Secure Harmonized Speed Under Byzantine Faults for Autonomous Vehicle Platoons Using Blockchain Technology

by

Noon Hussein

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Autonomous Vehicle (AV) platooning holds the promise of safer and more efficient road transportation. By coordinating the movements of a group of vehicles, platooning offers benefits such as reduced energy consumption, lower emissions, and improved traffic flow. However, the realization of these advantages hinges on the ability of platooning vehicles to reach a consensus and maintain secure, cooperative behavior.

Byzantine behavior [1,2], characterized by vehicles transmitting incorrect or conflicting information, threatens the integrity of platoon coordination. Vehicles within the platoon share vital data such as position, speed, and other relevant information to optimize their operation, ensuring safe and efficient driving. However, Byzantine behavior in AV platoons presents a critical challenge by disrupting coordinated operations. Consequently, the malicious transmission of conflicting information can lead to safety compromises, traffic disruptions, energy inefficiency, loss of trust, chain reactions of faults, and legal complexities [3,4]. In this light, this thesis delves into the challenges posed by Byzantine behavior within platoons and presents a robust solution using ConsenCar; a blockchain-based protocol for AV platoons which aims to address Byzantine faults in order to maintain reliable and secure platoon operations.

Recognizing the complex obstacles presented by Byzantine faults in these critical real-time systems, this research exploits the potential of blockchain technology to establish Byzantine Fault Tolerance (BFT) through Vehicle-to-Vehicle (V2V) communications over a Vehicular Ad hoc NETwork (VANET). The operational procedure of ConsenCar involves several stages, including proposal validation, decision-making, and eliminating faulty vehicles. In instances such as speed harmonization, the decentralized network framework enables vehicles to exchange messages to ultimately agree on a harmonized speed that maximizes safety and efficiency. Notably, ConsenCar is designed to detect and isolate vehicles displaying Byzantine behavior, ensuring that their actions do not compromise the integrity of decision-making. Consequently, ConsenCar results in a robust assurance that all non-faulty vehicles converge on unanimous decisions.

By testing ConsenCar on the speed harmonization operation, simulation results indicate that under the presence of Byzantine behavior, the protocol successfully detects and eliminates faulty vehicles, provided that more than two-thirds of the vehicles are non-faulty. This allows non-faulty vehicles to achieve secure harmonized speed and maintain safe platoon operations. As such, the protocol generalizes to secure other platooning operations, including splitting and merging, intersection negotiation, lane-changing, and others. The implications of this research are significant for the future of AV platooning, as it establishes BFT to enhance the safety, efficiency, and reliability of AV transportation, therefore paving the way for improved security and cooperative road ecosystems.

Acknowledgements

I wish to express profound appreciation to a number of individuals who have played a pivotal role in the successful completion of my thesis. First and foremost, I extend my heartfelt gratitude to my supervisor, Dr. Otman Basir. His unwavering support, expertise, and guidance throughout this research journey have been truly invaluable. The insightful feedback, constructive critique, and commitment to my academic growth that he provided have significantly molded the outcome of this thesis. Furthermore, I would like to convey my thanks to my professor at Qatar University, Dr. Ahmed Massoud. His unwavering dedication to research and academic advancement has left a deep impression on my academic voyage. His perceptive guidance and continuous motivation have enabled me to develop a more profound comprehension of the subject matter, greatly impacting the caliber of this thesis. I hold deep gratitude for the encouragement and direction I have gained from each of you. This thesis would have remained an unrealized endeavor without your collective contributions. Thank you for your belief in my abilities and for standing by me unfailingly during this academic pursuit.

Dedication

This thesis is dedicated to my beloved father, whose unwavering love, support, and guidance have been the driving force behind my academic pursuits. Your endless encouragement, belief in my abilities, and sacrifices have shaped me into the person I am today. Your unwavering faith in my potential and your constant presence in my life have provided me with the strength to overcome challenges and reach for my goals. I am forever grateful for your love and unwavering support.

To my family, friends, and loved ones, thank you for your encouragement, understanding, and support throughout this journey. Your belief in me and your words of encouragement have provided me with the strength and determination to overcome obstacles. Your presence in my life has made this journey all the more meaningful and memorable.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ACC Adaptive Cruise Control 7, 12, 15, 16

AI Artificial Intelligence 10, 17

AV Autonomous Vehicle iii, 1, 2, 4–12, 14, 28, 29, 41, 43, 49, 64, 65, 67

BFT Byzantine Fault Tolerance iii, viii, xi, 2, 3, 5, 29, 37, 39, 41, 43, 45, 49, 53, 62

BFT-ARM Byzantine Fault Tolerant Asynchronous Reliable Multicast 59, 60, 62

CAN Controller Area Network 23, 27

CAV Connected Autonomous Vehicle xi, 21, 22

DoS Denial of Service 22, 23, 25, 26

LiDAR Light Detection and Ranging 17, 27

PoS Proof-of-Stake 41

PoW Proof-of-Work 42

SAE Society of Automotive Engineers ix, 1, 7, 13–16

V2I Vehicle-to-Infrastructure 17, 22, 23, 26, 37

V2V Vehicle-to-Vehicle iii, ix, 2–4, 17, 19, 22, 23, 28, 37, 38, 45, 48, 57, 64

VANET Vehicular Ad hoc NETwork iii, ix, xi, 2–5, 17–22, 29, 36–39, 42, 43, 45, 59, 64, 65

Chapter 1

Introduction

1.1 Overview

A self-driving or AV uses technology to partially or entirely take on driving tasks in place of a human operator while navigating around potential road hazards and adapting to the flow of traffic [14]. AVs have emerged as a transformative technology in the transportation industry, leading to significant advancements in vehicle automation and communication technologies. One such example is enabling the realization of AV platooning—a concept where vehicles travel together synchronously, utilizing automation and inter-vehicle communication.

The SAE has introduced a widely-adopted classification system consisting of six levels that categorize AVs based on the level of human intervention required [15]. This classification system, also utilized by the US National Highway Traffic Safety Administration, ranges from SAE Level 0, which represents vehicles without any automation features and requires complete driver control, to SAE Level 5, which describes fully autonomous vehicles capable of performing safety-critical tasks without driver intervention. A Level 5 AV can operate under any weather conditions, at any time of day, on any type of road, and is majorly projected to be realized around 2030 [16, 17].

The emergence of AV platooning offers numerous benefits, including space-saving on congested roadways, enhanced safety, reduced energy consumption for transporting goods, lowered travel costs, and decreased greenhouse gas emissions. By reducing or eliminating the human factor in driving, a significantly safer driving environment can be created. According to a report by McKinsey & Company [18], this reduction could potentially result in a 90% decrease in crashes, saving an estimated $190 billion annually.

Due to the use of wireless communications and broadcasting in platooning, platoon communications are vulnerable to a wide range of cyberattacks, which makes ensuring the safety of AVs a critical challenge. Specifically, Byzantine behavior [1,2] poses challenges to consensus by transmitting incorrect or conflicting information to other vehicles in a platoon, thereby jeopardizing road safety, impeding the benefits of energy-efficient and smooth traffic flow, compromising system trust, and causing legal complexities [3,4]. In this light, this thesis delves into the challenges posed by Byzantine behavior within platoons and presents a robust solution using ConsenCar; a blockchain-based protocol for AV platoons. ConsenCar aims to address Byzantine faults in AV platoons in order to maintain reliable and secure operations within the platoon.

disrupting the organization and safe operation of the platoon, potentially leading to accidents or collisions. In this light, this thesis presents a solution to accurately detect and manage Byzantine behavior in platoons of AVs using blockchain technology, with the primary goal of securing the safety of human occupants and other individuals on the road. It is hypothesized that there exists a single vehicle compromising the security of the platoon through Byzantine behavior, which poses threats to integrity and human safety. Consequently, the platoon can fail to maintain safe operations, potentially posing threats to integrity and safety. To solve the consensus problem for such self-organizing system, the solution assumes partial synchronization, fixed timeouts, known participants, and unforgeable signatures to meet design requirements of agreement, integrity, termination, correctness, validity, and provability.

Introduced in this work is the ConsenCar protocol, which focuses on reliably detecting failures in systems where a single failure is deemed intolerable. To achieve this, the developed protocol leverages V2V communication over a VANET and implements BFT in platoon operations. The platoon management scheme employed by the proposed protocol distributes the responsibility of validating properties across all members of the platoon, thereby eliminating the leading vehicle as a single point of failure. By adopting this approach, ConsenCar that the consensus problem can be successfully resolved for the platoon, as long as over two-thirds of the vehicles in the platoon are non-faulty.

The blockchain structure of ConsenCar reveals that the algorithm can generalize to secure various platooning operations, such as splitting and merging, intersection negotiation, lane-changing, overtaking, and more. By modifying the algorithm to tailor to each operation, ConsenCar can provide secure platooning operations under Byzantine attacks, thereby ensuring the well-being of all road users.

1.2 Problem Statement

This thesis addresses the risks posed by Byzantine behavior in vehicular platoons, which can lead to disorganization and potential collisions or accidents, endangering human safety and compromising the integrity of the platoon. In particular, the research hypothesizes the presence of a single vehicle within a platoon engaging in Byzantine behavior, which could jeopardize security, safety and performance, in addition to any third-party communications with the leader.

1.3 Aim and Objectives

To overcome the various risks of Byzantine failure, this thesis proposes the development and adoption of ConsenCar; a distributed management system utilizing blockchain to secure vehicular communication through a consensus-based approach. By integrating BFT principles, the platooning system aims to detect and manage Byzantine failures effectively, provided that more than two-thirds of platooning vehicles are non-faulty. This ensures safe operation and improves its resistance against adversarial attacks.

Given the challenges posed by Byzantine behavior in vehicular platooning, this thesis aims to achieve several key objectives to enhance the safety and integrity of the platoon. The primary objectives are as follows:

1. Investigating the impact of Byzantine behavior on platoon safety, integrity, and performance, particularly in scenarios where a single faulty vehicle undermines the entire system.

2. Designing and implementing ConsenCar; the distributed consensus-based platoon management system, by leveraging blockchain and V2V communication over VANET.

3. Evaluating the effectiveness of the proposed system in tolerating Byzantine faults while minimizing overhead and message transmission delays.

4. Assessing the system's ability to identify and isolate faulty vehicles, thereby ensuring the platoon's safe and smooth operation.

1.4 Contributions

The contributions of this work are as follows:

- Comprehensive Literature Review: This work provides a comprehensive background and history into the development of AVs, from the 1500s until the present day. Further, the impact of AVs is explored on various industries, and security issues faced by AVs and AV platoons are detailed alongside open challenges in securing AVs.

- Decentralized Platoon Management: This work proposes a decentralized approach to platoon management. Unlike traditional leader-based management systems which rely on a single point of failure, the approach disperses the process of validating platoon properties among all members. This decentralized approach eliminates the vulnerability associated with a single leader and enhances the resilience and fault tolerance of the platoon.

- Enhanced Security and Safety: The developed ConsenCar protocol aims to ensure the security and safety of AV platoons by accurately detecting and managing Byzantine behavior through secure V2V communication. Utilizing unforgeable signed messages, it can effectively identify Byzantine vehicles through collaborative decision-making to maintain security and safety.

- Collaborative Consensus-Based Operation: The ConsenCar protocol utilizes consensus-based operation to ensure safe collaborative decision-making. Utilizing the shared ledger in blockchain, platoon members share information and collectively make decisions, thereby promoting integrity and efficient operation, even in the presence of Byzantine faults. In contrast to similar protocols, ConsenCar does not require expensive re-chaining or changing a suspected Proposer[1] in the event of a failure.

- Communication Overhead: As a crucial aspect of evaluating performance, the number of transmitted messages required for each consensus round is minimized. ConsenCar establishes secure operation under the limited bandwidth in VANETs which cannot tolerate high communication delays. Therefore, decision-making processes are performed promptly.

1.5 Structure

The thesis will be structured as follows:

[1] The term ”Proposer” is one of seven roles assigned to members in a platoon, and is defined in Chapter 4.

- Chapter 2 traces the history and development of AVs, explores the impact of AVs on various industries and discusses their challenges and limitations. Additionally, it reviews relevant literature on AVs and their security vulnerabilities, highlighting the need for robust security measures.

- Chapter 3 presents an extensive exposition on Byzantine cyberattacks, BFT and its applications in distributed systems. Moreover, it introduces blockchain technology and its key features and advantages of enhancing the security and reliability of AVs. It also proposes functional requirements for BFT platooning and delves into research gaps and challenges in BFT solutions for VANETs.

- Chapter 4 introduces the ConsenCar protocol; the proposed solution to establish secure harmonized speed under Byzantine cyberattacks in AV platoons. It describes the problem formulation, key components and mechanisms to ensure secure harmonized speed of the platoon as well as operational constraints.

- Chapter 5 describes the simulation set-up and methodology used to assess the effectiveness of the ConsenCar protocol. It analyzes simulation results and discusses the throughput of the protocol.

- Finally, Chapter 6 summarizes the main findings and implications of the research and identifies future research directions and areas for further investigation.

Chapter 2

Background & Literature Review

2.1 History of Autonomous Vehicles

Centuries before the invention of the automobile, the first design of a self-driving vehicle was introduced by Leonardo da Vinci [5]. Since then, AVs have majorly impacted the automobile industry, urban planning, traffic, insurance, labor market and other fields. Although the first design dates back to the 16^{th} century, the continuous technological progression is mainly classified in this section by decade to improve clarity and readability.

Table 2.1: Milestones in Autonomous Technology

1500s	First Prototype of the AV—Leonardo's Cart.
1925	Demonstration of a radio-controlled car—Francis Houdina.
1933	Prototype autopilot tested in a 21,000 km flight—Mechanical Mike.
1939	Futurama exhibit by Norman Bel Geddes at the New York World's Fair.
1948	Cruise Control invented—Ralph Teetor.
1961	First AV to embed a camera—Stanford Cart.
1986	Concept of platooning is introduced—Partners for Advanced Transit and Highways.
1992	First vehicle platooning experiments successfully conducted—Partners for Advanced Transit and Highways.
1995	Semi-autonomous vehicle completes a 98.2% autonomous cross-country journey—NavLab5.
1998	Toyota introduces Adaptive Cruise Control (ACC).
2003	Toyota Prius offers automatic reverse parallel-parking assistance.
2012	First license issued to a self-driving car in the United States—Google.
2014	SAE International publishes a 6-level classification system for on-road driving automation systems.
2015	Tesla introduces Autopilot feature in a software update.
2016	First self-driving taxi service in the world—nuTonomy Taxi.

1500s

The first prototype of the AV took place centuries prior to the first vehicle. In the 1500s, Leonardo da Vinci designed and built a cart that was capable of moving independently without needing to be pushed or pulled. As shown in Fig. 2.1, power was provided through high-tension springs, whereas steering was set in advance such that the cart moved in a predefined route. As a distant precursor to the car, Leonardo's Cart [5] is often referred to as the world's first robot [19].

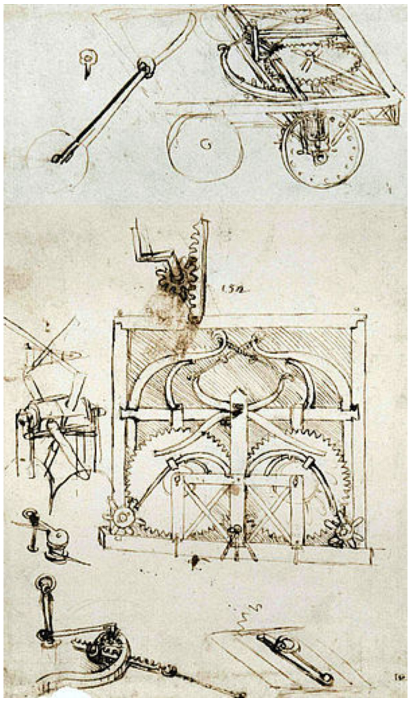


Figure 2.1: Original Design of Leonardo's Cart; the First Prototype of the AV (1500s) [5]

1920—1939

In the absence of a driver at the steering wheel, a radio-controlled car was demonstrated in 1925 by Francis Houdina through the streets of Manhattan. According to a New York Times report, the car was capable of starting its engine, shifting gears, and sounding its horn "as if a phantom hand were at the wheel [20]." Equipped with a transmitting antenna, the car received radio impulses from an operator present in another car that followed it.

The received signals were then sent from the antenna to circuit-breakers, which operated small motors that controlled compact electric motors responsible for guiding every motion of the vehicle. Despite offering a glimpse into the future, the operator lost control of the vehicle twice during the trip, resulting in a collision with another car which interrupted its trajectory.

Two events marked the progression of AV technology in the 1930s. Firstly, lengthy travel durations motivated the creation of long-range aircraft autopilot systems. Considering that, Sperry Gyroscope Co. designed a prototype autopilot known as Mechanical Mike. In 1933, Wiley Post used the prototype for a 21,000 km global flight, during which gyroscopes monitored the aircraft's heading and synchronized with the controls to guarantee precise orientation [21]. In the meantime, gyroscopes remain a fundamental component of inertial navigation in AV technology [22, 23].

Secondly, an early representation of automated guided cars was Norman Bel Geddes' Futurama Exhibit sponsored by General Motors at the 1939 World's Fair. Presented at the exhibit was his self-driving electric car model, controlled through radio-guided electromagnetic fields, and maneuvered using roadway-embedded magnetized metal spikes. After his model came to life in 1958 [24], Bel Geddes later promoted advancements in highway design and transportation, foreshadowed the Interstate Highway System, and debated that the process of driving should eliminate the presence of humans in his book Magic Motorways [25], published in 1940[1].

1940—1959

While riding a car driven by his lawyer, Ralph Teetor was inspired to invent what is presently known as Cruise Control. The blind inventor found a motive in the irritating rocking motion of the car to create a speed-control device. Illustrated in Fig. 2.2 [6], the first "Speedostat"[2] included a speed selector integrated into the dashboard, which was linked to a mechanism in the engine compartment driven by the drive shaft. As the speed chosen by the driver approached, the governor mechanism overcame the resistance of a spring, initiating a vacuum-driven piston capable of applying pressure to the gas pedal. The design was further modified by incorporating a speed lock mechanism: an electromagnetic motor maintaining dialed-in speed until the brake pedal was tapped. Receiving his first patent on a speed control device in 1945, his device was fully developed by 1948, and

[1] Though published in 1940, predictions made by the author had been arbitrarily dated ahead to 1960; 20 years from the year of publication.

[2] Early names for Teetor's invention included Controlmatic, Touchomatic, Pressomatic, and Speedostat before people settled on Cruise Control.

became commercially available in 1958 on Chrysler's Imperial, New Yorker and Winsdor models [26].

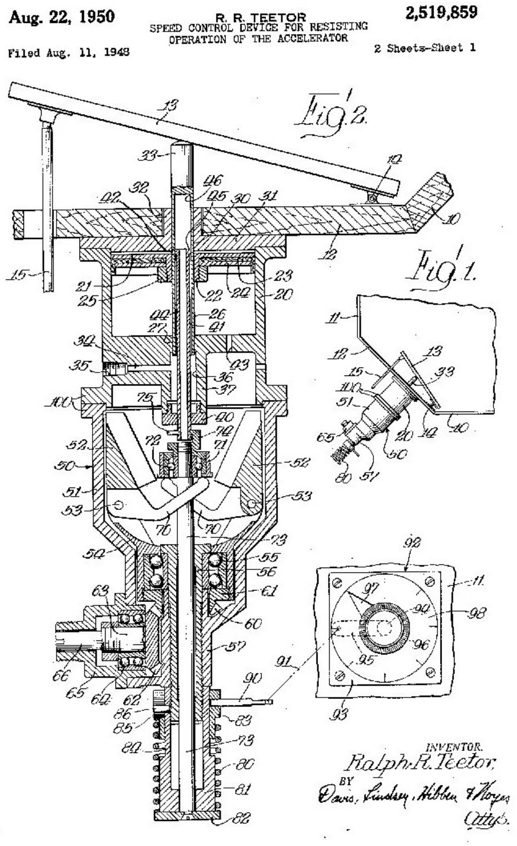


Figure 2.2: Speedostat; a Speed Control Device to Resist Operation of the Accelerator [6]

1960—1979

During the peak of the Space Race in 1961, scientists began exploring the concept of landing vehicles on the moon. The notion of a lunar rover controlled from a distance was suggested by James Adams at Stanford University to support his research on remote vehicle control through video information. Fitted with a television camera and capable of autonomously detecting and following a line on the ground, the Stanford Cart shown in Fig. 2.3 [7] was the first to embed a camera—a vital element in AVs today [27]. In highly controlled experiments, the cart relied on its camera and an early version of Artificial Intelligence (AI) to follow white lines and avoid obstacles placed in its path by the 1970s [28]. Through controllability tests of various combinations of communication delay and speed, it was found that steering command delays increased the likelihood of over-steering and losing control of the vehicle. Also, Adams explained that with a communication delay of about 2.5 s, which corresponds to the round-trip to the moon, the operator was incapable of reliably controlling the vehicle if it travels faster than about 0.3 km/h [29].

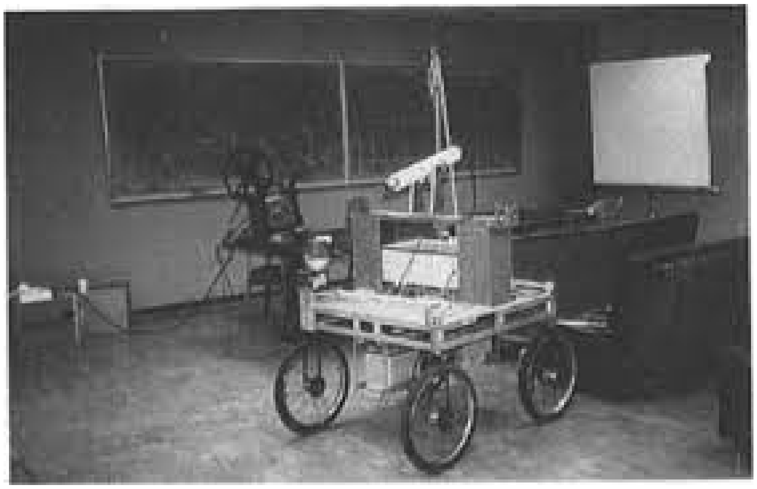


Figure 2.3: The Stanford Cart [7]

Early automation has proven the technical feasibility of automated driving, but their high cost and low demand for radio and wire-controlled vehicles gradually diminished their presence in modern AVs. In particular, the installment cost for guided-vehicle highways was estimated to be as high as $200,000 per lane-mile. If fully installed, such road upgrade could have added more than 40% to the cost of the American Interstate Highway System; already the largest public works project in history [30].

1980—1999

The Partners for Advanced Transit and Highways project introduced the concept of a vehicular platoon in 1986 [31]. The project demonstrated the advantages of the new technology, as there was a notion that platooning would improve road capacity, trip delays, accidents, vehicle breakdowns and energy consumption. By 1992, initial tests of vehicle platooning were successfully completed in San Diego, which demonstrated a four-vehicle platoon capability [32].

An ACC system was initially introduced by Toyota in 1998 [33]. The system works by detecting other vehicles in its lane, therefore automatically accelerating or decelerating depending on the distance between the two vehicles.

2000—2009

The 21st century marked many milestones at which the focus moved from academic research to industrial development. Building on previous technology, this century witnessed major improvements in safe practices and effective performance in complex and unpredictable human environments. As the main goal is to establish safer operation and improved traffic flow, the commuting experience is transformed, sharply reducing the human element from high-risk environments and streamlining industries.

As self-parking systems began to emerge, they demonstrated the capability of handling relatively challenging conditions, such as parallel parking in tight spaces. In 2003, the hybrid Toyota Prius offered automatic reverse parallel-parking assistance. Using front and rear cameras, it estimated the dimensions of the parking space and decided whether it can fit within it or not, whereas front and rear sensors estimated the proximity of surrounding vehicles [34]. By 2006, it featured parking sensors and could detect minute objects. Since then, competitors started promoting similar advancements in their vehicles. For instance, Ford Motor Company launched Active Park Assist in 2009, whereas BMW followed a year later.

2010—2019

Google privately began developing AVs under its Self-Driving Car Project in 2009. By the end of 2010, Google Cars had logged over 225,308 km on city streets and highways. In May 2012, the first licensed test of a self-driving car in the United States took place [8]. As shown in Fig. 2.4, a modified Toyota Prius was sent to the Nevada Department of Motor Vehicles [35]. Although engineers overtook control twice during the drive, the car passed the test. Consequently, Google announced in May 2014 their plans to build 100 self-driving car prototypes as a first step to providing safe commuting technology to the public.

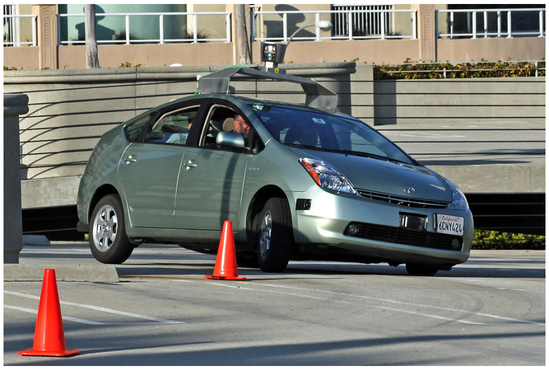


Figure 2.4: Google Car Tested in the Nevada Department of Motor Vehicles [8]

SAE International published J3016 in 2014, a 6-level classification system for on-road automation systems ranging from fully manual to fully automated [15]. Since its release, manufacturers started adopting the Ground Vehicle Standard as a taxonomy reference for vehicle automation. For instance, Tesla reported in 2016 that their vehicles were supplied with the essential machinery to enable full autonomy (SAE Level 5) safely. These include eight surrounding cameras, twelve ultrasonic sensors as well as a forward-facing radar.

Tesla Autopilot was introduced in late 2015 as a semi-autonomous feature. In highways and freeways, the feature allowed hands-free control, and was delivered as an overnight software update for electric vehicles [36, 37]. Due to its existing customer base, Tesla gathers more data from its Autopilot system in a single day than what Google has gathered since 2009 from its autonomous driving program. Moreover, despite claims that accidents will be reduced by 50% with Autopilot on, the company repeatedly stated that the feature was not autonomous, as constant driver engagement is required. In June 2016, the National Highway Traffic Safety Administration in the United States started investigating a fatal collision involving Autopilot with a semi-truck. With only a camera and radar system, Tesla later hinted that the system may have been unable to distinguish the white side of the truck from the bright sky. Two and a half months later, another fatal crash was reported as a result of colliding into a slow-moving street sweeper on a highway [38]. Ultimately, the crash uncovered a deadly risk Tesla was running by marketing its Autopilot as nearly autonomous when software malfunctions are yet to be resolved. Yet, it must be emphasized that the driver was not adequately fulfilling their co-pilot responsibility of monitoring the system and intervening when needed. Consequently, this reveals uncertainties about shared human-machine control as well as the implementation of fully autonomous machines that do not rely on human control. However, the National Highway Traffic Safety Administration reported in 2015 that 94% of traffic accidents are results of human error [39]. Hence, removing the human element to create a safer driving environment mandates further advancements in autonomous technology.

In August 2016, nuTonomy introduced the world's first self-driving taxi service [9, 40], showcased in Fig. 2.5 [9], within Singapore. Though the presence of an engineer was a required safety measure, select members of the public were able to book the vehicles through a smartphone application.

Figure 2.5: nuTonomy Taxi [9]

Moreover, claims that autonomous technology is a safer alternative to traditional driving were challenged in 2018 when an Uber Volvo XC90 prototype killed 49-year-old Elaine Herzberg in Arizona. Although she was jaywalking and wearing dark clothes at night with no reflective gear, the driver was charged with negligent homicide in September 2020 due to inattentiveness [41]. Despite the incident, the commercial production of AVs has not ceased. Instead, the state saw the launch of Waymo One, a commercial robotaxi by Waymo, in the same year of the accident [42]. The fully public and fully autonomous service is available for commuters in Phoenix to request through the Waymo One application.

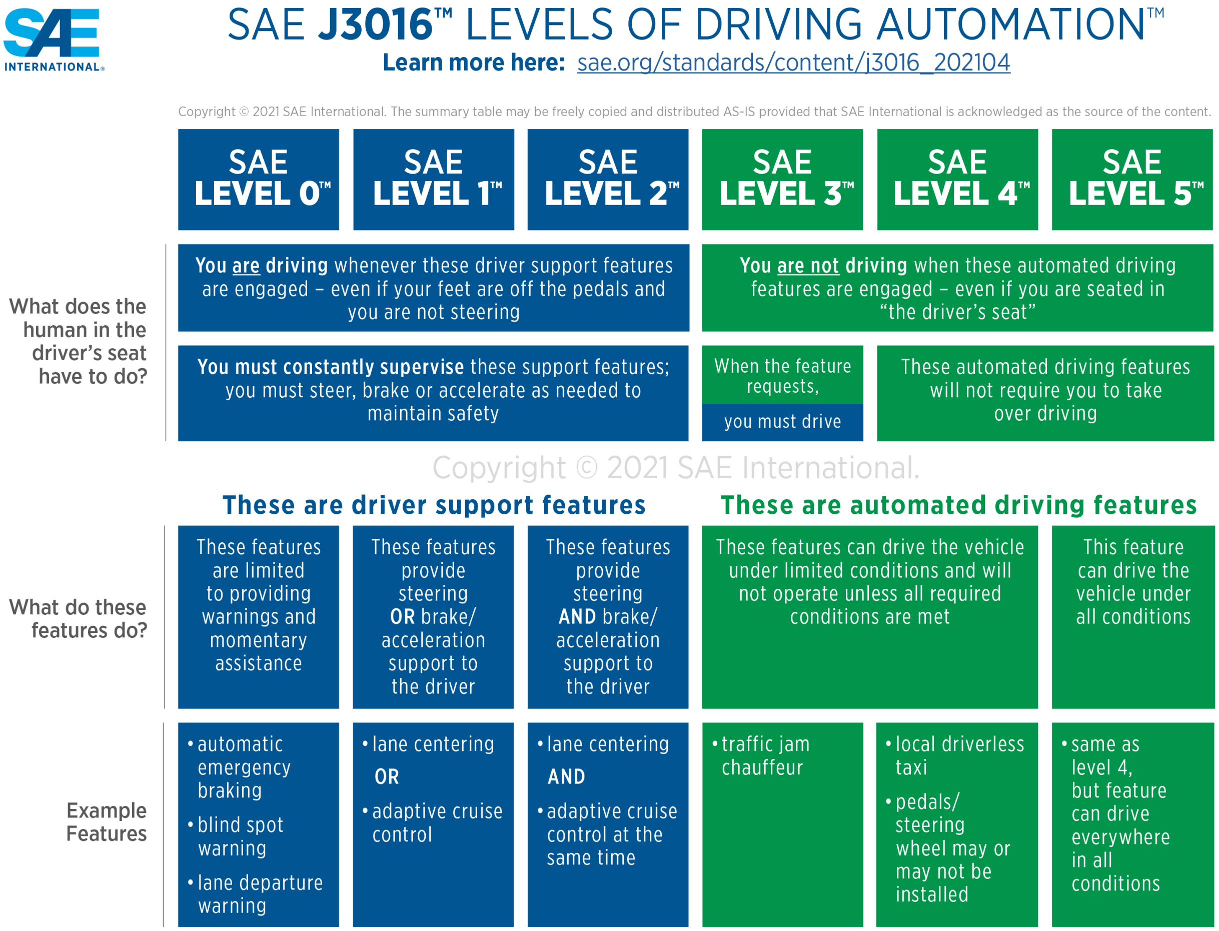
In 2019, the United States passed laws permitting AVs in twenty-nine states [43], whereas the European Union had defined Regulation (EU) 2019/2144 to be effective starting 2022 for automated and fully automated vehicles [44].

AVs are expected to bring about different advantages, with road safety being the most important. That is, accidents resulting from impaired driving are projected to decrease significantly. Nonetheless, safe driving mandates more than the mere knowledge of human presence behind the wheel, for how they are likely to act and respond to stimuli must also be known. Even amongst breakthrough innovations, it is to be noted that SAE Level 5, denoting full automation, is yet to be reached. Consequently, users must understand the perceived risk of overtrusting vehicles which are yet to achieve full autonomy.

2.2 Society of Automotive Engineers (SAE) Levels of Driving Automation

Autonomous driving capability is a spectrum of functionality that encompasses basic driver support to complete driving autonomy. Originally published in 2014, the SAE J3016 chart defines six levels of autonomous driving capability. As shown in Fig. 2.6, the first three levels (Levels 0-2) are defined as ”Driver Support Systems,” whereas Levels 3-5 describe actual ”Automated Driving Systems.”

As of 2023, no vehicle has reached SAE Level 5 autonomy, and many are convinced that this level is far too early from becoming a reality. Nevertheless, the chart reflects that remote driving assistance has become a crucial advancement in the journey towards achieving Level 5 autonomy. In addition, it clarifies driver support features, which include lane centering, brake/acceleration support and ACCs. For Levels 0-2, a human is always driving, as automated driving is not a function until Level 3 is reached.



SAE **J3016**™ LEVELS OF DRIVING AUTOMATION™

Learn more here: sae.org/standards/content/j3016_202104

	SAE LEVEL 0™	SAE LEVEL 1™	SAE LEVEL 2™	SAE LEVEL 3™	SAE LEVEL 4™	SAE LEVEL 5™
What does the human in the driver's seat have to do?	**You are driving** whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			**You are not driving** when these automated driving features are engaged – even if you are seated in "the driver's seat"		
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	
	These are driver support features			**These are automated driving features**		
What do these features do?	These features are limited to providing warnings and momentary assistance	These features provide steering **OR** brake/ acceleration support to the driver	These features provide steering **AND** brake/ acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met		This feature can drive the vehicle under all conditions
Example Features	• automatic emergency braking • blind spot warning • lane departure warning	• lane centering **OR** • adaptive cruise control	• lane centering **AND** • adaptive cruise control at the same time	• traffic jam chauffeur	• local driverless taxi • pedals/ steering wheel may or may not be installed	• same as level 4, but feature can drive everywhere in all conditions

Figure 2.6: SAE J3016 Levels of Driving Automation [10]

The majority of automotive manufacturers have integrated and commercialized various degrees of SAE Level 0-2 autonomy, such as "Intelligent Cruise Control," "Smart Cruise Control," "Intelligent Lane Control," "Lane-Keeping Assist System" or "ACC." These features assist human drivers in maintaining safe control over their vehicles, or reduce reaction time to a pending accident. Currently, many manufacturers deliver such functionality as a standard instead of an add-on.

Reaching Level 5 autonomy requires the vehicle to be capable of driving at any time of the day under any weather condition on every road type. The latter two requirements demonstrate the primary challenges for self-driving technology. That is, weather conditions reduce or even eliminate the effectiveness of on-board sensors. For instance, falling snow gathering on the front of a Light Detection and Ranging (LiDAR) unit makes it incapable of returning 3D data for the perception engine. Similarly, navigating roads in the absence of markers imposes a challenge on the on-board AI.

2.3 Vehicular Ad-hoc NETwork (VANET)

2.3.1 Definition

The wireless ad-hoc network is the parent field of all ad-hoc networks. As presented in Fig. 2.7, VANET is a sibling of Mobile Ad-hoc Network, which independently organizes its communication system without relying on other infrastructure. Given their easy and basic communication method, mobile ad-hoc networks are most commonly used in military applications. Likewise, VANET is similar along with few differences to accommodate unique vehicle characteristics. VANET comprises mobile nodes and road-side units, as illustrated in Fig. 2.8. Mobile nodes (or on-board units) are vehicle-embedded sensors used for signal processing and data sharing to and from road-side units. Road-side units are fixed gateway units that enable communications between mobile nodes and the servers or Internet. The most important service provided by VANET is road safety, as it reduces accidents by sharing data through the Internet. This is accomplished through its two main types of communication models; V2V and Vehicle-to-Infrastructure (V2I) [45].

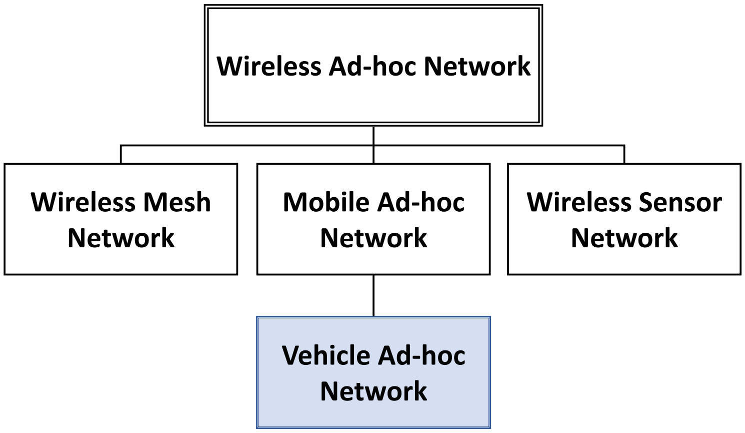


Figure 2.7: Classification of Ad-hoc Networks

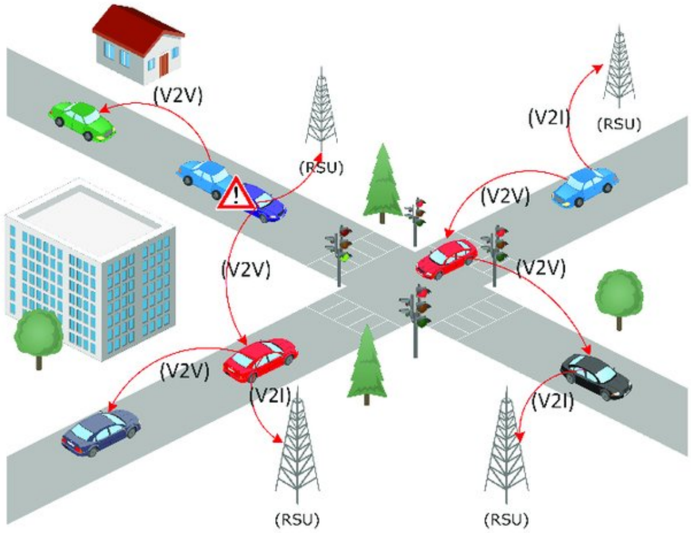


Figure 2.8: VANET Communication Architecture [11]

2.3.2 Communication Topologies

Due to its linear structure, communication and sensing capabilities are described by the range in the forward and backward direction for each vehicle within the platoon. To demonstrate, Fig. 2.9 illustrates three suitable communication topologies:

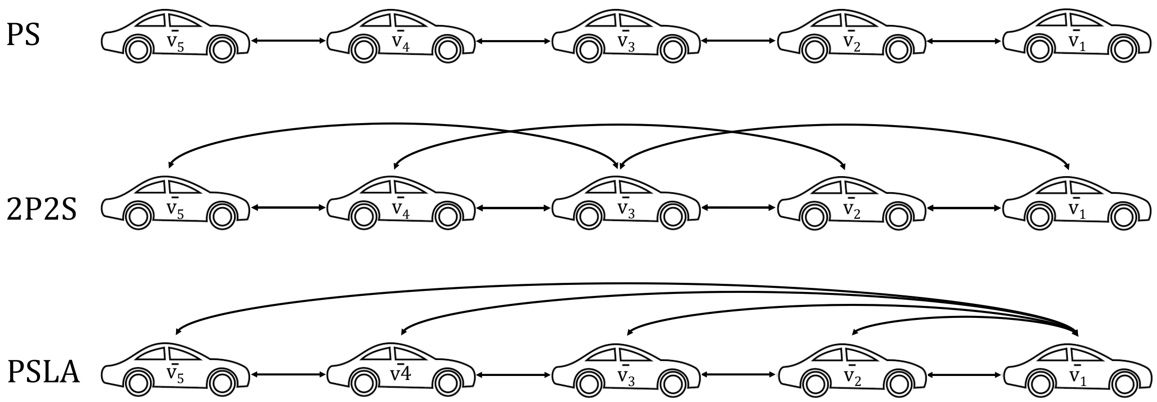


Figure 2.9: Platoon Communication Topologies; where Arrows Indicate Direct V2V Communication

- PS: Predecessor - Successor. Sensing topology is crucial for the verification of sent claim of a vehicle, including its identity or physical presence. For simplicity, sensing the license plate of predecessor and successor vehicles only is considered to authenticate a vehicle.
- 2P2S: Two-Predecessor - Two-Sucessor.
- PSLA: Predecessor - Successor - Leader to All.

2.3.3 Applications

Many research activities targeted the development of VANET applications and usage models for [46–48]. Since more people are spending their time on the road, the latest developed VANET-based applications related include online file sharing, real-time video updates as well as Internet-based entertainment through road-side units or V2V communications. In addition, these applications are classified as non-safety and safety applications [49]

Non-safety Applications

Non-safety applications provide comfort and traffic efficiency for commuters. Classifiable as value-added services, these applications include automatic toll collection, site-based services such as target localization, and Internet connectivity facility.

Safety Applications

Safety applications protect human lives on the road by delivering safety-related information on time to receivers in order to prevent accidents. These applications are shown in Table 2.2, whereas their corresponding message types are described as follows:

1. Information Messages: comprise messages related to construction zones while driving on the freeway, notifications about toll collection points, and messages indicating speed limits, among other messages.

2. Assistance Messages: assist the driver during their journey. Include lane switching, navigation, and cooperative collision avoidance information which are the most critical, as they warn the driver to slow down in order to avoid uncertain conditions.

3. Warning Messages: include traffic signal information ahead, toll points or bad road condition warnings.

Table 2.2: Safety Applications of VANET [12]

Application	Description
Traffic signal violation	Alerts vehicles of dangerous situations.
Intersection collision warning	Sends information about intersection points.
Turn assistance	Helps the driver turn the vehicle.
Blind spot warning	Warns about the existence of a vehicle located in an area not visible to the driver.
Pedestrian crossing information	Sends pedestrian crossing information in a path.
Lane changes warning	Ensures a clear lane for entry.
Forward collision warning	Warns the driver of a slower vehicle in the front.
Do-no-pass warning	Warns the driver of overtaking safety.
Post-crash	Alerts the driver that a crash has occurred.
Emergency service vehicle	Clears path for emergency vehicles.
Curve speed	Alerts the driver of road curves ahead.
Wrong way	Alerts the driver of movement in the wrong way.
Work zone	Alerts the driver of a work zone area ahead.

Furthermore, communication security in VANET is a vital concern, as safety-critical applications directly influence human lives. Compared to other wireless communications, ad-hoc networks experience a higher number of security issues due to their dynamic nature. Since there is no pre-existing infrastructure for ad-hoc networks, controlling their security becomes challenging. Similar to all computer systems, VANET is susceptible to data security issues, including confidentiality, integrity, availability, and authenticity.

2.4 Security Threats to Vehicular Platooning

Platoon security includes physical security, cybersecurity and vehicle privacy [50]. If any security elements are compromised, attackers can obtain personal or financial gain. For instance, a vehicle that is left with an unattended key compromises physical security. Moreover, unencrypted communication between two platooning vehicles targets cybersecurity, as it exposes transmitted information to sniffing attacks. In this section, only wireless communication cybersecurity for safety purposes is considered.

Network platoon security is of extreme importance due to the safety-critical information being shared in the platoon. Thus, the detection and mitigation of attacks is important to secure the system of vehicles. In the scope of vehicle network security, research identified attacks both VANETs and CAVs [51, 52], [53–55]. These attacks were further categorized into five domains: confidentiality, integrity, availability, authenticity and non-repudiation [51, 56, 57]. Further, trust management has been largely explored in the literature [53, 58], in which properties of most significance include decentralization, real-time constraints, information sparsity, scalability, privacy and robustness. Although these trust models form trusted node clusters of nodes which communicate well in areas of dense populations, they are volatile to low node concentrations. Nevertheless, the way trust can be incorporated into platoons is a concept that is notably absent from existing literature.

Vehicular platoon communication is exposed to different cybersecurity threats. Direct threats aim to disrupt integrity to decrease efficiency and passenger comfort. In contrast, indirect threats aim to separate or prevent platoon formation. Other passive threats may aim to steal user, vehicle and load information. Table 2.3 addresses state-of-the-art on VANET, CAV and platoon cybersecurity, highlighting potential threats posed on their networks.

Table 2.3: State-of-the-Art Review on VANET, CAV and Platoon Cybersecurity

Survey	Key Points	Discussed Attacks

[51]	• Security and privacy attacks and mechanisms in VANET. • Groups attack together according to broken security requirements or attributes; confidentiality, integrity, availability, authenticity, and non-repudiation.	Denial of Service (DoS), jamming, malware, broadcast tampering, black/grey hole, greedy behavior, sniffing, spamming, traffic analysis, Sybil, tunneling, GPS spoofing, free-riding, modification, masquerade, replay and repudiation.
[52]	• Analysis of truck platoon attack surfaces. • Summary of corresponding defense systems along with defense gaps for identified attacks. • Research issues to design more resilient platoons.	Illusion, masquerade, repudiation, man-in-the-middle, GPS spoofing, session hijacking, covert channel, worm/black hole, packet dropping, location disclosure, DoS, Sybil, jamming, sniffing and modification.
[53]	• Security analysis of Intelligent Transportation Systems. • In-depth risk analysis. • Recommendations for securing testbeds.	Sensor spoofing and jamming, information theft, sniffing, malware on vehicles and infrastructure.
[54]	• VANET security for trust management; discussion and highlights of recent open research questions. • REPLACE [59]; a trust-based platoon service recommendation scheme.	Does not specify attacks, but rather a wide range of trust management methods.
[55]	• Sensor, controller, and in-vehicle network security issues and solutions. • Connectivity technologies: applications and security issues. • Impact analysis of cyberattacks on CAVs and solutions to enhance their security.	Sensor spoofing and jamming, saturation, cancellation, replay, sniffing, modification, DoS, traffic analysis and jamming.

2.4.1 Confidentiality

Information transmitted within platooning vehicles can be private and confidential. Network sniffing breaks the confidentiality for V2V and V2I communications, as an attacker is capable of listening to exchanged information between the leading vehicle and members [60]. For example, an attacker can masquerade as a legitimate network user through a stolen or forged ID to gain legitimate access to the network, therefore affecting road-side units, trusted authority and platoon-enabled vehicles.

2.4.2 Integrity

Masquerade attacks jeopardize the integrity of leaders, members, road-side units, trusted authorities and other assets. Replay attacks jeopardize the integrity of road-side units, leaders, members and joiners/leavers by re-transmitting past messages within the network. In addition, modification attacks target members and road-side units by transmitting altered messages to vehicles [61]. For example, members may falsely change their speeds, resulting in fatal crashes.

2.4.3 Availability

Availability is a significant security element since critical information is communicated with members, such as the leader's speed and location. Compromising such elements results in disabling platooning services. Jamming attacks aim to disrupt V2V and V2I connections by injecting random noise into communication frequencies. DoS attacks target join/leavers and road-side units by flooding the network with requests that cannot be cleared quickly enough. Malware can compromise availability in different ways, one of which is by targeting platoon-enabled vehicles, road-side units and trusted authorities preventing legitimate service usage.

2.4.4 Authentication

GPS and sensor spoofing may target vehicles or road-side units. Spoofed GPS provides inaccurate location information by overpowering the authentic signal [62]. Sensors can be spoofed in different ways, such as by using malware to alter their outputs, or by exploiting weak points to access the Controller Area Network (CAN) bus. In many cases, their design is very simple and contains no security features, hence exposing them to attackers. On the other hand, Sybil attacks [63, 64] compromise authenticity by targeting the leader, members and road-side units from within the network. Thus, the leader and road-side units cannot differentiate between real and ghost vehicles. Although authenticated nodes are assumed to have established trustworthiness, they do not guarantee the legitimacy of actions taken by those nodes.

2.5 Attacks on Platoon Communications

Table 2.4 summarizes threats to platoons along with compromised security elements, as found in the literature.

Table 2.4: Threats to Platoons as Found in the Literature

Attack	Compromised Element	Attack Summary
Sniffing [60,65]	Confidentiality & privacy	Accesses transmitted information through an unsecured network within the platoon. Leads to data theft and privacy violations, and can be the root cause of other succeeding attacks.
Masquerade [65,66]	Integrity	Gains unauthorized network access through legitimate access identification by posing as an authorized vehicle. Leads to destabilization, false representation and reputation damage.
Replay [67–70]	Integrity	Replays old messages into the network. Leads to destabilization due to the reception of conflicting information.
Modification [71,72]	Integrity & availability	Alters transmitted data before forwarding it to recipients, such as by sending fake maneuver requests to platoon members. Leads to destabilization and can eliminate the availability of original messages.

Jamming [67, 73]	Availability	Prevents all local area communications on platoon frequencies. Leads to the disbandment of platoon members due to communication interference.
DoS [74, 75]	Availability	Floods the network with traffic or sends information which triggers a crash, making the shared network too busy to process legitimate requests, and therefore inaccessible to authorized users. Disabled communication disrupts formed platoons and prevents users from joining or creating platoons.
Malware [76, 77]	Availability	Prevents platooning capabilities, and can enable performing other attacks, including data theft, sensor spoofing and DoS, thereby compromising more security elements.
Spoofing & jamming sensors [62, 70, 78]	Authentication & availability	Inject malware or directly attack sensors; spoofing targets the physical layer to cause certain behaviors in the firmware which lead to false sensing, whereas jamming leads to blocking sensor operation.
Sybil [63, 64]	Authentication	Creates and uses ghost network vehicles to defeat trust, such as by attempting to join the platoon [63] or disrupting its density through corrupting inter-vehicle distances [64]. Leads to destabilization and prevents authentic members from joining.

2.5.1 Security Mechanisms

A diverse array of mechanisms exists to secure wireless vehicular platoon communications from a range of attacks. Table 2.5 introduces some security mechanisms alongside corresponding targeted attacks.

Table 2.5: Security Mechanisms and Open Challenges for Mitigating Attacks on Platoons

Security Mechanism	Targeted Attack	Open Challenge
Private & public keys	Sniffing, replay, modification.	Large-scale testing to assess the effectiveness against the cost of current key creation and distribution methods.
Road-side units	Masquerade, modification	More insight into road-side unit network security, including the detection of malicious road-side units is needed.
Control algorithms	Replay, modification, DoS, sybil	Finding the most efficient location for deployment and algorithm usage.
Hybrid communications	Replay, modification, jamming, sybil.	More research into V2I visual light and wireless radio communications is required.
On-board system security	Malware, spoofing & jamming sensors	Optimizing the deployment of security measures without compensating response.

2.5.2 Private and Public Keys

Public and private keys are used to encrypt and decrypt data, respectively, by encoding messages with predetermined algorithms which further enhance security and prevent replay attacks by adding signatures and timestamps to messages. Public key infrastructure utilizes a single key to secure a network. Available to all members, the key may periodically change to prevent external threat actors from network sniffing, data theft and other attacks. Though changing periodically, such keys may still be obtained by threat actors. On the other hand, private keys prevent modification, DoS [79] and Sybil attacks if connected to secure user IDs. Nonetheless, this mechanism becomes challenging when members are required to exchange keys during the unknown presence of an attacker. In platooning, a proposed solution is employing quantized fading channel randomness [80], which utilizes multi-path fading to rapidly generate identical private keys without necessitating key transmission. Since the pathway of the sniffer differs from that of an authorized user, the system becomes robust against sniffing, hence protecting the private key from unauthorized access.

To enable the aforementioned mechanism, it is required to implement additional hardware to platoon vehicles. Alternatively, road-side units can participate in private key exchange between members, though the installment of more infrastructure is required. Ultimately, key usage must be easy, cost-effective to distribute securely, and must maintain security unless accessed by an attacker [81].

2.5.3 Road-Side Units

As an adjacent infrastructure of the network, road-side units coordinate platoons as well as private and public keys by connecting platooning vehicles to road users and platoon services providers. In addition, they are capable of distributing secret keys in case direct communication is desired. Not only do these units establish improved connectivity, but they can also monitor driving behavior within the platoon, which can facilitate the detection of many attacks.

Given their advantages, road-side units remain of limited authority, for their primary use is secret key distribution to authorized users. Although this improves control over key acquisition and updating, it remains a challenge to efficiently detect and eliminate faulty units without risking damage to the network. Also, another challenge arises when handling network areas of low road-side unit density, as platoons cannot rely on them to update from a trusted authority.

2.5.4 Control Algorithms

To mitigate attacks on platoons, control algorithms can be used to detect abnormal driver behavior. Effective against replay, modification and Sybil attacks, this mechanism can detect risky behavior and communication caused by these attacks by continuously communicating with on-board sensors [82]. Further, such algorithms are applicable on the CAN bus to protect against sensor spoofing. Similar to private keys, control algorithms are beneficial in the sense that they can be easily updated at low costs.

To identify legitimate message sources, these algorithms may mandate source code authentication through encryption techniques, which can significantly increase platoon security against the injection of false data into communication channels [60, 67, 68]. Furthermore, vehicular platoon disruption attacks interfere with the natural movements of a platoon. As a countermeasure, attack detection algorithms are used to periodically monitor positions from multiple sources, including LiDAR systems and/or GPS sensor data, to ensure the correctness of each member's positional information. All in all, the real-time processing of information requires optimizing algorithm usage within platoons without compromising performance, to ensure safety and stability.

2.5.5 Hybrid Communication

The use of radio waves and visible light prevents jamming attacks. Motivated by this, hybrid systems are capable of preventing jamming and modification attacks by utilizing an additional channel to confirm the reception of information. Despite that the small inter-vehicle spacing allows the use of visible light communications, the possibility of external light interference still remains, causing the communication to be blocked. As such, combining visible light and IEEE 802.11p can establish secure visible light communications in platoons. That is, the hybrid communication pattern must receive both transmissions before performing any actions. To demonstrate, in the case of wireless communication jamming on 802.11p, the system switches to utilize visible light only, which relies on multiple communication types, until it re-establishes a secure connection.

2.5.6 On-Board System Security

Sensor spoofing and jamming attack prevention is feasible through the employment of multiple sensors to detect and report possible attacks. Lastly, malware-infected files may be difficult to detect, but may be prevented in several ways, including through the installment of firewalls and antivirus software on the on-board computer. Such risk may also be mitigated by only allowing authorized components to communicate with specific necessary nodes. Altogether, additional research regarding the adequate placement and deployment of such algorithms is necessary to maximize their effectiveness.

To summarize, this chapter encompasses a literature review addressing diverse aspects of AVs. It initiates with a historical account of AVs, tracing back to Leonardo da Vinci's 16^{th}-century design, moving forward to ramifications across sectors like the automobile industry, urban planning, traffic, insurance, and the labor market. Noteworthy advancements in AV technology are highlighted in this chapter, alongside an examination of trust management in AVs. Further, the spectrum of AV applications is explored, spanning safety and non-safety domains, while underscoring the significance of communication security within V2V networks. It also touches on the challenges and security strategies linked to on-board system security. In essence, a comprehensive overview of fundamental facets and open challenges within the realm of AVs is presented. In the next chapter, an in-depth exploration is offered concerning Byzantine cyberattacks, BFT and their utility within distributed systems. Additionally, an introduction is provided to blockchain technology, elucidating its essential attributes and the benefits it offers in enhancing the security, safety and reliability of AVs. Furthermore, the next chapter puts forward operational prerequisites for BFT platooning, while extensively examining unaddressed research areas and complexities associated with implementing such solutions for VANETs.

Chapter 3

The Byzantine Car

3.1 The Byzantine Generals' Problem

3.1.1 Definition

A reliable system must effectively manage malfunctioning components that convey contradictory information to its various segments. The challenge of dealing with such problem is conceptualized as the Byzantine Generals' Problem [1]. In this scenario, multiple divisions of a Byzantine army are camped with their troops around an enemy city, each led by its own general. Communication between generals is solely possible through messengers. Their task is to collaboratively devise a battle strategy based on their collective observations of the enemy. Nevertheless, some generals could be disloyal and aim to hinder the loyal ones from reaching a consensus. The study in Lamport's work demonstrates that the resolution of this problem, merely through oral messages, is feasible if and only if over two-thirds of those generals remain loyal. This implies that a lone traitor has the capacity to disrupt the decision-making of two faithful generals [1]. Nonetheless, other approaches to solve this problem for any number of generals and potential traitors will be explored in this chapter.

To guarantee the prevention of such behavior, the generals must follow an algorithm which:

1. Guarantees unanimous agreement among all loyal generals regarding the same course of action—all loyal generals must follow the adopted algorithm, unlike the traitors who may not do so. This condition must be guaranteed regardless of the traitorous generals' actions.

2. Ensures that a few traitorous generals are incapable of causing the adoption of a bad plan by loyal generals—not only must loyal generals agree on a plan, but the plan must also be reasonable.

Since Condition 2 is difficult to formalize, the generals' decision-making process is analyzed. Let $v(i)$ be the information in which the i^{th} general conveys. Every general follows a strategy to combine information $v(1), v(2), \ldots v(n)$ into one plan of action, where n describes the number of generals. Condition 1 can be satisfied by using the same method to combine information for all generals, whereas Condition 2 can be satisfied by adopting a robust method. If the loyal generals are nearly evenly split among various decision options, a minor number of traitors can influence the final decision, making it infeasible to label either decision as bad. Since traitorous generals transmit conflicting values to different generals, satisfying Condition 1 requires satisfying the following conditions:

(a) All loyal generals receive identical values $v(1), v(2), \ldots\ v(n)$—Condition 1 suggests that each general cannot automatically rely on $v(i)$ received directly from the i^{th} general due to the possibility of a disloyal i^{th} general conveying different values to various generals. This circumstance could lead to the adoption of a value $v(i)$ that deviates from the original, even if the i^{th} general is loyal. Therefore, a stipulation for every i is:

(b) In the event the i^{th} general is loyal, every loya; general must employ $v(i)$ as $v(i)$.

Based on the conditions above, the problem can be restricted to how one general communicates their value to other generals. This will be described in terms of an order sent to lieutenants of a commanding general, obtaining the Byzantine Generals' Problem:
Byzantine Generals' Problem—a commander must send an order to their $n-1$ lieutenants such that

IC1. The same order is obeyed by all loyal lieutenants.

IC2. Each loyal lieutenant obeys the commanding general's order, provided the commanding general is loyal.

The above conditions are known as the interactive consistency conditions, in which a loyal commander implies that IC1 follows from IC2.

3.1.2 Impossibility Conditions

The difficulty within the Byzantine Generals' Problem arises from the circumstance that if the generals are limited to sharing oral messages, a viable solution becomes impossible unless over two-thirds of the generals are loyal. Given the case of three generals, there is no feasible solution when a single traitor is present. With oral messages, the contents are fully controlled by the sender, therefore enabling a traitor to send any possible message. In Fig. 3.1, the loyal commander sends an "attack" order, but the traitorous Lieutenant 2 reports a "retreat" order instead to Lieutenant 1. To satisfy IC2, L1 must obey the "attack" order. Further, Fig. 3.2 shows a traitorous commander that sends conflicting commands to each lieutenant. Since L1 cannot identify the traitor nor the order sent to L2, both scenarios are identical for L1. If the traitor continuously lies, L1 will be incapable of distinguishing between the two scenarios, therefore forcing them to obey the "attack" order in both.

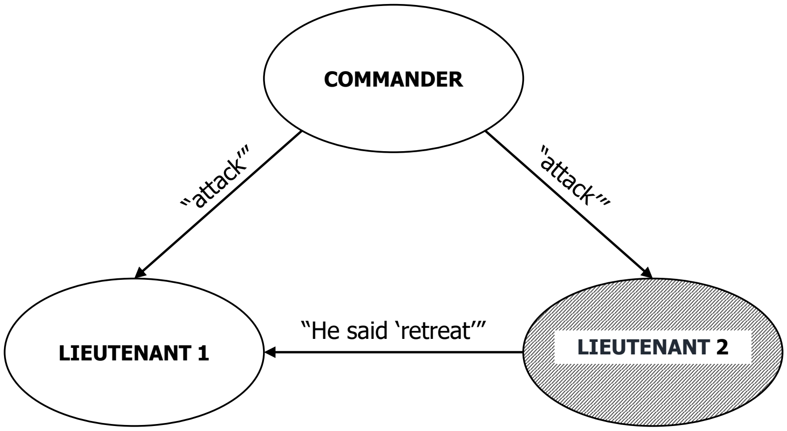


Figure 3.1: Byzantine Generals' Problem Where Lieutenant 2 is a Traitor

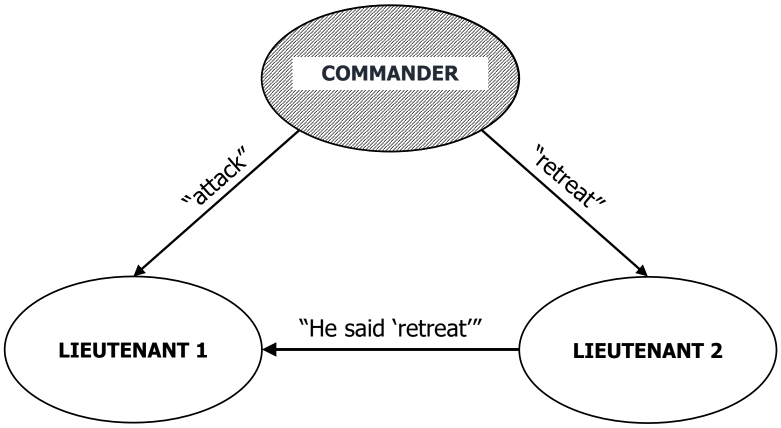


Figure 3.2: Byzantine Generals' Problem Where the Commander is a Traitor

Similarly, a "retreat" order sent to L2 must be obeyed, regardless of the order conveyed from the commander through L1, even if L1 is loyal. L2 must therefore obey the "retreat" order in the second scenario, whereas L1 follows the "attack" order, which violates condition IC1. Thus, no solution exists when a single Byzantine traitor is present within a group of three generals[1].

The root of the Byzantine Generals' Problem may be assumed to be the requirement to reach an exact agreement. However, this assumption is incorrect. Rather than attempting to reach an agreement on a specific plan of action, it can be assumed that the generals are only required to reach an agreement on an approximate attack time. In particular, considering that the commander dictates the attack time t, the subsequent requirements need to be fulfilled:

IC1'. Each loyal lieutenant launches their attack within a 5-minute timeframe of one another.

IC2'. If the commander is loyal, all loyal lieutenants attack within 5 minutes of t specified in the order.

Just like the Byzantine Generals' Problem, this cannot be solved if two-thirds of the generals or less are loyal. To demonstrate this, suppose the commander wants to send an "attack" order at $t = 1:00$ and a "retreat" order at $t = 2:00$. Through the assumed algorithm, every lieutenant follows these steps to obtain the order:

[1] For a rigorous proof of the infeasibility of a three-general solution to withstand a solitary traitor, refer to [83].

1. After receiving t, perform one of the following:
    (a) If $t \leq 1:10$, then attack.
    (b) If $t \geq 1:50$, then retreat.
    (c) Otherwise, move to step 2.
2. Inquire about the decision made by the other lieutenant in step 1:
    (a) If a decision has been made, then make the same decision.
    (b) Otherwise, retreat.

According to IC2', a loyal commander ensures that the correct order will be obtained by a loyal lieutenant in step 1, therefore satisfying IC2. Consequently, IC1 will follow from IC2, which necessitates only proving IC1 under the assumption of a traitorous commander. As there is at most a single traitor, both lieutenants are loyal, so if a lieutenant attacks in step 1, IC1' implies that the other lieutenant cannot decide otherwise. Since both will decide the same value by step 2, this satisfies IC1. This establishes a solution involving three generals for the Problem, which can endure a single traitor. However, this scenario is not achievable. Consequently, an algorithm that upholds IC1' and IC2' while faced with one traitor is infeasible. Thus, simulating m generals using a single general can be used to prove that no solution tolerates m traitors within fewer than $3m + 1$ generals.

3.1.3 Byzantine Fault Tolerance in Computer Systems

The use of majority voting to construct a reliable computer system is based on the assumption that all the same output will be produced by non-faulty processors, which is true provided they use the same input. From majority voting, a reliable system can be achieved if:

1. The exact input is used by each non-faulty processor.
2. The input unit is non-faulty, which allows the use of its values by non-faulty processes.

These are based on IC1 and IC2, where the input unit represents the commander and the lieutenants represent the processors. In order to guarantee the same input and solve the Byzantine Generals' Problem, a communication must be established among the processors. If the input is critical, several input devices should be used to provide redundant data.

However, this does not guarantee reliability, as non-faulty processors must also use that data to produce the same output.

For a non-faulty input device that generates different values due to being read while its value is changing, the non-faulty processors must obtain a reasonable input value. Within computer systems, the problem lies in implementing a message-passing system with the following assumptions:

A1. All messages sent by non-faulty processors are delivered correctly. Since the failure of a communication line is also possible, the algorithms guarantee fault tolerance of at most m failures, whether processor or communication line failures[2]. Under the assumption that a failed communication line cannot cause signed message forgery, the signed message is therefore insensitive to the failure of a communication line.

A2. A processor can determine the source of a received message. To prevent masquerade attacks, inter-process communications must occur over fixed lines instead of message-switching networks. If A4 and signed messages are assumed, A2 is not needed, as masquerading another processor implies forging its messages.

A3. The absence of a message is detectable. This is only possible through a timeout convention, which requires:

   (a) A fixed maximum time for message generation and transmission.

   (b) Synchronized sender and receiver clocks within a fixed maximum error.

Suppose generals are only allowed to take action in the following circumstances:

1. At a fixed initial time for all generals.

2. Upon receiving a message.

3. When a randomly set amount of time has elapsed.

If messages can be sent arbitrarily rapidly, no such algorithm can solve the Problem, even if failure is restricted to the failure to send a message. By fixing the transmission time for input processors, a processor can calculate the timeout one time for each message.

Unless periodically re-synchronized, clocks are likely to arbitrarily drift over time, regardless of the initial synchronization accuracy.

[2] Failure of all communication lines attached to the same processor is equivalent to the failure of a single processor.

A4. Signatures of a non-faulty processor cannot be forged. This implies that the function used must contain the following properties:

   (a) If a processor is non-faulty, no faulty processor can generate its signature.

   (b) Given the message and X, any processor can determine if X equals the signed message.

Since the signed message is simply a data item, (a) cannot be guaranteed. However, the probability of its violation can be minimized in order to improve the reliability of the system through:

1. Random malfunction. By suitably randomizing the sign function, the probability of generating a correct signature from a random processor malfunction will equal the reciprocal of the number of possible signatures.

2. Malicious intelligence. If it guides a faulty processor, signature construction becomes a cryptography problem [84, 85].

3.2 Solutions to the Byzantine Generals' Problem

In critical infrastructures, network reliability has a vital role in the credibility of the monitoring system. In this section, existing solutions to the Byzantine Generals' Problem will be explored. Initially, it will be shown that other wireless network solutions do not function at all or properly with platoon networks. Then, VANET-specific solutions will be assessed to determine their feasibility. To prove wireless network solutions infeasible, the characteristics of VANETs are first presented:

1. Mobility: VANETs consist of mobile and fixed nodes. Based on variable speed, the geographical location of mobile nodes may change at any time. Therefore, the network topology is dynamic, but does not affect the overall network operation.

2. Wireless: the wireless channel is unstable and is exposed to security threats.

3. Multi-hop: due to the limited transmission power and signal coverage, as well as communication with other nodes outside the coverage.

4. Self-organization: according to the distributed network algorithm, nodes move rapidly and independently of the composition of the network.

5. High dynamics: based on their needs, mobile nodes are capable of opening and closing the network when the antenna coverage affects the network.

In addition to its quickly changing topology, Table 3.6 further distinguishes VANET from other wireless networks.

Table 3.6: Comparison Between VANET and Wireless Networks [13]

Parameter	Cellular Wireless Network	VANET
Topology	Fixed	Dynamic, flexible
Infrastructure	Yes	Little
Safety and quality of service	Better	Poor
Speed of configuration	Slow	Fast
Cost	High	Low
Lifetime	Long	Short
Route selection and maintenance	Easy	Difficult
Robustness	Low	High
Relay equipment	Basic station, wired backbone network	Wireless nodes, wireless backbone network
Backbone network characteristics	High speed, reliable	Low capacity, high error

Due to the major differences in topology between VANET and other wireless networks, it can be stated that BFT solutions which are feasible for wireless network applications are infeasible for VANET applications. As a specific application of VANET, platoons are critical real-time systems that directly impact human safety. Based on the security threats and attacks presented in Sections 2.4 and 2.5, the importance of adopting a VANET-specific solution is further emphasized by comparing VANET with mobile ad-hoc networks in Table 3.7.

Although both networks are dynamic, self-configuring and do not require any fixed infrastructure to operate, mobile ad-hoc networks are intended for use when a fixed network infrastructure is unavailable or impractical, such as in disaster relief situations. On the other hand, VANETs are specifically designed to support V2V and V2I communications and networking. As such, the two networks may require different communication protocols

Table 3.7: Comparison Between VANET and Mobile Ad-hoc Networks

Parameter	Mobile Ad-hoc Network	VANET
Dynamic Topology	Changes slowly	Changes frequently and very rapidly
Mobility	Low	High
Node density	Sparse	Dense, frequently variable
Cost	Low	High
Bandwidth	100 kps	1000 kps
Range	Up to 100 m	Up to 500 m
Node lifetime	Depends on power resource	Depends on vehicle lifetime
Multi-hop routing	Available	Weakly available
Node moving pattern	Random	Regular, constrained by road
Addressing scheme	Attribute-based	Location-based
Position acquisition	Ultrasonic sensor	GPS, radar

and technologies to function effectively.

For example, the higher node density and faster node movement rates in VANET introduce challenges in communication and networking. Additionally, the network often requires stricter requirements for latency, reliability, and security due to its safety-critical nature. Nevertheless, VANET solutions have additional requirements, such as supporting V2V communication at high vehicle speeds as well as accounting for interference and signal degradation. These requirements are not necessarily present in mobile ad-hoc networks, which renders such solutions that do not address these issues unsuitable for use in VANETs. Therefore, it is important to consider the specific requirements and characteristics of VANET when designing solutions. In this case, Table 3.8 presents functional requirements to secure platoons against Byzantine faults.

Table 3.8: Functional Requirements for BFT Platooning

#	Requirement	Description
RQ1	Agreement	All non-faulty vehicles decide the same value.
RQ2	Integrity	All non-faulty vehicles decide only once.
RQ3	Termination	All non-faulty vehicles decide before timeout, which requires: (1) A predetermined upper limit for message generation and transmission; (2) Synchronized sender and receiver clocks within a fixed maximum error. As such, any absent messages can be detected.
RQ4	Correctness	All messages sent by non-faulty processors are delivered correctly. If not, the network must tolerate at most m failures, whether processor or communication line failures.
RQ5	Validity	The decision of a non-faulty recipient is validated by a non-faulty validator. The terms ”recipient” and ”validator” will be defined in Chapter 4.
RQ6	Provability	Any vehicle can verify the validity of a decision, which implies that all vehicles can determine the source of a received message.

Based on the above VANET requirements, Table 3.9 surveys literature presenting BFT approaches to secure platoons networks.

Table 3.9: Literature Survey of BFT Solutions for VANET

Research	Summary	Violated Requirement(s)
[86]	The protocol enables all non-faulty processing elements to reach consensus while minimizing information exchange and tolerating the largest number of faulty processing elements.	**RQ4:** Only considering processing elements damaged is insufficient for highly reliable VANET. In practice, the transmission medium may be crashed, omitted, or Byzantine damaged. **RQ6:** After a decision has been made, there is no evidence that other vehicles can verify the validity of a decision.

[87]	The privacy-complaint incentive announcement blockchain network is based on an anonymous vehicular announcement aggregation protocol which utilizes a threshold ring signature to preserve user privacy.	**RQ1:** In case of a road event, the authors failed to address the issue of establishing a shared strategy among vehicles. **RQ3:** In that case, it cannot be guaranteed that all non-faulty vehicles decide before timeout, since no common strategy has been established.
[88]	The algorithm employs a proof-of-eligibility test to validate the presence of a cluster of vehicles near the source of information.	**RQ1:** Since it mandates fixed members to reach an agreement, the solution is impractical in platoon networks of dynamic nature. Also, each server contains a unique node list that records the identities of multiple servers. If a server fails to vote before timeout, it will be removed from the list and will not be taken into account in upcoming rounds. However, this does not necessarily mean the server is faulty. **RQ2:** A previously deleted server due to timeout is unable to decide in future rounds. **RQ3:** For networks of low vehicle density ($\leq$ 200), the agreement and sign request phase failed due to timeout. Hence, the authors do not recommend vehicles producing announcements-aggregated packets in low density and high threshold values due to cryptographic time consumption. **RQ4:** The algorithm mandates a consensus from more than two-thirds of the nodes regarding the list of members and event value. Nonetheless, this is applicable to the synchronous, ”oral message” configuration of Lamport’s algorithm, whereas this work involves an asynchronous setup with signed messages.

[89]	A decentralized peer-to-peer federated learning method with the safeguarding of AV privacy utilizes the secret sharing scheme that is publicly verifiable for encrypted share verification.	**RQ3:** If a vehicle does not respond before timeout, it will be classified as a Byzantine node, although this does not necessarily mean the vehicle is Byzantine faulty. **RQ4:** Leaders are capable of stealing aggregation results.
[90]	The blockchain-enabled framework supervises driving while protecting vehicular information through a long short-term memory model that uses a BFT Proof-of-Stake (PoS) consensus mechanism.	**RQ2:** Since the agreement primarily depends on shared information, inactive information sharing due to lack of enthusiasm may cause some vehicles to not decide at all. **RQ3:** Highly dynamic road conditions of different road-side units may cause traffic congestion depending on the time and location. As such, sending excessive volumes of data to a single road-side unit may cause data loss and/or delays in decision-making.
[91]	The consortium blockchain secures resource sharing in vehicular edge computing through multi-step smart contracts. It applies a BFT-based PoS protocol to establish consensus.	**RQ3:** No timeout was specified for the decision-making of non-faulty vehicles. **RQ6:** No evidence was found that decisions can be validated by any vehicle.
[92]	The real-time blockchain provides accountability through smart contract services, such as automated toll fee settlement, reservation and payment for slots, fuel payment, and renewal of insurance.	**RQ5, RQ6:** Although replicas wait for a timeout, they do not know which replicas are non-faulty, which yields very poor performance.

[93]	The blockchain infrastructure provides forensic services for accident investigations. It addresses privacy, storage overhead and membership management issues in blockchain through pseudonyms, vehicular public key infrastructure and a fragmented ledger.	**RQ1, RQ2:** A mechanism should be introduced to identify malicious vehicles, and the participation of various entities in forensic blockchains must be regulated to ensure agreement and integrity. **RQ3:** No timeout was specified for the decision-making of non-faulty vehicles. **RQ5, RQ6:** Data availability is constrained by personal storage and shared counterparts, since there are no existing mechanisms to guarantee the presence of critical forensic information within the blockchain.
[94]	The blockchain features local physically-verified transactions to secure VANET without requiring constant communication with road-side units or other components.	**RQ3:** Scalability issues face such real-time systems, which show that Proof-of-Work (PoW) is an unacceptable consensus method for VANET if time is a critical aspect. **RQ1, RQ2, RQ4-6:** With higher than average delays introduced, this would result in no history updates for the time period, which could compromise correct data delivery to non-faulty vehicles and ultimately decision-making and agreement.

3.3 Challenges and Motivation

As shown in Table 3.9, the violated requirements impact latency, reliability and safety in a platoon network. Since communications can be critical for maintaining safe operation, the solutions above are not well-suited for mitigating Byzantine faults in platoon networks.

By design, the solutions in [86, 89] do not provide satisfactory levels of integrity, efficiency, reliability or scalability for the safety-critical platoon networks compared to other blockchain solutions presented in Table 3.9. For instance, the two cannot prevent leaders from stealing aggregation results. Nevertheless, the challenges in existing solutions are summarized below:

1. Privacy: the transparency of blockchain means that information stored on it is available to everyone, as shown in [88, 93, 94]. Other solutions [86] also do not ensure privacy.

2. Heavy-weight encryption usage: results in communication and authentication latency in addition to high power consumption, as devices are of limited processing power.

3. Accountability: misbehaving vehicles should be penalized in the reputation system. Although it is an an important factor for AVs, existing solutions [86–93] do not fully address this issue.

3.4 Proposed Solution

The linear spatial structure and dynamic changes within a platoon in real-time introduce constraints to the topology of the conventional consensus problem. Therefore, an analysis of the proposed solution will be conducted statically and dynamically to formalize consensus guarantees within a platoon. In addition to conventional consensus where vehicles communicate their states, vehicles also monitor and quantify distinct conditions of adjacent vehicles, such as their velocity, which contributes to the consensus protocol.

Effectively managing faults rather than aiming to tolerate them is crucial for this application, as one misbehaving node could prevent the correct execution of an operation in which consensus was reached. For instance, if platoon members agree to accelerate, a single disobedient member will block the operation entirely. In this case, the misbehaving vehicle should be detected and penalized in the system.

In general, leader-based management inherits a single-point failure by design, which poses severe security and safety risks to platoon vehicles. As a result, this work proposes a blockchain-based platoon management scheme to establish safe BFT platoon operation.

It is hypothesized that a single vehicle exists which compromises the security of the platoon through Byzantine behavior, which can pose threats to integrity and human safety. Therefore, the goal of this work is to develop a solution to accurately detect and manage Byzantine behavior in platoons of AVs, with the ultimate objective of ensuring the safety of human passengers as well as other road users. The platoon coordination algorithm disseminates the verification of characteristics among all participants, thus eradicating the role of the leader as a single point of failure. Specifically,:

- The distributed consensus scheme specifically designed for platoon applications over VANET is presented in Chapter 4.

- The feasibility of the scheme is illustrated by testing on speed harmonization operation in Chapter 5.
- Finally, conclusions and future research directions are presented in Chapter 6.

Chapter 4

The ConsenCar Protocol

4.1 Problem Formulation

This work addresses consensus using V2V communication over VANET to provide BFT. Byzantine behavior challenges consensus by transmitting incorrect or conflicting information to other vehicles in the platoon, which shares information about their position, speed, and other relevant data. As a result, the Byzantine platoon can become disorganized and fail to maintain safe operation, potentially leading to collisions or other accidents. Here, the presence of a single vehicle that compromises the security of the platoon through Byzantine behavior is hypothesized, which poses threats to integrity and human safety.

To establish safe platoon operation under Byzantine cyberattacks, an **automated** and **immutable** method to **share information** in the **decentralized** system is required to solve the **consensus** problem. Rather than tolerating them, the solution aims to detect faulty vehicles and eliminate them from the platoon. In this light, the developed algorithm detects the presence of Byzantine behavior in the demonstrated platoon management scheme. As such, a preventative measure can be taken to keep track of Byzantine behavior and penalize participating vehicles in a reputation system accessible by authorities.

4.2 Consensus

4.2.1 Conventional Consensus

In a conventional consensus, each vehicle must provide a reliable and correctly ordered broadcast. Referring to Table 3.8, the consensus problem is solved if RQs 1-3 are met, and a vehicle that meets those requirements is non-faulty. However, this application also requires the decision to be correct, which introduces RQs 4-6.

4.2.2 Assumptions

Addressing consensus challenges in self-organizing systems becomes complex when vehicles cease communication or engage in sending harmful messages. In response to these difficulties, the following assumptions are made:

1. Partial synchronization: although message delays are unknown and unbounded, all messages will ultimately reach its destination prior to the specified deadline.

2. Known participants: every participant knows all other participants who are allowed to vote.

3. Signatures: unforgeable, used by every participant to verify the sender of a message.

As compared to conventional consensus systems where all nodes are capable of validating and voting on every request, not all platoon vehicles can validate the same set of requested transitions. As an example, vehicles nearing the rear end of a platoon are typically detectable solely by the platoon's trailing vehicle.

Hence, an elaborate role allocation for vehicles is adopted in this consensus system, which contains seven distinct roles:

1. *Requester:* requests an operation.

2. Receiver: processes requests received from a *Requester*.

3. *Responder:* responds to the *Requester* when consensus is reached.

4. *Proposer:* proposes a new system state.

5. *Validator:* validates the newly proposed state.

6. *Acceptor:* votes for the proposed state.
7. *Learner:* receives the accepted state.

- Each vehicle v_x is a *Requester*.
- Each platoon vehicle p_x is an *Acceptor* and *Learner*.
- The first and last platoon vehicles p_1 and p_N are *Receivers*, *Responders* and *Proposers*.
- Each direct neighbor of a vehicle v_x is a *Validator* for v_x, since it can read its license plate.

4.2.3 Zero-Fault Tolerance

As long as a sufficient majority is correctly operating, conventional consensus can tolerate a certain number of failures. This way, the decisions of non-faulty nodes are not influenced by those which are faulty, therefore maintaining a consistent system state. Nonetheless, aforementioned assumptions are not applicable to this safety-critical application, as the safety of a single passenger carries more weight than any majority decision. Therefore, a single failure cannot be simply overseen, but must be detected and addressed to ensure the safety of human passengers and other road users.

Utilizing the detailed role assignment above, The algorithm necessitates *every* participant to unanimously concur on a single state transition. In this scenario, the objective is not to tolerate failures; instead, it is to identify whether a consensus-based unanimous decision was attained or if consensus was not reached. In the latter case, the vehicles implicated in the failed decision-making process are accountable. A vehicle held responsible might have encountered issues transmitting its vote, attempted to transmit a malicious message, or voted against the decision.

4.3 The ConsenCar Protocol

ConsenCar is a protocol designed specifically for cyber-physical systems, such as platoons, which reliably detects failures in systems where a single failure is intolerable.

In Fig. 4.1, the consensus-based platoon management is demonstrated for the speed harmonization function, where platoon property validation is distributed across all members. As shown in Fig. 2.9, it considers three V2V communication topologies, and assumes synchronized message delivery with adequate timeouts.

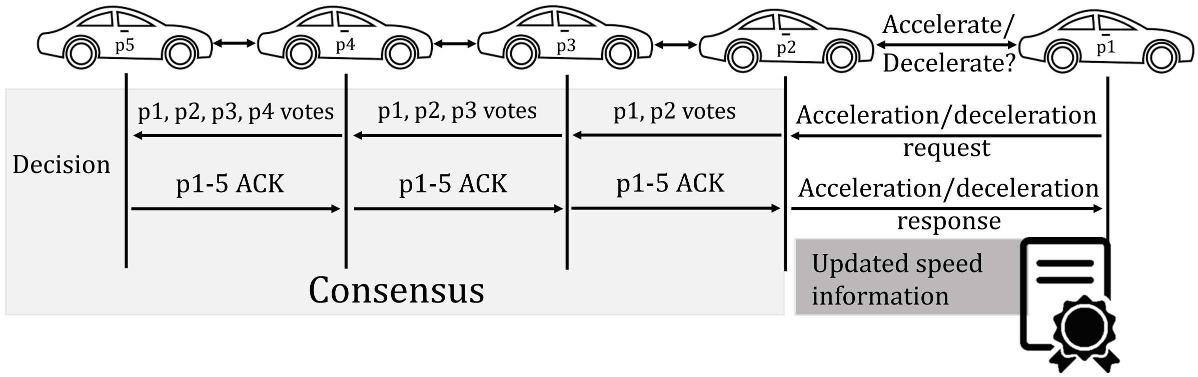


Figure 4.1: Consensus-Based Speed Harmonization. P_1 requests to move at a different speed (accelerate/decelerate). The acceleration/deceleration request is forwarded to every vehicle in the platoon, prompting them to cast their votes on the proposal. If a unanimous agreement is reached, action is finally taken and current speed is updated.

Rather than broadcasting messages, ConsenCar uses single-hop transmission, where each hop confirms the messages passed through previous hops. Specifically, this protocol is characterized by:

1. Node order: nodes are ordered and divided into:
   - *Acceptors*: first $2f + 1$ nodes.
   - *Learners*: last f nodes.
2. Requests: the source node sends hop-by-hop CH messages to the *Acceptor* (node $2f + 1$) through CH messages. Upon receiving and accepting the CH, the *Acceptor* will send three messages:
   - Client reply.
   - ACK message to the source node.
   - CH to Learners.

3. Handling failures: after sending a $< CH >$, each node sets a timer. If the timer expires before receiving an $< ACK >$, it will issue an $< SPT >$ to the source node. The source then reorders the chain to append malicious nodes.

As such, two types of consensus rounds are considered for the platoon scenario:

1. Planned action: requires the agreement of all vehicles.
2. Failure identification: requires the agreement of $f + 1$ vehicles.

At most, is is assumed that f failures can occur, and that all other participating vehicles are non-faulty. Although such reordering is possible, it is infeasible in platoons where the spatial location of nodes affects routes and connections. Instead, it requires overtaking maneuvers to adjust the position of vehicles. Moreover, all requests must be sent to the source, which increases the risk of failure in case the source is faulty.

The ConsenCar protocol successfully terminates if and only if all participating Acceptors agree on the same value. Otherwise, the protocol detects the Acceptor responsible for the failing consensus so that non-faulty nodes can take appropriate action. Nonetheless, the underlying network topology determines the maximum failures f that can be reliably detected by the protocol. Compared to other algorithms such as BChain [95], they cannot guarantee successful termination in the first round, as they could require up to $3q$ rounds, where q is the number of faulty nodes.

4.4 How Blockchain Establishes BFT

To establish robust BFT in platoon networks, one must provide the necessary level of immutability, scalability and flexibility while efficiently handling high transaction volumes. Motivated by this, the tamper-proof technology allows for the creation of highly adaptive decentralized AV networks through consensus algorithms to verify the order and content of transactions.

Each node in the platoon network is like a general; waiting for orders to execute. Since there is no middleman to execute an order on behalf of a node, each node must individually make a decision. In this sense, blockchain creates a layer of trust without needing to trust every node, by collaborating with other nodes to reach an agreement on the truth before it is recorded. If one node is unsure about a substance of the communication, others can verify it using what they know to be true. Once one node records it, a copy is sent to all other nodes in the network, producing redundant data.

The following are key blockchain characteristics that enable BFT operation:

- Immutability: describes the inability of a blockchain to be modified once it has been recorded. It ensures that the record of transactions and the current state of the network cannot be altered or tampered with, which ensures integrity in the network.
- Consensus: describes the process by which participants reach agreement on the contents of the shared ledger. It establishes integrity and reliability in the ledger by ensuring that all parties agree on the transactions and current state of the network.
- Smart contracts: self-executing contracts that facilitate, verify, and enforce the negotiation or performance of a contract. These ensure platoon security by reducing the need for manual intervention.
- Shared information ledger: a database that is shared and replicated among all participants in the network. It stores and tracks the movement and status of vehicles in real-time, which improves safety and efficiency.
- Decentralization: describes the distribution of authority and power among multiple parties, rather than being concentrated in a central authority. It increases transparency and accountability, as all parties have access to the same information. This builds trust among participants and makes engaging in fraudulent activity more difficult for any participant. Also, decentralization has the ability of ensuring that no single point of failure exists.

4.5 Modes of Operation

4.5.1 Normal Operation

Under normal conditions, all non-faulty vehicles must agree on a single proposal, which could consist of any planned platoon operation. Here, three types of messages are used: $< CH >, < ACK >, < NAK >$.

The structure of each message is found in Eq. 4.1. For simplicity, it is assumed that a vehicle specification is authenticated only through license plates by predecessor and successor vehicles, using distributed certification (blockchain).

$$< T, s, l_s, v_s, v_r, m, \sigma > \tag{4.1}$$

where T is the message type ($< CH >, < ACK >, < NAK >$), s is the sequence number of the ongoing consensus round, l_s is the sender's license plate number, v_s is the sender's speed, v_r is the receiver's speed (if present), m is the proposal to vote upon, and σ is the signature of the sender. The execution of the protocol is detailed in the following steps:

1. A Proposer puts forward a newly planned operation ($< CH >$), then directs it towards the tail of the platoon.

2. Every intermediate vehicle or Validator validates the proposal, and submits its vote by appending a unique $< CH >$ to the proposed $< CH >$ message. If a Validator receives multiple chained $< CH >$ messages, it will start by validating every vote through the verification of these predicates:

    (a) Every message contains the same sequence number as the current sequence number.

    (b) v_s of the current message matches v_r of the previous message.

    (c) σ is valid using the public key which corresponds to l_s.

3. When the final Acceptor (other Proposer) receives the proposal alongside all votes, it makes a decision. If the proposal is agreed upon by all voters, then consensus is reached and it sends back an $< ACK >$ alongside all received signatures to the other Acceptors (now Learners). In case one vote against the proposal exists, the final Acceptor opts not to proceed with the proposal and replies back with an $< NAK >$ alongside all signatures.

4. Every Learner receives and validates $< ACK >$ and $< NAK >$ signatures and makes a decision, until the final Learner (the Proposer) is reached.

Based on Fig. 4.1, the critical assumption is made that every vehicle is capable of sending messages to its neighboring $f + 1$ vehicles in one direction. As a result, this ensures that every vehicles receives an $< ACK >$ indicating a successful consensus round. If such assumption is not made, it cannot be guaranteed that proposal decisions are successfully transmitted to all vehicles. Therefore, the assumption confines the possible adopted topologies to 2P2S.

Example: Speed Harmonization

In this context, speed harmonization refers to the process by which the speed of multiple vehicles in a platoon is synchronized or adjusted through the use of advanced communication and control technologies. Mainly, the leader of the platoon constitutes the Requester in this scenario, as illustrated in Fig. 4.1.

Assuming that the Leader requests a change in speed, the algorithm presented in Fig. 4.2 is run for each request to prove the validity of ConsenCar.

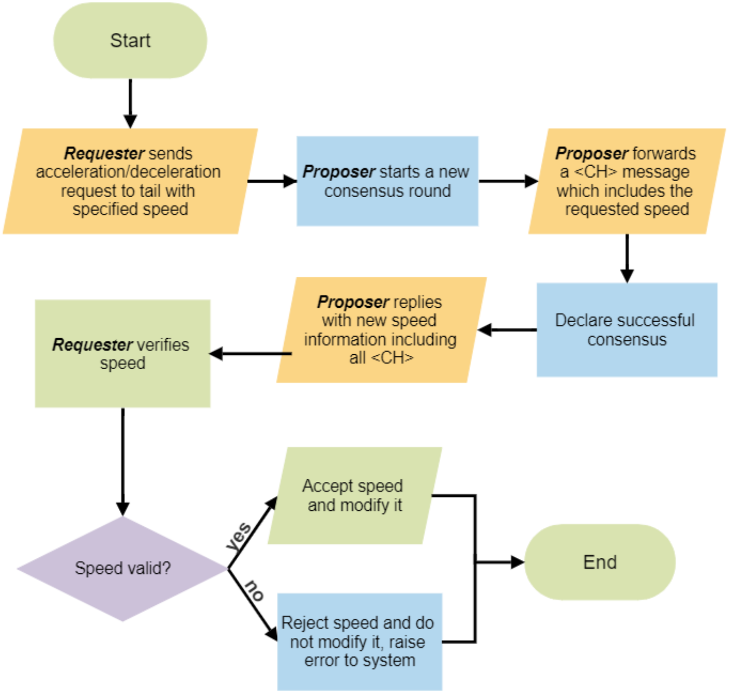

Start
Requester sends acceleration/deceleration request to tail with specified speed
Proposer starts a new consensus round
Proposer forwards a <CH> message which includes the requested speed
Declare successful consensus
Proposer replies with new speed information including all <CH>
Requester verifies speed
Speed valid?
yes
no
Accept speed and modify it
Reject speed and do not modify it, raise error to system
End

Figure 4.2: Normal Operation Algorithm for Speed Harmonization

Once the decision has been made, both s_s and s_r verify that each vehicle has adjusted its speed to the specified speed. Given the assumption that signatures are unforgeable, if the Requester cannot validate the speed, it will be considered an external error. As such, the requested speed will be rejected, and an error message will be raised in the system. The execution of this algorithm is presented in Chapter 5. However, with small modifications, this protocol generalizes to secure other platoon operations. For instance, the *join* function can be secured by authenticating the license plate of a joining vehicle, then running the same algorithm to reach a consensus on the new join request.

4.5.2 BFT Operation

The flowchart in Fig. 4.3 represents the algorithm which the system will follow in case consensus cannot be reached by vehicles. Assuming that message signatures are unforgeable, consensus can be reached if and only if *all* vehicles respond a) correctly; and b) before the specified timer runs out. Any violations will produce a) a $< NAK >$; or b) a timeout. Consequently, all vehicles are reliably capable of detecting a failed consensus.

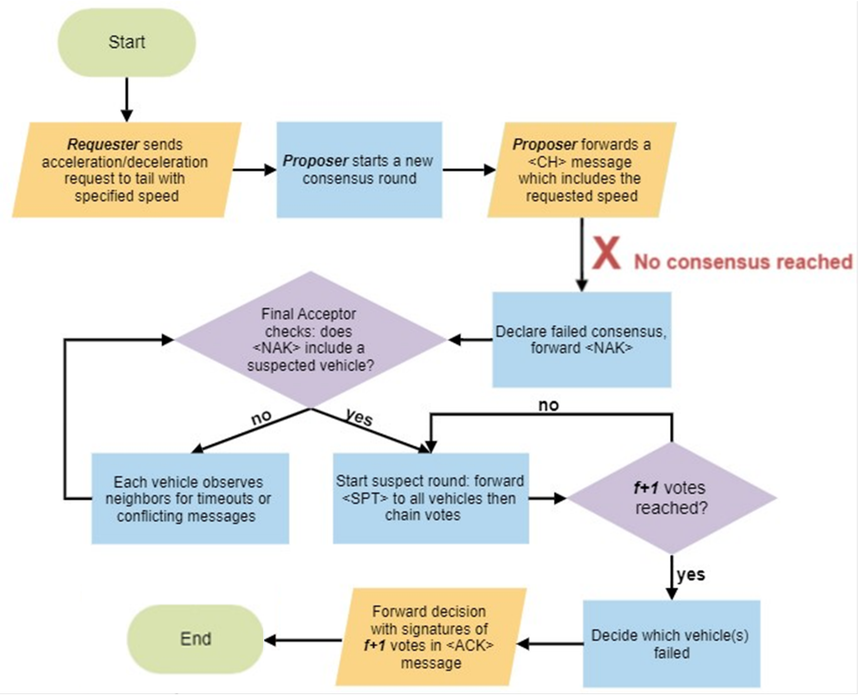

Start
Requester sends acceleration/deceleration request to tail with specified speed
Proposer starts a new consensus round
Proposer forwards a <CH> message which includes the requested speed
X No consensus reached
Declare failed consensus, forward <NAK>
Final Acceptor checks: does <NAK> include a suspected vehicle?
no
yes
no
Each vehicle observes neighbors for timeouts or conflicting messages
Start suspect round: forward <SPT> to all vehicles then chain votes
f+1 votes reached?
yes
Decide which vehicle(s) failed
Forward decision with signatures of f+1 votes in <ACK> message
End

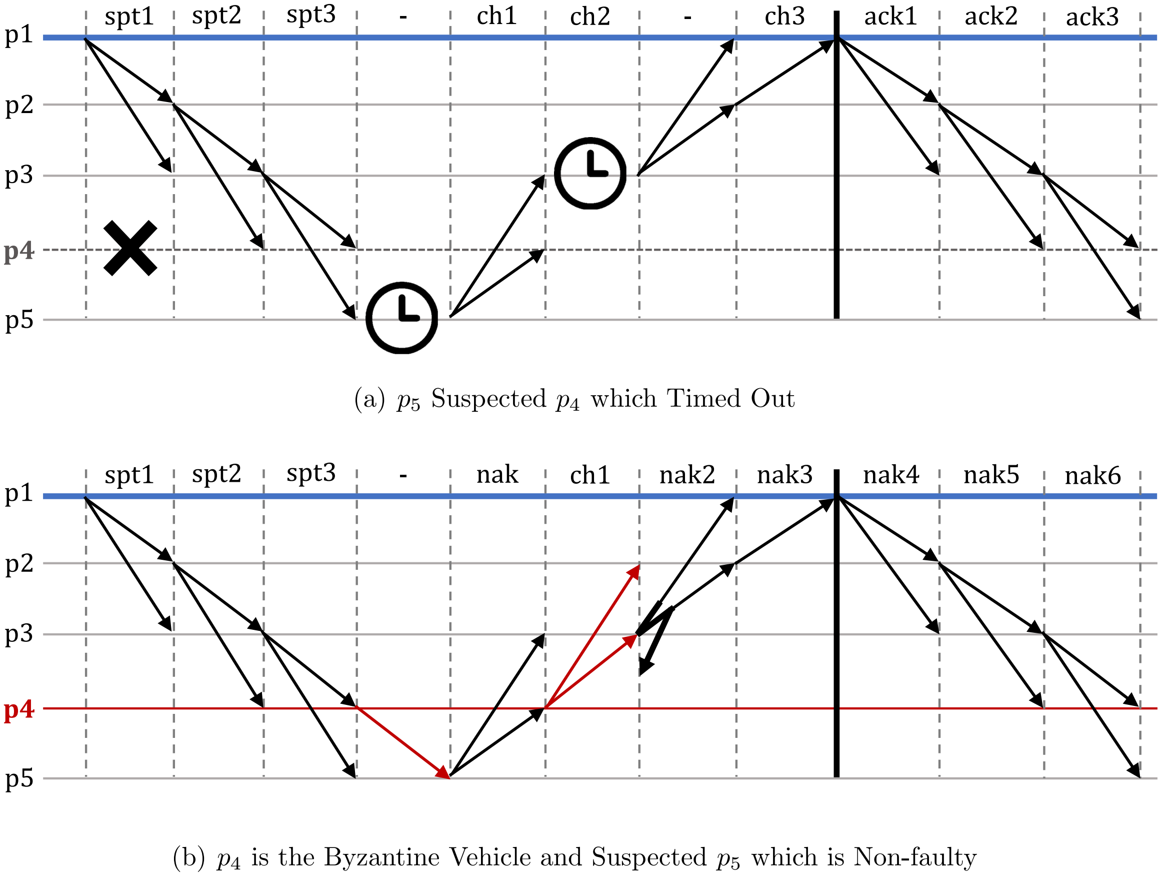
Figure 4.3: Byzantine Fault Tolerance Algorithm for Speed Harmonization

Every vehicle sets a timer that aligns with the anticipated time needed for the consensus process among the remaining platoon vehicles. This time can be calculated through Eq. 4.2:

$$t_{TO} = (N - i) * \tau \tag{4.2}$$

where $(N - i)$ is the number of vehicles awaiting to cast their votes and τ is a constant timeout period applicable to each vehicle. For this application, τ is set to 500 ms [96–98], whereas the transmission and processing latency between each two vehicles is assumed to be 5 ms. If and only if the successor vehicle is capable of providing a proof of consensus (valid $< ACK >$) before t_{TO} runs out, it decides the consensus value, then forwards an $< ACK >$. Otherwise, it decides the failure of consensus, then forwards a $< NAK >$ alongside the ID of the suspected vehicle(s), if any are suspected to have timed out or deviated from the protocol.

A timeout of p_5 is illustrated in Fig. 4.4. p_3 waits until p_4 times out before forwarding a $< NAK >$. When p_1 returns the $< NAK >$, p_2 will forward it to p_4 through p_3, providing it with another opportunity to reply. Regardless of an early $< NAK >$, it is important to forward the messages to every vehicle. In case p_4 fails to respond before the timer of p_5 runs out, p_5 will decide the failure of consensus.



(a) p_5 Suspected p_4 which Timed Out

(b) p_4 is the Byzantine Vehicle and Suspected p_5 which is Non-faulty

Figure 4.4: Suspect Round Proposed by p_1 to Identify the Byzantine Vehicle

To reliably detect f failures, $f + 1$ bidirectional communication hops are required for the network topology, which requires the adoption of the 2P2S topology (Fig. 2.9) for

$f = 1$.

Following an unsuccessful consensus round, the final Acceptor checks the $< NAK >$ message for any suspects. If suspects are found, a suspect round requiring $f + 1$ votes from neighbors of the suspect is started. Given a maximum of f faults, $f + 1$ votes are required such that a minimum of one vote is correct.

Before chaining votes, $< SPT >$ messages are conveyed to each vehicle. By doing so, each vehicle will know all the suspects, whereas neighbors of each suspect will be capable of observing its timeout or conflicting messages prior to voting against it. In Fig. 4.4(a), p_1 triggers a suspect round due to the timeout of p_4. Under normal operating conditions, p_3 suspects p_4 for the timeout, whereas p_1 starts a suspect round by forwarding an $< SPT >$ after the consensus fails with p_4 as a suspect. p_5 witnesses another p_4 timeout and uses a $< CH >$ to vote against it. Once p_3 receives the $< CH >$ message and p_4 does not respond before timeout, p_3 will similarly proceed to vote against p_4. Thus, reaching $f + 1$ votes will have p_1 decide that p_4 has timed out. It then forwards an $< ACK >$ containing the decision alongside signatures corresponding to the $f + 1$ votes. This process is repeated for all suspects in the platoon, until all malicious vehicles are identified. Finally, non-faulty vehicles use this information to eliminate the faulty vehicle from the platoon by splitting and re-merging the platoon.

4.6 Operational Constraints

Reaching consensus is merely a virtual agreement to execute a specific action. In reality, these actions may be physically blocked, slowed down, or altered due to external constraints.

Even if consensus is reached, it may be physically infeasible to safely execute the action. During execution, a vehicle that is supposed to perform a specific action may need to defer, stop or perform an alternative action to ensure passenger safety; the ultimate priority. Beyond such external influences, any vehicle has the potential to encounter a severe malfunction or intentionally obstruct platoon activities solely through its physical existence.

Further, the behavior of the faulty vehicle on the road in this work was modeled in a rather stable manner to facilitate the demonstration of platoon splitting and re-merging when a fault is detected. In reality, its behavior may not be as stable or predictable, which requires the adoption of predictive and/or emergency response algorithms in addition to ConsenCar.

Therefore, this work aims to reliably detect and eliminate failures, which creates the foundation for other collaborative decision-making processes necessary to increase road safety. For instance, keeping track of faulty vehicles in a reputation system can help block constantly misbehaving vehicles from joining and/or participating in voting processes. Nonetheless, non-faulty vehicles which are forced to deviate from the consensus must be distinguished from faulty vehicles. It should also be considered that faulty vehicles may transmit false messages, but not all vehicles can sense the root cause of the deviation, which increases the complexity of the problem.

Chapter 5

Implementation, Testing and Analysis

5.1 Implementation

The V2V network was simulated on OMNeT++; an event-based network simulator, with the INET Framework; an open-source OMNetT++ model library offering protocols, agents, and various other model components for communication network simulation. INET is particularly beneficial for designing and testing new protocols, which is the case for this work.

5.2 Platoon Velocity Simulation

The platoon velocity simulation adopts the 2P2S topology; the only topology in which ConsenCar can adopt. At the start of the simulation, the leading vehicle is located at a distance from the remaining vehicles, which means that $p2-5$ must move towards the leading vehicle first before the entire platoon drives together at a harmonized speed. As seen in Fig. 5.1, it requires the non-faulty vehicles 10 s approximately to start driving altogether at a harmonized speed. Nonetheless, only $p2, p3$ and $p5$ initially move at a harmonized speed ($t < 10s$) since $p4$ is a Byzantine node.

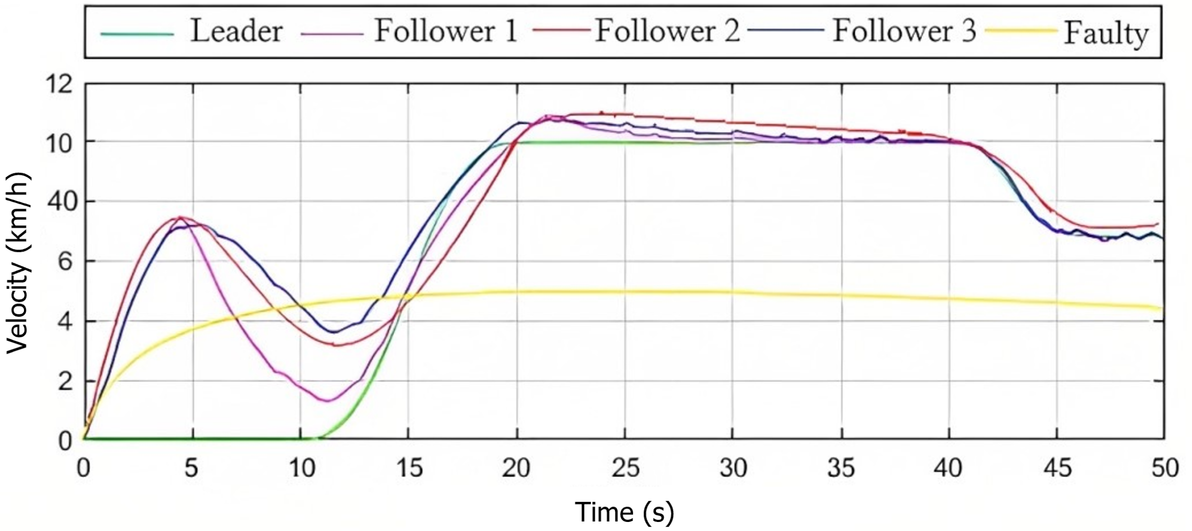


Figure 5.1: Velocity Simulation of Vehicles

Employing the ConsenCar protocol, it can be seen that $p2, p3$ and $p5$ successfully accelerated and decelerated ($t < 10s$) despite the presence of $p4$ as a Byzantine node in the platoon. Since more than two-thirds of the 4-vehicle platoon is non-faulty, ConsenCar is capable of securing platooning communications. Thus, the malicious votes made by $p4$ during acceleration and deceleration processes do not affect the decision-making of the three non-faulty vehicles.

As the vehicles approach $p1$, $p1$ transmits an acceleration request to the 5-vehicle platoon. Following the algorithm in Fig. 4.3, all vehicles vote on this request except for $p4$ which fails to reply, thereby resulting in a timeout. Since the consensus failed, a consensus round started in which $f + 1$ votes were required to decide that $p4$ is faulty. Following Fig. 4.4, consensus decision-making successfully declared $p4$ as faulty, and its vote did not affect the operation of the remaining vehicles. Note that not only does $p4$ transmit conflicting information to disrupt the platoon, but also does not follow the speed in which the 4-vehicle and 5-vehicle platoons agree upon. Therefore, $p5$ overtakes $p4$ to move in harmony with other non-faulty vehicles. Lastly, it can be seen that the non-faulty vehicles accelerate together to reach 50 km/h at around $t = 20s$ then decelerate together to about 35 km/h at around $t = 45s$ without any interruptions from $p4$ which has been left behind.

It is worth mentioning, however, that the velocity of the faulty vehicle was simulated in a steadier and less unpredictable manner here for simplicity. In reality, its behavior may not be predictable, which mandates the use of ConsenCar for more platooning functions, such as adaptive response to failure so that non-faulty vehicles can appropriately collaborate in real-time to avoid accidents with the faulty vehicle.

5.3 Communication Overhead

5.3.1 Number of Messages

Traditional consensus protocols usually assess the rate at which handled requests contribute to the replication of state machines. Nevertheless, ConsenCar focuses on the necessary number of transmitted messages for complete execution of a single consensus round, due to the limited bandwidth in VANETs.

It is observed in Fig. 5.2 that the number of transmitted messages does not depend on the number of failures, yet dependent on the maximum number f. Assuming the transmission of $2N$ messages where an $< ACK >$ is received for each message sent, the use of a reliable consensus-based approach mandates additional communication overhead as compared to a leader-based approach. To demonstrate, Fig. 5.2 depicts that the communication overhead for decentralized approaches is higher than that of leader-based approaches. However, both Byzantine Fault Tolerant Asynchronous Reliable Multicast (BFT-ARM) [99] and ConsenCar are linear to the vehicle count. Yet, the message distribution is more balanced in ConsenCar as compared to the leader-based approaches (BFT-ARM and BChain [95]), in which the leader is responsible for handling all messages, thereby increasing protocol execution risks in case the leader is faulty.

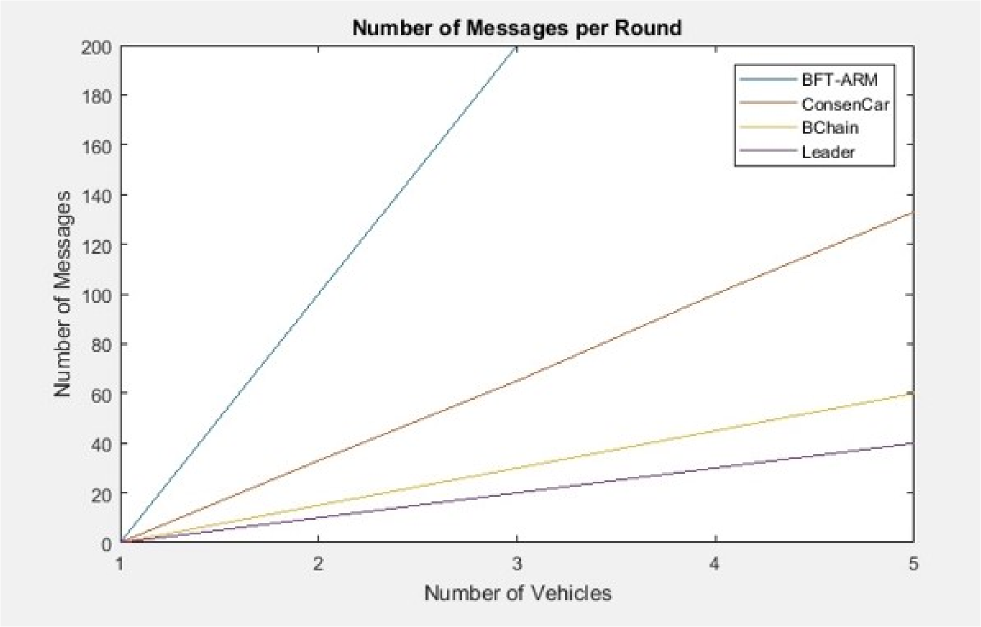


Figure 5.2: Number of Messages per Consensus Round

Conceptually, ConsenCar is similar to that of BChain in the sense of transmitting messages hop-by-hop rather than broadcasting them, where every hop confirms messages of past hops. Nevertheless, BChain is not presented here for comparison as a consensus protocol for platoons, as it is a general consensus protocol that is infeasible for platooning applications. Further, it cannot guarantee first round termination, as it requires at most $3q$ rounds for successful termination.

5.3.2 Consensus Time

The total time needed to reach consensus is another crucial performance metric. As illustrated in Fig. 5.3, the transmission and processing latency between each two vehicles was assumed to be 5 ms, while the timeout value was fixed to 500 ms. From Fig. 5.1, it can be seen that harmonizing the platoon for a modified speed is a smooth and fast process, which is ideal for ensuring road and passenger safety. In terms of scalability, the estimated total consensus time for a large platoon would be around 2 s. Compared to BFT-ARM, this time is always faster due to the reduced number of rounds necessary to reach consensus.

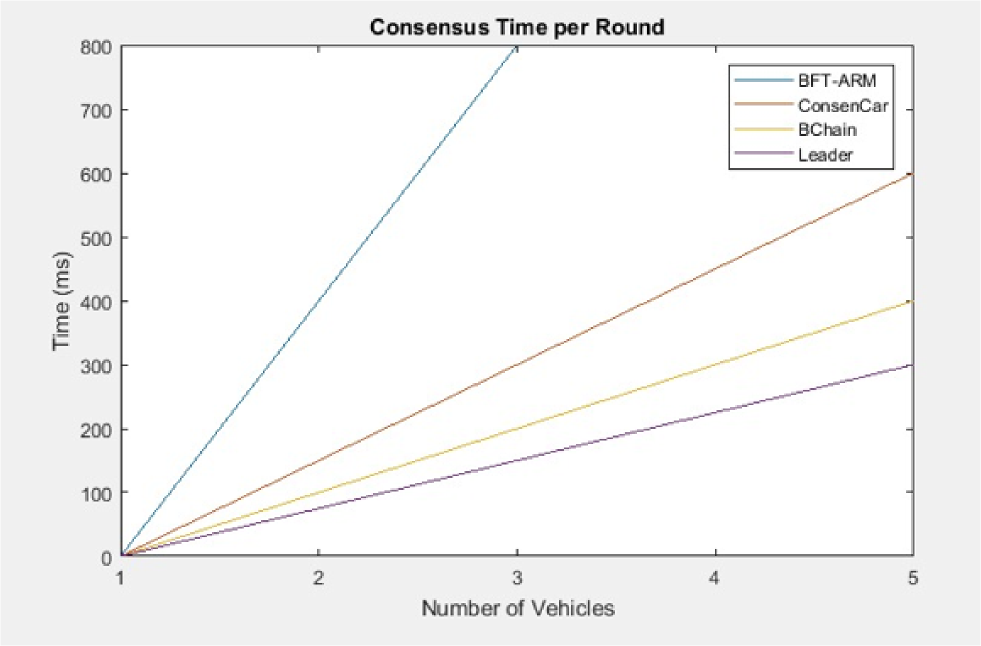


Figure 5.3: Total Consensus Time

Finally, the proposed blockchain-based consensus protocol generalizes to secure other platoon operations, such as splitting and merging, intersection negotiation, lane-changing, overtaking and others. With small changes to tailor to each operation, ConsenCar can generalize to provide secure platooning operations under Byzantine attacks, thereby guaranteeing the welfare of all road users.

5.3.3 Verifiable Speed Information

By utilizing signatures, vehicles can generate and update speed information that was signed by all vehicles. Subsequently, vehicles that are not part of the platoon can request and authenticate the information by verifying the signatures to ensure their alignment with the agreement.

5.4 Consensus Safety

The platoon management offered by ConsenCar is a safer, consensus-based alternative to other protocols which rely on majority-based consensus methods. Specifically, ConsenCar requires (a) the agreement of all vehicles on the same decision; and (b) the verification of that decision by all vehicles. Hence, failures and unintended vehicle interactions can be reliably detected and prevented. Further, the sequential execution ensures the validity of transmitted messages in an ordered chain, thereby avoiding message collision and retransmission.

The predetermined timeouts ensure that a definite agreement is reached within a bounded timeframe during the consensus process. In comparison to other algorithms such as BChain, expensive re-chaining is required in the event of a failure. Moreover, ConsenCar does not require *view changes* in which a suspected Proposer is changed, as compared to BFT-ARM. Instead, any identified failure will be tolerated by the platoon without the need to change the Proposer. Therefore, ConsenCar simplifies the consensus process by only using two kinds of consensus rounds. Based on the assumption of unforgeable signatures, achieving a "wrong" decision is impossible if one vehicle at least is non-faulty. Nonetheless, failures can be reliably detected if the vehicle reaches $f + 1$ neighbors in every direction. Consequently, the failed vehicle is identified and safely eliminated from the platoon such that it is no longer capable of compromising the real-time system.

5.5 Verification of Design Requirements

Based on the design requirements in Table 3.8, design requirements; agreement, integrity, termination, correctness, validity, and provability were met by the proposed algorithm. Table 5.1 below verifies that ConsenCar successfully meets all listed design requirements.

Table 5.1: Verification of Functional Requirements for BFT Platooning

#	Requirement	Verification of Requirement
RQ1	Agreement	All non-faulty vehicles decided the same value before taking action regardless of the faulty vehicle, as proven by Fig. 5.1.
RQ2	Integrity	Each non-faulty vehicle voted only once by transmitting its vote to the next vehicle and receiving an $< ACK >$ message.

RQ3	Termination	• All non-faulty vehicles decided before the timeout, as proven by the consensus reached by the 4-vehicle and 5-vehicle platoons in Fig. 5.1. • Faulty vehicles which failed to do so triggered a consensus round to detect the source of failure, as proven by the successful tolerance and overtaking of $p4$ in Fig. 5.1 after deciding it was faulty to leave it behind.
RQ4	Correctness	• All messages sent by non-faulty processors were delivered correctly, as proven by $< ACK >$ messages received by each sender. • The protocol successfully tolerated the failure of $p4$.
RQ5	Validity	Decision of non-faulty recipients were validated by non-faulty Validators when deciding that the suspect was faulty as well as before an acceleration/deceleration action was taken.
RQ6	Provability	The shared ledger allowed vehicles to determine the source of a received message, and therefore verify the validity of each decision.

Chapter 6

Conclusions and Future Directions

6.1 Conclusions

The ConsenCar protocol detects and manages Byzantine faults in platoons of AVs, therefore ensuring the safety of human passengers and other road users. The protocol leverages blockchain and V2V communication over VANET to provide distributed consensus-based platoon management. Simulation results show that ConsenCar successfully meets all design requirements (RQ 1-6), and efficiently secures the platoon against Byzantine cyberattacks. Most importantly, the ability of ConsenCar to generalize to secure other platoon operations makes it a versatile solution for ensuring the welfare of all road users.

This thesis investigates and addresses the risks posed by Byzantine behavior in vehicular platoons, which can lead to disorganization and collisions or accidents, therefore endangering human safety and compromising the integrity of the platoon. The research hypothesizes the presence of a single vehicle within a platoon engaging in Byzantine behavior, which could threaten the security, safety, and performance of such coordinated formation, in addition to any third-party communications with the leader.

This study encompasses a comprehensive literature review, tracing the historical evolution of AVs from the 1500s to the present era. It examines the wide-ranging impact of AVs on various industries while dissecting security concerns and open challenges inherent to AVs and AV platoons. Additionally, it introduces a decentralized platoon management approach, which replaces the traditional leader-based system with a distributed validation framework, thus eradicating single points of failure and heightening platoon resilience.

Notably, the ConsenCar protocol emerges as a solution aimed at fortifying the security and safety of AV platoons. Employing unforgeable signed messages, it proficiently detects Byzantine behavior via collaborative decision-making, assuring integrity in platoons. Collaborative consensus-based operations, powered by the shared ledger in blockchain, foster efficient and integrative decision-making, even amidst Byzantine faults, without necessitating extensive re-chaining or leadership changes. Moreover, the protocol prioritizes communication efficiency by minimizing the number of transmitted messages per consensus round, ensuring timely and secure decision-making within the constraints of VANET communication bandwidth.

ConsenCar assumes partial synchronization, fixed timeouts, known participants, and unforgeable signatures to solve the consensus problem for such self-organizing systems. Given these assumptions, the protocol establishes secure harmonized speed under Byzantine cyberattacks in AV platoons, where it utilizes signatures to generate and update speed information that was signed by all vehicles. Nonetheless, vehicles that are not part of the platoon can request and authenticate the information by verifying the signatures to ensure their alignment with the agreement.

The proposed ConsenCar protocol accurately detects and manages Byzantine behavior in platoons of AVs to ensure human the safety. By testing the algorithm on speed harmonization, its effectiveness in tolerating Byzantine faults was proven while minimizing overhead and message transmission delays. Additionally, testing assessed its ability to identify and isolate faulty vehicles, thereby ensuring the platoon's safe and smooth operation.

Simulation results show that the proposed protocol successfully meets all six design requirements and therefore efficiently secures the platoon against Byzantine cyberattacks. Moreover, when compared to other protocols, such as BFT-ARM and BChain, the design of ConsenCar introduces improved performance with minimum delays and communication overhead specifically for AV platooning applications. As such, the findings emphasize that the protocol is a feasible solution to accurately detect and eliminate Byzantine behavior in platoons of AVs.

6.2 Future Directions

In reality, a single failure cannot be tolerated when human safety is involved. As discussed in Chapters 2-3, platoon operations face multiple challenges in the presence of external constraints and critical failures. For this reason, testing the system in real-world scenarios allows for a comprehensive evaluation of its performance, reliability, and security under different conditions, such as varying network connectivity and environmental factors in the presence of a Byzantine attack. Hence, it allows for the identification of any potential flaws or limitations that must be addressed before deploying the system on a larger scale.

Looking ahead, the split platoon operation could be improved by introducing real-time behavioral analysis to safely eliminate vehicles whose behavior is less stable or less predictable. In situations where a faulty vehicle within a platoon refuses to leave, the implementation of a split function becomes crucial to safeguard the security and integrity of the entire platooning system. To achieve this, requests for compliance may be initiated by the platoon leader or authoritative nodes. These requests are directed towards the faulty vehicle to adhere to the platooning protocols and voluntarily leave the platoon. To allow for the possibility of rectification, a grace period could be granted to the faulty vehicle. During this period, the platoon leader and authoritative nodes closely monitor the behavior of the faulty vehicle to evaluate its compliance with the required protocols. The grace period serves as an opportunity for the operator to address the faults and rectify the situation. If the faulty vehicle persists in its refusal to leave the platoon after the grace period ends, the platoon leader shall collaborate with authoritative nodes to enforce its removal from the platoon. Various measures can be employed, such as temporarily disabling its access to platooning benefits, restricting its communication privileges within the platoon, or employing other methods to isolate it from the rest of the platoon. Once the faulty vehicle is successfully removed from the platoon, the split platoon function comes into effect. The remaining vehicles within the platoon undergo a process of reconfiguration and adaptation. They form new subgroups or adjust their platooning parameters to ensure the continuity of platoon operations. This enables the platoon to adapt to the changes in its composition and effectively compensate for the absence of the faulty vehicle.

Furthermore, a reputation system can establish connections with authoritative nodes for continuous monitoring and to properly punish misbehaving nodes. These nodes should have the power to take actions that discourage or penalize behaviors that compromise security. To achieve this goal, one approach is to implement consensus-based penalties to collectively agree to penalize misbehaving nodes. This can be achieved by applying consensus rules that result in rejecting or ignoring transactions or blocks propagated by the misbehaving node. By reaching a consensus on excluding or penalizing the misbehaving node, the authoritative nodes can effectively restrict its participation and influence within the network. Another method is the use of staking or collateral mechanisms. In such systems, nodes are required to lock a certain amount of value or tokens as collateral. If a node is found to be misbehaving or violating the rules of the network, the authoritative nodes can confiscate or slash a portion of the misbehaving node's stake as a form of punishment. This economic disincentive encourages nodes to act by the network's rules and discourages malicious behavior. Nonetheless, misbehaving nodes can also receive negative reputation scores, and authoritative nodes can collectively agree to impose penalties or restrictions on nodes with low reputation scores. These penalties may include reduced privileges, increased validation requirements, or temporary exclusion from participating in the network. In extreme cases of severe misbehavior or malicious activity, authoritative nodes may decide to temporarily or permanently exclude misbehaving nodes from the network. This can be achieved by banning the IP address or public key of the ode from participating in the network or by blacklisting the node in a distributed reputation database. Exclusion from the network serves as a strong deterrent and protects its security and integrity.

Lastly, it is important to acknowledge and understand the human factors associated with the operation of the algorithm under normal conditions and Byzantine behavior. Investigating these factors is crucial for optimizing the user experience and addressing any potential challenges or limitations that may arise during operation. By studying these factors, it can be ensured that the platooning system is user-friendly and capable of handling critical scenarios effectively in its environment without sudden human intervention, which may disrupt AV platoon operations.

References

[1] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[2] L. Lamport. The weak byzantine generals problem. *J. ACM*, 30(3):668–676, jul 1983.

[3] Mohammadreza Zohrevandi, Tanja Vos, Hajo A Reijers, and Wil MP van der Aalst. The byzantine generals problem for blockchain. In *Business Process Management*, pages 255–272. Springer, 2018.

[4] Gilly Leshed and Andrew Monk. The governance of blockchain systems: Mapping governance strategies and potential for disruption. *First Monday*, 22(12), 2017.

[5] Carlo Pedretti. *The Codex Atlanticus of Leonardo da Vinci: A Catalogue of its Newly Restored Sheets*, pages 125–126. Johnson Reprint Corporation, New York, New York, 1978.

[6] Ralph R Teetor. Speed control device for resisting operation of the accelerator, Aug 1950.

[7] Hans P. Moravec. *The Stanford Cart and the CMU Rover*, page 407. Springer New York, New York, NY, 1990. https://doi.org/10.1007/978-1-4613-8997-2_30.

[8] Andreas Geiger. *Probabilistic Models for 3D Urban Scene Understanding from Movable Platforms*. Schriftenreihe / Institut für Mess- und Regelungstechnik, Karlsruher Institut für Technologie. KIT Scientific Publishing, 2014. https://books.google.ca/books?id=UMd5Tuc6tFEC.

[9] Evan Ackerman. Nutonomy to test world's first fully autonomous taxi service in singapore this year, April 2016. https://spectrum.ieee.org/nutonomy-to-launch-worlds-first-fully-autonomous-taxi-service-in-singapore-this-year.

[10] Sae levels of driving automation™ refined for clarity and international audience, May 2021. https://www.sae.org/blog/sae-j3016-update.

[11] Syed Sarmad Shah, Asad Waqar Malik, Anis U. Rahman, Sohail Iqbal, and Samee U. Khan. Time barrier-based emergency message dissemination in vehicular ad-hoc networks. *IEEE Access*, 7:16494–16503, 2019.

[12] Muhammad Rizwan Ghori, Kamal Z. Zamli, Nik Quosthoni, Muhammad Hisyam, and Mohamed Montaser. Vehicular ad-hoc network (vanet): Review. *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, 2018.

[13] Hong Zhang and Jie Li. Dynamical topology analysis of vanet based on complex networks theory. *Cybernetics and Information Technologies*, 14, 12 2014.

[14] James Anderson and et al. *Autonomous Vehicle Technology: A Guide for Policymakers*. Rand Corporation, Santa Monica, CA, 2016.

[15] Sae j3016: Taxonomy and definitions for terms related to onroad motor vehicle automated driving systems, 2014. https://www.sae.org/standards/content/j3016_201401/.

[16] PWC. Connected car study 2015: Racing ahead with autonomous cars and digital innovation, 2015.

[17] Shane Underwood. *Automated, Connected, and Electric Vehicle Systems: Expert Forecast and Roadmap for Sustainable Transportation*. Artech House, 2014.

[18] Matteo Bertoncello and Dominik Wee. Ten ways autonomous driving could redefine the automotive world. *McKinsey & Company*, 2015.

[19] Allison Lee Palmer. *Leonardo da Vinci: A Reference Guide to His Life and Works*, pages 75–77. Significant Figures in World History. Rowman & Littlefield Publishers, 2018. https://books.google.ca/books?id=-tVyDwAAQBAJ.

[20] Anthony M. Townsend. *Ghost Road: Beyond the Driverless Car*. W. W. Norton, 2020. https://books.google.ca/books?id=ZqK6DwAAQBAJ.

[21] Marc E Cook. Autopilot basics, Apr 2016. https://www.aopa.org/training-and-safety/students/crosscountry/special/autopilot-basics.

[22] Gp Capt K S Mathur. *Fiber Optics & Aviation : Integration*. Blue Rose Publishers, 2021. https://books.google.ca/books?id=9_JVEAAAQBAJ.

[23] Weisong Shi and Liangkai Liu. *Computing Systems for Autonomous Driving*. Springer International Publishing, 2021. https://books.google.ca/books?id=9ch0EAAAQBAJ.

[24] Ian Chow-Miller. *How Self-Driving Cars Work*. Everyday STEM. Cavendish Square Publishing LLC, 2018. https://books.google.ca/books?id=z9JoDwAAQBAJ.

[25] Norman Bell Geddes. *Magic Motorways*. Random House, 1940. https://books.google.ca/books?id=PTMFAAAAMAAJ.

[26] Jack Challoner. *1001 Inventions That Changed the World*. 1001 Series. Thunder Bay Press, 2022. https://books.google.ca/books?id=D_Q1EAAAQBAJ.

[27] Ahmed Nabil Belbachir. *Smart Cameras*. Springer US, 2009. https://books.google.ca/books?id=it5W3f7yqAgC.

[28] Levent Guvenc, Sheng Zhu, Sukru Yaren Gelbal, and Bilin Aksun-Guvenc. *Autonomous Road Vehicle Path Planning and Tracking Control*. IEEE Press Series on Control Systems Theory and Applications. Wiley, 2021. https://books.google.ca/books?id=G4BTEAAAQBAJ.

[29] Nils John Nilsson. *The Quest for Artificial Intelligence*. https://books.google.ca/books?id=nUJdAAAAQBAJ, year=2009, publisher=Cambridge University Press.

[30] Mark H Lytle. *The All-Consuming Nation: Chasing the American Dream Since World War II*. Oxford University Press, 2021. https://books.google.ca/books?id=muFDEAAAQBAJ.

[31] Rudolf Schreiner. Managing security in intelligent transport systems. *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, 2015.

[32] S.E. Shladover. Path at 20—history and major milestones. *IEEE Transactions on Intelligent Transportation Systems*, 8(4):584–592, 2007.

[33] Neville A. Stanton, Steven Landry, Giuseppe Di Bucchianico, and Andrea Vallicelli. *Advances in Human Aspects of Transportation: Proceedings of the AHFE 2016 International Conference on Human Factors in Transportation, July 27-31, 2016, Walt Disney World®, Florida, USA*. Advances in Intelligent Systems and Computing. Springer International Publishing, 2016. https://books.google.ca/books?id=-2ABDgAAQBAJ.

[34] Mark Garrett. *Encyclopedia of Transportation: Social Science and Policy*. SAGE Publications, 2014. https://books.google.ca/books?id=vnpNBAAAQBAJ.

[35] Michael Fallon. *Self-Driving Cars: The New Way Forward*. Nonfiction — Young Adult. Lerner Publishing Group, 2018. https://books.google.ca/books?id=kvFjDwAAQBAJ.

[36] Bhoopathi Rapolu. *The Race for Work: Escape Automation, Transform Your Career and Thrive in the Second Machine Age*. Bhoopathi Rapolu, 2019. https://books.google.ca/books?id=p6FhDwAAQBAJ.

[37] Noon Hussein and Ahmed Massoud. Electric vehicle fast chargers: Futuristic vision, market trends and requirements. In *2019 2nd International Conference on Smart Grid and Renewable Energy (SGRE)*, pages 1–6, 2019.

[38] Edward Niedermeyer. *Ludicrous: The Unvarnished Story of Tesla Motors*. BenBella Books, 2019. https://books.google.ca/books?id=h_SADwAAQBAJ.

[39] National Center for Statistics and Analysis. Traffic Safety Facts 2020: A Compilation of Motor Vehicle Crash Data. Technical Report DOT HS 813 375, National Highway Traffic Safety Administration, October 2022.

[40] Andreas Herrmann, Walter Brenner, and Rupert Stadler. *Autonomous Driving: How the Driverless Revolution will Change the World*. Emerald Publishing Limited, 2018. https://books.google.ca/books?id=J_xQDwAAQBAJ.

[41] Stephen J.A. Ward. *Handbook of Global Media Ethics*. Springer International Publishing, 2021. https://books.google.ca/books?id=6QBBEAAAQBAJ.

[42] Felix Dodds, Carolina Duque Chopitea, and Ranger Ruffins. *Tomorrow's People and New Technology: Changing How We Live Our Lives*. Taylor & Francis, 2021. https://books.google.ca/books?id=cn08EAAAQBAJ.

[43] M. Roach. *The History of Speed*. Simon & Schuster UK, 2020. https://books.google.ca/books?id=-0HLDwAAQBAJ.

[44] Regulation (eu) 2019/2144 of the european parliament and of the council of november 2019, Nov 2019. https://www.legislation.gov.uk/eur/2019/2144.

[45] Muhammad Naeem Tahir, Pekka Leviäkangas, and Marcos Katz. Connected vehicles: V2v and v2i road weather and traffic communication using cellular technologies. *Sensors*, 22(3):1142, 2022.

[46] Mahmood A Al-Shareeda, Mohammed Anbar, Selvakumar Manickam, and Ali A Yassin. Vppcs: Vanet-based privacy-preserving communication scheme. *IEEE Access*, 8:150914–150928, 2020.

[47] Wellington Lobato Junior, Denis Rosário, Eduardo Cerqueira, Leandro A Villas, and Mario Gerla. A game theory approach for platoon-based driving for multimedia transmission in vanets. *Wireless Communications and Mobile Computing*, 2018, 2018.

[48] Nivedita Kadam and Raja Sekhar Krovi. Machine learning approach of hybrid ksvn algorithm to detect ddos attack in vanet. *International Journal of Advanced Computer Science and Applications*, 12(7), 2021.

[49] Shamsul Jamel Elias, Shahirah Hatim, Mohamad Darus, Shapina Abdullah, Jamaluddin Jasmis, R.Badlishah Ahmad, and Adam Wong Yoon Khang. Congestion control in vehicular adhoc network: A survey. *Indonesian Journal of Electrical Engineering and Computer Science*, 13:1280–1285, 03 2019.

[50] S. Ellwanger and E. Wohlfarth. Truck platooning application. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 966–971, 2017.

[51] Zhaojun Lu, Gang Qu, and Zhenglin Liu. A survey on recent advances in vehicular network security, trust, and privacy. *IEEE Transactions on Intelligent Transportation Systems*, 20(2):760–776, 2019.

[52] Amrita Ghosal, Sang Uk Sagong, Subir Halder, Kalana Sahabandu, Mauro Conti, Radha Poovendran, and Linda Bushnell. Truck platoon security: State-of-the-art and road ahead. *Computer Networks*, 185:107658, 2021.

[53] Vic Harkness, Joel Clark, Rich Perry, Terry Ip, and Imran Mughal. Future threats to its networks and cav infrastructure, 2020. https://www.f-secure.com/content/dam/f-secure/en/consulting/our-thinking/collaterals/digital/f-secureLABS_tlp-white-lazarus-threat-intel-report.pdf.

[54] Rasheed Hussain, Jooyoung Lee, and Sherali Zeadally. Trust in vanet: A survey of current solutions and future research opportunities. *IEEE Transactions on Intelligent Transportation Systems*, 22(5):2553–2571, 2021.

[55] Zhendong Wang, Haoran Wei, Jianda Wang, Xiaoming Zeng, and Yuchao Chang. Security issues and solutions for connected and autonomous vehicles in a sustainable city: A survey. *Sustainability*, 14(19):12409, 2022.

[56] Mohamed Nidhal Mejri, Jalel Ben-Othman, and Mohamed Hamdi. Survey on vanet security challenges and possible cryptographic solutions. *Vehicular Communications*, 1(2):53–66, 2014.

[57] Mohammed Saeed Al-kahtani. Survey on security attacks in vehicular ad hoc networks (vanets). In *2012 6th International Conference on Signal Processing and Communication Systems*, pages 1–9, 2012.

[58] Farhan Ahmad, Fatih Kurugollu, Chaker Abdelaziz Kerrache, Sakir Sezer, and Lu Liu. Notrino: A novel hybrid trust management scheme for internet-of-vehicles. *IEEE Transactions on Vehicular Technology*, 70(9):9244–9257, 2021.

[59] Hao Hu, Rongxing Lu, Zonghua Zhang, and Jun Shao. Replace: A reliable trust-based platoon service recommendation scheme in vanet. *IEEE Transactions on Vehicular Technology*, 66(2):1786–1797, 2017.

[60] Kai Li, Lingyun Lu, Wei Ni, Eduardo Tovar, and Mohsen Guizani. Cooperative secret key generation for platoon-based vehicular communications. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6, 2019.

[61] Ankur Sarker, Chenxi Qiu, and Haiying Shen. Quick and autonomous platoon maintenance in vehicle dynamics for distributed vehicle platoon networks. In *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 203–208, 2017.

[62] Mingshun Sun, Ali Al-Hashimi, Ming Li, and Ryan Gerdes. Impacts of constrained sensing and communication based attacks on vehicular platoons. *IEEE Transactions on Vehicular Technology*, 69(5):4773–4787, 2020.

[63] Jesty Santhosh and Sriram Sankaran. Defending against sybil attacks in vehicular platoons. In *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6, 2019.

[64] Roghieh A. Biroon, Zoleikha Abdollahi Biron, and Pierluigi Pisu. False data injection attack in a platoon of cacc: Real-time detection and isolation with a pde approach. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):8692–8703, 2022.

[65] Pengfei Zhu, Konglin Zhu, and Lin Zhang. Security analysis of lte-v2x and a platooning case study. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 532–537, 2020.

[66] Feng Jiang, Buren Qi, Tianhao Wu, Konglin Zhu, and Lin Zhang. Cpss: Cp-abe based platoon secure sensing scheme against cyber-attacks. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3218–3223, 2019.

[67] Seyhan Ucar, Sinem Coleri Ergen, and Oznur Ozkasap. Ieee 802.11p and visible light hybrid communication based secure autonomous platoon. *IEEE Transactions on Vehicular Technology*, 67(9):8667–8681, 2018.

[68] Nabila Bermad, Salah Zemmoudj, and Mawloud Omar. 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)*, pages 1–6, 2019.

[69] Roberto Merco, Zoleikha Abdollahi Biron, and Pierluigi Pisu. Replay attack detection in a platoon of connected vehicles with cooperative adaptive cruise control. In *2018 Annual American Control Conference (ACC)*, pages 5582–5587, 2018.

[70] Chien-Ming Chen, Bin Xiang, Yining Liu, and King-Hang Wang. A secure authentication protocol for internet of vehicles. *IEEE Access*, 7:12047–12057, 2019.

[71] Fangyi Wan, Ting Ma, Yi Hua, Bin Liao, and Xinlin Qing. Secure distributed estimation under byzantine attack and manipulation attack. *Engineering Applications of Artificial Intelligence*, 116:105384, 2022.

[72] Philipp Kremer, Ipsita Koley, Soumyajit Dey, and Sangyoung Park. State estimation for attack detection in vehicle platoon using vanet and controller model. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8, 2020.

[73] Yaodan Hu, Haoqi Shan, Raj Gautam Dutta, and Yier Jin. Protecting platoons from stealthy jamming attack. In *2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, pages 1–6, 2020.

[74] Dan Zhang, Ye-Ping Shen, Si-Quan Zhou, Xi-Wang Dong, and Li Yu. Distributed secure platoon control of connected vehicles subject to dos attack: Theory and application. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(11):7269–7278, 2021.

[75] Shahida Malik, Praveen Bandi, and Weiqing Sun. An experimental study of denial of service attack against platoon of smart vehicles. In *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*, pages 23–30, 2021.

[76] Duc Tran Le, Khanh Quoc Dang, Quyen Le Thi Nguyen, Soha Alhelaly, and Ammar Muthanna. A behavior-based malware spreading model for vehicle-to-vehicle communications in vanet networks. *Electronics*, 10(19), 2021.

[77] Asad Waqar Malik, Zahid Anwar, and Anis U. Rahman. A novel framework for studying the business impact of ransomware on connected vehicles. *IEEE Internet of Things Journal*, pages 1–1, 2022.

[78] Hyogon Kim and Taeho Kim. Vehicle-to-vehicle (v2v) message content plausibility check for platoons through low-power beaconing. *Sensors*, 19(24), 2019.

[79] Noon Hussein and Armstrong Nhlabatsi. Living in the dark: Mqtt-based exploitation of iot security vulnerabilities in zigbee networks for smart lighting control. *IoT*, 3(4):450–472, 2022. https://www.mdpi.com/2624-831X/3/4/24.

[80] Kai Li, Lingyun Lu, Wei Ni, Eduardo Tovar, and Mohsen Guizani. Secret key agreement for data dissemination in vehicular platoons. *IEEE Transactions on Vehicular Technology*, 68(9):9060–9073, 2019.

[81] Sean Joe Taylor, Farhan Ahmad, Hoang Nga Nguyen, Siraj Ahmed Shaikh, David Evans, and David Price. Vehicular platoon communication: Cybersecurity threats and open challenges. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 19–26, 2021.

[82] Alberto Petrillo, Antonio Pescapé, and Stefania Santini. A collaborative approach for improving the security of vehicular scenarios: The case of platooning. *Computer Communications*, 122:59–75, 2018.

[83] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *Journal of the ACM*, 27(2):228–234, 1980.

[84] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[85] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[86] Shu-Ching Wang, Ya-Jung Lin, , and Kuo-Qin Yan and. Reaching byzantine agreement underlying vanet. *KSII Transactions on Internet and Information Systems*, 13(7):3351–3368, July 2019.

[87] Lun Li, Jiqiang Liu, Lichen Cheng, Shuo Qiu, Wei Wang, Xiangliang Zhang, and Zonghua Zhang. Creditcoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 19(7):2204–2220, 2018.

[88] Huiye Liu, Chung-Wei Lin, Eunsuk Kang, Shinichi Shiraishi, and Douglas M. Blough. A byzantine-tolerant distributed consensus algorithm for connected vehicles using proof-of-eligibility. New York, NY, USA, 2019. Association for Computing Machinery.

[89] Jin-Hua Chen, Min-Rong Chen, Guo-Qiang Zeng, and Jia-Si Weng. Bdfl: A byzantine-fault-tolerance decentralized federated learning method for autonomous vehicle. *IEEE Transactions on Vehicular Technology*, 70(9):8639–8652, 2021.

[90] Le Xia, Yao Sun, Rafiq Swash, Lina Mohjazi, Lei Zhang, and Muhammad Ali Imran. Smart and secure cav networks empowered by ai-enabled blockchain: The next frontier for intelligent safe driving assessment. *IEEE Network*, 36(1):197–204, 2022.

[91] Siming Wang, Dongdong Ye, Xumin Huang, Rong Yu, Yongjian Wang, and Yan Zhang. Consortium blockchain for secure resource sharing in vehicular edge computing: A contract-based approach. *IEEE Transactions on Network Science and Engineering*, 8(2):1189–1201, 2021.

[92] R. Ramaguru, M. Sindhu, and M. Sethumadhavan. Blockchain for the internet of vehicles. *Communications in Computer and Information Science*, page 412–423, 2019.

[93] Mumin Cebe, Enes Erdin, Kemal Akkaya, Hidayet Aksu, and Selcuk Uluagac. Block4forensic: An integrated lightweight blockchain framework for forensics applications of connected vehicles. *IEEE Communications Magazine*, 56(10):50–57, 2018.

[94] Matthew Wagner and Bruce McMillin. Cyber-physical transactions: A method for securing vanets with blockchains. In *2018 IEEE 23^{r}d Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 64–73, 2018.

[95] Sisi Duan, Hein Meling, Sean Peisert, and Haibin Zhang. Bchain: Byzantine replication with high throughput and embedded reconfiguration. *Lecture Notes in Computer Science*, page 91–106, 2014.

[96] Aviv Zohar and Yonatan Sompolinsky. Blockchain consensus algorithms in the wild. *arXiv preprint arXiv:1503.08837*, 2015.

[97] Antonio-Javier Fernández-Ahumada, Raúl Morales-Téllez, and Pedro García-Teodoro. Performance analysis of consensus protocols for permissioned blockchains. *IEEE Access*, 7:12776–12785, 2019.

[98] Xueping Li, Ping Jiang, Yang Chen, Xiaoyong Luo, and Xiaolin Gui. Blockchain-based consensus mechanisms: A survey. *IEEE Transactions on Industrial Informatics*, 16(6):4209–4229, 2020.

[99] M. Wegner, W. Xu, R. Kapitza, and L. Wolf. Byzantine consensus in vehicle platooning via inter-vehicle communication. Technical report, Humboldt University Berlin, Berlin, Germany, 3 2016.