Thesis for The Degree of Licentiate of Engineering

A Simulation-based Methodology to Assess the Impact of Jamming Attacks on Interconnected Automated Road Vehicles

Mateen Malik

Division of Computer and Network systems
Department of Computer Science & Engineering
Chalmers University of Technology and Gothenburg University
Gothenburg, Sweden, 2024

A Simulation-based Methodology to Assess the Impact of Jamming Attacks on Interconnected Automated Road Vehicles

Mateen Malik

Department of Computer Science & Engineering
Division of Computer and Network systems
Chalmers University of Technology and Gothenburg University
Gothenburg, Sweden

This thesis has been prepared using LaTeX.
Printed by Chalmers Reproservice,
Gothenburg, Sweden 2024.

"Securing interconnected safety-critical systems, is a relentless journey, not a destination."
- Anonymous

Abstract

This thesis addresses security benchmarking of Cooperative Driving Automation (CDA) applications, focusing on simulation-based assessment of the consequences of jamming attacks. CDA systems are expected to improve the safety, fuel efficiency, and passenger comfort of future road vehicles. These systems rely on data received wirelessly from other vehicles and roadside installations and must, therefore, be resilient to attacks conducted via the wireless channel.

We propose a framework for benchmarking the resilience of CDA applications against various types of jamming attacks through simulations. To this end, we have developed a simulation engine for communication-based fault and attack injection experiments called ComFASE, which utilizes four existing simulators: Plexe, Veins, OMNet++, and SUMO.

We illustrate our benchmarking approach by conducting a series of simulations where we study the impact of different types of jamming attacks on a longitudinal control algorithm for platooning, which is provided in the Plexe framework. We propose and investigate simulation models for five types of jamming attacks: *delay attack*, *denial-of-service (DoS) attack*, *barrage jamming*, *deceptive jamming*, and *destructive interference*. We implement these models in a simulation model of the physical layer of the IEEE 802.11p standard provided in Veins.

We emphasize that the work presented in this thesis constitutes a first step towards defining a framework for security benchmarking of CDA applications. Such a framework must consider various aspects related to system design, environmental conditions, attack types, and system use cases. We address only a few of these aspects, specifically for jamming attacks: the *driving scenario*, the *attack model*, and the *attack model parameters*.

Our attack models are based on three primary parameters: *attack duration*, *attack start-time*, and *attack value*. The first two are defined in relation to the time axis of the driving scenario, while the attack value determines the nature or strength of the attack. Our results show that a significant number of the simulated attacks caused collisions among the vehicles in the platoon. They also show that the outcome of the simulations is highly dependent on the driving scenario, the attack types, and the attack model parameters.

Keywords: network simulators, vehicle simulators, attack injection, jamming attacks, Cooperative Driving Automation (CDA), connected automated road vehicles, simulation-based testing

List of Publications

Appended publications

This thesis is based on the following publications:

[A] **Mateen Malik**, Mehdi Maleki, Peter Folkesson, Behrooz Sangchoolie, Johan Karlsson "ComFASE: A Tool for Evaluating the Effects of V2V Communication Faults and Attacks on Automated Vehicles"
52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2022.

[B] Mehdi Maleki, **Mateen Malik**, Peter Folkesson, Behrooz Sangchoolie, Johan Karlsson "Modeling and Evaluating the Effects of Jamming Attacks on Connected Automated Road Vehicles"
Pacific Rim International Symposium on Dependable Computing (PRDC), 2022.

[C] **Mateen Malik**, Behrooz Sangchoolie, Johan Karlsson "A Simulation-based Security Benchmarking Approach for Assessing Cooperative Driving Automation (CDA) Applications"
EAI INTSYS 2024 8th EAI International Conference on Intelligent Transport Systems.

Other publications

The following publication is not included in the main licentiate thesis.

[a] **Mateen Malik**, Maytheewat Aramrattana, Mehdi Maleki, Peter Folkesson, Behrooz Sangchoolie, Johan Karlsson "Simulation-based Evaluation of a Remotely Operated Road Vehicle under Transmission Delays and Denial-of-Service Attacks"
Pacific Rim International Symposium on Dependable Computing (PRDC), 2023

Acknowledgment

First of all, I would like to thank my main supervisors, Johan Karlsson from Chalmers Technical University and Behrooz Sangchoolie, who is my industry supervisor from RISE Research Institutes of Sweden, for their patience, unwavering support, and guidance. You are my source of inspiration, and I look forward to continuing my research endeavor under your supervision. I am also very thankful to my examiner, Ioannis Sourdis from Chalmers Technical University, whose support and constructive feedback have been invaluable.

Equally, I would like to express my gratitude to all my co-authors, Peter Folkesson, Mehdi Maleki, and Maytheewat Aramrattana whose meaningful contributions and insights have been of immense value. I sincerely thank Mehdi Maleki, a co-author of most of my published papers and a dear friend. Thank you, Mehdi, for our outstanding scientific collaboration and fruitful discussions.

I especially want to thank my manager, Åsa Olsson, at RISE Research Institutes of Sweden. Her unwavering support has been invaluable to me. Whenever I faced challenges or uncertainties, Åsa was always there to help me.

I am grateful to the VALU3S (Grant Agreement No. 876852), SUNRISE (Grant Agreement No. 01069573), and AGRARSENSE (Grant Agreement No. 101095835) research projects. Their provision of resources and international collaboration opportunities not only facilitated the progress of my research but also played a crucial role in enabling the publication of my work.

Lastly and most importantly, I would like to express my deepest gratitude to my family, especially my wife, Qurat-Ul Ain, for her love, understanding, and unwavering support. Her support has been a constant source of strength and motivation throughout this research journey. I am also thankful to my two children, Hashim Malik, who is five years old, and Ayzal Malik, who is three years old, for their patience and understanding during my periods of intense work. I will make it up to you guys. I also would like to thank my Mom, Shamim Akhtar, and my brother, Mubeen Malik, and his family for their unwavering belief in me and unconditional support. I, in fact, cannot thank you enough.

Contents

Chapter 1

Thesis Summary

1 Introduction

Road vehicles have evolved from mechanical machines into interconnected cyber-physical systems that offer improved safety, fuel consumption, and driver assistance. The development of vehicle-to-vehicle (V2V) and vehicle-to-everything (V2X) communication technology provides pivotal steps in facilitating cooperative driving automation (CDA) applications [1]. CDA facilitates tasks such as platooning, intersection management, and cooperative lane changes, which improve traffic efficiency and safety by sharing real-time information. Although CDA is expected to improve the safety, efficiency, and dependability of automated and connected vehicle transportation systems, additional research is required to tackle the safety and security challenges associated with systems that depend on sensor data sent via wireless networks.

Protecting a cooperative driving application from security threats is a demanding and expensive endeavor that requires many labor-intensive activities [2], [3]. This thesis focuses on one such activity: *conducting simulation-based assessments to evaluate a cooperative system's resilience against security attacks, particularly those targeting its wireless communication channel through radio frequency jamming.*

Given the substantial costs and test repeatability issues associated with proving ground and real-world testing, simulation-based testing has become a common approach for assessing performance, safety and security attributes of CDA applications like platooning [4], [5]. Legal bodies, such as the UNECE (United Nations Economic Commission for Europe) that develops internationally harmonized regulations and standards [6]–[8], along with automotive OEMs (Original Equipment Manufacturers) [9], [10], are increasingly adopting simulation-based testing for the verification and validation of CDA applications.

Since CDA applications rely on wireless communication for cooperation and coordination, securing against potential cybersecurity attacks is crucial. According to the communication jamming taxonomy proposed by Lichtmann et al., [11], security attacks directed against the physical layer of the communication channel can be broadly classified into two main categories: *jamming attacks* and *cyberattacks*. In a jamming attack, the adversary transmits a radio signal that interferes with the legitimate signal to prevent the successful delivery of data packets. A jamming attack can, therefore, be viewed as a *denial-of-service* attack against the wireless communication system. In a cyberattack, on the other hand, the adversary aims to compromise the system by sending erroneous or deceptive information in packets that do not violate the specification for the physical layer. A comprehensive review of different types of cyberattacks and potential countermeasures is provided by El-Rewini et al. [2].

As noted by Lichtmann et al. [11], both jamming attacks and cyberattacks often serve a more sinister purpose than just a denial-of-service (DoS), e.g., the ultimate goal of an attack against a CDA application can be to cause vehicle collisions and traffic jams. Jamming attacks represent a significant category of security threats because they are relatively simple to execute. They require minimal knowledge about the targeted system, primarily limited to the protocol specifications for the wireless network, which are typically accessible in public

standardization documents.

This thesis introduces and demonstrates simulation models for five jamming attacks: *delay attack*, *denial-of-service (DoS) attack*, *deceptive jamming*, *barrage jamming* and *destructive interference*. These jamming techniques represent different types of jamming in terms of the capabilities that an adversary needs for their implementation. We conduct our attack injection experiments using the ComFASE simulation engine, which has been developed in our research group at RISE [12]. ComFASE utilize a combination of four established frameworks, Plexe [4], OMNet++ [13], Veins [14], and SUMO [15].

The work presented in this thesis is intended as an initial contribution towards defining a *simulation-based security benchmarking framework* for assessment of CDA applications concerning their ability to operate safely in the presence of jamming attacks. Work on benchmarks for assessing, testing, and evaluating essential system properties has a history that goes back several decades. Examples include benchmarks for computing performance [16], transaction processing [17], [18], dependability [19], and security [20]. In the field of intelligent transportation systems, several papers have addressed simulation-based assessment of the resilience against security attacks for CDA systems, mainly for platooning systems [2], [3], [21]–[24]. However, only a few of these papers specifically deal with jamming attacks [24]–[27] and, to our knowledge, no previous paper has discussed the idea of defining security benchmarks for simulation-based assessment of CDA applications.

1.1 Research Approach

Our research approach uses a simulation-based testing methodology. Simulation-based test methods enable the execution of extreme driving scenarios that would be costly and potentially dangerous to set up in the real world. Conducting these scenarios outside of a controlled environment could result in significant property damage or human injuries. Typical scenarios, such as vehicles abruptly accelerating or performing emergency braking, are ideal for testing in a simulation-based environment. This test method facilitates the early identification of vulnerabilities and testing under various conditions, ensuring that CDA applications are robust, resilient to cyberattacks, and able to operate safely in real-world scenarios.

Our approach to attack injection testing is similar to the traditional fault injection testing method, where faults are introduced into a system to evaluate its safety. Attack injection testing technique deliberately introduces malicious data or corrupts legitimate information to evaluate a system's resilience against specific types of security attacks. This technique can be used in the real world as well as in a simulation-based environment to verify and validate the safety of interconnected automated vehicles under attack.

Real-world testing offers several benefits; it verifies the system's reliability under true operating conditions and can reveal problems like hardware malfunctions that is not possible to detect in simulations. Real-world testing also have some limitations such as it can pose significant safety risks, add substantial costs of physical testing, and make it challenging to replicate the tests.

In contrast, simulation-based testing offers several advantages, including lower costs, reproducibility, test automation, the ability to explore edge cases safely, and the early identification of design flaws. These benefits contribute to the effectiveness of simulation-based testing in developing and evaluating interconnected vehicle systems. However, the quality of the results obtained from the simulation testing is tightly connected to the fidelity of the models with respect to the actual system.

We developed a ComFASE tool, a communication-based fault and attacks simulation engine that allows detailed modeling and simulation of jamming attacks against cooperative vehicular systems. Using ComFASE, we designed and executed extensive test campaigns to investigate the impact of such attacks on the safety of a platooning system.

A long-term goal of this research is to investigate the possibility of defining standardized procedures for testing and evaluating the safety of CDA applications. As an initial step towards this goal, we propose a tentative framework for benchmarking the resilience of CDA applications against jamming attacks. The primary components of our security benchmarking framework include attack models and driving scenarios. The attack models simulate different types of jamming attacks, such as barrage jamming and destructive interference. The key parameters of our jamming attack models include *attack start-time*, *attack duration*, and *attack value*. The attack start time indicates the start of an attack. The attack duration specifies how long an attack remains active. The attack value is defined differently for different jamming attacks types. In general, the attack value represents the power, strength, or intensity of an attack.

1.2 Simulation of Jamming Attacks

We propose simulation models for five types of jamming attacks and use them to study their impact on a platooning application. The types of jamming attacks we model are: *denial-of-service (DoS) attacks*, *barrage jamming*, *deceptive jamming*, *destructive interference*, and *delay attacks*. All these jamming attacks are implemented by modifying the physical layer simulation model provided in the Veins simulation framework.

In the context of this thesis, a **denial-of-service (DoS) attack** is one that completely blocks the wireless communication channels for an extended time. We model DoS attacks by manipulating the value of a parameter called *propagation delay* which is defined in the physical layer model provided in Veins.

In **barrage jamming** attacks, the adversary transmits high-power noise over a broad spectrum of frequencies and, as a result, partially or completely blocks the transmission or reception of legitimate signals. We model the barrage jamming attacks by manipulating a Veins parameter called *Noise*.

Deceptive jamming involves transmitting fake signals that mimic legitimate signals in frequency and power, confusing receivers and consequently causing them to process false or misleading information. We model the deceptive jamming attacks by manipulating a Veins parameter called *Interference*.

Destructive interference occurs when a well-crafted malicious signal in terms of time, frequency, and space is transmitted to interfere with legitimate signals, causing signal distortion or signal cancellation. We model the destructive interference attacks by integrating a destructiveness D parameter in the Veins simulator.

Communication delays can occur when messages are intercepted and prevented from reaching their intended recipients. These intercepted messages are then retransmitted at a later time to confuse the system. We model delay attacks by manipulating the Veins parameter called *Propagation delay*, i.e., the same parameter used for modelling DoS attacks.

1.3 Research Contributions

The primary contributions of this thesis are:

C1 Development of simulation engine ComFASE (Communication-based Fault and Attack Simulation Engine) designed for comprehensive modeling, configuration, and experimentation with two types of jamming attack models. These models include *delay attacks*, *denial-of-service (DoS) attacks. (Paper A)*

C2 Extending ComFASE with three jamming attack models, *barrage jamming*, *destructive jamming*, and *deceptive jamming*. This enables us to perform an in-depth analysis of the impact of jamming attacks on a platooning system consisting of four vehicles using the sinusoidal driving scenario. *(Paper B)*

C3 Examining of how the various parameters of the attack model, such as *attack start time*, *attack duration*, and *attack value*, influence the results of the simulations. *(Paper A, Paper B, Paper C)*

C4 Analysing and comparison between two different driving scenarios (i.e., sinusoidal and braking) and their influence on the outcomes of barrage jamming attacks on a platoon of four vehicles. *(Paper C)*

C5 A proposal for a conceptual framework for the future development of simulation-based security benchmarking for platooning and other CDA systems. *(Paper C)*

1.4 Research Questions

The platooning application, which is the target of our attack injection experiments, is implemented in the Plexe simulation framework. Our study focuses on injecting jamming attacks and assessing the jamming resilience of this platooning application. This raises specific research questions that will be explored in this thesis. The first research question is formulated as follows.

Q1 To what extent does the choice of jamming technique affect the likelihood that an attacker succeeds in causing the vehicles in the platoon to collide?

We address Q1 in Paper A, Paper B, and Paper C. Paper A provides a comprehensive analysis of the effects of communication delays and denial of service (DoS) attacks. Paper B examines the impact of *barrage jamming*, *deceptive jamming*, and *destructive interference* attacks. Paper C analyses how *barrage jamming* affects different vehicles in a platoon.

Q2 How do the parameters of our attack models, i.e., the attack start-time, the attack duration, and the attack value, affect the outcomes of the jamming attacks?

In the second research question, we aim to investigate how the attack model parameters affect the distribution of the outcomes. As previously mentioned, our attack models include three basic parameters: *attack start-time*, *attack duration*, and *attack value*. The attack start time and attack duration are defined in relation to the time axis of a driving scenario that describes the motions of the simulated system of vehicles.

The definition of the attack value varies depending on the jamming technique. For example, in *barrage jamming*, the attack value represents the power of the interfering noise signal. In *destructive interference*, the attack value represents the amount of destructiveness that influences the legitimate signal.

We address Q2 in Paper A, Paper B, and Paper C. In Paper A, we examine the impact of the attack parameters for *delay* and *DoS* attacks. Paper B examines the impact of the attack parameters for *barrage jamming*, *deceptive jamming*, and *destructive interference* attacks. Similarly, Paper C analyzes the impact of attack parameters presented in question two for *barrage jamming* and *destructive interference* attacks (see Fig. 2).

Q3 How does the driving scenario influence the outcomes of jamming attacks on a platoon?

The third research question highlights the impact of the driving scenario on the outcome of the jamming attacks. We investigate this question by conducting jamming attacks for two driving scenarios: the *sinusoidal* and the *braking* scenarios. These scenarios are included in the Plexe simulation framework and have previously been used in several research studies related to vehicle platooning [3], [28]–[36]. We address Q3 in Paper C.

Q4 Are certain vehicles in the platoon more likely to cause collisions?

The fourth research question focuses on attacks that lead to collisions, more specifically, on identifying the first vehicle in the platoon that hits another

vehicle from behind (rear-end collision). We refer to this vehicle as the *collider vehicle*. We expect data on the collider vehicle distribution to be an essential input to our future work, which aims to develop an attack-resilient cooperative adaptive cruise controller. We address Q4 in paper C.

1.5 Thesis Structure

The remainder of this thesis is organized as follows. Chapter 2 provides short descriptions of background information that is essential to our work, including an overview of related work. Chapter 3 summarizes the three conference papers that constitute the main contributions of the thesis. Chapters 2, 3, and 4 contain reprints of these papers.

2 Background

This section provides essential background information that serves as a starting point for our work. In Section 2.1, we provide an overview of the platooning application and the CACC controller that we investigate in our attack injection experiments. Section 2.2 provides an overview of the wireless communication concepts relevant to our work. Section 2.3 gives an overview of different jamming techniques described in the literature. We conclude this chapter with a presentation of related work presented in section 2.4.

2.1 Platooning Application

Platooning is a cooperative driving technology in which a group of vehicles, known as a platoon, travels closely together at high speeds, maintaining a small distance between each other. The vehicles in a platoon are equipped with advanced communication systems and cooperative cruise controllers that allow them to share information and coordinate their movements.

The attack injection experiments described in this thesis have all been conducted with a platooning application that is available in the Plexe framework. This platooning application is based on a CACC controller that implements the longitudinal control law described by Rajamani et al [37]. We refer to this controller simply as CACC in paper A and B. However, since Plexe includes several types of CACC controllers, we decided to rename this controller to P1 (Plexe 1) in Paper C to avoid confusion with other controllers.

The P1 controller consists of an upper-level controller and a lower-level controller. The upper-level controller determines the desired acceleration for each vehicle in the platoon to maintain the desired spacing between the vehicles and ensure the platoon's string stability. String stability of a platoon refers to the ability of a group of coordinating vehicles to travel with the desired velocity and maintain a close distance to achieve the vehicle's on-road performance and efficiency [21]. A platoon has string stability if disturbances are not amplified when propagating along the vehicle string [38]. A string-stable platoon ensures that the following vehicles decelerate in a controlled manner without overreacting. If the platoon is string unstable, the second vehicle might

brake harder than necessary, causing the third vehicle to brake even harder, potentially leading to a chain reaction of emergency braking or even collisions. The lower-level controller translates the desired acceleration into throttle and brake commands.

In the P1 controller, each vehicle receives information from the lead vehicle and the preceding vehicle in the platoon via the wireless network. This information includes the controller's desired acceleration (m/s^2), the vehicle's actual acceleration (m/s^2), speed (m/s), XY position (m), and the time at which the data has been measured (s). The controller equation that computes the desired acceleration of the i-th vehicle in a platoon is given by [4] given by

$$\begin{aligned} \ddot{x}_{i\_des} &= (1 - C_1)\ddot{x}_{i-1} + C_1\ddot{x}_l \\ &\quad - (2\xi - C_1(\xi + \sqrt{\xi^2 - 1}))\omega_n\dot{\epsilon}_i \\ &\quad - (\xi + \sqrt{\xi^2 - 1})\omega_n C_1(v_i - v_l) - \omega_n^2\epsilon_i \end{aligned} \tag{1.1}$$

C_1 is the weighting factor that takes on values between 0 and 1 where the default value is set to 0.5, ξ is the damping ratio and can be set to 1 for critical damping, and ω_n is the controller's bandwidth, where the default value is set to 0.2Hz [4]. $\ddot{x}$ denotes the longitudinal acceleration of the vehicle where $\ddot{x}_{i-1}$ represents the acceleration of the preceding vehicle and $\ddot{x}_l$ represents the acceleration of the lead vehicle. Similarly, v_i is the longitudinal velocity of the i_{th} vehicle, and v_l is the longitudinal velocity of the lead vehicle.

The first four terms of equation (1.1) consist of information received from other vehicles via V2V communication. The fifth term consists of two parameters, ω_n^2 and ϵ_i, where ω_n^2 is the control gain, and ϵ_i is the longitudinal spacing error of the ith vehicle, which is calculated using equation (1.2), where x_i is the position of the ith vehicle, x_{i-1} is the position of the preceding vehicle, and L is the desired spacing.

$$\epsilon_i = x_i - x_{i-1} + L \tag{1.2}$$

The ϵ_i solely relies on sensor information acquired by each vehicle's own radar and is therefore unaffected by any communication loss. Note that the P1 controller mainly uses the radar to maintain consistent spacing between vehicles to ensure string stability. It does not use the radar to achieve collision avoidance, as is done in ACC controllers. More details about setting the controller parameters, such as engine and driver parameters, can be found in the API section of the Plexe webpage [39].

In addition to the CACC controller described above, Plexe includes implementations of three other cooperative cruise controllers [39]. These controllers are known as 'Flatbed' [40], 'Ploeg' [41], and 'Consensus' [42].

2.2 Wireless Communication

In Section 2.2.1, we give a short overview of the IEEE standards that provide wireless access in vehicular environments (WAVE)while elaborating more on data transmission and reception at the physical layer, antenna characteristics, and wireless channel behavior in Section 2.2.2.

2.2.1 Wireless Access in Vehicular Environments

IEEE has developed a family of standards for Wireless Access in Vehicular Environments (WAVE) to enable vehicles to share information about their status and environment through dedicated short-range communications (DSRC) to improve safety, traffic management, and efficiency.

The key standards for WAVE consist of the following IEEE standards:

- IEEE 1609.1 defines the application layer, including message formatting and data exchange.
- IEEE 1609.2 defines the security protocols for V2V and V2X communications.
- IEEE 1609.3 covers networking services for WAVE, such as routing and message forwarding.
- IEEE 1609.4 defines the upper Media Access Control (MAC) layer.
- IEEE 802.11*p* defines the lower MAC and physical (PHY) layers.

The Veins simulation framework provides a MAC and PHY layer models for 1609.4 and 802.11*p*, respectively. We implement the jamming attacks by manipulating specific parameters in the simulation model of the IEEE 802.11p physical layer implemented in Veins. Specific to our experimentation setup, we use a message update frequency of 10 Hz. This means that all vehicles in the platoon broadcast exactly one message each during a time period of 0.1 seconds, where each message consists of of 200 bytes. Thus, a jamming attack that blocks all communication during 1 second would result in the loss of 40 messages.

2.2.2 V2V Physical Layer Communication

IEEE 802.11p Transceiver This section summarizes the transmitter and receiver blocks of a typical IEEE 802.11p system, see Fig. 1. IEEE 802.11p operates in the 5.9 GHz frequency band with a bandwidth of 10 MHz per channel and uses orthogonal frequency division multiplexing (OFDM) to transmit data efficiently. OFDM divides the 10 MHz channel into 64 subcarriers, of which 52 are utilized: 48 is allocated to the data subcarriers and 4 to pilot subcarriers. The pilot subcarriers play an essential role in wireless OFDM-based systems. They provide a reference for accurately estimating and compensating for frequency offset and phase noise in the received signal. The remaining 12 subcarriers act as guard bands to prevent interference with adjacent channels.

The forward error correction (FEC) scheme used for IEEE 802.11p transmission is convolutional code with industry-standard generator polynomials $g_o = 133$ and $g_1 = 171$ with supported code rates of 1/2, 2/3 and 3/4 [43]. FEC is a technique used in digital communication systems to improve the reliability of data transmission over noisy or unreliable channels. The idea behind FEC is that the transmitter adds redundant error-correcting codes to the original data

before sending it. These redundant bits allow the receiver to detect and correct errors that occur during transmission without the need for retransmission [44].

Code rates higher than 1/2 are achieved through puncturing [43], a process that selectively removes certain bits from the encoded data stream. By eliminating specific bits, puncturing effectively increases the code rate without changing the structure of the convolutional code [45]. The receiver can recover the original data by knowing which bits were punctured and applying error correction. Interleaving is a process that comes after puncturing and before modulation. It rearranges the bits across the transmission frame and aims to spread out consecutive bits over different sub-carriers and time slots. This helps mitigate burst errors that may affect groups of consecutive bits due to interference, fading, or other channel disturbances. The de-interleaving reorders the bits to their original positions at the receiver, reversing the interleaving pattern applied at the transmitter.

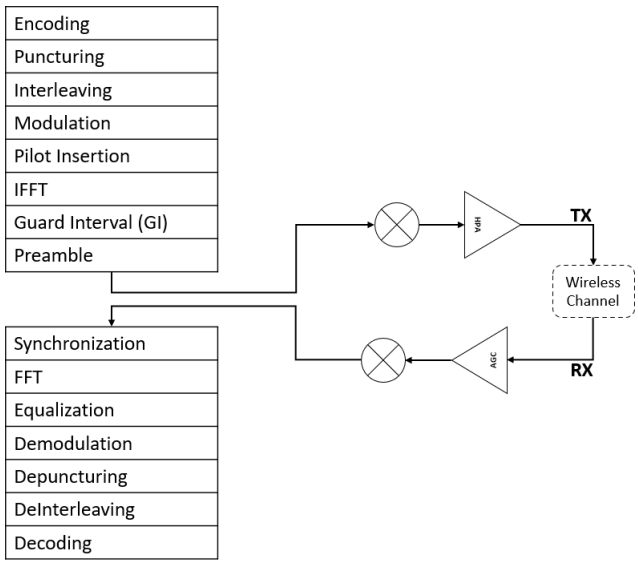

Figure 1: A simplified block diagram of IEEE 802.11p physical layer transceiver design [46], [47].

For transmission, several modulation schemes are used in the context of V2V communication [48]. These modulation schemes include binary phase shift keying (BPSK), quadrature phase shift keying (QPSK), and quadrature amplitude modulation such as 16-QAM and 64-QAM [43].

After modulation, known reference signals called pilots are added to specific subcarriers for channel estimation. An inverse fast fourier transform (IFFT) is then performed to convert the signal from the frequency domain to the time domain, enabling multi-carrier transmission. Guard interval is used to mitigate inter-symbol interference (ISI), where a portion of the time domain signal is copied from the end and appended to the beginning of the signal.

Before forming the OFDM symbol, a predefined sequence of time domain

signals, known as the preamble, is added for synchronization, initial channel estimation (e.g., fading, attenuation, and phase shifts), and receiver initialization to correctly interpret incoming signals. After completing these processing steps, the OFDM symbol is amplified using a high-power amplifier (HPA) before transmission (see Fig. 1).

The modulation schemes used in the Veins simulation framework are BPSK and QPSK, depending on the required data rate and the wireless channel conditions. BPSK is more resistant to noise and multipath fading, while QPSK offers a higher data rate than BPSK is less resilient to noise and interference. It has a coding rate of 1/2 and a data rate of 6 Mb/s with a channel spacing of 10 MHz [14].

In OFDM-based communication systems, the receiver evaluates whether the received signal has been correctly decoded and matches the originally transmitted data. There is an upper bound for the number of erroneous bits in a packet that a receiver can correct to preserve the integrity of the data. If the number of erroneous bits exceeds that threshold, the packet cannot be decoded correctly, preventing the successful recovery of the transmitted data.

BER is a metric that quantifies the ratio of bits received in error to the total number of bits transmitted. BER is calculated based on the signal interference and noise ratio (SINR) and the given modulation scheme. SINR is used to determine the quality of the signal received. It is the ratio between the power of the legitimate signal and the total power of noise and interference. Noise is the unwanted noise, that is, channel noise, parasitic noise, and interference is the transmission of the neighboring channels using the same frequencies see Eq. 1.3.

In the Veins simulations, there is a module at the physical layer called the ‘decider, which computes the packet's SINR and inputs it into a bit error model to calculate the BER. This model calculates the bit error rate based on the modulation scheme and SINR. The BER is then compared with a randomly generated number between 0.0 and 1.0. If the generated number exceeds the BER, the packet is considered error-free and passed to the next layer [49]. Veins utilizes BER and a random number generator to simulate the unpredictable nature of real-world errors.

$$SINR = \frac{SignalPower}{InterferencePower + NoisePower} \tag{1.3}$$

Wireless Channel A wireless channel is the medium through which wireless communication signals are transported from a transmitter to a receiver. It consists of three essential elements: the transmit antenna, the air medium, and the receiver antenna.

There are different types of *antennas* used for signal transmission, such as *monopole* and *directional* antennas [50] that are used to transmit and receive the data to or from the wireless channel. The monopole antenna transmits the electromagnetic waves equally in all directions, whereas in the case of a directional antenna, the concentration of electromagnetic waves is in one direction [50]. In the platooning application model, which is part of our

simulation environment, all vehicles are equipped with monopole antennas to transmit/receive signals [51]. If the received signal power is above the antenna's sensitivity threshold, the signal is sent to the physical layer for processing [14]. The default threshold value in Veins is -95 dBm.

A wireless channel occupies a given set of frequencies within a frequency band in the electromagnetic spectrum. The key factors that affect the signal in the wireless channel include *(i)* the distance *'d'* between transmitter and receiver, *(ii)* the sensitivity of the receiver antenna, *(iii)* the wavelength 'λ' of the transmitted signal, and *(iv)* reflections from objects in the environment such as road, buildings, and trees. The Veins simulator includes three predefined channel models [52] for calculating signal attenuation which is the reduction in signal strength as a signal travels through a wireless channel. These models are *Two-Ray Interference Model*, *Obstacle Shadowing*, and *Free Space Path Loss (FSPL)* models. The Two-Ray Interference Model considers two primary propagation paths from the transmitter to the receiver: direct line-of-sight (LOS) and ground-reflected path. In the obstacle shadowing environment model, the signal attenuation is caused by physical obstacles, such as buildings, trees, or vehicles, that partially or fully obstruct the line of sight between the transmitter and receiver.

Finally, the Free Space Path Loss (FSPL) model attenuates the signal as it travels in free space without any obstacles or reflections. FSPL is determined by the distance d between the transmitter and receiver and the signal's wavelength λ. This model is widely used in wireless communication to estimate signal loss under ideal line-of-sight conditions [49]. The mathematical representation of *FSPL* is called the *Friis transmission equation* shown in Eq. 1.4.

$$Pr[dBm] = Pt[dBm] + Gt[dB] + Gr[dB] - \sum Lx[dB] \tag{1.4}$$

In this equation, the received power Pr $[dBm]$ is calculated based on the transmitted power Pt $[dBm]$ that is delivered to the transmitting antenna; Gt $[dB]$ and Gr $[dB]$ are the transmitter and receiver antenna gains, respectively. The $\sum Lx$ $[dB]$ represents the total losses caused by the environment. In Eq. 3.2, the dB, or decibel, is a logarithmic unit used to express the ratio between two values in the context of power used to describe gain or loss, whereas the dBm is an absolute measurement that indicates actual power with reference to 1 milliwatt.

Our experiments use the FSPL model because our experimental scenario involves a platoon of four vehicles driving on a highway with no oncoming traffic or nearby buildings. Additionally, the distances between the vehicles are relatively short, making FSPL a suitable model for initially assessing the impact of jamming on vehicle safety. Testing with other propagation models such as *Two-Ray Interference Model*, *Obstacle Shadowing*, *Rayleigh* and *Rician Fading* models are also of interest and will be considered as part of our future work.

2.3 Jamming Techniques

Jamming of analog signals is carried out in the wireless channel. Lichtman et al. [11] categorize various types of communication jamming techniques based on their methods, signal characteristics, and intended effects.

In **barrage jamming** or **noise jamming** all subcarriers are jammed by transmitting additive white gaussian noise (AWGN) to degrade the received SINR. As barrage jamming targets a broad frequency range, it is suitable for attacking systems that use frequency-hopping and spread spectrum communications. **Spot jamming** focuses noise on a specific frequency, while **sweep jamming** rapidly shifts noise power to disrupt multiple channels over time.

Pilot jamming is a type of barrage jamming attack where pilot symbols are the main target of the attacker. These symbols are inserted at specific subcarriers or time slots and help the receiver to estimate the channel response and perform synchronization and equalization. In a pilot jamming attack, the noise power is evenly distributed between all pilot subcarriers. **Pulse jamming** is a type of barrage jamming attack that disrupts synchronization by sending high-energy pulses.

In **Reactive jamming** the attacker actively listens to a communication channel and transmits interference only when legitimate signals are detected. This technique allows the jammer to conserve energy and focus its attack on actual transmissions, making it harder to detect and mitigate than constant or proactive jamming. **Deceptive jamming** involves transmitting fake signals, such as in *replay attacks* or *spoofing*, to confuse receivers in differentiating the legitimate signal from the malicious signal.

In **nulling, cancellation** or **destructive interference** attacks, the jammer transmits a structured waveform designed to cancel out the target signal by creating destructive interference. This is achieved by transmitting a signal that is identical to the target signal in time and frequency but shifted in phase by 180 degrees. As a result, when the two signals overlap, they effectively cancel each other out, leaving only noise for the receiver. In **pilot nulling** attacks, pilot symbols are the main target of the attackers. In pilot nulling, the attacker seeks to cancel the pilot tone of the OFDM signal to bring the pilot tone power close to zero.

Jammer equipment has different capabilities that can be classified according to *time-correlation*, *protocol-awareness*, *learning*, and *spoofing* [11]. The term *time-correlation* indicates that the jamming signal sent is aligned with the legitimate or target signal in time. *Protocol awareness* describes the attacker's understanding of the legitimate signal's protocol. In the context of *machine learning*, learning refers to the process by which the attacker's system derives knowledge from data. Lastly, *spoofing* involves impersonating legitimate signals to obtain unauthorized access or benefit.

Among the five types of jamming attacks studied in this thesis, DoS attack and barrage jamming are *non-correlated* and *non-protocol aware* and can, therefore, be carried out without detailed knowledge of the communication protocol and targeted signal. However, destructive interference and deceptive jamming require *time-correlated* and *protocol-aware* capabilities.

2.4 Related Work

We adress four areas of related research in this section. Section 2.4.1 covers studies that investigate the impact of jamming attacks in wireless communication systems. Section 2.4.2 describe work focusing specifically on the impact of jamming attacks in platooning systems, while Section 2.4.3 covers work that aim to improve the resilience of CACC controllers to jamming. In Section 2.4.4, we briefly describe previous work on security benchmarking.

2.4.1 Jamming Techniques

Researchers in V2V communication systems have recognized the increasing challenges posed by cybersecurity threats and the critical importance of information security [2], [23], [53]. This section presents some of the studies most relevant to our work.

Moser et al. [25] studied the impact of signal cancellation attacks where the attacker's signal interferes destructively with the legitimate signal. They demonstrated that the signal cancellation attack could effectively attenuate the signals up to 40 dB. Moser et al. demonstrated through their experiments that cancellation or destructive interference attacks are feasible, a notion previously deemed impossible in past studies. They emphasized the importance of considering the possibility of signal cancellation attacks when assessing the security of advanced cooperative systems.

Clancy [27] studied the performance of OFDM transmission, *pilot jamming*, and *pilot nulling* attacks. According to the results obtained in this work, pilot jamming is roughly 2 dB more efficient than barrage jamming, and pilot nulling is approximately 7.5 dB more efficient than barrage jamming.

Mahal et al. [26] studied the impact of nulling attacks on cyclic prefixes in single-carrier frequency division multiple access (SC-FDMA) communication, which is employ for up-links in 4G and 5G mobile communication standards. Cyclic prefixes involve adding a copy of the end of a signal to the beginning of the signal to mitigate inter-symbol inference. (ISI).

Patounas et al. [54] studied the prevention, detection, and mitigation of DoS attacks on IEEE 802.11p-based communication of a vehicle platoon. They implemented and tested intrusion detection and handling mechanisms against barrage jamming and data falsification attacks.

2.4.2 Simulation-based Assessment of Cooperative Cruise Controllers

Alipour-Fanid et al. [21] investigated the impact of the attacker's location when performing a reactive jamming attack on cooperative driving. They used a high-level model of the IEEE 802.11p protocol to study the impact of jamming attacks on a cooperative cruise controller implemented in MATLAB. They showed that targeting the vehicle behind the lead vehicle is most effective for an attacker to destabilize the string stability of the platoon.

Alipour-Fanid et al.'s study is similar to our work on performing jamming attacks on the IEEE 802.11p communication protocol, where the same cooper-

ative cruise controller is evaluated. Their flexible jamming model represents a wide range of jamming signal scenarios implemented in MATLAB.

We model and implement detailed *barrage* and *destructive interference* attacks at the physical layer of the communication system modeled in Veins. Veins provides high-fidelity wireless communication models. Moreover, the attacker model implemented by Alipour-Fanid et al. is based on the additive Gaussian random noise (AWGN).

van-der Heijden et al. [3] proposed a novel attacker model and use it to evaluate the resilience and effectiveness of three cooperative cruise controllers provided in Plexe. One of these controllers is the same as the one evaluated in our work. Their results show that this CACC controller is highly sensitive to jamming attacks. Their work resembles ours in conducting the simulations using the Plexe framework. However, while they model the impact of jamming attacks as lost messages at the application level, we simulate the attacks at the physical layer.

Another aspect of our simulations is the granularity. The granularity of the attack parameter values for our test campaigns is relatively high, i.e., our attack model parameter's step size was significantly smaller than those used in comparable studies [3], [21]. In addition, we classified the experimental results using the deceleration profiles and collision incidents, while the other studies used the speed profile to classify the severity of the outcome. As part of future work, we plan on extending our classification scheme to allow direct comparisons with these results and other future studies.

2.4.3 Improving Jamming Resilience of CACC controllers

The control algorithms of cooperative vehicles must be built resilient to jamming attacks to ensure the safety, operational continuity, security, and regulatory compliance of autonomous vehicle systems.

In a recent paper, Segata et al. [22] argue that no single communication technology can achieve the level of reliability that is required for advanced cooperative driving applications. Hence, they propose a fallback and recovery mechanism based on the assumption that future vehicles will be equipped with multiple communication interfaces, such as IEEE 802.11p, Visible Light Communication (VLC), and LTE-based Cellular V2X (C-V2X). This mechanism ensures that vehicles can safely transition to autonomous or manual driving. The authors show that the proposed fallback and recovery mechanism are feasible. However, designing such a system requires careful consideration, as poor design choices can lead to instability or even collisions.

Rens van-der et al. [3] developed an evaluation framework for assessing the resilience of Plexe-implemented cooperative cruise controllers against jamming attacks. Based on their experimental findings, the authors suggested a graceful degradation from a cooperative cruise controller to an adaptive cruise control as a potential mitigation strategy.

Shahriar et al. [55] also proposed a synchronized braking mechanism in the cooperative cruise controller implemented in the Plexe simulation. This mechanism is a type of emergency braking and acts as a fail-safe mechanism

to avoid rear-end collisions. They did not test their safety mechanism against jamming attacks. However, their focus is to avoid rear-end collisions that could occur in case of braking due to the cooperative cruise controller's small inter-vehicle distances.

2.4.4 Security Benchmarking

Researchers have proposed various security benchmarking frameworks across different cybersecurity domains. Oliveira et al. [20] introduced a two-phase benchmarking framework specifically for web service frameworks (WSFs), emphasizing security qualification and trustworthiness assessment.

Similarly, Anisetti et al. [56] developed a security benchmark to evaluate the security assurance of OpenStack, an open-source cloud infrastructure. Additionally, Braun et al. presented NETCARBENCH [57], a benchmark designed to assess and compare the techniques and tools used to develop in-vehicle communication networks. To the best of our knowledge, no prior research has focused on establishing security benchmarks for the simulation-based assessment of Cooperative Driving Automation (CDA) applications.

3 Summary of Appended Papers

In this chapter, we summarize all the publications included in this thesis. Figure 2 provides an overview of these publications.

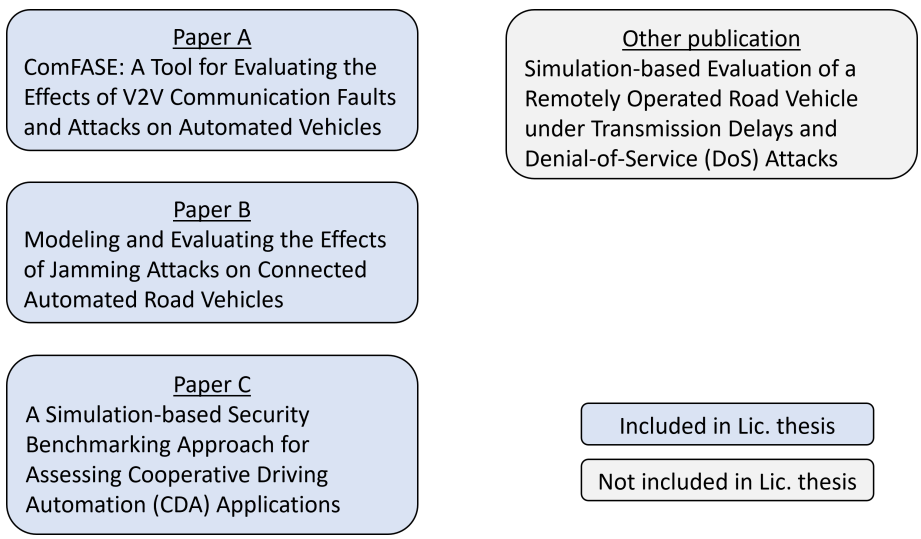

Figure 2: Overview of the research publications.

3.1 ComFASE: A Tool for Evaluating the Effects of V2V Communication Faults and Attacks on Automated Vehicles

In this paper, we introduce ComFASE, a versatile fault and attack simulation engine for studying consequences and safety implications of communication failures in interconnected automated vehicular systems. The tool is flexible in modelling different types of faults and attacks that may compromise the reliability of wireless messages. It enables detailed simulations to assess the safety implications of cybersecurity attacks and communication faults in realistic traffic scenarios. To this end, ComFASE utilizes four existing simulation environments: Plexe (for simulation of platooning systems), Veins (a vehicular network simulator), SUMO (a traffic simulator), and OMNeT++ (a networks simulator).

The tool provides support for automatically running long series of fault or attack injection experiments, commonly known as fault injection or attack injection campaigns. The conduct of a campaign is divided into three phases: configuration, execution, and result classification.

During configuration, the user defines a traffic scenario, sets various parameters in the communication model, and provides a campaign vector. The traffic scenario can be tuned with respect to various system parameters dealing with road conditions, vehicle features, system size, scenario maneuvers, and simulation time. ComFASE currently supports simulation of the physical (PHY) and media access (MAC) layers of the IEEE 802.11p standard for wireless access in vehicular environments (WAVE). The user can configure the communication model by selecting one of three wireless channel models, the packet size, and the beaconing period.

The campaign vector includes information about the selected attack model, the vehicles to be subjected to attacks, and the attack model parameters. The latter includes the attack start time, attack duration and attack value. The attack's start time and duration are defined in relation to the time axis of the traffic scenario, whereas the attack value depends on the attack model. Another important part of the configuration phase is the execution of the golden run, which generates a profile of the system's behavior under fault-free circumstances. The data collected during the golden run is later used for classification of the outcomes of the attack simulations.

In the execution phase, ComFase runs the attack campaign defined by the configuration data automatically without human intervention. During the simulations, data is collected from SUMO about the movements of the vehicles in the investigated system, including velocities, accelerations, decelerations and collision incidents. In the result classification phase, automated analyses are performed to classify the outcomes of the attack simulations according to their severity. These analyses are performed by comparing the behavior of the target system during the simulated attacks with its behavior during the golden run.

To demonstrate the tool, we present results from a series of simulation experiments, where we injected delay and denial-of-service attacks on wireless messages exchanged between vehicles in a platooning application. The results

show how different variants of attacks and attack model parameters influence the platooning system regarding collision incidents versus benign, negligible and non-effective outcomes of the attacks.

Statement of Contribution

This work is a collaborative effort between my colleague, Mehdi Maleki, and me, with valuable input from my supervisors, Behrooz Sangchoolie and Johan Karlsson. My colleague and I developed the fault and attack injection tool, ComFASE, configured the test campaigns and performed the analysis. Additionally, I took the lead role in conceptualizing the project and writing the paper.

3.2 Modeling and Evaluating the Effects of Jamming Attacks on Connected Automated Road Vehicles

In this paper, we propose and utilize simulation models to examine the impact of three types of jamming attacks: *destructive interference*, *barrage jamming*, and *deceptive jamming*. The primary objective of this study is to evaluate the impact of these attacks on vehicle safety by analyzing collision incidents and deceleration profiles of the vehicles caused by the communication loss.

Our finding reveals that jamming attacks pose significant risks to the stability and safety of platooning systems equipped with CACC controllers, which rely solely on communication and don't have fallback mechanisms to handle communication failures.

We conducted three attack injection test campaigns to evaluate the impact of destructive interference attacks. In the first test campaign, vehicle 2 was targeted; in the second campaign, vehicle 4 was targeted; and in the last one, all vehicles were targeted. When all vehicles were targeted, 27.5% of all experiments resulted in collisions. When vehicle 4 was targeted, 26% resulted in collisions, and when vehicle 2 was targeted, 7% resulted in collisions.

We observed that vehicle 4 was significantly more vulnerable to destructive interference attacks than vehicle 2. This high vulnerability is primarily due to the distance between the target and leader vehicles. In this study, we used the free space path loss (FSPL) environment model, where signal attenuation strongly depends on the distance between the transmitter and receiver. Being the farthest vehicle from vehicle 1, vehicle 4 experiences the highest level of signal attenuation, making it particularly susceptible to the impact of the injected attacks.

We also conducted barrage jamming attacks on all vehicles in the platoon, where 48% of the experiments resulted in collisions. To better understand the impact of the barrage jamming attacks, we identified the vehicles responsible for these collisions. Our analysis revealed that vehicles 2, 3, and 4 accounted for 41%, 43%, and 16% of the collisions, respectively. This outcome highlights how the barrage jamming attacks can disrupt the coordination and safety of the platoon. We also injected deceptive jamming attacks where 47% of the total experiments resulted in collisions.

These results, where many experiments resulted in collisions, emphasize the need for robust error-handling mechanisms in CACC controllers to mitigate the risks of jamming attacks. Our findings suggest that the current implementation of the CACC model we tested lacks sufficient resilience to message loss caused by jamming attacks. By demonstrating the impact of these attacks in a controlled simulation environment, the study underscores the importance of evaluating platooning applications under jamming attacks. This work provides valuable insights for designing and developing more secure and resilient communication protocols and control algorithms for connected automated vehicles.

Our future research will focus on developing techniques to improve the resilience of cooperative cruise controllers used in platooning and other autonomous driving applications against jamming attacks.

Statement of Contribution

This paper is a collaborative effort with my co-author Mehdi Maleki. Mehdi Maleki served as the first author and contributed to the development of the tool. I led the project in terms of conceptualizing the idea, conducting the literature review, and particularly focusing on attack modeling. Additionally, I took the lead role in writing the paper. I received valuable feedback from my supervisors, Behrooz Sangchoolie and Johan Karlsson, which helped to enhance the quality of our work for publication. I'm also proud to mention that this paper received the second-best paper award from the PRDC program.

3.3 A Simulation-based Security Benchmarking Approach for Assessing Cooperative Driving Automation (CDA) Applications

The work presented in this paper is intended as an initial contribution towards a definition of *security benchmarks* for simulation-based assessment of CDA applications concerning their ability to operate safely in the presence of jamming attacks. In general, the primary motivation for defining benchmarks for computer-based systems is to provide a widely accepted and easy-to-use procedure for evaluating or comparing system implementations, components, or design solutions. Regarding basic concepts and main objectives, security benchmarking is akin to the closely related field of dependability benchmarking.

Since security benchmarking is a novel topic in the context of CDA applications, we would like to emphasize that our benchmarking framework is intended as a tentative example of how security benchmarks for assessing the resilience of a CDA application against jamming attacks could be defined. This is not intended as a final solution but as a starting point for a wider effort to develop security benchmarks for CDA applications, including benchmarks for attacks other than jamming attacks.

The core components of our proposed security benchmark are the driving scenario and the attack model. To illustrate the role these components would play in future definitions of jamming resilience benchmarks, we utilized two driving scenarios, braking and sinusoidal, as stimuli to evaluate the robustness

of a platooning application. In addition, we injected barrage jamming attacks into the vehicle communication system based on the IEEE 802.11p protocol. Other system components influencing the evaluation, such as the wireless communication model, wireless channel model, and the number of vehicles, are kept constant throughout the testing and evaluation process.

We demonstrate that barrage jamming attacks can compromise safety, leading to emergency braking and collisions among platooning vehicles. Our findings also indicate that the severity of barrage jamming attacks varies depending on the driving scenario, with the most severe impacts, such as collisions, occurring when the attack is started during vehicle acceleration. This outcome is strongly connected to the design of the CACC controller model and explains why the platooning system is more vulnerable to attacks during an acceleration period.

When utilizing the specific CACC controller [22], the lead vehicle periodically sends acceleration and deceleration commands to the platoon's following vehicles. In case of communication loss, the following vehicles continue accelerating, decelerating, or keeping a constant speed according to the last received command. If a jamming attack begins to block the communication channel during an acceleration period, the affected vehicles will continue to accelerate and cause collision when the lead vehicle decelerates.

The *attack start-time* is not the only attack parameter that influences the likelihood of a collision. The *attack duration* and *attack value* are other attack parameters that influence the outcome. The longer attacks are generally more likely to cause a collision. However, attack durations longer than a certain threshold do not significantly increase the number of severe outcomes. The higher attack values contribute to greater signal distortion, which can eventually cause communication loss. This loss significantly contributes to collisions when vehicles accelerate. We observe fewer collisions for attacks initiated when the vehicles are braking because the communication loss happens already when the vehicles have started to reduce their speed.

Statement of Contribution

This is collaborative work with my supervisor, Behrooz Sangchoolie and Johan Karlsson. I was responsible for developing the concept of the security benchmarking framework and connecting it to my research topic, conducting test campaigns, performing the literature review, and analyzing the outcome of the experiments. Additionally, I took the lead in writing the paper.