UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

GUILHERME HENRIQUE MENGE DE AMANDO

AN ENVIRONMENT FOR TESTING V2X COMMUNICATION SCENARIOS

RECIFE
22 de setembro de 2023

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

GUILHERME HENRIQUE MENGE DE AMANDO

AN ENVIRONMENT FOR TESTING V2X COMMUNICATION SCENARIOS

Trabalho apresentado ao Programa de Graduação em Sistemas de Informação do CENTRO DE INFORMÁTICA da UNIVERSIDADE FEDERAL DE PERNAMBUCO como requisito para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: *Prof. Abel Guilhermino da Silva Filho*

RECIFE
22 de setembro de 2023

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Amando, Guilherme Henrique Menge de.
An environment for testing V2X communication scenarios / Guilherme Henrique Menge de Amando. - Recife, 2023.
36p : il., tab.

Orientador(a): Abel Guilhermino da Silva Filho
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Sistemas de Informação - Bacharelado, 2023.

1. automotives. 2. intervehicular networks. 3. v2x. 4. autonomous vehicles. 5. cybersecurity. I. Silva Filho, Abel Guilhermino da. (Orientação). II. Título.

000 CDD (22.ed.)

GUILHERME HENRIQUE MENGE DE AMANDO

AN ENVIRONMENT FOR TESTING V2X COMMUNICATION SCENARIOS

Trabalho apresentado ao Programa de Graduação em Sistemas de Informação do CENTRO DE INFORMÁTICA da UNIVERSIDADE FEDERAL DE PERNAMBUCO como requisito para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado em: 22 / 09 / 2023.

BANCA EXAMINADORA

_____

Prof. Abel Guilhermino da Silva Filho

UNIVERSIDADE FEDERAL DE PERNAMBUCO

_____

Prof. Divanilson Rodrigo de Sousa Campelo

UNIVERSIDADE FEDERAL DE PERNAMBUCO

Resumo

A maioria dos carros tem entre 10 e 15 ECUs que controlam funções, incluindo a motorização e câmbio do carro. Carros de luxo chegam 70 ECUs para apoiar as tecnologias mais recentes, como os carros autônomos. Os automóveis estão cada vez mais conectados por meio de diversas tecnologias, resultando em risco de ataques através de diferentes vetores. O invasor pode migrar da comunicação V2X, back-end ITS, aplicativos de terceiros e outras tecnologias sem fio. Os pesquisadores sugerem que os invasores provavelmente vão ter sucesso se tiverem acesso à rede de um veículo. Nossa pesquisa propõe um ambiente de testes para simular a comunicação V2X, além de um estudo de caso sobre possíveis ataques cibernéticos e eventuais contramedidas associadas.

Palavras-chave: Automotives, Intervehicular communication, V2X, Cybersecurity, Platoon, Autonomous vehicles, Self-driving cars, OBU, RSU

Abstract

Most cars have 10-15 ECUs that control functions, including powertrain. Luxury cars have up to 70 ECUs to support the most recent technologies, such as self-driving (autonomous) cars. Autos are increasingly connected through various technologies, resulting in risks of attacking through different vectors. The attacker may pivot from V2X communication, ITS back-end, third-party apps, and other wireless technologies. Researchers suggest attackers are likely to succeed when they have access to a vehicle's network. This research proposes a testing environment to simulate V2X communication (OBUs and RSUs) and a case study of possible cyberattacks and associated countermeasures (mitigations).

Keywords: Automotives, Intervehicular communication, V2X, Cybersecurity, Platoon, Autonomous vehicles, Self-driving cars, OBU, RSU

Contents

List of Figures

List of Tables

1 Introduction

The modern automotive industry is currently producing highly complex computers on wheels. As a result, sensors and communication devices are growing in number. Increasing the amount of those types of devices in our cars increases the risks associated with cyberattacks. Modern vehicles have between 10 and 70 electronic control units (ECUs), depending on the price of the vehicle. These devices are responsible for controlling many functions of vehicles, including the powertrain. Cars are also increasingly connected through a wide variety of technologies. In certain cases, the communication of vehicle systems occurs via message-oriented transmission protocols that lack encryption or authentication, as specified in (MUCH, 2016). Attempts have been made to implement security features to mitigate risks.

V2X communication, or Vehicle to Everything communication, is the technology that allows vehicles to communicate with each other, with infrastructure, and with pedestrians. Shared information includes speed, location, and heading. Figure 1 illustrates the dynamics of V2X connectivity.

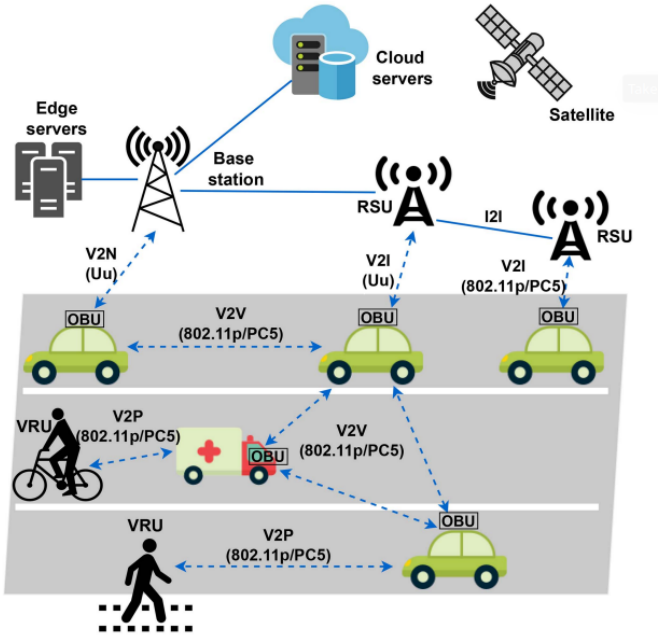

Figure 1: V2X connectivity example. Source: (SEDAR et al., 2023)

Warnings about traffic congestion, accidents, and pedestrians crossing the road are also part of the information exchanged between the entities. V2X communication has the potential to improve road safety, reduce traffic congestion, and make driving more efficient. There is also V2V communication, or Vehicle to Vehicle communication, which is a subset of V2X communication that allows vehicles to communicate directly with each other. This direct communication shares information more quickly and accurately than through a centralized infrastructure. V2V communication is a crucial technology for the development of autonomous

vehicles and its systems.

Vehicular Ad hoc NETworks (VANET) is not restricted to V2V communications. A vehicle within the VANET network will be communicating with a wide range of different nearby nodes such as pedestrians, cameras, and other adjacent infrastructure usually known as Roadside Unit (RSU). (LEE; ATKISON, 2021)

As detailed in a recent survey focused on testing V2X communication (WANG, J. et al., 2019), those technologies are getting popular while still in an exploratory stage. The problems of traffic safety and information security brought about by V2X applications needs attention. Before marketization, the reliability and maturity of the technology are essential and must be rigorously tested and verified. Therefore, testing is an integral part of V2X technology.

Dynamic network topologies, such as the scenario observed in V2X communication, result in discontinuous transmissions. One of the most significant challenges in the area is dealing with this situation. There are components known as roadside units (RSU) designed to play a critical role in communication, providing continuous transmission coverage and permanent connectivity. However, deploying roadside units have significant challenges, especially when security risks are at stake. Balancing performance and cost, respecting some target Quality of Service (QoS) such as energy consumption, service coverage, throughput, and low latency is quite challenging.

Those new technologies introduce new security risks. Vehicles are increasingly connected to the internet, which makes them even more vulnerable to cyberattacks. Attackers could use V2X communication to disrupt traffic, cause accidents, or steal personal information. Recent work has shown that once an attacker has access to the vehicle's internal network, it is not difficult to gain control of it (MILLER; VALASEK, 2015). Pivoting between protocols is a significant concern. Considering a scenario where an opponent already has access to a Wi-Fi network generated by the vehicle, pivoting from the Ethernet network to critical infrastructure (including other protocols) should also be a significant concern.

The number of automotive cybersecurity incidents has been increasing over the years. Upstream Security's 2020 Automotive Cybersecurity Report (UPSTREAM, 2022) shows a 99 percent increase in automotive cyber incidents between 2019 and 2020. Threats to vehicles regarding their communication channels are responsible for 89,3 percent of the incidents, vehicle code is associated with 87,7 percent of the cases. More than half of the incidents had vulnerabilities that could be exploited, and almost all of them were vulnerable to attacks via external connections.

Recent incidents involving connected and automated vehicles (CAVs) are good examples of the abundance of interfaces and the lack of attention to security and associated risks. Tesla's entire fleet takeover without direct connection with the vehicles is an excellent example of a security flaw (HUGHES, 2017). Many automotive manufacturers and technology companies have established bug bounty programs to encourage researchers to find and report vulnerabilities. Tesla, for example, has awarded large sums of money to researchers who have responsibly disclosed vulnerabilities in their vehicles.

Identifying reasonable methods to simulate V2X communication scenarios is the first step in studying the dynamics of vehicle interaction and the associated security risks. Developing solutions to improve the monitoring of the devices used during the tests of eventual attack

techniques is essential to comprehend the information exchanged between them. The creation of defense mechanisms, even if in a testing environment, will be explored further in this work.

The number of new components connected via IP/Ethernet protocol in current vehicles is larger every year. Higher data rates are necessary for developing new functions associated with V2V communication, augmented reality, self-driving cars, and audio-video technology. (MUCH, 2016) There are challenges and opportunities associated with the growth of connected devices in modern automotives:

Challenges

1. Security: unauthorized access to vehicle systems or data (which could cause accidents and many other problems).
2. Privacy: steal data about the driver and passengers, which could be used for marketing or other malicious purposes without their consent.
3. Malfunction: the devices can malfunction, which could also lead to safety issues.

Opportunities

1. Improved safety: Connected devices can be used to improve safety features, such as collision avoidance and lane departure warning.
2. Increased convenience: Connected devices can be used to provide features such as navigation, entertainment, and remote vehicle control.
3. Reduced costs: Connected devices can be used to reduce costs by improving efficiency and reducing the need for human intervention.

Which attack techniques associated with V2V communication are most likely to succeed? Which platform could be used to create a testbed for such scenarios? How could a researcher simulate attacks on those networks? How could we improve security by implementing software protection? In order to answer those questions, our work focuses on creating an enronment to facilitate testing attack scenarios, such as:

1. DoS (Denial of Service) attacks: Attackers could flood the network, preventing legitimate messages from being properly delivered.
2. Spoofing: Attackers could spoof the identity of a device, allowing them to send false messages or intercept legitimate ones.
3. Man-in-the-middle (MITM) attacks and data theft: Modern vehicles might handle sensitive information, such as location, and vehicle identification numbers, which the attackers could steal.

Preventing attackers from succeeding in using these techniques is essential to guarantee the security of the users. For example, in a Denial of Service attack (DoS) scenario, a threshold could be implemented on the communication protocol to avoid flooding attacks. In this case, blocking reception if the number of received messages exceeds a limit in a certain period would be a possible mitigation. Other threat models and attack scenarios should be considered.

In this rapidly evolving landscape of modern transportation, V2X (Vehicle-to-Everything) communication is a pivotal technological advancement with the potential to revolutionize road safety, traffic management, and the overall driving experience. As the deployment of V2X systems gains momentum, the criticality of ensuring their reliability, robustness, and compatibility becomes evident. The creation of assertive testing environments emerges as an imperative solution.

The complexity of V2X communication scenarios, involving intricate interactions between vehicles, infrastructure, and advanced communication protocols, necessitates comprehensive testing that mirrors real-world conditions. Assertive testing environments go beyond rudimentary validation, striving to emulate the intricate dynamics of actual vehicular networks. By meticulously recreating various driving scenarios, communication behaviors, and potential challenges, these environments enable researchers, engineers, and policymakers to subject V2X systems to rigorous scrutiny.

Establishing assertive testing environments empowers the industry to identify vulnerabilities, fine-tune communication protocols and validate safety mechanisms while reducing researchers' costs. This proactive approach not only enhances the performance and security of V2X systems but also instills confidence in their real-world implementation. Ultimately, the creation of such environments is not just a technical requirement but a strategic investment in the safety, efficiency, and innovation of our future roadways.

Our work goes through the most recent research regarding automotive V2X communication technologies. Then, we discuss the implementation of an environment to help test and debug automotive communication scenarios. Finally, we hypothesize that it is possible to build a low-cost and efficient architecture to simulate possible attack scenarios and its countermeasures.

2 Technical background

Studying modern vehicles' communication infrastructure and the associated technology was essential during the first' weeks of research. This knowledge was necessary to comprehend attacking tactics and techniques related to automotive networks. Attackers could work with a wide range of vectors, from physical interventions in the vehicle or nearby areas (such as image spoofing with objects) to server-side vulnerability exploitation (very far from the vehicle). (MUCH, 2016)

The ECUs (Electronic Control Units) are the devices responsible for controlling the vehicle's functions, sometimes accountable for simple sensors, other times dealing with complex critical calculations. The automotive industry currently relies on various technologies to connect its cars. There is a huge growth in the number of parts integrated by different network protocols (and associated technologies). The non-availability of space and problems with fault detection impacted the development of a distributed system. Components connect to different types of buses. (HVANTH; VALLI; GANESAN, 2012)

When analyzing the integration of various buses in vehicular networks, it is crucial to consider data compatibility, bus protocols, synchronization, and performance requirements. This process involves creating and implementing suitable gateway modules, ensuring data integrity, translating protocols, and efficient routing.

Optimizing the functionality of systems, streamlining data exchange, and enhancing communication between subsystems are the main goals of developing integrations. They promote interoperability, allowing different components from various manufacturers to work together smoothly. This integration also makes future scalability possible and enables the integration of new technologies and functionalities in vehicles. In our work, we will further discuss the consequences of these features on the vehicle's security.

Integrating different buses in vehicular networks is crucial in achieving efficient and reliable communication, coordination, and control within a vehicle. It is an important aspect of designing and building modern automotive systems capable of supporting advanced features, connectivity, and safety requirements. (SEDAR et al., 2023)

2.1 Vehicular connectivity protocols

Various communication technologies and standards make up vehicular connectivity protocols, enabling vehicles to connect with infrastructure and allowing internal components to communicate. These protocols facilitate the exchange of information, data, and commands, improving driving safety, efficiency, and enjoyment. They encompass networks within a vehicle and those between vehicles. (KARAGIANNIS et al., 2011)

2.1.1 Intra-vehicular Networks

This kind of vehicular network refers to the communication systems within a vehicle, enabling interaction among the various components, sensors, control units, and subsystems. These networks are vital in managing and coordinating the vehicle's internal operations. Some of the commonly used intravehicular protocols include Controller Area Network (CAN), Local

Interconnect Network (LIN), MOST, FlexRay, Ethernet, and IEEE 802.11p (WAVE).

The evolution of network protocols used inside automobiles has basically progressed from LIN (Local Interconnect Network) to CAN (Controller Area Network) and further to Ethernet. Each protocol represents a significant advancement in terms of data rate, complexity, and functionality.

LIN a legacy protocol, which operates at data rates of up to 20 Kbps and is primarily used for communication with non-safety-critical devices, such as interior lighting, window control, and seat adjustment. LIN is characterized by its simplicity, cost-effectiveness, and ease of implementation, making it suitable for basic communication in entry-level vehicles.

As well as LIN, CAN emerged in the mid-1980s and quickly became the dominant protocol for in-vehicle communication due to its robustness and reliability. It operates at data rates of up to 1 Mbps and supports both high-speed (CAN-HS) and low-speed (CAN-LS) variants. CAN is used for real-time, critical communication between electronic control units (ECUs) in safety-critical systems like engine control, braking, and transmission.The protocol's message-based structure allows for prioritization of messages and collision detection, ensuring efficient data exchange.

Ethernet, widely used in computer networks, found its way into automotive systems to meet the increasing demand for high-speed data communication and integration of advanced features. Automotive Ethernet encompasses multiple variants, including 100BASE-T1 (100 Mbps) and 1000BASE-T1 (1 Gbps), operating over a single twisted-pair cable. It offers significantly higher data rates, up to 2.5 Gbps, enabling efficient communication for advanced driver-assistance systems (ADAS), infotainment, and telematics. (MORTAZAVI; SCHLEICHER; GERFERS, 2018)

Automotive Ethernet also allows for the convergence of multiple communication protocols onto a single network, reducing wiring complexity and weight. Its ability to support high bandwidth and facilitate data-intensive applications makes it crucial for autonomous driving and connected car functionalities. The evolution from LIN to CAN and Ethernet reflects the automotive industry's need for increasingly sophisticated communication protocols to accommodate the growing complexity of in-vehicle systems and functionalities. Each protocol's adoption has been driven by the desire for improved performance, reliability, and future scalability, paving the way for enhanced safety, efficiency, and user experience in modern vehicles. This technology is meant to be used in vehicles and supports a range of applications, including advanced driver-assistance systems (ADAS), infotainment, telematics, and other connected car features. It has high data rate communication capabilities, including speeds up to 10 Gbps. (MORTAZAVI; SCHLEICHER; GERFERS, 2018)

100BASE-T1 (IEEE 802.3bw) operates at 100 Mbps and is used in automotive applications that require lower bandwidth than 1000BASE-T1, 100BASE-TX, and 1000BASE-T. 1000BASE-T is the standard Gigabit Ethernet protocol used in various industries, providing 1 Gbps data rate over four twisted-pair cables. 100BASE-T1 (IEEE 802.3bw) operates at 100 Mbps and is used in automotive applications that require lower bandwidth than 1000BASE-T1, 100BASE-TX, and 1000BASE-T. 1000BASE-T is the standard Gigabit Ethernet protocol used in various industries, providing 1 Gbps data rate over four twisted-pair cables.

This technology provides sufficient bandwidth to handle data-intensive applications, ensur-

ing smooth, real-time communication among various automotive systems. It also has a slight wiring complexity, and support for scalability, allowing for integrating future technologies and features in automotive systems. Fast and reliable communication is crucial for safety-critical applications like ADAS, and 1000BASE-T1 meets these requirements.

The following table presents a comparison of various intravehicular network protocols used in vehicles. Understanding the features and applications of these protocols is crucial for designing dependable and efficient automotive systems. The table offers significant details on frequently used intravehicular network protocols, aiding in decision-making regarding their implementation.

Name vs Specif.	**LIN**	**CAN**	**MOST**	**FlexRay**	**Ethernet (automotive) *****
Used On	Low-Level Subnets	Soft Real-Time	Audio and video applications	Hard Real-Time	Generic
Example of use	Door locking control	Engine Control; Driving assistance	Multimedia protocol	Emergency Systems	Infotainment; Parking cameras
Architecture	Single-Master; Master-Slave	Multi-Master	Master-Slave	Multi-Master	Flexible
Transfer mode	Sync.	Async.	Both	Both	Async.
Data rate	20 kBit/s	1 MBit/s	25, 50 and 150MBit/s	10 MBit/s	10 MBit/s - 2.5 GBit/s

Table 1: Comparing Automotive Intravehicular Network Protocols

2.1.2 Inter-vehicular Networks

Inter-vehicular networks enable communication between vehicles (V2V) and between vehicles and infrastructure (V2I). These networks enable the exchange of safety-critical information, traffic data, and cooperative applications. (MALIK; SAHU, 2018) Currently, there are two main types of communication technologies used:

1. Dedicated Short Range Communication (DSRC): A dedicated wireless communication protocol based on IEEE 802.11p, specifically designed for V2V and V2I communication. It operates in the 5.9 GHz band and enables direct and broadcast-based communication for safety and non-safety applications.

2. Cellular Vehicle-to-Everything (C-V2X): Utilizes cellular networks (4G LTE and 5G) to enable V2X communication. It provides both direct communication (PC5) and network-based communication (Uu). C-V2X offers enhanced capabilities for safety, traffic efficiency, and infotainment applications. (WANG, J. et al., 2019)

Vehicular connectivity protocols are the backbone of intelligent transportation systems. They facilitate communication between vehicles and surrounding infrastructure, which is essential in realizing the vision of safer, more efficient, and connected vehicles on the roads. (WANG, J. et al., 2019) The main aspects of each protocol are compared in the table below.

Name vs Specif.	**DSRC**	**CV2X**
Used On	V2X Communication	V2X Communication
Example of use	Direction control and ACC	Internet
Architecture	Decentralized (Adhoc)	Flexible
Transfer mode	Both	Both
Data rate	27 MBit/s	up to 20 GBit/s

Table 2: Comparing Automotive Inter-vehicular Network Protocols

Those connectivity protocols are crucial for enhancing road safety, improving traffic efficiency, and enhancing the overall driving experience. Some of the key objectives of using such technology are:

1. Safety: Vehicular connectivity protocols enable the exchange of real-time safety-critical information, such as collision warnings, emergency braking, and traffic signal status. This helps prevent accidents, reduce fatalities, and improve overall road safety.

2. Traffic Efficiency: By sharing traffic and road condition information, vehicular connectivity protocols facilitate intelligent traffic management, congestion mitigation, and optimized routing, leading to smoother traffic flow and reduced travel times.

3. Cooperative Applications: Vehicular connectivity protocols enable the implementation of cooperative applications, such as platooning (grouping of vehicles for improved fuel efficiency), intersection management, and cooperative adaptive cruise control. These applications enhance traffic efficiency, reduce emissions, and improve fuel economy.

4. Infotainment and Comfort: By providing connectivity between vehicles and external infrastructure, protocols enable a wide range of infotainment services, including live traffic updates, streaming media, remote diagnostics, and remote vehicle control. This enhances the driving experience and passenger comfort.

2.2 Security standards in the Automotive Industry

In the interconnected and technologically advanced landscape of the automotive industry, the paramount importance of security standards has never been more pronounced. As vehicles evolve into intricate networks of software, sensors, and communication interfaces, the need to safeguard these systems from malicious intrusions and vulnerabilities becomes a critical imperative. This is where security standards step in as the guiding compass, charting the course for ensuring the integrity, privacy, and resilience of automotive technologies.

The convergence of cutting-edge technologies such as connectivity, autonomous driving, and smart infotainment systems has brought both unprecedented opportunities and potential risks. The introduction of security standards serves as a cohesive framework, orchestrating a harmonized response to the multifaceted challenges that arise. These standards delineate the requisite protocols, practices, and measures that must be adhered to at every phase of a vehicle's lifecycle – from design and manufacturing to operation and maintenance.

In this era of connected vehicles and data-driven mobility, security standards provide not only a shield against cyber threats but also a beacon of assurance for manufacturers, regulators, and consumers alike. They stimulate innovation within a secure framework, engendering a culture of cybersecurity consciousness across the automotive ecosystem. This section is a journey through the landscape of security standards applied in the automotive industry, unraveling their significance, evolution, and implications for shaping the future of safer and more resilient vehicular technologies.

The International Organization for Standardization (ISO) is an international nongovernmental organization comprising national standards bodies focused on developing and publishing a wide range of proprietary, industrial, and commercial standards. ISO 26262-1:2018 addresses the functional safety of electrical and electronic systems within road vehicles. It focuses on developing safety-related systems and aims to minimize potential hazards caused by malfunctioning electronics, reducing the risk of accidents and injuries. This standard is crucial for the automotive industry as it provides a systematic approach to managing safety throughout the development lifecycle of automotive systems.

2.2.1 ISO 26262-1:2018: audience and stakeholders

This standard impacts various stakeholders within the automotive industry who are involved in developing, producing, and supplying electrical and electronic systems in road vehicles. The main goal of the standard is to ensure that these systems are functionally safe and minimize the likelihood of any potential risks arising from their malfunction.

The audience includes, automotive manufacturers, suppliers, development teams, quality assurance, consumers, and other interested parties. Legal entities involved in automotive manufacturing and supply chain may be impacted regarding liability and responsibility for safety-related incidents. Compliance with this standard might help demonstrate due diligence in addressing safety concerns.

The ultimate beneficiaries are consumers and road users. Compliance with the standard helps ensure the safety of vehicles and reduces the risk of accidents and injuries caused by electronic malfunctions. Regulatory bodies responsible for automotive safety standards

may use ISO 26262-1 as a reference for setting safety regulations and requirements in the automotive industry.

This ISO standard also affects suppliers of electrical and electronic components, subsystems, and vehicle software. They must adhere to the standard's safety requirements and contribute to the overall safety of the end product.

Overall, ISO 26262-1:2018 has a wide-ranging impact on the automotive industry and its stakeholders, aiming to enhance the safety of vehicles and foster a safety-centric approach to developing electrical and electronic systems used in road vehicles. There is a wide range of entities involved in producing safer automobiles.

Main directives

1. Safety Lifecycle: ISO 26262-1 defines a safety lifecycle with specific phases, activities, and work products. This lifecycle approach ensures that safety considerations are systematically integrated from concept to decommissioning of the vehicle.

2. ASIL Classification: The standard introduces the Automotive Safety Integrity Level (ASIL) classification, which categorizes safety requirements based on their potential impact on safety. ASIL A represents the lowest impact, while ASIL D represents the highest.

3. Safety Requirements: ISO 26262-1 emphasizes the importance of safety requirements engineering, ensuring that functional safety requirements are well-defined and properly documented.

4. Verification and Validation: The standard outlines methods for verification and validation of safety requirements, system architecture, and safety mechanisms to ensure their correct implementation.

5. Safety Case: ISO 26262-1 requires manufacturers to develop a safety case that demonstrates the safety of the system through evidence and argumentation.

6. Change Management: The standard emphasizes the importance of managing changes to safety-related systems to ensure that safety integrity is maintained throughout the vehicle's lifecycle.

In summary, ISO 26262-1:2018 is of paramount importance to the automotive industry, as it provides a comprehensive framework for ensuring functional safety in vehicles. By adhering to this standard, manufacturers can systematically manage safety risks, improve product reliability, and enhance overall road safety for the benefit of drivers, passengers, and pedestrians.

2.3 Automation levels

ADAS makes up a set of driver assistance systems, providing several levels of automation, which range from alerts so that the driver understands the context of the vehicle, up to the execution of actions in systems using Machine Learn algorithms.

As seen in [Nandavar et al. 2023], the adoption of ADAS reduces incidents on roads and improves vehicle driving safety. Therefore, to corroborate with this fact, some of the systems that help to improve the steering reliability through ADAS:

- Adaptive Cruise Control (ACC): When driving a vehicle, legislation requires that a distance is respected in relation to other cars on the track, as if If braking is necessary, those involved will have time for the total stop to occur. safely. To achieve this, ACC helps define the correct spacing of according to the driver's current speed.

- Lane Keeping Assist (LKA): On highways with vehicles in opposite or with multiple lanes, it is necessary for the driver to redouble the attention so that an accident does not occur. Therefore, systems like LKA are used to prevent the vehicle from accidentally transgressing the direction, putting passenger safety at risk. In this way, when detecting unusual behavior, the system issues a warning to the driver, which should urgently regain control of the vehicle.

- Electronic Stability Control (ESC): Acts on monitoring and control of vehicle stability and, if a loss of control is detected, age, whether due to a curve taken at too high a speed or lack of grip, can act by reducing engine capacity and applying the brakes with greater pressure and in an asymmetrical way.

- Advanced Emergency Braking (AEB): Responsible for providing emergency assistance braking, monitoring the relative distance to an obstacle in front of the vehicle.

Furthermore, the system can issue alerts about the need to apply the brakes and In case of driver inactivity, it can automatically activate the brakes of emergency.

The levels of automation, or interaction between the vehicles and humans, can be defined on up to 6 scales, according to *U.S. National Highway Traffic Safety Administration (NHTSA)*, which goes from zero to five, in ascending order according to the level of automation, where at level zero, there is no automation and, at level 5, there is no longer the need for a human driver (NHTSA, 2017).

1. At Level 0, the vehicle system can only provide some alerts or relevant information about the driver's driving through a ADAS (Advanced Driver Assistance Systems), for example. Therefore, in this modality, the driver is responsible for monitoring and driving the vehicle.

2. When moving to level 1, ADAS will be able to issue brake and acceleration interventions in the event of a lane change, for example, using ADAS systems such as ACC and LKA, previously described. Therefore, at level 1, the driver must still control and always be attentive.

3. At level 2, in addition to the braking and acceleration controls present at level 1, ADAS can control the vehicle's lateral displacement in systems such as *Highway Pilot* in which automation can help keep the vehicle in the lane. However, the driver must still monitor and drive the vehicle.

4. Upon reaching level 3, the vehicle can drive autonomously. However, many scenarios can result in the system's operational loss, which passes the responsibility for the operation to the driver, who must always be attentive.

5. At Level 4, the driver's action becomes optional. It is still possible to take control if you wish, but the vehicle's systems can maintain safe operation without needing external interventions.

6. Finally, at Level 5, the difference from the previous level is that there is no longer the option for a human operator to drive the vehicle. In this way, an autonomous and self-sufficient system carries out all operations safely.

2.4 Platoon Control, Roadside Units, and Onboard Units

Platoon control involves a group of vehicles traveling closely together in a coordinated manner, aiming to enhance traffic flow efficiency, reduce congestion, and enhance fuel efficiency through collaborative and automated driving. This concept is imperative for developing autonomous driving systems, which have attracted significant investment from the automotive industry.

Roadside Units (RSUs) and Onboard Units (OBUs) are fundamental components of Intelligent Transportation Systems (ITS) and vehicular networks. Both are essential in enabling V2X (Vehicle-to-Everything) communication and advanced intelligent transportation infrastructure. (CHEN et al., 2018)

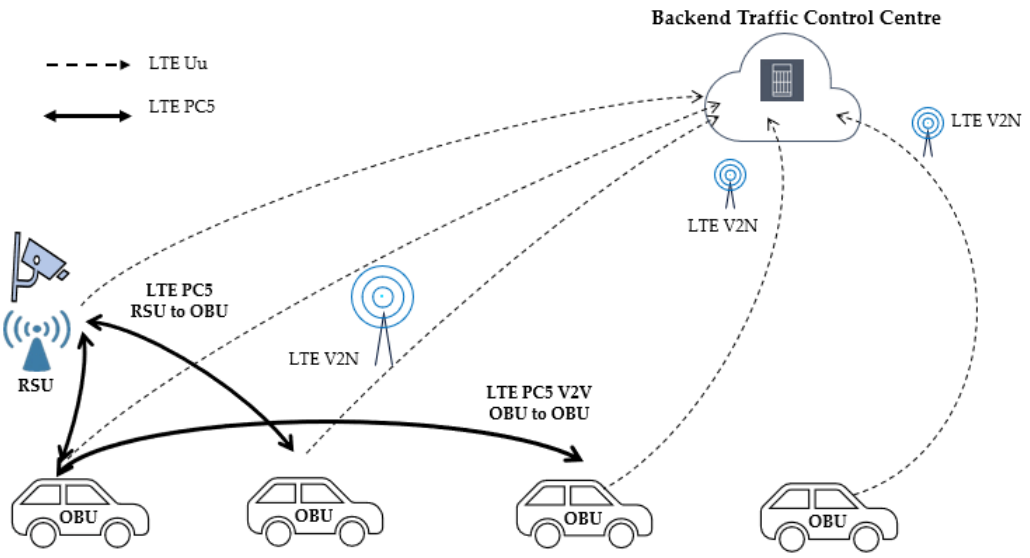

Figure 2: Platoon control system and its devices Source: (MIAO; VIRTUSIO; HUA, 2021)

In summary, RSUs and OBUs are integral to modern transportation systems and the advancement of V2X communication. RSUs act as communication hubs facilitating data exchange between vehicles and infrastructure, while OBUs enable vehicles to communicate with each other and the surrounding environment. These components contribute to safer, more efficient, and more intelligent transportation systems by enabling real-time data exchange, traffic management, safety applications, and more. Figure 2 depicts the communication dynamics.

3 Related work

As mentioned in (CHEN et al., 2018), current communication models rely on recent research material. For example, (MILANÉS et al., 2014) proposes a joint control communication design to achieve reliable vehicle platooning. The results were exciting, as they were very close to the expected parameters. In Figure 3 it is possible to observe the efficiency of ACC assisted by V2V communication (using DSRC devices), implementing their solution in a real-life scenario. The acceleration and braking curves are similar to the expected results.

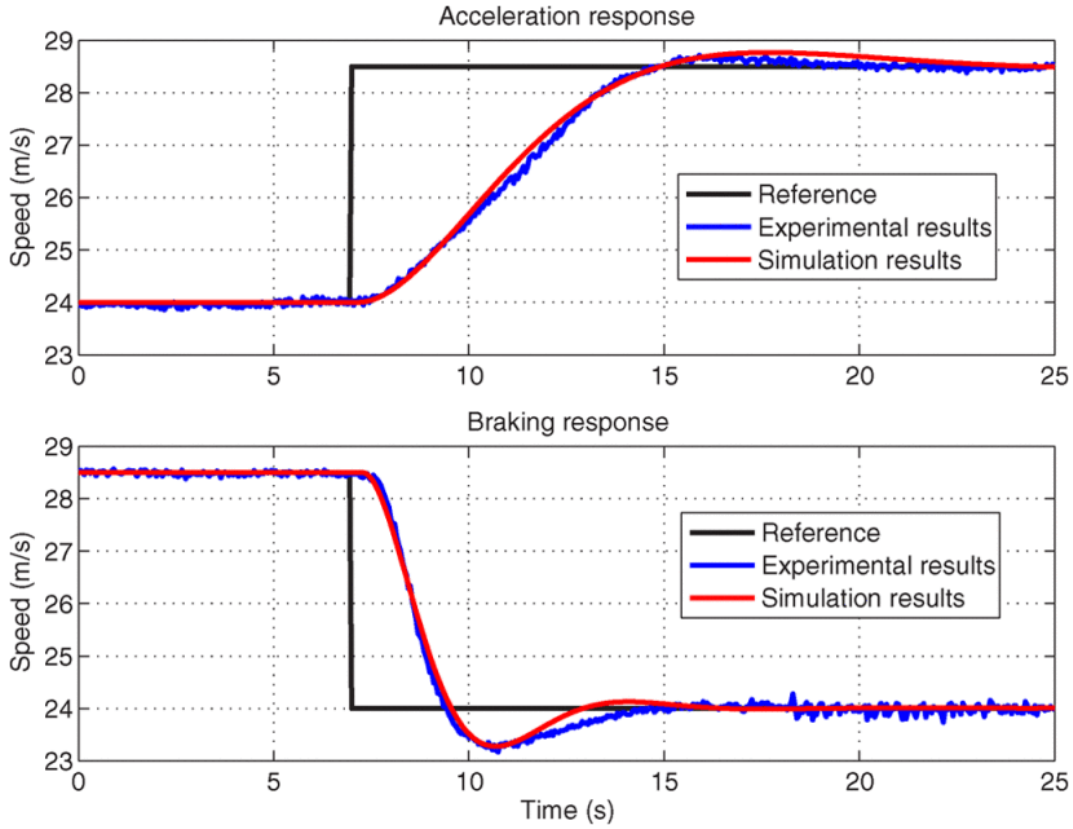

Figure 3: Vehicle's response for acceleration and braking maneuvers. Source: (MILANÉS et al., 2014)

With the implementation of the Cooperative Adaptive Cruise Control (CACC) (an extension to the adaptive cruise control (ACC) concept using Vehicle-to-Everything (V2X) communication), the results are much more consistent and demonstrated by (MILANÉS et al., 2014). The research compares ACC and CACC using three variables: speed, acceleration, and time gap. Reduced gap variability was demonstrated and is visible when comparing Figures 4 and 5.

The average time gap (in seconds) is much smaller in Figure 5 (with CACC). It is noticeable that the variation is also much smaller in the scenario using V2V communication to support the ACC.

The results are conclusive, and a proof of the importance of V2V communication in improving Platoon Control Systems' performance. (GUO; WEN, 2016), for example, establish a

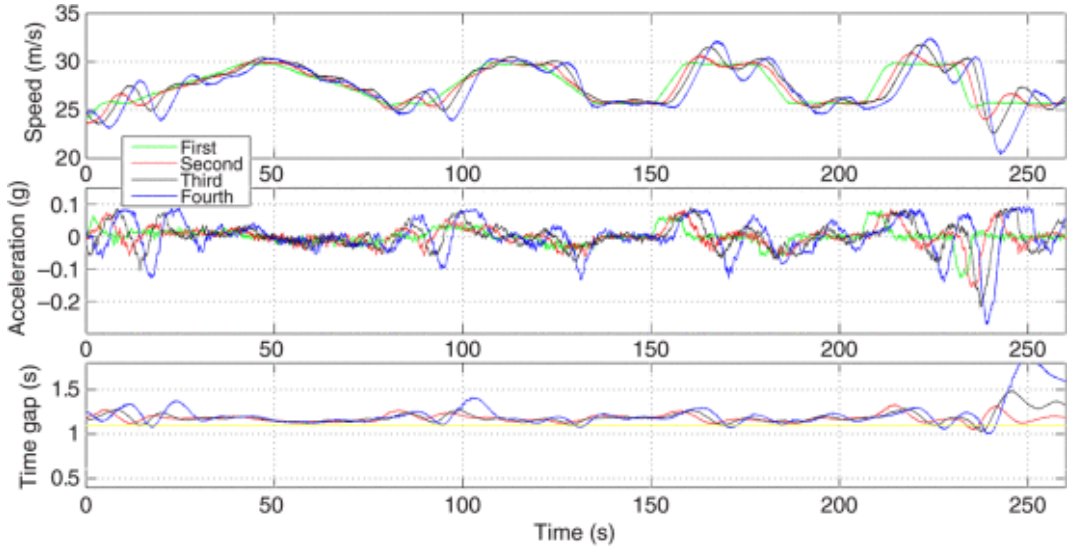

Figure 4: Detail of the last deceleration for the ACC test. Source: (MILANÉS et al., 2014)

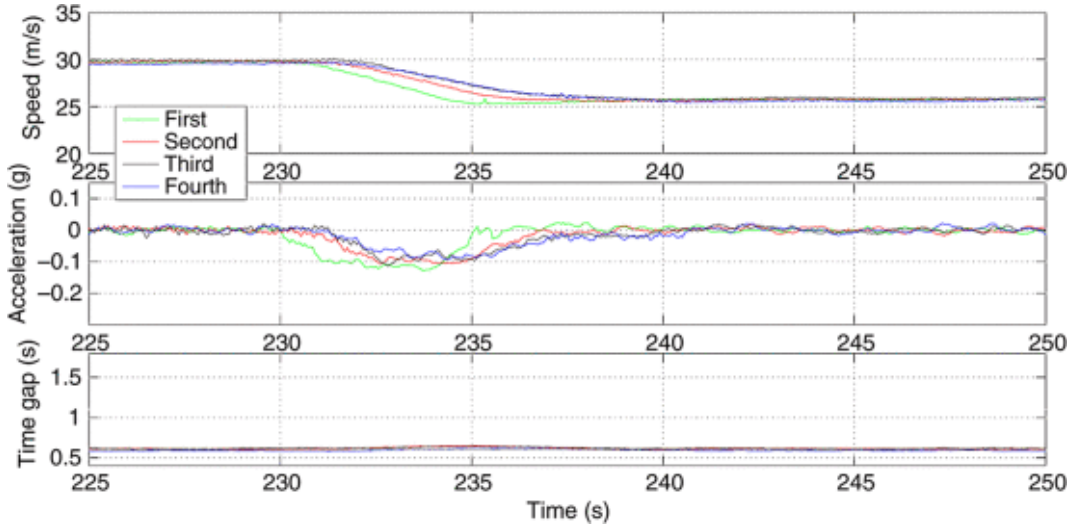

Figure 5: Detail of the last deceleration for the CACC test. Source: (MILANÉS et al., 2014)

network access scheduling and platoon control codesign framework to resolve network access conflicts in Vehicular Ad Hoc Networks (VANETs).

(JIA; NGODUY, 2016), for example, has demonstrated a cooperative microscopic traffic model considering V2X communication. The research also observes the effects of general vehicular communications to the vehicle's cooperative driving.

Connected devices in modern automotive are a trend that is likely to continue in the coming years. All the associated challenges and opportunities are directly affected by security (SEDAR et al., 2023).

Security concerns are growing with the adoption of IP and Ethernet protocols. (MORTAZAVI; SCHLEICHER; GERFERS, 2018) Pivoting from one network to another is a significant issue when developing a cybersecurity solution. In its survey, (SEDAR et al., 2023) used keywords-based search queries to fetch pertinent publications dealing with proactive, reactive, and AI/ML-based defense approaches in vehicular communications security. The results have shown a huge growth in the number of publications.

Another good example is the implementation of automated lane changing maneuvers in a V2X communication environment to reduce the probabilities of collisions. (MILANÉS et al., 2014) has also provided insights on the safe execution of such features.

It is important to note that some of the studies mentioned above focused mainly on theoretical analysis and did not conduct experiments on realistic V2X communications, and might be appliable to our research. (CHEN et al., 2018)

During the last two years, researchers have demonstrated how easy it might be for a vulnerability in a manufacturer's servers to lead to complete access and control of a vehicle. (HUGHES, 2017) In a specific incident with Tesla's infrastructure, Jason Hughes took over an entire fleet of cars. A single system called "mothership" was responsible for communicating with all of them. This vulnerable system allowed the security researcher to send commands directly to the vehicles.

Centralizing control features inside cloud environments might considerably increase associated risks and potential threats, as demonstrated in (TAYLOR et al., 2021). In the article (UÇAR; ERGEN; ÖZKASAP, 2017), the author explores platoon management using the current dominant IEEE 802.11p (DSRC) and hybrid DSRC-Visible Light Communication (VLC).

Testing is an active research field to ensure platoon stability. Furthermore, before the practical deployment of vehicular platoons, DSRC and hybrid DSRC-VLC based management protocols need to be analyzed in the presence of attackers. The main objective is to identify security vulnerabilities of DSRC and DSRC-VLC-based platoons under the jamming and fake platoon maneuver. The research clearly demonstrates that DSRC is highly vulnerable to such attacks. Although VLC limits the effect of adversaries, hybrid architecture still suffers from jamming, fake maneuver and other attacks as described in (GAO et al., 2022), (MALIK; KHAN, et al., 2023), (BERMAD; ZEMMOUDJ; OMAR, 2019), (WANG, R. et al., 2022).

Denial of service (DoS) attacks and uncertain dynamics related to Platoon Systems disrupt communication channels, potentially leading to data loss and jeopardizing control performance, vehicle stability, and safety. To ensure platoon stability during DoS attacks, researchers introduce differente methods to reduce the impact of packet loss, as seen in (PETRENKOV; AGAFONOV, 2021), (WANG, R. et al., 2022). Other major concern is the lack of

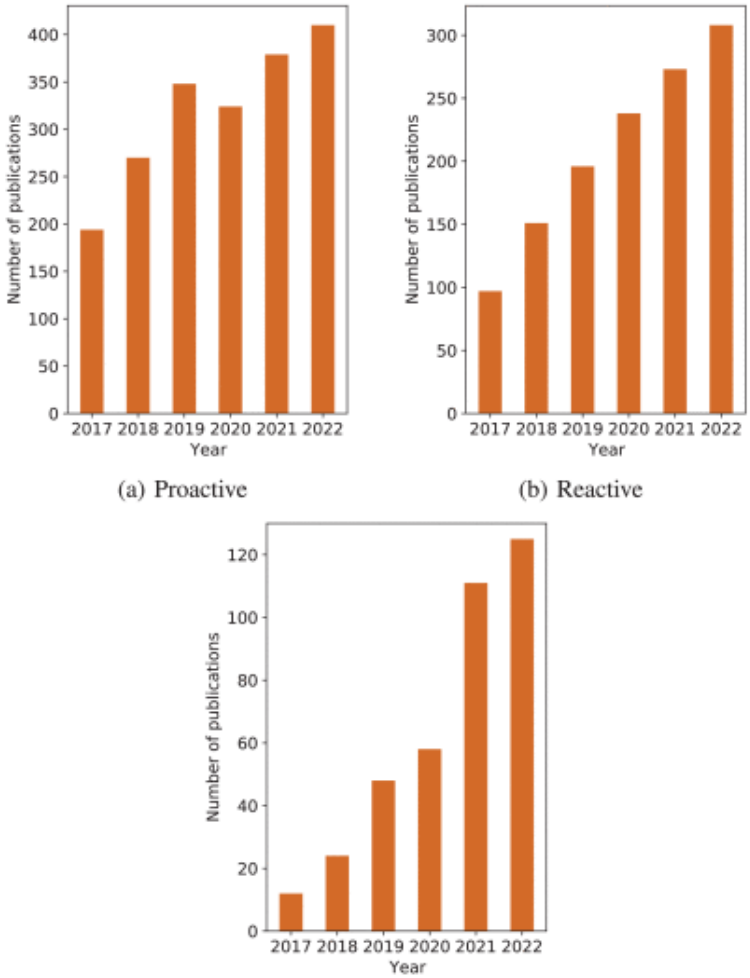

Figure 6: Number of publications per year for (a) proactive, (b) reactive, and (c) AI/ML defense mechanisms pertaining to vehicular communications security. Source: (SEDAR et al., 2023)

standardized authentication. Most recent research suggests using Software Defined Networks (SDN) and Aggregated Message Authentication Codes (AMACs) techniques in 5G-V2X network to reduce handover signaling overhead and communication delay during authentication. (LI; LAI, 2020)

Data-driven-based distributed security control approaches have shown to be effective with vehicle-following platoons in the presence of denial-of-service attacks. (CHE; DENG; LIU, 2020). Distributed resilient observers are designed to estimate the trajectories of the leader-vehicle. Based on the developed observers, data-driven controllers are proposed for the vehicle-followers. Specifically, a data-driven approach is introduced to identify controller parameters without relying on the knowledge of system dynamics.

Security platoon control protocols are implemented differently. Some of them consider the length of lost packets, others analyze internal stability and designs controllers. Simulation results from (UÇAR; ERGEN; ÖZKASAP, 2017) demonstrate the effectiveness of the proposed approach. Possible implementations of authentication include blockchain authentication, as demonstrated in (CARVAJAL-ROCA; SHI; WANG, 2022).

Most of the work observed relates to proprietary technology and expensive equipment. Cutting-edge technologies, such as 5G networks (as observed in (MUSTAFA; YAO HUANG, 2020), (LIU et al., 2021), and (LI; LAI, 2020)), represent a more significant barrier for researchers, as the cost and availability of equipment are not attractive.

4 An environment for testing V2X communication

The swift evolution of vehicular technology, driven by the proliferation of connected and autonomous vehicles, has ushered in an era where Vehicle-to-Everything (V2X) communication is pivotal. V2X communication promises safer and more efficient roadways, where vehicles and infrastructure seamlessly exchange critical data to enhance road safety, traffic management, and the overall driving experience. However, the reliability and robustness of V2X communication systems require rigorous testing and validation in affordable and efficient testing environments before deployment on real-world roads.

Implementing an environment that facilitates testing and debugging V2X communication is a relevant achievement, especially in a scenario with multiple vehicles, such as platoon structures. Through careful orchestration of hardware, software, and simulation tools, this environment seeks to emulate the intricate dynamics of vehicular networks, enabling researchers and practitioners to scrutinize the performance, security, and interoperability of V2X systems.

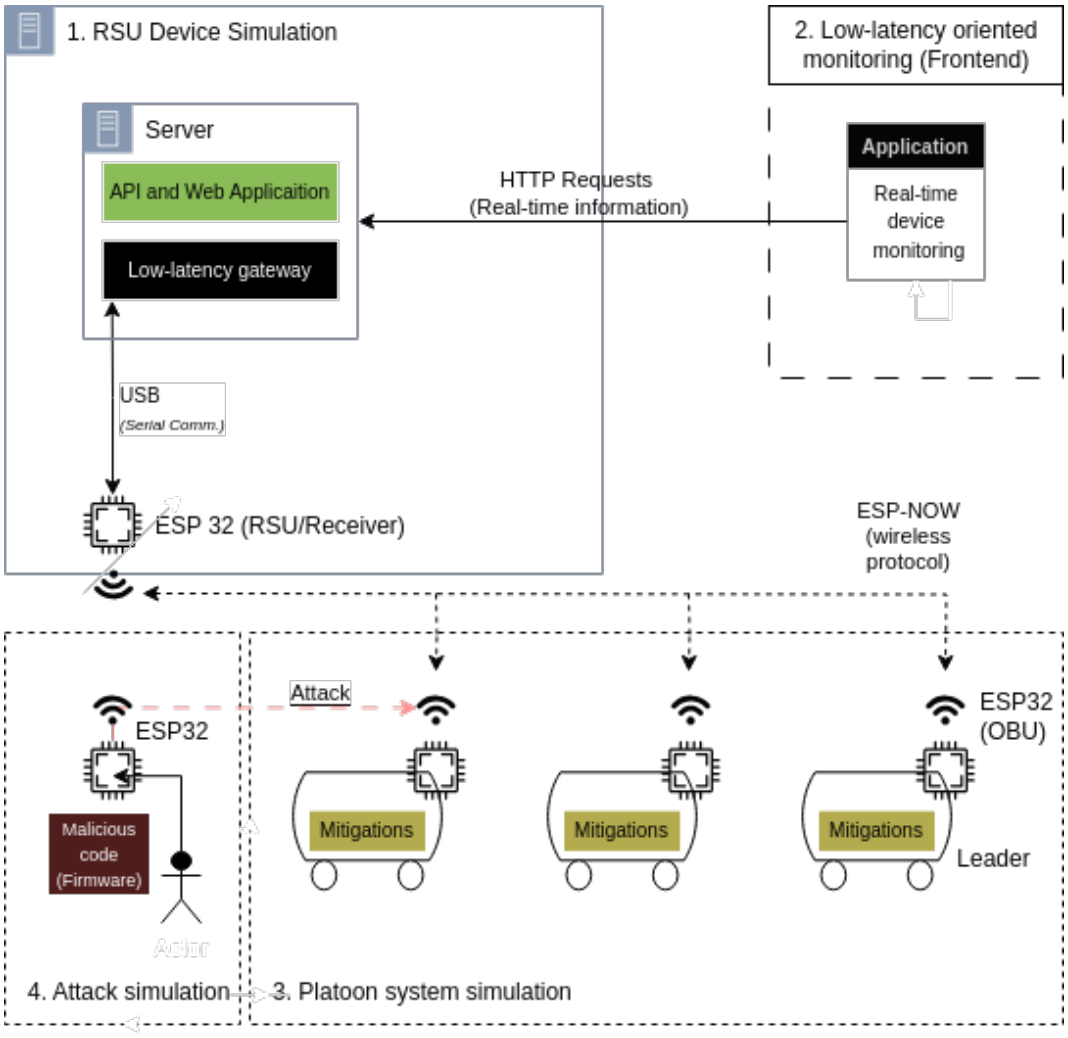

Figure 7: Proposed architechture for simulating V2X communication using ESP32 boards.

This section comprehensively overviews the design considerations, components, and methodologies underlying possible V2X communication testing environments. It aims to equip readers with the insights necessary for effective and reliable testing of V2X communication protocols and applications, as demonstrated in our proposal (Figure 7). The power of V2X communication testing environments becomes a necessity and a strategic advantage as the automotive landscape evolves and more researchers can work on testing automotive networks' security.

The successful implementation and validation of V2X (Vehicle-to-Everything) communication scenarios demand a robust and well-structured testing environment that accurately emulates the intricacies of real-world vehicular networks. Pursuing new developments in vehicular technology and ensuring the safety, efficiency, and reliability of the V2X communication system could be easier with an architecture of the testing environment.

The proposed architecture is essential for accurately replicating and assessing vehicular communication dynamics and specific scenarios, such as security incidents. By synthesizing various hardware, software, and simulation components, this architecture aims to simulate the complex interplay of vehicles (OBUs, for example), roadside units (RSUs), and communication protocols that characterize V2X scenarios..

Through a systematic breakdown of the architecture's key elements, design considerations, and interconnections, this section provides a guide for creating a testing environment for V2X communication. As we observe the complexities of the proposed architecture and the implementation of communication protocols, real-time simulation tools and logging features are demonstrated in Figure 6. Such contributions are valuable for building AI models in future research regarding anomaly detection using AI.

This section aims to elucidate the architecture's technical aspects and emphasize the broader significance of its role in shaping the future of connected and autonomous vehicles. The proposed architecture vividly explores and refines V2X communication. With the cooperation of fellow undergraduate and graduate students, engaged with developing automotive systems and security solutions, it was possible to establish the main aspects of our proposal (Figure 7):

1. **RSU Device simulation**
    (a) **Low-latency gateway** - responsible for collecting low-latency data from the ESP32 used as receiver, as well as providing the collected information to consumers such as our API and Web Application.
    (b) **API and Web Application** - stores and serves data to applications such as real-time monitoring.
2. **Low-latency oriented monitoring application** - focused on providing real-time information from the connected devices.
3. **Platoon system simulation**
    (a) **Using ESP32 boards to simulate OBU and RSU behavior**
4. **Attack simulation** - on the next section, we will discuss how to test attack scenarios and possible mitigations.

Developing the API and web application is crucial in creating an assertive solution for future research. As we focus on cybersecurity, testing scenarios with possible attacks and their mitigations should be easier with tools to assist with data consumption. The low-latency monitoring application, even though part of our proposed architecture, it is not part of the scope of our research. In the next section, details of each component will be disclosed.

4.1 RSU Device Simulation

While navigating through the nuances of RSU simulation, readers will gain insights into the technical intricacies and comprehend the broader implications of RSUs in shaping the future of connected mobility. This section endeavors to unravel the transformative potential of RSUs while equipping readers with the essential knowledge to model and simulate these critical components of the modern road ecosystem.

Through a comprehensive exploration of simulation techniques, software tools, and real-world deployment scenarios, this section also aims to provide a profound understanding of how to replicate the behavior of a Roadside Unit. This implementation of a V2X simulated environment focuses on receiving, storing, and providing data for testing and debugging purposes. Our choices of technologies were based on a good balance between performance and a good learning curve.

4.1.1 Low-latency gateway

This section of our architecture is part of a parallel research currently being developed by students from the Informatics Center of the Federal University of Pernambuco. The group focuses on automotive industry research and innovation. Currently, most students require working in a reduced environment for testing V2X communication in general. Figure 8 is an example of a testing environment using an electric toy car for creating a reduced environment focused on Platoon Control Systems.

The capture of messages between vehicles is accomplished through a serial connection (USB) utilizing an ESP32 that is purpose-programmed for this function. It is imperative to note that the ESP32 is an RSU simulator and will be programmed to receive Platoon Control System-related messages. The API developed during this research, represented by a green box in Figure 7, receives information via HTTP POST requests forwarded by the Low-latency gateway. Our solution stores and serves the information for real-time monitoring and other applications.

4.1.2 API and web application for debugging and testing purposes

The roadside unit simulator (back-end application) was designed to store and serve data for AI model training, data science, and real-time debugging. The information includes sensor and real-time data from the vehicles. This data is stored in a database and made available to AI models for training and inference. The application would also provide a primary user interface and an API for data scientists to explore and visualize the data. Additionally, the application would allow engineers to debug V2X communication in real-time.

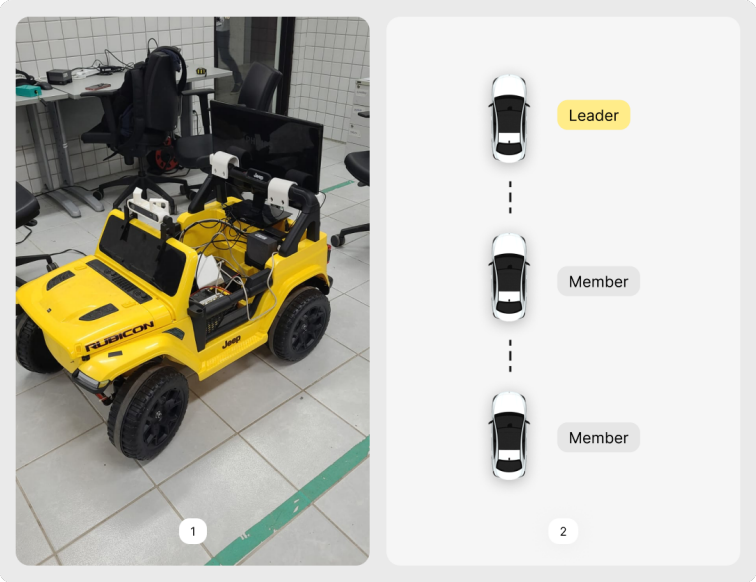

Figure 8: Reduced environment for testing Platoon Control Systems

A roadside unit simulator would be a valuable tool for studying safety and efficiency mechanisms of transportation systems. By storing and serving data for AI model training, the application would help to develop more accurate and reliable AI models. The application would make it easier to explore and visualize data by providing a user interface for data scientists. Furthermore, by allowing engineers to debug V2X communication in real time, the application would help ensure autonomous vehicles' safety. As demonstrated by (MILANÉS et al., 2014), V2X communication directly impacts ACC implementations.

Defining the framework to develop our application was the first challenge to start our research. After identifying the due dates of the project, as well as the number of necessary endpoints and the amount of data to be processed, programming in Python has shown to be the best choice. We have chosen a Python framework called Flask. It has been proven to be the best option compared to Django and other MVC-based frameworks. If you need to build a complex web application, Flask may have limitations. However, if you are looking for a lightweight, flexible, and easy-to-use framework for building smaller web apps and APIs, then Flask was the most interesting option.

As demonstrated in Figure 9, our application is build using a Python framework called Flask. Three endpoints serve our primary web application:

1. Route: GET '/' - Action: Lists the vehicles in the page.
2. Route: GET '/reset' - Action: Resets the database.
3. Route: GET '/vehicle/<name>' - Action: Displays all the logs from a specific vehicle in the page

However, the most important routes of our applications are associated with the functionalities relying on the API. The functions are similar to our primary web application. However, there

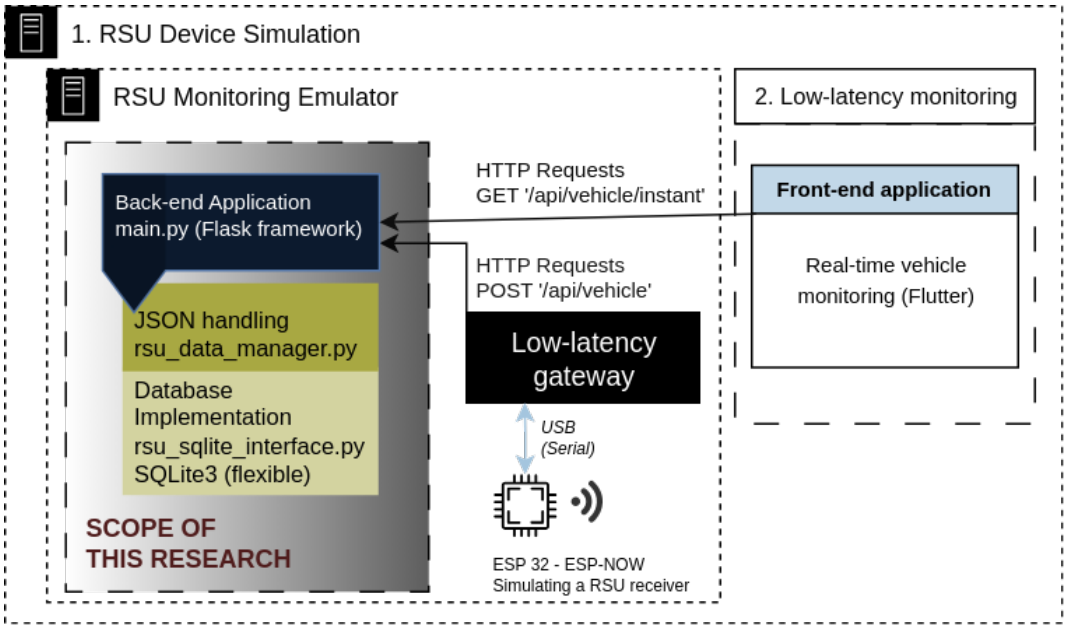

Figure 9: API and web application architecture

are some formatting standards that were followed to facilitate the parallel research.

1. Route: GET '/api/vehicle/instant' - Action: Returns a JSON with the list of connected vehicle and real-time information about them. The format is guided by the developed front-end application. This API was developed to receive more information in the future. The Overview information was the minimum viable product of our research. It contains:

   (a) Name of the vehicle

   (b) Is the vehicle connected (in the past X seconds)? Is the vehicle the leader of the Platoon?

   (c) Angle, Velocity and Distance

   (d) Velocity Setpoint and Distance Setpoint, important to evaluate the efficiency of maneuvers, for example.

   Listing 1 is an example of JSON received by the front-end application when sending a GET request on this endpoint. It was especially designed to be used by the real-time monitoring application, as it uses a separate table with only the last message of each vehicle to provide latest status in reduced latency.

```
{
  "vehicles": [
    {
      "overview": {
        "name": "test_vehicle_2",
        "isConnected": false,
        "isLeader": false,
```

```
        "angle": {
          "icon": "speed",
          "value": "-12"
        },
        "distance": {
          "icon": "speed",
          "value": "2 m"
        },
        "distanceSetpoint": {
          "icon": "bullseye-arrow",
          "value": "2 m"
        },
        "velocity": {
          "icon": "speed",
          "value": "12 RPM"
        },
        "velocitySetpoint": {
          "icon": "bullseye-arrow",
          "value": "12 RPM"
        }
      }
    }
  ]
}
```

Listing 1: JSON format example for the front-end application.

2. Route: POST '/api/vehicle' - Receives a JSON in a specific format from the Low-latency gateway. The information is collected from the vehicles by the ESP32 used as a receiver for the ESP-NOW protocol.

   Listing 2 is an example of JSON sent via POST Request to our API by the Low-latency gateway, developed in parallel with this research.

```
{
    "sender":"test_vehicle_name",
    "receiver": "test_vehicle_name_2",
    "data":{
        "isLeader": "False",
        "velocity": "12",
        "angle": "-12",
        "distance": "2",
        "velocitySetpoint": "12",
        "distanceSetpoint": "2"
    }
}
```

Listing 2: JSON format example of received data from the Low-latency gateway.

3. Route: GET '/api/vehicle/<name>' - Returns a crude array of dictionaries, each one of them is a message stored in the database.

4.2 Real-time device monitoring research

One of the solution's requirements was displaying data in real-time or with the minimum possible latency. An average latency of 2 seconds is acceptable on the front-end visualization. With that in mind, a local HTTP server hosts the BFF framework. The front-end application makes regular (parameter-defined) calls to the server, updating data on the application.

Sustaining real-time parameters and updates is challenging, especially considering the growing data traffic. The application prioritizes instant data. In consequence, lightweight and robust frameworks are imperative. Those requirements were imperative while developing the API and Web application responsible for serving the data.

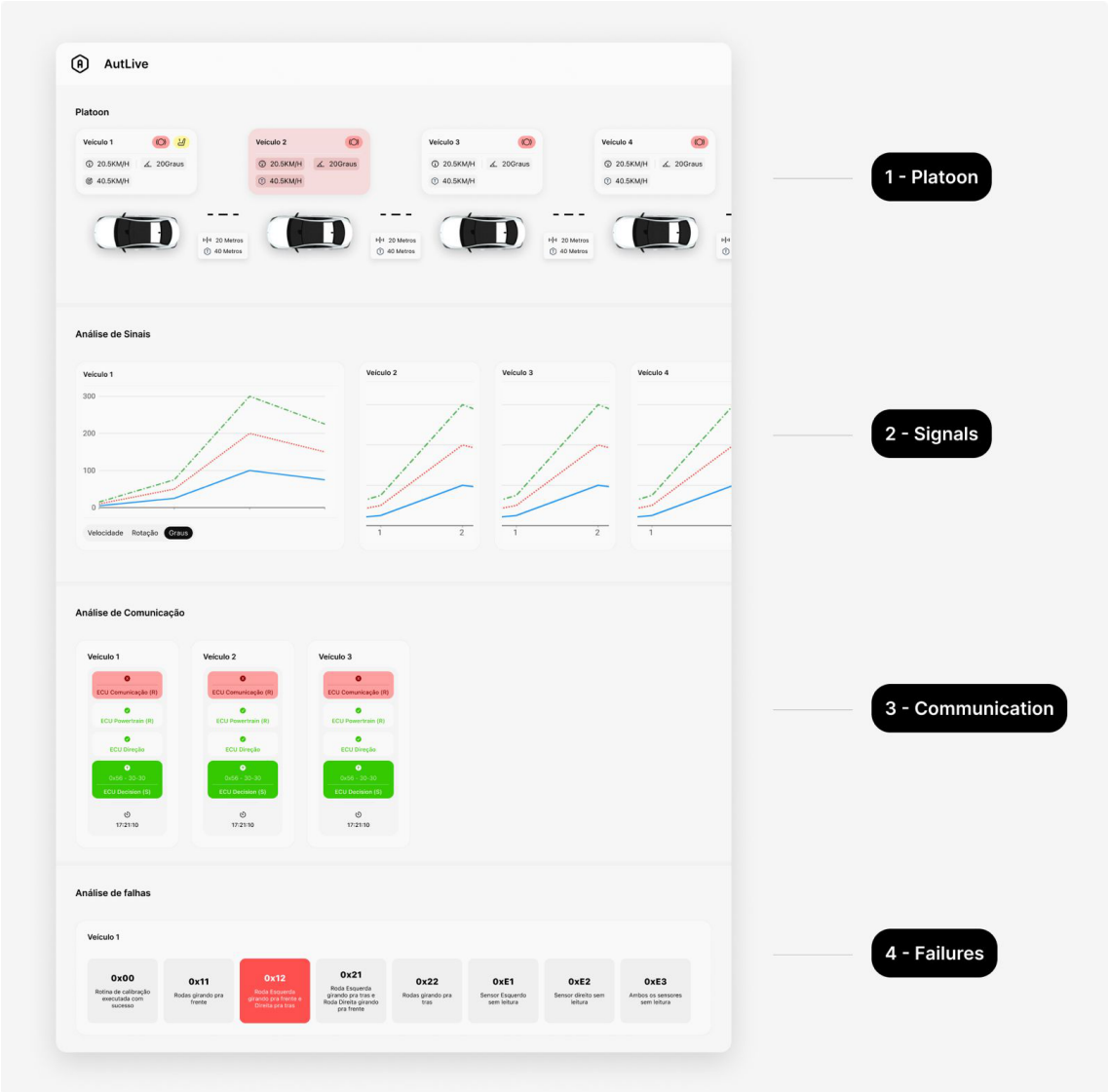

Figure 10: Prototype developed by students from LIVE-UFPE

There are numerous front-end development technologies currently available. Flutter was

chosen because of its flexibility and simplicity, especially considering cross-platform deployment. In addition to the requirements, Flutter provides features and improvements beyond cross-platform development (compared to React Native, for example, (WU, 2018)).

As it is not a native framework for the target platform, the performance is a relevant factor that must be considered, depending on the number of visual components presented on the screen and constant updates. As described by (WU, 2018), Flutter tends to achieve better results than React Native, especially in scenarios with large lists and elements that require high computational resource usage. Another relevant feature is generating Windows/MacOS applications and web apps. Changing operating systems does not impact the use of the application.

Created components are updated in "real-time", speeding up the development process (layout errors can be quickly identified and fixed). Figure 10 is a high-fidelity front-end prototype developed by Samuel Souza, who works in cooperation with students from LIVE-UFPE, a research group focused on smart cities and related topics.

4.3 Platoon system simulation using ESP32 boards

To simulate a V2X environment, we need to create a model of the vehicles, the infrastructure, and the communication network. We can then use this model to simulate different scenarios, such as accidents, traffic jams, and cyberattacks.

ESP32 is a wise choice for simulating V2X communication due to its versatility, capabilities, and cost-effectiveness (especially considering our scenario). Here are some reasons why ESP32 boards are well-suited for this purpose:

- Wireless Communication: ESP32 boards come equipped with built-in Wi-Fi and Bluetooth capabilities. This makes them suitable for simulating wireless communication scenarios between vehicles, similar to how V2V communication occurs.
- V2V Protocols Support: ESP32 boards can be programmed to support various V2V communication protocols, such as DSRC (Dedicated Short-Range Communication) or IEEE 802.11p. This allows for realistic simulation of communication exchanges between vehicles.
- Real-Time Capabilities: ESP32 boards offer real-time processing capabilities, enabling them to process and respond to messages in a time-sensitive manner, which is crucial in V2V scenarios to ensure safety and coordination.
- Multi-Threaded Operation: ESP32 supports multi-threading, allowing you to simulate multiple vehicles with individual threads, each executing its own communication behavior. This is important for creating a more realistic simulation of simultaneous V2V communications.
- Low-Cost Solution and customizability

  - ESP32 boards are relatively affordable, making them a cost-effective choice for creating a fleet of simulated vehicles. This affordability enables scalability in creating larger-scale simulations.

  - Those boards are fully programmable. Associated with different programming languages and frameworks, it allows you to customize their behavior to match specific V2V scenarios or communication protocols.

- Integration with Sensors: ESP32 boards can be easily integrated with various sensors like GPS, accelerometers, and cameras. This allows you to simulate sensor data exchange alongside V2V communication.

- Community Support and rapid prototyping

  - ESP32 has a large and active community, which means you can find ample resources, libraries, and tutorials to aid in developing your V2V simulation setup.

  - The availability of development environments and tools like the Arduino framework makes rapid prototyping of V2V communication scenarios feasible.

- Energy Efficiency and Scalability

  - ESP32 boards offer power-saving modes and efficient energy consumption, which can be important when simulating battery-operated vehicles.

  - Due to their low cost and availability, you can create a network of interconnected ESP32-based simulated vehicles to study more complex V2V scenarios.

In summary, ESP32 boards provide a convenient and cost-effective platform to simulate intervehicular communication scenarios. Their wireless communication capabilities, real-time processing, and integration potential make them well-suited for emulating V2V communication protocols and behaviors, helping researchers and developers assess and improve V2V communication systems in a controlled environment.

4.4 Attack simulation using ESP32

Testing the security and resilience of V2X (Vehicle-to-Everything) systems using ESP32 boards can be done effectively by simulating attacks. This is a cost-effective way to perform such tests. In this section, we have demonstrated an environment that emulates V2X communication behavior. The environment provides an emulated RSU which captures valuable data for analysis. To simulate any kind of cyberattack in this environment, some general steps outlined below should be followed.

1. **Preparing the Environment** Gather the necessary hardware, including ESP32 boards, computers, and any other equipment required to implement our proposed environment.

(a) At least four ESP32 boards:
   i. 1 with the malicious firmware, responsible for attacking
   ii. 2 of them simulating V2X usual communication
   iii. 1 as a receiver to enable the low-level gateway application to send real-time messages to our back-end application.

(b) A server with any linux distribution with the following services:
   i. Back-end application: Follow the instructions available at the repository. Deploy the Flask application very fast with Gunicorn.
   ii. Low-level gateway: Responsible for collecting the messages from the ESP32 receiver and sending them via POST request to the back-end application
   iii. Front-end application for real-time monitoring: developed to improve monitoring conditions for researchers.

It is imperative to conduct experiments in a controlled testing environment, such as a lab or testbed, for maximum safety and accuracy.

2. **Identifying Attack Scenarios** This is the most important step of security testing. Specific attack scenarios should be defined. These could include attacks on message integrity, confidentiality, availability, or other aspects of V2X communication.

3. **Develop Attack Code** Write code for the ESP32 boards to simulate the chosen attacks. Depending on the type of attacks you want to simulate, this might involve sending malicious messages, spoofing GPS data, jamming wireless signals, or other attack techniques. In the next section we will further discuss an example of malicious code.

4. **Implement Security Measures** If you're testing a V2X system's security features, implement the necessary security measures on the ESP32 boards to protect against the simulated attacks. For example, use encryption and authentication protocols, or simpler solutions like rate limiters to avoid flooding, for example.

5. **Data Collection using the proposed infrastructure** Execute the attack scenarios while monitoring and collecting data from the ESP32 boards using the proposed infrastructure. The results will be logged and displayed in real-time using the provided solution.

6. **Iterate and Improve** It will be possible to analyze the data collected to identify anomalies and weaknesses in the V2X system. Modify your attack code or security measures to refine the testing process. Repeat the testing and data collection process until you have a good understanding of the system's strengths and weaknesses.

7. **Document and Report** Document the results of the simulations, including the technique's name, the impact on the V2X system, and any recommendations for improving security. Create a detailed report that can be shared with stakeholders or used to improve the V2X system's security.

8. **Repeat and Validate** Continue to iterate and validate the simulations as new attack vectors or vulnerabilities in V2X systems are identified by researchers.

Testing should comply with relevant regulations and ethical guidelines. Obtain any necessary permissions or approvals for conducting these simulations, especially if they interfere with real-world V2X communications. Be sure to implement safety measures to prevent any unintended consequences or interference with real-world V2X communication.

It is important to keep in mind that conducting security testing and simulations in V2X communication is a complex task that necessitates a thorough comprehension of both V2X technology and cybersecurity principles. Furthermore, it is crucial to prioritize safety and ethical considerations when conducting these simulations, in order to prevent any potential harm or legal issues.

5 Testing V2X cyberattacks and possible mitigations

As previously discussed in this paper, Vehicle-to-everything (V2X) communication is a rapidly emerging technology that has the potential to revolutionize transportation safety and efficiency. V2X allows vehicles to communicate with each other, with roadside infrastructure, and much more. Sharing information about the vehicles' location, speed, intentions, and the surrounding environment is part of its functions.

However, V2X communication enables many different attack vectors. Our case study will demonstrate a simple attack vector that could disrupt or turn off connectivity, resulting in safety hazards. We will also demonstrate a possible mitigation to prevent this specific scenario.

5.1 Flood attack scenario

In this section, our primary goal is to execute a DoS attack, flooding the ESP-NOW protocol with useless messages. Here is a simple attack example that could test the effectiveness of any mitigation measures and monitoring systems.

The code below takes advantage of the Arduino IDE, providing excellent abstraction. It starts up the ESP32 unit and its connectivity protocols (WiFi and ESP-NOW - as seen in the setup() function). Later on, using the loop function, we remove the delay and any control variables to generate a fast and infinite message sender, flooding the ESP-NOW protocol.

```
1  // ESP-NOW flood generating example
2
3  // Common libraries
4  #include <ESP32WiFi.h>
5  #include <espnow.h>
6
7  // Constants
8  const char *ssid = "YOUR_SSID";
9  const char *password = "YOUR_PASSWORD";
10 const uint64_t peer_address = 0x0000000000000001;
11
12 // Starting Wi-Fi and ESP-NOW
13 void setup() {
14   Serial.begin(115200);
15   WiFi.begin(ssid, password);
16   while (WiFi.status() != WL_CONNECTED) {
17     delay(500);
18     Serial.print(".");
19   }
20   Serial.println();
21   Serial.println("Conectado ao Wi-Fi");
22
23   esp_now_init();
24   esp_now_set_peer_info(peer_address, ESP_NOW_ROLE_SLAVE, NULL, 0);
25  esp_now_register_recv_cb(onReceive);
26 }
27
28 // Loop
```

```
29 void loop() {
30     // Send a packet
31     char message[] = "Hello world!";
32     esp_now_send(peer_address, message, sizeof(message));
33 
34     // Wait X seconds - commented to avoid any interval and generate flooding with
           the loop
35     // delay(1000);
36 
37 }
```

Listing 3: Firmware for ESP32 devices to generate DoS attacks on ESP-NOW protocol networks.

The complexity of the exploitation depends on the complexity of the associated vulnerability. In this case, we are taking advantage of the architecture of ESP-NOW protocol, which leads to simple exploitation opportunities. As a simples multicast protocol, without reasonable security features, there is a wide range of attack vectors that could be exploited (such as Man-in-the-middle (MITM)and Spoofing).

5.2 Implementing security mitigations

There are multiple possible implementations to be included in the device's firmware to fix this kind of vulnerability. Below there are are some mitigations for DoS/Flooding attacks in general:

1. Use a rate limiter. prevents a device from sending too many messages in a short period of time.

2. Use a message filter. allows you to ignore specific messages or messages from specific sources.

3. Use encryption: makes it more difficult for attackers to intercept and manipulate messages.

4. Monitor your network for suspicious activity: allows you to detect attacks early and take steps to mitigate them.

On our research, we demonstrate a method to implement a mitigation against flooding attacks ESP-NOW protocol. In this case, we will take advantage of a rate limiter. The code below implements the rate limiter in the callback function associated with receiving packets. It is a simple implementation for testing purposes, and could also be implemented using different mechanisms under different platforms.

```
1 // Loop
2 void loop() {
3     // Send a packet
4     char message[] = "Hello world!";
5     esp_now_send(peer_address, message, sizeof(message));
6 
```

```
7       // Wait X seconds - commented to avoid any interval and generate flooding with
            the loop
8       // delay(1000);
9
10  }
11
12  // Define the constants
13  const int max_messages_per_second = 10;
14
15  // Callback for received packets
16  void onReceive(uint8_t *data, uint16_t len) {
17    Serial.print("Received: ");
18    for (int i = 0; i < len; i++) {
19      Serial.print(data[i]);
20    }
21    Serial.println();
22
23    // Check if the rate limiter has been exceeded
24    if (esp_now_get_tx_queue_len(0) >= max_messages_per_second) {
25      Serial.println("Rate limiter exceeded.");
26      return;
27    }
```

Listing 4: Firmware for ESP32 devices to generate DoS attacks on ESP-NOW protocol networks.

In Line 24 of Listing 5.2, the number of messages in the queue of the ESP-NOW protocol is validated. The code prints an error if it is larger than the maximum of messages per second (defined as a variable in Line 13). At that point, the code could prevent the device from reading the messages using a timeout. This implementation is not a complex example of a security mechanism but illustrates how the firmware could be adjusted to avoid flooding. There are some benefits of using a rate limiter:

1. Improve the performance of a system by preventing it from being overwhelmed by requests.
2. Helps to prevent legitimate users from being denied service due to flooding attacks.
3. Protects a system from security threats, such as DoS attacks.

However, there are also some drawbacks to rate limiting usage, in general, that should be considered:

1. Difficult to configure correctly.
2. Consequently slows down legitimate traffic.
3. It is not very difficult to be bypassed by attackers.

Overall, rate limiting effectively prevents flooding attacks and is a good starting point. However, it is crucial to consider more complex alternatives and implement a comprehensive security approach. Considering the V2X communication scenario, we have identified other alternatives to prevent DoS in general. These methods have caught our attention because they effectively prevent different attacks associated with malicious network traffic.

1. Traffic shaping: Used to control the flow of traffic on a network. This can be used to prevent certain types of traffic, such as flooding traffic, from reaching a system.

2. Intrusion detection systems (IDSs): Used to detect malicious traffic, such as flooding traffic. IDSs can be used to alert administrators about flooding attacks and to take steps to mitigate them, taking advantages of detection rules to identify anomalies.

3. Intrusion prevention systems (IPSs): IPSs are similar to IDSs, but they can also take active steps to mitigate attacks, such as blocking malicious traffic.

This research provides information and tools to facilitate testing V2X communication scenarios and develop such defense mechanisms. It is essential to understand that cybersecurity is always a work in progress. Penetration testing, threat hunting, and other security-related activities are imperative to build a safer environment on automotive networks.

6 Conclusion

Automotive networks are considerably new in the world of information warfare. However, we have seen that even though, to the best of our knowledge, there have been no accidents caused by cyberattacks. Researchers have shown that it is possible to develop remote exploits without user interaction by reverse engineering a modern car (with the proper time, effort, and knowledge). A few of them could even remotely control the car's physical functions. Exploiting centralized infrastructure might spread worm malware across cars and the V2X environment. In case of any malicious intentions, they could cause widespread defacement and potentially many deaths.

In order to support security standards, such as ISO 26262:2018, it is necessary to design security features and mitigations for a broad spectrum of attack vectors. Proper encryption methods and availability are the most important aspects to be implemented and tested. Cooperative adaptive cruise control systems play an essential role in improving the performance of autonomous vehicles. Testing scenarios in reduced environments is imperative to achieve the desired results.

This research demonstrates that it is possible to create a testing environment for V2X communication scenarios with considerably small financial resources using ESP32 boards. In addition to the details of the architecture and implementations, we provide a case study of our proposal, in which we demonstrate a cyberattack and its possible mitigations using the provided infrastructure.

While working on this research with undergraduate and graduate students from UFPE (Federal University of Pernambuco), debugging tools with rich logging and real-time monitoring features were imperative to developing security solutions. For example, developing AI models for automotive intrusion detection systems demands centralized information sources, such as the developed web applications and API, with structured information from the devices' communication. Real-time monitoring of testing environments is also essential and will support future research on simulating V2X communication scenarios and observing the results. Real-time charts can easily be generated by using reference (setpoint) and vehicle (real) values, as shown in Figure 3.

References

BERMAD, Nabila; ZEMMOUDJ, Salah; OMAR, Mawloud. Securing Vehicular Platooning against Vehicle Platooning Disruption (VPD) Attacks. In: 2019 8th International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN). [S.l.: s.n.], 2019. P. 1–6. DOI: 10.23919/PEMWN47208.2019.8986956.

CAN Specification. Version 2.0: Robert Bosch GmbH, 1991.

CARVAJAL-ROCA, Ivan E.; SHI, Jinming; WANG, Jian. A Blockchain-based Lightweight Authentication Protocol for Vehicular Platoons. In: 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring). [S.l.: s.n.], 2022. P. 1–6. DOI: 10.1109/VTC2022-Spring54318.2022.9860654.

CHE, Weiwei; DENG, Chao; LIU, Dan. Data-Driven-Based Distributed Security Control for Vehicle-Following Platoon. In: 2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS). [S.l.: s.n.], 2020. P. 1109–1113. DOI: 10.1109/DDCLS49620.2020.9275217.

CHEN, Wenbo et al. Platoon control for connected vehicles based on the V2X communications: Design and implementation. In: 2018 Chinese Control And Decision Conference (CCDC). [S.l.: s.n.], 2018. P. 6552–6557. DOI: 10.1109/CCDC.2018.8408282.

GAO, Kai et al. False Data Injection Attack Detection in a Platoon of CACC in RSU. In: 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). [S.l.: s.n.], 2022. P. 1324–1329. DOI: 10.1109/TrustCom56396.2022.00186.

GUO, Ge; WEN, Shixi. Communication Scheduling and Control of a Platoon of Vehicles in VANETs. **IEEE Transactions on Intelligent Transportation Systems**, v. 17, n. 6, p. 1551–1563, 2016. DOI: 10.1109/TITS.2015.2505407.

HUGHES, John. **Tesla Network Vulnerability Report - 2017-03-24 (Annotated)**. Accessed: June 16th of 2023. Mar. 2017. Available from: <https://docs.google.com/document/d/1yXni1GoD93q8mX-yom7JLBn0Q8tPOQz2A_y3m3LJi8o>.

HVANTH, Ris; VALLI, D.; GANESAN, K. Design of an In-Vehicle Network (Using LIN, CAN and FlexRay), Gateway and its Diagnostics Using Vector CANoe. **American Journal of Signal Processing**, Scientific and Academic Publishing, v. 1, n. 2, p. 40–45, Feb. 2012. DOI: 10.5923/j.ajsp.20110102.07. Available from: <https://doi.org/10.5923>.

JIA, Dongyao; NGODUY, Dong. Enhanced cooperative car-following traffic model with the combination of V2V and V2I communication. **Transportation Research Part B: Methodological**, v. 90, p. 172–191, 2016. ISSN 0191-2615. DOI: https://doi.org/10.1016/j.trb.2016.03.008. Available from: <https://www.sciencedirect.com/science/article/pii/S0191261515302563>.

KARAGIANNIS, Georgios et al. Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions. **IEEE Communications Surveys & Tutorials**, v. 13, n. 4, p. 584–616, 2011. DOI: 10.1109/SURV.2011.061411.00019.

LEE, Michael; ATKISON, Travis. VANET applications: Past, present, and future. **Vehicular Communications**, v. 28, p. 100310, 2021. ISSN 2214-2096. DOI: https://doi.org/10.1016/j.vehcom.2020.100310. Available from: <https://www.sciencedirect.com/science/article/pii/S2214209620300814>.

LI, Guanjie; LAI, Chengzhe. Platoon Handover Authentication in 5G-V2X : IEEE CNS 20 Poster. In: 2020 IEEE Conference on Communications and Network Security (CNS). [S.l.: s.n.], 2020. P. 1–2. DOI: 10.1109/CNS48642.2020.9162271.

LIU, Feifei et al. Secure Vehicle Platooning Protocol for 5G C-V2X. In: 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). [S.l.: s.n.], 2021. P. 868–875. DOI: 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00123.

MALIK, Abdul; KHAN, Muhammad Zahid, et al. An Efficient Approach for the Detection and Prevention of Gray-Hole Attacks in VANETs. **IEEE Access**, v. 11, p. 46691–46706, 2023. DOI: 10.1109/ACCESS.2023.3274650.

MALIK, Suman; SAHU, Prasant Kumar. Study on wireless communication aspect of VANETs. In: 2018 IEEE MTT-S International Microwave and RF Conference (IMaRC). [S.l.: s.n.], 2018. P. 1–4. DOI: 10.1109/IMaRC.2018.8877354.

MIAO, Lili; VIRTUSIO, John; HUA, Kai-Lung. PC5-Based Cellular-V2X Evolution and Deployment. **Sensors**, v. 21, p. 843, Jan. 2021. DOI: 10.3390/s21030843.

MILANÉS, Vicente et al. Cooperative Adaptive Cruise Control in Real Traffic Situations. **IEEE Transactions on Intelligent Transportation Systems**, v. 15, n. 1, p. 296–305, 2014. DOI: 10.1109/TITS.2013.2278494.

MILLER, Charlie; VALASEK, Chris. Remote exploitation of an unaltered passenger vehicle. **Black Hat USA**, v. 2015, S 91, p. 1–91, 2015.

MORTAZAVI, Sanaz; SCHLEICHER, Detlef; GERFERS, Friedel. Modeling and Verification of Automotive Multi-Gig Ethernet Communication up to 2.5 Gbps and the Corresponding EMC Analysis. In: 2018 IEEE Symposium on Electromagnetic Compatibility, Signal Integrity and Power Integrity (EMC, SI & PI). [S.l.: s.n.], 2018. P. 329–334. DOI: 10.1109/EMCSI.2018.8495375.

MUCH, Alexander. Automotive Security: Challenges, Standards, and Solutions. **Software Quality Professional**, v. 18, n. 4, p. 4–12, Sept. 2016. Available from: <https://www.proquest.com/docview/1817024845>.

MUSTAFA, Iqra; YAO HUANG, Ching. Lightweight cryptographic URLLC for 5G-V2X. In: 2020 International Symposium on Networks, Computers and Communications (ISNCC). [S.l.: s.n.], 2020. P. 1–6. DOI: 10.1109/ISNCC49221.2020.9297191.

NANDAVAR, Sonali et al. Exploring the factors influencing acquisition and learning experiences of cars fitted with advanced driver assistance systems (ADAS). **Transportation research part F: traffic psychology and behaviour**, Elsevier, v. 94, p. 341–352, 2023.

NHTSA. **The Evolution of Automated Safety Technologies**. [S.l.: s.n.], 2017. https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety, Last accessed on 2023-08-03.

PETRENKOV, Denis; AGAFONOV, Anton. Anomaly Detection in Vehicle Platoon with Third-Order Consensus Control. In: 2021 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). [S.l.: s.n.], 2021. P. 0463–0466. DOI: 10.1109/USBEREIT51232.2021.9455022.

SEDAR, Roshan et al. A Comprehensive Survey of V2X Cybersecurity Mechanisms and Future Research Paths. **IEEE Open Journal of the Communications Society**, v. 4, p. 325–391, 2023. DOI: 10.1109/OJCOMS.2023.3239115.

TAYLOR, Sean Joe et al. Vehicular Platoon Communication: Cybersecurity Threats and Open Challenges. In: 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). [S.l.: s.n.], 2021. P. 19–26. DOI: 10.1109/DSN-W52860.2021.00015.

UÇAR, Seyhan; ERGEN, Sinem Çöleri; ÖZKASAP, Öznur. Security vulnerabilities of autonomous platoons. In: 2017 25th Signal Processing and Communications Applications Conference (SIU). [S.l.: s.n.], 2017. P. 1–4. DOI: 10.1109/SIU.2017.7960322.

UPSTREAM. **Global Automotive Cybersecurity Report**. Accessed: June 15th of 2023. 2022. Available from: <https://info.upstream.auto/hubfs/Security_Report/Security_Report_2022/Upstream_Security-Global_Automotive_Cybersecurity_Report_2022.pdf>.

WANG, Jian et al. A Survey of Vehicle to Everything (V2X) Testing. **Sensors**, v. 19, n. 2, 2019. ISSN 1424-8220. DOI: 10.3390/s19020334. Available from: <https://www.mdpi.com/1424-8220/19/2/334>.

WANG, Rongzhen et al. Security Platoon Control of Connected Vehicle Systems under DoS Attacks and Dynamic Uncertainty. In: IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society. [S.l.: s.n.], 2022. P. 1–5. DOI: 10.1109/IECON49645.2022.9968650.

WU, Wenhao. React Native vs Flutter, Cross-platforms mobile application frameworks. Metropolia Ammattikorkeakoulu, 2018.