc. IEEE Conf. Commun. Netw. Security*, San Francisco, CA, USA, 2014, pp. 472–480.

[8] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," in *Proc. Black Hat USA*, 2015, pp. 1–91.

[9] S. Nie, L. Liu, and Y. Du, "Free-fall: Hacking tesla from wireless to CAN bus," in *Proc. Briefing Black Hat USA*, 2017, pp. 1–16.

[10] "Upstream Security's 2020 Global Automotive Cybersecurity Report." [Online]. Available: https://upstream.auto/upstream-security-global-automotive-cybersecurity-report-2020/ (accessed Feb. 2020).

[11] C. Patsakis, K. Dellios, and M. Bouroche, "Towards a distributed secure in-vehicle communication architecture for modern vehicles," *Comput. Security*, vol. 40, pp. 60–74, Feb. 2014.

[12] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "CANAuth—A simple, backward compatible broadcast authentication protocol for CAN bus," in *Proc. ECRYPT Workshop Lightweight Cryptogr.*, 2011, p. 20.

[13] P. Mundhenk *et al.*, "Security in automotive networks: Lightweight authentication and authorization," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 22, no. 2, pp. 1–27, 2017.

[14] A.-I. Radu and F. D. Garcia, "LeiA: A lightweight authentication protocol for CAN," in *Proc. Eur. Symp. Res. Comput. Security*, 2016, pp. 283–300.

[15] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures," in *Proc. Int. Conf. Comput. Safety Rel. Security*, 2008, pp. 235–248.

[16] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive CAN bus," in *Proc. World Congr. Ind. Control Syst. Security (WCICSS)*, London, U.K., 2015, pp. 45–49.

[17] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Montreal, QC, Canada, 2016, pp. 130–139.

[18] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Baden-Baden, Germany, 2011, pp. 1110–1115.

[19] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in *Proc. IEEE 2nd Int. Forum Res. Technol. Soc. Ind. Leveraging Better Tomorrow (RTSI)*, Bologna, Italy, 2016, pp. 1–6.

[20] M.-J. Kang and J.-W. Kang, "A novel intrusion detection method using deep neural network for in-vehicle network security," in *Proc. IEEE 83rd Veh. Technol. Conf. (VTC Spring)*, Nanjing, China, 2016, pp. 1–5.

[21] D. Stabili, M. Marchetti, and M. Colajanni, "Detecting attacks to internal vehicle networks through hamming distance," in *Proc. AEIT Int. Annu. Conf.*, Cagliari, Italy, 2017, pp. 1–6.

[22] H. Lee, S. H. Jeong, and H. K. Kim, "OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame," in *Proc. 15th Annu. Conf. Privacy Security Trust (PST)*, Calgary, AB, Canada, 2017, pp. 57–66.

[23] K.-T. Cho and K. G. Shin, "Viden: Attacker identification on in-vehicle networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 1109–1123.

[24] 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, pp. 2114–2129, 2018.

[25] M. Kneib and C. Huth, "Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2018, pp. 787–800.

[26] 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, pp. 2300–2314, 2019.

[27] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *Proc. 25th USENIX Security Symp. (USENIX Security)*, 2016, pp. 911–927.

[28] S. Mohalik *et al.*, "Model checking based analysis of end-to-end latency in embedded, real-time systems with clock drifts," in *Proc. 45th Annu. Des. Autom. Conf.*, Anaheim, CA, USA, 2008, pp. 296–299.

[29] S. Zander and S. J. Murdoch, "An improved clock-skew measurement technique for revealing hidden services," in *Proc. USENIX Security Symp.*, 2008, pp. 211–225.

[30] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *IEEE Trans. Dependable Secure Comput.*, vol. 2, no. 2, pp. 93–108, Apr.–Jun. 2005.

[31] T. Rakthanmanon *et al.*, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2012, pp. 262–270.

[32] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. KDD Workshop*, vol. 10. Seattle, WA, USA, 1994, pp. 359–370.

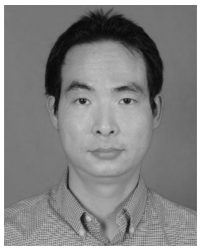
[33] H.-R. Choi and T.-Y. Kim, "Directional dynamic time warping for gesture recognition," in *Proc. 2nd Int. Conf. Multimedia Syst. Signal Process.*, 2017, pp. 22–25.
[34] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, and E. Keogh, "Generalizing DTW to the multi-dimensional case requires an adaptive approach," *Data Min. Knowl. Discov.*, vol. 31, no. 1, pp. 1–31, 2017.
[35] S. Checkoway *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. USENIX Security Symp.*, San Francisco, CA, USA, 2011, p. 6.
[36] C. Miller and C. Valasek, "Adventures in automotive networks and control units," in *Proc. DEFCON*, 2013, pp. 260–264.
[37] A. Greenberg, *Hackers Remotely Kill a Jeep on the Highway—With Me in it*, vol. 7, Wired, New York, NY, USA, 2015, p. 21.
[38] K.-T. Cho and K. G. Shin, "Error handling of in-vehicle networks makes them vulnerable," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2016, pp. 1044–1055.
[39] E. Evenchick, "Hopping on the CAN bus: Automotive security and the CANard toolkit," in *Proc. Black Hat Asia*, 2015, pp. 1–31.
[40] S. Jana and S. K. Kasera, "On fast and accurate detection of unauthorized wireless access points using clock skews," *IEEE Trans. Mobile Comput.*, vol. 9, no. 3, pp. 449–462, Mar. 2010.
[41] 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.
[42] J. Zhou, G. Xie, S. Yu, and R. Li, "Clock-based sender identification and attack detection for automotive CAN network," *IEEE Access*, vol. 9, pp. 2665–2679, 2020.
[43] J. Ning and J. Liu, "An experimental study towards attacker identification in automotive networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, 2019, pp. 1–6.
[44] Y. Xun, Y. Zhao, and J. Liu, "VehicleEIDS: A novel external intrusion detection system based on vehicle voltage signals," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 2124–2133, Feb. 2022.

Yilin Zhao (Graduate Student Member, IEEE) received the B.S. degree in computer science and technology from Qingdao Agricultural University, Qingdao, China, in 2019.

His research interests cover vehicular network security and machine learning.

Yijie Xun (Member, IEEE) received the B.S. degree from Shanxi University, Taiyuan, Shanxi, China, in 2016, and the Ph.D. degree from Xidian University, Xi'an, Shannxi, China, in 2021.

He is currently an Associate Professor with the School of Cybersecurity, Northwestern Polytechnical University, Xi'an. His research interests include vehicular network security and machine learning.



Jiajia Liu (Senior Member, IEEE) received the B.S. degree in computer science from Harbin Institute of Technology, Harbin, China, in 2004, the M.S. degree in computer science from Xidian University, Xi'an, China, in 2009, and the Ph.D. degree in information sciences from Tohoku University, Sendai, Japan, in 2012.

He is a Full Professor (Vice Dean) with the School of Cybersecurity, Northwestern Polytechnical University, Xi'an, China. He has published more than 180 peer-reviewed papers in many high-quality publications, including prestigious IEEE journals and conferences. His research interests cover a wide range of areas, including intelligent and connected vehicles, mobile/edge/cloud computing and storage, Internet of Things security, wireless and mobile ad hoc networks, and space–air–ground integrated networks.

Prof. Liu received the IEEE VTS Early Career Award in 2019, the IEEE ComSoc Asia–Pacific Outstanding Young Researcher Award in 2017, the IEEE ComSoc Asia–Pacific Outstanding Paper Award in 2019, the Niwa Yasujiro Outstanding Paper Award in 2012, the Best Paper Awards from many international conferences, including IEEE flagship events, such as IEEE GLOBECOM in 2016 and 2019, IEEE WCNC in 2012 and 2014, IEEE WiMob in 2019, and IEEE IC-NIDC in 2018. He was a recipient of the Tohoku University President Award 2013. He has been actively joining the society activities, like serving as an Associate Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS since May 2018, IEEE TRANSACTIONS ON COMPUTERS from October 2015 to June 2017 and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY since January 2016, an Editor for IEEE NETWORK since July 2015 and IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING since January 2019, a Guest Editor of top ranking international journals, such as IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, *IEEE Vehicular Technology Magazine*, and IEEE INTERNET OF THINGS JOURNAL, and serving as the technical program committees of numerous international conferences, such as the Leading Symposium Co-Chair of AHSN Symposium for GLOBECOM 2017, CRN Symposium for ICC 2018, and AHSN Symposium for ICC 2019. He is the Chair of IEEE IOT-AHSN TC and a Distinguished Lecturer of IEEE Communications Society and Vehicular Technology Society.