CAN-SQUARE - Decimeter Level Localization of Electronic Control Units on CAN Buses

Bogdan Groza, Pal-Stefan Murvay, Lucian Popa, Camil Jichici

Politehnica University Timisoara, Faculty of Automatics and Computers
{bogdan.groza, pal-stefan.murvay, lucian.popa,
camil.jichici}@aut.upt.ro

Abstract. The CAN bus survived inside cars for more than three decades due to its simplicity and effectiveness while protecting it calls for solutions that are equally simple and effective. In this work we propose an efficient mechanism that achieves decimeter-level precision in localizing Electronic Control Units (ECUs) on the CAN bus. The proposed methodology requires two connections at the ends of the bus and a single rising edge, i.e., the start of a dominant bit. Since several such rising edges are present in every frame, malicious devices may be easily localized with high accuracy from single frame injections. Our methodology requires only elementary computations, e.g., additions and multiplications, which are trivial to perform and implement. We prove the feasibility of the proposed methodology inside a real car and perform more demanding experiments in a laboratory setup where we record modest overlaps only between nodes that are 10 cm apart. We prove resilience against replacement and insertion attacks as well as against temperature variations in the range of 0-60°C.

1 Introduction and Motivation

There is really not much more that needs to be said to convince readers on the insecurity of modern vehicles and their communication buses, e.g., [17], [2], [18], etc. It is apparent that a bus designed by BOSCH in the 80s, the Controller Area Network (CAN), cannot cope with modern security needs. With new vulnerabilities reported each year, the challenge in designing security for this widely spread bus that proved its efficiency for more than three decades remains open. The difficulty of embedding cryptographic elements inside the 64-bit CAN frames was so tremendous that researchers looked at various alternatives such as authentication data embedded in the ID field [11], [24], covert timing channels [25], etc. The industry did not hesitate to proceed in this direction as proved by recently released AUTOSAR standards for secure in-vehicle communication [1] which introduces truncated authentication tags of 24-28 bits in each frame and a 0-8 bits freshness parameter (see SecOC profiles 1-3 in [1]). Hopefully, the time for such compromises may come to an end with the release of CAN-FD that carries 512 bit payloads and sets room for regular sized cryptographic elements. Clearly, the high-bandwidth CAN-FD will make the use of CAN inside cars even more attractive and the deployment of cryptographic security easier.

But adding cryptography is far from solving the problem as ECUs may be compromised and engaged in malicious activities, i.e., impersonating other nodes, cryptographic keys may be extracted by memory dumps or side-channel attacks, etc. In this

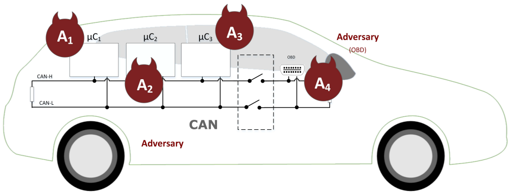


Fig. 1. Addressed setting: an in-vehicle CAN bus with adversaries at several possible locations

context, using physical layer techniques to identify [3], localize [20] and eventually isolate nodes [9] is of prime importance.

Motivation and contributions. In-vehicle networks are controlled environments where it is somewhat unlikely that an adversary will tap the bus at any random location due to obvious physical access difficulties. All of the attacks reported so far come from open ports, e.g., the OBD port, corrupted control units, e.g., the infotainment or telematic unit, etc. Therefore the most likely scenario is that an attack will originate at a predictable location. Figure 1 suggests such a setting in which an in-vehicle CAN bus is tapped by an adversary on the OBD port or possibly by corrupting an existing electronic control unit (insertion of malicious nodes or tapping the bus in accessible locations is not excluded from our experiments). In-vehicle networks are not necessarily flat, i.e., multiple buses can be linked via gateways, a case in which the solution presented in this work can be easily extended by placing probes at the ends of each bus. Freshly emerged works [9] have proved the feasibility of disconnecting parts of the bus in real time without damaging real-time communication. In this context, localizing intruders on the bus becomes an immediate problem. In this work, we propose a technique based on frame timing and signal characteristics. This technique was previously explored in [20] but the results were not so successful in detecting node replacement and insertions on the bus. We use the same setting as the one in [20] and achieve high precision in localizing the devices regardless to the type of attack: node corruption, replacement or insertion. If intruder nodes can be localized then other countermeasures such as bus segment disconnection can be applied. Moreover, given the small localization error, a visual inspection will immediately point to the maliciously planted device. We briefly summarize the contributions of our work as follows:

1. we prove localization to be feasible with high precision, i.e., most of the times in the order of a decimeter or less according to the inter-distance matrix, even when sampling from the CAN-H line alone without needing the additional CAN-L voltage (this halves the wiring costs),
2. while other proposals require statistical tests or more demanding machine learning algorithms, we reduce computational costs to virtually nothing, i.e., our localization scheme requires only buffering some samples and one subtraction/division for each sample (these are trivial to implement),
3. we prove the resilience of the technique in front of node replacement and insertions, a possible attack scenario or an innocuous circumstance called by a faulty device, to which previous works did not offer much resilience,

4. we provide experiments on a setup built with a new professional CAN bus cable and prove the resilience of the proposed method in front of multiple adversaries and environmental changes, i.e., temperature variations, in addition to validating against our previous dataset from [20] where such an analysis was not performed,
5. we further endorse the use of double bus taps which originates from the work in [9] although in a very distinct context, i.e., that of isolating segments of the bus, which opens numerous roads for future security solutions.

The rest of the paper is organized as follows. We briefly present the related work in the next subsection. Section 2 provides some background on CAN and a comprehensive description of our experimental setup. In Section 3 we begin by presenting the methodology in our work and then proceed to the experimental results in Section 4. Finally, Section 5 holds the conclusions of our work.

1.1 Related works

Since it was not designed to include security mechanisms, the CAN standard provides no means of uniquely identifying the transmitter of a frame which opens door for attacks. An analysis of CAN bus attacker capabilities can be found in [8]. To overcome this design limitation, various solutions have been proposed starting from regular cryptographic security which has been also recently adopted by the industry [1], placing additional ECUs to act as gateways [10], destroying malicious frames by legitimate senders that recognize their IDs [6] and, as a distinctive line of work, physical fingerprinting CAN nodes and determining their location on the bus which we discuss next.

One research direction considered by multiple related works is fingerprinting CAN nodes based on their unique physical layer behavior. CAN physical signaling is influenced by unique characteristics of CAN transceiver chips and even by power supply circuitry. The idea of fingerprinting CAN nodes based on voltage measurements of the CAN differential signal was introduced in [19]. The paper illustrates the concept by applying simple signal processing on a dataset obtained by sampling signals from a CAN bus working at a bit rate of 125 kbps with an oscilloscope at a sample rate of 2 GS/s. The results obtained in this initial work are improved by Choi et al. in [4]. They extract a set of 17 features from the signals sampled at 2.5 GS/s on a 500 kbps CAN bus and use classification algorithms to fingerprint and identify nodes. Using a similar setup, the same authors propose using only voltage data from rising and falling edges of the recorded frames [5]. In a more pragmatic approach, the Viden mechanism proposed in [3] is based on a very low sampling rate (50 kS/s) and its efficiency is demonstrated by a proof of concept implementation. Viden uses multiple measurements of dominant levels sampled at different points during the frame transmission to build voltage profiles for uniquely identifying transmitter nodes.

The automotive industry also showed interest in this type of approaches. A series of papers authored by Kneib et al. comes as a proof of research efforts at Bosch on this subject. In Scission [14], further improvements are made on using voltage signatures for fingerprinting by focusing on characteristics exhibited by signals around rising and falling edges sampled from CAN frame sections following the arbitration field. Further improvements are presented in EASI [16] which is tailored to the capabilities of automotive-grade microcontrollers. The required sampling rate in this case is as low as 2 MS/s when using randomly interleaved sampling. Their more recent proposal, VALID [23], aims to reduce the required sampling rate even more for the purpose of achieving a solution that can be implemented on currently available automotive-grade microcontrollers. In [15], the authors also evaluate the effect of temperature on the accuracy of their proposed mechanisms. The effect of environmental factors, e.g. temperature, voltage, was also considered by authors of [7] in the design of SIMPLE, a voltage-based IDS that accounts for temperature and voltage variations by updating node fingerprints.

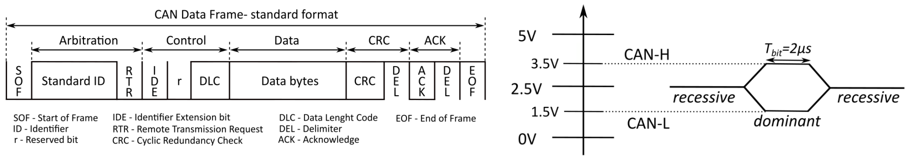
The work in [20] uses features extracted from voltage data to estimate the location of the transmitter for several attack scenarios with adversaries at various locations on the bus. A different approach proposed for adversary localization and isolation by having a bus guardian control relays placed near each node is detailed in [9]. Rumez et al. [21] take another approach and propose the use of time domain reflectometry to evaluate the network structure and estimate node locations. Their approach is based on measuring the network response to a pulse which is sent when the network is offline, i.e., before starting communication. While their results prove the ability to identify disconnected nodes or newly added network nodes, the mechanism is unable to correlate message transmissions to node locations on the bus.

2 CAN Background and Experimental Setup

This section gives a brief background on CAN then proceeds to a detailed presentation of our experimental testbed.

2.1 CAN Background

The CAN protocol is still the most widely used for communication between ECUs found inside contemporary vehicles. Its simple two-wire (CAN-High, CAN-Low) physical layer, support for bit rates up to 1 Mbit/s (a maximum bit rate of 500 kbit/s is employed for in-vehicle application) and maximum payload of 8 bytes make it suitable for a wide range of applications [12], [13]. The standard CAN data frame, depicted in Figure 2 (i) is divided into several main fields: arbitration, control, data, CRC and acknowledgment. The name of the arbitration field comes from its use in the arbitration mechanism which is employed for deciding which node should win the bus in case two or more nodes start transmission at the same time. A dominant bit, i.e. SOF (Start-of-Frame) marks the beginning of the frame followed by the 11 bit frame identifier (ID) which is used to identify CAN messages and, as part of the arbitration field, also contributes to frame arbitration, i.e. lower IDs indicate higher priority. The payload length (DLC) is encoded in the control field. The data field is followed by a 15 bit CRC and the ACK bit which is used to ensure that a transmitted frame was properly received by network nodes. An extended data frame also exists which allows the use of 29 bit IDs with no changes in the payload size. A newer extension of the protocol, i.e. CAN-FD (CAN with Flexible Data-rate), allows the use of bit rates higher than 1 Mbps after the arbitration field along with payloads of up to 64 bytes. Currently available CAN-FD compatible transceivers are capable of bit rates of up to 8 Mbps.



(i) (ii)

Fig. 2. Standard CAN frame format (i) and physical bit representation (ii)

At the physical layer, CAN is implemented as a 2-wire differential line. Data representation at the physical layer is based on two symbols: recessive and dominant. Figure 2 (ii) depicts bit representation at the physical level for high-speed CAN. The dominant state (logical '0'), is reached when the CAN-High and the CAN-Low lines are actively driven by the transceiver. During the dominant state the CAN-H line reaches a voltage of about $3.5V$ while the CAN-L line goes down to $1.5V$. These values may differ (e.g., when working with $3.3V$ levels) and will still properly represent a dominant bit as long as the differential voltage is above $0.9V$. In contrast, the recessive (1) state occurs when the bus is not driven, both lines exhibiting a similar voltage level which is usually around $2.5V$.

2.2 Experimental Setup

An abstract representation of our experimental setup is shown in Figure 3. The CAN bus is constructed from a professional CAN cable with a total length of 5 meters. The bus is divided into 9 segments with lengths of 10-10-50-50-100-100-130-30-20 cm providing 10 connection points in the form of DB9 connectors. Due to space constraints, we defer the pictures with our setup for Appendix A. In Scenario A we used 10 devices, each connected to one of the entry points, while in Scenario B we use 5 legitimate devices connected to the following 5 connection points, A, C, E, G, I and the rest of the points are dedicated for adversaries. This along with some of the distances depicted in the figure are detailed in the next section which addresses the methodology in our work.

The employed network nodes consist of two types of USB-to-CAN devices: 5 x USB-CAN modules from SYSTEC electronic and 2xVN5610A from Vector Informatik. The USB-CAN modules are equipped with a high-speed CAN transceiver capable of transmitting data at up to 1 Mbps, while the VN5610A devices support both high-speed CAN and CAN-FD. In our experiments the bit rate was configured to 500 kbps for all devices considering that this is the recommended data rate for use in high-speed CAN buses in automotive [22]. Periodic data frames were transmitted from the USB-CAN modules using the PcanView, application from SYSTEC electronic and from the VN5610A devices using Vector CANoe environment. For bus monitoring and sample acquisition we used a PicoScope 5444D with four probes connected to the two bus lines, i.e., CAN-Low and CAN-High, on both ends of the bus, close to the termination on each side as shown on the left side of Figure 3. We configured the PicoScope to sample voltage data at 250 MS/s which is the maximum rate when using all 4 channels.

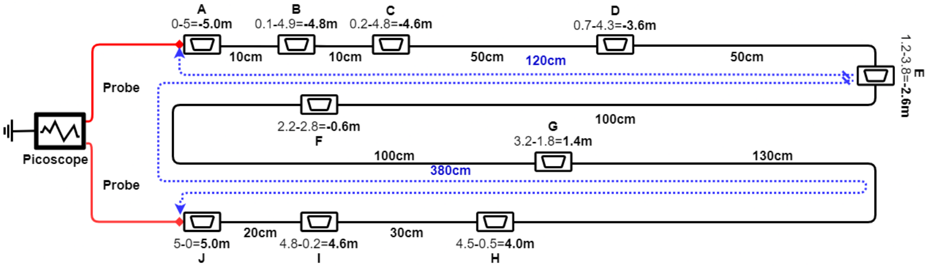


Fig. 3. Abstract view of the experimental setup with nodes connected to the CAN bus in 10 distinct locations (white boxes) and a PicoScope

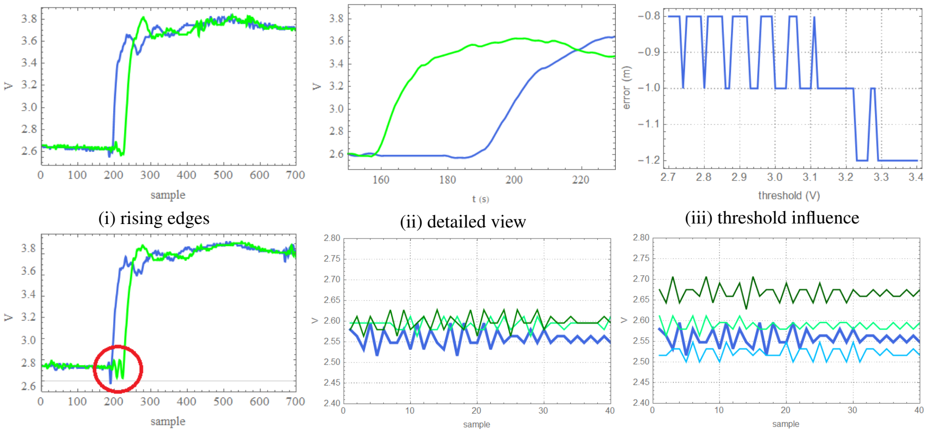
3 Methodology and Results

We now present the proposed methodology for fingerprint extraction and discuss some limitations in existing approaches.

3.1 Concept and Limitations in Previous Approaches

To localize the nodes on the bus we use the difference in propagation time to the left and right probes. The difference in propagation time, referred by us as *differential delay* and denoted as Δ can be used to compute the exact position of each node on the network. The *differential delay* multiplied with the propagation speed, i.e., $5ns/m$, leads to the *differential position* denoted as π which can be also obtained by subtracting the distance to the right probe from the distance to the left probe. Figure 3 depicts the *differential position* of each node. For example, node A is located 0cm from the left probe and 5m from the right probe which results in 0-5=-5m. Node E is located 120cm from the left and 380cm from the right probe which results in 120-380=-260cm, etc. The *differential position* can be immediately converted to the physical position toward the left probe by adding the cable length and dividing by 2, i.e., the physical position to the left probe is $\frac{\pi+\ell}{2}$ where ℓ is the cable length.

The more complicated task is the correct estimation of the propagation delay for which in our previous work [20] we used a threshold based separation that did not cope well with adversarial attacks on the bus. Figure 4 presents some details regarding voltage levels which lead to issues in computing differential delays based on a simple threshold. While the voltage appears to rise sharply on both the left and right sides of the bus (i) the detailed view in (ii) proves that the rising slopes are not perfectly parallel and by very small changes in the threshold used to calculate the differential delay (in the order of $0.1V$ or less) significant changes in the reported distance may occur. As a concrete numerical example, in (iii) we show that localization errors may vary from 80cm up to 1.2m which is very high for a threshold between $2.7V$ and $3.4V$. So ideally, one would choose the lowest possible threshold value. Based on the data that we recorded, setting a threshold as low as $0.2V$ above the expected CAN-H voltage for a recessive state ($2.5V$) which gave the small error was most of the times unusable due to electrical



(i) rising edges

(ii) detailed view

(iii) threshold influence

(iv) anomaly at rising start

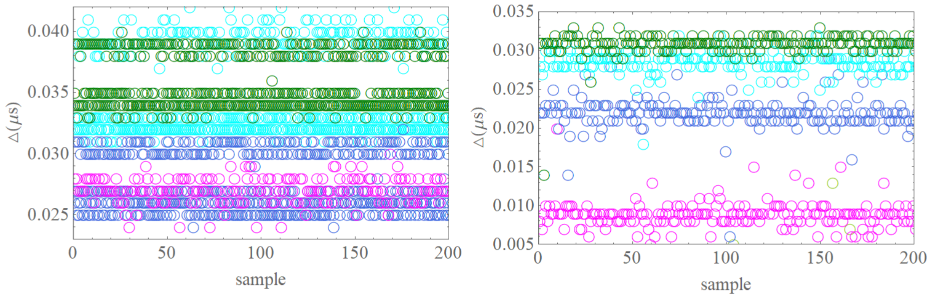
(v) recessive with adversaries

(vi) recessive with temperature

Fig. 4. Voltage values on the left and right rising edges (i), detail view of the rising portion (ii), threshold influence on delay (iii), anomaly before rising time (iv), recessive level variation with adversaries tapping the bus (v) and temperature variations (vi)

noise on the bus. For example, in (iv) we show anomalies during the start of the bit that may greatly affect the computed differential delay (note that the voltage actually dips before the bit starts which would result in a delayed identification of the start when using a threshold), then in (v) we show the recessive voltage (before the starting bit of the same node) for a clean bus (blue) and the same bus with 2 adversaries (green) and 3 adversaries (dark green) while (vi) shows the voltage when the temperature variates between 0°C (light blue), 24°C (blue), 50°C (green) and 60°C (dark green). On a CAN bus, nodes may disconnect when entering in a bus-off state and thus the recessive voltage may change due to innocuous circumstances. Clearly, temperature variations are even more common as components may heat-up while running and a car may operate in various climatic conditions. Fluctuations in the order of $0.2V$ or even more are to be expected so using a simple voltage threshold as we previously used in [20] is not sufficient to cope with adversaries and environmental variations. To achieve resilience to such variations, we need to use the recessive voltage of the bus and subtract it from the threshold. Essentially, this is what we do in CAN-SQUARE and in addition we also use a sliding window w to jump over short-lived fluctuations of the voltage.

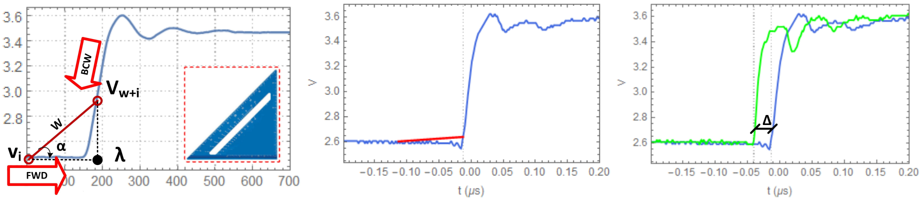
As a more practical example regarding delays, Figure 5 contrasts between the delays computed with a threshold based separation shown in (i), similar to the attempt in [20], and the slope-based separation from CAN-SQUARE (this work) shown in (ii). Note that the purple points significantly overlap with the blue points on the left side of the figure. On the right side of the plot, by using the methodology in this work, the separation becomes clearer with no overlap between the magenta and blue points. Small overlaps remain between the cyan and green points but these correspond to nodes separated by only 20cm of wire. As it will be shown later in our experiments, by averaging over multiple values, separation will be possible even for nodes that are 10cm apart.



(i) threshold based separation

(ii) slope based separation

Fig. 5. Separation based on thresholds (i) and based on slopes (ii) as proposed in this work



(i) start detection concept

(ii) actual start (left channel)

(iii) resulting differential delay

Fig. 6. CAN-SQUARE bit start-time extraction: concept (i), actual start on the left channel (ii) and the resulting differential delay of $0.028\mu s$ (iii)

3.2 Intrusion detection and localization algorithm

Let the voltage samples recorded from the left side of the bus be $\widetilde{v}_l = \{\widetilde{v}_l[0], \widetilde{v}_l[1], \widetilde{v}_l[2],$ ..., $\widetilde{v}_l[b-1]\}$ and the voltage from the right side of the bus be $\widetilde{v}_r = \{\widetilde{v}_r[0], \widetilde{v}_r[1], \widetilde{v}_r[2],$ ..., $\widetilde{v}_r[b-1]\}$ at time $\widetilde{t} = \{\widetilde{t}[0], \widetilde{t}[1], \widetilde{t}[2], ..., \widetilde{t}[b-1]\}$ (the time runs identically for both sides of the bus). As a general notation, we use a tilde to separate between vectors and scalars. We assume a buffer of size b which in our experiments was set at 2-4 thousand samples to cover the duration of 1 bit, e.g., for a 500 kbps CAN the duration of a bit is $2\mu s$ which would require a buffer of 4000 values at a 500MS/s sampling rate (note that we are interested only in the start of the bit so the buffer does not need to cover the entire bit duration). Since the recording is done at some fixed sample rate δ, then the time for sample i is actually $i \times \delta$. Then for a fixed window $w < b$ we define the left and right slopes of the signal as: $\widetilde{s}_l[i] = \frac{\widetilde{v}_l[i]-\widetilde{v}_l[i+w]}{\widetilde{t}[i]-\widetilde{t}[i+w]} = \frac{\widetilde{v}_l[i+w]-\widetilde{v}_l[i]}{w\delta}$, $\widetilde{s}_r[i] = \frac{\widetilde{v}_r[i]-\widetilde{v}_r[i+w]}{\widetilde{t}[i]-\widetilde{t}[i+w]} = \frac{\widetilde{v}_r[i+w]-\widetilde{v}_r[i]}{w\delta}$, $\forall i \in [0..b-w-1]$. That is, computing the slope requires only one subtraction and one division (considering that the sampling rate is fixed and thus $w\delta$ is constant). Having the left and right slopes defined, for a target slope α we define the *fingerprint indexes* λ_l, λ_r as the minimum indexes for which it holds that $s_l[\lambda_l] > \alpha$ and $s_r[\lambda_r] > \alpha$. The *differential delay* of a rising edge is the time difference between the two fingerprint indexes, i.e., $\Delta = \widetilde{t}[\lambda_l] - \widetilde{t}[\lambda_r] = (\lambda_l - \lambda_r)\delta$ and the position of the sender node is $\pi = \frac{\Delta}{5\times10^{-9}}$ (we consider the usual propagation speed of $5ns/m$).

Figure 6 (i) provides a suggestive graphical overview on how the timing fingerprint is constructed. This figure suggests a window w and a slope $\alpha = 1$, i.e., $\tan(45°)$, formed between voltage levels at indexes i and $i + w$. This figure also explains the title of our work which is a reminiscence of a *speed square*, i.e., a triangular marking tool, as shown in the right side of the figure. This simple procedure came out as the best to get a clean cut of the node distances on the bus. In our experiments we use two types of moving squares, i.e., the forward square FWD-SQUARE and the backward BCW-SQUARE which are moving from left to right and right to left respectively (the conducted experiments showed that the BCW-SQUARE method is more precise). Due to space constraints, we defer the pseudocode description of the FWD and BCW-SQUARE to Appendix B. Figures 6 (ii) and (iii) show the actual indexes at which the signal is cut on the left and right channels for a window set to $w = 100$ in our experiments. The actual fingerprint, i.e., the timing difference between the two indexes on the left and right channels, is $\varDelta = 0.028\mu s$. This difference was recorded for a node that is located 20cm from one end of the bus and 480cm from the other. Given a signal propagation speed of $5ns$ per meter this results in theory in a theoretical differential delay of about $23ns$ (the measured $28ns$ in the experiments are very close to the expected value and is in fact a worst case from our experiments).

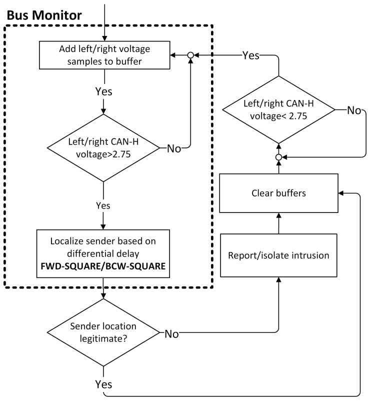
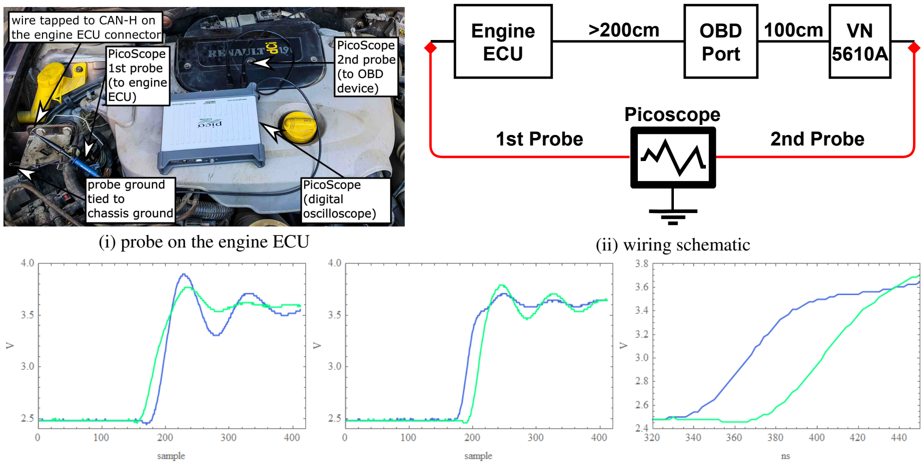

Bus Monitor
Add left/right voltage samples to buffer
Yes
Left/right CAN-H voltage>2.75
No
Yes
Localize sender based on differential delay
FWD-SQUARE/BCW-SQUARE
Sender location legitimate?
No
Yes
Report/isolate intrusion
Clear buffers
Left/right CAN-H voltage< 2.75
No
Yes

Fig. 7. Flowchart of the intrusion localization procedure

Having the description of the localization procedure, the conceptual description of the proposed intrusion detection and localization algorithm easily follows. Figure 7 provides an overview for this. The algorithm continuously adds voltage samples from the bus to a buffer and checks that the CAN-H voltage on the left and right sides did not exceed $2.75V$ (this threshold is selected based on CAN physical layer specification in ISO 14229-2 which set this as the minimum CAN-H voltage during a dominant bit). Once such change occurs, the algorithm proceeds to the forward or backward square algorithms which will localize the nodes. If the recorded location is not a legitimate one, then an intrusion will be signaled (possibly isolated with techniques such as those presented in [9]). Regardless, the buffer is subsequently cleaned and the monitor waits for the voltage to drop below the $2.75V$ which will happen when returning to a recessive state. The monitoring continues in the same fashion. As stated, the pseudocode description of the FWD and BCW-SQUARE can be found in Appendix B.

4 Experimental evaluation

In this section we present the scenarios that we consider and then provide concrete experimental data for each of them.



(i) probe on the engine ECU

(ii) wiring schematic

(iii) delay for engine ECU

(iv) delay for OBD transmitter

(v) detail on delay

Fig. 8. Experiments inside a Renault Megane: (i) probe on the engine ECU, (ii) wiring schematic, (iii) differential delays for the engine ECU, (iv) differential delays for the OBD transmitter and (v) detailed view of delays

4.1 Evaluation scenarios

As an additional step to prove the correctness of our approach, we also verify the methodology inside a real car. Note that the correctness of our approach is supported by the laws of physics which cannot be refuted by practical deployments of CAN buses inside cars. Propagation delays are also used in numerous security applications, e.g., distance-bounding protocols. Still, experimenting inside the car gives convincing arguments in support of our approach. For this purpose we placed two bus taps inside a Renault Megane as depicted in Figures 8 (i) and (ii): one in the vicinity of the engine ECU which called for additional wirings done by us and the other on the OBD port which was already accessible and linked to the engine ECU. To verify that the differential delays are similar, besides the incoming frames from the engine ECU we also inject frames from the OBD port with a VN5610A. Figures 8 (iii) and (iv) show the differential delays from the engine ECU as well as from the VN5610A plugged on the OBD port. Note that the differential delays are identical while the blue and green edges are flipped between (iii) and (iv) since the engine ECU and the VN5610A are placed at opposite sides of the bus. The measured differential delays, detailed in Figure 8 (v), are of around 34ns which corresponds to an interdistance of 6.8m that further translates to a physical distance of 3.4m (as later explained in subsection 4.2). This roughly corresponds to the wire length of slightly more than 2m from the engine ECU, that we could measure with a tape line inside the car, plus the 1m extension of the OBD wire where we placed the second probe.

Since it is much harder to perform insertion and replacement attacks at various distances inside the car due to access difficulties and it is also harder to control temperature variations in the range of 0-60°C, we perform the rest of the data collection on an experimental bus. The attacks and environmental changes that we account for are realistic and the laboratory setup facilitates the collection of a much larger experimental basis. Concretely, we test the proposed mechanism on two distinct network configurations based on [20]: a 10 ECU network in Scenario A and a 5 ECU network in scenario B. Figure 9 shows a schematic representation of ECU locations in each scenario. The first network configuration allows data collection from a larger number of ECUs while the second network configuration allows more insertion points for adversary nodes. We apply our mechanism both on our previous datasets from [20] and on fresh datasets from the newly implemented network (pictured in Appendix A) that follows the same configuration with a professional CAN bus cable. On this new setup we also test the response to multiple adversaries and environmental variations, i.e., temperature changes which are known to influence voltage fingerprints.

Scenario A: Replacement attacks on a large network with 10 ECUs. It is the first legitimate network containing 10 ECUs for which we use our datasets from [20]. This is a somewhat high number of ECUs for a single bus. Note that while more than 100 ECUs are claimed to be present inside some cars, and this is indeed correct, they are always grouped on several CAN buses that may further communicate via a gateway.

Attack scenario A.1: Multiple ECU replacements in the 10 ECUs (identical devices). This represents the first alteration of our clean network in which we consider the malicious (or innocuous action in case that ECU replacement is done by an authorized garage) to be the *replacement* of some existing ECUs which is emulated simply by mixing the devices from the first setup.

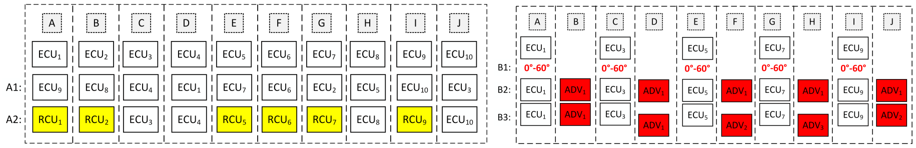
Attack scenario A.2: Multiple ECU replacements in the 10 ECUs (distinct devices). This scenario pushes the limit of the previous by replacing 6 out of the 10 legitimate ECUs from Scenario 1 with distinct devices.

Scenario B: Single or multiple insertions and temperature variations on a smaller network with 5 ECUs. Since this scenario requires multiple measurements as well as open locations in the network, i.e., for the insertion attacks, we will use a smaller network. Note that this is still a realistic number of ECUs since existing reference works such as [7], [14] have physically fingerprinted real cars that had 4-6 ECUs. Also, the results that we obtain hold even for larger number of ECUs as they are comparable to the 10 ECU network in Scenario A.

Attack Scenario B.1: Temperature variations. In this scenario we keep our experimental setup inside a box, at 50°C and 60°C in order to observe the influence of environmental temperature. This scenario is very realistic given the various conditions in which cars, and CAN buses in particular, may operate.

Attack Scenario B.2: Single insertions in the clean network with 5 ECUs (distinct devices). This scenario tries to determine how fingerprinting will be altered by the addition of new ECUs to the bus and thus we consider inserting two distinct devices (one at a time) on the clean network from Scenario B.

Attack Scenario B.3: Multiple insertions in the clean network with 5 ECUs (distinct devices). Each node that is added to the bus will change the impedance which immediately affects propagation timings. Previously, we were only concerned with single adversaries on the bus, now we extend the experiments with data for 2 and 3 adversaries. Such a scenario is less likely, but we need to determine its influence on timings.


A B C D E F G H I J
ECU$_1$ ECU$_2$ ECU$_3$ ECU$_4$ ECU$_5$ ECU$_6$ ECU$_7$ ECU$_8$ ECU$_9$ ECU$_{10}$
A1: ECU$_9$ ECU$_8$ ECU$_4$ ECU$_1$ ECU$_7$ ECU$_6$ ECU$_2$ ECU$_5$ ECU$_{10}$ ECU$_3$
A2: RCU$_1$ RCU$_2$ ECU$_3$ ECU$_4$ RCU$_5$ RCU$_6$ RCU$_7$ ECU$_8$ RCU$_9$ ECU$_{10}$
A B C D E F G H I J
ECU$_1$ ECU$_3$ ECU$_5$ ECU$_7$ ECU$_9$
B1: 0°-60° 0°-60° 0°-60° 0°-60° 0°-60°
B2: ECU$_1$ ADV$_1$ ECU$_3$ ADV$_1$ ECU$_5$ ADV$_1$ ECU$_7$ ADV$_1$ ECU$_9$ ADV$_1$
B3: ECU$_1$ ADV$_1$ ECU$_3$ ADV$_1$ ECU$_5$ ADV$_2$ ECU$_7$ ADV$_3$ ECU$_9$ ADV$_2$

(i) Scenario A

(ii) Scenario B

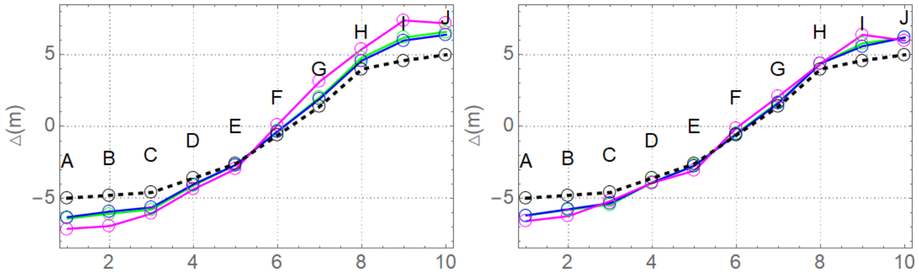
Fig. 9. ECU placement in Scenario A (i) and Scenario B (ii)

4.2 Results

We first prove the robustness of our localization algorithms in front of network modifications, i.e., ECU replacement and position changes, for the large 10 ECU network. Then we proceed to a finer grained analysis against replacement, multiple ECU insertions and temperature variations on the 5 ECU network.

As a general procedure, we quantify the distances between locations π', π'' on the bus as the Euclidean distance between the evaluated locations of the reporting ECUs to the bus ends, i.e., $\mathbb{D}^{\blacklozenge}(\pi', \pi'') = \left\{ \sqrt{(\delta(\pi') - \delta(\pi''))^2} : \forall \pi', \pi'' \in \{A, B, ..., J\} \right\}, \blacklozenge \in \{\text{intra}, \text{inter}\}$. The intra-distances refer to distances between experiments performed on the same (clean) network while inter-distance refer to distances between measurements taken on the clean network when compared to the network affected by adversarial/environmental actions. Whenever $\pi' = \pi''$, i.e., the evaluated positions are the same, the distance represents an intra-distance and whenever $\pi' \neq \pi''$, i.e., the evaluated positions are distinct, the distance represents an inter-distance. The distances $\mathbb{D}^{\blacklozenge}(\pi', \pi'')$ is further computed as the mean of 1000 random experiments and in the following tables we are going to present either the mean interdistance over single rising edges denoted as $\overline{\mathbb{D}}_1$ or the mean value of the medians for 15 consecutive rising edges denoted as $\widetilde{\mathbb{D}}_{15}$. The reason for choosing the median of 15 consecutive values is that an 8 byte CAN frame will have an average of 15 transitions from 0 to 1 (due to the stuffing rule, each 5 consecutive identical bits will be followed by a 6-th that differs). Thus, 15 rising edges will be generally available in each frame to establish the node location.

Important note. Since we work with the differential delays at the two bus ends, the position of each node $\delta(\pi), \forall \pi \in \{A, B, ..., J\}$ is reported in the range of $[-5m, 5m]$ to which some error is added due to cable imperfections, measurement imprecisions and noise that affects our algorithms. The inter-distance $\mathbb{D}^{\blacklozenge}(\pi', \pi'')$ may report values of up to 10 meters plus some measurement error for a 5 meter cable which may seem puzzling but it is nevertheless correct. For example in case of node A placed at -5m and node J at +5m we have $\mathbb{D}^{\text{inter}}(A, J) = \sqrt{(-5-5)^2} = 10m$. The *physical distance* $\mathbb{D}^{\text{phy}}(\pi', \pi'')$ between two nodes is actually half the inter-distance, i.e., $\mathbb{D}^{\text{phy}}(\pi', \pi'') = \frac{\mathbb{D}^{\text{inter}}(\pi', \pi'')}{2}$. This can be easily proved as follows. Consider positions π', π'' and position π' located at distance d'_l from the left side of the bus and d'_r from the right side, while π'' is at d''_l from the left and d''_r from the right. The physical distance between them is $\mathbb{D}^{\text{phy}}(\pi', \pi'') = \sqrt{(d'_l - d''_l)^2}$ and since for our cable of length ℓ we have $d'_l = \ell - d'_r, d''_l = \ell - d''_r$ it also holds that $\mathbb{D}^{\text{phy}}(\pi', \pi'') = \sqrt{(d'_l - d''_l)^2} = \sqrt{(\ell - d'_r - \ell + d''_r)^2} = \sqrt{(d''_r - d'_r)^2} = \sqrt{(d'_r - d''_r)^2}$. The inter-

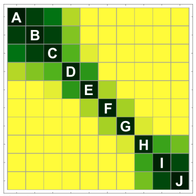


(i) forward square

(ii) backward square

Fig. 10. Reported distances over 3 distinct experiments in Scenario A: clean network (blue), replacements with identical nodes (green) and replacements with distinct nodes (magenta) with the forward (i) and backward (ii) square method

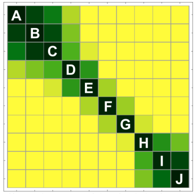
distance however is $\mathbb{D}^{\text{inter}}(\pi', \pi'') = \sqrt{[(d'_l - d'_r) - (d''_l - d''_r)]^2}$ and by substituting d'_r, d''_r for cable of length ℓ we get $\mathbb{D}^{\text{inter}}(\pi', \pi'')=\sqrt{[(d'_l - \ell + d'_l) - (d''_l - \ell + d''_l)]^2}$ $= \sqrt{(2d'_l - 2d''_l)^2} = 2\sqrt{(d'_l - d''_l)^2} = 2\mathbb{D}^{\text{phy}}(\pi', \pi'') \Rightarrow \mathbb{D}^{\text{phy}}(\pi', \pi'') = \frac{\mathbb{D}^{\text{inter}}(\pi', \pi'')}{2}\square$.



(i) clean

$\overline{\mathbb{D}}_1$	A	B	C	D	E	F	G	H	I	J
A	**0.17**	0.42	0.77	2.30	3.48	5.73	7.91	10.61	11.91	12.38
B	0.40	**0.22**	0.43	1.91	3.09	5.33	7.51	10.21	11.52	11.98
C	0.76	0.42	**0.24**	1.57	2.72	4.96	7.17	9.84	11.18	11.62
D	2.30	1.92	1.55	**0.24**	1.18	3.41	5.59	8.28	9.61	10.07
E	3.47	3.11	2.74	1.19	**0.23**	2.22	4.41	7.09	8.42	8.91
F	5.72	5.33	4.99	3.39	2.22	**0.24**	2.17	4.85	6.18	6.65
G	7.90	7.51	7.14	5.65	4.41	2.18	**0.26**	2.64	4.02	4.47
H	10.58	10.18	9.82	8.26	7.10	4.86	2.65	**0.28**	1.34	1.79
I	11.92	11.53	11.16	9.59	8.43	6.16	4.00	1.33	**0.22**	0.48
J	12.38	11.99	11.61	10.08	8.89	6.65	4.45	1.82	0.48	**0.19**

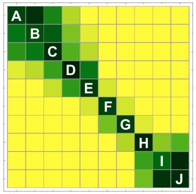
$\widetilde{\mathbb{D}}_{15}$	A	B	C	D	E	F	G	H	I	J
A	**0.03**	0.38	0.76	2.32	3.49	5.75	7.94	10.59	11.92	12.41
B	0.38	**0.05**	0.38	1.93	3.11	5.35	7.56	10.21	11.53	12.02
C	0.76	0.38	**0.08**	1.54	2.72	4.98	7.18	9.83	11.15	11.65
D	2.32	1.93	1.56	**0.09**	1.17	3.42	5.62	8.28	9.59	10.09
E	3.49	3.11	2.72	1.17	**0.09**	2.26	4.45	7.10	8.42	8.92
F	5.75	5.37	4.98	3.42	2.26	**0.08**	2.19	4.85	6.16	6.66
G	7.94	7.55	7.17	5.62	4.44	2.19	**0.08**	2.64	3.97	4.46
H	10.59	10.21	9.83	8.28	7.10	4.84	2.65	**0.05**	1.31	1.81
I	11.91	11.53	11.15	9.60	8.42	6.16	3.97	1.32	**0.09**	0.49
J	12.41	12.02	11.65	10.09	8.92	6.65	4.46	1.81	0.49	**0.02**



(ii) replacement identical devices

$\overline{\mathbb{D}}_1$	A	B	C	D	E	F	G	H	I	J
A	**0.19**	0.45	0.86	2.29	3.47	5.66	7.83	10.51	11.72	12.36
B	0.41	**0.20**	0.49	1.89	3.08	5.28	7.45	10.09	11.31	11.96
C	0.79	0.37	**0.24**	1.52	2.70	4.90	7.08	9.77	10.95	11.61
D	2.31	1.88	1.47	**0.22**	1.16	3.34	5.51	8.18	9.39	10.03
E	3.47	3.06	2.63	1.19	**0.23**	2.17	4.33	7.02	8.20	8.86
F	5.72	5.29	4.88	3.44	2.24	**0.24**	2.10	4.78	5.98	6.62
G	7.90	7.49	7.07	5.63	4.45	2.23	**0.24**	2.57	3.77	4.41
H	10.58	10.14	9.73	8.27	7.13	4.94	2.74	**0.24**	1.11	1.77
I	11.90	11.49	11.06	9.61	8.45	6.28	4.09	1.41	**0.27**	0.46
J	12.37	11.98	11.52	10.08	8.89	6.72	4.55	1.88	0.69	**0.20**

$\widetilde{\mathbb{D}}_{15}$	A	B	C	D	E	F	G	H	I	J
A	**0.04**	0.41	0.85	2.27	3.46	5.68	7.83	10.52	11.72	12.39
B	0.38	**0.05**	0.46	1.89	3.08	5.30	7.44	10.13	11.33	12.00
C	0.76	0.34	**0.10**	1.51	2.69	4.91	7.07	9.76	10.95	11.62
D	2.31	1.90	1.47	**0.09**	1.14	3.36	5.50	8.20	9.40	10.07
E	3.49	3.08	2.63	1.21	**0.09**	2.18	4.34	7.02	8.23	8.89
F	5.75	5.34	4.89	3.47	2.29	**0.11**	2.08	4.77	5.98	6.64
G	7.94	7.53	7.08	5.67	4.48	2.26	**0.12**	2.57	3.77	4.45
H	10.59	10.18	9.74	8.31	7.14	4.91	2.76	**0.09**	1.13	1.79
I	11.91	11.50	11.06	9.64	8.45	6.23	4.08	1.40	**0.20**	0.47
J	12.41	11.99	11.55	10.13	8.94	6.73	4.58	1.89	0.69	**0.02**

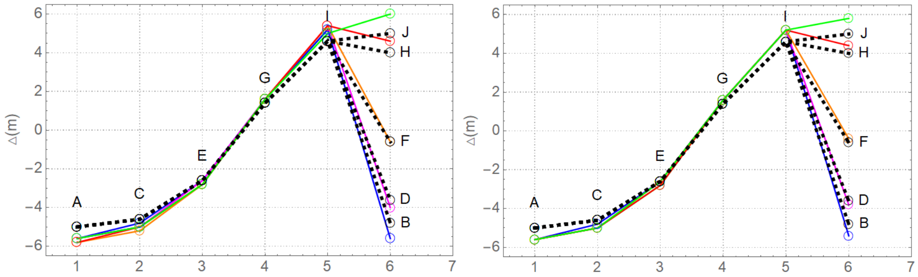


(iii) replacement distinct devices

$\overline{\mathbb{D}}_1$	A	B	C	D	E	F	G	H	I	J
A	**0.44**	0.33	0.98	2.29	3.12	6.11	8.30	10.60	12.57	12.14
B	0.79	**0.50**	0.61	1.88	2.73	5.71	7.93	10.20	12.22	11.76
C	1.16	0.83	**0.31**	1.54	2.36	5.37	7.55	9.85	11.84	11.39
D	2.71	2.37	1.29	**0.22**	0.82	3.79	6.00	8.29	10.29	9.83
E	3.86	3.57	2.47	1.21	**0.40**	2.60	4.82	7.12	9.10	8.68
F	6.13	5.79	4.72	3.44	2.61	**0.44**	2.58	4.86	6.89	6.40
G	8.34	8.00	6.90	5.64	4.79	1.82	**0.41**	2.68	4.69	4.21
H	10.97	10.64	9.59	8.30	7.48	4.49	2.28	**0.27**	1.98	1.54
I	12.30	11.96	10.90	9.64	8.81	5.79	3.62	1.33	**0.68**	0.27
J	12.77	12.43	11.39	10.11	9.26	6.28	4.08	1.78	0.34	**0.29**

$\widetilde{\mathbb{D}}_{15}$	A	B	C	D	E	F	G	H	I	J
A	**0.36**	0.11	1.00	2.27	3.13	6.10	8.31	10.60	12.59	12.15
B	0.75	**0.43**	0.62	1.88	2.75	5.72	7.92	10.21	12.20	11.76
C	1.13	0.82	**0.23**	1.51	2.37	5.33	7.54	9.83	11.82	11.38
D	2.68	2.36	1.31	**0.09**	0.81	3.78	5.98	8.28	10.27	9.82
E	3.86	3.52	2.48	1.21	**0.36**	2.60	4.80	7.11	9.09	8.65
F	6.11	5.79	4.74	3.47	2.61	**0.35**	2.54	4.85	6.83	6.40
G	8.31	7.98	6.94	5.67	4.81	1.84	**0.35**	2.66	4.63	4.21
H	10.96	10.63	9.59	8.32	7.47	4.49	2.29	**0.08**	1.99	1.55
I	12.29	11.97	10.91	9.64	8.78	5.81	3.61	1.31	**0.67**	0.23
J	12.78	12.45	11.41	10.14	9.28	6.30	4.10	1.80	0.18	**0.26**

Fig. 11. Reported distances over 3 experiments in Scenarios A.1 and A.2: clean network, replacements with the identical devices and replacement with distinct devices



(i) TJA0

(ii) TJA1

Fig. 12. Insertion attacks: TJA0 (i) vs. TJA1 (ii) with the BCW-SQUARE method

Scenario A. In Figure 10 we present the intra and inter-distances results for the clean 10 ECU network (blue) in as well as for replacements with identical devices (green) and replacement with distinct devices (magenta). The black dotted line denotes the real position of the nodes. The backward square has a slightly improved accuracy. For replacements with the same devices there is almost no change in the reported distances. When replacing with distinct devices the change becomes visible and the locations may shift with at most 30cm. ECU replacement is a rare procedure inside a vehicle and if such change occurs, it will likely be done with identical devices. In Figure 11 we present the intra and inter-distance between the three configurations as numerical values and also as heatmap for the values on the left. Additional plots for replacements in Scenario A can be found in Appendix C.

To establish a more concrete view on the accuracy of the localization methods, in Tables 1 and 2 we present numerical data on the estimated distances as medians $\mathbf{M}$, means μ over all the reported distances and errors with respect to the true location on the network. Note that as the impedance of the cable does affect the propagation speed which we consider to be fixed at a reference of $5ns/m$, and thus the reported distance may vary based on the cable impedance, we expect for such variations to be present. The fact that the errors at the bus ends, i.e., locations A vs. J are symmetric prove that our method has very good precision. Finally, the accuracy can be corrected by interpolating with the expected error but this would be out of scope.

Scenario B. In Scenario B we investigate both single and multiple node insertions attacks as well as the influence of temperature variations. First, in Figure 12 we present the influence of single node insertions based on applying the new methodology on our past dataset from [20]. Insertions are performed with two distinct transceivers TJA0 and TJA1. For brevity, we defer part of the numerical data for Scenario B to Appendix D.

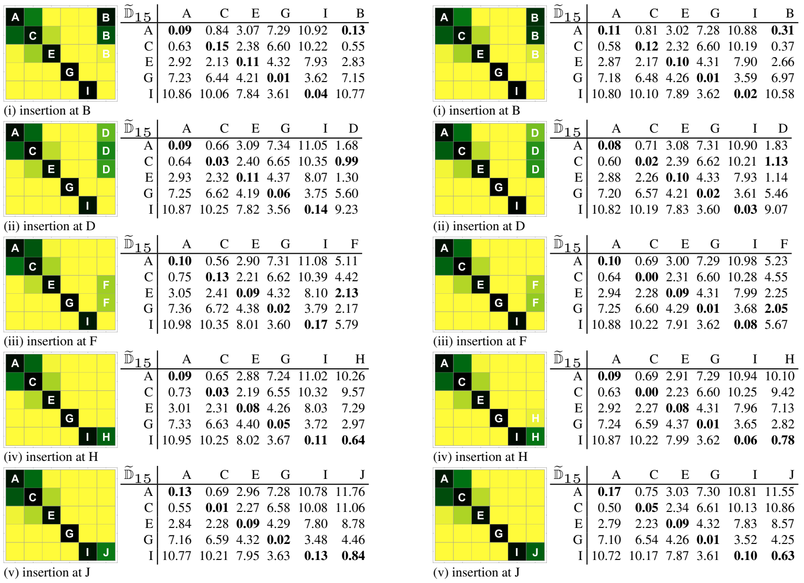
Figure 13 presents these in terms of inter-distances when using the backward square method. The first device, i.e., TJA0, has a slightly larger effect on the distances but the results are close. This shows that off-line calibration during production with innocuous adversarial devices may be useful in calibrating the detection algorithm for future attacks by unknown devices. It can be easily seen from the heatmaps that the adversary device easily positions close to the target node while it is still possible to distinguish it from the legitimate node.

Table 1. Scenarios A.1 and A.2 - node replacements FWD-SQUARE $\alpha = 2.5, w = 100$

Scenario	A	err.	B	err.	C	err.	D	err.	E	err.	F	err.	G	err.	H	err.	I	err.	J	err.
clean ntw. ($\mathbf{M}$)	-6.40	1.40	-6.00	1.20	-5.80	1.20	-4.20	0.60	-2.60	0.00	-0.40	0.20	2.00	0.60	4.80	0.80	6.20	1.60	6.60	1.60
clean ntw. (μ)	-6.42	1.42	-6.09	1.29	-5.79	1.19	-4.10	0.50	-2.68	0.08	-0.31	0.29	2.04	0.64	4.80	0.80	6.20	1.60	6.60	1.60
replacement-same ($\mathbf{M}$)	-6.40	1.40	-6.00	1.20	-5.60	1.00	-4.00	0.40	-2.60	0.00	-0.40	0.20	2.00	0.60	4.60	0.60	6.00	1.40	6.40	1.40
replacement-same (μ)	-6.32	1.32	-5.94	1.14	-5.64	1.04	-4.03	0.43	-2.67	0.07	-0.33	0.27	1.94	0.54	4.60	0.60	6.00	1.40	6.40	1.40
replacement-distinct ($\mathbf{M}$)	-7.20	2.20	-7.00	2.20	-6.00	1.40	-4.40	0.80	-3.00	0.40	0.20	0.80	3.20	1.80	5.40	1.40	7.40	2.80	7.20	2.20
replacement-distinct (μ)	-7.14	2.14	-6.95	2.15	-6.07	1.47	-4.37	0.77	-2.96	0.36	0.13	0.73	3.15	1.75	5.40	1.40	7.40	2.80	7.20	2.20

Table 2. Scenarios A.1 and A.2 - node replacements BCW-SQUARE $\alpha = 1, w = 25$

Scenario	A	err.	B	err.	C	err.	D	err.	E	err.	F	err.	G	err.	H	err.	I	err.	J	err.
clean ntw. ($\mathbf{M}$)	-6.20	1.20	-5.80	1.00	-5.40	0.80	-3.80	0.20	-2.80	0.20	-0.40	0.20	1.80	0.40	4.40	0.40	5.80	1.20	6.20	1.20
clean ntw. (μ)	-6.21	1.21	-5.81	1.01	-5.45	0.85	-3.89	0.29	-2.71	0.11	-0.47	0.13	1.72	0.32	4.40	0.40	5.80	1.20	6.20	1.20
replacement-same ($\mathbf{M}$)	-6.20	1.20	-5.80	1.00	-5.40	0.80	-4.00	0.40	-2.80	0.20	-0.60	0.00	1.60	0.20	4.40	0.40	5.60	1.00	6.20	1.20
replacement-same (μ)	-6.20	1.20	-5.77	0.97	-5.35	0.75	-3.91	0.31	-2.73	0.13	-0.54	0.06	1.64	0.24	4.40	0.40	5.60	1.00	6.20	1.20
replacement-distinct ($\mathbf{M}$)	-6.60	1.60	-6.20	1.40	-5.20	0.60	-4.00	0.40	-3.00	0.40	0.00	0.60	2.00	0.60	4.40	0.40	6.40	1.80	6.00	1.00
replacement-distinct (μ)	-6.60	1.60	-6.25	1.45	-5.20	0.60	-3.92	0.32	-3.08	0.48	-0.09	0.51	2.11	0.71	4.40	0.40	6.40	1.80	6.00	1.00



(i) insertion at B

$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I	B
A	**0.09**	0.84	3.07	7.29	10.92	**0.13**
C	0.63	**0.15**	2.38	6.60	10.22	0.55
E	2.92	2.13	**0.11**	4.32	7.93	2.83
G	7.23	6.44	4.21	**0.01**	3.62	7.15
I	10.86	10.06	7.84	3.61	**0.04**	10.77

(ii) insertion at D

$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I	D
A	**0.09**	0.66	3.09	7.34	11.05	1.68
C	0.64	**0.03**	2.40	6.65	10.35	**0.99**
E	2.93	2.32	**0.11**	4.37	8.07	1.30
G	7.25	6.62	4.19	**0.06**	3.75	5.60
I	10.87	10.25	7.82	3.56	**0.14**	9.23

(iii) insertion at F

$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I	F
A	**0.10**	0.56	2.90	7.31	11.08	5.11
C	0.75	**0.13**	2.21	6.62	10.39	4.42
E	3.05	2.41	**0.09**	4.32	8.10	**2.13**
G	7.36	6.72	4.38	**0.02**	3.79	2.17
I	10.98	10.35	8.01	3.60	**0.17**	5.79

(iv) insertion at H

$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I	H
A	**0.09**	0.65	2.88	7.24	11.02	10.26
C	0.73	**0.03**	2.19	6.55	10.32	9.57
E	3.01	2.31	**0.08**	4.26	8.03	7.29
G	7.33	6.63	4.40	**0.05**	3.72	2.97
I	10.95	10.25	8.02	3.67	**0.11**	**0.64**

(v) insertion at J

$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I	J
A	**0.13**	0.69	2.96	7.28	10.78	11.76
C	0.55	**0.01**	2.27	6.58	10.08	11.06
E	2.84	2.28	**0.09**	4.29	7.80	8.78
G	7.16	6.59	4.32	**0.02**	3.48	4.46
I	10.77	10.21	7.95	3.63	**0.13**	**0.84**

(i) insertion at B

$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I	B
A	**0.11**	0.81	3.02	7.28	10.88	**0.31**
C	0.58	**0.12**	2.32	6.60	10.19	0.37
E	2.87	2.17	**0.10**	4.31	7.90	2.66
G	7.18	6.48	4.26	**0.01**	3.59	6.97
I	10.80	10.10	7.89	3.62	**0.02**	10.58

(ii) insertion at D

$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I	D
A	**0.08**	0.71	3.08	7.31	10.90	1.83
C	0.60	**0.02**	2.39	6.62	10.21	**1.13**
E	2.88	2.26	**0.10**	4.33	7.93	1.14
G	7.20	6.57	4.21	**0.02**	3.61	5.46
I	10.82	10.19	7.83	3.60	**0.03**	9.07

(iii) insertion at F

$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I	F
A	**0.10**	0.69	3.00	7.29	10.98	5.23
C	0.64	**0.00**	2.31	6.60	10.28	4.55
E	2.94	2.28	**0.09**	4.31	7.99	2.25
G	7.25	6.60	4.29	**0.01**	3.68	**2.05**
I	10.88	10.22	7.91	3.62	**0.08**	5.67

(iv) insertion at H

$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I	H
A	**0.09**	0.69	2.91	7.29	10.94	10.10
C	0.63	**0.00**	2.23	6.60	10.25	9.42
E	2.92	2.27	**0.08**	4.31	7.96	7.13
G	7.24	6.59	4.37	**0.01**	3.65	2.82
I	10.87	10.22	7.99	3.62	**0.06**	**0.78**

(v) insertion at J

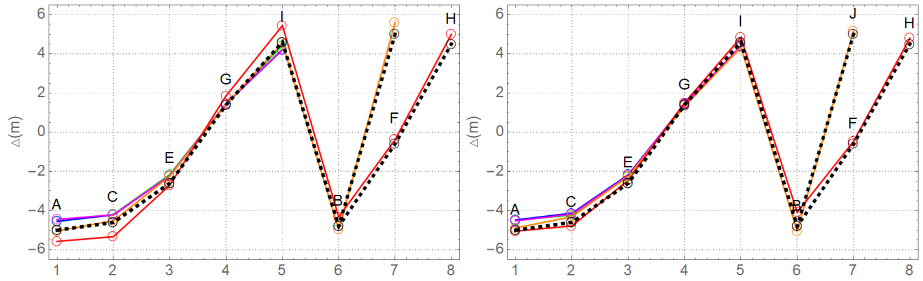
$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I	J
A	**0.17**	0.75	3.03	7.30	10.81	11.55
C	0.50	**0.05**	2.34	6.61	10.13	10.86
E	2.79	2.23	**0.09**	4.32	7.83	8.57
G	7.10	6.54	4.26	**0.01**	3.52	4.25
I	10.72	10.17	7.87	3.61	**0.10**	**0.63**

Fig. 13. Reported inter-distances in case of the node insertion attack from Scenario B.2 with TJA0 (left) and TJA1 (right)

We now extend these experiments with new ones in which we account for temperature variations and multiple adversaries. In Figure 14 we present the reported distances over the 5 distinct experiments: clean network (blue), heated 50°C (blue), heated 60°C (magenta), 2 adversaries (orange) and 3 adversaries (red). The black dotted line denotes the real position of the nodes. Figure 15 shows the inter and intra-distances both as numerical values and heatmaps with the more effective backward square method.

5 Conclusions

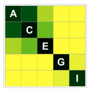
The methodology proposed in this work is very simple and extremely effective in localizing nodes on the CAN bus. Since a single rising edge is sufficient and one frame



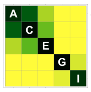
(i) forward square

(ii) backward square

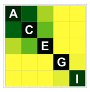
Fig. 14. Reported distances over 5 distinct experiments in Scenario B: clean network (blue), heated 50C (blue), heated 60C (magenta), 2 adversaries (orange) and 3 adversaries (red) with the forward (i) and backward (ii) square method



$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I
A	**0.31**	0.52	2.23	6.12	8.88
C	0.53	**0.13**	1.72	5.65	8.41
E	2.22	1.71	**0.06**	3.92	6.67
G	6.14	5.64	3.91	**0.09**	2.78
I	8.90	8.37	6.65	2.75	**0.33**

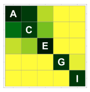


$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I
A	**0.27**	0.58	2.21	6.18	8.75
C	0.61	**0.08**	1.73	5.68	8.27
E	2.30	1.63	**0.06**	3.95	6.52
G	6.22	5.55	3.93	**0.05**	2.61
I	8.96	8.31	6.69	2.73	**0.33**

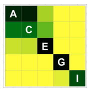


$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I
A	**0.24**	0.56	2.20	6.19	9.03
C	0.63	**0.11**	1.69	5.68	8.52
E	2.34	1.69	**0.04**	3.95	6.79
G	6.26	5.60	3.94	**0.03**	2.89
I	9.01	8.33	6.71	2.72	**0.39**

(i) inter-distances clean network, 0C, 60C

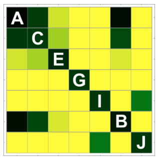


$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I
A	**0.22**	0.39	2.33	5.80	8.91
C	0.72	**0.41**	1.83	5.32	8.41
E	2.45	2.07	**0.18**	3.57	6.72
G	6.37	5.99	3.80	**0.35**	2.76
I	9.11	8.74	6.58	3.04	**0.37**

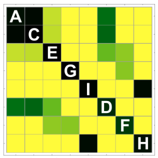


$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I
A	**0.26**	0.20	2.19	6.17	9.39
C	0.77	**0.72**	1.68	5.68	8.89
E	2.50	2.44	**0.03**	3.95	7.17
G	6.40	6.35	3.95	**0.04**	3.26
I	9.15	9.09	6.71	2.71	**0.51**

(ii) inter-distances 2 and 3 adversaries from clean network



$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I	B	J
A	**0.03**	0.39	2.55	6.05	9.15	**0.09**	9.94
C	0.37	**0.37**	2.15	5.68	8.76	0.49	9.52
E	2.56	2.18	**0.27**	3.51	6.57	2.65	7.36
G	6.02	5.69	3.47	**0.36**	3.09	6.14	3.85
I	9.16	8.74	6.61	3.07	**0.36**	9.22	**0.81**
B	0.09	0.47	2.63	6.11	9.22	**0.17**	10.04
J	9.96	9.55	7.35	3.88	0.79	10.02	**0.37**



$\widetilde{\mathbb{D}}_{15}$	A	C	E	G	I	D	F	H
A	**0.10**	0.06	2.44	6.44	9.67	0.72	4.44	9.61
C	0.05	**0.00**	2.39	6.39	9.62	**0.66**	4.42	9.56
E	2.46	2.40	**0.00**	4.00	7.22	1.70	2.01	7.16
G	6.45	6.39	3.99	**0.00**	3.22	5.71	**1.97**	3.17
I	9.67	9.62	7.22	3.22	**0.03**	8.94	5.20	**0.07**
D	0.73	0.66	1.72	5.72	8.94	**0.28**	3.74	8.87
F	4.47	4.41	2.02	1.96	5.19	3.74	**0.38**	5.13
H	9.64	9.56	7.17	3.17	0.08	8.89	5.16	**0.11**

(iii) intra-distances 2 and 3 adversaries

Fig. 15. Reported distances over 3 experiments in Scenario B.1 and B.3: (i) clean network and temperature variations, (ii) inter-distances to the clean network with 2, 3 adversaries and (iii) intra-distances for 2, 3 adversaries

carries more than a dozen such edges, localization can be done after a single frame with extremely high accuracy. The correctness of our approach is confirmed by data collection in a real car where we used two probes: one connected near the OBD port and the other near the engine ECU where we did a minor modification by slightly extending the wire from the existing connector. Further experiments performed on a realistic laboratory setup suggest decimeter level precision with slight overlaps only between nodes that are 10cm apart. This short localization range clearly sets room for physical inspection of the exact device that is responsible for injecting frames on the bus. The proposed method seems to be highly resilient to temperature variations despite voltage changes on the bus. The computational overhead is also insignificant, the only possible limitation is the required high sampling rate but this is clearly achievable with modern signal processing devices.

References

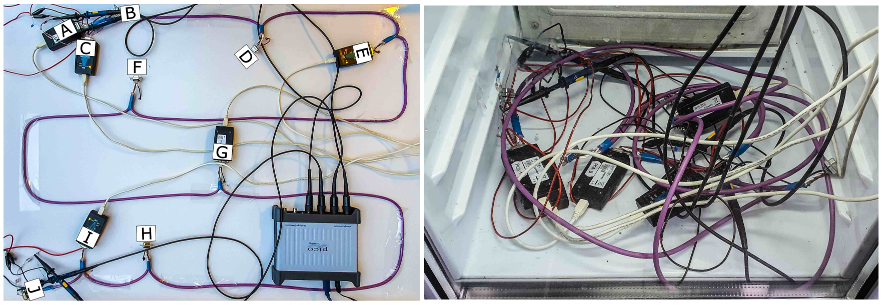
1. AUTOSAR: Specification of Secure Onboard Communication, 4.3.1 edn. (2017)
2. Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., Kohno, T., et al.: Comprehensive Experimental Analyses of Automotive Attack Surfaces. In: USENIX Security Symposium. San Francisco (2011)
3. Cho, K.T., Shin, K.G.: Viden: Attacker Identification on In-Vehicle Networks. In: ACM SIGSAC Conf. on Computer and Communications Security. pp. 1109–1123. ACM (2017)
4. Choi, W., Jo, H.J., Woo, S., Chun, J.Y., Park, J., Lee, D.H.: Identifying ECUs Using Inimitable Characteristics of Signals in Controller Area Networks. IEEE Transactions on Vehicular Technology **67**(6), 4757–4770 (June 2018)
5. Choi, W., Joo, K., Jo, H.J., Park, M.C., Lee, D.H.: VoltageIDS: Low-Level Communication Characteristics for Automotive Intrusion Detection System. IEEE Transactions on Information Forensics and Security (2018)
6. Dagan, T., Wool, A.: Parrot, a software-only anti-spoofing defense system for the CAN bus. ESCAR EUROPE p. 34 (2016)
7. Foruhandeh, M., Man, Y., Gerdes, R., Li, M., Chantem, T.: SIMPLE: Single-frame based physical layer identification for intrusion detection and prevention on in-vehicle networks. In: Proc. of 35th Annual Computer Security Applications Conference. pp. 229–244 (2019)
8. Fröschle, S., Stühring, A.: Analyzing the capabilities of the CAN attacker. In: European Symposium on Research in Computer Security. pp. 464–482. Springer (2017)
9. Groza, B., Popa, L., Murvay, P.S., Yuval, E., Shabtai, A.: CANARY - a reactive defense mechanism for Controller Area Networks based on Active RelaYs. In: 30th USENIX Security Symposium (2021)
10. Humayed, A., Li, F., Lin, J., Luo, B.: CANSentry: Securing CAN-Based Cyber-Physical Systems against Denial and Spoofing Attacks. In: European Symposium on Research in Computer Security. pp. 153–173. Springer (2020)
11. Humayed, A., Luo, B.: Using ID-Hopping to Defend Against Targeted DoS on CAN. In: Intl. Workshop on Safe Control of Connected and Autonomous Vehicles. p. 19–26. ACM (2017)
12. ISO: 11898-1–Road vehicles–Controller area network (CAN)–Part 1: Data link layer and physical signalling. Tech. rep., International Organization for Standardization (2015)
13. ISO: 11898-2, Road vehicles Controller area network (CAN) Part 2: High-speed medium access unit. Tech. rep., International Organization for Standardization (2016)
14. Kneib, M., Huth, C.: Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 787–800. ACM (2018)
15. Kneib, M., Schell, O., Huth, C.: On the Robustness of Signal Characteristic-Based Sender Identification. arXiv preprint arXiv:1911.09881 (2019)
16. Kneib, M., Schell, O., Huth, C.: EASI: Edge-based sender identification on resource-constrained platforms for automotive networks. In: Network and Distributed System Security Symposium (NDSS). pp. 1–16 (2020)
17. Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., et al.: Experimental security analysis of a modern automobile. In: Security and Privacy (SP), 2010 IEEE Symposium on. pp. 447–462. IEEE (2010)
18. Miller, C., Valasek, C.: Adventures in automotive networks and control units. DEF CON **21**, 260–264 (2013)
19. Murvay, P.S., Groza, B.: Source Identification Using Signal Characteristics in Controller Area Networks. IEEE Signal Process. Lett. **21**(4), 395–399 (2014)
20. Murvay, P.S., Groza, B.: TIDAL-CAN: Differential Timing based Intrusion Detection And Localization for Controller Area Network. IEEE Access **8**, 68895–68912 (2020)

21. Rumez, M., Dürrwang, J., Brecht, T., Steinshorn, T., Neugebauer, P., Kriesten, R., Sax, E.: CAN Radar: Sensing Physical Devices in CAN Networks based on Time Domain Reflectometry (2019)
22. SAE: J2284-3 High-Speed CAN (HSC) for Vehicle Applications at 500 KBPS. Standard, SAE International (Nov 2016)
23. Schell, O., Kneib, M.: VALID: Voltage-Based Lightweight Intrusion Detection for the Controller Area Network. 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) pp. 225–232 (2020)
24. Wu, W., Kurachi, R., Zeng, G., Matsubara, Y., Takada, H., Li, R., Li, K.: IDH-CAN: A Hardware-Based ID Hopping CAN Mechanism With Enhanced Security for Automotive Real-Time Applications. IEEE Access **6**, 54607–54623 (2018)
25. Ying, X., Bernieri, G., Conti, M., Poovendran, R.: TACAN: Transmitter authentication through covert channels in controller area networks. In: Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems. pp. 23–34. ACM (2019)

Appendix A - Experimental setup

Figure 16 (i) provides a depiction of our newly built experimental setup which uses an industry grade CAN bus cable. The bus is terminated at each end by a split termination as commonly employed in industry applications with two 60Ω resistors in series (totaling 120Ω) and a capacitor of $10nF$ to remove noise.

To avoid overloading the picture, only 5 devices are connected to the bus which corresponds to the clean network in Scenario B. Figure 16 (ii) shows the network placed inside the refrigerator where it was kept for 1 hour. We intentionally placed the cable and devices in the refrigerator with no attempt to preserve the bus geometry as in the original setup. Somewhat surprising for us, even if the geometry of the bus was changed drastically and the temperature dropped from room temperature 24°C to 0°C, the impact on the reported lengths was insignificant (variations in the order of several centimeters at most). To record data at higher temperature, the clean setup was placed inside a sealed box to avoid heat dissipation and 4 hair-driers were used to heat it for 30 minutes at 50°C and 60°C.



(i) clean network setup

(ii) inside the refrigerator at 0°C

Fig. 16. The clean network (i) and the network dropped inside a refrigerator at 0°C (ii)

Appendix B - BCW and FWD-SQUARE algorithms

Algorithm 1 presents the bus monitor which reads voltage samples on CAN-H to the left and right sides of the bus v_l, v_r and appends them to the buffers $\widetilde{v}_l, \widetilde{v}_r$ (lines 2-3) until a threshold τ is exceeded on both side (line 5). The threshold τ was set to $2.75V$ which is the minimum acceptable dominant voltage on CAN-H according to ISO specifications. When this threshold is met, the FWD or BCW functions extract the time of the rising edge to the left and right of the bus, i.e., t_l, t_r, and the position π is computed (lines 6-8).

Algorithms 2, 3 present the FWD and BCW functions. The FWD-SQUARE function proceeds from the left to the end of the array (indexes 0 to $b - 1$) until the slope exceeds the value of α (lines 3-4). The BCW-SQUARE function first proceeds from the left to right until the voltage reaches the threshold τ to avoid a start on a bit plateau (line 3). Then the index is decremented until the slope drops below the value of α (line 5).

Algorithm 1 Bus Monitor

```
1:  procedure MonitorLocation($v_l, v_r, t$)
2:      $\widetilde{v}_l \leftarrow \mathrm{add}(\widetilde{v}_l, v_l)$
3:      $\widetilde{v}_r \leftarrow \mathrm{add}(\widetilde{v}_r, v_r)$
4:      $\widetilde{t} \leftarrow \mathrm{add}(\widetilde{t}, t)$
5:      if $v_l \geq \tau \wedge v_r \geq \tau$ then
6:          $t_l \leftarrow \mathrm{SQUARE}(\widetilde{v}_l, \widetilde{t}, w, \alpha)$
7:          $t_r \leftarrow \mathrm{SQUARE}(\widetilde{v}_r, \widetilde{t}, w, \alpha)$
8:          $\pi \leftarrow (t_l - t_r) \times (5 \times 10^{-9} s/m)^{-1}$
9:      else
10:         $\pi \leftarrow \perp$
11:     end if
12:     return $\pi$
13: end procedure
```

Algorithm 2 FWD SQUARE

```
1: function FWDSQUARE($\widetilde{v}, \widetilde{t}, w, \alpha$)
2:     $i \leftarrow 0$
3:     while $(\widetilde{v}[i] - \widetilde{v}[i - w])/w\delta < \alpha$ do
4:         $i \leftarrow i + 1$
5:     end while
6:     return $\widetilde{t}[i]$
7: end function
```

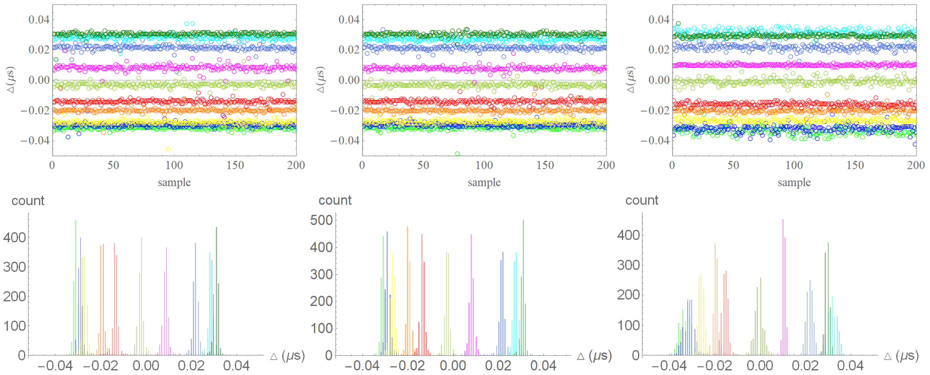
Algorithm 3 BCW SQUARE

```
1: function BCWSQUARE($\widetilde{v}, \widetilde{t}, w, \alpha$)
2:     $i \leftarrow 0$
3:     while $\widetilde{v}[i] < \tau$ do $i \leftarrow i + 1$
4:     end while
5:     while $(\widetilde{v}[i] - \widetilde{v}[i - w])/w\delta > \alpha$ do
6:         $i \leftarrow i - 1$
7:     end while
8:     return $\widetilde{t}[i]$
9: end function
```

Appendix C - Complementary data regarding distances

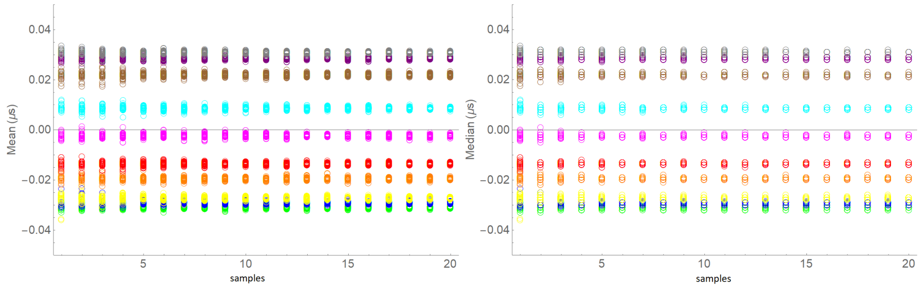
In Figure 17 we also present the raw distances and their histogram distributions as computed for Scenario A for the 10 ECUs. Note that there are overlaps between the first three and the last two devices, but these are separated by only 10cm and respectively 20cm of wire. This is an extremely small distance and even so, the devices can be distinguished over multiple samples.

Figure 18 shows the convergence of the mean values in contrast to the median values with the number of samples. It can be easily seen that the median value converges faster, generally a dozen samples being sufficient to establish the location and these can be extracted from a single frame. The plots are for the BCW-SQUARE method applied on the nodes in Scenario B. The FWD-SQUARE method has lesser accuracy as previously discussed.



(i) Scenario A (clean network) (ii) Scenario A (replacements, same) (iii) Scenario A (replacements, distinct)

Fig. 17. Reported distances for the 10 devices in Scenario A and their histogram distributions



(i) Mean (ii) Median

Fig. 18. Convergence of mean (i) and median (ii) values toward the real distance

Appendix D - Additional numerical data for Scenario B

Tables 3 and 4 give the numerical values as medians $\mathbf{M}$ and means μ over all the collected samples for each node with the forward and backward square methods. The backward square method is more accurate.

Tables 5 and 6 provide the true distances along with the resulting errors. Again, note that since no cable has exactly the $5ns/m$ propagation speed, small variations are expected. The results clearly indicate that the professional CAN bus cable has lower propagation delays and the distances appear smaller than in the previous experiments. The FWD-SQUARE provided less accuracy and we have attempted a software interpolation to increase the sampling rate by 2x-8x but the benefits were little, the BCW-SQUARE remaining still more accurate.

Interestingly, the distances are almost unaffected by temperature variations. The effects of 2 adversaries are similarly low, only when 3 adversaries are connected to the bus the distances are more visibly affected. Such a scenario with 3 adversaries would be less likely on an in-vehicle bus.

Table 3. Scenario B.2 - single insertions FWD-SQUARE $\alpha = 2, w = 100$

Scenario	A	err.	C	err.	E	err.	G	err.	I	err.
clean ($\mathbf{M}$)	-6.00	1.00	-5.40	0.80	-2.80	0.20	1.80	0.40	5.60	1.00
clean (μ)	-5.95	0.95	-5.35	0.75	-2.75	0.15	1.80	0.40	5.67	1.07
TJA0-B ($\mathbf{M}$)	-6.00	1.00	-5.20	0.60	-2.60	0.00	2.00	0.60	5.80	1.20
TJA0-B (μ)	-5.94	0.94	-5.21	0.61	-2.54	0.06	1.95	0.55	5.74	1.14
TJA0-D ($\mathbf{M}$)	-6.20	1.20	-5.60	1.00	-2.40	0.20	2.00	0.60	6.00	1.40
TJA0-D (μ)	-6.19	1.19	-5.61	1.01	-2.42	0.18	2.09	0.69	5.94	1.34
TJA0-F ($\mathbf{M}$)	-6.20	1.20	-5.60	1.00	-3.00	0.40	2.20	0.80	6.00	1.40
TJA0-F (μ)	-6.23	1.23	-5.67	1.07	-3.01	0.41	2.10	0.70	6.03	1.43
TJA0-H ($\mathbf{M}$)	-6.20	1.20	-5.60	1.00	-3.00	0.40	1.60	0.20	6.00	1.40
TJA0-H (μ)	-6.13	1.13	-5.58	0.98	-3.00	0.40	1.51	0.11	5.94	1.34
TJA0-J ($\mathbf{M}$)	-6.00	1.00	-5.40	0.80	-3.00	0.40	1.60	0.20	5.40	0.80
TJA0-J (μ)	-5.99	0.99	-5.39	0.79	-2.90	0.30	1.59	0.19	5.41	0.81
TJA1-B ($\mathbf{M}$)	-5.80	0.80	-5.20	0.60	-2.60	0.00	1.80	0.40	5.60	1.00
TJA1-B (μ)	-5.82	0.82	-5.20	0.60	-2.63	0.03	1.86	0.46	5.51	0.91
TJA1-D ($\mathbf{M}$)	-6.00	1.00	-5.40	0.80	-2.60	0.00	2.00	0.60	5.60	1.00
TJA1-D (μ)	-5.97	0.97	-5.41	0.81	-2.52	0.08	1.96	0.56	5.63	1.03
TJA1-F ($\mathbf{M}$)	-6.00	1.00	-5.40	0.80	-2.80	0.20	2.00	0.60	5.60	1.00
TJA1-F (μ)	-5.95	0.95	-5.41	0.81	-2.89	0.29	1.96	0.56	5.69	1.09
TJA1-H ($\mathbf{M}$)	-6.00	1.00	-5.40	0.80	-2.80	0.20	1.60	0.20	5.80	1.20
TJA1-H (μ)	-5.92	0.92	-5.38	0.78	-2.89	0.29	1.63	0.23	5.70	1.10
TJA1-J ($\mathbf{M}$)	-5.80	0.80	-5.20	0.60	-2.80	0.20	1.80	0.40	5.40	0.80
TJA1-J (μ)	-5.79	0.79	-5.24	0.64	-2.80	0.20	1.70	0.30	5.47	0.87

Table 4. Scenario B.2 - single insertions BCW-SQUARE $\alpha = 1, w = 25$

Scenario	A	err.	C	err.	E	err.	G	err.	I	err.
clean ($\mathbf{M}$)	-5.60	0.60	-5.00	0.40	-2.80	0.20	1.60	0.20	5.20	0.60
clean (μ)	-5.69	0.69	-5.01	0.41	-2.70	0.10	1.60	0.20	5.25	0.65
TJA0-B ($\mathbf{M}$)	-5.60	0.60	-4.80	0.20	-2.60	0.00	1.60	0.20	5.20	0.60
TJA0-B (μ)	-5.65	0.65	-4.84	0.24	-2.63	0.03	1.63	0.23	5.24	0.64
TJA0-D ($\mathbf{M}$)	-5.60	0.60	-5.00	0.40	-2.60	0.00	1.60	0.20	5.40	0.80
TJA0-D (μ)	-5.66	0.66	-5.03	0.43	-2.61	0.01	1.65	0.25	5.35	0.75
TJA0-F ($\mathbf{M}$)	-5.80	0.80	-5.20	0.60	-2.80	0.20	1.60	0.20	5.40	0.80
TJA0-F (μ)	-5.76	0.76	-5.13	0.53	-2.77	0.17	1.64	0.24	5.39	0.79
TJA0-H ($\mathbf{M}$)	-5.80	0.80	-5.00	0.40	-2.80	0.20	1.60	0.20	5.40	0.80
TJA0-H (μ)	-5.71	0.71	-5.04	0.44	-2.80	0.20	1.52	0.12	5.32	0.72
TJA0-J ($\mathbf{M}$)	-5.60	0.60	-5.00	0.40	-2.80	0.20	1.60	0.20	5.00	0.40
TJA0-J (μ)	-5.53	0.53	-4.99	0.39	-2.74	0.14	1.57	0.17	5.07	0.47
TJA1-B ($\mathbf{M}$)	-5.60	0.60	-4.80	0.20	-2.60	0.00	1.60	0.20	5.20	0.60
TJA1-B (μ)	-5.56	0.56	-4.88	0.28	-2.68	0.08	1.60	0.20	5.19	0.59
TJA1-D ($\mathbf{M}$)	-5.60	0.60	-5.00	0.40	-2.60	0.00	1.60	0.20	5.20	0.60
TJA1-D (μ)	-5.60	0.60	-4.95	0.35	-2.66	0.06	1.64	0.24	5.25	0.65
TJA1-F ($\mathbf{M}$)	-5.60	0.60	-5.00	0.40	-2.60	0.00	1.60	0.20	5.20	0.60
TJA1-F (μ)	-5.67	0.67	-5.01	0.41	-2.69	0.09	1.61	0.21	5.28	0.68
TJA1-H ($\mathbf{M}$)	-5.60	0.60	-5.00	0.40	-2.80	0.20	1.60	0.20	5.20	0.60
TJA1-H (μ)	-5.65	0.65	-4.99	0.39	-2.73	0.13	1.58	0.18	5.25	0.65
TJA1-J ($\mathbf{M}$)	-5.60	0.60	-5.00	0.40	-2.60	0.00	1.60	0.20	5.20	0.60
TJA1-J (μ)	-5.50	0.50	-4.93	0.33	-2.67	0.07	1.63	0.23	5.10	0.50

Table 5. Scenarios B.1 and B.3 temperature variations and multiple insertions FWD SQUARE $\alpha = 2, w = 200$ (8x)

Scenario	A	err.	C	err.	E	err.	G	err.	I	err.
cln. ntw. ($\mathbf{M}$)	-4.60	0.40	-4.20	0.40	-2.20	0.40	1.50	0.10	4.50	0.10
cln. ntw. (μ)	-4.56	0.44	-4.19	0.41	-2.15	0.45	1.46	0.06	4.42	0.18
cln. ntw. 0 $^\circ$C ($\mathbf{M}$)	-4.50	0.50	-4.20	0.40	-2.20	0.40	1.50	0.10	4.50	0.10
cln. ntw. 0 $^\circ$C (μ)	-4.50	0.50	-4.17	0.43	-2.19	0.41	1.49	0.09	4.48	0.12
cln. ntw. 50 $^\circ$C ($\mathbf{M}$)	-4.60	0.40	-4.20	0.40	-2.20	0.40	1.50	0.10	4.40	0.20
cln. ntw. 50 $^\circ$C (μ)	-4.54	0.46	-4.22	0.38	-2.21	0.39	1.46	0.06	4.21	0.39
cln. ntw. 60 $^\circ$C ($\mathbf{M}$)	-4.50	0.50	-4.20	0.40	-2.20	0.40	1.50	0.10	4.40	0.20
cln. ntw. 60 $^\circ$C (μ)	-4.45	0.55	-4.21	0.39	-2.21	0.39	1.46	0.06	4.24	0.36
2 adv. B, J ($\mathbf{M}$)	-5.00	0.00	-4.50	0.10	-2.20	0.40	1.40	0.00	4.50	0.10
2 adv. B, J (μ)	-5.05	0.05	-4.52	0.08	-2.25	0.35	1.43	0.03	4.54	0.06
3 adv. D,F,H ($\mathbf{M}$)	-5.60	0.60	-5.30	0.70	-2.70	0.10	1.90	0.50	5.50	0.90
3 adv. D,F,H (μ)	-5.57	0.57	-5.33	0.73	-2.70	0.10	1.86	0.46	5.44	0.84

Table 6. Scenario B.1 and B.3 temperature variations and multiple insertions BCW SQUARE $\alpha = 0.25, w = 25$

Scenario	A	err.	C	err.	E	err.	G	err.	I	err.
cln. ntw. ($\mathbf{M}$)	-4.80	0.20	-4.00	0.60	-2.40	0.20	1.60	0.20	4.00	0.60
cln. ntw. (μ)	-4.50	0.50	-4.23	0.37	-2.27	0.33	1.41	0.01	4.36	0.24
cln. ntw. 0 $^\circ$C ($\mathbf{M}$)	-4.80	0.20	-4.00	0.60	-2.40	0.20	1.60	0.20	4.00	0.60
cln. ntw. 0 $^\circ$C (μ)	-4.51	0.49	-4.11	0.49	-2.19	0.41	1.41	0.01	4.33	0.27
cln. ntw. 50 $^\circ$C ($\mathbf{M}$)	-4.80	0.20	-4.00	0.60	-2.40	0.20	1.60	0.20	4.00	0.60
cln. ntw. 50 $^\circ$C (μ)	-4.48	0.52	-4.13	0.47	-2.17	0.43	1.43	0.03	4.39	0.21
cln. ntw. 60 $^\circ$C ($\mathbf{M}$)	-4.80	0.20	-4.00	0.60	-2.40	0.20	1.60	0.20	4.80	0.20
cln. ntw. 60 $^\circ$C (μ)	-4.53	0.47	-4.18	0.42	-2.22	0.38	1.45	0.05	4.40	0.20
2 adv. B, J ($\mathbf{M}$)	-4.80	0.20	-4.80	0.20	-2.40	0.20	1.60	0.20	4.00	0.60
2 adv. B, J (μ)	-4.88	0.12	-4.31	0.29	-2.24	0.36	1.32	0.08	4.33	0.27
3 adv. D,F,H ($\mathbf{M}$)	-4.80	0.20	-4.80	0.20	-2.40	0.20	1.60	0.20	4.80	0.20
3 adv. D,F,H (μ)	-5.05	0.05	-4.79	0.19	-2.44	0.16	1.51	0.11	4.84	0.24