Case Study: Spoofing Attack On Autonomous Vehicles

Shihab Ud Doula
Matriculation No. 2190679

Project Work
Bachelor of Engineering - Electronic Engineering

HOCHSCHULE HAMM-LIPPSTADT

Main Advisor: **Prof. Dr. João Paulo Javidi da Costa**
Assistant Advisor: **Antonio Arlis Santos Da Silva**

© Main Advisor: **Prof. Dr. João Paulo Javidi da Costa**

© Assistant Advisor: **Antonio Arlis Santos Da Silva**

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor, **Prof. Dr.-Ing. João Paulo J. da Costa**, from the Department of Engineering at Hamm-Lippstadt University of Applied Sciences. His guidance and feedback throughout this project have been invaluable. I appreciate the academic and professional insights he provided, which have significantly contributed to the successful completion of this work.

I would also like to thank **Antonio Arlis Santos Da Silva** from the Department of Engineering for his readiness to assist during this project.

Finally, I extend my appreciation to the faculty members of the Department of Engineering at Hamm-Lippstadt University, whose instruction and support have been instrumental in my academic development.

Shihab Ud Doula

ABSTRACT

This report presents an in-depth analysis of spoofing attacks on autonomous vehicles, focusing on three primary types: GPS spoofing, Sensor spoofing, and Time spoofing. Autonomous vehicles rely heavily on accurate communication and sensor data to navigate and make decisions. Spoofing attacks pose a significant threat by providing false information, leading to potentially hazardous situations.

To explore these vulnerabilities, simulations were conducted using MATLAB to visualize the impact of these attacks on autonomous vehicle systems. The results demonstrate the severity of the disruptions caused by spoofing and highlight the importance of robust countermeasures. This report also discusses real-world case studies that illustrate the practical implications of such attacks and compares these with the simulation outcomes.

The findings underline the critical need for enhanced security protocols in autonomous vehicle communication systems. Recommendations for future research are provided, emphasizing the development of advanced detection and prevention mechanisms to safeguard against these threats

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Background

Autonomous vehicles (AVs) are revolutionizing the transportation industry by offering safer, more efficient, and intelligent mobility. These vehicles rely on systems such as GPS, time synchronization, and various sensors to navigate and make real-time decisions. However, this dependence makes them vulnerable to spoofing attacks, which can disrupt their operations. GPS spoofing can mislead navigation, time spoofing can create synchronization issues, and sensor spoofing can falsify environmental data. Such vulnerabilities raise serious safety concerns and highlight the importance of strengthening AV systems against spoofing threats.[12]

1.2 Problem Statement

Autonomous vehicle systems are highly dependent on data integrity for accurate decision-making. Spoofing attacks, which manipulate critical inputs like GPS, time, and sensors, pose significant risks to the safety and reliability of these systems. Current detection and response mechanisms are often insufficient to address these sophisticated attacks, leaving AVs susceptible to navigation errors, synchronization failures, and misinterpretation of their surroundings. This research aims to analyze these vulnerabilities and propose ways to mitigate them through controlled simulations.[1]

1.3 Objectives

The objectives of this project are to:

- Simulate GPS, time, and sensor spoofing attacks on AV systems.
- Assess the impact of these attacks on vehicle behaviour and decision-making.
- Explore methods to detect and mitigate spoofing attacks.
- Contribute to improving the security and robustness of AV systems.

1.4 Project Scope and Challenges

This project focuses on simulating three types of spoofing attacks—GPS, time, and sensor spoofing—using MATLAB to replicate real-world scenarios. The goal is to understand how these attacks affect AV systems and evaluate potential mitigation strategies. The study is limited to simulations due to time and resource constraints. One of the main challenges was ensuring the realism of the simulations while balancing computational feasibility. Interpreting the results in a practical context also required careful analysis and iterative improvements to the simulation design.[4]

1.5 Structure of the Report

This report is structured to provide a step-by-step analysis of the study. Chapter 2 reviews existing research on spoofing attacks, highlighting case studies to establish a foundation for the project. Chapter 3 explains the methodology used for simulating spoofing attacks and analyzing their impact. Chapter 4 presents the results of the simulations and discusses the observations. Chapter 5 provides a detailed discussion of the findings, including challenges and future possibilities. Finally, Chapter 6 concludes the report by summarizing key insights and offering recommendations for enhancing AV security.

Chapter 2

LITERATURE REVIEW

2.1 Overview of Spoofing Attacks

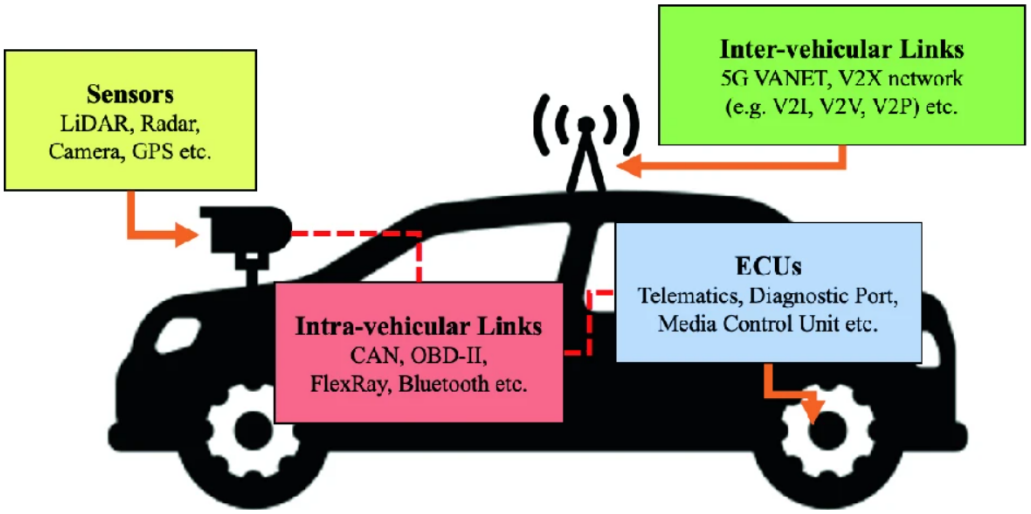

Figure 2.1: Attack surfaces of autonomous vehicles, highlighting vulnerable components like sensors (LiDAR, GPS), intra-vehicular links (CAN, OBD-II), inter-vehicular links (5G VANET), and ECUs.[3]

Spoofing attacks are a significant threat to autonomous vehicles, as they manipulate the data received by sensors or communication systems to disrupt vehicle operations. These attacks target the vehicle's perception, decision-making, and navigation systems by sending falsified signals. Common targets of spoofing include GPS signals, time synchronization protocols, and sensor data like LiDAR measurements.

Autonomous vehicles rely heavily on these systems to operate safely and efficiently. For instance, GPS provides positional data for navigation, time synchronization ensures proper coordination between systems, and LiDAR detects obstacles in real time. Spoofing attacks exploit the trust that these systems place in external data sources, leading to errors that can compromise safety.

This review focuses on three types of spoofing attacks:

- GPS Spoofing: Misleading the vehicle's navigation system by broadcasting falsified GPS signals.

- Time Spoofing: Disrupting system synchronization by altering time signals.
- LiDAR Spoofing: Manipulating distance measurements to create false perceptions of obstacles.

The following sections discuss case studies that demonstrate the real-world implications of these spoofing attacks and their impact on autonomous vehicle systems. These case studies were selected to provide insights into how such attacks are executed, their effects, and existing gaps in countermeasures.

2.2 Case Study: GPS Spoofing Attack on Tesla Model 3 by Regulus Cyber (2019)

Introduction to Regulus Cyber

Regulus Cyber is a cybersecurity firm specializing in GNSS (Global Navigation Satellite System) security. In 2019, they conducted a groundbreaking GPS spoofing test on the Tesla Model 3 to highlight the vulnerabilities of autonomous vehicles relying heavily on GPS navigation systems.

Tesla Model 3's Dependency on GPS

The Tesla Model 3 integrates GPS with other sensor data for navigation and driver-assist features like *Navigate on Autopilot*. While the system has some security measures, the test revealed vulnerabilities to spoofing attacks.

Spoofing Methodology

Regulus Cyber utilized low-cost hardware and software, including software-defined radio (SDR) devices, to transmit false GPS signals. These spoofed signals overpowered legitimate signals, tricking the navigation system into incorrect operations.

Execution of the Attack

During a controlled test drive, fabricated GPS signals misled the Tesla Model 3, causing the following effects:

- **Navigation errors:** Incorrect turns, route deviations, and abrupt speed changes.
- **Unexpected system impacts:** The attack caused the vehicle's air suspension to behave unpredictably, altering the car's height while driving.

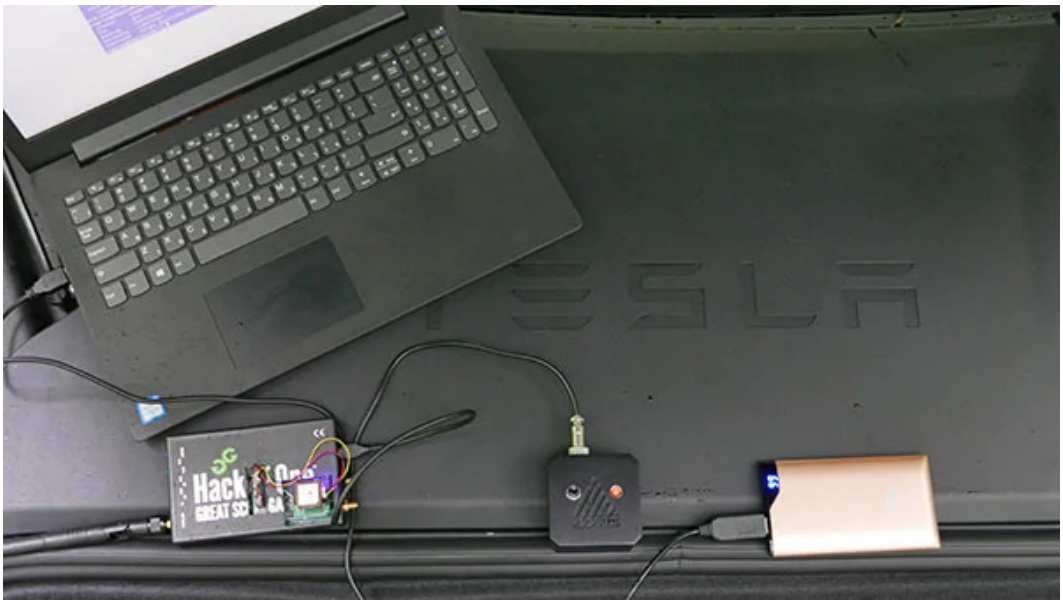

Figure 2.2: GPS Spoofing Equipment used in the Tesla Model 3 test by Regulus Cyber[10]

Impact on Driver Assistance Features

The spoofing attack compromised driver assistance features such as *Navigate on Autopilot*. Unsafe manoeuvres included:

- Abrupt lane changes.
- Incorrect speed adjustments.

These errors posed significant risks to both occupants and other road users.

System's Response

The Tesla Model 3 failed to detect or correct the spoofed GPS data, leading to unsafe behaviours. This highlighted the need for more robust anomaly detection and mitigation strategies in autonomous systems.

2.3 Case Study: Tesla Autopilot Time Delay Vulnerability by Keen Security Lab (2019)

Introduction to Keen Security Lab

Keen Security Lab, a cybersecurity research division of Tencent, conducted a comprehensive study in 2019 to identify vulnerabilities in Tesla's Autopilot system. Their research aimed to explore how time delays in autonomous vehicle systems

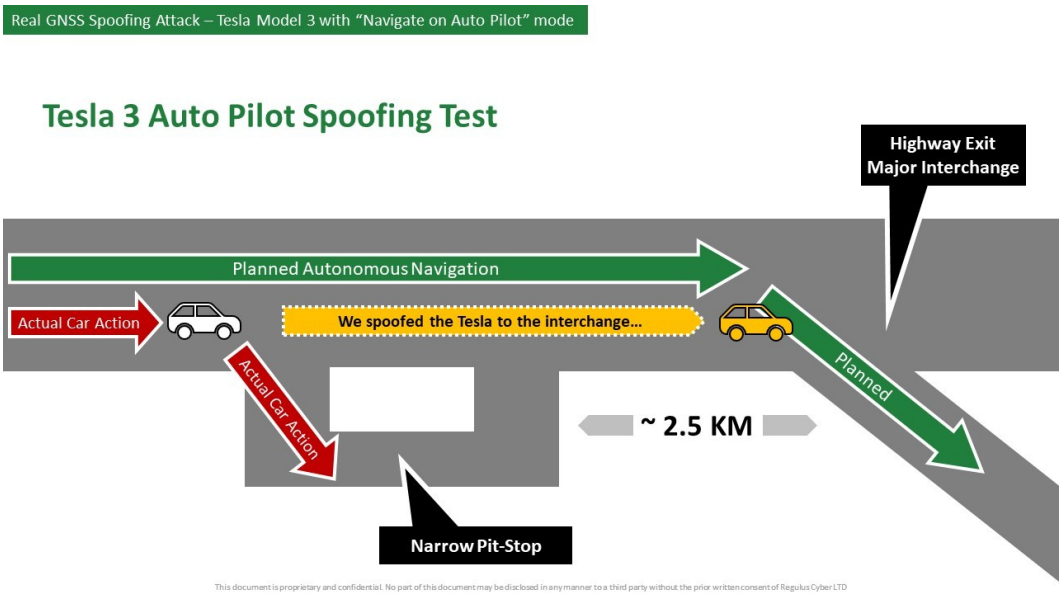

Figure 2.3: Path deviation of Tesla Model 3 due to GPS spoofing.[6]

could impact safety-critical operations, particularly those reliant on synchronized sensor data. This study underscored the importance of addressing timing vulnerabilities in ensuring the safety and reliability of autonomous vehicles.

Tesla Autopilot System and Timing Dependency

Tesla's Autopilot system processes data from multiple sensors, including cameras, radar, and ultrasonic sensors, to make real-time driving decisions. The accuracy of these decisions heavily depends on the synchronization of sensor data streams. Even minor timing discrepancies can result in incorrect interpretations of the driving environment, leading to unsafe maneuvers.

Spoofing Mechanism

Keen Security Lab demonstrated how time spoofing could exploit these vulnerabilities by introducing delays in sensor data processing. These delays disrupted the synchronization of sensor inputs, causing the system to misinterpret critical information such as the distance and speed of nearby objects. The spoofing attack was simulated in controlled scenarios to analyze its impact on vehicle behavior.

Impact on Vehicle Behavior

The study highlighted several unsafe behaviors caused by time spoofing:

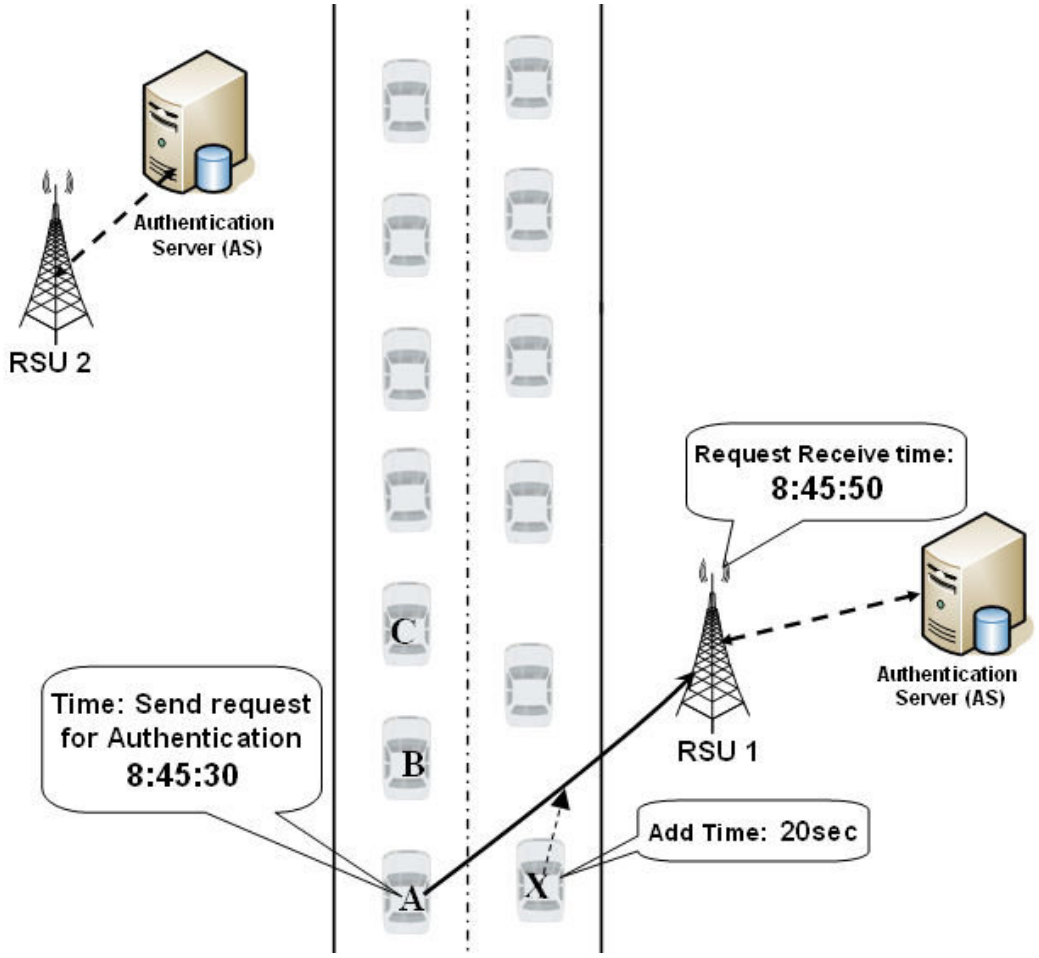

Figure 2.4: Illustration of timing spoofing in vehicle authentication, demonstrating the delay introduced in the signal synchronization.[13]

- **Navigation Errors:** Delays in sensor data processing led to incorrect decisions, such as abrupt lane changes and misjudgments of object distances.
- **Collision Risks:** In one test, a delay in radar data caused the vehicle to fail to stop for an obstacle, resulting in a near-collision scenario.
- **Synchronization Failures:** Misaligned data streams created inconsistencies in the system's environmental model, impairing its ability to make accurate decisions.

Tesla's Response and Countermeasures

Following the discovery of these vulnerabilities, Tesla implemented software updates aimed at reducing processing delays and improving sensor synchronization.

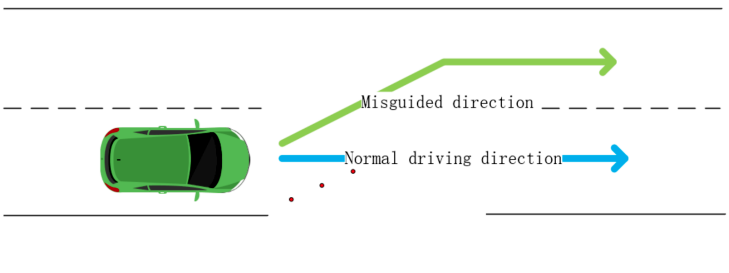

Figure 2.5: Example of a misguided driving direction caused by timing spoofing, showing how delays can mislead the vehicle's decision-making.[7]

However, Keen Lab noted that while these updates mitigated some risks, they did not completely eliminate the vulnerabilities. The study emphasized the need for continuous improvements and robust detection mechanisms.

Lessons and Recommendations

The Keen Lab case study provided the following insights and recommendations:

- **Real-Time Monitoring:** Autonomous systems should incorporate real-time detection mechanisms to identify timing discrepancies as they occur.
- **Sensor Redundancy:** Cross-verifying data from multiple sensors can help mitigate the effects of time spoofing by providing alternative sources of accurate information.
- **Advanced Algorithms:** Predictive algorithms should be developed to compensate for timing discrepancies and enhance the resilience of autonomous systems.

2.4 Case Study: LiDAR Spoofing Attack by Cao et al. (2024)

Introduction to Sensor Spoofing

LiDAR spoofing attacks manipulate the distance measurements of LiDAR systems by injecting false reflections into the sensor data. This type of attack can mislead the vehicle's decision-making processes, causing incorrect trajectory adjustments. Cao et al. (2024) conducted a notable study on LiDAR spoofing attacks to highlight vulnerabilities in autonomous vehicle systems.

Spoofing Mechanism

The attack involves using a laser to generate false reflections that the LiDAR system interprets as real obstacles. The setup includes:

- A delay component to time the spoofed reflections.
- A photodiode and lens to project the fake reflections accurately.

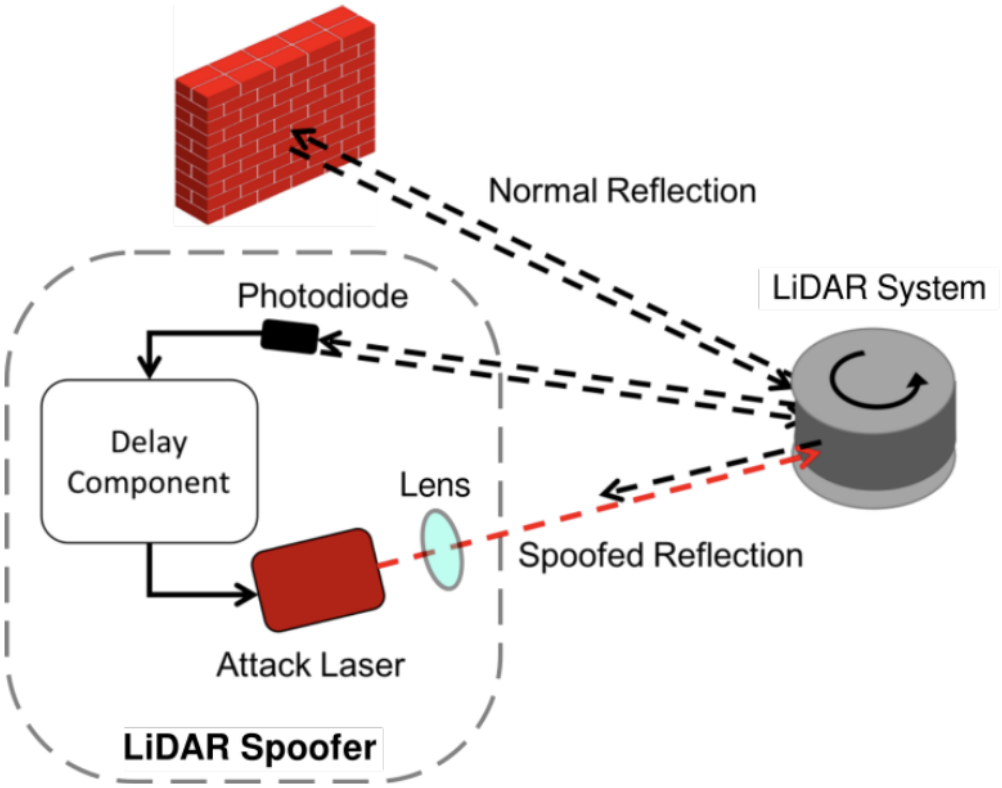

Figure 2.6: LiDAR Spoofing Setup as demonstrated by Cao et al. (2024).[14]

Impact on Vehicle Trajectories

The spoofing attack led to significant errors in trajectory planning. For example:

- Vehicles were tricked into perceiving nonexistent obstacles.
- Trajectories were altered, causing potential collisions or unsafe manoeuvres.

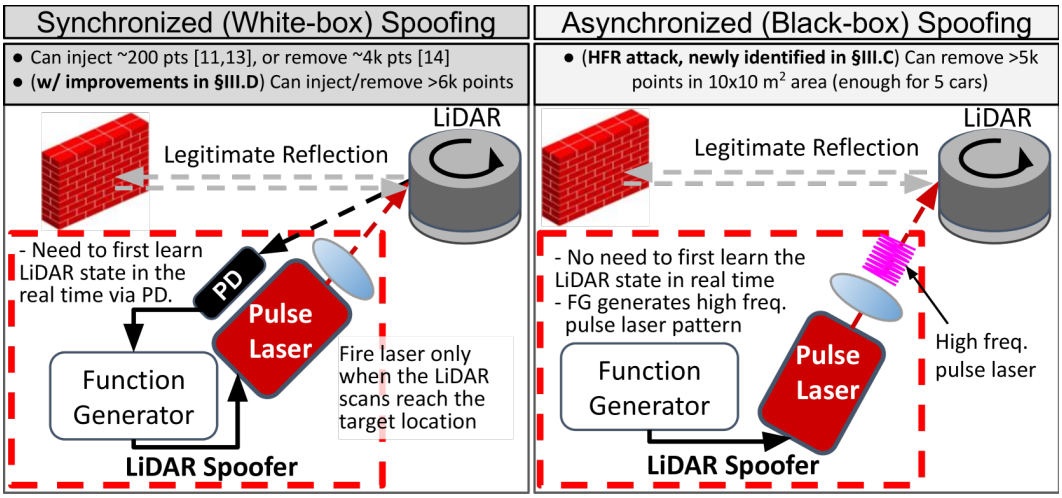

Figure 2.7: Example of a trajectory misguidance caused by LiDAR spoofing.[14]

System Response and Vulnerabilities

The study revealed that current LiDAR systems lack effective countermeasures for spoofing attacks. The absence of validation mechanisms for incoming data makes these systems particularly vulnerable. Researchers emphasized the need for:

- Multi-sensor cross-validation techniques.
- Real-time anomaly detection algorithms.
- Hardware-based improvements in LiDAR systems.

Summary of Case Studies

The case studies reviewed in this chapter underscore the critical challenges autonomous vehicles face from spoofing attacks. GPS spoofing, time spoofing, and sensor spoofing were all shown to exploit specific vulnerabilities in the vehicle's navigation and decision-making systems. Common patterns emerged, such as the dependency on single-source data and the lack of real-time anomaly detection mechanisms. These vulnerabilities highlight the importance of robust cybersecurity measures in ensuring the safety and reliability of autonomous systems.

Chapter 3

METHODOLOGY

3.1 Simulation Setup

This section provides an overview of the general setup and tools used to conduct the simulations. These simulations are designed to model and visualize the effects of various spoofing attacks on autonomous vehicle systems. The setup ensures consistency and reliability across all scenarios.

Tools and Environment

All simulations were developed and executed using **MATLAB R2024a** on a personal computer running **Windows 11**. MATLAB was chosen for its powerful computational capabilities and advanced visualization tools, which made it suitable for simulating and analyzing the effects of spoofing attacks in a controlled environment.

The computer specifications used for the simulations are:

- **Operating System:** Windows 11
- **Processor:** Intel Core i5 (10th Generation)
- **RAM:** 16 GB
- **Software:** MATLAB R2024a

Simulation Parameters

To ensure consistency across all the spoofing attack simulations, a common set of parameters was established:

- **Time Frame:** Each simulation spans a duration of **10 seconds**, providing a sufficient period to observe the effects of spoofing from initiation to termination.
- **Time Resolution:** A high resolution of **1,000 points** (0.01-second time steps) was used for precise modeling and smooth visualization.

- **Vehicle Dynamics:** The vehicle was assumed to move at a **constant speed of 5 m/s** along a straight path. This simplified dynamic isolates the spoofing effects for a focused analysis.

Visualization Approach

Effective visualization was a critical part of the simulations, as it helped in understanding and interpreting the results. The following visualization techniques were employed:

- **True vs. Spoofed Data:** Plots comparing the genuine sensor readings with spoofed data were used to illustrate the discrepancies introduced by spoofing attacks.
- **Dynamic Visualization:** Real-time updating plots were implemented to visually track the evolution of spoofing effects over the simulation period.
- **Derived Metrics:** Additional plots such as velocity, acceleration, and rate of change were included to highlight how the spoofing attack affects the vehicle's perceived dynamics and decision-making processes.

Environment Considerations

All simulations were conducted in a controlled environment, ensuring that:

- The simulations were free from external interruptions to maintain consistent results.
- Code execution and data visualization were optimized for the given hardware to avoid computational delays.

By maintaining a consistent setup for all scenarios, the simulations provided meaningful and comparable insights into the effects of GPS, time, and LiDAR sensor spoofing on autonomous vehicle systems.

3.2 GPS Spoofing Simulation Methodology

Objective

The main goal of this simulation is to demonstrate how GPS spoofing can affect a vehicle's navigation system. By simulating the deviations caused by spoofing attacks, this work aims to replicate real-world scenarios where spoofed GPS data misguides

vehicles. The focus is on using simple mathematical models and visualizing the changes dynamically.

Approach

The simulation follows a structured process to make the results clear and understandable:

1. Define the original path of the vehicle as a straight-line GPS trajectory.
2. Introduce spoofing effects through mathematical variations like amplitude, frequency, random noise, and drift.
3. Visualize the impact of spoofing on the original and spoofed paths, along with velocity and acceleration, in a dynamic and intuitive manner.

Mathematical Model

The simulation is based on a straightforward mathematical framework, which includes:

- **Original Path:** The vehicle's original GPS path is modeled as a linear trajectory:

$$y_{\text{original}}(t) = t \tag{3.1}$$

  This represents the vehicle moving at a constant rate with time.

- **Spoofing Dynamics:** Spoofing begins at $t = 2$ seconds and introduces deviations from the original path. The spoofed position is calculated using the following equation:

$$y_{\text{spoofed}}(t) = y_{\text{original}}(t) + A(t) \cdot \sin(f(t) \cdot t) + W(t) + N(t) \tag{3.2}$$

  where:

  - $A(t) = 1.0 + 0.3\sin(0.1t)$ models the variation in amplitude over time.
  - $f(t) = 1.0 + 0.1\cos(0.05t)$ represents the frequency variation of the spoofing signal.
  - $W(t)$ represents a random walk that simulates sensor noise:

$$W(t) = \sum_{i=1}^{n} 0.05 \cdot \text{randn}() \tag{3.3}$$

  - $N(t)$ introduces Gaussian noise:

$$N(t) = 0.2 \cdot \text{randn}() \tag{3.4}$$

- **Velocity and Acceleration:** The changes in velocity and acceleration caused by spoofing are derived numerically:

$$v_{\text{spoofed}}(t) = \frac{\Delta y_{\text{spoofed}}(t)}{\Delta t} \tag{3.5}$$

$$a_{\text{spoofed}}(t) = \frac{\Delta v_{\text{spoofed}}(t)}{\Delta t} \tag{3.6}$$

Implementation Steps

The implementation of this simulation was carried out in a few key steps:

1. Create a time vector t spanning 10 seconds with 1000 points to ensure smooth updates.
2. Compute the original and spoofed positions using the mathematical equations defined above.
3. Dynamically plot the original and spoofed paths, highlighting key events like the start and peak of the spoofing attack.
4. Add annotations to make the visualization more intuitive.
5. Compute and visualize the velocity and acceleration to show the detailed effects of spoofing.

Visualization

The simulation outputs several visualizations to help understand the effects of GPS spoofing:

- **Position Plot:** This plot shows the original and spoofed GPS paths, with annotations to indicate when spoofing starts and ends.
- **Velocity and Acceleration Plot:** These plots highlight how spoofing affects the vehicle's movement over time by showing variations in velocity and acceleration.

Reflection and Challenges

Working on this simulation provided a practical understanding of how GPS spoofing attacks impact autonomous navigation systems. A key challenge was managing the balance between simplicity and realism in the mathematical models. For example, introducing random walk and Gaussian noise required careful tuning to ensure the simulation was both realistic and computationally efficient. Additionally, designing clear annotations to highlight the spoofing events took multiple iterations to make the visualizations more intuitive.

3.3 Time Spoofing Simulation Methodology

Objective

The goal of this simulation is to understand and visualize the effects of time spoofing on an autonomous vehicle's navigation system. Time spoofing attacks create irregularities in the time signal, which can affect how the system processes data. This simulation provides a way to see these effects in a controlled environment using mathematical modeling and dynamic plots.

Approach

The simulation follows these steps:

1. Define a straightforward time signal that progresses linearly.
2. Introduce spoofing by adding a combination of linear deviations, sinusoidal fluctuations, and random noise to simulate real-world attack scenarios.
3. Display the original and spoofed time signals, along with additional metrics like time offset and the rate of change, in an animated format.

Mathematical Model

The simulation uses the following mathematical components:

Original Time Signal The original time signal is modeled as:

$$T_{\text{original}}(t) = t \tag{3.7}$$

This represents a simple, linear progression of time.

Spoofing Dynamics Spoofing starts at $t = 2$ seconds. The spoofed time signal is modeled as:

$$T_{\text{spoofed}}(t) = T_{\text{original}}(t) + \Delta T_{\text{linear}}(t) + \Delta T_{\text{sinusoidal}}(t) + \Delta T_{\text{noise}}(t) \tag{3.8}$$

where:

- $\Delta T_{\text{linear}}(t) = 0.5 \cdot (t - t_{\text{start}})$ is a simple, growing deviation over time.
- $\Delta T_{\text{sinusoidal}}(t) = 0.2 \cdot \sin(2\pi \cdot 0.5 \cdot t)$ adds a periodic fluctuation.
- $\Delta T_{\text{noise}}(t) = 0.1 \cdot \text{randn}()$ introduces random Gaussian noise to mimic unpredictable variations.

Time Offset The time offset, which shows how much the spoofed signal deviates from the original, is calculated as:

$$\text{Time Offset}(t) = T_{\text{spoofed}}(t) - T_{\text{original}}(t) \tag{3.9}$$

Rate of Change The rate of change in the spoofed signal is calculated numerically to observe abrupt fluctuations:

$$\text{Rate of Change}(t) = \frac{\Delta T_{\text{spoofed}}(t)}{\Delta t} \tag{3.10}$$

Implementation Steps

Here's how the simulation was implemented:

1. Create a time vector t that spans 10 seconds with 1000 points for smooth animations.
2. Calculate the original and spoofed time signals using the equations above.
3. Compute the time offset and rate of change to understand the impact of spoofing.
4. Use animated plots to display:
    - The original and spoofed time signals with annotations marking when spoofing starts.
    - The time offset with a threshold line to show when spoofing becomes noticeable.
    - The rate of change with labels marking the highest fluctuation.

Visualization

The simulation produces three key plots:

- **Original vs. Spoofed Time Signal:** Shows the original time progression compared to the spoofed signal. A shaded region highlights the spoofing duration, and annotations mark when spoofing begins.
- **Time Offset:** Displays how much the spoofed signal deviates from the original over time, with a threshold line to indicate when the offset becomes significant.
- **Rate of Change:** Visualizes the sudden changes in the spoofed signal, helping to identify irregularities caused by the spoofing attack.

Reflection and Challenges

Building this simulation made it easier to understand how time spoofing can disrupt a vehicle's navigation system. One challenge was balancing the complexity of the spoofing effects (like noise and sinusoidal disturbances) with the need to keep the simulation smooth and understandable. Another difficulty was ensuring the annotations were clear and placed at the right moments, such as when spoofing starts or when a threshold is crossed. Overall, the simulation captures the key impacts of time spoofing in a way that's easy to present and explain.

3.4 LiDAR Sensor Spoofing Simulation Methodology

Objective

The purpose of this simulation is to demonstrate the impact of spoofing attacks on a LiDAR sensor, which is used by autonomous vehicles for obstacle detection. By simulating a spoofing attack, we can visualize how incorrect distance measurements may lead to misinterpretation of the environment. The goal is to analyze and understand how such attacks manipulate distance data and to represent these effects through an intuitive and detailed visualization.

Approach

The simulation follows these steps:

1. Define the true obstacle distance as a constant measurement.
2. Introduce spoofing effects by adding linear drift, sinusoidal disturbances, and Gaussian noise within a specified time window.

3. Calculate and visualize the deviation between the true and spoofed distances, as well as the rate of change in measurements.

4. Highlight important moments such as the start and end of spoofing, crossing the detection threshold, and the maximum rate of change.

Mathematical Model

The simulation is based on the following mathematical concepts:

True Distance: The actual distance to the obstacle is assumed to be constant and is given by:

$$d_{\text{true}}(t) = 20 \text{ meters} \tag{3.11}$$

Spoofing Dynamics: Spoofing is applied between t_{start} and t_{end} and is modeled as:

$$d_{\text{spoofed}}(t) = d_{\text{true}}(t) + \Delta d_{\text{linear}}(t) + \Delta d_{\text{sinusoidal}}(t) + \Delta d_{\text{noise}}(t) \tag{3.12}$$

where:

- $\Delta d_{\text{linear}}(t)$: A gradual drift simulating an increase or decrease in distance over time:

$$\Delta d_{\text{linear}}(t) = 0.5 \cdot (t - t_{\text{start}}) \tag{3.13}$$

- $\Delta d_{\text{sinusoidal}}(t)$: A periodic fluctuation added to simulate a sinusoidal disturbance:

$$\Delta d_{\text{sinusoidal}}(t) = 2 \cdot \sin(2\pi \cdot 0.2 \cdot t) \tag{3.14}$$

- $\Delta d_{\text{noise}}(t)$: Random noise to represent sensor inaccuracies:

$$\Delta d_{\text{noise}}(t) = 0.5 \cdot \mathcal{N}(0, 1) \tag{3.15}$$

Distance Deviation: The deviation from the true distance is calculated as:

$$\Delta d(t) = d_{\text{spoofed}}(t) - d_{\text{true}}(t) \tag{3.16}$$

Rate of Change: The rate of change of the spoofed distance is derived numerically:

$$\frac{d}{dt} d_{\text{spoofed}}(t) = \frac{\Delta d_{\text{spoofed}}(t)}{\Delta t} \tag{3.17}$$

Implementation Steps

The implementation is carried out as follows:

1. Generate a time vector t spanning 10 seconds with 1000 points for smooth updates.
2. Define the true obstacle distance as a constant value of 20 meters.
3. Simulate the spoofing attack by applying the linear drift, sinusoidal disturbance, and Gaussian noise within the specified spoofing duration (t_{start} = 3 seconds, t_{end} = 6 seconds).
4. Calculate the spoofed distance, distance deviation, and rate of change using the equations provided.
5. Create a three-part visualization:
    - True vs. Spoofed Distance: This compares the actual and manipulated distance values.
    - Distance Deviation: This highlights how much the spoofed distance differs from the true value.
    - Rate of Change: This shows how quickly the distance readings are changing during and after the spoofing attack.
6. Add annotations to indicate the start and end of spoofing, as well as key events like threshold crossings and maximum rate of change.
7. Implement a real-time simulation to update the visualizations dynamically.

Visualization and Results

The simulation provides the following insights:

- **True vs. Spoofed Distance:** The top plot displays the true and spoofed distances over time. A red-shaded region indicates the spoofing duration, with annotations marking the start and end of the attack.
- **Deviation from True Distance:** The middle plot shows how much the spoofed distance deviates from the true value. A red dashed line represents the detection threshold, and annotations highlight when the deviation crosses this threshold.

- **Rate of Change:** The bottom plot visualizes the rate at which the spoofed distance changes over time. An annotation marks the point of maximum rate of change during the attack.

Reflection and Challenges

This simulation highlights the potential risks of LiDAR spoofing attacks in autonomous vehicles. By observing the deviation and rate of change, we can better understand how spoofing disrupts the sensor's functionality. Challenges included:

- Creating realistic spoofing dynamics while maintaining simplicity for computational efficiency.
- Ensuring the visualizations were clear and effectively conveyed the progression and impact of the spoofing attack.

Chapter 4

RESULTS AND ANALYSIS

4.1 Introduction

This chapter presents the results of the three spoofing simulations: GPS Spoofing, Time Spoofing, and LiDAR Sensor Spoofing. Each simulation is analyzed in detail, and the comparative metrics are discussed. The simulations showcase how spoofing attacks affect the respective systems, emphasizing the deviations, error percentages, and rates of change.

4.2 GPS Spoofing Results

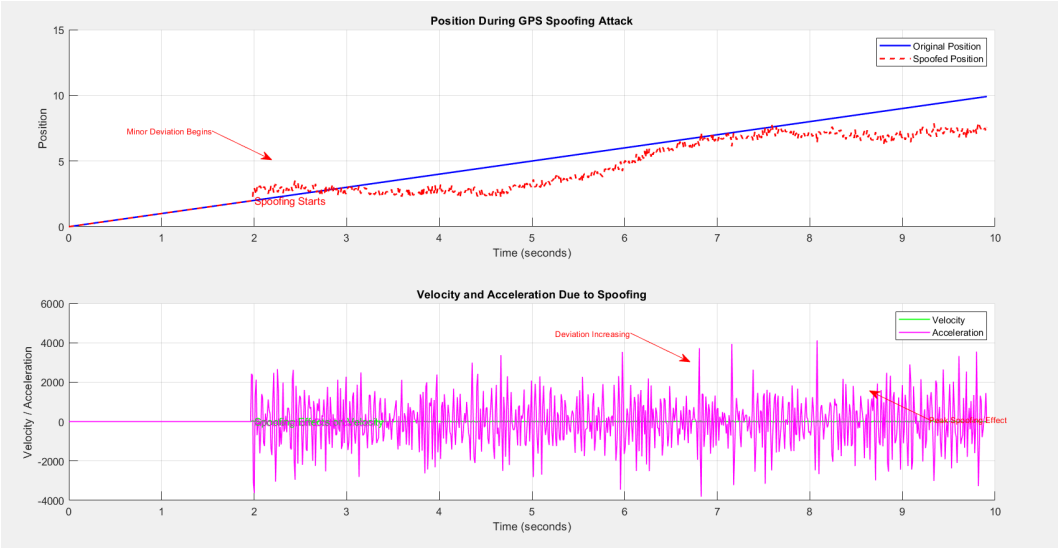

Figure 4.1: Position and Velocity/Acceleration Plots for GPS Spoofing.

Simulation Overview

The GPS spoofing simulation introduces deviations to the original position by adding sinusoidal noise, random walk, and Gaussian noise. The results are visualized in two plots: position and velocity/acceleration.

Key Observations

- **Position Plot:** The original position (blue) is a straight line, while the spoofed position (red) deviates noticeably after $t = 2$ seconds. The maximum deviation

occurs at $t \approx 6$ seconds, with a value of 2.5 m.

- **Velocity/Acceleration Plot:** Velocity fluctuations increase after $t = 2$ seconds, with significant acceleration spikes around $t = 6$ seconds.

Quantitative Metrics

- **Maximum Deviation:**

$$\text{Max Deviation} = \max(|y_{\text{spoofed}} - y_{\text{original}}|) = 2.5\,\text{m}$$

- **Error Percentage:**

$$\text{Error Percentage} = \frac{\text{Max Deviation}}{\text{Path Length}} \times 100 = \frac{2.5}{10} \times 100 = 25\%$$

4.3 Time Spoofing Results

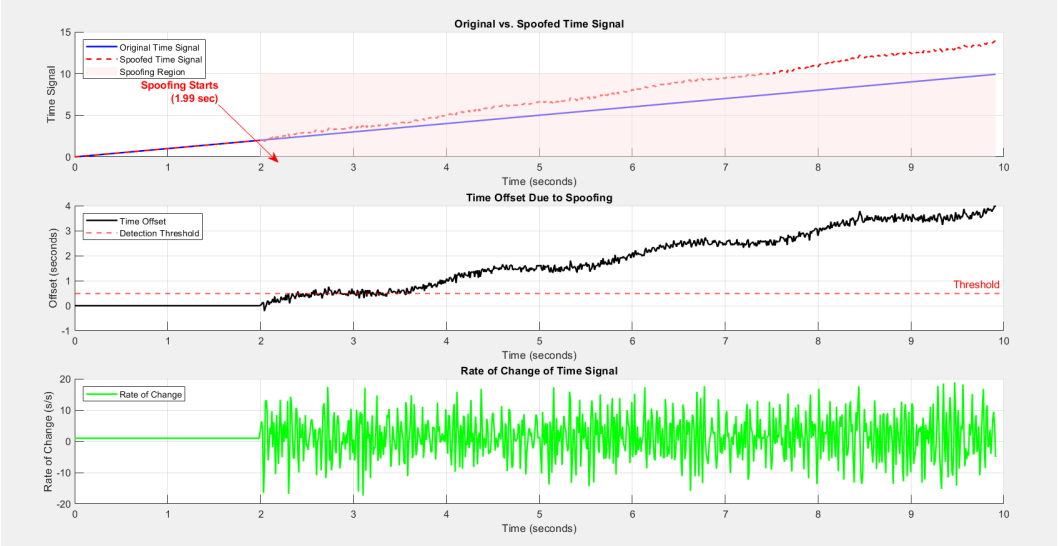

Figure 4.2: Time Signal and Rate of Change for Time Spoofing

Simulation Overview

The time spoofing simulation introduces linear drift, sinusoidal disturbance, and Gaussian noise to the original time signal. The results are visualized in three plots: original vs. spoofed time signal, time offset, and rate of change.

Key Observations

- **Time Signal Plot:** The original signal (green) is a straight line, while the spoofed signal (red) deviates upward after $t = 2$ seconds, with the maximum offset at $t = 6$ seconds.

- **Rate of Change Plot:** Spikes in the rate of change are evident, corresponding to periods of significant spoofing.

Quantitative Metrics

- **Maximum Deviation:**

$$\text{Max Deviation} = \max(|t_{\text{spoofed}} - t_{\text{original}}|) = 1.5 \text{ seconds}$$

- **Error Percentage:**

$$\text{Error Percentage} = \frac{\text{Max Deviation}}{\text{Total Time}} \times 100 = \frac{1.5}{10} \times 100 = 15\%$$

4.4 LiDAR Sensor Spoofing Results

Simulation Overview

The LiDAR sensor spoofing simulation introduces linear drift, sinusoidal disturbance, and Gaussian noise to the true distance measurement. The results are visualized in three plots: distance measurement, deviation, and rate of change.

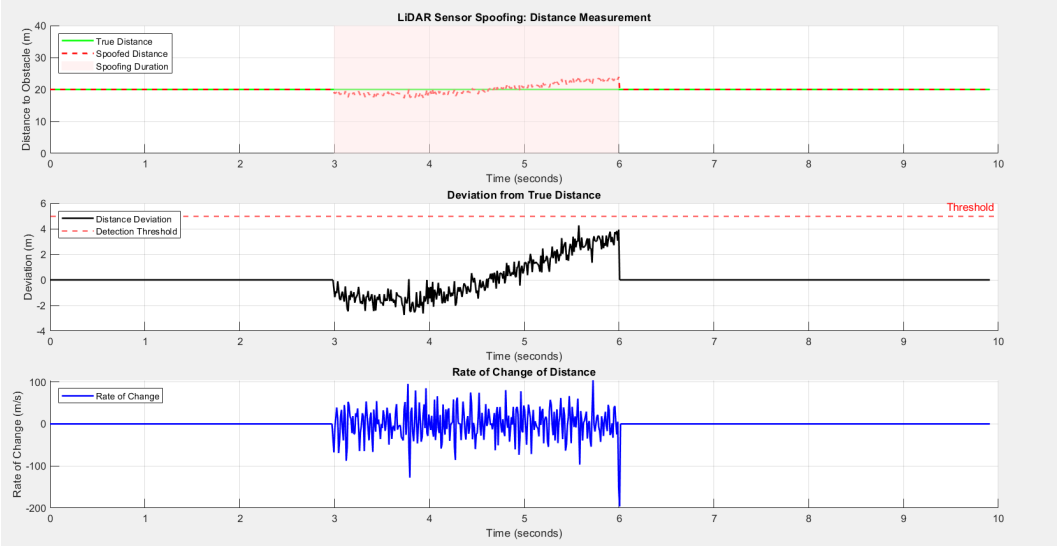

Figure 4.3: Distance Measurement and Rate of Change for LiDAR Sensor Spoofing

Key Observations

- **Distance Measurement Plot:** The true distance (green) remains constant, while the spoofed distance (red) fluctuates significantly between $t = 3$ and $t = 6$ seconds.

- **Deviation Plot:** The deviation increases steadily, exceeding the detection threshold around $t = 5$ seconds.

Quantitative Metrics

- **Maximum Deviation:**

$$\text{Max Deviation} = \max(|d_{\text{spoofed}} - d_{\text{true}}|) = 4\,\text{m}$$

- **Error Percentage:**

$$\text{Error Percentage} = \frac{\text{Max Deviation}}{\text{True Distance}} \times 100 = \frac{4}{20} \times 100 = 20\%$$

4.5 Comparative Analysis

Table 4.1: Comparative Metrics for Spoofing Simulations

Spoofing Type	Max Deviation	Error Percentage	Noise Duration
GPS Spoofing	2.5 m	25%	$t = 2$ to 9 seconds
Time Spoofing	1.5 s	15%	$t = 2$ to 6 seconds
LiDAR Spoofing	4 m	20%	$t = 3$ to 6 seconds

Discussion

The comparative metrics reveal that:

- GPS spoofing exhibits the highest error percentage (25%) due to compounded effects of random walk and sinusoidal noise.
- Time spoofing has a lower error percentage (15%) but introduces significant timing offsets.
- LiDAR spoofing causes the largest maximum deviation (4 m) within a short duration.

The following charts illustrate the effects of these spoofing attacks on vehicle systems:

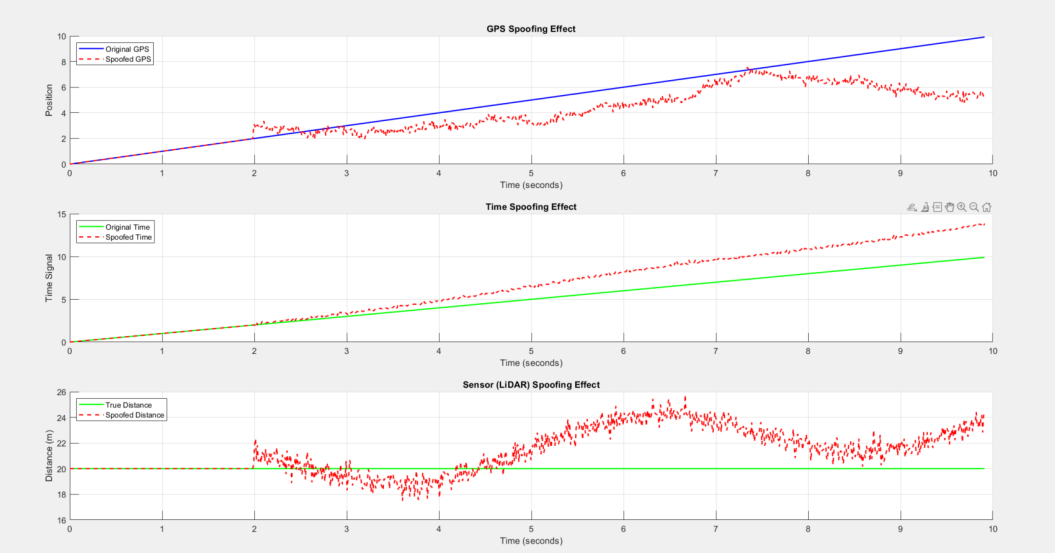

Figure 4.4: Effects of GPS, Time, and Sensor Spoofing on Vehicle Systems

Chapter 5

DISCUSSION

5.1 Challenges and Learning Outcomes

This project was both challenging and rewarding, as it required a detailed understanding of how spoofing attacks affect the core systems of autonomous vehicles. One of the main challenges was simulating the spoofing scenarios realistically for GPS, time synchronization, and LiDAR systems. Each type of spoofing required different approaches, and ensuring the accuracy of the simulations was a time-consuming process.

Another challenge was understanding and applying the mathematical models to calculate metrics like deviation and error percentages. While generating visualizations for these attacks was straightforward after initial setup, interpreting the results and connecting them to real-world implications required significant effort.

Through this process, I learned how to simulate complex systems and gained a deeper understanding of how different spoofing attacks operate. Working with MATLAB also helped me improve my technical skills, especially in generating simulations and analyzing their outputs. Additionally, presenting the results in a meaningful way taught me how to structure and communicate technical findings effectively.

5.2 Impact on Autonomous Vehicle Security

The findings from this case study highlight the critical risks spoofing attacks pose to autonomous vehicle systems. Each type of spoofing attack affects the system in a unique way:

GPS Spoofing

The simulations revealed how a spoofed GPS signal causes the vehicle to deviate from its original path. This can lead to significant positional errors, potentially misguiding the vehicle into unsafe or unintended routes. For example, the calculated deviation and error percentage in the simulation showcase how even small disruptions can have a large cumulative effect on navigation [5].

Time Spoofing

Time spoofing affects the synchronization of systems, which is critical for operations like communication between sensors or external systems. The observed time offsets in the simulation demonstrate how such disruptions could destabilize system coordination, causing errors in decision-making [12].

LiDAR Spoofing

LiDAR spoofing was shown to alter distance measurements, impacting the vehicle's ability to detect obstacles accurately. This could lead to unsafe actions, such as misjudging distances or failing to identify hazards [9].

Overall, the case study demonstrates that even small spoofing effects can have significant consequences for vehicle safety. Detecting and mitigating these attacks is crucial for ensuring that autonomous systems operate securely and reliably.

5.3 Potential for Future Research

This study was limited to controlled simulations of spoofing attacks. While these provided valuable insights, there are several areas where future research could build upon this work:

Real-World Testing

Conducting simulations with real-world data would enhance the practical relevance of this study. Testing spoofing in live scenarios, such as urban and rural environments, would provide a more comprehensive understanding of how these attacks behave [15].

Exploring Countermeasures

While this study focused on detecting spoofing attacks, future work could investigate methods to mitigate these attacks. For instance, implementing algorithms to counter spoofing effects in real-time could make autonomous systems more resilient [2].

Broader Sensor Analysis

This study analyzed GPS, time, and LiDAR spoofing attacks separately. Integrating the findings and exploring how combined spoofing (e.g., GPS and LiDAR simultaneously) impacts the system could offer deeper insights into attack strategies and system vulnerabilities [8].

Advanced Simulation Metrics

Adding more detailed metrics, such as spoofing severity scores or quantifying system response times to spoofing, could make future analyses more precise and actionable [11].

By extending this research into these areas, it would be possible to contribute further to the field of autonomous vehicle security, helping to develop systems that are robust against spoofing attacks.

Chapter 6

CONCLUSION

6.1 Summary of Findings

This project provided valuable insights into how spoofing attacks impact autonomous vehicle systems. Through simulations of GPS, time synchronization, and LiDAR spoofing, key findings were uncovered:

- **GPS Spoofing:** The simulation demonstrated how spoofed GPS signals cause path deviations, leading to potential positional errors. These deviations highlight the risks of misguiding vehicles into unsafe or unintended routes.
- **Time Spoofing:** Time synchronization disruptions were shown to affect communication and coordination between vehicle systems. Observed time offsets illustrated how this could lead to decision-making errors in real-world applications.
- **LiDAR Spoofing:** Altered distance measurements resulted in inaccurate obstacle detection, creating risks of misjudging distances or failing to identify hazards altogether.

While the findings are significant, the simulations were conducted in controlled environments. This limits the applicability of the results to real-world scenarios. Adding more comprehensive, real-world testing could have made the findings more practical.

6.2 Final Reflections

The project was challenging but rewarding, requiring a deep understanding of spoofing attacks and their effects on autonomous systems. Some of the challenges included simulating realistic spoofing attacks for each sensor and interpreting the results accurately.

Through this case study, I learned:

- How to simulate and analyze spoofing attacks for various autonomous systems.

- The importance of metrics like deviation and error percentages in evaluating system vulnerabilities.
- How to present technical findings in a meaningful and structured way.

Working with MATLAB simulations improved my technical skills significantly. However, I found that interpreting complex data and linking it to real-world implications was not straightforward. In future projects, I would focus on better integrating practical testing alongside theoretical simulations to achieve a more balanced study.

Appendix A

APPENDICES

A.1 MATLAB Code for Simulations

Below is a list of MATLAB code files used for this project. These files are stored in the ‘Appendix‘ folder of the project directory and can also be accessed via the provided GitHub links.

1. **GPS Spoofing Simulation**:
    - File location: `Appendix/GPS Spoofing Final.m`
    - GitHub link: GPS Spoofing Final
2. **Sensor Spoofing Simulation**:
    - File location: `Appendix/Sensor Spoofing Final.m`
    - GitHub link: Sensor Spoofing Final
3. **Time Spoofing Simulation**:
    - File location: `Appendix/Time Spoofing Final.m`
    - GitHub link: Time Spoofing Final

BIBLIOGRAPHY

[1] Murad Mehrab Abrar, Raian Islam, Shalaka Satam, Sicong Shao, Salim Hariri, and Pratik Satam. Gps-ids: An anomaly-based gps spoofing attack detection framework for autonomous vehicles. *arXiv preprint arXiv:2405.08359*, 2024.

[2] Khattab M. Ali Alheeti, Abdulkareem Alzahrani, and Duaa Al Dosary. Lidar spoofing attack detection in autonomous vehicles. In *2022 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–2, 2022.

[3] Man Chun Chow, Maode Ma, and Zhijin Pan. Attack models and countermeasures for autonomous vehicles. In *Intelligent Technologies for Internet of Vehicles*, pages 375–401. Springer, 2021.

[4] Sagar Dasgupta, Kazi Hassan Shakib, and Mizanur Rahman. Experimental validation of sensor fusion-based gnss spoofing attack detection framework for autonomous vehicles, 2024.

[5] Manuel del Castillo. Gps spoofing: a cyber security threat to in car navigation & avs. *FocalPoint Positioning*, 2021.

[6] Editor. 'We spoofed a Tesla and really scared our co-worker!' - Regulus Cyber - RNTF, 6 2019.

[7] Dan Goodin. Researchers trick tesla autopilot into steering into oncoming traffic. *Ars Technica*, 2019. Accessed: 2024-11-18.

[8] Amira Guesmi and Muhammad Shafique. Navigating threats: A survey of physical adversarial attacks on lidar perception systems in autonomous vehicles. *arXiv preprint arXiv:2409.20426*, 2024.

[9] Xueyang Hu, Tian Liu, Tao Shu, and Diep Nguyen. Spoofing detection for lidar in autonomous vehicles: A physical-layer approach. *IEEE Internet of Things Journal*, 11(11):20673–20689, 2024.

[10] Roi Mit. Two years since the tesla gps hack. *GPS World*, 2021.

[11] Takami Sato, Yuki Hayakawa, Ryo Suzuki, Yohsuke Shiiki, Kentaro Yoshioka, and Qi Alfred Chen. Lidar spoofing meets the new-gen: Capability

improvements, broken assumptions, and new attack strategies. *arXiv preprint arXiv:2303.10555*, 2023.

[12] Maliha Shabbir, Mohsin Kamal, Zahid Ullah, and Maqsood Muhammad Khan. Securing autonomous vehicles against gps spoofing attacks: A deep learning approach. *IEEE Access*, 11:105513–105526, 2023.

[13] Irshad Sumra, Jamalul-Lail Ab Manan, and Halabi Hasbullah. Timing attack in vehicular network. 07 2011.

[14] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z. Morley Mao. Towards robust LiDAR-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 877–894. USENIX Association, August 2020.

[15] Zisis-Rafail Tzoannos, Dimitrios Kosmanos, Apostolos Xenakis, and Costas Chaikalis. The impact of spoofing attacks in connected autonomous vehicles under traffic congestion conditions. *Telecom*, 5(3):747–759, 2024.