DOI:

https://doi.org/10.1007/s10586-022-03821-x

Mitigating Voltage Fingerprint Spoofing Attacks on the Controller Area Network Bus

Wassila Lalouani
Department of Computer and Information Science
Towson University
Towson, Maryland, USA
wlalouani@towson.edu

Yi Dang, and Mohamed Younis
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, Maryland, USA
{dangyi1, younis}@umbc.edu

***Abstract*— The Controller Area Network (CAN) bus suffers security vulnerabilities that allow message spoofing and masquerading Electronic Control Units (ECUs). A popular provision for mitigating these vulnerabilities is through the use of machine learning (ML) to derive ECU fingerprints based on the physical properties of bus signals. Particularly, voltage-based intrusion detection systems associate the message transmitter with its voltage fingerprint to detect conflicting logical ECU identifiers in the presence of cyberattacks. However, the signal characteristics depend on the operating conditions and hence the fingerprints need to be adapted overtime by online training of the underlying ML model. An adversary may exploit such a shortcoming to superimpose training data based on its own transmissions and thus bypass the protection mechanism. Such an attack not only allows device impersonation but also leads to rejecting transmissions of a legitimate ECU. This paper proposes an effective approach to thwart these attack scenarios. Our approach introduces unpredictably-scheduled transmissions involving one or multiple ECUs to confuse the adversary and ensure the generation of a legitimate fingerprinting dataset for online training. We validate the robustness of our approach using data collected from a real vehicle and show that it outperforms a prominent competing scheme by over 30% in terms of identifying malicious ECUs when the attacker could overwrite 50% of the retraining transmissions.**

***Keywords*—Cyber physical systems, CAN bus, security, fingerprinting, device authentication, impersonation attack.**

I. INTRODUCTION

A cyber-physical system (CPS) reflects the realization of a control system using a networked set of modules that operate in a distributed manner. The CAN bus is commonly used for interconnecting ECUs within a CPS. In particular, the CAN bus is the de facto standard for the automotive industry, where ECUs represent embedded automotive systems like engine control, anti-lock braking, traction control, etc. [1]-[3]. Inter-ECU communication over the CAN bus allows coordinated operation that directly impacts on-road safety, maintainability, and fuel consumption. In fact, with the increased adaptation of autonomous vehicles, the criticality of the CAN bus becomes even more prominent given the disengagement of drivers when making decisions. Not surprisingly, such criticality elevates the importance of protecting the CAN bus against cyberattacks.

The security analysis of the CAN bus has revealed that it is vulnerable to ECU impersonation and message spoofing attacks [4]-[13]. The broadcast nature of the CAN communication protocol, and the lack of strong authentication and data integrity mechanism prevent the detection of these serious attacks. An attacker can masquerade an ECU and transmit malicious data or control information that may lead a CPS to take inappropriate decisions causing serious consequences. Although a CAN frame can be encrypted to ensure authentication and integrity using message authentication codes (MACs) or digital signatures, this imposes excessive computational overhead and is not suited for the resource-constrained embedded CPS modules. The most prominent defense strategy exploits the physical characteristics of the ECU signals to define a unique device fingerprint that serves as a means for identity verification [14][15]. Machine learning techniques are often used for devising the fingerprints based on collected measurements during system operation. Popular fingerprinting methodologies include clock variation, signal voltage, and message traffic based features [16]-[21]. The fingerprints are then used by intrusion detection systems (IDS) to classify legitimate and malicious transmissions. Voltage-based schemes are deemed to be the most robust IDS for CAN buses [18]-[20].

Although device fingerprinting enables frame-level ECU authentication and is lightweight in terms of computational overhead, it has been shown that the adversary may associate its own fingerprint with legitimate ECUs [22][23]. While voltage-based fingerprints can be defined offline before the system deployment, the signal characteristics are found to vary based on environmental conditions, e.g., changes in temperature, and hence the fingerprint needs to be adjusted through online training of the underlying machine learning model [24]. The online training period allows the attacker to interfere with the measurement collection and gradually transforms the fingerprint to a version that reflects such an attacker rather than the legitimate ECU. The attacker takes advantage of the predictability (periodicity) of the ECU transmissions, which is typical for real-time control systems. The impersonation attack can be launched to delegitimize the messages transmitted by a victim ECU or just masquerading such an ECU to inject faulty data. It has been shown in [18][20], that existing voltage-based IDS fail to mitigate such impersonation attacks.

To overcome the aforementioned shortcoming, this paper promotes a novel approach for Mitigating Attempts to Spoof

voltage-based fingerprints (MAS). The basic idea of MAS is to expose the attacker by introducing: (i) unpredictably scheduled transmissions, and (ii) unconventional fingerprints by triggering simultaneous transmissions of more than one ECU. The objective is to generate a clean online fingerprinting dataset based on individual and combined transmissions of multiple ECUs. By provisioning for transmissions that an attacker cannot predict their senders, MAS enables the collection of uncorrupted measurements that can be used for online training. The combined transmissions are meant to confuse the adversary about which ECU is supposed to transmit and also allows the detection of interference attempts based on the increased voltage level. MAS not only can detect fingerprint poisoning (hijacking) attacks, but also can flag suspicious message headers without dropping legitimate data frames. Our main contributions are summarized as follows:

- We point out a new attack on the CAN bus where a malicious ECU can interfere with the frame header to change the voltage fingerprint and cause the rejection of messages of legitimate nodes.
- We propose MAS, a novel mechanism for detecting and mitigating malicious attempts to impersonate or blacklist legitimate CAN nodes by falsifying their voltage fingerprints.
- We validate the effectiveness of our MAS mechanism using a dataset collected from ten automotive-grade ECUs.

The reminder of the paper is organized as follows. The next section discusses the related work. Section III discusses the system and problem models. Section IV presents our approach. Section V reports the validation results. Finally, the paper is concluded in section VI.

II. Related Work

The CAN-Bus has been widely utilized in automotive systems, domestic appliances, and medical devices. A CAN-Bus message does not have origin and destination addresses; instead it reflects a broadcast so that each ECU can send and receive packets to/from the bus. This communication technique increases the network elasticity which means that if a new ECU is to be added to the network, it will be configured easily and does not require any changes to the network infrastructure and other nodes. However, the simplicity and agility of a CAN-Bus also increases the risk of intrusion [25]. To address this problem, quite a few IDS have been developed to protect the security of the CAN-Bus network. The underlying principle of these IDS is to characterize the normal operation involving legit nodes. By using such a characterization as a fingerprint malicious behavior could be detected [26]-[32].

Existing IDS techniques for the CAN-Bus vary in terms of the features that are used for fingerprint generation, and can generally be classified as (i) Entropy-based IDS (EIDS), (ii) Clock-based IDS (CIDS), and Voltage-based IDS (VIDS). EIDS an information theoretic based anomaly detection approach. An example of EIDS is the work of Müter and Asaj [16] where the data recorded from the in-vehicle network during normal operation is used to calculate the Shannon entropy. Deviations from such a baseline entropy are flagged as potential intrusions. Hence, EIDS is a system-wide assessment methodology and does not identify the malicious nodes given the broadcast nature of the CAN-bus [35]. On the other hand, CIDS defines fingerprints based on the clock that usually exists in all digital systems. Specifically, the uniqueness of the clock skew and clock offset of a given ECU is exploited to identify the source of a CAN-Bus transmission [17]. Cho and Shin [2], and Young et al. [26] proposed methods that leverage the periodic behavior of CAN-Bus messages to fingerprint each ECU in the network. Messaging rate is used by Koyama et al. [30] to detect malicious traffic, and by Longari et al. [31] to thwart attacks that opt to disconnect an ECU from the bus by interfering with its transmission (and hence enable spoofing attacks). In [32] anomalies are detected based on deviation from a pre-specified timing behavior, while Bi et al. [33] consider both the data and time characteristics of bus frames to classify abnormal ECU activities. Despite being superior to EIDS in terms of ability to fingerprint the individual ECUs, CIDS is not suitable when ECU's transmission pattern cannot be characterized.

Like CIDS, VIDS relies on device-level fingerprinting. Material and design imperfections of an ECU make the voltage on the bus exhibit some unique features for each transmitter. VIDS has higher accuracy and is easier to apply than EIDS and CIDS. Hence, quite a few VIDS-based techniques have been developed, such as Viden [18], voltageIDS [20], and Scission [21]. All these techniques extract features of the ECU's voltage and then apply machine learning to define a model that serves as a fingerprint. The extracted features vary. Viden focuses on the control frames to learn voltage features of target ECU. It excludes the ACK phase of each data frame. Scission opts to reduce the sampling rate and the considered features by using rising edges, falling edges and time between consecutive dominant voltages (logic zero). Since the voltage profile changes with the environment and operating conditions (e.g., temperature and supply voltage), the machine learning model must get updated and retrained periodically. VIDS is effective against a single-actor based denial of service or spoofing attacks that opt to force the victim into the bus-off mode [34].

Some work has also considered attacks that defeat VIDS. SIMPLE [22] applies a poisoning attack which could fool VIDS that uses multi-frame training data by injecting messages without raising suspicion. The attack targets the retraining data of a legit ECU, where malicious messages are injected in a gradually increasing frequency in order to transform the fingerprint of such a legit ECU to that of the attacker node. Such an attack is referred to as hill-climbing. SIMPLE also offers a method to overcome such a stealthy attack by extracting features from data frames. DUET [23] is another attack model, which makes the attacker transmit messages with target ECU simultaneously, so that it can poison the training data of the fingerprint model. The underlying assumption for SIMPLE and DUET is that the attacker can predict when a victim node will transmit. MAS overcomes these poisoning attacks by introducing unpredictability not only at the level of the individual transmitters but also on their multiplicity. While RAID also tries to introduce unpredictability to counter DUET,

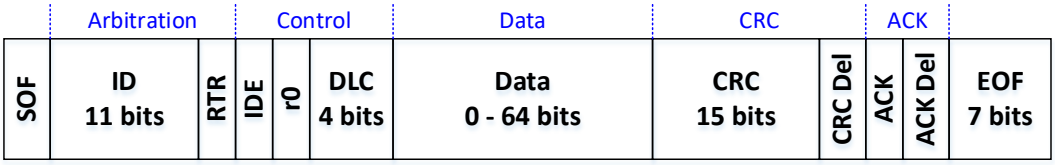

Fig. 1: The structure of a data frame in CAN-Bus, figure redrawn based on [36].

it does so by randomly changing the message ID [23]. In addition to being hard to provision, RAID also changes the bus access priority and hence risks the timeliness properties of the CPS. To the best of our knowledge, MAS is the only countermeasure for training data poisoning of VIDS that not only detects the attack but also identifies the attacker node.

III. System Model and Preliminaries

A. Background

The CAN-Bus is a multi-master event-triggered system, where a message can be transmitted by a node without the need of a predefined schedule. There are four frame types, namely, data, remote, overload, and error [16]. Error and overload frames are for dealing with errors on the CAN-Bus. A remote frame is used to request data from a specific ECU. The data frame is the most commonly used and carries the data from a transmitter to a receiver. Fig. 1 shows the CAN-Bus data frame structure. A data frame consists of the following bit fields: (i) start of frame (SOF), which reflects one dominant bit (logic zero), (i) an arbitration field which consists of 12 bits, (iii) a control field which has 6 bits, (iv) a data field that ranges from 0 to 64 bytes, (v) Cyclic Redundancy Check (CRC) for error detection, (vi) ACK field (2 bits), and (vii) an end of frame (7 bits). The RTR bit, within the arbitration field, defines whether this frame is a data frame or a remote frame. The data field can be in the length of zero (remote frame) to eight bytes and the control field specifies the length of the data frame.

The CAN-Bus standard employs a simple arbitration procedure to prevent any two ECUs from concurrently transmitting their message frames on the bus. Each message is assigned an identifier, which is utilized to define its priority. The lower the identifier value is the higher priority a message has. Thus, the ECU which has a lower ID will access the bus while other contending nodes will terminate their transmission and wait for the bus to be free. The ID comparison is conducted bit-by-bit. To transmit a message, an ECU first synchronizes with the SOF field. For the ID field, each ECU sends an ID bit and then reads it back from the bus. If an ECU, ξ_i, reads a dominant bit (zero value) after it transmits a recessive bit (value of one), it concludes that a higher priority ECU is trying to access the bus. In such a case, ξ_i loses the arbitration and stops transmitting; otherwise, ξ_i continues to transmit the message.

All nodes on a CAN-bus will send ACK when they receive a data message, regardless of the intended recipient of the message. If any node transmits '0' as an ACK, the bus will show '0', and consequently the sender concludes successful delivery and does not retransmit. Every ECU has two error counters: Transmit Error Counter (TEC) and Receive Error Counter (REC). When an error is experienced during transmission, the sender's TEC is increased by 8, and the REC of other nodes on the bus (receivers) are increased by 1. When an error is detected by a receiver, its REC grows by 8. When the message is correctly transmitted, both TEC and REC of the sender and receiver nodes are decremented by 1. A node's state is defined by its TEC and REC. There are three states in CAN-Bus: error active, error passive, and bus off. In the latter, the node cannot transmit/receive any frames and is disconnected from the bus.

B. System and Adversary Models

We assume a CPS that consists of multiple ECUs interconnected through a CAN-Bus. To support authentication of the source of a message transmission on the bus, a voltage-based IDS is employed. The IDS is in essence network-based where the bus traffic is monitored at the physical-layer and the fingerprints of ECUs are validated using machine learning. The IDS can be applied by a certain ECU or by all ECUs. The latter scenario is captured in Fig. 2, where each ECU, ξ_i, maintains a machine learning model that references the voltage profile of each other ECU, ξ_s, when ξ_s transmits on the bus. Given the real-time control aspect of a CPS, the system realizes a closed loop where sensor data is periodically collected, processed and acted upon within certain time constraints. Hence, the bus access profile of an ECU is periodic where at the system design time each ECU is allocated sufficient bandwidth (access time).

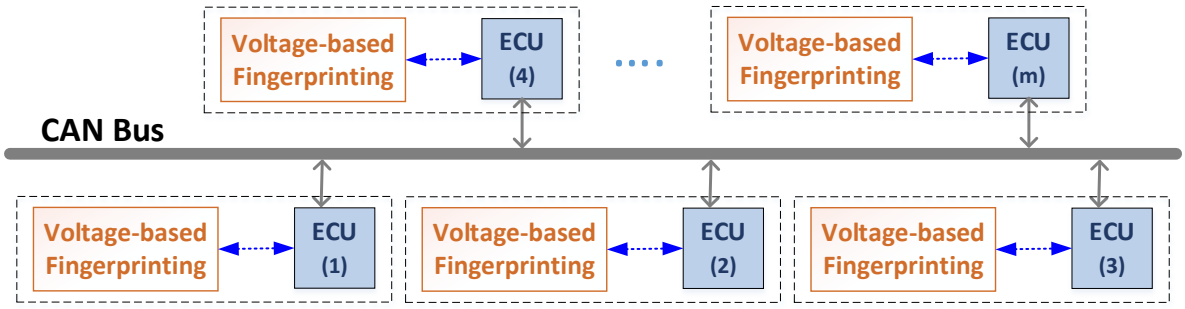

Fig. 2. The assumed CAN-Bus based CPS architecture.

To ensure system stability, the bus capacity usually exceeds the collective demand of all ECUs. MAS exploits such property in mitigating cyberattacks, as explained in Section V.

To attack the considered CPS, an adversary opts to destabilize the system by injecting false data, depriving the ECUs from getting data, or disrupting the coordination among the ECUs. To achieve these goals, the adversary strives to implant malicious ECUs that can be exploited in launching attacks. For that one of two approaches could be pursued [37]. The first is to install malicious ECUs on the CAN-Bus or change the firmware of existing ECUs; such a scenario could be feasible by getting physical access to the system, e.g., when an electrical vehicle is being serviced at a repair shop. The second scenario reflects remote access to the CPS, where a legit ECU is hacked and loaded with a software to implement the desired attacks [38]. In essence, the adversary will exploit the interface of a CPS to access ECUs, e.g., through Vehicle-to-Vehicle or Vehicle-to-Infrastructure communications. The adversary is assumed to be aware of the voltage fingerprinting based IDS employed by the CPS.

IV. Voltage Fingerprint Falsification Attacks

Variations in the signal voltage are found to be distinct among the CAN-Bus transceivers and hence can serve as physical-layer based fingerprints to distinguish among the connected devices. The features capturing the statistical and physical characteristics of signals are employed by the IDS to determine any impersonation or falsification in data packets. Such a fingerprinting mechanism is deemed to be very robust. Yet, a recent study by Bhatia et al. [23] has pointed out the vulnerability of voltage-based IDS to impersonation attacks, as summarized below. We also point out an additional voltage fingerprint falsification attack that is geared for blacklisting legit nodes. MAS tackles both attacks as detailed in Section V.

A. ECU Impersonation Attack

The voltage profile of an ECU usually varies due to electronics aging and changes in the ambient conditions, e.g., temperature. Therefore, it is necessary to retrain the machine learning model to update the ECU fingerprint and avoid false positive and false negative classifications by the IDS. Such retaining is conducted while the system is in operation and constitutes a vulnerability. Basically, poisoned data could be injected by corrupting the voltage measurements, if an attacker can determine the retraining time. DUET [23] constitutes a realization of such poisoning scenario involving a pair of malicious ECUs; through coordinated actions the malicious ECU pair can get the IDS to use wrong voltage measurements for updating the fingerprint of a victim node. The key condition for the attack to succeed is the ability of the adversary to predict: (i) when a victim ECU transmits, and (2) when retraining takes place. The former is facilitated by the periodic nature of real-time tasks where an ECU transmits at a constant rate. The retraining time can vary from one CPS to another and hence it is application dependent. For example, in a vehicle retraining could be conducted after startup or when the engine temperature reaches a certain level. The main idea of the attack is to get a malicious ECU to simultaneously transmit along with the victim, causing the observed voltage on the bus to deviate from that when only the victim ECU transmits.

Fig. 3(a) shows the attack steps where one of the malicious ECU acts as a helper by sending a message with a preceded ID in order to force the victim to wait. Such a step is meant to allow the second malicious ECU (threat actor) to synchronize its transmission with the victim, realizing that jitters could be experienced and the exact time for a victim transmission could slightly vary from one period to another. The actor will then transmit along with the victim and hence the voltage will reflect two rather than one transmission. By using the voltage of the combined transmission for training, the fingerprint of the victim will be transformed to a one that reflects "victim & actor". The threat actor needs to be in the passive error state so that its transmission reflects an error frame which does not interfere with the victim's transmission from a bus control point of view, i.e., will not cause the victim ECU to back off. Such an error passive state can be initially reached by a simultaneous transmission by the helper and actor; subsequent return to such a state would be a byproduct of the actor's transmission with

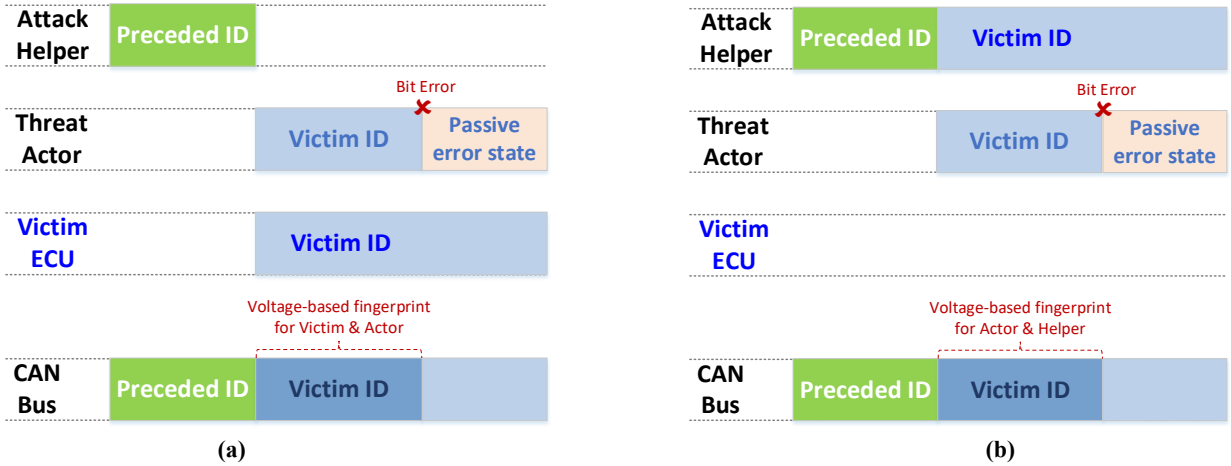

Fig. 3. Illustrating the DUET attack on the voltage-based IDS [23], where (a) shows poisoning the fingerprint of the victim ECU, and (b) illustrates how the victim is impersonated after its fingerprint is poisoned (hijacked).

the victim. Fig. 3(b) shows a consensual effect of the fingerprint poisoning where the malicious ECU pair impersonates the victim by simultaneously transmitting data frames, relying on the fact that the IDS cannot differentiate between the fingerprints of "victim & actor" and "actor & helper".

B. ECU Isolation Attack

In addition to the possible victim node impersonation discussed above, here we point out a simpler and serious attack scenario. The idea is that the malicious ECU pair will apply the steps in Fig. 3(a) during normal operation and not just during retraining time. The main goal of the attack is to blacklist the victim node, where a simultaneous transmission by a malicious ECU causes the exhibited fingerprint to mismatch what the IDS expects. By flagging the fingerprint mismatch the IDS will classify the data frame as suspicious and cause the victim node to be ignored. Such an attack scenario can be quite damaging to the CPS application since it hinders coordination between the victim node and the rest of the system; in fact, the consequences could be grave depending on the victim node's role within the CPS. Imagine targeting the steering control or brake subsystem of a vehicle with such an attack. We note that deriving fingerprints from the data part of the frame would not be effective since the victim is not expected to send the same payload every time; hence a data field cannot be the base of fingerprints that qualify acceptance or rejection of a transmission given the failure of authentication of the message ID.

It is worth noting that the node isolation attack can also be realized by forcing the victim to a bus-off state [6]. It has been shown in [13] that such an attack can be launched in a stealthy manner where the attacker sends a message with the same ID of the victim, and transmits all dominant bits in the data field. The victim ECU will detect an error and increment its TEC by 8. After several attack rounds, the victim ECU will go into a bus-off state and become isolated.

V. Detailed MAS Design

To mitigate the aforementioned attacks, MAS strives to provide a clean training dataset for fingerprinting each ECU. Such a dataset is generated using aperiodic transmissions based on a schedule that is mutually agreed upon among the ECUs. The clean dataset will allow the detection of fingerprint falsification attacks since it deprives the adversary of the opportunity for knowing when these transmissions are made. Furthermore, MAS engages one or multiple ECUs in each of the aperiodic transmissions. Having multiple ECUs simultaneously transmitting further enables the detection of fingerprint poisoning attempts since the impact on the voltage profile by the adversarial node will become noticeable and cannot stay stealthy. MAS is explained in the balance of this section.

A. Fingerprint Update Scheduling

The success of the aforementioned fingerprinting falsification attacks fundamentally depends on the adversary's ability to predict when a victim node usually transmits. Generally, a CPS involves a set of periodic tasks that run at rates that depend on the specific role that the individual task plays in realizing the CPS application. Scheduling these tasks and the associated communication traffic is typically based on static priorities that are monotonic with the rate that they are invoked at [39]. Fig. 4(a) shows an example, where node ξ_1 gets the highest priority since its period is the smallest. The priority of ξ_2 exceeds ξ_3, and both need to transmit less frequently than ξ_1. A least common multiple of all periods is usually determined where the scheduleability of the system is analyzed for meeting the latency constraints. In most designs, the capacity of shared resources, e.g., the communication bus, is not fully utilized to avoid risk of overstressing the system and also to support low priority non-critical tasks. For example, in an electrical vehicle entertainment-related data is given low priority on the CAN-Bus. MAS leverages these CPS design characteristics in mitigating fingerprinting falsification attacks.

MAS opts to maintain uncorrupted ECU fingerprints that can be employed by the IDS. As pointed out earlier, the attacker targets the training datasets that are used for fingerprint updates. MAS counters such an attack strategy by scheduling the data measurements in a manner that cannot be predicted by snooping on the CAN-Bus. The main idea is to designate certain transmissions for model training and schedule them in a manner that cannot be inferred without reverse engineering of the CPS design, something that we assume to be impractical without gaining full access to the system which would constitute a

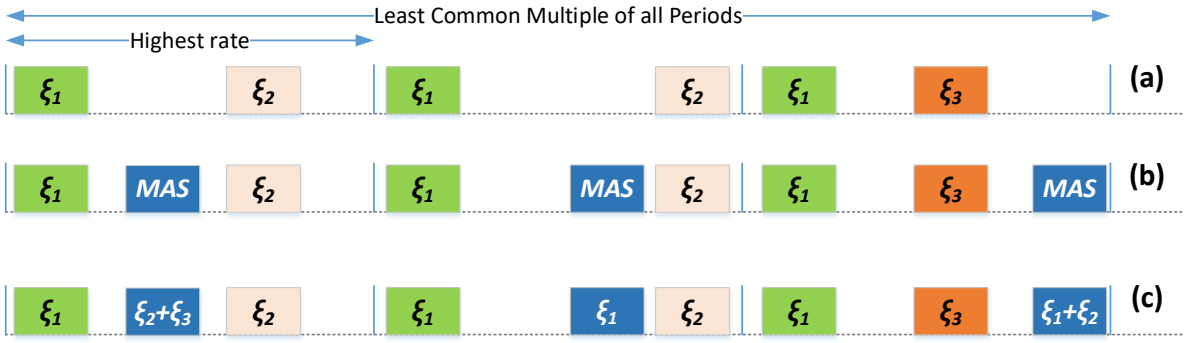

Fig. 4. An illustrative example of how transmissions appear on the CAN-Bus, where: (a) reflects a typical rate monotonic schedule of real-time communication with the priority being monotonic with the rate, i.e., $\xi_1 > \xi_2 > \xi_3$, (b) shows how MAS can be provisioned as a bandwidth preserving server for scheduling aperiodic transmissions, and (c) depicts a possible selection of ECUs by MAS for solo and combined transmissions.

different level of threats that is outside the scope of this paper or the scope of cyberattacks as a matter of fact. MAS considers training-related transmissions as aperiodic bus activities and employs a well-established mechanism for scheduling them. Specifically, MAS introduces a bandwidth-persevering server [39] to support training data collection. Such a mechanism strives to allocate some bus bandwidth for aperiodic, i.e., training-related, transmissions without interfering with periodic, application-critical, tasks. Fig 4(b) shows an example for when retraining messages could be scheduled by MAS.

A bandwidth server is in essence a periodic task that will run to support MAS. In each invocation, the server will check a queue of aperiodic (training) transmissions. In the context of CAN-Bus the bandwidth server task will have the least priority so that it does not interfere with the periodic, essential, transmissions. Each aperiodic training transmission will be associated with specific one or multiple ECUs, as explained later. We note that MAS will be running on each ECU so that all copies reach the same conclusion on when to send and which node will transmit; yet the designated ECUs for the transmission will indeed access the bus. There is no need for explicit synchronization among the individual copies of the MAS bandwidth server since they all will be monitoring the CAN-Bus broadcast. MAS is advocating either a constant or total bandwidth server implementation. The former ensures bus access time for training transmissions, defined as a percentage of the bus capacity. Such a percentage obviously will depend on how often retraining is to be conducted and the normal bus load, which depends on the ECU types and count in the CPS. A total bandwidth server, on the other hand, allows such minimum capacity to dynamically grow based on the load on the bus. The advantage of such a server configuration is that it can allow the training transmission frequency to be adaptively increased based on the environmental changes, yet at the price of increased complexity.

B. Determining Transmitters

In MAS, the introduced unpredictability is not only due to the transmission scheduling but also to determining which node participates. MAS pursues two types of aperiodic bus transmissions, namely training and deceptive. The former is designated for individual ECUs and is geared for collecting voltage measurements to update the ECU fingerprints. As mentioned earlier, retraining is necessary to factor in changes in the voltage profile due to environmental effects such as temperature. Unlike conventional approaches, MAS does not rely on the ordinary, periodic, transmissions of nodes to update the fingerprinting models. Instead, MAS provisions for training-specific transmissions that cannot be guessed by an adversary. Such a scheme evades any attempts to corrupt the training data and hijack the fingerprint of victim nodes. Moreover, MAS enables the model update to be continual and thus facilitates fingerprint correction in case the adversary accidentally succeeds in false data injection through some of the retraining transmissions.

The deceptive type of aperiodic transmissions is meant to detect fingerprint poisoning attempts by an attacker. As noted in [23], the voltage variation between one transmission and two

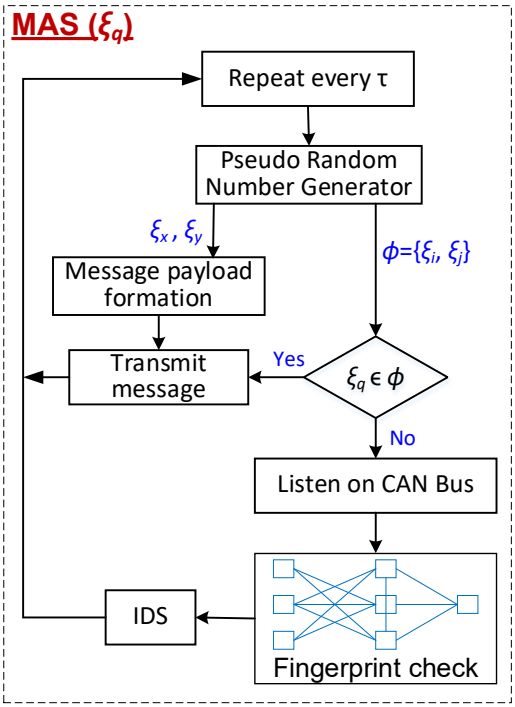

Fig. 5. A block diagram description of how MAS generates and checks deceptive transmissions.

simultaneous transmissions is noticeable and can be detected through the use of thresholds. Yet, when a third transmission is simultaneously made, the voltage leap cannot be noticed. In Section VI, we show results from our experiments that confirm such a property. MAS exploits such a property to confuse the adversary and detect fingerprinting manipulation attempts. Specifically, a deceptive transmission in MAS involves two participants. Hence, MAS introduces new fingerprints corresponding to combinations of two ECUs. Fig 4(c) shows an example of ECU designation for the individual retraining messages provisioned by MAS. A fingerprinting falsification attack targeting a node will be detected by MAS by comparing the voltage profiles of single and dual node transmissions. To elaborate if an adversary could simultaneously transmit during a training frame (an aperiodic transmission) of node ξ_i, the system will detect such an attack given that MAS develops fingerprints for dual node transmissions; the adversarial transmission will also increase the voltage shift in the header and will be flagged by the IDS. Note that targeting a deceptive transmission will not affect the voltage profile much and would not cause MAS to fail in detecting attacks against the training transmissions.

The data payload of a deceptive message will also vary to avoid replays. In MAS, each ECU will employ a pseudo random number generator (PRNG) with the same seed. To generate a deceptive message, each ECU generates two random numbers reflecting the IDs of participants in the deceptive transmission, i.e., the random numbers are generated in the range [1, m], where m is the number of ECUs in the CPS. The message payload will be the concatenated bit-pattern of two additional random numbers, for a total of 64 bits which

corresponds to the data field in the CAN frame shown in Fig. 1. By employing the same PRNG and using similar seed values, all MAS copies at the various ECUs will generate the same deceptive message and identify the same set of transmission participants. By running the same bandwidth-preserving server on all ECUs, the transmission time will be synchronized. Recall that all ECUs receive transmissions on the CAN-Bus and thus, they can stay synchronized. The bandwidth-preserving server will have the least priority in terms of bus-access so that the aperiodic transmissions do not interfere with the periodic, critical, ones; the latter are subject to timing constraints and should be getting privilege as noted earlier.

Fig. 5 provides a block diagram summary of MAS' procedure for generating deceptive frames. The steps are iterated every τ time units, which reflects the periodicity of the bandwidth preserving server. In each iteration, the PRNG is used to generate two pairs of ECU IDs, namely, (ξ_i, ξ_j) and (ξ_x, ξ_y). The latter is used to form the data payload of a deceptive message. If the node happens to be among the randomly selected participants, i.e., ξ_i or ξ_j, the message will be sent; else the node will just snoop on the CAN bus and match the fingerprint of the deceptive transmission with that reflecting the combination of ξ_i and ξ_j.

C. Identifying Transmitters

As stated in Section III-B, voltage-based fingerprinting proved to be a robust means for identifying the source of transmissions on the bus. MAS strives to thwart the threat of falsification of device fingerprints by provisioning unpredictable transmissions to collect retraining data. We note that MAS does not presume a certain machine learning technique or specific features for defining the fingerprints. In other words, contemporary schemes in the literature, e.g., [18], may be applied, where MAS only opts to ensure that the retraining of the fingerprinting model is based on uncorrupted data. As discussed in the previous section, MAS introduces deceptive transmissions that are used to detect malicious interference of the aperiodic transmissions of individual ECUs. Such interference detection is based on comparison of the voltage profile of a retraining-related transmission of a node ξ_i and the profile when ξ_i co-transmit a deceptive frame with another ECU. For that, MAS again extracts time-domain features and applies voltage-based fingerprinting techniques to define a signature for each pair of co-transmitting ECUs. Specifically, we compute the mean of the voltage level for each low-to-high/high-to-low transitions.

Identifying an ECU is usually done by extracting the features on its transmission and applying the machine learning classifier to find the closest fingerprint among those of existing nodes. This corresponds to using the machine learning model for testing. The same applies when two nodes co-transmits. Since the fingerprint of an attacker could be unknown, e.g., when a malicious ECU is installed in a stealthy manner, it is important to prevent misclassification of a simultaneous transmission of an attacker along with a victim node ξ_v, from being matched to either ξ_v or any pair of nodes that includes ξ_v. In other words, determining the ECU that most probably sent a frame does not allow distinguishing an unknown fingerprint from the already-known legitimate ones. MAS tackles such an issue by the use of Fisher-Discriminant Analysis (FDA) to find a transformation matrix W that determines the most discriminant features among the training samples of distinct ECUs. The matrix W transforms the features of the transmitted frame in order to compute the Mahalanobis distance to the kernel of an ECU fingerprint, where the kernel here reflects the mean and variance over the feature space. The Mahalanobis distance, Δ, is computed using:

$$\Delta = \sqrt{(F-\mu)^T \sigma^{-1} (F-\mu)} \quad (1)$$

where μ, and σ, respectively, are the mean and variance. If Δ exceeds a certain threshold, the fingerprint is deemed to be unknown. On the other hand, if Δ is closer to another ECU, it indicates a sign of an impersonation attempt.

VI. Security Analysis

The most notable attacks on CAN buses are: (1) ECU impersonation, (2) message injection, (3) voltage corruption, (3) bus off, and (4) ECU isolation. The first and second on the list are obvious. The third reflects attempts to corrupt the voltage-based fingerprint. A variant of such an attack is called hill-climbing which opts to modify the fingerprint using multiple frames. The hill-climbing attack is not applicable in the case of MAS since the fingerprint is extracted based on a single frame. In the bus-off attack a malicious unit intentionally interferes with the target ECU to advance the error counter and causes transition to the bus-off state. Finally, the fourth attack is kind of a denial of service by making simultaneous transmission with the victim so that other nodes do not recognize the fingerprint of the victim ECU and reject its messages. We note that detecting impersonation attempts and identifying the attacker would suffice for mitigating the first 4 attacks on the list. Hence, in the following we analyze the robustness of MAS against impersonation and isolation attacks.

Lemma #1: MAS detects individual or collusive impersonation attempts.

Proof: In order to impersonate a victim node, the attacker has to poison the fingerprint training data; otherwise the IDS will detect the mismatch between the fingerprint of the victim and the transmitting (malicious) ECUs. To poison the training data, the adversary has to predict the exact time of MAS' aperiodic transmissions and the exact random data to be transmitted. Given the irregular pattern of the aperiodic frames, the adversary may try the following strategies either alone or with the help of an accomplice:

i. *Target the periodic transmission of the victim*: Since MAS uses aperiodic transmissions to collect training data, the fingerprint of the victim node will not be impacted by interfering with periodic transmissions. Hence, this strategy will fail since the attack will be detected by the IDS as the fingerprint of the attacker does not match that of the victim.
ii. *Interfere with all victim node's transmissions*: The adversary would continually snoop on the bus and transmit when recognizing the arbitration part of the victim frame. Here, the adversary's transmission cannot be synchronized

with the victim and only the data part of the frame can be targeted. Random data has to be transmitted given the adversary's unawareness of the actual data in the victim mode frame. This will trigger bus collision between the victim transmission and that of the attacker. However, as such collision repeats it could be flagged and attributed to the presence of malicious behavior. Also the adversary will not succeed in replacing the training data, but rather prevents collecting it.

Thus, impersonation attempts will be exposed.

Lemma #2: MAS detects attacks that opt to isolate a node.

Proof: The main goal of the attack is to blacklist the victim node, where a simultaneous transmission by a malicious ECU causes the exhibited fingerprint to mismatch what the IDS expects. Since the attack requires synchronizing with the victim node, a malicious ECU pair could only apply the steps in Fig. 3(a) during normal operation (periodic transmissions) and not during retraining time (aperiodic transmissions). MAS can detect such an attack for the following since: (a) the dataset for forming fingerprints is clean, where the adversary cannot track aperiodic transmissions to poison the training data; (b) even if the adversary can interfere with aperiodic transmissions of a victim node, MAS will detect such interference by recognizing the difference between the voltage profile (fingerprint) of two nodes and a single node transmission. Hence, attempts to isolate the victim cannot be stealthy and will be detected by MAS.

VII. Performance Validation

A. Implementation Setup

In our experiment, we used voltage data that has been collected from the vehicle electronic system of a Nissan Sentra [22]. The dataset assumes that the frame ID is representing the ECU identifier. We have decoded the packet in order to extract the message IDs; those employed in the experiment were from the following groups {374, 375}, {644, 645, 646}, {386}, {533, 534}, and {849}. We associate for each group an ECU ID. Thus we identify five ECUs. MAS had to identify the correct group for a transmitting ECU. We also collected results for combined transmissions by reconstructing the bimodal distribution from the original ECU voltage. Then we employed the feature extraction mechanism suggested in [22]. We evaluate MAS' ability to detect stealthy fingerprint falsification attacks, in term of the following:

- *Aperiodic transmission predictability*: The success of MAS is based on the unpredictability of the training transmissions which prevents the synchronization of the attacker message with the victim's message. We assume the attacker employs a Long Short-Term Memory (LSTM) model in order to infer the pattern of the aperiodic transmissions. To evaluate this metric, the adversary is assumed to have knowledge of the schedule of some previous aperiodic transmissions, and try to infer the next aperiodic transmission. In essence, this metric measures the accuracy of the attacker's prediction of the fingerprint retraining data collection. We considered the bus free time based on the considered dataset to determine the schedule of MAS related transmission. The LSTM has two layers with 4 cells; its last layer includes a softmax with two outputs indicating whether there is aperiodic transmission or not.
- *Victim identification accuracy*: The success of MAS in mitigating targeted attacks, is based on the unpredictability of the transmitting ECU. To assess this metric, we assume that the attacker employs an LSTM in order to uncover the pattern of the specific victim aperiodic transmission. We distinguish such LSTM from the one used by the adversary for predicting training (aperiodic) transmissions by using the suffix V and S, respectively, i.e., $LSTM_V$ for victim identification and $LSTM_S$ for predicting aperiodic transmissions Assuming the adversary's full knowledge of the aperiodic schedule timing, we aim to gauge the accuracy of the attacker for associating the specific victim to its own slot in the schedule. The employed $LSTM_V$ includes two layers with 4 cells and a softmax layer with the cardinality of the possible combination of ECU.

To schedule the aperiodic (MAS generated) transmissions, we consider the bus free time and randomly allocate some of that time for MAS such that only a fraction of the available bandwidth is utilized. Overall, the number of MAS messages is less than the periodic transmission count, which is consistent with practice as the training overhead should not be dominant. In the simulation, we also vary the ECU count, the number of deceptive (combined) transmissions, and the attacker's knowledge of training related transmissions. The validation experiments and results are discussed next.

B. Simulation Results

Figure 6 gauges the effectiveness of MAS in terms of the success rate in detecting fingerprint interference attempts. Such success depends on the model accuracy of the number of ECU participants in combined transmissions. Recall that MAS analyzes the Mahalanobis distance, Δ, to distinguish between individual (single) and simultaneous (combined) transmissions. As indicated by the results, the accuracy of identifying a *single* ECU is 99%. For *combined* transmissions, the fewer the number of engaged ECUs, the more accurate the identification becomes. MAS sustains acceptable distinction even for six combined transmissions. The IDS can also discriminate between a *mix* of individual and combined transmissions. MAS achieves such a high success rate in detecting individual and combined transmissions since the voltage distribution is significantly different when a node sends alone and with other accomplices. Figure 7 confirms such a conclusion and reports the kernel density distribution of voltage for distinct configurations where we reach unimodal Gaussian distribution for a single ECU and bimodal for combined transmission. When two and three nodes send similar headers, the distribution becomes bimodal and spans higher voltages. Our results are consistent with the literature [18][23]. This indicates MAS' higher success rate of detecting the participation of an adversarial node in a transmission, e.g., to poison the training data.

The adversary's ability to predict the training transmission schedule is assessed in Figure 8. In this experiment, the adversary is assumed to know that MAS is applied and be able

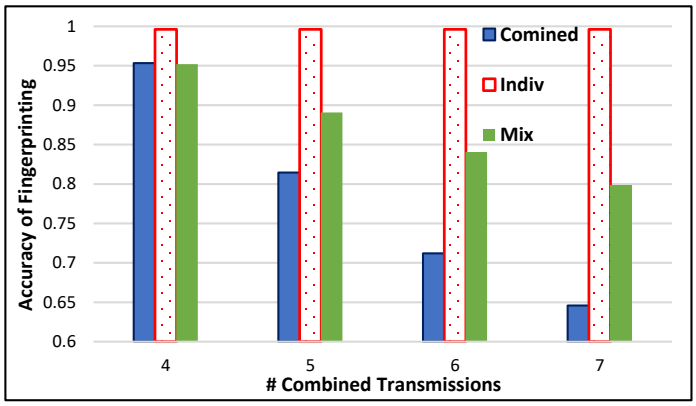

Fig. 6: Fingerprinting accuracy for distinct combination of ECUs.

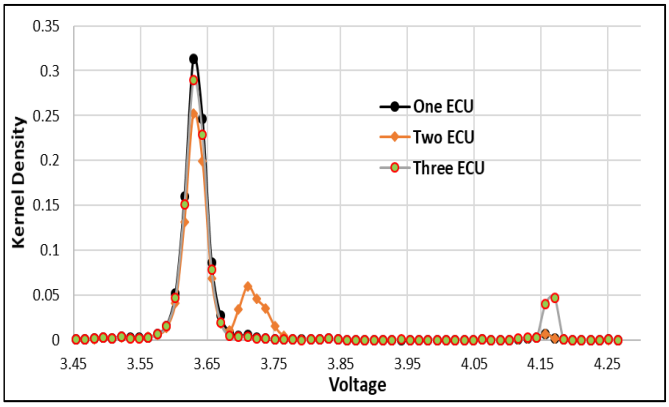

Fig. 7: Kernel distribution for the distinct cardinality of combined transmissions.

to distinguish between periodic and aperiodic transmissions. As noted earlier, we have implemented $LSTM_S$ that is trained using some aperiodic transmission timings to predict when the next transmission will be. In the figure, the adversary has varying levels of knowledge of prior training-related (aperiodic) transmissions, where 50% implies that the attacker knows the schedule of half of the aperiodic transmissions. Basically, as the adversary is able to overhear some of the previous aperiodic transmissions, and try to predict the upcoming ones. The figure demonstrates the robustness of MAS against attempts to poison the fingerprint training data. The randomized aperiodic transmission schedule significantly limits the ability of an attacker to predict when to interfere, where the accuracy of $LSTM_S$ is quite low and reflects just random guesses. The accuracy does not improve even if the LSTM is trained with more data as long as the percentage does not change.

To gauge the effect of MAS' deceptive aperiodic transmission, we have implemented an additional LSTM, denoted, $LSTM_V$, that an adversary could employ to predict when a victim ECU will transmit next. This is a different LSTM from the one discussed about for predicting the training schedule. Here, the adversary is assumed to know the aperiodic schedule of training messages, which is not even feasible as shown by Figure 8. Figure 9 reports the results which reflect the adversary's ability to identify the aperiodic transmission of a targeted victim node. The figure clearly depicts a very low prediction accuracy which confirms the adversary's inability to impersonate and/or invalidate the legit ECU fingerprint during the training stage. When collectively considering the results of both figures 8 and 9, it can be concluded that the accuracy for predicting and identifying the aperiodic transmission of a victim node is in the single digit (less than 10%) percent. Indeed, with such a poisoning ratio it is impossible for the adversary to corrupt the fingerprinting dataset.

C. *Comparison with Competing Approaches*

We now compare MAS with other state-of-the-art VIDSs, namely, Viden [18], Scission [21], Simple [22] and RAID [23]. To protect the CAN-bus system, the IDS should be able to detect the attack or raise an alarm when an attacker tries to evade the provisioned protection, and ideally identify the malicious actor. Yet, identifying the attacker is also the most difficult feature to realize in an IDS. Table 1 compares the capabilities of MAS to the aforementioned competing schemes. As indicated by Table 1, MAS can identify the malicious actor in all attack scenarios, something other schemes cannot do. Such capability is attributed to the fact that MAS extracts the ECU fingerprint from a single frame, retrains fingerprint from legit messages, identifies the source of message based on the

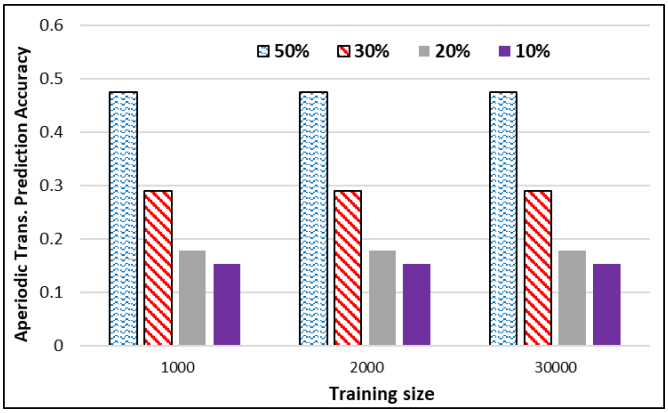

Fig. 8: Adversary's prediction accuracy for training related transmission.

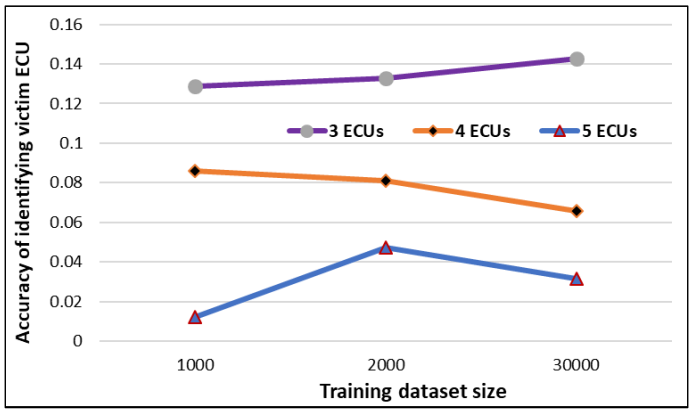

Fig. 9: Success rate for predicting the training transmission of a victim ECU.

frame's data field which could not be manipulated and provisions aperiodicity of the training data collection.

Table 1: Comparing the capabilities of MAS to those of competing schemes

IDS / Attack	Viden	Scission	Simple	RAID	MAS
Message Injection	I.S	I.S	I.S	I.S	I.S
Impersonation	I.S	I.S	I.S	I.S	I.S
Hill-climbing	V	V	D	I.S	I.S
Voltage Corruption	V	V	V	P	I.S
ECU Isolation	V	V	V	V	I.S
Bus off	V	V	V	V	I.S

I.S: Identify source; V: Vulnerable; D: Detection; P: Prevent

Due to changes in the operating environment, e.g., variations in temperature and voltage input, the voltage-based fingerprint of the existing ECUs will change and the underlying VIDS must be updated. Viden and Scission extract fingerprint from multiple frames and hence becomes susceptible to hill-climbing attacks, where an attacker gradually increases the rate of malicious messages (voltage data poisoning) in order to associate the fingerprint of the victim ECU with that of the attacker. Simple and MAS extract fingerprints from a single frame and hence are able to detect hill-climbing attacks. For the voltage corruption attack, a malicious node transmits a message simultaneously with the victim ECU causing the voltage features of such an ECU to be manipulated without detection. RAID leverages the ID extension bit in the CAN bus frame to make the ECU ID unpredictable to the attacker. MAS does much better than RAID by: (i) randomizing the schedule of the retraining transmissions in order to lower the probability of corrupting the fingerprint dataset, and (ii) injecting descriptive transmissions to confuse and expose attackers, and (iii) detecting corruption attempts and identifying the source of the malicious message by checking the sender's fingerprint of the data field of the message. We also note that all existing VIDS are vulnerable to bus-off attacks except MAS since MAS can identify the attacker and block it before the error count reaches the bus-off threshold.

Since RAID is the most recently published and is more capable than other existing schemes, we have compared the performance of MAS and RAID. Figure 10 shows the success rate of attacker identification when MAS or RAID are applied. We have assumed that the adversary targets transmissions used for collecting retraining data, i.e., launch the DUET attack explained in Section IV. The adversary's success ratio is varied from 0.2-0.8, as noted on the *x*-axis in Figure 10. Such a success ratio reflects the adversary's ability of distinguishing the transmissions of the targeted ECU, which are normally periodic as also being considered by RAID. When detecting the DUET attack, RAID discards the transmission and hence loses training data. In other words, the attack diminishes the size of collected retraining data rather than poisoning it. Consequently, as indicated by the results in Figure 10, the fingerprinting accuracy is reduced and RAID's ability to identify the malicious node is significantly degraded. Such degradation is almost proportional to the data poisoning rate, i.e., DUET attack attempts. MAS on the other hand stays robust, where the retraining transmissions are no longer predictable to the adversary. To highlight the major performance edge that MAS has, let us consider the case when the attacker could target 50% of the training-related transmissions. For such a case, RAID's attack identification accuracy drops to 70% while MAS suffers no notable impact and thus has 30% advantage over RAID. In Figure 10, we show two cases reflecting the probability that the attacker's attempt happens to accidently match some of the actual retraining transmission, e.g., using $LSTM_S$, leading to discarding it. Such a probability is set for 0.2 and 0.3, which are even higher than what is shown in Fig. 8. For the latter the attacker identification accuracy is slightly impacted at high data poisoning, yet consistently stays above 93% and significantly outperforms RAID.

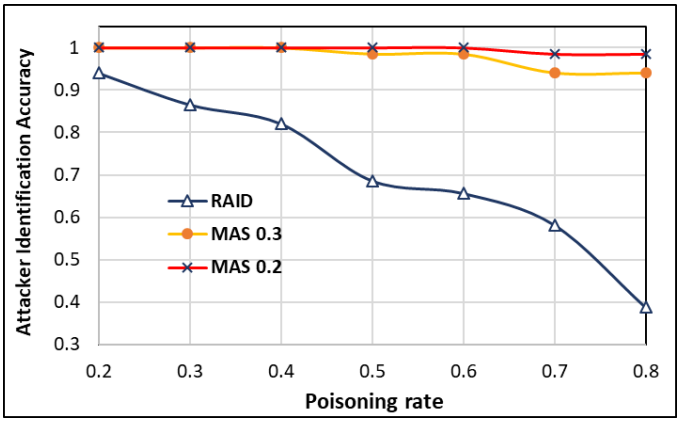

Fig. 10. Comparing MAS to RAID in terms of their robustness under poisoning attacks launched during the collection of retraining data.

VIII. CONCLUSIONS

This paper has focused on the CAN bus, which is widely used by the automotive industry. The CAN bus is deemed to be vulnerable to node masquerading and message spoofing. The popular approach for mitigating such vulnerability is to derive a voltage-based fingerprint for each ECU based on the properties of bus signals. Yet, prior work showed such fingerprinting protection could be broken by a pair of collusive attackers that target the system at the time of fingerprinting update and successfully impersonate a victim node. We have further pointed out another attack that could cause a victim ECU to become isolated. To counter the aforementioned attacks, this paper has presented MAS. The key design principle of MAS is to degrade the adversary's ability for poisoning the data collected for forming/updating the fingerprint models. The validation results using data from a vehicle electronic system, have confirmed the effectiveness and viability of MAS.

The effectiveness of MAS can be further extended by factoring in other ECU and bus characteristics, e.g., incorporating clock-based fingerprinting capabilities. We envision that a combined VIDS and CIDS could better safeguard a CAN bus. We plan to investigate such a promising research direction in the future. We also plan to investigate the growing attack surface with the integration of CPS systems with the 5G technology and beyond [40].

REFERENCE

[1] M. Bozda, M. Samie, S. Aslam and I. Jennions. "Evaluation of CAN Bus Security Challenges," *Sensors*, Vol. 20, No. 8, pp. 2364, 2020.

[2] J. Liu, S. Zhang, W. Sun, and Y. Shi, "In-vehicle network attacks and countermeasures: challenges and future directions," *IEEE Network*, Vol. 31, No. 5, pp. 50–58, 2017.

[3] O. Avatefipour. "Physical-Fingerprinting of Electronic Control Unit (ECU) Based on Machine Learning Algorithm for In-Vehicle Network Communication Protocol 'CAN-BUS'," *MS Thesis*, Dept. of Computer Engineering, University of Michigan-Dearborn, 2017.

[4] Tencent Keen Security Lab, "Tencent keen security lab: experimental security research of Tesla autopilot," 2019, https://keenlab.tencent.com/en/2019/03/29/Tencent-Keen-Security-Lab-Experimental-Security-Research-of-\Tesla-Autopilot/.

[5] R.-P. Weinmann and B. Schmotzle, "Tbone – a zero-click exploit for Tesla MCUs," 2020, https://kunnamon.io/tbone/tbone-v1.0-redacted.pdf.

[6] K.-T. Cho and K. G. Shin, "Error Handling of In-vehicle Networks Makes Them Vulnerable," *Proc. the ACM SIGSAC Conf. on Computer and Comm. Security (CCS '16)*, pp. 1044–1055, Oct. 2016.

[7] H. Wen, Q. A. Chen, and Z. Lin, "Plug-N-Pwned: Comprehensive Vulnerability Analysis of OBD-II Dongles as A New Over-the-Air Attack Surface in Automotive IoT," *Proc. of the 29th USENIX Conference on Security Symposium*, Article 54, pp. 949–965, 2020.

[8] U. Ezeobi, H. Olufowobi, C. Young, J. Zambreno, and G. Bloom, "Reverse Engineering Controller Area Network Messages using Unsupervised Machine Learning," *IEEE Consumer Electronics Magazine*, pp. 1–1, 2020.

[9] H. Olufowobi, C. Young, J. Zambreno, and G. Bloom, "SAIDuCANT: Specification-Based Automotive Intrusion Detection Using Controller Area Network (CAN) Timing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1484–1494, Feb. 2020.

[10] M. D. Pese, T. Stacer, C. A. Campos, E. Newberry, D. Chen, and ´ K. G. Shin, "LibreCAN: Automated CAN Message Translator," *Proc of the ACM SIGSAC Conf, on Comp, & Comm. Security*, 2019, pp. 2283–2300.

[11] S. Kulandaivel, T. Goyal, A. K. Agrawal, and V. Sekar, "CANvas: Fast and Inexpensive Automotive Network Mapping," *Proc. the 28th USENIX Security Symposium (USENIX Security 19)*, Aug. 2019, pp. 389–405.

[12] SF. Lokman, A.T. Othman, and MH. Abu-Bakar, "Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review," *J. Wireless Com. Network*, Vol. 2019, #184, 2019.

[13] K. Iehira, H. Inoue and K. Ishida, "Spoofing attack using bus-off attacks against a specific ECU of the CAN bus," *Proc. 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 2018.

[14] Q. Gu, D. Formby, S. Ji, H. Cam and R. Beyah, "Fingerprinting for Cyber-Physical System Security: Device Physics Matters Too," *IEEE Security & Privacy*, vol. 16, no. 5, pp. 49-59, September/October 2018.

[15] D. Formby, P. Srinivasan, A. M. Leonard, J. D. Rogers and R. A. Beyah. "Who's in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems." Proc. of *NDSS*, 2016.

[16] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," *Proc. IEEE Intell. Veh. Symp.*, Jun. 2011, pp. 1110–1115.

[17] K.-T. Cho, and K. G. Shin. "Fingerprinting electronic control units for vehicle intrusion detection," *Proc. 25th USENIX Security Symposium (USENIX Security 16)*, 2016.

[18] K.-T. Cho and K. G Shin, "Viden: Attacker Identification on In- Vehicle Networks," *Proc. of the ACM SIGSAC Conference on Computer and Communications Security*, 2017, 1109–1123.

[19] Y. Yang, Z. Duan, and M. Tehranipoor, "Identify a Spoofing Attack on an In-Vehicle CAN Bus Based on the Deep Features of an ECU Fingerprint Signal," *Smart Cities*, Vol. *3*, pp. 17-30, 2020.

[20] W. Choi, et al., "VoltageIDS: Low-level communication characteristics for automotive intrusion detection system," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2114–2129, 2018.

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

[22] M. Foruhandeh, et al., "SIMPLE: Single-frame based physical layer identification for intrusion detection and prevention on in-vehicle networks," *Proc. the Annual Computer Security Applications Conference (ACSAC)*, 2019, pp. 229–244.

[23] R. Bhatia, V. Kumar, K. Serag, Z. B. Celik, M. Payer and D. Xu, "Evading Voltage-Based Intrusion Detection on Automotive CAN" *Proc. of Network and Distributed System Security Symp. (NDSS)*, Feb. 2021.

[24] M. Tian, R. Jiang, C. Xing, H. Qu, Q. Lu and X. Zhou, "Exploiting Temperature-Varied ECU Fingerprints for Source Identification in In-vehicle Network Intrusion Detection," *Proc. IEEE 38th Int'l Performance Computing and Communications Conference (IPCCC)*, 2019.

[25] C. Young, J. Zambreno, H. Olufowobi, and G. Bloom, "Survey of Automotive Controller Area Network Intrusion Detection Systems," *IEEE Design Test*, vol. 36, no. 6, pp. 48–55, Dec. 2019.

[26] C. Young, H. Olufowobi, G. Bloom, and J. Zambreno, "Automotive Intrusion Detection Based on Constant CAN Message Frequencies Across Vehicle Driving Modes," *Proc. of ACM Workshop on Automotive Cybersecurity*, Mar. 2019, pp. 9–14.

[27] Q. Zhao, M. Chen, Z. Gu, S. Luan, H. Zeng, and S. Chakrabory, "CAN Bus Intrusion Detection Based on Auxiliary Classifier GAN and Out-of-distribution Detection," *ACM Trans. Embed. Comput. Syst.*, Vol. 21, No. 4, Article 45, 30 pages, July 2022.

[28] X. Li, F. Liu, D. Li, T. Hu, M. Han, "Illegal Intrusion Detection for In-Vehicle CAN Bus Based on Immunology Principle," *Symmetry*, Vol. 14, No. 8, Article1532, 2022.

[29] Z. Bi, G. Xu, G. Xu, M. Tian, R. Jiang, and S. Zhang, "Intrusion Detection Method for In-Vehicle CAN Bus Based on Message and Time Transfer Matrix," *Security and Communication Networks*, vol. 2022, Article ID 2554280, 19 pages, 2022.

[30] T. Koyama, T. Shibahara, K. Hasegawa, Y. Okano, M. Tanaka, and Y. Oshima, "Anomaly Detection for Mixed Transmission CAN Messages Using Quantized Intervals and Absolute Difference of Payloads," *Proc. of ACM Workshop on Automotive Cybersecurity*, Mar. 2019, pp. 19–24.

[31] S. Longari, M. Penco, M. Carminati, and S. Zanero, "CopyCAN: An Error-Handling Protocol Based Intrusion Detection System for Controller Area Network," *Proc. of ACM Workshop on Cyber-Physical Systems Security & Privacy*, 2019, pp. 39–50.

[32] H. Olufowobi, C. Young, J. Zambreno, and G. Bloom, "SAIDuCANT: Specification-Based Automotive Intrusion Detection Using Controller Area Network (CAN) Timing," *IEEE Transactions on Vehicular Technology*, Vol. 69, No. 2, pp. 1484–1494, Feb. 2020.

[33] Z. Bi, G. Xu, G. Xu, M. Tian, R. Jiang, and S. Zhang, "Intrusion Detection Method for In-Vehicle CAN Bus Based on Message and Time Transfer Matrix," *Security and Communication Networks*, vol. 2022, Article ID 2554280, 19 pages, 2022.

[34] G. Bloom "WeepingCAN: A Stealthy CAN Bus-off Attack," *Proc. of the Network and Distributed System Security Symposium (NDSS 2022)*, San Diego, CA, April 2022.

[35] M. Marchetti, D. Stabili, A. Guido and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," *Proc IEEE 2nd Int'l Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, 2016.

[36] S. Corrigan, "Introduction to the Controller Area Network (CAN)," *Technical Report # SLOA101B*, Texas Instrument, May 2016.

[37] M. Hamad, and V. Prevelakis, "SAVTA: A Hybrid Vehicular Threat Model: Overview and Case Study," *Information*, 11(5), pp. 273, 2020.

[38] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Proc. of Black Hat*, Las Vegas, NV, Aug. 2015.

[39] J. W. S. Liu, *Real-Time Systems, 1/e*. Prentice Hall, USA, 2005.

[40] A. Mughaid, S. AlZu'bi, A. Alnajjar, et al. "Improved dropping attacks detecting system in 5g networks using machine learning and deep learning approaches," *Multimed Tools Appl.*, 22 pages, Sept. 2022.

IX. STATEMENTS AND DECLARATIONS

Funding: The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing Interests: The authors have no relevant financial or non-financial interests to disclose.

Author Contributions: All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Wassila Lalouani, and Yi.

Deng. The first draft of the manuscript was written by Mohamed Younis and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Data Availability Statement: The datasets analyzed during the current study are available in [22].

I. BIOGRAPHY

Wassila Lalouani is currently an assistant professor in the Department of Computer and Information Science, Towson University. She got her PhD in Computer Science from the University of Maryland Baltimore County. Her research interest includes network management and protocols, machine learning, and network security.

Yi Dang received the B.S degree in Electronic Information Engineering from Taiyuan University of Technology, Taiyuan, China, in 2003. He received his MS degree in Electrical Engineering from UMBC in December 2021. His primary research focus is on wireless communication and security.

Mohamed F. Younis is currently a professor in the department of computer science and electrical engineering at the university of Maryland Baltimore County (UMBC). He received his Ph.D. degree in computer science from New Jersey Institute of Technology, USA. Before joining UMBC, he was with the Advanced Systems Technology Group, an Aerospace Electronic Systems R&D organization of Honeywell International Inc. While at Honeywell he led multiple projects for building integrated fault tolerant avionics and dependable computing infrastructure. He also participated in the development of the Redundancy Management System, which is a key component of the Vehicle and Mission Computer for NASA's X-33 space launch vehicle. Dr. Younis' technical interest includes network architectures and protocols, wireless sensor networks, embedded systems, fault tolerant computing, secure communication and distributed real-time systems. He has published over 300 technical papers in refereed conferences and journals. Dr. Younis has seven granted and three pending patents. In addition, he serves/served on the editorial board of multiple journals and the organizing and technical program committees of numerous conferences. Dr. Younis is a Fellow of the IEEE and the IEEE communications society.