**VEHICLE TO X COMMUNICATION**

**Submitted by:**

**PAWAN CHANDRA**

**885887810**

**A logo of a university

Description automatically generated**

**California State University, Fullerton**

**CPSC-597**

**Fall 2023**

**Advisor: Dr. Sampson Akwafuo**

***Reviewer: if you have one, name goes here***

**Department of Computer Science**

**California State University Fullerton**

****

**Department of Computer Science**

This project has been satisfactorily demonstrated and is of suitable form

This project report is acceptable in partial fulfilment of the requirements for the award of Master of Science degree in Computer Science.

|  |  |  |
| --- | --- | --- |
| **Vehicle to X Communication** | | |
| Project Title | | |
| **Pawan Chandra** |
| Student Name |
| **Dr. Sampson Akwafuo** |
| Advisor's Name |
|  | |
| Advisor's signature | Date |
| **Your reviewer name goes here** |
| Reviewer's name |
|  | |
| Reviewer's signature | Date |

**Table of Contents**

[***Table of Figures 4***](#_heading=h.1fob9te)

[***Abstract 5***](#_heading=h.3znysh7)

[***1. Introduction 1***](#_heading=h.2et92p0)

[**1.1 Motivation 2**](#_heading=h.nxmgkola95cx)

[**1.2 Objectives 3**](#_heading=h.3dy6vkm)

[***2. Literature Review 6***](#_heading=h.4d34og8)

[***3. Methodology 8***](#_heading=h.26in1rg)

[***4. Results and Discussion 12***](#_heading=h.44sinio)

[**4.1 Network 12**](#_heading=h.8wxh105zk82s)

[**4.2 Camera and Artificial Intelligence 12**](#_heading=h.z337ya)

[**4.3 Arduino Microcontroller 12**](#_heading=h.3j2qqm3)

[**4.3 Others 13**](#_heading=h.5k3gsn4bn4yb)

[**Discussion 13**](#_heading=h.ko7knzi1x0r6)

[***5. Conclusion 14***](#_heading=h.1y810tw)

[**5.1 Future Work 14**](#_heading=h.4i7ojhp)

[**5.2 Limitations 14**](#_heading=h.2xcytpi)

[5.2.1 Getting Location 14](#_heading=h.lz4hbl1zzfn8)

[5.2.2 Processing Power 14](#_heading=h.4dzxw398mzd3)

[5.2.3 Funding 15](#_heading=h.dnqygbrew6zk)

[5.2.4 Knowledge 15](#_heading=h.y72t2ov075n5)

[***References 16***](#_heading=h.1ci93xb)

# Table of Figures

[Figure 1 – Understanding Levels of Autonomous Vehicles 1](https://docs.google.com/document/d/1UMv1v3luNGYa46WBBGLQgjIKwis-8Qn7/edit#heading=h.35nkun2)

[Figure 2 – V2V V2I V2P and V2X Communication 3](https://docs.google.com/document/d/1UMv1v3luNGYa46WBBGLQgjIKwis-8Qn7/edit#heading=h.1ksv4uv)

[Figure 3 – Project Layout and Approach Diagram](https://docs.google.com/document/d/1UMv1v3luNGYa46WBBGLQgjIKwis-8Qn7/edit#heading=h.1ksv4uv) 5

[Figure 2 – Optical Encoder used to calculate speed](https://docs.google.com/document/d/1UMv1v3luNGYa46WBBGLQgjIKwis-8Qn7/edit#heading=h.1ksv4uv) 14

# Abstract

While there are many companies racing to produce self-driving cars, the incentive to update level 0 and level 1 autonomous vehicles are close to none. The cost of uprooting the infrastructure and implementing technologies such as radars, lidar, cameras, etc along with implementing the computer needed to process data far outweigh the benefits. It is much more cost effective to purchase a vehicle with update and integrated components versus upgrading an old car. With that being said, there some components that are worth upgrading in older non autonomous vehicles that allow drivers to benefit from the data collected by autonomous vehicles. According to IEEE, Vehicle-to-vehicle (V2V) communication is a technology that enables vehicles to communicate with each other without relying on third-party networks like cellular networks. This communication is what allows levels 3 and above autonomous cars to share data such as traffic flow, road hazards, traffic light status and more. In this paper I will discuss how the implementation of Vehicle to X communication can be implemented on a regular non autonomous vehicle and attempt to show the benefits of having just that.

# Introduction

The idea of autonomous vehicles is to some a scary thought. Plagued with ethical dilemmas, safety concerns, and life and death situations where the fatal conditions all come down to the team of engineers who implemented, hopefully, the correct algorithm, infrastructure, security protocols, hardware/technology etc.[1]. There is also an issue of different levels of autonomous vehicles on the road, each running their own separate software and firmware and manufactures using perhaps different standards.

There are 6 different levels of autonomous vehicles ranging from levels 0 – 5 [2, p. 6], [3]. Level 0 autonomy refers to a vehicle that does not have any autonomous features or capabilities. In other words, the vehicle is completely controlled by the driver, and all driving tasks such as steering, braking, and accelerating must be performed by the driver. Level 1 uses advanced driver assistance systems (ADAS) for things such as assisted steering, braking, **or** acceleration[4]. Adaptive or smart cruise control is an example of level 1. Level 2 involves the usage of ADAS that can take over steering, acceleration, **and** braking. Think Tesla’s Autopilot feature.

A diagram of a car steering wheel

Description automatically generated

***Figure 1 – Understanding Levels of Autonomous Vehicles***

Level 3 and above involves less and less human intervention. This is made possible with the help of modern technology such as radar, LiDAR, cameras etc., collecting and sharing data between all capable vehicles using Vehicle to Vehicle (V2V) communication technology to relay important data such as road hazards or slowdowns[3], [5]. Level 4 vehicles can operate autonomously in predefined geographic areas or under specific conditions, such as on certain well-mapped and controlled roads or in particular weather conditions. These vehicles can handle all aspects of driving, including acceleration, braking, steering, and monitoring the environment, without human intervention, but only within their defined operational design domain (ODD). If the vehicle encounters a situation or environment that falls outside its ODD or encounters a technical issue, it will typically prompt the driver to take control. In Level 4, a human driver may still be necessary for certain situations.

Level 5 autonomous vehicles are designed to operate autonomously in all conditions and environments, without any human intervention required. They do not have a steering wheel or pedals, and there is no need for a human driver to be present. These vehicles are theoretically capable of handling any driving scenario, including complex urban environments, adverse weather conditions, and off-road terrain.

## 1.1 Motivation

In late 2020 Waymo, a self-driving autonomous vehicle company owned, released their autonomous level 4 taxis to the public for usage[11]. In January of 2023, Mercedes had gotten approved to release their first level 3 autonomous vehicles in the United States[6]. Though this may sound like great news to some, what is to happen with older vehicles with outdated technologies. Like many cell phones, when a new device comes to the market and an upgrade is due, our only options are to replace our old device with the new due to the non-modular architecture and infrastructure.

This creates a massive e-waste and shortage of valuable and limited resources at the cost of unnecessary upgrade. There are many variables to consider and unfortunately technologies do not consider reuse, upgrade, or modularity causing old vehicles to become obsolete. Currently, there are no incentives for upgrading your vehicles due to the cost and vast differences in the technologies used between level 0 and level 3.

A car on a crosswalk

Description automatically generated

***Figure 2 – V2V, V2I, V2P, and V2X Communication***

Instead of creating a fully automated modular system, we can bridge the gap between the autonomous and non-autonomous vehicles by utilizing data obtained and processed from levels 3 and above autonomous vehicles and relaying that information to all vehicles nearby. This method will utilize the network communication device within the vehicles and infrastructure also known as Vehicle to X communication or V2X.

## 1.2 Objectives

Vehicle communication, which involves vehicles sharing information with each other and with infrastructure is a critical component of future transportation systems, particularly in the context of connected and autonomous vehicles (CAVs). However, there are several challenges and problems associated with vehicle communication:

1. **Cybersecurity Concerns:** As vehicles become more interconnected, they become potential targets for cyberattacks. Hackers could exploit vulnerabilities in communication systems to gain control of vehicles, disrupt traffic, or steal sensitive data.
2. **Privacy Issues:** Collecting and sharing data among vehicles and infrastructure could raise privacy concerns. Individuals may be uncomfortable with the idea of their vehicles constantly transmitting information about their location and driving habits.
3. **Standardization:** The development of common communication standards and protocols is essential to ensure interoperability between vehicles from different manufacturers. Achieving such standardization can be challenging.
4. **Reliability and Redundancy:** Vehicle communication systems must be highly reliable, especially in safety-critical situations. Failures or interruptions in communication could have severe consequences.
5. **Scalability:** As the number of connected vehicles increases, the communication infrastructure must be able to handle the growing volume of data and messages without congestion or latency issues.
6. **Spectrum Allocation:** The radio frequency spectrum used for vehicle communication is a limited and valuable resource. Allocating sufficient spectrum for V2X communication while avoiding interference with other wireless technologies is a complex task.
7. **Infrastructure Costs:** Building and maintaining the necessary infrastructure to support V2X communication, such as roadside units and communication networks, can be costly and require significant investment.
8. **Geographical Coverage:** Ensuring consistent communication coverage across urban, suburban, and rural areas can be challenging. Remote or sparsely populated regions may have limited access to V2X communication infrastructure.
9. **Data Management:** Managing the vast amount of data generated by connected vehicles and ensuring its secure storage, processing, and analysis present significant challenges.
10. **Latency:** In safety-critical applications, such as collision avoidance systems, low-latency communication is essential. Ensuring that messages are delivered quickly and reliably can be difficult, especially in congested networks.

Though there are many concerns with communication devices, the primary objective of this project is to focus on:

* Standardization
* Reliability and Redundancy
* Data Management
* Cybersecurity

A screenshot of a computer screen

Description automatically generated

***Figure******3 – Project Layout and Approach Diagram***

In Figure 3, it accurately describes the testing environment I plan to implement. Here you can see 4 different cases I would like to tackle such as pedestrian crossing, road hazard detection, traffic light awareness, and communication from one server to another given the speed of traffic. In all the 4 different cases, the blue car, which is the non-autonomous vehicle, will be the focus as the green car communicates with the servers and surroundings.

# 

# Literature Review

According to NSTSA.gov, “Vehicle-to-vehicle (V2V) communication enables vehicles to wirelessly exchange information about their speed, location, and heading[5]. The technology behind V2V communication allows vehicles to broadcast and receive omni-directional messages (up to 10 times per second), creating a 360-degree “awareness” of other vehicles in proximity”. Essentially this exchange of data does not require all vehicles to have radar, lidar, cameras, or other advanced technology. If the vehicle ahead has the capable technologies, then it can relay messages to nearby vehicles acting as their vision.

However like most communication devices, there are issues that come with it such as packet loss due to interference and collisions in the congested network. Another challenge would be scalability as noted by Shi, Sung “Recent studies on the performance of IEEE 802.11 p-based V2V communication system have identified its scalability issue with high density of vehicles, e.g., multi-lane highway scenarios”[7]. M. Muhammad and G. Safdar explored using cellular networks as part of the V2V called Cellular-V2X (C–V2X) networks outside of the IEEE 802.11p standard running in the 5.9 GHz frequency band[9].

They go on to explain “C– V2X includes two modes: longer-range, higher latency. communication via the cellular network known as C–V2N, and low latency, direct communication referred to as C– V2V/I/P”[9]. The IEEE standard of using 5.9 GHz for vehicle communication is primarily referred to as DSRC or dedicated short range communication. The problem here is that the higher frequency is equal to shorter range communication. There isn’t an exact number in terms of distance as the radio frequency is not in a vacuum and faces many obstacles when sending and receiving.

The biggest obstacle would be not only range but the rate. The rate at which data is processed vs the speed of a vehicle approaching a pedestrian can be detrimental. Some ways to mitigate this might be installing many network devices near residential areas, by intersections, for improved connectivity. Shi and Sung elaborate stating the issue might be due to insufficient amount of radio spectrum allocated to V2V communication though with recent updates to 5 GHz and even 6 GHz networks readily available, this issue might be mitigated with advances in network technology.

Outside of network issues, the speed and mobility of the vehicle can also affect the performance of communication. *Cristina, et al. (2017) “Unfortunately, the challenging radio conditions derived from the mobility of vehicles, their relatively high speed with respect to pedestrians, the dynamic topology of vehicular wireless networks and its higher likelihood to produce inter-vehicular line-of-sight blockage are factors that pose significant challenges to be dealt with”*[8]*.*

Because the market and research for autonomous vehicles are still relatively early, currently there are no incentives going towards creating a system where non autonomous vehicles too can be autonomous with perhaps a hypothetical modular device. Another reason why current solutions are not being discussed is the cost. The cost of implementing newer gadgets such as lasers, radars, cameras, etc. into older vehicles might be too expensive, require uprooting the entire architecture, or even compromising the vehicle frame.

# Methodology

The project will require extensive planning from procurement to architecture/framework to hardware components and software capability/cross platforms. I will be following the waterfall methodology as I will be setting my own requirements and I have a clear idea on what steps I need to take to accomplish my goal and meet my objective. Waterfall can be effective for small projects with a limited scope and well-understood technology. In such cases, the overhead of iterative development and continuous feedback may not be necessary.

Software Requirements

Requirements for creating autonomous vehicles using a raspberry pi.

1. Real-time sensor data processing: The software should be able to efficiently process real-time data from various sensors, such as cameras, LiDAR, and ultrasonic sensors.
2. Object detection and recognition: The software should be capable of detecting and recognizing objects in the vehicle's surroundings, including vehicles, pedestrians, traffic signs, and traffic lights.
3. Path planning and navigation: The software should include algorithms for generating optimal paths and making navigation decisions based on the detected objects and environmental conditions.
4. Collision avoidance: The software should implement collision detection algorithms and take appropriate actions to avoid potential collisions with obstacles or other vehicles.
5. Localization and mapping: The software should incorporate algorithms for accurately estimating the vehicle's position and creating maps of the environment in real-time.
6. Lane detection and keeping: The software should include algorithms for detecting and tracking lanes on the road and keeping the vehicle within the designated lanes.
7. Human-machine interface: The software should provide a user-friendly interface for controlling and monitoring the autonomous vehicle, including features like status indicators, alerts, and manual override options.
8. Data logging and analysis: The software should have the capability to log sensor data, vehicle status, and decision-making processes for later analysis and debugging.
9. Redundancy and fault tolerance: The software should incorporate mechanisms for detecting failures and implementing backup systems to ensure safe operation in case of software or hardware malfunctions.
10. Integration with external systems: The software should be designed to integrate with other external systems, such as traffic management systems, vehicle-to-vehicle communication, and infrastructure networks, to enable cooperative and coordinated driving.

Communication Requirements:

1. Communication Protocols: The system should support standard communication protocols such as CAN (Controller Area Network) or Ethernet for exchanging information between various vehicle components.
2. Vehicle-to-Vehicle (V2V) Communication: The system should enable vehicles to communicate with each other to exchange information such as position, speed, acceleration, and intentions, allowing for cooperative driving and collision avoidance.
3. Vehicle-to-Infrastructure (V2I) Communication: The system should facilitate communication between vehicles and roadside infrastructure, such as traffic signals, road sensors, and toll booths, to provide information about road conditions, traffic congestion, and infrastructure-related alerts.
4. Data Exchange Format: The system should define a standardized format for data exchange between vehicles and infrastructure systems, ensuring interoperability and compatibility.
5. Latency and Reliability: The communication system should have low latency and high reliability to ensure timely and accurate information exchange between vehicles and infrastructure systems.
6. Range and Coverage: The system should provide sufficient range and coverage to facilitate communication over a wide area, enabling vehicles to exchange data even in challenging environments.
7. Bandwidth and Throughput: The communication system should offer adequate bandwidth and throughput to support the exchange of large volumes of data, including high-definition maps, sensor data, and real-time updates.
8. Quality of Service (QoS): The system should prioritize critical communication messages to ensure their timely delivery, especially for safety-critical applications such as collision warnings or emergency notifications.
9. Scalability: The communication system should be scalable to accommodate a growing number of vehicles and infrastructure systems, ensuring efficient and reliable communication in large-scale deployments.

The following subsections provide detail on the hardware and software that make up the environment needed to implement this project.

Hardware in the Environment

The following bulleted list provides a breakdown of the hardware anticipated for use in this project.

1. Raspberry Pi - series of small, single-board computers (SBCs) developed by the Raspberry Pi Foundation. The raspberry pi’s will be used on vehicles which will have 3 functions.
   1. Attach sensors to the GPIO.
   2. Used to create autonomous vehicles by processing data from said sensors.
   3. Used as communication to Send and Receive data.
   4. Traffic light communication (V2I).
   5. Personal Computer - There will be a main server, my laptop, where all data will get processed such as vehicle status, location coordinates, speed, error/log messages etc.
2. Craft Materials - In order to create an environment to test the vehicles, I will need to procure materials to create a track.

Software in the Environment

The following bulleted list provides a breakdown of the software anticipated for use in the project.

1. C++ & Python - Python will serve as the main programming language both for network-based communication (sending and receiving packets) and programming the raspberry pi vehicles to run on the given track.
2. MySQL - will handle the data, coordinates, boolean/error flags, and logs.
3. Visual Studio Code - I will most likely be using a lightweight IDE to code and handle all other tasks.
4. GitHub - for source control.

# Results and Discussion

## 4.1 Network

Testing and sending coordinates to a server using python was pretty straight forward and successful. Python has many network features and so creating a tcp client and server was simple. This also goes for using python on each of the hardware devices such as the vehicles, traffic signals, and mobile phones. The coordinates received were saved to a text file, which can be accessed from both the autonomous and non autonomous vehicles, this includes the live data from the traffic light and other hardware components. I used a boolean for the traffic light function

## 

## 4.2 Camera and Artificial Intelligence

Setting up the Camera took the majority of the time. Finding the correct camera with the largest wide angle view that also is supported by raspberry pi took a bit of trial and error. There was also the issue of mounting the camera at the best possible location on the vehicle as the view of the front of the car while capturing the lanes were pretty difficult. Next, programming the camera to first detect lanes required advanced knowledge of color and vector manipulation. After obtaining the best image, implementing a gray scale and hyper contrast, I was able to detect the lanes. This allowed the autonomous vehicle to stay inside the lane. Implementing Artificial intelligence after feeding it training data of basic traffic lights and stop signs, I was able to detect toy stop signs and streetlights with the camera I had already set up.

## 

## 4.3 Arduino Microcontroller

Using the arduino uno microcontroller, I controlled all the small movements from the motor and programmed the vehicles turning left, right, and the sharpness of the turn. Then connecting the raspberry pi to the arduino uno creating a master-slave configuration, I was able to send signals from the GPIO pins to the arduino board and control the turns. The camera would detect shifts when the lane begins to turn, causing the program to send signals to the controller to turn.

## 

## 4.3 Others

The cars and the traffic lights were controlled via the raspberry pi. The servers on each of the devices would send signals to the main server which was my desktop. The testing environment was my room.

## 

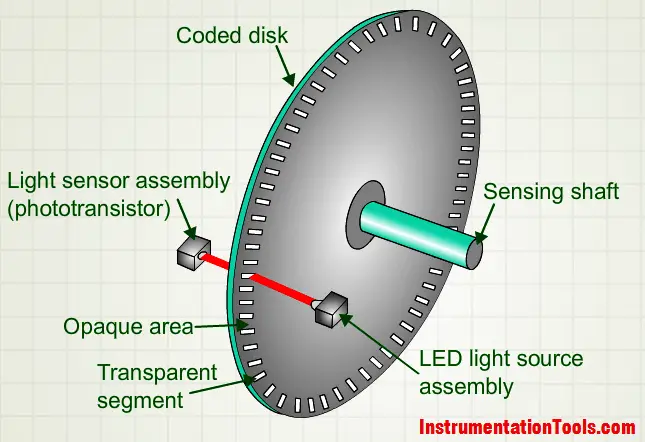
## Discussion

Due to my inability to obtain the location of the vehicles and other components, I was unable to complete most of the tests. The environment and all the components are in working order but without getting an exact coordinate of the vehicles relative to the test environment, it became much more difficult to produce the results for my project.

# Conclusion

## 5.1 Future Work

Future works include utilizing python and the latest version of opencv for easier API implementation as python is the better option for machine learning programming. Also changing the raspberry pi camera to a webcam where you have a wider field of vision can help detect objects much easier. With the two implementations mentioned, communication with the server will be faster and integration of python script with python object oriented code can streamline the process. Creating a car that can switch lanes would make for an ideal situation as creating a test environment would be a lot easier. The test environment could be a loop or a circuit with multiple lanes and having both autonomous and non autonomous vehicles running as you place obstacles on the track. Lastly, The most important option is to find ways to calculate distance from the starting point and obtain coordinates relative to that. One way to obtain coordinates is using an optical encoder. By using a an optical encoder we can calculate distance based off of speed. The distance can be used to get a coordinate.



***Figure 5 – Optical Encoder used to calculate speed***

## 5.2 Limitations

### 5.2.1 Getting Location

In order to obtain the location of a small scale vehicle, the options are very limited. The usage of a global position system of GPS cannot be feasible as the GPS depends on the number of satellites, latency, location, and interference. On top of that, GPS devices aren’t accurate enough to calculate distance due to its inconvenience. The other option was to use trigonometry by triangulation to calculate distance based on radio frequency and the distance between main server, start point, and vehicle location; however, this would be a project itself. I chose to use the stepper motor to get the coordinates of the autonomous vehicle but this makes it difficult for the non autonomous vehicle to interact with the traffic light as the location of traffic light must be hard defined/set.

### 5.2.2 Processing Power

The raspberry pi is a solid option for a small scale project, however due to the nature of this project, there were many times where the CPU utilization of the raspberry pi was at 100%. Being at 100% raises the temperature and slows down the computation needed to verify the object in front while also using sensors to drive and send data. This can be mitigated by using a raspberry pi 4 and above as the newer raspberry pi’s are a 64-bit processor and have a higher ram from factory. This will help lower the temperature and the CPU utilization.

### 5.2.3 Funding

During the time of the project, the raspberry pi’s were limited and therefore extremely overpriced. I was able to get my hands on older raspberry pi models however, as mentioned above, they were limited due to the processing power. One of the raspberry pi’s I bought ended up overheating and killing the processor during the installation of opencv. Also spending part of the budget on solder equipment, circuit boards, car models, 3D printing components etc.

### 5.2.4 Knowledge

In computer science, you are very limited to hardware knowledge. Things such as embedded systems, circuitry, microcontrollers, hardware components, and programming these components aren’t taught as part of the computer science program. In this project having some knowledge of the topics mentioned would save a lot of time during the planning, implementing, and testing phase. Knowing how to use semaphores, programming GPIO pins, and utilizing other features of a microcontroller would be ideal.

# 

# References

[1] “Autonomous Vehicles - The Nexus - Urbanism Next.” https://www.urbanismnext.org/technologies/autonomous-vehicles (accessed Sep. 17, 2023).

[2] “The 6 Levels of Vehicle Autonomy Explained | Synopsys Automotive.” https://www.synopsys.com/automotive/autonomous-driving-levels.html (accessed Sep. 17, 2023).

[3] “Vehicle-to-Vehicle Communication | NHTSA.” https://www.nhtsa.gov/technology-innovation/vehicle-vehicle-communication (accessed Sep. 17, 2023).

[4] S. Campbell *et al.*, “Sensor Technology in Autonomous Vehicles : A review,” in *2018 29th Irish Signals and Systems Conference (ISSC)*, Jun. 2018, pp. 1–4. doi: 10.1109/ISSC.2018.8585340.

[5] R. Jurgen, *V2V/V2I Communications for Improved Road Safety and Efficiency*. Warrendale, UNITED STATES: SAE International, 2012. Accessed: Sep. 17, 2023. [Online]. Available: http://ebookcentral.proquest.com/lib/fullerton/detail.action?docID=5341866

[6] “2024 Mercedes-Benz EQS Sedan and S-Class Will Debut Drive Pilot in the U.S.,” *Car and Driver*, Jan. 26, 2023. https://www.caranddriver.com/news/a42672470/2024-mercedes-benz-eqs-s-class-drive-pilot-autonomous-us-debut/ (accessed Sep. 17, 2023).

[7] L. Shi and K. W. Sung, “Spectrum Requirement for Vehicle-to-Vehicle Communication for Traffic Safety,” in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, May 2014, pp. 1–5. doi: 10.1109/VTCSpring.2014.7023107.

[8] W. Li, X. Wang, and B. Moran, “Wireless Signal Travel Distance Estimation Using Non-Coprime Wavelengths,” *IEEE Signal Process. Lett.*, vol. 24, no. 1, pp. 27–31, Jan. 2017, doi: 10.1109/LSP.2016.2632708.

[9] M. Muhammad and G. A. Safdar, “5G-based V2V broadcast communications: A security perspective,” *Array*, vol. 11, p. 100084, Sep. 2021, doi: 10.1016/j.array.2021.100084.

[10] S. Zeadally, J. Guerrero, and J. Contreras, “A tutorial survey on vehicle-to-vehicle communications,” *Telecommun. Syst.*, vol. 73, no. 3, pp. 469–489, Mar. 2020, doi: 10.1007/s11235-019-00639-8.

[11] “Waypoint - The official Waymo blog: Waymo is opening its fully driverless service to the general public in Phoenix,” Waypoint – The official Waymo blog. Accessed: Sep. 30, 2023. [Online]. Available: https://waymo.com/blog/2020/10/waymo-is-opening-its-fully-driverless.html