VECAEP: A Hands-on Exploration Platform for Vehicular Communication Attacks

Darshith Madvinkodi Prakash, Bhagawat Baanav Yedla Ravi, Srivalli Boddupalli, and Sandip Ray
Department of ECE, University of Florida, Gainesville, FL 32611, USA.
darshith.madvink@ufl.edu, b.yedlaravi@ufl.edu, bodsrivalli12@ufl.edu, sandip@ece.ufl.edu.

2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring) | 979-8-3503-1114-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/VTC2023-Spring57618.2023.10199804

***Abstract*—Vehicular communication systems and their applications have rapidly grown in recent years with the proliferation of cooperative applications. Unfortunately, vehicular network applications can be susceptible to cybersecurity attacks, disrupting the vehicular ecosystem or even causing fatal injuries. Unfortunately, platforms to enable realistic exploration of these vulnerabilities are limited. We address this critical need through the design of a new exploration platform, VECAEP, to enable comprehension of communication attacks. VECAEP permits the user to explore diverse communication attacks and comprehend interactions of different attack parameters and their impacts on the attack. We demonstrate VECAEP with attacks on Cooperative Adaptive Cruise Control.**

I. INTRODUCTION

Modern vehicles include advanced communication capabilities such as Dedicated Short Range Communication (DSRC) and Cellular Vehicle to Everything (C-V2X). Vehicular communications, referred to as V2X, include interactions with other vehicles (V2V), infrastructure components (V2I), and other electronic systems (V2IoT). V2X holds the promise of drastically improving road safety, efficiency, and comfort by enabling vehicles to exchange information about traffic and road conditions. However, an unfortunate impact is the increased vulnerability of vehicles to cybersecurity compromises by vastly increasing the attack surface that can be exploited by an adversary to disrupt the vehicular ecosystem or cause catastrophic accidents. The proliferation of advanced communication technologies critically depends on our ability to comprehend and defend against security subversions.

In spite of this critical need, awareness of cybersecurity concerns in vehicular systems has been limited, even among cybersecurity researchers and practitioners. While several high-profile papers have been published demonstrating compromises to vehicular systems and transportation infrastructure [1], [2], these works still remain niche topics. A key reason is the lack of infrastructure to get a hands-on understanding of how to perform cybersecurity attacks on automotive systems. Mastery of security concepts requires the user to play with the system being hacked, explore various settings, and observe the impact; simply learning the concepts from books or research papers or watching demonstrations of experts performing attacks is inadequate [3]. On the other hand, such exploration must be performed in a controlled environment: it is infeasible and unsafe (and expensive) for a novice user to learn vehicle hacking using actual vehicles.

This project has been partially supported by the National Science Foundation under Grants CNS-1908549 and SATC-2221900.

In this paper, we present an exploration platform, VECAEP, that enables the exploration of attacks on vehicular communication. VECAEP includes a hardware prototype and software tools to provide hands-on experience performing cyber security attacks on vehicular communication applications. We demonstrate VECAEP in the exploration of communication attacks on Cooperative Adaptive Cruise Control.

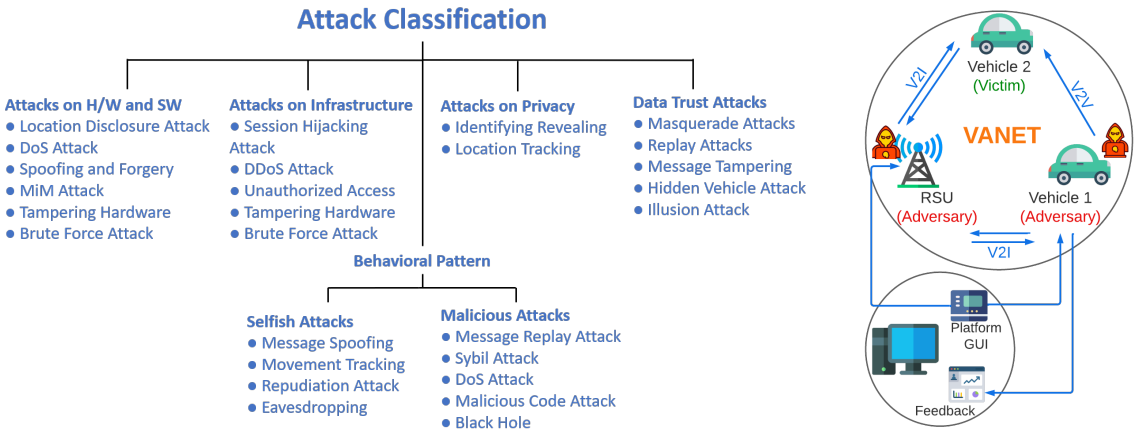
The paper makes the following important contributions:

- To our knowledge, VECAEP is the first exploration platform to enable interactive exploration of vehicular communication attacks.
- In doing so we identify some of the challenges in creating these attacks in a way that can be explored particularly by an inexperienced user.
- We are demonstrating this platform on realistic and representative attacks.

The remainder of this paper is structured as follows. Section II discusses related work. Section III highlights the high-level architecture of VECAEP. Section IV presents VECAEP application for exploring attacks on CACC. We conclude in Section V. For the interested reader, a demonstration video is available showcasing a number of VECAEP features [4].

II. RELATED WORK

Automotive security has garnered significant research interest over the past few years [5]–[8]. However, we are not aware of a hands-on exploration platform to evaluate and perform V2X attacks. Some works used automotive simulators in analyzing security. Heijden et al. [9], demonstrates the use of the Plexe simulator toolkit in the analysis of attacks on CACC controllers. These works portray the analysis of best-case attacks and their effects on vehicular systems. The CONVINCE [10] framework uses synthesis and validation algorithms, modeling, and simulation through a heterogeneous simulator for inter-vehicle communication and intra-vehicle environments. However, neither framework enables hands-on experience to interplay with attack attributes. Ravi et al. [11] enables interactive hands-on exploration of automotive ranging sensor attacks. However, this framework was limited to ultrasonic sensors.



(a) (b)

Fig. 1: Automotive Communication Attacks and VECAEP Infrastructure. (a) Taxonomy of Automotive Communication Attacks in the scope of VECAEP. (b) User-level View of VECAEP. The VECAEP front-end includes a platform GUI and Feedback module that interacts with the platform VANET.

Fig. 2: Physical platform showing the leading, following vehicle, Road Side Unit (RSU), and traffic signs.

III. VECAEP ARCHITECTURE

Fig. 1 shows a pictorial representation of the VECAEP platform design. It includes vehicles and roadside units (RSU) that interact through a vehicular ad hoc network (VANET) to exchange information. Components can be added without disrupting the top-level interface. The user can interact with the VANET through a frontend that can be used to configure attack parameters and provide feedback. The vehicles and RSU are deployed on a pill-shaped track (Fig. 2). The user can introduce attacks at any point during the vehicular movements, by choosing a vehicle or the RSU to be malicious (see below).

A. VECAEP *Frontend*

The frontend accepts user inputs for vehicular controls and V2X attacks. Examples of control parameters include the speed and acceleration of the leading vehicle, selection of desired control models for trials, V2X attack units. Additionally, it provides necessary analytical data as feedback to the user for comprehending attacks, *e.g.*, data from multiple components are aggregated and analyzed to determine the effectiveness of the attack (See Section IV).

B. *VANET Functionality*

Fig. 3 shows the VANET supports interaction among the leading and following vehicles, as well as the RSU. VECAEP enables the user to explore scenarios where either the leading vehicle or the RSU is treated as malicious. The leading module identifies the start and stop conditions for vehicular movement, via inputs from frontend or the steering control unit. The steering control unit determines the necessary conditions for the lane following the algorithm and start/stop conditions on the physical platform: subsequently, the speed control unit actuates the proxy car motors based on inputs provided by the user. The communication unit is responsible for establishing an ad hoc network and exchanging data between vehicles and other nodes in the network. The sensor unit determines the gap between the leading and following vehicles through equipped sensors on the following vehicle. The control unit uses the leading vehicle's acceleration data shared through the communication unit, gap data obtained from the sensor unit, and the current velocity and acceleration data through its speed control unit in a loop to compute the desired acceleration of the following vehicle. Updated acceleration values are fed back to the speed control unit to drive the motors. The RSU module shows the effect of a cybersecurity attack by an infrastructure node. It comprises a network tool controlled by the frontend and is capable of performing communication attacks on the nodes in the network using the communication unit.

C. *Implementation Details*

The VECAEP platform is implemented using Raspberry Pi 4 Model B boards mounted on SunFounder Picar-X robot car [12] chassis that serves as a proxy for the vehicles. RSU attached to a traffic signal replica is also implemented using the Raspberry Pi board. The physical platform has a pill-shaped black lane on which the proxy cars run for multiple exploration iterations. The cars reset to new positions through the curved ends of the pill-shaped lane after every iteration (referred to as the *Reset phase*). The vehicular ad-hoc network is implemented using IEEE 802.11ac hardware on Raspberry

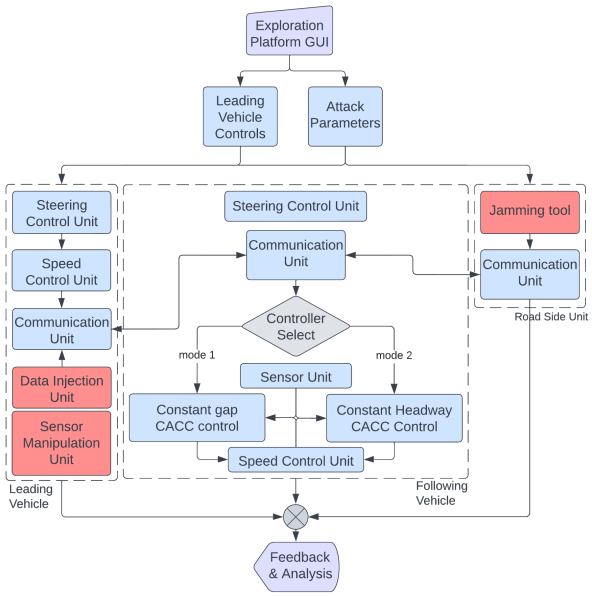


Fig. 3: High-level Architecture of VeCAEP. It involves coordination among possibly adversarial and victim vehicles as well as a possibly adversarial RSU.

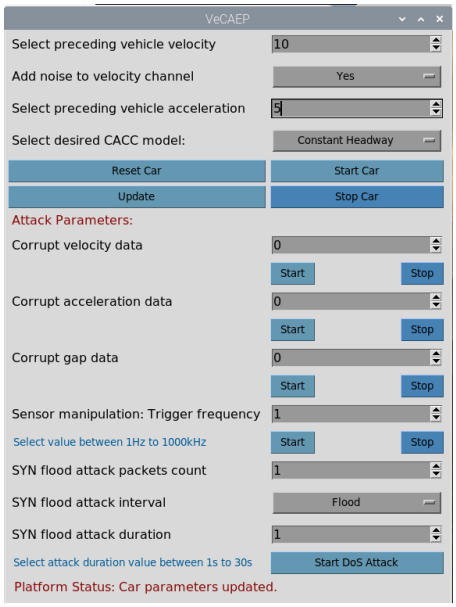


Fig. 4: VeCAEP Frontend. Users can modify parameters and orchestrate attacks. Feedback is also provided to tune parameters to lead to an enhanced attack.

Pi. The Raspberry Pi nodes are configured to operate on frequency band channel 48, which operates at 5.24 GHz. The ad hoc mesh network is created using Better Approach to Mobile Ad-hoc Networking (BATMAN) [13]. Upon power-up, the proxy cars and RSU automatically create and connect to the BATMAN-adv mesh network.

Steering and Speed Control: The vehicle is propelled by two DC motors on the rear wheels of the proxy car. SunFounder Picar-X provides a software library for controlling the servo and DC motors on the proxy car [12]. The speed control unit acts as a lower-level controller, determining the throttle and/or brake commands. It receives acceleration values as input from frontend and computes the desired velocity values using standard kinematics. The steering control unit comprises servo motors and infrared (IR) sensors. The IR sensor array is used with the line-following algorithm to keep the proxy cars on the track. The IR sensor array on the proxy car also detects the start/stop points on the physical platform.

IV. VeCAEP Case Study: Attack on CACC

Although VeCAEP is designed as a generic exploration platform for vehicular communication attacks, it is instructive to see it in action for a specific application. To showcase that, we have put together a demonstration illustrating its use for exploring attacks on CACC. In this section, we briefly recount that demonstration. For the interested reader, a video of the entire demonstration is available online [4]. CACC is a simple two-vehicle car following application, where the following car maintains a constant (time or distance) headway to the leading car based on relative position, velocity, and acceleration values. Position and velocity values are computed through sensors, while the acceleration is communicated by V2X messages. [14] The CACC implementation we use is by Amoozadeh *et al.* [15], which is a constant time headway implementation targeting a time headway of $0.55s$ on stable CACC engagement. Fig. 4 shows the VeCAEP frontend enabling the user to play with various attacks on CACC. Note that it provides several bells and whistles, *e.g.*, the addition of noise, type of attack, communication frequency, etc.

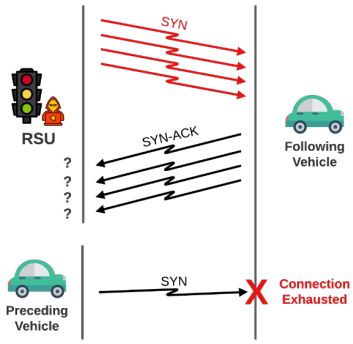


Fig. 5: SYN flood attack. It leads to half-opened connections consuming all server resources, leading to Denial-of-Service.

A. Attack Models

a) Jamming and Flooding: Jamming/Flooding attacks are a popular variant of Denial-of-Service (DoS) attacks in vehicular networks which disrupt communication by overloading the hardware capability [16]. Fig. 5 depicts a SYN flood jamming attack on the target vehicle in VeCAEP. The attacker is an RSU that sends connection requests faster than the vehicle can process and does not send back ACK signals to

the server. This results in the server waiting on the connection requests made, consuming resources on the targeted server and rendering it unresponsive.

b) Spoofing: Spoofing attacks, a form of sensor manipulation attacks cause a false perception of an imaginary object ahead [9]. The sensors used here operate on the time of flight principle, *i.e.*, by measuring the time taken from triggering the sound wave to the reception of the echo, to calculate the distance between sensor hardware and the obstacle ahead. The attacker exploits this principle using additional ultrasonic sensors to perform spoofing attacks. In VeCAEP, the attack sensor is attached at the rear end of the preceding car (adversary). This sensor points towards the target (following) car's sensor and performs an attack by triggering sound waves faster than the bounced-back echo signal of the target vehicle's sensor.

c) Data Injection Attacks: The attack involves a vehicle compromised by an attacker to send mutated packets to the target vehicle, possibly with a falsified payload. The impact of an attack on CACC scenarios depends on the type of fabricated data exchanged between the vehicles and the type of controller used [17]. VeCAEP provides user access to the communication unit on the leading car to fabricate the communication data shared between the proxy cars. The communication data include the vehicle statistics of the preceding proxy car: velocity, acceleration, and position.

Remark 1: The platform uses a kali Linux network tool called hping3 [18] performing communication attacks. The tool runs on the RSU node of the ad-hoc network by taking inputs from the VeCAEP application GUI. Flooding attacks are performed by providing inputs like the number of packets sent to the target vehicle, packet interval, and the duration of an attack. During an attack, the connection between proxy cars is disrupted, causing a loss of Basic Safety Message (BSM) packets on the following car, which might lead to performance degradation or cause a collision.

B. Feedback and Analysis

The novelty of VeCAEP is primarily in the way it enables a user to explore the attack and its impact. For instance, the attack analysis tool uses the gap trends of the two vehicles to analyze the quality of attacks. During jamming attacks, the time difference between the reception of data packets is used to identify the success of the SYN flood attack. Similarly, the difference in trend between the safe gap and the current gap is used to determine performance degradation or collision possibility during sensor manipulation and data injection attacks. Figs. 6, 7, 8, and 9 show representative plots for a benign run and three attack runs respectively.[1] Each run is analyzed by VeCAEP to determine how the attack performed, *e.g.*, in Fig. 8, the feedback includes the observation that the trend between the time interval 35s(+1.6654337e9) to 42s(+1.6654337e9) shows misbehavior in gap values perceived by the following vehicle.

[1]The plots are unsmooth due to noise added to the channels as an input.

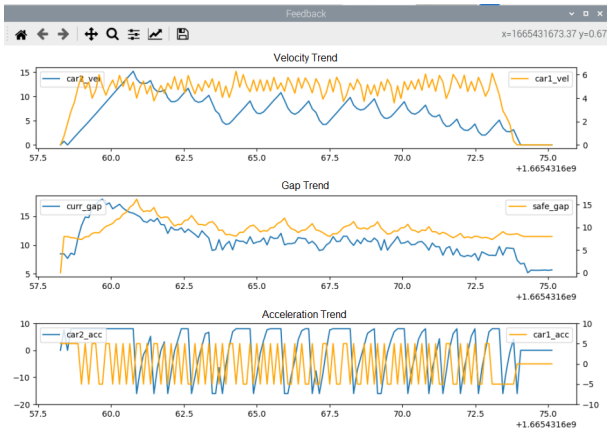


Fig. 6: Benign Scenario Feedback. Leading vehicle (car1), Following vehicle (car2) velocities, acceleration, and gap are shown.

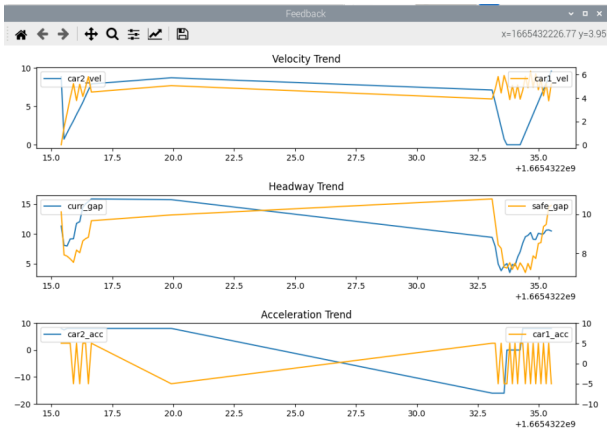


Fig. 7: Jamming Attack Scenario Feedback. Between 17.5s and 32.5s (+1.6654322e9), there is a loss of data transfer.

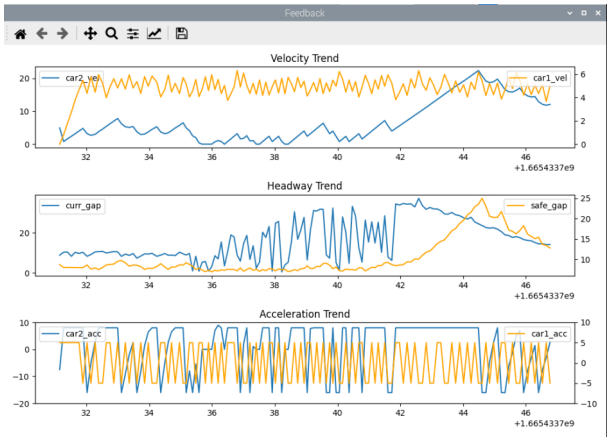


Fig. 8: Sensor Manipulation Attack Scenario Feedback. Spoofed signals are crafted and sent by the preceding vehicle. Between 35s to 42s (+1.6654337e9), the headway/gap value perceived by the following vehicle increases significantly, showing that the attack is effective.

Fig. 9: Data Injection Attack Scenario Feedback. The car2 safe_gap plot intersects with the actual gap (curr_gap) *e.g.*, at 64s (+1.6654e9) indicating performance degradation.

Remark 2: (Implementation Challenges) Since the CACC is a longitudinal control application, there is a need for a steering mechanism to keep the cars in a straight line [19]. Identifying start and end points by the car to prevent proxy cars from going off track or to have enough runway is non-trivial. Addressing these issues involved implementing a steering control unit with an array of infrared sensors. Furthermore, accessing mesh nodes from a non-mesh device to realize the communication was addressed by additional WiFi hardware that was used to create a bridge node. The values used in high-level CACC controller equations like minimum safe gap, and preceding and following vehicle's deceleration, were configured to the exploration platform requirements based on feedback obtained from multiple trials.

C. VeCAEP *Effectiveness*

VeCAEP enables the user to configure various sequences of attack scenarios and provide fine-grained analysis of the behavior of the victim vehicles. Note that although it uses a miniaturized environment for deployment in laboratory environments, the effect and conclusions correspond to real-life scenarios. For instance, in the jamming attack scenario, there is a loss in data between 17.35 and 32.5 which makes the acceleration of the lead car appear to increase, contrary to the ground truth. The corresponding car following action results in a reduced gap as shown in Fig.7. Correspondingly, in a sensor manipulation attack scenario, a spoofed signal indicates the closer presence of the leading car. To avoid a collision the following car tries to increase the gap (Fig. 8). These effects correspond to the physical reality of such attacks under real-life scenarios [5], [6].

V. Conclusion and Future Work

We have presented to our knowledge the first platform for hands-on exploration of vehicular communication attacks. VeCAEP is a configurable, extensible exploration platform for helping novice users explore V2X communications and their security implications. Developing such a platform entails challenges beyond simply replicating the attacks: the system must enable the user to play with different parameters, comprehend the impact of a certain parameter setting, and provide useful feedback. Furthermore, the attack impact as visualized in the platform should be consistent with a real-world attack, while accounting for the miniaturization of the platform and the simplification of the proxy cars compared to real vehicles. We showed how to address these challenges in VeCAEP. We also demonstrated the viability of VeCAEP in the exploration of realistic attacks on CACC and V2X implementations.

In future work, we will integrate more attacks with VeCAEP. We will also consider replacing proxy vehicle hardware with wider chassis that can accommodate industry-standard DSRC/C-V2X modems as communication units.

References

[1] S. Checkoway *et al.*, "Comprehensive experimental analyses of automotive attack surfaces." in *USENIX Security Symposium*, vol. 4. San Francisco, 2011.
[2] C. Miller and C. Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle," in *BlackHat*, 2015.
[3] S. Bhunia and M. Tehranipoor, *Hardware Security: A Hands-on Learning Approach*. Morgan Kaufmann, 2018.
[4] D. M. Prakash, "Vecaep: A hands-on exploration platform for vehicular communication (v2x) attacks," YouTube, urlhttps://www.youtube.com/watch?v=YAvMjm6eUbc.
[5] A. Ghosal and M. Conti, "Security issues and challenges in v2x: A survey," *Computer Networks*, vol. 169, p. 107093, 2020.
[6] B. Mokhtar and M. Azab, "Survey on security issues in vehicular ad hoc networks, alexandria eng," *J*, vol. 54, no. 4, pp. 1115–1126, 2015.
[7] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: Vanets and iov," *Ad Hoc Networks*, vol. 61, pp. 33–50, 2017.
[8] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Transactions on Intelligent transportation systems*, vol. 16, no. 2, pp. 546–556, 2014.
[9] R. Van der Heijden, T. Lukaseder, and F. Kargl, "Analyzing attacks on cooperative adaptive cruise control (cacc)," in *2017 IEEE VNC*. IEEE, 2017, pp. 45–52.
[10] B. Zheng, C.-W. Lin, H. Yu, H. Liang, and Q. Zhu, "Convince: A cross-layer modeling, exploration and validation framework for next-generation connected vehicles," in *2016 IEEE/ACM ICCAD*. IEEE, 2016, pp. 1–8.
[11] B. B. Y. Ravi, M. R. Kabir, N. Mishra, S. Boddupalli, and S. Ray, "Autohal: An exploration platform for ranging sensor attacks on automotive systems," in *2022 ICCE*. IEEE, 2022, pp. 1–2.
[12] GitHub, "Sunfounder/picar-x at v2.0," https://github.com/sunfounder/picar-x/tree/v2.0.
[13] Batman-Adv, "Batman-adv the linux kernel documentation."
[14] K. C. Dey, L. Yan, X. Wang, Y. Wang, H. Shen, M. Chowdhury, L. Yu, C. Qiu, and V. Soundararaj, "A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (cacc)," *IEEE T-ITS*, vol. 17, no. 2, pp. 491–509, 2015.
[15] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon management with cooperative adaptive cruise control enabled by vanet," *Vehicular communications*, vol. 2, no. 2, pp. 110–123, 2015.
[16] S. Malik, P. Bandi, and W. Sun, "An experimental study of denial of service attack against platoon of smart vehicles," in *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*, 2021, pp. 23–30.
[17] R. A. Biroon, Z. A. Biron, and P. Pisu, "False data injection attack in a platoon of cacc: real-time detection and isolation with a pde approach," *IEEE T-ITS*, vol. 23, no. 7, pp. 8692–8703, 2021.
[18] Kali Linux, "Hping3: Kali linux tools," https://www.kali.org/tools/hping3/, Last accessed on 2022-12-02.
[19] Z. Wang, G. Wu, and M. J. Barth, "A review on cooperative adaptive cruise control (cacc) systems: Architectures, controls, and applications," in *2018 21st ITSC*. IEEE, 2018, pp. 2884–2891.