PanoptiCANs - Adversary-resilient Architectures for Controller Area Networks

Bogdan Groza[1], Lucian Popa[1], Tudor Andreica[1], Pal-Stefan Murvay[1],
Asaf Shabtai[2], and Yuval Elovici[2]

[1] Politehnica University Timisoara, Faculty of Automatics and Computers
{bogdan.groza, lucian.popa, tudor.andreica,
pal-stefan.murvay}@aut.upt.ro
[2] Ben-Gurion University of the Negev, Israel
{shabtaia, elovici}@bgu.ac.il

Abstract. Inspired by Jeremy Bentham's *panopticon*, i.e., an institutional building design in which a single security guard is able to monitor all detainees while they are unable to tell if they are being watched, we design the PanoptiCANs — a series of adversary-resilient CAN bus architectures. While DoS attacks are impossible to prevent on a regular bus topology, the PanoptiCANs are able to actively respond to them, as well as to generic attacks, by air gapping the network. The proposed modifications allow a bus guardian to monitor and isolate intruders on the bus while all traffic is redirected so that legitimate nodes carry on their tasks without significant disturbances. A decentralized version delegates these abilities to regular nodes, reducing costs and wire lengths, while also being able to localize and isolate the intruders much faster. We prove the effectiveness of the proposed topologies on an experimental setup with automotive grade controllers and collected in-vehicle traffic data. With the most effective architecture, intruders are isolated in a few milliseconds following single frame injections.

1 Introduction and motivation

Starting with the security incidents reported almost a decade ago [1], [2], [3], cars and the Controller Area Network (CAN) bus, in particular, have become an engaging research subject for security professionals. The CAN bus is the most widely-used in-vehicle communication layer with a history that spans over more than three decades. BOSCH, the original designer of CAN, started to work on CAN-FD since 2011 [4]. This extension is now available and increases the bandwidth of CAN and the size of its frames. More recently, in 2018, the CAN in Automation (CiA) association of users and manufacturers started the specification for CAN XL [5], a layer which extends the bandwidth even further. So it is clear that the CAN bus is here to stay and will be present in cars and various industries for the decades that follow. The modifications proposed in this work are compatible with future extensions of CAN and may be adapted for different electrical specifications.

There are several attack entry points that have to be considered in modern cars: adversaries may remotely corrupt an existing in-vehicle unit, tap the bus at some location that is more accessible and, much more commonly, connect to an available interface

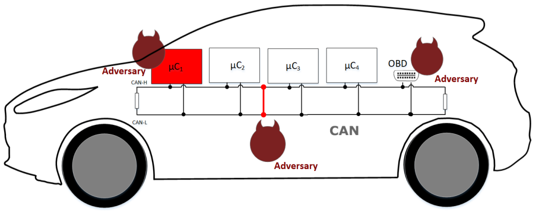


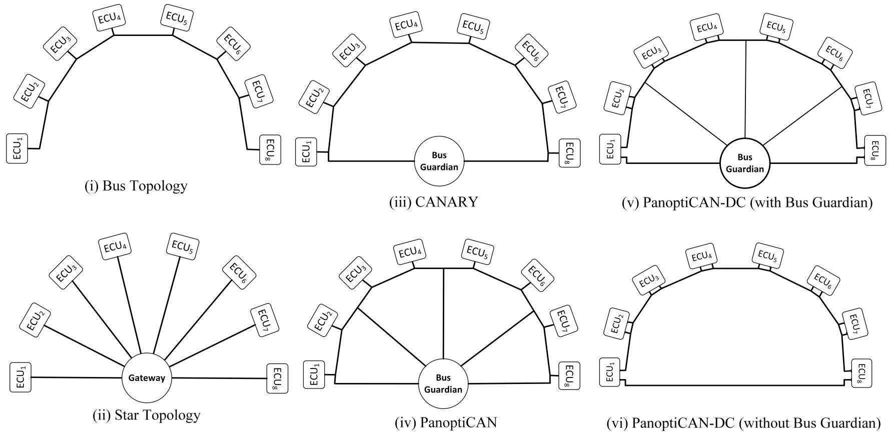
Fig. 1. Addressed setting: adversarial actions on CAN bus

such as the OBD port. Figure 1 suggests such scenarios by depicting a CAN bus inside a car and several adversaries. In response to this, there have been many efforts to secure the CAN bus, e.g., a brief summary can be found in [6]. The large majority of these works target either the introduction of some cryptographic payload in the frame or the development of intrusion detection systems that may separate between legitimate and adversarial traffic. We briefly enumerate some solutions in the related works section.

In-vehicle networks are heterogeneous, that is, both low and high-end controllers are present which makes it difficult to design solutions that can be ported on all devices that are plugged to the bus. In this context, intruder isolation, by air gaping the network with active relays and creating a physical separation between bus segments, is highly efficient in preventing attacks. Also, Denial-of-Service (DoS) cannot be stopped at all without such modifications as DoS prevention depends on the topology. With or without cryptography or intrusion detection systems in place, there is no way to prevent DoS attacks on CAN buses as long as a bus topology is employed since all nodes have unrestricted access to the communication medium. This not only allows a malicious node to send high-priority frames, but it also empowers malicious nodes to manipulate frames sent by other legitimate nodes with the goal of increasing their error counters and placing them into a bus-off state. Ultimately, a malicious node can keep the bus in a dominant state causing a complete blackout and no node will be able to send legitimate frames. The only way to prevent DoS attacks on the CAN bus is by architectural changes which have already been suggested in [7]. But such changes have to be done in a clever way so that the great advantages of a bus topology are not lost. Notably, the bus topology is cheap to implement and makes it easy to install nodes by simply plugging them to the wires. This is what made CAN the most desirable communication interface for in-vehicle networks and simply changing the bus topology to a star topology may not be so appealing (not to mention that it turns the central gateway into a single point of failure).

The PanoptiCAN: concept and design. Designed by Jeremy Bentham in the 18-th century [8], the *panopticon* (from Greek *panoptes*, i.e., all-seeing) is an institutional building that allows a single security guard to observe all detainees in the building while they are unable to tell whether they are being watched. Similarly, in a PanoptiCAN, the Bus Guardian (a trusted device attached to the bus) is able to monitor traffic and isolate each node by using active relays that change the topology of the network. We are not specifically interested that nodes remain unaware of being watched, what is important is that isolated nodes are still able to receive traffic from the rest of the network in order to keep the vehicle functional. In principle, a node should not be able to tell that it was cut-out from the bus by analyzing incoming traffic, neither should the rest of the nodes, since incoming traffic will perfectly mimic the full bus. This should be interpreted in a constructive sense, i.e., all CAN frames will arrive regardless of the intruder intervention. It is out of scope for this work if the adversary or legitimate nodes can decide whether such isolation took place based on physical characteristics, i.e., voltage levels [9], clock skews [10], or other fine-grained characteristics. Addressing this issue would lead to unnecessary complications. It does not seem to matter much if the adversary knows that it is isolated and the same holds for legitimate ECUs (Electronic Control Units) for which this is irrelevant as long as they receive the rest of the CAN packets and are able to deliver their own legitimate packets in time. Consequently, what maters is that the intrusion is observed, isolated, and all legitimate traffic remains largely unaltered, reaching its destination. The main advantage of our construction is that we can preserve all existing in-vehicle functionalities unaltered as all legitimate ECUs will have access to all in-vehicle traffic. In the decentralized version of the PanoptiCAN we renounce on the Bus Guardian in order to simplify wiring. In this case, the legitimate nodes are empowered to isolate the intruder and reconstruct traffic in other parts of the network. We keep the Bus Guardian optional in this version.

To put our contribution into context, in Figure 2 we provide a simplified view of some network configurations for CAN: the commonly employed bus topology (i), a star topology (ii), the recently proposed DoS-resilient topology [7] of CANARY (iii) and the PanoptiCAN (iv) along with its distributed version with (v) and without a Bus Guardian (vi) which are the contributions of this work. To clarify the context, we now briefly discuss the advantages and disadvantages of these topologies. Controller Area Networks usually follow a bus topology. Star topologies have been commonly suggested as an alternative to increase the resilience of CAN [11], [12], [13], e.g., they do not allow a DoS to propagate over the bus, but they are more expensive, they introduce a single point of failure and they cannot retrofit existing vehicles. A newly suggested option, CANARY [7], allows dynamic topology changes by using active relays. In principle, CANARY is a mixed bus-ring topology where a Bus Guardian taps the two bus ends and bridges between the left and right sides of the network. Although from [7] CANARY may give the impression of a star topology, the lines running from the Bus Guardian to the relays are not CAN bus wires that carry data, but regular copper wires that carry a voltage signal that triggers the relays, and thus, the network topology is still a bus. A ring topology is formed by the Bus Guardian which links the left and right sides of the network. To these existing topologies, we add three more powerful topologies: the PanoptiCAN and its decentralized versions with or without the Bus Guardian. The PanoptiCAN is a mixed bus-star topology since the Bus Guardian taps the bus in several points that allow him to record/replay traffic. The decentralized version of the PanoptiCAN is a mixed bus-daisy-chain topology which does not require a Bus Guardian and greatly simplifies wiring, thus reducing costs, but also improves on intrusion localization speed, all these at the cost of an additional transceiver for each node which is inexpensive. In the light of the above, Table 1 provides a brief summary on the operation principles, advantages and disadvantages of the discussed architectures. Briefly, CANARY and the PanoptiCANs provide a switchable topology that is resilient to DoS and many other types of attacks. The decentralized version of the PanoptiCAN improves significantly in terms of localization speed and wiring requirements.



(i) Bus Topology

(iii) CANARY

(v) PanoptiCAN-DC (with Bus Guardian)

(ii) Star Topology

(iv) PanoptiCAN

(vi) PanoptiCAN-DC (without Bus Guardian)

Fig. 2. Simplified view of some existing/proposed topologies for CAN: (i) bus, (ii) star, (iii) CANARY, (iv) PanoptiCAN, (v) PanoptiCAN-DC with a bus guardian, and (vi) PanoptiCAN-DC without a bus guardian

Table 1. Advantages and disadvantages of existing topologies, CANARY and the PanoptiCANs

Topology	**Operating principle**	**Advantages**	**Disadvantages**
Regular Bus	nodes wired to the same line	cheap and easy to deploy	no intruder isolation, DoS vulnerable
Regular Star	nodes wired to a gateway	good node isolation (DoS resilient)	expensive gateways, require one channel for each node, more wires
CANARY	cut bus segments and/or load-balance the network in case of attacks	can isolate intruders, DoS resilient, can retrofit existing networks	more difficult/expensive wiring, requires bus guardian
PanoptiCAN	bus guardian isolates/monitors each node in case of attacks	can isolate intruders, DoS resilient, can retrofit existing networks	more difficult/expensive wiring, requires a bus guardian
PanoptiCAN-DC	nodes locally switch the bus to a daisy-chain topology	can isolate intruders, DoS resilient, simple wiring, distributed, bus guardian optional	requires one additional transceiver for each node

Summary of contributions. Briefly, our work contributes in five relevant directions with respect to existing works:

1. we propose several switchable architectures that are resilient to adversary attacks and which are more effective than previous approaches, i.e., the PanoptiCAN and its decentralized version PanoptiCAN-DC,
2. we improve on the wirings which are both difficult to manage inside cars and nonetheless expensive, this improvement is both in the way we wire the *bus canaries*, i.e., the double relay-resistor structure initially proposed by [7], but also in the topology of the decentralized PanoptiCAN-DC which requires far simpler wirings compared to both CANARY and PanoptiCAN,

3. we improve on the localization speed significantly with the decentralized PanoptiCAN-DC which is capable to localize the adversary almost instantaneously following a single frame injection without needing the Bus Guardian intervention to probe the network and locate the intrusion,
4. we specifically focus on preventing the more insidious DoS attack caused by *error inflicting adversaries* that modify legitimate frames to lead sender nodes into Bus-off and also account for the possibility of *multiple adversaries* on the bus,
5. last but not least, by this work we also push more in the direction of *adversary resilient topologies* that *react* on adversarial actions, opening road for protecting vehicles against intrusions by actively air gapping the networks.

The rest of the paper is organized as follows. In Section 2 we present some basics on CAN buses and discuss related works. Section 3 introduces the design that we propose for the PanoptiCAN and one immediate simplification which greatly reduces wiring costs. Then in Section 4 we present the evaluation scenarios we consider and Section 5 contains the experimental results. Finally, Section 6 holds the conclusion of our work.

2 Background and related works

In this section we introduce some basic background on CAN buses and discuss related works on attack and countermeasures for CAN.

2.1 CAN basics

The physical layer of the CAN bus consists in two differential lines linked at the ends with 120Ω termination resistors as illustrated in the left side of Figure 3. The CAN protocol supports bit rates of up to 1Mbps using frames with a specific layout. The frame starts with an arbitration field of 11 bits (or 29 bits in extended frames), contains a payload which is up to 64 bits (or up to 512 bits in CAN-FD) and a 15-bit CRC (or up to 21 bits in CAN-FD). All nodes communicating on the bus must comply with the error management mechanism required by the standard to assure undisturbed communication in the presence of faulty transmitters or receivers. For this, there are two counters, TEC (transmit error counter) and REC (receive error counter), incremented each time a CAN error is observed or decremented after a frame is successfully received. The detection of a frame transmission error is signaled by the transmitter or receivers using error frames which consist in 6 consecutive dominant bits for active error flags or 6 consecutive recessive bits for passive error flags. These flags violate the specified stuffing rule, notifying in this way all nodes of the error. Nodes can be in one of the three defined error states: error active, error passive and bus-off. If the TEC and REC counters are both lower than 128, the node is in the error active state and can transmit active error flags. If at least one of the counters is greater than 127 the node is in the error passive state and can transmit only passive error flags. If the TEC counter is greater than 255, the node will disconnect from the bus, entering in a Bus-off state. The right side of Figure 3 shows the CAN error states and transition conditions. It is notable to mention that the error management mechanism was exploited to force legitimate ECUs in Bus-off [14] or as a defense measure against adversarial ECUs [15].

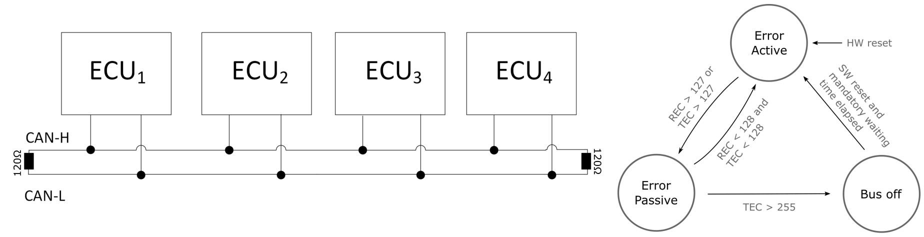


Fig. 3. Basic depiction of a CAN bus (left) and the CAN error state machine (right)

2.2 Related works

Since the CAN bus does not include sender authentication or other security mechanisms by default, nodes which are communicating on CAN are vulnerable to several types of attacks. Vulnerabilities exposed by [2], [3] or more recently [16], [17] and many others showed that messages from genuine ECUs can be easily spoofed or that adversarial frames can be injected from remote in order to take control of critical vehicle functionalities. Authenticating incoming data has been proposed by numerous works and recently by industry standards, e.g., AUTOSAR [18], [19], etc.

However, even if security is in place and intrusions are detected, there is still room for DoS attacks due to the wired-AND behaviour of the CAN bus. The simplest form of DoS attack, mentioned as early as the work in [20], exploits the CAN arbitration mechanism by continuously sending frames with high priority which hinders legitimate transmissions. A partial solution against DoS attacks is the use of ID-hopping techniques which modify frame identifiers/priorities through a secure procedure. This was first proposed by Humayed and Luo [21] which used a software-based implementation while a dedicated CAN controller which provides increased ID entropy was proposed in [22]. The use of ordered encryption for the same purpose was recently suggested in [23]. This type of solutions will not work against an adversary that disrupts legitimate frames or which writes the highest priority ID, i.e., 0x00, on the bus. Such specialized attacks were more recently analyzed in [24] and [25] where CAN frames were manipulated to prevent correct interpretation of CAN symbols. This type of attack can target specific messages or nodes on the bus [25] sending them into Bus-off. This vulnerability was previously demonstrated in [14]. Resetting the ECU error counters was suggested as a countermeasure but this will nullify the error confinement capabilities of CAN.

Our work can be also linked to related works that address the reliability of CAN buses. Bus Guardians were used in [26], [27] to increase the reliability of CAN by monitoring the electrical signal on the bus. In this context, the Bus Guardian is not responsible with intrusion detection or triggering relays to disconnect parts of the bus. The idea of using relays to disconnect sections of the CAN bus was employed in the context of fault detection [28], [29] and [30] where relays were used to simulate broken wires. These works do not use *bus canaries* that maintain connectivity on the bus and do not address security countermeasures by the use of relays. So far, CANARY [7] is the only proposal that addresses DoS attacks by disconnecting bus segments with the use of a Bus Guardian which monitors the network and triggers the *bus canaries*. We have already argued in the introduction how our work improves on this and more details will follow in the next sections.

3 Design details

This section presents the design of the PanoptiCAN and of its decentralized version which improves the intruder localization time and simplifies the wirings.

3.1 Engineering goals

First, we underline our engineering goals. With the designs proposed in this work we mainly try to improve in two directions: reducing the relay triggering rate and reducing the wiring complexity. These are vital for practical adoption of the proposed technology.

Relay triggering will induce errors on legitimate nodes due to electrical disturbances on the bus. It has been shown in [7] that the error counters remain well below the Error Passive threshold and will never reach the Bus Off state. While this means that the solution in [7] is safe to use, it still seems preferable to keep the relay triggering rate as low as possible especially since some relays will include mechanical parts that may be damaged after repeated use. Due to the more efficient placement of relays and bus taps, the PanoptiCANs are able to isolate nodes efficiently without the need to load-balance the network as in the case of CANARY [7] where the relays need to be triggered at fast rates when performing the load-balancing defense. The PanoptiCANs do not require load balancing as the adversary can be immediately isolated, and more, PanoptiCAN-DC can isolate the adversary even faster than the regular PanoptiCAN.

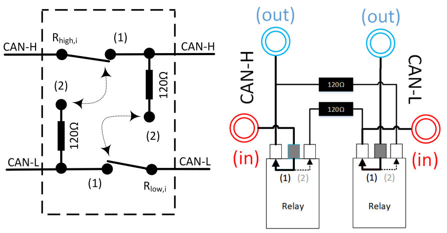


Fig. 4. The original relay schematic from CANARY [7] (left) and the actual wiring for PanoptiCAN relays (right)

Wiring is another issue. Current cars may have around 2.2 km of wires that connect hundreds of sensors and control units, according to recent estimates from the industry [31]. CANARY [7] may call only for a small fraction compared to this. But still, each extra wire induces cost and additional difficulties in mounting it inside the car. The setup used in [7] is also somewhat simpler having only 5 ECUs guarded by 8 relays, but for the PanoptiCAN, in this work, we develop a setup that is almost twice as large by using 8 ECUs and 24 relays. This could double the wiring demand, but we improve both by using a more efficient scheme for wiring the relays (discussed in the next section) as well as by a simpler design. The PanoptiCAN-DC makes the Bus Guardian optional and its wiring is much more simpler than all previous approaches. The relay structures used in CANARY [7], which we will call *bus canaries* or simply *canaries* in what follows, are a double relay-resistor pair which are capable of actively cutting adversaries from the bus, i.e., simply by splitting the bus into two or more sub-buses that

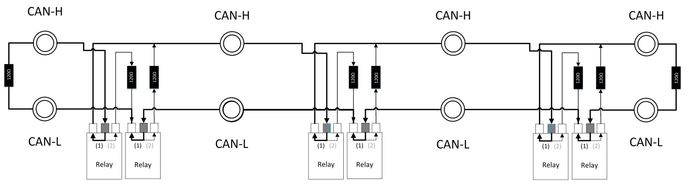


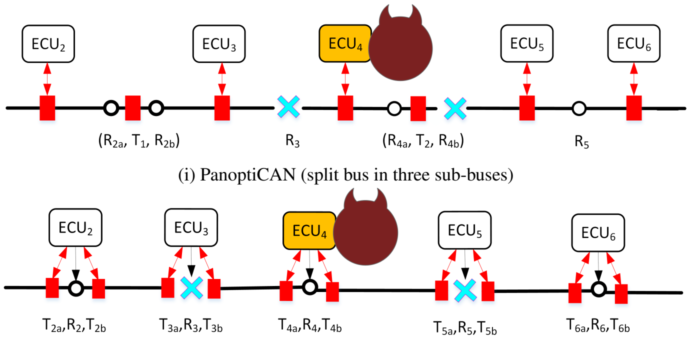
Fig. 5. Bus wiring for PanoptiCANs (4 connection points for ECUs)

are still compliant to the CAN standard which requires a 120Ω termination at the end of the lines. We use similar *bus canaries* in our work but with some wiring simplification that is more suitable for the relays that we use in our setup. During the implementation, we noticed that the resistors can be directly linked to the pins of the relays which results in a more compact *bus canary* with less wirings and a much more intuitive connection to the bus. Figure 4 contrasts between the original schematic from [7] and the wiring of the *bus canaries* from the current work. The two components are essentially identical but the new wiring from our work is much simpler and more suitable for the off-the shelf relays.

This wiring of the *bus canary* makes it much easier to generalize the schematic for a bus topology as depicted in Figure 5. This bus-like depiction is more suitable for implementation purposes and it also shows that bus modifications are not very complicated for practice, i.e., there are two relay-resistor pairs in each location where the bus needs to be split. Having clarified the exact wiring scheme, in the exposition that follows we will switch to a simplified view of the wiring which is more intuitive. To get a more concrete image on how intruder isolation can be performed in the PanoptiCAN and PanoptiCAN-DC, Figure 6 gives a brief overview on intruder isolation for a bus segment in case when ECU_4 becomes adversarial. The red-filled rectangles denote transceivers on the bus, while the black circles are inactive *canaries* and a cross denotes a triggered *canary*. The PanoptiCAN will use canaries R_3 and $R_{4,b}$ triggered by the Bus Guardian while the PanoptiCAN-DC will use canaries R_3 and R_5 triggered by legitimate nodes ECU_3 and ECU_5. Consequently, the PanoptiCAN will split the bus into 3 sub-buses while the PanoptiCAN-DC will switch the bus nearby ECU_3 and ECU_5 into a daisy-chain topology.

3.2 PanoptiCAN: topology and procedures

Having *canaries* as a starting point, the design of the PanoptiCAN is straight-forward: each ECU is placed between two canaries and a bus tap that is linked or multiplexed to the Bus Guardian. Additionally, two transceivers are placed at the two bus ends. The left side of Figure 7 contains a graphical depiction of an 8 ECU PanoptiCAN and can be easily extended to any number of nodes. In a network of n ECUs, i.e., $\mathsf{ECU}_i, i = 1..n$ one *canary*, i.e., R_i, is placed after each odd numbered ECU and two *canaries* with a *tap* in the middle are placed after each even numbered ECU, i.e., $(\mathsf{R}_{i,a}, T_{i/2}, \mathsf{R}_{i,b})$. To isolate an odd numbered ECU, e.g., $\mathsf{ECU}_i, i = 2k+1$, the canaries $\mathsf{R}_{i-1,a}$ and R_i will be triggered. The only exception is ECU_1 for which only R_1 has to be triggered since it is



(i) PanoptiCAN (split bus in three sub-buses)

(ii) PanoptiCAN-DC (switch bus to a daisy-chain topology)

Fig. 6. Intruder isolation in the PanoptiCAN (i) and PanoptiCAN-DC (ii)

at the beginning of the bus. Incoming traffic from odd-numbered ECUs can be recorded from bus tap $T_{(i-1)/2+1}, i = 2k+1$. To isolate an even-numbered ECU, canaries R_{i-1} and $\mathsf{R}_{i,b}$ will be triggered. Again, the exception is the ECU at the end of the bus, even or odd, a case in which only the relay which precedes it will be triggered, i.e., $\mathsf{R}_{i-1,*}$ (here $*$ is a placeholder which is void for even numbered ECUs and a for an odd number ECU). Traffic from even numbered ECUs will be recorded at tap $T_{i/2}$. In this topology, the Bus Guardian can efficiently determine the location of the adversary and isolate it by using a divide and conquer strategy, i.e., split the network in two by triggering the relay in the middle and see from which side the intrusion packets originate, etc. A simpler option is by isolating each ECU one at a time and see if the corrupted traffic originates from the corresponding ECU.

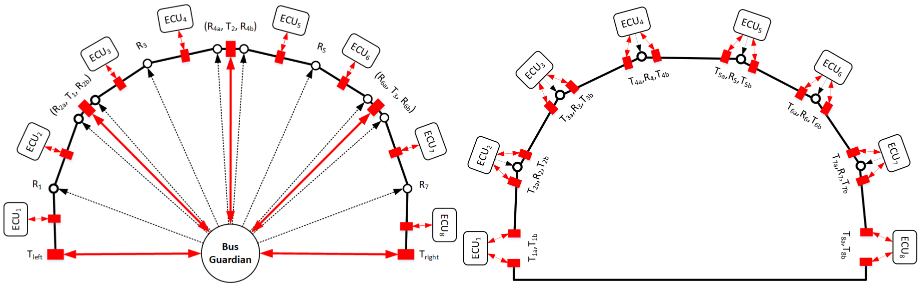


Fig. 7. A PanoptiCAN (left) and a decentralized PanoptiCAN-DC without a Bus Guardian (right) with 8 ECUs

3.3 PanoptiCAN-DC: a more efficient, decentralized design

The design of CANARY and PanoptiCAN share a common difficulty in wiring the Bus Guardian to each of the *canaries*. Each *canary* requires two wires to be controlled and given the placement of the *canaries* along the bus, this results in a wiring harness that is in principle equivalent to a star topology although these are not CAN wires and the number of transceivers is reduced compared to a star topology.

To further improve our concept, we introduced a decentralized version of the PanoptiCAN, which we call PanoptiCAN-DC, in which we greatly reduce the wirings by letting each node be in control of its own *canary*. This sets room for using much shorter wires between each node and the *canary* nearby. However, in this case we cannot let the nodes to simply cut the bus in their vicinity since we need traffic to further propagate between the resulting sub-buses, we need a much more clever solution for this. For this purpose we use a daisy-chain topology that will still allow each node to communicate while a DoS is no longer feasible as long as nodes will filter incoming traffic and will not propagate intruder frames further into the network. The daisy-chain topology will require an additional transceiver on each of the ECUs.

The right side of Figure 7 shows the topology of the PanoptiCAN-DC, the decentralized version of the PanoptiCAN. The design is symmetric, each ECU has two bus taps and a *canary* in the middle, the two ECUs at left and right ends of the bus communicate on a private CAN and do not have a *canary*. For a network of n ECUs, each $\mathsf{ECU}_i, i = 1..n$ has two transceivers, i.e., $T_{i,a}, T_{i,b}$ and one *canary*, i.e., R_i, except for ECU_1 and ECU_n. In case an intrusion occurs, each $\mathsf{ECU}_i, i = 2..n-1$ will trigger its *canary* R_i and cut the bus at his location. Then it will filter and redirect traffic from one side to another. In this way, a DoS attack no longer propagates into the network. The Bus Guardian is optional in this design and required only to retransmit traffic from one part to the other of the network when more than one adversary is present.

4 Adversary model and evaluation scenarios

In this section we discuss the adversary model and the scenarios for which we further evaluate the performance of the proposed solution.

4.1 Adversary model

We assume the existence of an adversary that has full control over the communication channel, but we do refine this model for the specific needs of our setup. If one node becomes adversarial and all the traffic that it sends is bogus then the node will be localized and disconnected from the network. If the intruder plugs into the network in the vicinity of a legitimate node and isolating the intruder is not possible, then the best that we can do is to isolate the intruder on the segment with the legitimate node.

Two adversarial actions that were commonly considered by the literature are fuzzing the bus in which the adversary injects random CAN frames that have random IDs or data fields and replay attacks in which the adversary injects existing IDs with identical or randomized data-fields. While each node may run its own IDS and ignore attack packets, Denial of Service (DoS) attacks are much more complicated to address. Also, cryptography provides a good solution in response to first two types of attacks but it is fully ineffective against DoS attacks. As already mentioned in the introduction, a DoS can be caused either by *flooding* the bus with high priority identifiers as well as by *distorting* legitimate frames which will increase the error counters of legitimate nodes. Since these attacks are more dangerous, we focus on them in what follows.

4.2 Attack response capabilities

Both the PanoptiCAN and its distributed version can isolate any single ECU if it becomes corrupted. Both schemes can respond even to insidious attacks such as frame distortions (that can place legitimate ECUs into Bus-off) and check whether the attack originates on the specific ECU or has been forged from another bus segment. However, it will not be possible to separate between the legitimate ECU and the adversary as long as the adversary taps the bus on the same segment as the legitimate ECU. Such situations should be rare as physical access to vehicle wires is not so immediate (most of the attacks reported so far come from open connections such as the OBD port or from corrupted units such as vehicle telematics). Finally, if the adversary can tap the bus at any point inside the vehicle, then he may use the same connection point as the legitimate ECU making the separation impossible anyway.

Both the PanoptiCAN and its distributed version can address the case of multiple adversaries. For the PanoptiCAN, isolating multiple adversaries depends on the number of bus taps. To isolate all segments of the bus, n bus taps would be needed, which will make the PanoptiCAN capable to switch from a bus to a star topology. In our design however, we considered only $n/2$ bus taps which makes it possible to isolate at most $n/4 + 1$ adversaries. Figure 8 (i) clarifies why this is the case. If $\mathbf{Adv}_1$ is the corrupted ECU_1 then it will no longer be possible to separately isolate ECU_2 from ECU_3 since there is only one tap left, i.e., T_1, that connects to their segment. So the next adversary that can be isolated is $\mathbf{Adv}_2$ on ECU_4. The same reasoning goes further and the next adversary that can be isolated is $\mathbf{Adv}_3$ on ECU_8, i.e., $8/4 + 1 = 3$ adversaries isolated in zones Z_1, Z_2 and Z_3 as depicted in the figure. Worst case however, if we isolate adversaries at ECU_2 and ECU_7 then it is no longer possible to isolate any of the ECUs 3, 4, 5 or 6, since there is a single tap, i.e., $T2$, available. So in the worst case, $n/4$ adversaries can be isolated. Since in-vehicle networks are controlled environments and only a small number of corrupted units is expected, $n/4$ seems a good reference point. To generalize on this, Figure 8 (ii) and (iii) explores the theoretical possibilities for the adversary locations and the amount of these which can be successfully isolated. Note that while modern vehicles may have more than 100 ECUs, these are never connected to the same bus, they are always organized in sub-networks of usually less than a dozen ECUs. We considered a network of 16-24 ECUs which is very large, usually there are less than a dozen nodes on the same bus. In this network we add 1-6 adversaries and this results in an exponential increase for the possible placements of the adversaries, i.e., up to about 5×10^7 possible locations. In theory, k adversaries may cover $\binom{n}{n-k}$ bus segments (assuming that the order of the adversaries does not matter) and $n/4$ adversaries can be configured in $\binom{n}{n/4}$ locations. But PanoptiCAN can isolate about 25% of its nodes, so up to 6 nodes can be isolated in the 24-node PanoptiCAN while the smaller 16-node PanoptiCAN can isolate up to 4 nodes.

For the distributed version, PanoptiCAN-DC, the situation is further improved. Due to the autonomous action of each node, any number of adversaries can be isolated. However, if there is no bus guardian to redirect traffic from one segment to another, two

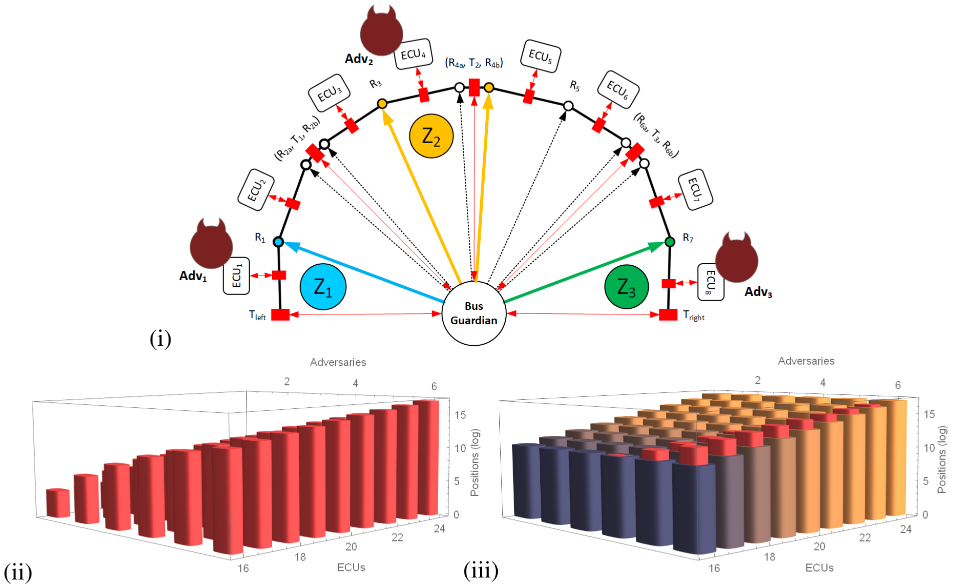


Fig. 8. Example of adversary placement and corresponding isolated zones (i), possible adversary locations (ii) and overlay with locations that can be isolated (iii) in a network with 16-24 nodes and 2-6 adversaries (the z-axis of the plot is base 2 logarithmic)

adversaries may cause a DoS that will completely cut all the bus segments in between. For example, by using Figure 8 (i) as a reference, if ECU_2 and ECU_7 are corrupted, then they will be immediately isolated by their neighboring ECUs, but if they cause a DoS, then no traffic can be recovered from any of the ECUs 3, 4, 5 and 6. For this, a Bus Guardian may be added in the distributed version to redirect traffic. We do believe however that multiple adversaries will be rare on in-vehicle networks and the simplicity of the PanoptiCAN-DC is a much greater advantage.

4.3 Expected response to DoS attacks

We now set a brief theoretical framework for understanding channel behavior in case of a DoS caused by a flooding attack on the network. We are interested in determining the localization time and the delays induced on legitimate packets before and after the adversary is isolated.

Let λ_{adv} be the arrival rate for adversarial frames on a bus which can accommodate up to λ_{bus} frames and let λ_{leg} be the rate of legitimate frames on the bus. Obviously, $\lambda_{leg} < \lambda_{bus}$ and in most real-world applications the frame rate of the bus is half of the maximum bus rate [32]. In practice, CAN buses may have a load of around several thousands frames per second. Since most practical in-vehicle deployments use a 500kbps bandwidth and are kept below a 50% busload, a rate of around 2000 frames per second can be expected for legitimate traffic. Assuming that the adversary floods the bus with packets with higher priority, the maximum arrival rate for legitimate frames during a Dos attack is: $\lambda_{leg}^{\max} = \min(\lambda_{leg}, \lambda_{bus} - \lambda_{adv})$.

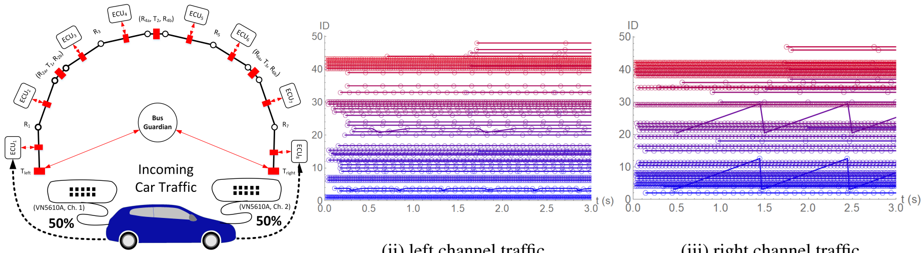
Clearly, by flooding the bus at a maximum rate, the adversary can make the maximum arrival rate for legitimate frames drop to 0. Fortunately, this happens only as long as the adversary is not yet isolated. To isolate the intruder, assuming on-event based localization, the PanoptiCAN will need $\log_2 n$ frames, since the fastest way to isolate an intruder in an n node network is by performing a binary search, while the PanoptiCAN-DC can perform the isolation following a single intruder frame, since the neighboring nodes will immediately trigger their relays when an intrusion is detected. This leads to the following localization time for the two schemes: $\varTheta = \log_2 n \times \lambda_{adv}^{-1}$, $\varTheta^{dc} = t_{frame}$. The isolation is thus much faster with the distributed version. The delays encountered for the two schemes during the isolation process are also distinct. For the PanoptiCAN, the intruder will share the same bus with some of the legitimate nodes until the isolation is completed, thus, some legitimate frames may not be received until this happens. The PanoptiCAN-DC has to send all frames over multiple hops of the daisy-chain, but all the legitimate frames will arrive on the bus. These are expressed in the following relations: $\varDelta_{\neg isol} = \frac{t_{frame}}{h(\lambda_{bus} - \lambda_{leg} - \lambda_{adv})}$, which accounts for the remaining bandwidth following existing legitimate traffic and the traffic caused by the adversary, and $\varDelta_{\neg isol}^{dc} = \frac{n}{2} t_{frame}$, which accounts for the worst case in which a frame has to be retransmitted over $n/2$ nodes in case of the PanoptiCAN-DC. We use $\neg isol$ as a placeholder to denote that the isolation process started but the intruder is not yet isolated. Here h denotes the Heaviside step function, i.e., a zero for negative arguments or a one for positive arguments. Thus, as long as the rate of legitimate frames plus the adversary traffic exceeds the bus rate, $h(\lambda_{bus} - \lambda_{leg} - \lambda_{adv})$ will return a 0 leading to a maximum delay $\varDelta_{\neg isol} = \infty$. Once the intruder is isolated, there will be at most one hop for the PanoptiCAN as well as for the PanoptiCAN-DC which leads for both schemes to a low transmission delay of twice the time of the frame, i.e., $\varDelta_{isol} = 2t_{frame}$, since, in the worst case, the frame has to be retransmitted by the Bus Guardian or by the node near the intruder which is in charge of the isolation.

5 Experiments and results

In this section we discuss experiments with the proposed defense mechanism. Due to space constraints, the full description of the experimental model that we developed is deferred to Appendix A and more experiments can be found in Appendix B.

5.1 Recorded in-vehicle traffic

In our experiments, we use real world in-vehicle traffic that was collected by us from a high-end vehicle. The CAN bus was set at 500kbps, the usual bandwidth inside cars. The busload generally stayed between 30-50% which is usual inside vehicles. Given that an adversary located at the bus ends would be trivial to isolate, we choose to split the legitimate traffic into two traces that are sent to the left and right sides of the bus. Thus, the 90 identifiers where split in half and allocated to the left and right side of the



(i) test setup (ii) left channel traffic (iii) right channel traffic

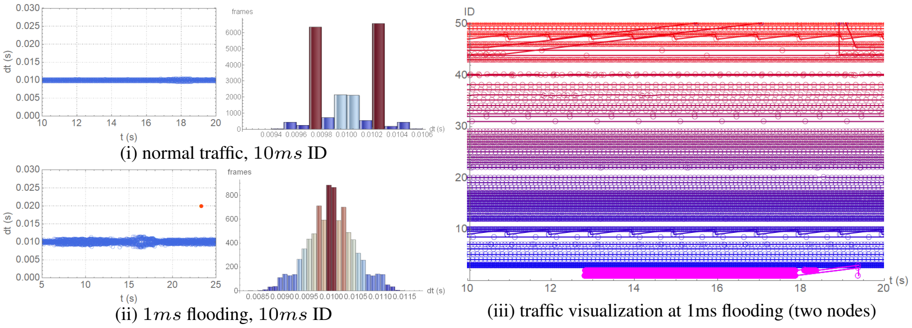
Fig. 9. IDs from the collected in-vehicle trace arriving on the left and right channels

experimental model. The in-vehicle traffic is reproduced inside the network with Vector's VN5610A which is an industry standard tool that allows real-time retransmission of in-vehicle traffic at micro-second accuracy. Figure 9 (i) shows a brief schematic of our test setup and the in-vehicle traffic arriving on the left (ii) and right channel (iii). There are less than 50 IDs on each channel. The cycles are well preserved with a small exception of an ID on the right channel which exhibits some cycle variations. This is not unexpected, while most in-vehicle traffic is cyclic in nature, on-event frames may also occur. In the plots from Figure 9 (ii) and (iii), the number of the ID represents its rank (the order of the ID based on its priority) by which it is placed on the ordinate (y-axis) to which we add the deviation of the current timestamp from the previous. Adding this deviation makes it much easier to spot the occurrence of a DoS as the delayed ID appear higher on the plot. For each ID, the abscissa (x-axis) is the timestamp at which it occurs (a small circle is used as a marker on the plot).

5.2 Response to DoS attacks in the experiments

In our testbed we broadcast legitimate in-vehicle traffic on the left and right sides of the bus as suggested in Figure 9.

PanoptiCAN response to DoS by flooding. We first test the response of the system in case of a flooding caused by two nodes, i.e., a DDoS attack (we use two nodes since the related solution in [7] cannot respond to multiple adversaries). We set ECU_2 and ECU_4 to flood the bus with one high priority message sent at $1ms$. This flooding attack will cause visible delays on the rest of the frames but the bus is still around 50% free so all regular traffic is still there. Figure 10 (i) and (ii) show the comparative effects of a flooding attack on an ID with a cycle time of $10ms$ and normal traffic (orange dots denote delayed frames). The CAN bus shows very good resilience, the delays induced by this flooding are very small making the arrival time of this high priority ID to deviate around 1ms, i.e., less than 10%. Figure 10 (iii) provides the traffic visualization for 50 IDs in case of the attack, the adversary high priority IDs can be seen in magenta at the bottom of the figure. Indeed there are very little disturbances during the attack for all the IDs. As an example, the flooding attack was programmed to last for 5 seconds, i.e., between the 13th and the 18th second as it can be seen in Figure 10. The PanoptiCAN can isolate the adversary in a few milliseconds as we discuss in Appendix B (the time to trigger the relays is $5ms$ according to the datasheet).

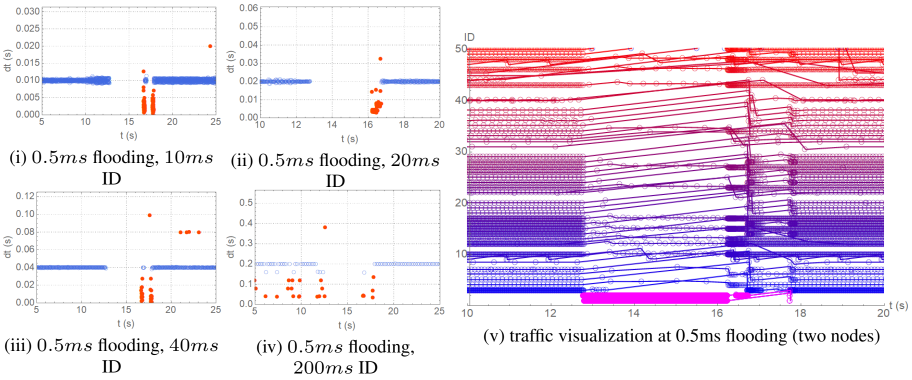


(i) normal traffic, $10ms$ ID

(ii) $1ms$ flooding, $10ms$ ID

(iii) traffic visualization at 1ms flooding (two nodes)

Fig. 10. Flooding effects on IDs with a cycle of $10ms$ (i), (ii) and visualization for 50 IDs (iii)



(i) $0.5ms$ flooding, $10ms$ ID

(ii) $0.5ms$ flooding, $20ms$ ID

(iii) $0.5ms$ flooding, $40ms$ ID

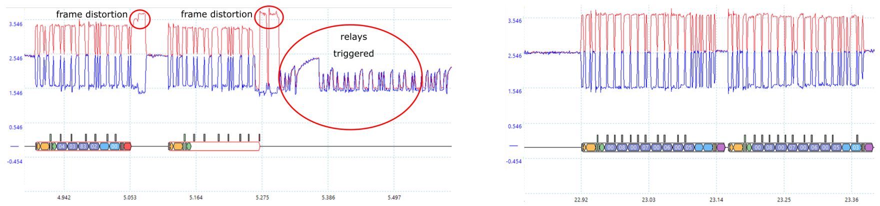
(iv) $0.5ms$ flooding, $200ms$ ID

(v) traffic visualization at 0.5ms flooding (two nodes)

Fig. 11. DDoS effects on IDs with a cycle time of (i) 10ms, (ii) 20ms, (iii) 40ms, (iv) 200ms and visualization for 50 IDs (v)

When we double the number of messages sent by ECU_2 and ECU_4, i.e., 0.5ms flooding instead of a 1ms flooding, the effects are far more dramatic. Doubling the adversarial messages leads to one message being sent each $250\mu s$ (this is roughly the duration of a CAN frame when the bus is set at 500kbps). There will be little or no space at all for legitimate frames which leads to a full DoS. Figure 11 (i), (ii), (iii) and (iv) show the effects of the DDoS attack on four IDs with a cycle time of $10ms$, $20ms$, $40ms$ and $200ms$. For all of them, as long as the adversary is not isolated there will be no frame that reaches the bus. Figure 11 (v) provides traffic visualization for 50 IDs in case of the 0.5ms flooding which leads to a DoS. The effects are very similar on all IDs, only in rare situations some of them manage to enter the bus. Again however, the PanoptiCAN will easily locate the two intruders and isolate them, restoring all traffic back to normal from the 18th second onward as can be seen in Figure 11 (v).

PanoptiCAN-DC response to attacks. The distributed version of the PanoptiCAN offers a much faster response to attacks since all relays will be triggered simultaneously, once an attack frame is detected, and the legitimate nodes will switch the bus to a daisy-chain topology which will no longer allow intruder frames to propagate. The only shortcoming is that multiple adversaries can isolate the bus segments between them by performing a DDoS attack, e.g., by flooding. The same limitation occurs with CANARY [7] and it can only be solved by placing a Bus Guardian to redirect frames from various parts of the bus as in the regular Panopti-CAN. For the PanoptiCAN-DC the Bus Guardian will simply redirect frames without requiring it to trigger the relays. To get a more concrete image on frame distortion attacks, in Figure 12 (i) we depict frames disrupted by the adversary. By setting the last consecutive bits to more than 6 zeros, a stuff error occurs and all nodes respond with an error flag. In this way the legitimate node will be forced to enter the bus off state. However, the PantoptiCAN-DC can easily isolate the adversary and frames returning to normal, as can be seen in Figure 12 (ii), after about 17ms once the adversary is isolated (the time to trigger the relays is 5ms, but since multiple relays are triggered as the 8 nodes do not react at the same time, it takes 17ms for the bus traffic to be restored to normal). Without intruder isolation, it is very hard to tell whether the node is the victim of an attack or he indeed has problems at the transceiver level. More, it is not possible to tell where is the intruder located on the bus. PanoptiCAN-DC solves both problems by isolating bus segments in a daisy-chain manner.



(i) frames altered by the adversary and relay action

(ii) frames restored to normal

Fig. 12. Two frames altered by the adversary followed by relay action (i) and frames restored (ii)

6 Conclusion

Relays are able to provide an efficient active defense mechanism against generic intrusions and DoS attacks in particular. The cost of relays is small and the experiments prove they do not impede regular traffic if properly deployed. In this work we provided some new conceptual architectures with relays that provide good alternatives for intruder isolation by air-gaping CAN networks. The PanoptiCAN provides an adversary-resilient CAN bus architecture that can actively circumvent many types of attacks, including DoS attacks which are difficult to address due to the usual topology of CAN (bus) and its error confinement mechanism. The decentralized version of the Panopti-CAN improves significantly by reducing wiring costs and providing a faster response to intrusions. This comes with a slight disadvantage, i.e., less resilience in front of a DDoS attack, but the cost reduction it offers may be more important since multi-adversary scenarios are less likely in a controlled environment such as in-vehicle networks. We thus emphasize that switchable bus daisy-chain topologies may be practical for preventing intrusions and hope that our work paves way in using such topologies.

References

1. K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Experimental security analysis of a modern automobile," in *Security and Privacy (SP), 2010 IEEE Symposium on*. IEEE, 2010, pp. 447–462.
2. S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, "Comprehensive experimental analyses of automotive attack surfaces." in *USENIX Security Symposium*. San Francisco, 2011.
3. C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," *Black Hat USA*, 2014.
4. *CAN FD - The basic idea*, CAN in Automation (CiA). [Online]. Available: https://www.can-cia.org/can-knowledge/can/can-fd/
5. *Controller Area Network Extra Long (CAN XL)*, CAN in Automation (CiA). [Online]. Available: https://www.can-cia.org/can-knowledge/can/can-xl/
6. B. Groza and P.-S. Murvay, "Security Solutions for the Controller Area Network: Bringing Authentication to In-Vehicle Networks," *IEEE Vehicular Technology Magazine*, vol. 13, no. 1, pp. 40–47, 2018.
7. B. Groza, L. Popa, P.-S. Murvay, E. Yuval, and A. Shabtai, "CANARY - a reactive defense mechanism for Controller Area Networks based on Active RelaYs," in *30th USENIX Security Symposium*, 2021.
8. J. Bentham, *Panopticon - a plan of management for a Panopticon penitentiary-house*, 1791.
9. K.-T. Cho and K. G. Shin, "Viden: Attacker identification on in-vehicle networks," in *Conference on Computer and Communications Security*. ACM, 2017, pp. 1109–1123.
10. ——, "Fingerprinting electronic control units for vehicle intrusion detection," in *25th USENIX Security Symposium*, 2016.
11. M. Barranco, G. Rodriguez-Navas, J. Proenza, and L. Almeida, "CANcentrate: An active star topology for CAN networks," in *International Workshop on Factory Communication Systems*. IEEE, 2004, pp. 219–228.
12. M. Barranco, L. Almeida, and J. Proenza, "ReCANcentrate: A replicated star topology for CAN networks," in *Conf. on Emerging Technologies and Factory Automation*, vol. 2. IEEE, 2005.
13. R. Obermaisser and R. Kammerer, "A router for improved fault isolation, scalability and diagnosis in CAN," in *2010 8th IEEE International Conference on Industrial Informatics*, July 2010, pp. 123–129.
14. K.-T. Cho and K. G. Shin, "Error handling of in-vehicle networks makes them vulnerable," in *Conference on Computer and Communications Security*. ACM, 2016, pp. 1044–1055.
15. D. Souma, A. Mori, H. Yamamoto, and Y. Hata, "Counter attacks for bus-off attacks," in *Conference on Computer Safety, Reliability, and Security*. Springer, 2018, pp. 319–330.
16. S. Nie, L. Liu, and Y. Du, "Free-Fall: Hacking Tesla from Wireless to CAN Bus," *Black Hat USA*, 2017.
17. S. Nie, L. Liu, Y. Du, and W. Zhang, "Over-the-Air: How we Remotely Compromised the Gateway, BCM, and Autopilot ECUs of Tesla Cars," *Black Hat USA*, 2018.
18. *Specification of Secure Onboard Communication*, 4th ed., AUTOSAR, 2017.
19. *Specification of Crypto Service Manager*, AUTOSAR, 11 2020, r20-11.
20. M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Workshop on Embedded Security in Cars*. Bochum, 2004.
21. A. Humayed and B. Luo, "Using ID-Hopping to Defend Against Targeted DoS on CAN," in *Proceedings of the 1st International Workshop on Safe Control of Connected and Autonomous Vehicles*. New York, NY, USA: Association for Computing Machinery, 2017, p. 19–26.

22. W. Wu, R. Kurachi, G. Zeng, Y. Matsubara, H. Takada, R. Li, and K. Li, “IDH-CAN: A Hardware-Based ID Hopping CAN Mechanism With Enhanced Security for Automotive Real-Time Applications,” *IEEE Access*, vol. 6, pp. 54 607–54 623, 2018.
23. B. Groza, L. Popa, and P.-S. Murvay, “Highly Efficient Authentication for CAN by Identifier Reallocation With Ordered CMACs,” *IEEE Trans. on Vehicular Technology*, vol. 69, no. 6, pp. 6129–6140, 2020.
24. A. Palanca, E. Evenchick, F. Maggi, and S. Zanero, “A Stealth, Selective, Link-Layer Denial-of-Service Attack Against Automotive Networks,” in *Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2017, pp. 185–206.
25. P.-S. Murvay and B. Groza, “DoS Attacks on Controller Area Networks by Fault Injections from the Software Layer,” in *International Conference on Availability, Reliability and Security (ARES)*. ACM, 2017.
26. I. Broster and A. Burns, “An analysable bus-guardian for event-triggered communication,” in *RTSS 2003. 24th IEEE Real-Time Systems Symposium, 2003*. IEEE, 2003, pp. 410–419.
27. P. de Moraes, O. Saotome, and M. M. D. Santos, “Trends in Bus Guardian for Automotive Communication-CAN, TTP/C and Flexray,” SAE Technical Paper, Tech. Rep., 2011.
28. H. Sivencrona, T. Olsson, R. Johansson, and J. Torin, “RedCAN/sup TM/: simulations of two fault recovery algorithms for CAN,” in *10th IEEE Pacific Rim International Symposium on Dependable Computing*, March 2004, pp. 302–311.
29. L. Zhang, Y. Lei, and Q. Chang, “Intermittent connection fault diagnosis for CAN using data link layer information,” *IEEE Trans. on Industrial Electronics*, vol. 64, no. 3, pp. 2286–2295, 2016.
30. L. Zhang, F. Yang, and Y. Lei, “Tree-based intermittent connection fault diagnosis for controller area network,” *IEEE Trans. on Vehicular Technology*, vol. 68, no. 9, pp. 9151–9161, 2019.
31. U. Hoff and D. Scott, “Challenges for wiring harness development,” *CAN Newsletter*, pp. 14–19, 2020.
32. *Designing a CAN network*, CAN in Automation (CiA). [Online]. Available: https://www.can-cia.org/can-knowledge/can/design-can-network/
33. T. Hoppe, S. Kiltz, and J. Dittmann, “Security threats to automotive CAN networks–practical examples and selected short-term countermeasures,” in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2008, pp. 235–248.

Appendix A - Further details on the experimental model

The development of our experimental model was quite a laborious work as we implemented an 8 ECU network, a realistic size for what can be found inside modern vehicles. The experimental model, which is common for the PanoptiCAN and the PanoptiCAN-DC, included 8 regular ECUs, one Bus Guardian, up to 20 MCP2551 CAN transceivers, 22 relays, 22 120Ω resistors, 14 CAN wires and additionally 200 jumper wires, all these mounted on a 1000×700mm board. The exact number of components that are used in each of the network configurations is presented in Table 2. Figure 13 provides a detailed depiction of our experimental setup and a bus canary.

Appendix B - Results on an existing CANoe car simulation

To give a better image on the behavior of the current solution, we also test it against adversarial actions on a car simulation in the industry standard tool CANoe. This simulation environment was also used by [33] in one of the first reported attack on CAN

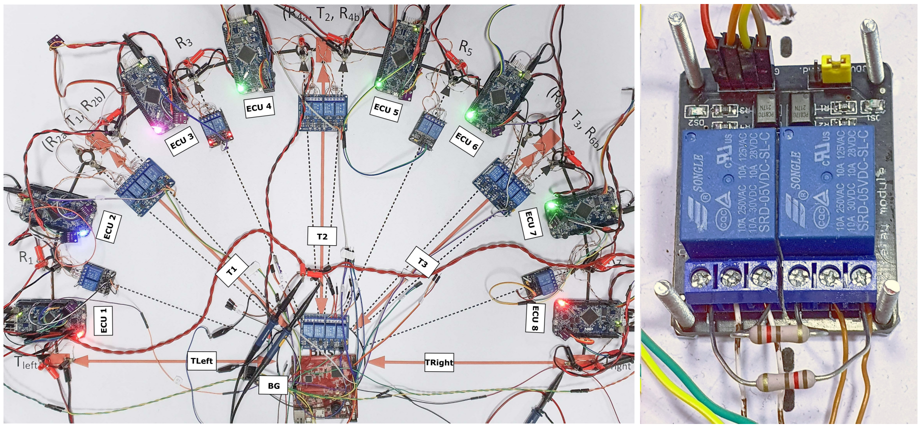


Fig. 13. The 8 ECU PanoptiCAN setup (left) and detailed view of a *bus canary* (right)

Table 2. Detailed component list for the PanoptiCAN and PanoptiCAN-DC

Component → **Setup** ↓	**S12XE**	**TC277**	**MCP2551**	**Relay blocks**	**120Ω resistors**	**CAN wires to BG**	**Jumper wires**
PanoptiCAN	8	1	12	20	22	5	~200
PanoptiCAN-DC w/ BG	8	1	20	12	16	5	~175
PanoptiCAN-DC w/o BG	8	0	16	12	16	0	~150

buses more than a decade ago and in [7] to prove functionality of relay-based isolation in CANARY. The simulation we use contains two buses, one for engine functionalities, e.g., ignition, ABS, etc. and the other for the car body, e.g., doors, lights etc.

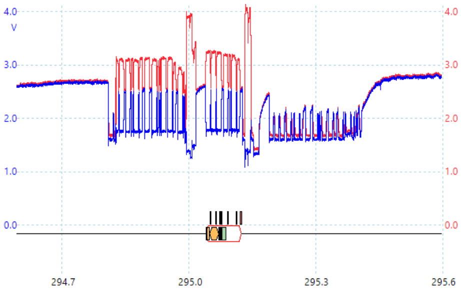


Fig. 14. Adversarial frames and relay action

CANARY reports that a load balancing speed of $50ms$ is needed in order to make the signals look identical in case of a full DoS on the bus [7]. With the PanoptiCAN, there is no need to load balance the network. Once the intruder is isolated, traffic can be easily redirected. A short window of opportunity exists for the adversary until it has been localized. This window of opportunity is however too small to make the attack effective. To clarify this, we outline the response of PanoptiCAN-DC which provides the fastest reaction time. Figure 14 depicts two adversarial DoS frames arriving on the bus. The relays are triggered during the 2-th frame which is actually destroyed by the relay action and as the adversary becomes isolated from the rest of the network. It

Table 3. Brief comparison between CANARY, PanoptiCAN and PanoptiCAN-DC

Topology	**Design**	**Retrofit**	**IDS**	**Transceivers**	**Canaries**	**Cost**	**DoS**	**DDoS**	**Fuz**	**DFuz**
CANARY	Centralized	Yes	Bus Guardian	$n+2$	$n-1$	high	✓	×	✓	×
PanoptiCAN	Centralized	Yes	Bus Guardian	$3n/2+1$	$3n/2-2$	high	✓	✓	✓	✓
PanoptiCAN-DC	Decentralized	No	all ECUs	$2n$	$n-2$	moderate	✓	×	✓	✓

takes less than $3ms$ until this frame is destroyed (the relays that we use have a response time of around $5ms$ according to the technical datasheet). A $3ms$ window of opportunity is too small for an adversary to cause any issues in this simulation. The time of the intrusion detection algorithm to process one frame is well under $100\mu s$ and thus insignificant compared to the relay operation time.

Appendix C - Attack resilience and quantitative comparison

The intrusion detection system (IDS) that we implemented checks for known IDs based on a Bloom filter that was trained to recognize legitimate IDs and DLCs (datafield lengths). A specific threshold, e.g., 2000 frames/s, on the busload is used to trigger the alarm if a flooding takes place while the TEC and REC counters are monitored to detect frame distortion attacks. The intrusion detection mechism is similar to the one we used in CANARY [7] and any other mechanism can be implemented behind the PanoptiCAN. Once an intrusion frame is detected, the localization algorithm starts. In case of PanoptiCAN, this algorithm performs a binary search starting with the canary in the middle of the bus (which splits the bus in half) and proceeds to the left or right according to the direction where the intrusion comes from (in case the intrusion comes from both directions, then both directions are to be inspected). For PanoptiCAN-DC, there is no need to run a localization algorithm since each node will trigger its canaries turning the bus into a daisy-chain (the canaries are disabled if no intrusion is detected after a specific timeout, 5 seconds in our implementation).

In Table 3 we provide a brief quantitative comparison of the two switchable topologies proposed in this work with our previous work CANARY [7]. In terms of attack resilience, we separate between attacks performed by single nodes, i.e., DoS and Fuzzing (Fuz) which stands for generic injections of frames with random content, and distributed versions of them caused by multiple nodes, i.e., DDoS and DFuz. CANARY [7] is in terms of wiring cost similar to a star topology but it requires only 2 additional transceivers to tap the two bus ends, i.e., $n+2$ transceivers for n nodes. The PanoptiCAN is a bit more expensive than CANARY in terms of transceivers since the Bus Guardian will tap the bus after each two consecutive nodes, this requires $n + n/2 + 1$ for n nodes, but it can isolate multiple nodes and thus DDoS and DFuzzing can be prevented. Finally, the decentralized PanoptiCAN-DC requires the same number of transceivers as a regular star, but none of the complex wiring of each node to the gateway, reducing significantly the costs of wiring. Its switchable daisy-chain/bus topology allows it to isolate any number of adversaries being resilient to DFuzzing but not to DDoS (as already explained in the work).