2025 28th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)

Mitigating DoS Attacks on CAN: A Priority-Raise Approach and Its Timing Analysis

Mingqing Zhang and Alejandro Masrur
Department of Computer Science, TU Chemnitz, Germany

***Abstract*—With an increasing level of interconnection, it has become easier for attackers to target in-vehicle buses in industrial and road vehicles alike. One of the most common attacks is Denial-of-Service (DoS), where the bus is flooded with messages disrupting critical functions and, thereby, jeopardizing the safety of operators and/or passengers. In this paper, we are concerned with DoS attacks on Controller Area Network (CAN) – one of the most popular buses – and propose a priority-raise approach (PRA) to mitigate their impact. Basically, we limit the priority of messages arriving through a gateway from outside the vehicle. If a DoS attack is started, this only affects the messages with lower priority than that used by the attacker. Affected messages can then switch to a higher priority reserved for them, i.e., they can raise their priority, to escape the attack. We investigate a centralized as well as a decentralized priority switching and analyze timing for a varying DoS severity.**

I. Introduction

Communication between vehicles, as well as to road-side/backend infrastructure, leading to increased concern about cybersecurity in both industrial trucks (such as forklifts [1] [2]) and road/passenger vehicles [3] [4] [5]. More specifically, an attacker can gain access to in-vehicle buses and disrupt critical functions, posing a threat not only to the safety of operators and passengers, but also to (unaware) bystanders [6]. This is particularly aggravated by increasing autonomous behavior in vehicles, including highly robotized industrial trucks and road vehicles capable of emergency braking and overtaking with minimal human intervention [7] [8]. Cyberattacks targeting these autonomous functions can have severe consequences and must be prevented or mitigated by all means necessary.

One of the most common cyberattacks is DoS, which floods the bus with malicious messages, disturbing or disrupting any communication. DoS attacks are particularly challenging to address because traditional cybersecurity measures such as encryption and authentication are ineffective. Consequently, disconnecting the bus from the external world often seems like the only viable solution. However, this approach cripples essential functionalities that depend on external information, such as real-time traffic updates or warnings received via vehicle-to-everything (V2X) channels in road vehicles. As a result, there is a growing interest in techniques that can mitigate the effects of DoS attacks while maintaining open communication channels with the external world.

In this paper, we focus on a mitigation technique for DoS attacks on CAN. Despite its limitation, such as a reduced payload (of only 8 bytes per frame) and transmission rates (of up to 500 kbps), CAN remains one of the most popular buses in the industry. This is largely due to its high robustness and low cost compared to other technologies such as CAN-FD, FlexRay and (Automotive) Ethernet.

CAN relies on bit-wise arbitration, enabling reliable implementation of (non-preemptive) fixed-priority message transmission. However, this characteristic also renders it vulnerable to DoS attacks, where an attacker can flood the bus with high-priority messages, obstructing legitimate communication. Generally, there are two ways to initiate a DoS attack on CAN. First, with physical access to the bus, the attacker can connect a malicious node, for example, featuring an altered CAN driver (i.e., with an altered firmware/CAN stack) that allows generating error messages on demand. This is otherwise not possible with a legitimate CAN driver. Second, without physical access to CAN, the attacker can still gain remote access through a gateway and flood the bus with high-priority messages. Whereas there is no possible defense against the first attack, it requires the attacker to physically manipulate the vehicle, which is considerably less probable.

Contributions. We consider the second DoS attack scenario with the attacker remotely accessing the bus through a gateway and propose a priority-raise approach (PRA) to mitigate consequences thereof. Basically, we restrict the range of priorities that can be sent through the gateway. If a DoS attack is started, this will only affect legitimate messages with lower priority than those allowed through the gateway, whereas higher-priority messages will experience no significant degradation. Note that this accounts for sufficient design flexibility. That is, under normal conditions, one may still want incoming messages from the gateway (e.g., warnings, traffic updates, etc.) to be prioritized over some of the internal messages.

On the other hand, however, we still want affected (lower-priority) messages to recover from a DoS attack, for which we allow them to switch to a predefined higher priority. To this end, we consider two possible ways of switching priorities: centralized and decentralized. With centralized PRA, once a DoS attack is detected, all messages switch to their reserved higher priority. With decentralized PRA, messages switch to their reserved higher priority individually as their watchdog timer elapses. We illustrate the benefits of the proposed approach based on an extensive evaluation for a varying attack severity and a case study from the automotive domain consisting of adaptive cruise control (ACC).

979-8-3315-2801-0/25/$31.00 ©2025 IEEE

II. Background

In this section, we provide an overview of the CAN bus and discuss the potential for DoS attacks, presenting a realistic attack model.

A. Controller Area Network (CAN)

Due to its robustness and low cost, CAN has been extensively used in the automotive industry, including industrial trucks and road/passenger vehicles. It is a multi-master bus system designed to simplify the wiring between ECUs (electronic control units) in a vehicle. CAN has been designed to ensure real-time behavior through a non-preemptive, fixed-priority transmission scheme, which utilizes bit-wise arbitration to resolve conflicts [9]. If two or more nodes start transmitting simultaneously (after detecting that the bus is idle), the node sending a dominant bit (i.e., a logical 0) wins arbitration over those sending recessive bits (i.e., a logical 1). This means that the lower the frame's ID, the higher its priority will be. Nodes or frames that lose arbitration will need to compete for the bus again after it becomes idle again. However, ongoing transmission of messages, regardless of priority, cannot be preempted.

B. Denial-of-Service (DoS) Attacks

The goal of a DoS attack is to render a service or system unavailable to its users, thereby disrupting normal operations and potentially causing malfunctions [10]. In the context of CAN, an attacker can launch a DoS attack by flooding the bus with a large number of frames/messages in a short period of time, disturbing or even disrupting normal communication. A DoS attack on the CAN can have severe consequences for vehicle safety, imperilling passengers and other road participants alike. For instance, if an attacker gains access to a vehicle's CAN bus, they can conduct a DoS attack by flooding the bus with excessive messages. This prevents legitimate messages, including brake or speed control messages, from being transmitted. As a result, the vehicle may malfunction, potentially leading to a crash.

C. Attack Model

Let us consider a typical automotive CAN configuration for road vehicles as shown in Fig. 1, where an attacker can gain access to the CAN bus through physical (via OBD-II port) or remote (via infotainment system through gateway) interfaces [11]. Even though attacks via the physical interface are nearly impossible to prevent, they are less likely because they require physical access to the vehicle. In contrast, remote attacks over wireless interfaces are more probable and thus constitute the focus of this paper.

III. Related Work

Generally there are two steps to counter DoS attacks on the CAN bus: intrusion detection and attack mitigation. Currently, most research focuses on intrusion detection rather than attack mitigation. There are three main categories of instruction detection systems (IDS):

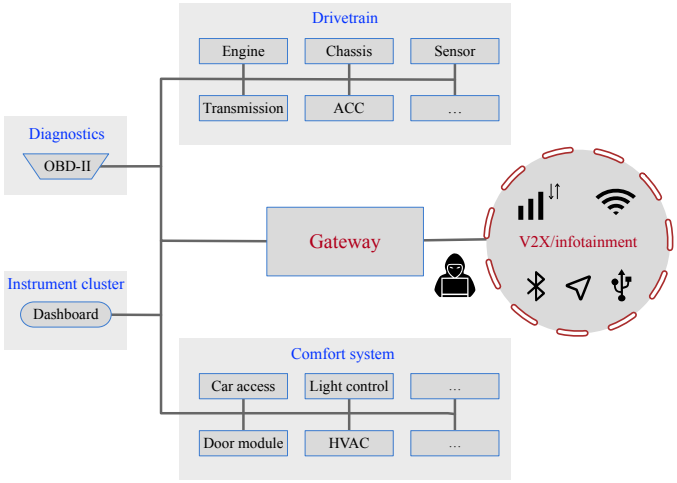


Fig. 1: A typical automotive CAN configuration [12] [13]

- ***Signature-based IDS*** identify an ongoing attack by comparing the network activities with signatures or patterns stored in a database [14]. The disadvantage of a signature-based IDS is that it can only detect attacks with known and documented signatures [15]. Signatures were extracted from standard ECU specifications in [14].
- ***Specification-based IDS*** use a so-called specification, which describes the general behavior of network components, to recognize attacks, i.e., deviations from the specified/expected behavior [16]. Specifications were acquired from CAN in [16] and evaluated against DoS attacks, showing that successful detection heavily depends on the behavior of corresponding ECUs.
- ***Anomaly-based IDS***, similarly, monitor activities on the bus and compare the same for deviations against so-called normal behavior profiles.

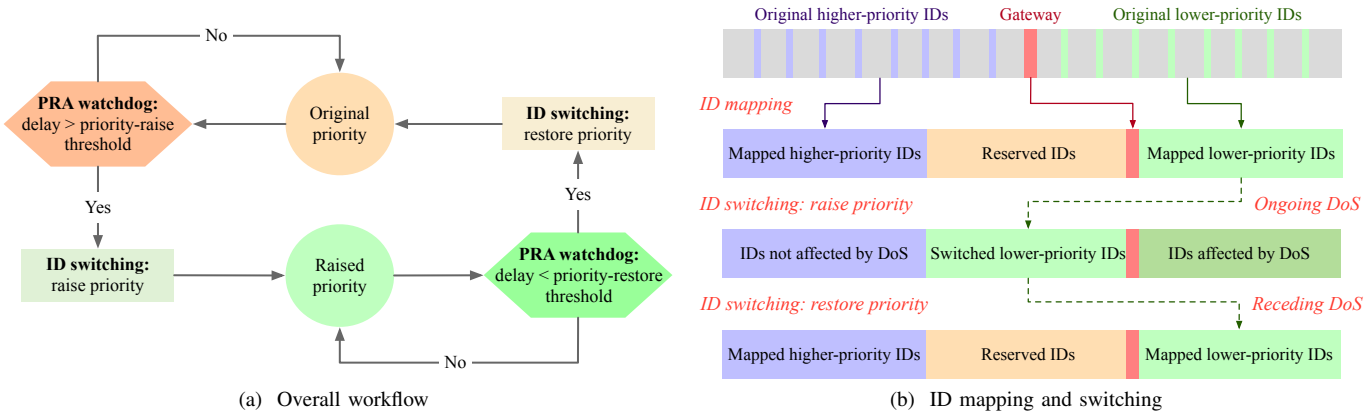
In this paper, instead of relying on IDS techniques, we focus on mitigating DoS attacks without modifying CAN protocol. We analyze the resulting timing properties of our proposed approach and present a real-world case study involving ACC.

IV. Priority-Raise Approach

Whereas the straightforward solution implies detaching CAN from the gateway, this also cancels all communications with the outside world crippling desired functionality that relies on external information such as traffic updates/warnings from other vehicles and/or supportive infrastructure. We rather opt to maintain all communication channels open, but restrict the range of IDs that are allowed through the gateway.

A. Overall workflow

Basically, a simple watchdog (timer) is implemented at the application layer (i.e., without changing the CAN stack/driver) on each endangered node/ECU, i.e., a node sending vulnerable messages – with lower priority than those arriving from the gateway. The watchdog then monitors the transmission delays of such vulnerable messages on the node. If the watchdog identifies unusual delays exceeding a specific threshold –



(a) Overall workflow

(b) ID mapping and switching

Fig. 2: Priority-Raise Approach (PRA)

called priority-raise threshold – of a vulnerable message, it is an indication of an ongoing DoS attack. This triggers what we refer to as *ID switching*, raising the priority of the corresponding message as illustrated in Fig 2a.

With a higher priority, the corresponding message vastly escapes the effects of the DoS attack. However, since CAN implements a non-preemptive transmission scheme, – the ongoing transmission cannot be preempted/canceled – the now lower-priority DoS messages can still block legitimate transmissions, but only for the duration of a single transmission.

If the blocking time caused by lower-priority DoS messages diminishes over time, it indicates that the DoS attack is receding. In other words, once the transmission delays fall within a certain threshold – known as the priority-restore threshold – over a pre-established time interval, an ID switching is triggered again to restore the original priorities.

B. ID mapping and switching

As shown in Fig 2b, originally all CAN messages are scattered in the ID space, where the message IDs allowed through the gateway typically reside in the middle range.

CAN messages are typically classified based on their priorities/IDs in relation to those allowed through the gateway. To facilitate the aforementioned ID switching, message IDs need to be mapped or redistributed prior to operation as shown in Fig 2b. In addition, vulnerable messages – those with lower priority than messages from the gateway – are assigned/reserved higher priorities/IDs to which they can switch in the event of an ongoing DoS attack.

C. Centralized vs. decentralized ID switching

As mentioned above, PRA can either trigger a centralized or a decentralized ID switching mechanism. In a centralized ID switching scenario, the priority of all vulnerable messages synchronously increases. In this process, the first watchdog detecting an ongoing DoS attack sends a broadcast message (clearly, with a priority higher than those from the gateway) to notify all other vulnerable messages to raise their priorities as well. While this approach ensures the preservation of relative priorities among vulnerable messages, it necessitates reserving an additional ID for the priority-raise command.

In contrast, in a decentralized ID switching, vulnerable messages raise their priorities individually as their corresponding watchdogs detect the ongoing DoS attack. This method eliminates the need for a centralized priority-raise command, thus obviating the requirement for an extra reserved ID. The disadvantage is that, depending on the order in which messages raise their IDs, there can be a short-term priority inversion. That is, if a lower-priority message raises its priority before a higher-priority message, the former will contribute to additional delay for the latter. This is, however, not entirely critical, since this additional delay caused on the higher-priority message will end up enforcing a priority raise on its own, which reinstates the priority order. In other words, the extra delay suffered by a particular message is bounded by PRA's priority-raise threshold alone.

Finally, note that again either a centralized or decentralized ID switching can be applied for restoring priorities when the DoS attack recedes.

V. Schedulability Analysis

A DoS attack basically disrupts communication leading to significant transmission delays and deadline misses. While the proposed PRA helps mitigate this effect, it also introduces some timing overhead, which we will now analyze.

The set of messages m_i sent over CAN is denote by M with $1 \leq i \leq n$ and n being the total number of messages in M. Without loss of generality, we assume that all m_i are sorted in the order of decreasing priority, i.e., the greater the index i, the lower the priority of the message. We denote an m_i's transmission time by c_i, which depends on CAN's bandwidth and the number of bits sent, i.e., overhead bits including the minimum inter-message separation, data and stuffing bits. Further, we describe the period of repetition of an m_i by p_i and its deadline by d_i with $d_i \leq p_i$.

Let us consider that all messages in M are vulnerable to a DoS attack and analyze the priority-raise approach. The

priority-restore case, i.e., when messages switch back to their original priorities, can be understood by analogy.

Now, we denote by τ_i the priority-raise threshold of a message m_i, i.e., if m_i requires longer than τ_i to be transmitted, a priority-raise process will commence. There are two possible behaviors depending on whether centralized or decentralized ID switching is implemented.

Centralized ID switching. The first watchdog detecting an ongoing DoS attack will send a high-priority message that triggers a synchronized ID switching for all vulnerable messages. We assume this message has the highest possible priority, minimizing switching delay, and that its transmission time is given by c_{sw}.

We now need to compute the longest possible busy period after ID switching, for which we assume that it takes the longest possible time equal to $\tau_{max} = \max_{i=1}^{n}(\tau_i)$ to start the process. This leads to a busy period as follows:

$$t'_{busy} = \tau_{max} + b_{max} + c_{sw} + \sum_{i=1}^{n} \left\lceil \frac{t'_{busy}}{p_i} \right\rceil c_i, \qquad (1)$$

where b_{max} is the maximum possible blocking time accounting for DoS messages at the moment of switching. Note that only one DoS message can affect ID switching, since the message sent to that end has the highest priority in the system. Hence, $b_{max} \leq c_{max}$ holds with c_{max} being the maximum transmission time of a message with full payload (i.e., 8 bytes). After switching IDs, DoS messages can only be sent when no legitimate message is being transmitted, and they have no further impact on t'_{busy}.

Next, We need to compute the worst-case response time (WCRT) of the k-th transmission of message m_i within this busy period. If m_i was not the message to trigger an ID switching, its WCRT can be easily computed. However, if m_i is the message having triggered the ID switching, its WCRT has to be computed as follows:

$$r'_{i,k} = \tau_i + b_{max} + c_{sw} + k \cdot c_i + \sum_{j=1}^{i-1} \left\lceil \frac{r'_{i,k}}{p_j} \right\rceil c_i, \qquad (2)$$

where again only one DoS message can affect the transmission of m_i once this has raised its priority. In addition, the priority-raise threshold τ_i, the delay for ID switching, and the time for switching c_{sw} will also affect m_i's WCRT.

Clearly, (2) results in the longest possible response time. Hence, the schedulability of CAN messages under centralized PRA can be guaranteed, if the following holds for $1 \leq i \leq n$ and $1 \leq k \leq 1 + \left\lfloor \frac{t'_{busy} - d_i}{p_i} \right\rfloor$ with t'_{busy} given as per (1):

$$r'_{i,k} \leq d_i + (k-1) \cdot p_i. \qquad (3)$$

Decentralized ID switching. If now every vulnerable message performs an individual ID switching whenever its corresponding watchdog detects the DoS attack, no high-priority message is sent and we can relax the computation of busy period to:

$$t''_{busy} = \tau_{max} + b_{max} + \sum_{i=1}^{n} \left\lceil \frac{t''_{busy}}{p_i} \right\rceil c_i, \qquad (4)$$

where b_{max} is defined as for the centralized case. In this case, since every m_i switches to a higher ID individually, the WCRT of m_i's k-th transmission can only be given by the expression:

$$r''_{i,k} = \tau_i + b_{max} + k \cdot c_i + \sum_{j=1}^{i-1} \left\lceil \frac{r'_{i,k}}{p_j} \right\rceil c_i, \qquad (5)$$

where all legitimate messages with higher priority than m_i are assumed to have already switched their IDs (prior to m_i), resulting in the longest possible contention for m_i. Finally, the schedulability of CAN messages under decentralized PRA can be guaranteed, if the following holds for $1 \leq i \leq n$ and $1 \leq k \leq 1 + \left\lfloor \frac{t''_{busy} - d_i}{p_i} \right\rfloor$ with t''_{busy} given as per (4):

$$r''_{i,k} \leq d_i + (k-1) \cdot p_i. \qquad (6)$$

Priority-Raise Thresholds. The designer can choose the value of τ_i based on the concept of DoS tardiness α_i, which is the amount of time by which m_i is allowed to miss its deadline under a DoS attack. Let us consider the case of centralized ID switching and define $r'_i = \max_{\forall k}(r'_{i,k})$, i.e., the longest possible WCRT for an m_i transmission. As a result, τ_i is given by the following expression:

$$\tau_i = d_i + \alpha_i - r'_i. \qquad (7)$$

Note that if τ_i is negative, the selected value of α_i cannot be guaranteed for m_i. Therefore, it is up to the designer to decide on how to proceed, which may ultimately lead to the conclusion that the system is infeasible under the given conditions. Priority-raise thresholds for decentralized ID switching can also be determined in a similar way.

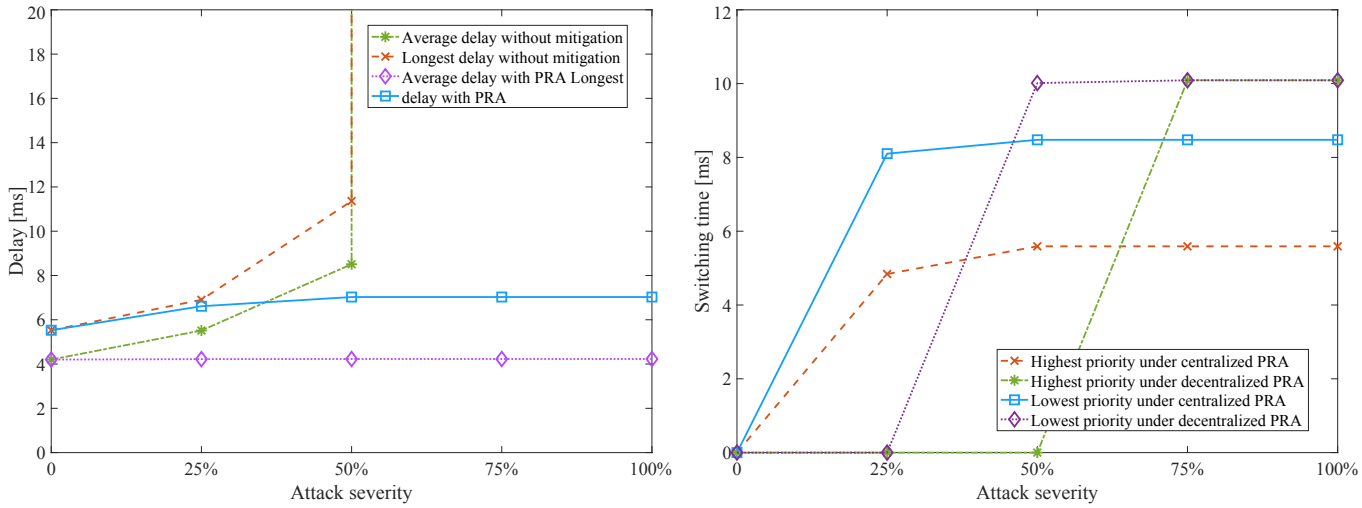
VI. Evaluation

In this section, we begin by presenting and discussing simulation results for various *attack severity*, which we define as the additional utilization generated by an DoS attack on the bus. Further, we delve into the advantages of the proposed approach through a case study involving adaptive cruise control.

A. Varying attack severity

We conducted simulations of DoS attacks with varying severity on CAN using MATLAB/Simulink. To this end, we utilized the real-world CAN message settings, where the message ID ranges from 0x050 - 0x5E0, which result in approximately 25% utilization [17]. Again, we assume that an attacker performs DoS attacks through the gateway, which only allows messages with IDs between 0x471 and 0x479 to pass through. Each of the following scenarios was simulated for a duration of $1000s$ with a granularity of $1\mu s$: DoS attack with no mitigation, DoS attack with centralized PRA and decentralized PRA.

For evaluating our proposed PRA, we configured priority-raise thresholds to half of the periods, assuming deadlines equal to periods. We focused on messages with original IDs in the range of 0x480 to 0x5E0 due to their relatively low priority compared to malicious messages arriving from the gateway. As mentioned earlier, we considered varying attack severity from



(a) Average and longest delays independent of message priority

(b) ID switching times for the highest- and the lowest-priority message

Fig. 3: Simulation results for real-world CAN message setting

0% (no attack) to 100% (the attack completely blocks the bus for vulnerable messages).

Fig. 3 summarizes our simulation results. In Fig. 3a, we first present the average and longest (communication) delays under a DoS attack, independent of message priority. With no mitigation, there is a significant increase in both average and longest delays from 25% to 50% attack severity. Delays become infinite from 50% attack severity onward, indicating that messages never reach their destination. In contrast, PRA maintains the average and longest delays relatively constant throughout the full range of attack severity. This behavior remains consistent regardless of whether a centralized or decentralized ID switching mechanism is implemented.

The ID switching times for both the highest-priority and the lowest-priority message under centralized and decentralized PRA can be observed in Fig. 3b. Both the highest- and the lowest-priority message switch IDs already at a 25% attack severity under centralized PRA. This occurs because one message triggers the ID switching for all vulnerable messages. On the other hand, the same messages undergo an ID switching at 50% and at 75% attack severity under decentralized PRA.

At this point, we can draw the general conclusion that when the attack severity is relatively low, our decentralized PRA outperforms centralized PRA with overall lesser switching time. This is attributed to the fact that many messages, which do not necessarily need to raise their priorities, are 'compelled' to switch under centralized PRA. However, with relatively high attack severity, centralized PRA demonstrates superior performance. This is because other messages, whose delays have not yet exceeded their corresponding priority-raise thresholds, will eventually need switch anyway.

B. ACC as case study

Adaptive cruise control (ACC) primarily functions to maintain a vehicle at a constant speed, as set by the driver – a feature known as speed control – independently of the road profile. However, when a slower vehicle is detected ahead, ACC adjusts the vehicle's speed to maintain a safe distance from the leading vehicle, referred to as spacing control.

In this section, we assess the effectiveness of the proposed Priority-Raise Algorithm (PRA) in mitigating DoS attacks within the context of a case study involving adaptive cruise control (ACC). In this scenario, a vehicle, referred to as the ego car, follows another vehicle, known as the lead car, on a highway while maintaining a safe relative distance or a set speed. The system utilizes radar sensors to detect the lead car and make necessary speed adjustments [18].

Attack Scenario. Suppose the ego vehicle is operating at spacing control, maintaining a safe distance from the lead vehicle ahead. If the leading vehicle suddenly accelerates, ACC will switch to speed control, prompting the ego vehicle to also accelerate until it reaches the driver-set speed. Now, consider a targeted DoS attack occurring precisely at this moment by flooding the bus with fake sensor messages containing greater values of the actual relative (inter-vehicle) distance than the actual ones. The ACC system, being misled by these false messages, may interpret that it is safe to speed up, potentially leading to a collision with the leading vehicle.

We simulated this scenario in Matlab/Simulink with a simulation time of $80s$ and a sample time of $0.1s$, assuming the ego follows the leading vehicle on a flat road with an acceleration range from $-3.5m/s^2$ to $2.5m/s^2$, as per [19].

In the scenario described earlier, we assume the worst-case situation, where the malicious messages from the attacker consistently win arbitration against genuine sensor messages. Consequently, the ACC controller relies on altered data, leading to potentially hazardous outcomes. We analyze the impact of the targeted DoS attack and the effectiveness of our proposed priority-raise approach (PRA) concerning the relative

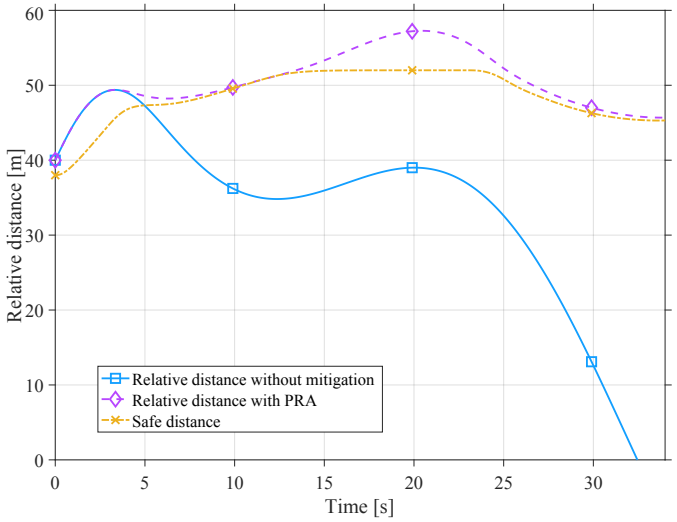


Fig. 4: ACC under targeted DoS attack

distance between the ego and leading vehicles. As depicted in Fig. 4, under a targeted DoS attack and in the absence of mitigation, the actual relative distance between vehicles drops below the safe distance from $5s$ all the way down to $0m$ at $33s$, i.e., a collision is inevitable. However, PRA detects the abnormal delay, which exceeds the priority-raise threshold, leading to priority raise. Following the ID switching, distance sensor messages with the newly switched ID are recognized and accepted by the ACC controller. Consequently, the ACC system resumes operation with legitimate sensor readings, maintaining a safe distance at all times. With our proposed PRA, the relative distance between vehicles remains within in a safe range throughout, successfully mitigating the impact of the targeted DoS attack.

VII. Conclusion

In this paper, we introduced our priority-raise approach (PRA) designed to detect and mitigate DoS attacks on the CAN bus. Unlike approaches that isolate the CAN bus from external world, thereby sacrificing desired functionalities, our method involves restricting IDs/priorities entering through the gateway. This offers significant design flexibility; under normal conditions, messages arriving through the gateway can still be assigned higher priority than (some of) the internal messages, as per the designer's discretion. However, during a DoS attack, when an anomalous delay is detected, vulnerable/affected CAN messages are switched to reserved[1] higher-priority IDs. We analyzed the schedulability of the proposed PRA, considering both centralized and decentralized ID switching mechanisms. Our evaluation results, comprising detailed simulations, demonstrate the efficacy of the proposed PRA in mitigating DoS attacks. Furthermore, we presented a case study discussing a targeted DoS attack on adaptive cruise control (ACC), illustrating how our approach can effectively counter such attacks without requiring modifications to the CAN stack or drivers. This lack of additional costs makes our approach particularly beneficial for cost-sensitive domains such as automotive applications.

[1]Note that the size of the original ID space is rarely a restriction, since CAN allows for 2^{11} possible IDs. Even with ID mapping, this is still sufficient for most applications.

References

[1] H. M. Boland, M. I. Burgett, A. J. Etienne, and R. M. S. III, "An Overview of CAN-BUS Development, Utilization, and Future Potential in Serial Network Messaging for Off-Road Mobile Equipment," in *Technology in Agriculture*, IntechOpen, 2021.

[2] V. Mullet, P. Sondi, and E. Ramat, "A Review of Cybersecurity Guidelines for Manufacturing Factories in Industry 4.0," *IEEE Access*, 2021.

[3] A. Aloseel, H. He, C. Shaw, and M. A. Khan, "Analytical Review of Cybersecurity for Embedded Systems," *IEEE Access*, 2021.

[4] V. K. Kukkala, S. V. Thiruloga, and S. Pasricha, "Roadmap for Cybersecurity in Autonomous Vehicles," *IEEE Consumer Electronics Magazine*, 2022.

[5] M. Zhang, P. Parsch, H. Hoffmann, and A. Masrur, "Analyzing CAN's Timing under Periodically Authenticated Encryption," in *Design, Automation & Test in Europe Conference & Exhibition*, 2022.

[6] M. Zhang and A. Masrur, "Improving Timing Behavior on Encrypted CAN Buses," in *IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, 2019.

[7] D. Krishna Murthy, M. Zhang, and A. Masrur, "Analyzing the Impact of Secure CAN Networks on Braking Dynamics of Cooperative Driving," in *22nd Euromicro Conference on Digital System Design*, 2019.

[8] M. Zhang, D. K. Murthy, P. Parsch, H. Hoffmann, P. H. Kindt, and A. Masrur, "A Periodic Authentication Scheme for Safety/Security Co-Design on CAN Buses," *IEEE Transactions on Vehicular Technology*, 2024.

[9] H. M. Song and H. K. Kim, "Discovering CAN Specification Using On-board Diagnostics," *IEEE Design & Test*, 2020.

[10] Z. Li, H. Zhang, H. Shahriar, D. Lo, K. Qian, M. Whitman, and F. Wu, "Denial of Service (DoS) Attack Detection: Performance Comparison of Supervised Machine Learning Algorithms," in *IEEE International Conference on Dependable, Autonomic and Secure Computing*, 2020.

[11] M. Bozdal, M. Samie, S. Aslam, and I. Jennions, "Evaluation of CAN Bus Security Challenges," *Sensors*, 2020.

[12] T. Huybrechts, Y. Vanommeslaeghe, D. Blontrock, G. Van Barel, and P. Hellinckx, "Automatic Reverse Engineering of CAN Bus Data Using Machine Learning Techniques," in *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*, 2018.

[13] S. F. Lokman, A. Othman, and M. Husaini, "Intrusion Detection System for Automotive Controller Area Network (CAN) Bus System: A Review," *EURASIP Journal on Wireless Communications and Networking*, 2019.

[14] S. Jin, J.-G. Chung, and Y. Xu, "Signature-based Intrusion Detection System (IDS) for In-vehicle CAN Bus Network," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2021.

[15] T. Hoppe, S. Kiltz, and J. Dittmann, "Applying Intrusion Detection to Automotive IT-early Insights and Remaining Challenges," *Journal of Information Assurance and Security*, 2009.

[16] H. Olufowobi, C. Young, J. Zambreno, and G. Bloom, "SAIDuCANT: Specification-Based Automotive Intrusion Detection Using Controller Area Network (CAN) Timing," *IEEE Transactions on Vehicular Technology*, 2020.

[17] R. De Andrade, K. N. Hodel, J. F. Justo, A. M. Laganá, M. M. Santos, and Z. Gu, "Analytical and Experimental Performance Evaluations of CAN-FD Bus," *IEEE Access*, 2018.

[18] L. Elmorshedy, B. Abdulhai, and I. Kamel, "Quantitative Evaluation of the Impacts of the Time Headway of Adaptive Cruise Control Systems on Congested Urban Freeways Using Different Car Following Models and Early Control Results," *IEEE Open Journal of Intelligent Transportation Systems*, 2022.

[19] M. Althoff, S. Maierhofer, and C. Pek, "Provably-Correct and Comfortable Adaptive Cruise Control," *IEEE Transactions on Intelligent Vehicles*, 2021.