Threat and Application of Frequency-Agile Radio Systems

Kexiong (Curtis) Zeng

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Engineering

Yaling Yang, Chair
Y. Thomas Hou
Wenjing Lou
Majid Manteghi
Ranveer Chandra
Gang Wang

August 31st, 2018
Blacksburg, Virginia

Keywords: GPS Spoofing, Road Navigation, Cognitive Radio, White Space Radio,
Maritime Mesh Network, Energy Harvesting

Threat and Application of Frequency-Agile Radio Systems

Kexiong (Curtis) Zeng

(ABSTRACT)

As traditional wireless systems that only operate on fixed frequency bands are reaching their capacity limits, advanced frequency-agile radio systems are developed for more efficient spectrum utilization. For example, white space radios dynamically leverage locally unused TV channels to provide high-speed long-distance connectivity. They have already been deployed to connect the unconnected in rural areas and developing countries. However, such application scenarios are still limited due to low commercial demand. Hence, exploring better applications for white space radios needs more effort. With the benefits come the threats. As frequency-agile radio systems (*e.g.*, software-defined radios) are flexible and become extremely low-cost and small-sized, it is very convenient for attackers to build attacking tools and launch wireless attacks using these radios. For example, civilian GPS signals can be easily spoofed by low-cost portable spoofers built with frequency-agile radio systems. In this dissertation, we study both the threat and application of frequency-agile radio systems. Specifically, our work focuses on the spoofing threat of frequency-agile radio towards GPS-based systems and the application of TV white space radio for ocean communications.

Firstly, we explore the feasibility of using frequency-agile radio to stealthily manipulate GPS-based road navigation systems without alerting human drivers. A novel attacking algorithm is proposed, where the frequency-agile radio transmits fake GPS signals to lead the victim to drive on a wrong path that looks very similar with the navigation route on the screen. The attack's feasibility is demonstrated with real-world taxi traces in Manhattan and Boston. We implement a low-cost portable GPS spoofer using an off-the-shelf frequency-agile radio platform to perform physical measurements and real-world driving tests, which shows the

low level of difficulty of launching the attack in real road environment. In order to study human-in-the-loop factor, a deceptive user study is conducted and the results show that 95% of the users do not recognize the stealthy attack. Possible countermeasures are summarized and sensor fusion defense is explored with preliminary tests.

Secondly, we study similar GPS spoofing attack in database-driven cognitive radio networks. In such a network, a secondary user queries the database for available spectrum based on its GPS location. By manipulating GPS locations of surrounding secondary users with a frequency-agile radio, an attacker can potentially cause serious primary user interference and denial-of-service to secondary users. The serious impact of such attacks is examined in simulations based on the WhiteSpaceFinder spectrum database. Inspired by the characteristics of the centralized system and the receiving capability of cognitive radios, a combination of three defense mechanisms are proposed to mitigate the location spoofing threat.

Thirdly, we explore the feasibility of building TV white space radio based on frequency-agile radio platform to provide connectivity on the ocean. We design and implement a low-cost low-power white space router ($523, 12 watts) customized for maritime applications. Its communication capability is confirmed by field link measurements and ocean-surface wave propagation simulations. We propose to combine this radio with an energy harvesting buoy so that the radio can operate independently on the ocean and form a wireless mesh network with other similar radios.

Threat and Application of Frequency-Agile Radio Systems

Kexiong (Curtis) Zeng

(GENERAL AUDIENCE ABSTRACT)

As traditional wireless systems, such as mobile phones and WiFi access points, only operate on some fixed frequency bands, it becomes increasingly crowded for those popular bands. Hence, for more efficient frequency resource utilization, frequency-agile radio systems that can dynamically operate on different frequency bands are developed. With these new technologies come new threats and applications, which are the focus of our work. On the one hand, as frequency-agile radio systems become low-cost and portable, attackers can easily launch wireless attacks with them. For example, we explored the feasibility, impact, and countermeasures for GPS spoofing attacks using frequency-agile radio systems in different scenarios. In a GPS spoofing attack, an attacker transmits false GPS signals to manipulate users' GPS receivers. This kind of attack can be very dangerous and even life-threatening if it is launched against critical GPS-based applications. For example, once GPS-based navigation systems in self-driving cars are stealthily manipulated by remote attackers, attackers can divert self-driving cars to pre-defined destinations or dangerous situations like wrong-way driving on highway. On the other hand, since there is rich under-utilized spectrum resource in remote areas with no broadband connection yet, frequency-agile radio systems can be used to provide broadband internet connectivity there. For example, based on frequency-agile radio platform, we developed a low-cost low-power wireless router that can dynamically operate on TV broadcasting band. It is able to provide high-speed wireless connection to a large area on the ocean. This technology has the potential to bring low-cost high-speed connection to people and industry on the ocean, which will facilitate various maritime applications.

Dedication

To my wife and mother

Acknowledgments

First of all, I would like to thank my advisor Dr. Yaling Yang for her guidance and support through all the past six years. She is very nice and caring. Everything she does is for the best of her students. During my PhD career, I spent a lot of time in exploring my own interests and research topics. She was always supportive and gave me the freedom to grow. Eventually, I am so happy to become who I want to be.

I would like to thank Dr. Gang Wang and Dr. Majid Manteghi for their hands-on guidance on our collaborated projects. I would like to thank Dr. Ranveer Chandra for his guidance during my internship at Microsoft Research. I also would like to thank all the other committee members, Dr. Y. Thomas Hou and Dr. Wenjing Lou, for their valuable feedback on my research work.

I must thank all my collaborators, Shinan Liu, Yuanchao Shu, Dong Wang, Yanzhi Dou, Haoyu Li, Ali Hosseini Fahraji, Pedram Loghmannia and Sihao Sun, for their hard work. Without their help, I can never build systems with such big workload.

I wanna thank my family members, especially my wife Hongying (Katherine) Kuai and my mom Juxiang Zhong, for their unconditional love and support.

My gratitude also extends to my roommate Xing Lu for his company during the last two years.

Grant Information

This dissertation was supported by National Science Foundation (NSF) under grants CNS-1228903, CNS-1054697, CNS-1547366, CNS-1527239, CNS-1750101, and CNS-1717028.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces motivation, challenges, contributions and an outline of the structure of this dissertation.

1.1 Motivation

In the past decade, thanks to the rapid deployment of wireless communication infrastructures and the proliferation of mobile devices, the society has been dramatically changed and people are living in a more convenient way than ever. Recently, a new wireless technology called frequency-agile radio is proposed. Frequency-agile radio can flexibly operate on various frequency bands with different access protocols.

The flexibility of frequency-agile radio enables many new exciting applications. One of such applications is dynamic spectrum access in TV white space. "White space" refers to the locally unused TV channels that are pre-allocated to TV broadcasting. There exists large contiguous white space band that includes, but is not limited to, 470 MHz to 698 MHz

with a single channel bandwidth of 6 MHz. White space technology is a promising solution for bringing high-speed long-distance connectivity to remote areas. It has already been employed by big technology companies like Microsoft and Facebook to connect rural areas in America [1] and developing countries in Africa [2].

Frequency-agile radio also creates new threats to wireless system security. One of such threats is spoofing attack against GPS-based systems. GPS is a space-based satellite radio navigation system, which provides location and time information to many critical applications (*e.g.*, auto-pilot systems, power grids, financial markets, etc.). However, civilian GPS does not have any authentication mechanism and is vulnerable to signal spoofing attack. An increasing number of GPS spoofing incidents are reported around the globe [3].

In this dissertation, we study one threat and one application of frequency-agile radio systems, which are GPS spoofing attack and white space radio. The details of the challenges we face and the contributions we make will be presented in the following section.

1.2 Challenges and Contributions

One big challenge comes from the unknown feasibility, impact, and countermeasures for GPS spoofing attack with frequency-agile radio towards various critical applications. It used to be difficult to launch GPS spoofing attack due to the requirement of expensive cumbersome hardware and advanced knowledge. However, with the development of low-cost portable frequency-agile radio platform and open-source attack software, launching GPS spoofing attack becomes much more convenient. Therefore, we take the first step towards exploring the feasibility, impact, and countermeasures of GPS spoofing attack in two critical application scenarios—road navigation systems and database-driven cognitive radio networks. In the first scenario, an attacker can stealthily manipulate road navigation systems by constantly

spoofing GPS signals in an intelligent way, which can potentially divert navigation users to pre-defined locations or dangerous situations (e.g., wrong-way driving). In the second scenario, an attacker can cause interference with critical infrastructures (e.g., military radars) or denial-of-service to the users by spoofing GPS locations of the networking devices.

The other challenge arises from building white space radio based on frequency-agile radio platform to provide connectivity in new scenarios. White space radio has been demonstrated a promising technology to connect low-population areas on land. But the use cases are very limited because the most populated areas do not have much available white space due to high usage of TV broadcasting service. Therefore, we explore a new application scenario for white space radio—providing connectivity on the ocean where there is rich white space spectrum available. Different from terrestrial environment, ocean environment brings many unique challenges, such as no solid ground, no power lines, and no cable connections. To address these challenges, we propose an energy harvesting maritime mesh network, which is based on energy harvesting buoys and low-cost low-power white space routers designed and implemented on our own.

1.2.1 GPS Spoofing in Road Navigation Systems

Mobile navigation services are used by billions of users around the globe today. While GPS spoofing is a known threat, it is not yet clear if spoofing attacks can truly manipulate road navigation systems. Existing works primarily focus on simple attacks by randomly setting user locations, which can easily trigger a routing instruction that contradicts with the physical road condition (*i.e.*, easily noticeable).

In this dissertation, we explore the feasibility of a stealthy manipulation attack against road navigation systems [4, 5]. The goal is to trigger the fake turn-by-turn navigation to guide

the victim to a wrong destination without being noticed. Our key idea is to slightly shift the GPS location so that the fake navigation route matches the shape of the actual roads and trigger physically possible instructions. To demonstrate the feasibility, we first perform controlled measurements by implementing a portable GPS spoofer and testing on real cars. Then, we design a searching algorithm to compute the GPS shift and the victim routes in real time. We perform extensive evaluations using a trace-driven simulation (600 taxi traces in Manhattan and Boston), and then validate the complete attack via real-world driving tests (attacking our own car). Finally, we conduct deceptive user studies using a driving simulator in both the US and China. We show that 95% of the participants follow the navigation to the wrong destination without recognizing the attack. Inspired by the results, we also explore the direction of sensor fusion based location verification as a defense mechanism [6].

1.2.2 GPS Spoofing in Database-Driven Cognitive Radio Networks

With the proliferation of mobile devices (*e.g.*, laptops, smartphones, etc.), the demand of wireless connectivity has exploded and the existing wireless networks (*e.g.*, cellular, WiFi, etc.) are stretching their capacity. The primary limitation of wireless network capacity is the spectrum scarcity. Based on the fact that some spectrum (*e.g.*, cellular and WiFi) is overcrowded while other spectrum (*e.g.*, broadcast TV and military radar) is underutilized, cognitive radio networks (CRNs) are proposed to ease the imbalance in spectrum utilization. In CRNs, there exists primary users (PUs) and secondary users (SUs). Primary users (*e.g.*, military radars, satellites, TV towers, etc.) have privileges of accessing dedicated spectrum all the time. Secondary users can only opportunistically access available spectrum without interfering with primary users.

FCC has enforced a centralized spectrum database that is in charge of spectrum management

in cognitive radio networks (database-driven CRNs), which currently includes white space networks in TV bands and newly proposed small cell networks in 3.5 GHz. In database-driven CRNs, a secondary user queries the database for available spectrum based on its GPS location. However, this spectrum-allocation mechanism creates a critical vulnerability to GPS spoofing attacks. More specifically, an adversary can take over the GPS of a group of surrounding secondary users, and then manipulate their GPS locations in the query process. Consequently, victim secondary users query the database with false locations and obtain incorrect spectrum information, which can potentially cause primary user interfering and denial-of-service attacks. In this dissertation, we take the first step towards examining the impact of GPS spoofing attacks in database-driven CRNs and propose corresponding attack detection and countermeasures [7, 8]. The results show that as the secondary user density increases, the consequences of primary user interference and denial-of-service become increasingly serious. By leveraging the characteristics of centralized system and the receiving capability of cognitive radios, a combination of proposed countermeasures can effectively mitigate the location spoofing threat.

1.2.3 Developing White Space Radios for Ocean Communications

Ubiquitous wireless connections on land have enabled people to enjoy Internet wherever they go. However, due to a lack of communication infrastructures, high-speed low-cost connectivity on the ocean is still lagging behind. Existing alternative maritime communication technologies all suffer from low-speed and/or high-cost issues.

In this dissertation, we apply white space radios to fill the connectivity gap on the ocean. An energy harvesting maritime mesh network is proposed, which consists of energy harvesting buoys and white space routers. The energy harvesting buoy can generate hundred-watt

power from the motion of ocean wave and current. We design and implement a low-cost and low-power white space router prototype, which is only $523 and consumes 12 watts power. By preliminary link measurements and ocean-surface wave propagation simulation, a 5-kilometer mesh backhaul link with 2 Mbps UDP throughput is predicated. These results have confirmed the feasibility of providing connectivity on the ocean with the proposed network based on white space routers powered by energy harvesting buoys.

1.3 Dissertation Organization

The remainder of this dissertation is organized as follows.

- Chapter 2 explores the feasibility of GPS spoofing attack in road navigation systems.
- Chapter 3 studies GPS spoofing attack and its countermeasures in database-driven cognitive radio networks.
- Chapter 4 presents our low-cost low-power white space radios connecting ocean.
- Chapter 5 draws conclusions.
- Chapter 6 discusses future work, which focuses on practical GPS spoofing countermeasures and mesh networking on the ocean.

Chapter 2

Towards Stealthy Manipulation of Road Navigation Systems

In this chapter, we study stealthy manipulation of GPS-based road navigation systems using frequency-agile radio and preliminary countermeasures.

2.1 Challenges and Contributions

Billions of users around the globe are relying on mobile navigation services today [9]. Ranging from map applications (*e.g.*, Google Maps, Waze) to taxi sharing platforms (*e.g.*, Uber, Lyft), these services depend on accurate and reliable GPS inputs. Recently, GPS systems also start to play a major role in navigating autonomous vehicles, with a key impact on the driving safety [10].

In the meantime, there has been a growing concern about the security of GPS applications. GPS is vulnerable to *spoofing attacks* where adversaries can inject falsified GPS signals to

control the victim's GPS device [11]. Such attacks did happen in the real-world, especially targeting drones and ships. For example, Humphreys *et al.* demonstrated a successful GPS spoofing attack against drones in 2012 [12]. In 2013, a luxury yacht was intentionally diverted from Monaco to Greece by spoofing its receiving GPS signals [13].

To understand the risks of GPS spoofing attacks, researchers have explored to build GPS spoofers to spoof drones, ships and wearable devices [14, 15, 16]. However, these works mainly focus on simple attacks by setting random locations in the target device [14, 15, 16]. Other works have examined GPS spoofing attacks on systems in the open environment (*e.g.,* open air/water) such as drones and ships [12, 13] where a simple GPS change could (stealthily) steer their navigation.

So far, it is still an open question regarding whether attackers can manipulate the *road navigation systems* by spoofing the GPS inputs. The problem is critical considering that navigation systems are actively used by billions of drivers on the road and play a key role in autonomous vehicles. At the same time, the problem is challenging given that most road navigation systems are used (or closely monitored) by human drivers. In addition, naive GPS manipulations are unlikely to succeed primarily because of the physical road constraints. For example, random GPS manipulation can easily create "physically impossible" navigation instructions (*e.g.*, turn left in the middle of a highway). Since the possibility of the attack is not yet clear, most civilian systems don't have any defense mechanisms in place.

In this work, we take systematic steps to explore the feasibility of manipulating road navigation systems *stealthy* by carefully crafting the spoofed GPS inputs. The goal is to manipulate the turn-by-turn navigation and guide a victim to a wrong destination without being noticed. The key intuition is that users are more likely to rely on GPS services when navigating in unfamiliar areas (confirmed via user study). In addition, most navigation systems display the "first-person" view which forces users to focus on the current road and the next turn.

To these ends, if an attacker identifies an attacking route that mimics the *shape* of the route displayed on the map, then it is possible to trigger navigation instructions that are consistent with the physical environment (*e.g.*, triggering the "turning right" prompt only when there is an actual right-turn ahead) to avoid alerting users.

To understand the attack feasibility, we take four key steps[1]. First, we implement a GPS spoofer to perform empirical measurements to understand the attackers' practical constraints and capacities. Second, we design the attacking algorithms and evaluate them based on empirical taxi driving traces. Third, we implement the system and validated it using real-world driving tests (the attacks are applied to the author's car, with careful protections and ethical reviews). Finally, we conduct "deceptive" user studies to examine the feasibility of the attack with other users (non-authors) in the loop and understand key factors to the success of the attack.

Measurements. We show that adversaries can build a portable spoofer with low costs (about $223), which can easily penetrate the car body to take control of the GPS navigation system. Our measurement shows that effective spoofing range is 40–50 meters and the target device can consistently latch onto the false signals without losing connections. The results suggest that adversaries can either place the spoofer inside/under the target car and remotely control the spoofer, or tailgate the target car in real time to perform spoofing.

Stealthy Attacking Algorithm. To make attack stealthy, we design searching algorithms that search for attacking routes in real-time. The algorithm crafts the GPS inputs to the target device such that the triggered navigation instruction and displayed routes on the map remain *consistent* with the physical road network. In the physical world, the victim who follows the instruction would be led to a wrong route (or a wrong destination). We evaluate

[1]Our study received the approval from our local IRB (#17-936).

algorithms using trace-driving simulations (600 taxi trips in total) from Manhattan [17] and Boston [18]. On average, our algorithm identified 1547 potential attacking routes for each target trip for the attacker to choose from. If the attacker aims to endanger the victim, the algorithm can successfully craft special attack route that contains wrong-ways for 99.8% of the trips. Finally, the algorithm also allows the attacker to pre-define a target destination area to lead the victim to.

Real-world Driving Test. We implemented the algorithm and tested it by attacking our own car in a real-world driving test. We have taken careful protection to ensure research ethics (*e.g.*, experiments after midnight in suburb areas, appropriate shield and power control). We demonstrate the feasibility of the attack to trigger the target navigation instructions in real-time while the victim (the author) is driving.

User Study. Finally, we examine the attack feasibility with users (non-authors) in the loop. Due to the risk of attacking real cars, we instead perform a *deceptive* experiment using a driving simulator. We customize the driving simulator to load a high-resolution 3D street map of real-world cities. We apply deception by phrasing the study as a "usability test of the driving software", while we perform spoofing attacks during the experiment (informed consent obtained afterwards). The user study ($N = 40$) was conducted in both the US and China with consistent results. We show the proposed attack is highly effective: 38 out of 40 participants (95%) follow the navigation to all the wrong destinations. Based on our results, we discuss possible solutions moving forward.

In summary, our work makes three key contributions.

- We propose a novel attack that manipulates the road navigation systems stealthily. The proposed algorithm is extensively evaluated using real-world taxi driving traces.
- We implement the attack algorithm and a low-cost portable GPS spoofer. Real-world

measurements and driving tests on the road confirm the attack feasibility.

- We conduct a user study to demonstrate the attack feasibility with human drivers in the loop. The results provide key insights into how common driving habits make users vulnerable.

We hope the results can help to raise the attention in the community to develop *practically deployable* defense mechanisms (*e.g.*, location verification, signal authentication, sensor fusion) to protect the massive GPS device users and emerging GPS-enabled autonomous systems.

2.2 Background and Threat Model

In this section, we start by providing the background of GPS spoofing attacks and describing the unique challenges in *road navigation scenarios*.

Global Positioning System (GPS). GPS is a space-based radio navigation system that provides the geolocation and time information. To date, it consists of 31 satellites in medium Earth orbit where each satellite is equipped with a synchronized atomic clock. Each satellite continuously broadcasts GPS information using Coarse/Acquisition (C/A) code on L1 band at 1575.42 MHz and encrypted precision (P/Y) code on L2 band at 1227.60MHz with 50 bps data rate. P(Y) code is used exclusively by authorized U.S. military receivers and C/A code is not encrypted for general civilian access.

GPS Spoofing Attacks. Civilian GPS is vulnerable to spoofing attacks. GPS spoofing attacks have two key steps: First, in the *takeover* step, attacker lures the victim GPS receiver to migrate from the legitimate signal to the spoofing signal. The takeover phase can be either brute-forced or smooth. In the former case, a spoofer simply transmits the false

signals at a high power, causing the victim to lose track of the satellites and lock on to the stronger spoofing signals. In contrast, smooth takeover begins by transmitting signals synchronized with the original ones and then gradually overpowering the original signal to cause the migration. The advantage of smooth takeover is the stealthiness since it will not generate abnormal jumps in the received signal strength. However, smooth takeover requires specialized hardware to real-time track and synchronize with the original signals at the victim's location (costly) [14, 19]. Next, in the second step, the attacker can manipulate the GPS receiver by either shifting the signals' arrival time or modifying the navigation messages [13, 19].

2.2.1 Threat Model

In this work, we explore a novel attack against *road navigation systems* by spoofing the GPS inputs. In this attack, the victim is a driver who uses a GPS navigation system (*e.g.*, a mobile app) while driving on the road. The victim can also be a person sitting in a GPS-enabled self-driving car. The attacker spoofs the signals of the victim's GPS receiver to manipulate the routing algorithm of navigation system. The attacker's goal is to guide the victim to take a wrong route without alerting the victim (*i.e.*, stealthy). The attack can be realized for three purposes.

- **Deviating Attack.** The attacker aims to guide the victim to follow a wrong route, but the attacker does not have a specific target destination. In practice, the attacker may detour ambulances or police cars to enter a *loop route*.
- **Targeted Deviating Attack.** The attacker aims to guide the victim to a *target destination* pre-defined by the attacker, for example, for ambush, robbery or stealing a self-driving car.

- **Endangering Attack.** The attacker aims to guide the victim into a dangerous situation, for example, entering the *wrong way* on a highway.

In our threat model, the attacker has no access to the *internal* software/hardware of the target GPS device or those of the navigation service. The attacker also cannot modify the navigation services or algorithms (*e.g.*, on Google Maps servers). In addition, we assume the attacker knows the victim's rough destination area (*e.g.*, a financial district, a hotel zone) or the checkpoint that the victim will bypass (*e.g.*, main bridges, tunnels, highway entrances). In later sections, we will justify why this assumption is reasonable and design our attack to tolerate the inaccurate estimation of the victim's destination. We focus on low-cost methods to launch the attack without the need for expensive and specialized hardware.

Compared to spoofing a drone or a ship [13, 12, 20, 15, 16]., there are unique challenges to manipulate the *road navigation systems*. First, road navigation attack has strict geographical constraints. It is far more challenging to perform GPS spoofing attacks in real-time while coping with road maps and vehicle speed limits. In addition, human drivers are in the loop of the attack, which makes a stealthy attack necessary.

The scope of the attack is limited to scenarios where users heavily rely on the GPS device for navigation. For example, when a user drives in a very *familiar* area (*e.g.*, commuting from home to work), the user is not necessarily relying on GPS information to navigate. We primarily target people who drive in an unfamiliar environment. In addition, the attack will be applicable to self-driving cars that rely on GPS and the physical-world road conditions for navigation (instead of the human drivers).

2.3 Measurement-Driven Feasibility Study

We start by performing real-world measurements to understand the constraints of the attacker's capacity in practice. The results will help to design the corresponding attacking algorithms in the later sections.

Portable GPS Spoofer. We implemented a portable GPS spoofer to perform *controlled* experiments. As shown in Figure 2.1. The spoofer consists of four components: a HackRF One-based frontend, a Raspberry Pi, a portable power source and an antenna. The whole spoofer can be placed in a small box and we use a pen as a reference to illustrate its small size. HackRF One is a Software Defined Radio (SDR). We connect it to an antenna with frequency range between 700 MHz to 2700 MHz that covers the civilian GPS band L1 (1575.42 MHz). A Raspberry Pi 3B (Quad Core 1.2GHz Broadcom BCM2837 64bit CPU, 1GB RAM) is used as a central server. It runs an SSH-enabled Raspbian Jessie operating system with a LAMP stack server. GPS satellite signals are generated by an open-source software called Wireless Attack Launch Box (WALB) [21] running on Raspberry Pi. The Raspberry Pi has a cellular network connection and supports remote access through SSH (Secure Shell). By controlling the Raspberry Pi, we can inject the real-time GPS location information either manually or using scripts. We use a 10000 mAh power bank as a power source for the entire system. All the components are available off-the-shelf. The total cost is about 223 US Dollars ($175+$35+$10+$3).

Measurement Setups. We seek to examine the GPS spoofing range, the takeover time delay, and the potential blockage effect from the car body. Before and during the measurements, we have taken active steps to ensure the research ethics and legality. First, the measurement was exclusively conducted in China. We obtained a temporary legal permission from the local radio regulation authority in Chengdu, China for conducting the experiments.

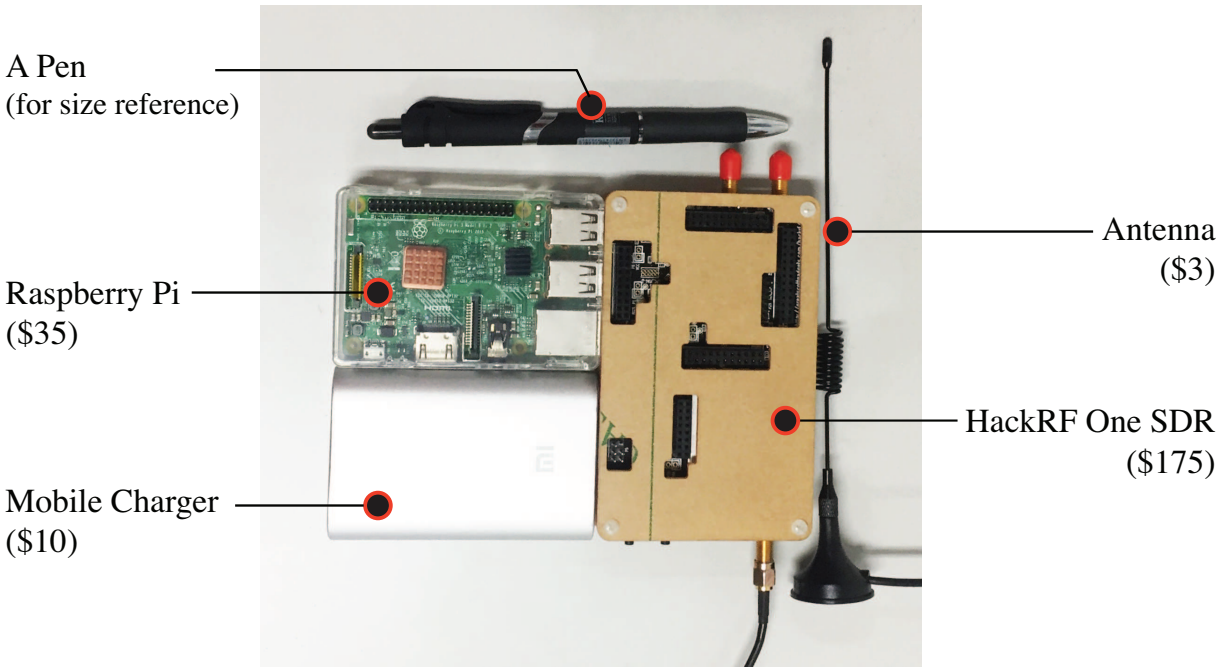
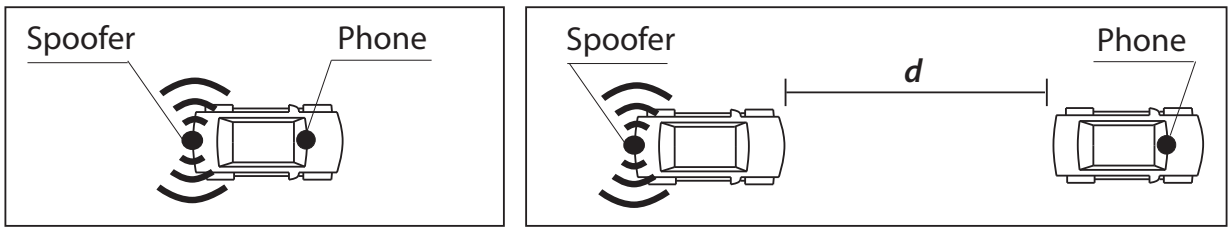


Figure 2.1: A low-cost portable GPS spoofer.



(a) Same-car Test (b) Two-car Test

Figure 2.2: Measurement setups.

Second, we performed the measurements in a large outdoor parking lot *after midnight* when there were no people or cars around (with the permission). Third, we have carefully tested the GPS signal strength at the edge of the parking lot to make sure the signals did not affect the outside areas.

Our measurement focuses on two possible attacking cases to spoof the GPS device in a moving car (Figure 2.2). First, the attacker can place the small spoofer in victim's car or stick the spoofer under the car. The attacker then can remotely login to the spoofer via SSH to perform the attack through a cellular connection. Second, if the spoofer cannot be

Distance (m)	10	20	30	40	50	60
Takeover Time (s)	59.2	37.6	41.2	62.4	35.0	-
Failure Rate	0	0	0	0	0.2	1.0

Table 2.1: Average takeover time and the failure rate.

attached to the victim's car, then the attacker may tailgate the victim's car by driving or flying a drone that carries the spoofer.

Same-Car Setting. In the same car setting, we place the smartphone (XIAOMI MIX2 with Android 8.0) as the victim GPS device in the dashboard area. Then we place the spoofer under the *backseat*, or in the *trunk*. At each position, we SSH the spoofer to take over the GPS lock of the phone. We repeat 10 times and calculate the average takeover time. The result shows that the average takeover time is slightly higher from the trunk (48 seconds) than that from the backseat (35 seconds), but the difference is minor. Note that the takeover is a one-time effort. Once the fake signal is locked in, the connection can sustain throughout the attack.

Two-Car Setting. Then we test to place the spoofer and the smartphone in two different cars, and examine the impact of distance d. We increase d by a step of 10 meters and measure the takeover time. Cars remain static during the measurement. As shown in Table 2.1, the distance does not significantly impact the takeover time, but it does affect the takeover success rate. When the distance is longer, the takeover is more likely to be unsuccessful. The effective spoofing range is 40–50 meters.

We performed additional tests to examine the potential blockage effect of other cars on the road. More specifically, we placed the spoofer and the smartphone in two different cars. Between these two cars, we placed three additional cars as the blockage. The result shows the average takeover time remains similar (41.2 seconds). To further examine the sustainability of the signal lock-in, we fix the location of the spoofer's car, and let the victim's car drive in circles (about 10 mph) while keeping a distance for 15 meters. After driving non-stop for 15

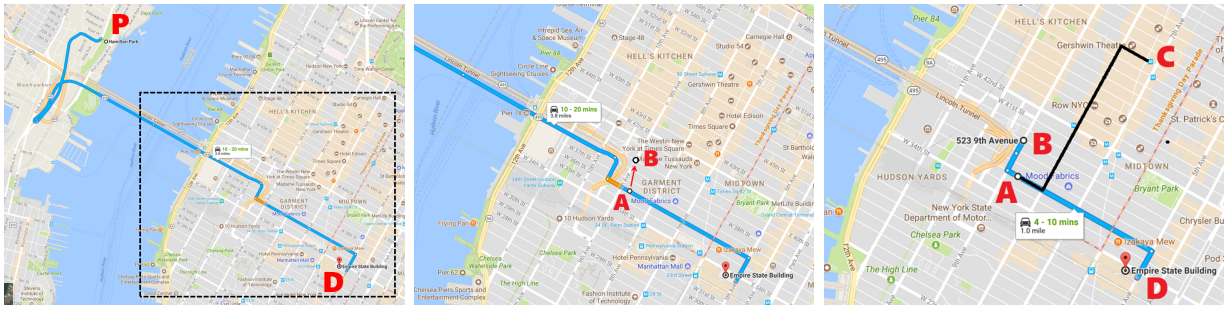
minutes, we did not observe any disconnections, which confirms the sustainability. Overall, the results demonstrate the possibility of performing the GPS spoofing attack in practice.

2.4 GPS Spoofing Attack Method

The measurement results demonstrate the initial feasibility, and the next question is how to make the attack more stealthy. Intuitively, if the attacker randomly changes the GPS information of the navigation device, the driver can easily notice the inconsistency between the *routing information* and *physical road condition*. For example, the spoofed GPS location may trigger the navigation system to instruct a "left turn", but there is no way to turn left on the actual road. In order to make the driver believe he is driving on the original route, the key is to find a virtual route that mimics the shapes of the real roads. In this way, it is possible for the navigation instructions to remain consistent with the physical world. Another contributing factor is that navigation systems typically display the *first person* view. The driver does not see the whole route, but instead, focuses on *the current route* and *the next turn*, which is likely to increase the attacker's chance of success.

2.4.1 The Walk-through Example

The victim is a traveler to the New York City who is not familiar with the area and thus relies on a GPS app to navigate. Figure 2.3a shows the victim is driving from Hamilton Park in New Jersey (P) to Empire State Building in Manhattan (D). Assume that an attacker takes over the victim's GPS receiver at the exit of the Lincoln Tunnel (A) as shown in Figure 2.3c. The attacker creates false GPS signals to set the GPS location to a nearby "ghost" location B. To cope with the false location drift, the navigation system will recalculate a new route



(a) Original navigation route Γ : $P \to D$ (b) Ghost location B (c) Actual path of the victim $A \to C$

Figure 2.3: An attack example: the victim's original navigation route is $P \to D$; At location A, the spoofer sets the GPS to a ghost location B which forces the navigation system to generate a new route $B \to D$. Following the turn-by-turn navigation, the victim actually travels from A to C in the physical world.

between B and D. We call the new route *ghost route*. On the physical road, the victim is still at location A and starts to follow the turn-by-turn navigation from the app. At the same time, the navigation app is constantly receiving the spoofed GPS signals. Eventually, the victim will end up at a different place C. Note that the shape of the $B \to D$ route is similar with that of the $A \to C$ route. Depending on the purpose of the attack, the attacker may pre-define the target destination C or simply aims to divert the victim from arriving the original destination D.

In practice, when the attacker changed the GPS information from A to B, it may or may not trigger the "recalculating" voice prompt in the navigation system. This depends on where B is positioned. If B still remains on the original route (but at a different location from A), then there will be no voice prompt. Otherwise, the voice prompt could be triggered. This turns out to be less of a problem. Our user study (Section 2.7) shows that users often encounter inaccurate GPS positioning (*e.g.*, urban canyon effect in big cities) and don't treat the one-time "recalculating" as an anomaly.

Symbol	**Definition**
G	A geographic area.
$R = \{r_i\}$	Road segments set.
$C = \{c_i\}$	Road segment connection set. $c_i = (r_i, r_{i+1})$.
$L = \{l_i\}$	Road segment length set. $l_i = \|r_i\|$.
$\Phi = \{\phi_i\}$	Connection turning angle set. $\phi_i = \phi(r_i, r_{i+1})$.
S	The merged segment $S_k = [r_i, ...r_{i+j}]$.
P, D, Γ	Starting point, destination, navigation route.
$\Gamma_o, \Gamma_g, \Gamma_v$	Original route, ghost route, victim route.
Loc_a, Loc_g	actual location, ghost location.
$\Omega_{driftDis}$	Max. drifted distance between Loc_g and Loc_a.
v_g, v_a	Ghost speed, actual speed.
Ω_{speed}	Max. speed scale factor $\|(v_g - v_a)\|/v_a \leq \Omega_{speed}$.

Table 2.2: Notation and definition.

2.4.2 Attack Formulation

A successful spoofing attack relies on a careful choice of the ghost location B. The ghost route $B \rightarrow D$ should fit the road map starting from A. In addition, the ghost location B should be close to A so that there will not be an obvious location change on the navigation map screen. In the following, we describe our attack objectives and constraints. Key notations are listed in Table 2.2.

Road Model. As shown in Figure 2.4, a geographic area G is represented by a set of road segments and connection points. R is a set of road segments, and $C = \{c_i = (r_i, r_{i+1})\}$ is a set of connection points. Road segments are inter-connected through connection points. L defines road segment length. Φ quantifies a connection point's turning angle. More specifically, $\phi_i = \phi(r_i, r_{i+1})$, $\phi_i \in [-\pi, \pi)$. We use the counterclockwise convention to calculate the angle [22]. $\phi_i > 0$ and $\phi_i < 0$ indicate a left and right turn respectively.

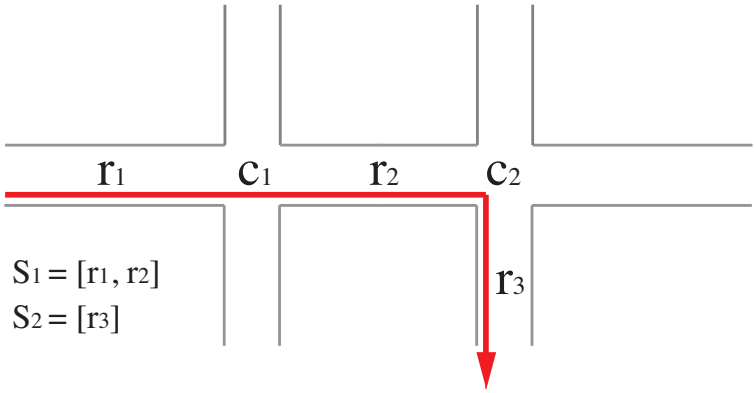


Figure 2.4: Road model example.

Navigation Route. Given a starting point and a destination point, a navigation route Γ is calculated by the navigation system represented by road segments: $\Gamma = (r_1, r_2, ..., r_n)$. In practice, navigation systems typically tell people to keep driving along the road crossing multiple segments before a turn is required. To this end, we further merge adjacent road segments. If the turning angle at connection point (r_i, r_{i+1}) is below a certain threshold θ (say 30°), these two road segments can be merged. After merging such road segments, the navigation route is rewritten as $\Gamma = (S_1, S_2, ..., S_m)$.

Consider a victim is following an *original route* Γ_o to a destination D. At some point, an attacker launches the spoofing attack to change the victim's GPS from its actual location Loc_a to a nearby ghost location Loc_g. This will trigger the navigation system to recalculate a new route from Loc_g to D as the *ghost route* $\Gamma_g = (S_{g_1}, S_{g_2}, ..., S_{g_m})$. Consequently, the victim will follow navigation instructions from Γ_g and will end up traversing a *victim route* $\Gamma_v = (S_{v_1}, S_{v_2}, ..., S_{v_m})$. In our attack, Γ_v should match Γ_g in terms of road segments and connections. Note that Γ_v might contain wrong-way segments (if S_{v_i}'s direction is against the traffic) or loops (if S_v has the same starting and ending point).

Attack Objective. Given the victim's current location Loc_a and destination D, the attack ATK aims to identify feasible victim routes and the associated ghost loca-

tion Loc_g and ghost route Γ_g. We define $O = ATK(G, D, Loc_a) = \{o_1, o_2, ..., o_k\}$, where $o_i = (\Gamma_{vi}, \Gamma_{g_i}, Loc_{g_i})$ such that Γ_{vi} matches Γ_{g_i}. If the attacker aims to divert the victim to a pre-defined destination area C, then the attacker only needs to search the o_i where Γ_{vi} bypasses C.

Constraints. The constraint Ω includes two elements. (1) Location drift constraint $\Omega_{driftDis}$ which defines the maximum drifted distance between Loc_g and Loc_a at the beginning of the attack, *i.e.*, $||Loc_g - Loc_a|| \leq \Omega_{driftDis}$. This is to avoid obvious location change on the navigation map screen. (2) Speed scale factor constraint Ω_{speed} that limits the ghost speed v_g within a reasonable range, *i.e.*, $|(v_g - v_a)|/v_a \leq \Omega_{speed}$. The above practical constraints can be set to different values by attackers in different situations, *e.g.*, depending on the awareness of the human users and the navigation system.

2.5 Detailed Attack Algorithm Design

Next, we describe the detailed design of our attack algorithm. The attack algorithm contains two key components: *road network construction* and *attack route search*. For any target geographic area, we construct the road network from public map data. This is a one-time effort and can be computed offline. In our study, we use the data from OpenStreetMap to build a road network G. Based on the graph, we introduce two algorithms to search the attack routes. The algorithms will return a list of potential attack-launching positions and the corresponding victim routes. Using the searching algorithms, the attacker can also specify a target destination (area) to divert the victim to.

2.5.1 Basic Attack Design

Given graph G, victim's current location Loc_a, destination D and constraints Ω, we design a *basic* search algorithm for the ghost locations and victim routes. Before introducing the algorithm, we clarify on a few assumptions. First, given a starting point and a destination, the attacker needs to compute a navigation route Γ similar to what the victim has. by querying the navigation service that the victim is using (*e.g.*, Google Maps APIs). In addition, the attacker knows the victim's actual location Loc_a. For the same-car setting (*e.g.*, spoofer is attached under the victim car), our spoofer is able to tell the fake GPS signals and the real signals apart, and send the victim's actual location back to the attacker. For the tailgating model, the victim is within the sight of the attacker, and thus Loc_a is known.

Regarding the victim's destination D, it is not necessarily the final destination. It can be simply a rough area (*e.g.*, financial district, hotel zone) or a location checkpoint (*e.g.*, main bridges, tunnels, highway entrances) that the victim will *bypass*. The intuition is simple: for two nearby destinations, the navigation system will return two routes whose *early portions* are similar (or even identical). With an estimated D, the attacker can generate a valid ghost route to match the early portion of the victim's route, which is sufficient to trigger the fake turn-by-turn navigation instructions. In practice, attackers may obtain D from different channels, such as the target user's social media location check-ins, destination broadcasting in taxi-hailing services, and identifying the checkpoints that the user must traverse (*e.g.*, the Lincoln Tunnel entrance when traveling between New Jersey and Manhattan). Technically, attackers can also probe the victim's destination area by sequentially drifting the ghost location and observing the reactive movements of the victim, which has shown to be feasible [13].

As illustrated by Algorithm 1, the basic algorithm begins by selecting a ghost location Loc_g

Algorithm 1: Basic attack algorithm

Require: $G, D, Loc_a, \Omega_{driftDis}, \Omega_{speed}$
Ensure: $O = \{o_1, o_2, ..., o_K\}$, $o_i = (\Gamma_v, \Gamma_g, Loc_g)_i$
Initialization: $O \leftarrow \emptyset$
Preprocessing: Find all candidate ghost current locations $\{Loc_{g_1}, Loc_{g_2}, ..., Loc_{g_N}\}$ within $\Omega_{driftDis}$ distance from Loc_a
for $i = 1$ to N **do**
    $\Gamma_g = (S_{g_1}, S_{g_2}, ..., S_{g_m})$, where Γ_g is obtained through an API getNavigationRoute(G, Loc_{g_i}, D)
    $U_0 = \{[r_{ac}]\}$, where $Loc_a \in r_{ac}$
    $U_1, U_2, ..., U_m \leftarrow \emptyset$
    for $j = 1$ to m **do**
        if $U_{j-1} == \emptyset$ **then**
            break
        end if
        for $u \in U_{j-1}$ **do**
            $v \leftarrow u.endpoint$
            for $s \in$ segments with starting point of v **do**
                if s has passed the search criteria **then**
                    Append $u.append(s)$ to U_j
                end if
            end for
        end for
    end for
end for
return O

from all the connection points within the distance bound $\Omega_{driftDis}$ from the actual location Loc_a. Then, a ghost navigation route $\Gamma_g = (S_{g_1}, S_{g_2}, ..., S_{g_m})$ from the ghost location to the destination is calculated. In order to find as many victim routes as possible, we traverse the graph from the actual location via an m-depth breadth-first search. We keep the candidate routes that satisfy the following criteria at every step:

- *Turn Pattern Matching:* To make sure the navigation instructions of the ghost route can be applied to the victim route, we need to match the turn patterns of the two routes: $\phi(S_{v_i}, S_{v_{i+1}})$ and $\phi(S_{g_i}, S_{g_{i+1}}) \in$ same maneuver instruction category.

- *Segment Length Matching:* Given a speed scale factor Ω_{speed}, the travel distance of the ghost should be within $(1 \pm \Omega_{speed})$ times the victim's actual travel distance on each segment, namely, $(1 - \Omega_{speed}) \cdot S_{v_i} \leq S_{g_i} \leq (1 + \Omega_{speed}) \cdot S_{v_i}$. This guarantees segment

length on the ghost and victim route is similar.

In the worst case, the computational complexity is exponential to the number of road segments connected by one intersection. However, thanks to the searching criteria, the unqualified victim routes can be terminated in the very early stage.

Algorithm 2: Iterative attack algorithm

Require: $G, D, \Omega_{driftDis}, \Omega_{speed}$, O_0, I, attack goal
Ensure: O_i, where $i = 1, 2, ..., I-1$

Initialization: $carryover\_\Gamma_v \leftarrow \emptyset$, $carryover\_\Gamma_g \leftarrow \emptyset$, $O_i \leftarrow \emptyset, i = 1, 2, ..., I$
for $i = 1$ to $I-1$ **do**
  if attack goal has been achieved **then**
    return
  end if
  $U_1, U_2, ..., U_m \leftarrow O_{i-1}$
  for $j = 1$ to m **do**
    if $U_j = \emptyset$ **then**
      break
    end if
    for u in U_j **do**
      $\Gamma_{g_u} \leftarrow O_{i-1}[u]$
      for $k = start_j$ to end_j **do**
        Append $basic\_attack(G, D, \Gamma_{g_u}[k])$ to O_i
        Append $\Gamma_{g_u}[:k]$ to $carryover\_\Gamma_g[u]$
        Append $\Gamma_{v_u}[:\hat{k}]$ to $carryover\_\Gamma_v[u]$
      end for
    end for
  end for
  Save $(O_i, carryover\_\Gamma_v, carryover\_\Gamma_g)$
end for
return

2.5.2 Iterative Attack Design

In basic attack, the attacker only shifts the GPS position once from Loc_a to Loc_g. Here, we propose an *iterative attack*, which allows the attacker to create multiple drifts at different locations, while the victim is driving. By iteratively applying the basic attack algorithm, the attack performance can be significantly improved since partially matched victim-ghost routes can be used for searching new routes as the victim moves. As shown in Algorithm 2, for

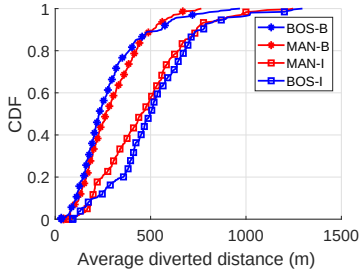
each iteration, we first check if the attack goal has been achieved. If not, we create another location shift on the new ghost route segments from the previous iteration, and apply the *basic searching algorithm*. The attacker goal can be "reaching a pre-defined destination" or "entering a wrong way", which helps to terminate the searching early.

2.5.3 Targeted Deviating Attack

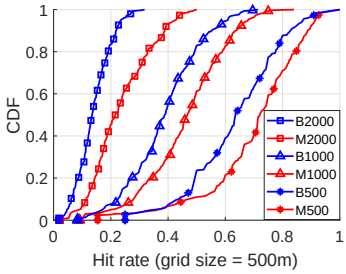
With the above searching algorithms, the attacker may launch the attack by specifying a target destination area. More specifically, attacker can divide the geographic area into grids (width w) and then pick one of the grids as the target destination. Then the attacker can run the basic or iterative algorithm to compute all the possible victim routes and identify those that bypass the pre-selected grid. The attacker can terminate the searching algorithm earlier once a victim route hits the destination grid. Intuitively, the success of the attack depends on the road map of the city and the size of the grid (w). There is also a limit on how far away the target destination can be set given the condition of the original route. We provide detailed evaluations in the next section.

2.6 Attack Evaluation

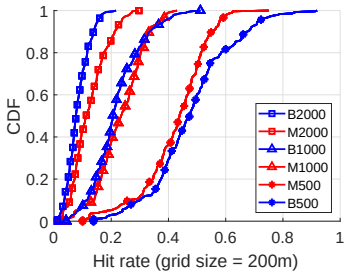
Next, we evaluate the proposed algorithms using both trace-driven simulations and real-world driving test. Our simulation is based on empirical driving traces collected from Manhattan and Boston. Given different attack goals, we seek to understand how well the algorithms can identify the qualified ghost routes and ghost locations. Then we implement algorithms and conduct real-world driving tests to validate the attack feasibility in real-time.



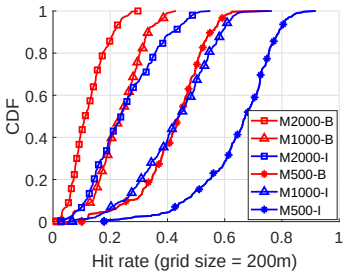
(a) Avg. diverted distance



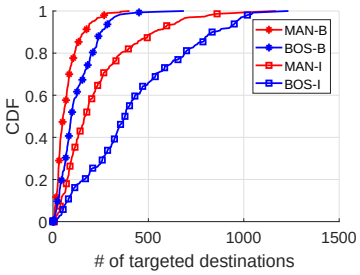
(b) Hit rate (grid size=500m, basic attack)



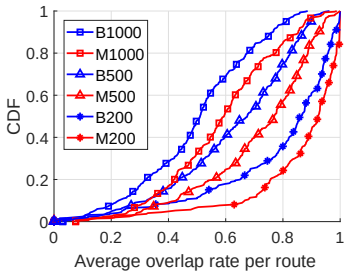
(c) Hit rate (grid size= 200m, basic attack)



(d) Hit rate (grid size= 200m, iterative attack)



(e) # of Diverted destinations



(f) Overlap rate: estimated vs. true routes

Figure 2.5: Attack results in Manhattan (MAN) and Boston (BOS). B = Basic Attack; I = Iterative Attack; M500 = Manhattan with a $r = 500m$ considered area or 500m-grid-size estimation area; B500 = Boston with a $r = 500m$ considered area or 500m-grid-size estimation area.

2.6.1 Simulation Experiments

Our attack is more suitable to run in the cities where the road networks are dense. We use the maps of Manhattan (NY) and Boston (MA) since the two cities have different road networks [23] to test our algorithm under different road conditions. For example, Manhattan has more regular grids with a 17.8° standard deviation of turn angles, while Boston has more curvy roads (20.5° standard deviation). In addition, Manhattan has a lower road segment density (51 segments/km^2) compared with that of Boston (227 segments/km^2). We construct the road network based on the OpenStreetMap database [23].

Driving Trace Dataset. To examine the attack performance on realistic driving trips, we obtain taxi trip datasets from NYC Taxi and Limousine Commission (TLC) [17] and the Boston taxi trace dataset used by MIT Challenge [18]. We randomly select 600 real-world taxi trips (300 per city). These traces cover the large area and various road types (visualization is in Appendix A.1). The average length of the routes is 900m in Manhattan (MAN) and 2000m in Boston (BOS).

Evaluation Configurations. For each taxi trip, we exhaustively run the search algorithm at *each road segment* to identify all the possible attack locations (and the corresponding ghost locations and victim routes). This provides a "ground-truth" on the possible attack options available to the attacker. Then we discuss how these options meet the attacker's goals.

For constraint parameters, we set the maximum drift distance $\Omega_{driftDis} = 400$m. A measurement study shows that a GPS drift of less than 400m is common during active driving [24]. In addition, given the speed limits in the two cities are 25 to 30 mph, we set $\Omega_{speed} = 0.2$ assuming a 5–6 mph speed offset is unnoticeable. For iterative attack, we run two iterations as a comparison with the basic attack. Our algorithm also requires calculating the "turning angle" to compare the shape of the roads. We follow Waze's standard [25] to identify the continuous road ($[-30°, 30°]$]), left/right-turn ($[30°, 170°]$), and U-turn ($[170°, 180°]$). We implement the algorithms in Python, and run the evaluation on a server with a 192GB RAM and 24 cores.

2.6.2 Evaluation Results

The performance metric depends on the specific goal of the attacker. Recall in our threat model (Section 2.2.1), we defined three types of attacks which need different evaluation

metrics. Below, our metrics are all based on each of the taxi trips (per-trip metric).

Deviating Attack. If the attacker simply aims to divert the victim from reaching the original destination, the evaluation metric will focus on the *number of victim routes* available to the attacker, and the *diverted distance* for each road segment on victim routes. More specifically, given road segment r_v and the original navigation route $\Gamma_o = (r_1, r_2, ..., r_n)$, the diverted distance for r_v is calculated as $\min_{i=1,2,...,n}\{||r_v - r_i||\}$, where $||r_v - r_i||$ is the distance between two road segments. By running the basic algorithm, we successfully identify at least one victim route for all the 600 taxi trips. On average, each trip has 335 qualified victim routes, indicating a wide range of attack opportunities. The iterative algorithm (iteration $i = 2$) identified many more victim routes (3,507 routes per trip). Note that for BOS-I, the results are based on 260 trips with distance capped at 6000m. Figure 2.5a shows average diverted distance per trip. Again, the iterative algorithm is able to identify victim routes that are further away from the victim's original routes. On average, about 40% of the trips can be diverted 500 meters away.

One specific goal of the Deviating Attack could be delaying the victim's trip by leading the victim to loop routes. Given a taxi trip, we examine whether there exists a victim route that contains a loop. Using the basic algorithm, we find at least one loop victim route for 256 out of 300 (85.33%) taxi trips in Manhattan, and 294 out of 300 (98%) trips in Boston.

Targeted Deviating Attack. If the attacker aims to divert the user to a pre-defined location, the evaluation metric will focus on *hit rate*. For a given taxi trip, the hit rate reflects how likely a victim route can bypass the attacker-defined destination to achieve targeted diverting. Given a taxi trip, we first circle an area around the taxi route as the considered attack area. The area is of a similar shape of the taxi route with a radius of r (*i.e.*, any location inside this area has a distance shorter than r to the taxi route). We divide

the area into grids (width w). The attacker can pick a grid inside the area as the target destination. Hit rate is the ratio of the grids that the victim can be diverted to over all the grids in the attack area. An illustration is available in Appendix A.2.

Figure 2.5b shows the hit rate of the basic attack. We set the grid size as w=500m and then vary the radius r of the considered area. The result shows that we can achieve about 70%, 47%, 20% median hit rate in Manhattan with r= 500m, 1000m, and 2000m respectively. This indicates that even a randomly selected destination grid is highly likely to be reachable. Not surprisingly, victim routes get sparser when it is further away from the original route. Note that even with 20% hit rate in 2000m range, if the attacker provides three candidate target destination grids, the success rate will be higher $1 - (1 - 0.2)^3 = 48.8\%$. Comparing Figure 2.5b and Figure 2.5c, we show that a larger grid leads to a higher hit rate. In practice, attacker can use a larger grid if he can tolerate some inaccuracy of the target destination *i.e*, the victim is led to a nearby area instead of the exact target location.

Figure 2.5d shows that the iterative attack algorithms can significantly increase the hit rate (blue lines) comparing to those of the basic algorithm (red lines). In addition, Figure 2.5e shows that iterative algorithm also significantly increases the total number of bypassed grids by all the victim routes, *i.e.* the number of potential *target destinations* for the attacker.

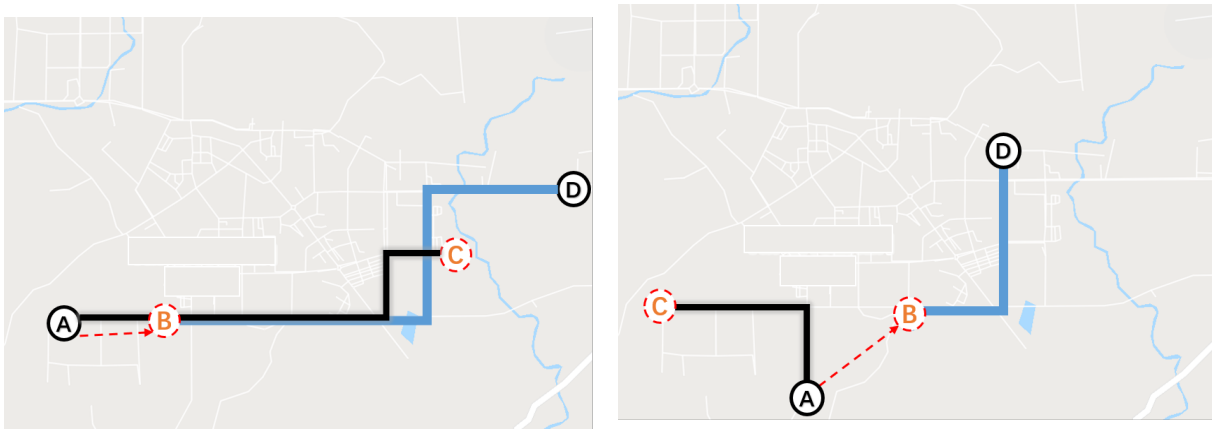
Endangering Attack Result. If the attacker aims to endanger the victim, then we focus on the *wrong-way rate*. Given a taxi trip, we aim to find at least one victim route that contains a wrong way segment. The basic algorithm identified a wrong-way victim route for 599 out of the 600 taxi trips (99.8%). Notably, 90.4% of trips have the victim routes that contain a highway type of wrong way segment, which incurs real danger.

Boston vs. Manhattan. Boston has denser road networks and irregular road shapes. Manhattan has a sparser and grid-like road network. The road network features affect the

attack performance. As shown in Figure 2.5b and Figure 2.5c, the smaller grid size helps Boston to reduce the hit rate deficit against Manhattan, since the dense road segments in Boston allow us to divert the victim to more precise destinations. In addition, since Boston has more irregular roads, it is more difficult to search for a long victim route that matches the ghost route. On the contrary, Manhattan's grid-like road structure yields a better match for long victim routes as shown in Figure 2.5a. Our attack works for small cities, but will yield fewer options for attackers (validated in our real-world driving test).

Original Destination Estimation. Recall that to run the attack algorithm, the attacker needs some knowledge about D, the original destination of the victim. Here, we evaluate the impact of the inaccurate estimation of D. More specifically, given a true D, we randomly set an estimated D' that is within 200m, 500m or 1000m. Using D', we generate the estimated route, and then calculate the overlapped portion with the original route. As shown in Figure 2.5f, even if the estimated destination is not accurate, there are enough overlapped segments (in the beginning) that can help to generate the victim routes. For example, even with 1000m error, the attacker can divert the victim using the first half of the ghost navigation route (medium 0.5 overlap rate).

Computation Time Delay. The ghost route searching can be completed within milliseconds for the basic attack. The average searching time for one ghost location candidate is 0.2ms in Manhattan and 0.3ms in Boston. The iterative attack takes a longer but acceptable time: 0.13s in Manhattan and 0.32s in Boston. Note that attacker can always pre-compute the route (within a minute) before the victim arrives the attack location.



(a) On-Route Attack (b) Off-Route Attack

Figure 2.6: The original routes and victim routes in the real-world driving tests.

2.6.3 Real-world Driving Tests

We implemented the full attack algorithm and validated the feasibility through real-world driving tests. Two authors performed the same-car attack using our own car. One author acted as the driver (victim) who strictly followed the navigation instructions from the Google Maps (v9.72.2) running on the phone (XIAOMI MIX2 with Android 8.0 and HUAWEI P8 with Android 6.0). The other author sat on the backseat to operate the spoofer and ran the attack algorithm on a laptop. As previously stated, the spoofer can tell apart the fake GPS signals with the real ones, and thus the attacker knows the true location of the victim. The goal of the real-world driving tests is to examine if the spoofer can trigger the fake navigation instruction in *real-time* right before users need to make a navigation decision.

Similar as early measurements, we obtained a legal permission from the local radio regulation authority, and conducted the experiments exclusively in China. In addition, we have taken active steps to make sure the spoofing signals did not affect innocent users or cars. More specifically, we performed our measurements in a suburb area *after midnight* when there were almost no other cars on the road. To minimize the impact of the spoofing signals, we reduce

the transmit power of the spoofer to the minimum (-40 dBm) and then use attenuators (30 dB) to reduce the signal strength after locking in. The metal structure of the car also acts as a shield to contain the spoofing signals (about 15 dB attenuation). In addition, there is another -42.41 dB free space propagation loss at a two-meter distance. This means, beyond two meters away from the car, the signal strength is already very weak (about -127.41 dBm), which cannot take the lock of any GPS devices.

In total, we tested on two different routes as shown in Figure 2.6. In both screenshots, lines $A \rightarrow D$ represent original routes. Blue lines stand for ghost routes, while black lines stand for victim routes. A is the user's actual location and B is the corresponding ghost location. C is the user's diverted destination, D is the original destination. In the first case (Figure 2.6a), the attacker set the ghost location to another location *on the original route*. Our test showed that this indeed can avoid triggering the "re-calculating" voice prompt. The route took nine minutes and the driver was successfully diverted to the pre-defined location 2.1 kilometers away from the original destination. In the second case (Figure 2.6b), the attacker set the ghost location *off the original route*, which triggered a "re-calculating" voice prompt. This time, the driver drove five minutes and was diverted 2.5 kilometers away. In both cases, the smartphone was locked to the spoofed signal without dropping once. The sequences of fake locations were fed to the phone smoothly with a 10Hz update frequency. Despite the potential cross-checks of heading and filters embedded in Google Maps, the navigation instructions were triggered in time.

2.7 Attacks with Human in the Loop

Next, we examine how stealthy the attack can be to human drivers (victims) through a user study. As previously stated, the attack focuses on people who drive in the unfamiliar

locations because they would be more likely to rely on the GPS navigation (instead of their own knowledge of the roads). We will also check the validity of this assumption in the user study. Our study cannot involve attacking human subjects when they drive real cars due to safety implications. Instead, we conduct a *deceptive* user study in a simulated environment using a customized driving simulator. Our study received the approval of our local IRB (#17-936).

2.7.1 User Study Methodology

Our user study examines three high-level research questions. *R1:* how do users use GPS navigation systems in practice? *R2:* under what conditions is the GPS spoofing attack more likely to deceive users successfully? *R3:* what are the user perceptions towards the GPS spoofing attack? We explore the answers with three key steps: pre-study survey, driving tests, and post-study interview. To avoid alerting the participants, we frame the study with a non-security purpose, stating that the study is to test the usability of our simulation software. We debrief users after the driving test to obtain the informed consent. The study takes about 50 minutes and we compensate each participant $10.

Pre-study Survey. The survey asks two questions: (1) how often do you use GPS navigation services when driving in familiar locations (*e.g.*, home and work) and unfamiliar locations (*e.g.*, visiting a new city). (2) what information provided by the navigation service do you primarily rely on during driving?

Driving Tests. To simulate a realistic driving scenario, we build a simulator by modifying a popular driving simulation game "Euro Truck Simulator II" (ETS II) [26]. We use ETS II for three reasons. First, the game presents the *first-person view* with realistic vehicle interior and dashboard. In addition to the front view, the participant can easily move the

view-angle (to see through the passenger window and the backseat) by moving the cursor. This provides a wide view range to the participant. Second, the simulator can load real-world maps where the 3D street view mimics the reality. Figure 2.7b and Figure 2.7c show the side-by-side companion of the game view (of a 3:1 map) and the actual street view (from Google Street View) at the same location. Because the street view is rendered in a high-resolution, the street signs and road names are clearly displayed. Third, the simulator SDK allows us to control the day-and-night settings and special weather conditions. We provide a demo video under this link[2].

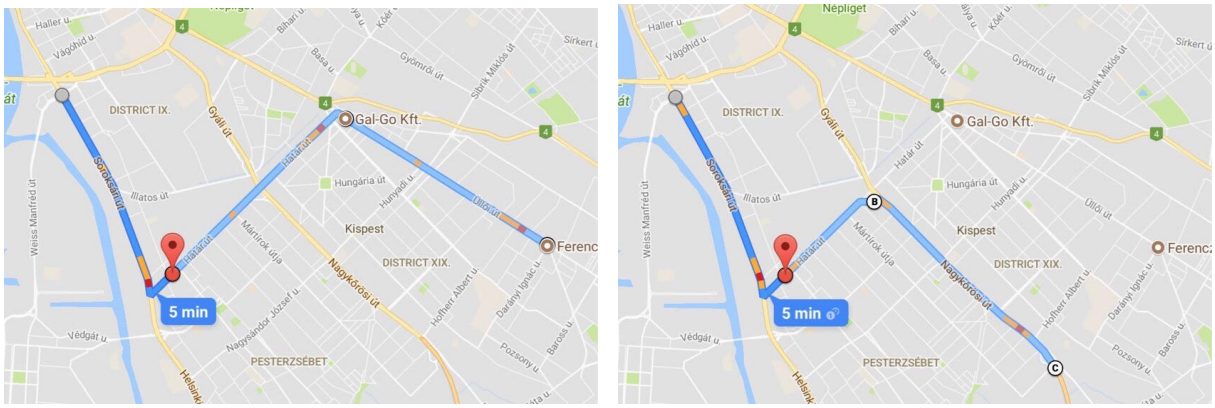
For the driving test, we simulate attacking a victim who drives in a new city. We display the driver's view on a 22 inch LED display (1920 x 1200) and load a 3:1 map of Budapest in Hungary [27], which is considered an unfamiliar city for our participants. At the same time, we run Google Maps on an Android smartphone as the navigation app. The app provides turn-by-turn navigation, and the voice prompt reads the street names. The smartphone is placed in front of the LED display (near the "dashboard" area) as shown in Figure 2.7a. For ethical and legal reasons, we cannot directly spoof the GPS signal of the smartphone. Instead, the smartphone runs a dedicated app (developed by us) to fetch GPS sequences from a server. The server reads the GPS information from the driving simulator in real time and generates fake locations for the smartphone. In this way, we can directly manipulate the GPS read of the smartphone for the user study.

To examine user reactions to the attack, we assign each participant driving tasks. The participants will drive to deliver packages to a given destination following the navigation of Google Maps. Figure 2.8 shows the driving routes used in our user study. Figure 2.8a shows the original route that the participant is supposed to take. Figure 2.8b shows the route to which the attacker aims to detour the participants. This route is chosen because it

[2]Demo: `https://www.dropbox.com/sh/h9zq8dpw6y0w12o/AABZiKCUOhe44Bu1CtHZzHLta`

(a) Experiment Setups (b) ETS II Game View (c) Google Street View

Figure 2.7: User study setups; The ETS II Game View is comparable to the Google Street View at the same location.



(a) Original Route (b) Victim Route

Figure 2.8: The original and victim route for the user study.

contains a high-way in the victim route, and only local-ways in the original route. These are the clear discrepancies for the victim to recognize. We tune two parameters: *driving time* (day or night) and *weather* (rainy or clear). The participant will deliver the package four times (on the same route) in this order: "rainy night", "clear night", "rainy day", and "clear day". This order makes it easier to recognize the attack in the end than at the beginning. The experiment stops whenever the participant recognizes the attack. Note that the attack covers the takeover phase when the phone loses the GPS signal for a while and then jumps to a new location.

To help the participants to get familiar with the driving simulator, we spend about 5–10

minutes to let the participants play with the simulator before the real tests. We also use the time to train the participants to "think-aloud" — expressing their thoughts and actions *verbally*. During the real test, we encourage the participants to think-aloud and record the audio.

Post-study Interview. In the interview, we first debrief the participants about the real purpose of the study. Second, we ask about their perceptions towards GPS spoofing attacks. Third, we let the participants comment on the key differences between using the driving simulator and their real-world driving. The participants can withdraw their data at any time and can still receive the full compensation.

Recruiting Participants. We performed the user study in both the U.S. and China. The user study materials have been translated into the respective languages of the participants. Given that the study requires the participants to physically come to the lab (and stay for about one hour), we cannot perform the study on a massive scale. With a limited scale, our goal is to recruit a diverse sample of users. We distribute our study information on social media, user study websites, and student mailing lists. We recruited 40 participants (20 in the U.S. and 20 in China). Among the 40 participants, there are 30 male and 10 female. 17 people are 26–35 years old, and 20 people are 18–25, and 3 people are 36–50. Regarding the driving experience, 22 people drive for <3 years, 16 people drive for 3–10 years, and 2 people drive for 10–20 years. Our participants are slightly biased towards tech-savvy users: 20 users (50%) have a Computer Science background.

2.7.2 User Study Results

Driving and Navigation Habits. *Users are more likely to use GPS navigation systems when traveling in unfamiliar areas.* We ask users to rate how often they use GPS in

"familiar", "not-too-familiar" and "unfamiliar" areas with a scale of 10 (1=never; 10=almost every time). The U.S. participants' the average score for unfamiliar places is much higher (7.85) than familiar locations (4.55). The results from China are consistent (10.0 vs. 3.93). This means, our attack may not be applicable to familiar area since people don't rely on GPS.

Users are more likely to rely on the voice prompt and visual instructions than the textual information. We present a Google Maps screen and ask which information the participant typically rely on to make driving decisions (a multi-choice question). In the U.S., 13 users (68.4%) choose voice prompt, 11 users (57.9%) rely on visual elements such as road shapes and arrows, and only 6 users (31.6%) choose textual information such as street names. The results from China are consistent. These results are in favor of our attack, which is designed to manipulate the voice and the visual elements.

User Reactions to GPS Spoofing Attacks. Our attack has achieved a high success rate (95%). Out of 40 people, only one U.S. participant and one Chinese participant recognized the attack. The rest 38 participants all finished the four rounds of driving tasks and followed the navigation to reach the wrong destinations.

Both participants recognized the attack because they detected certain inconsistency between the navigation information and the surrounding environment on the road. The U.S. participant (user#38, m, 18-25, driving <3 years) recognized the attack during the second round (clear night). He was driving on a high way with a gas station on his right when he realized that the Google Maps showed that he was on a local way without a gas station nearby. He also checked the street signs and recognized the inconsistent road names. The Chinese participant (user#5, m, 26-35, driving <3 years) recognized the attack during the first round (rainy night), alerted by the "highway and local way" inconsistency.

During the driving task, we observe that almost all the participants noticed when the GPS signals are lost during the takeover phase (about 30 seconds), but still kept driving on the road. Once the GPS signal came back, they continued to follow the navigation instructions. Our interview later shows most users have experienced malfunctioned GPS before, which is not enough to alert them.

User Perceptions to the Attack. During the interview, we find that *most users have experienced GPS malfunction in real life*. 95% of the users commented that they experienced GPS malfunction in real life such as losing GPS signals and wrong positioning. User#39 stated that she even had a car accident due to the poor GPS signals. Some users mentioned that it could be very challenging to check road signs constantly. For example, user#03 stated "*the roads in the U.S. all look similar. Sometimes I notice the road signs, but not when I drive fast*". In addition, users *do not understand how GPS spoofing works*, Among the 40 participants, only eight users can explain GPS spoofing correctly.

We encourage the participants to comment on the differences between using the simulator and real-world driving. The most common response is the usage of the keyboard and mouse to control the car for steering and acceleration. User#10 also commented that they can drive more recklessly in the simulation game: "*The most different part is that you are afraid of nothing. You are not afraid of red lights, crashing either.*" These are the limitations of the controlled and simulated studies.

Discussion. Overall, the results show that our attacks are highly effective even when human drivers are in the loop. The results also point out three types of inconsistencies that are likely to alert users: (1) inconsistency between highway and local ways; (2) inconsistent street names; (3) inconsistent landmarks (*e.g.*, gas station). More advanced attacks can further avoid the "highway - local way" inconsistency by filtering out such routes. The

Table 2.3: Comparison of different countermeasures.

Mechanism		$ Cost	Deployment Overhead	Effectiveness	Robustness
Modification-based	Encryption & authentication	High	High	High	High
	Ground infrastructures	High	High	High	High
	GPS receiver hardware	Medium	High	High	High
	GPS receiver software	Low	Low	Low	Low
Modification-free	External location verification	Low	Low	Low	Low
	Internal sensor fusion	Low	Low	Low	Low
	Computer vision	Low	Low	Medium	Unknown

other two factors depend on whether the driver has the habit (and has the time) to cross-check the surrounding environment. In addition, our interview reveals that most people have experienced GPS malfunction in real life, which makes them more tolerable to GPS inconsistencies. In addition, since people are more likely to rely on visual and voice prompt, it increases the attacker's probability of success. Our study still has limitations, which are discussed at the end of the chapter.

2.8 Countermeasures

Our study demonstrated the initial feasibility of manipulating the road navigation system through targeted GPS spoofing. The threat becomes more realistic as car-makers are adding auto-pilot features so that human drivers can be less involved (or completely disengaged) [28]. In the following, we discuss key directions of countermeasures and explored initial feasibility of sensor fusion based location verification.

2.8.1 Overview of Existing Countermeasures

In Table 2.3, we classify different methods based on whether (or how much) they require modifications to the existing GPS. Modification-based methods require changing either the GPS satellites, ground infrastructures, or the GPS receivers. Modification-free methods

typically don't need to change existing GPS, which make them more attractive to be adopted.

Modification-Based Approaches. First, the most effective solution is to upgrade the civilian GPS signals to use the P(Y) code encryption. Researchers also proposed signal authentication for next-generation GNSS (Global Navigation Satellite System) [29, 30]. However, this approach is extremely difficult to prevail in a short term, given the massive number of civilian GPS devices already shipped and deployed in the short term.

Second, trusted ground infrastructures to help GPS devices to verify the location and related techniques include trusted verifiers, distance bounding protocols [31, 32], multilateration [33], multi-receiver crowdsourcing [34] and physical-layer feature checks [35]. However, due to the constraints in government policies, and the significant costs, dedicated ground infrastructures are also unlikely to be widely deployed.

Finally, we can modify the GPS receivers. For example, the angle-of-arrival of signals can help to estimate the transmitter's location for authenticity check. This requires a large directional antenna array [36], or special moving antenna [37]. Such hardware modifications are not applicable to the billions of mobile phones. At the software level, consistency-check algorithms can help to detect the side effects of non-smooth GPS takeover [38, 39, 40]. In addition, the GPS receiver can also lock on additional satellites [41] or synchronize with other GPS receivers [11] to identify spoofing. However, these methods often suffer from the multi-path effect and are vulnerable to smooth takeovers [14].

Modification-Free Approaches. First, location verification can leverage existing GNSS signals (*e.g.*, Galileo, GLONASS, Beidou) [42], and wireless network signals [43]. These external location verifications help but cannot stop the attacker completely because civilian GNSS signals are also unencrypted. The attacker can perform multi-signal jamming or spoofing against both signals [14]. Similarly, the network location is based on the MAC

address of the WiFi or cell tower ID, which can also be jammed or spoofed [44, 45].

In addition, a navigation system may cross-check the GPS locations with dead reckoning results based on inertial measurement unit (IMU) sensors (*e.g.*, accelerometer, gyroscope, magnetometer) [46, 47]. However, this method in general suffers from accumulative IMU sensor errors and becomes ineffective as the time drifts.

Computer Vision Based Location Verification. We believe a promising defense direction is to use computer vision techniques to automatically cross-examine the physical-world landmarks and street signs with the digital maps. Recall that in our user study, the two participants recognized the attack in a similar way. Given the proliferation of cameras/LIDARs on mobile devices and vehicles, vision-based location verification only requires software level upgrade. So far, vision-based techniques can accurately localize vehicles (up to 3m) using visual odometry and road maps [48, 49]. SLAM (Simultaneous Localization And Mapping) can also localize images based on geo-referenced street view databases [50].

What remains unknown is the *robustness* of vision-based methods against adversarial manipulations. Recent works [51, 52] demonstrated that image classifiers can be easily fooled by adding small adversarial noises to the input (*e.g.*, a street sign image). In our scenario, although it is very unlikely for adversaries to modify all the *physical* street signs and landmarks along the road, the high sensitivity of image classifiers is still a potential concern. Recently, researchers have proposed methods to enhance the robustness of image classifiers [53, 54, 55]. Further research is needed to understand the feasibility of vision-based location verification.

2.8.2 Inertial Sensor and Network Location Based Defense

To understand how well the inertial sensor fusion method can defend against the stealthy attack, we perform a preliminary experiment using real-world driving data collected by 4

users in Boston [23]. Specifically, the raw data contains GPS location, network location from WiFi or cellular towers, acceleration, angular velocity, and geo-magnetic traces from 4 users who use different smartphone models. In the preprocessing step, we perform device-vehicle coordinate alignment and sensor drift calibration.

The defense goal is to verify the reported GPS locations in real time. In other words, the defense mechanism should be able to classify under-attack driving (positive) and normal (negative) driving based on the real-time sensor data. Hence, we generate under-attack sensor data in two steps: (1) We launch a basic attack ($\Omega_{driftDis} = 400m$, $\Omega_{speed} = 0.2$) against the user at a random time point, which finds a ghost location and a ghost route very similar to the real driving route. (2) After that, we replace GPS locations collected on the victim route with ghost GPS locations according to the scale factor $\Omega_{speed} = 0.2$, while retaining other sensor data and network locations. In this way, we can construct the sensor trace where the victim exactly followed ghost navigation instructions and traversed the victim route (actual route in the dataset). Both under-attack and normal sensor traces are sliced into chunks (with a variable-sized time window) to feed a classifier. The entire dataset is split into training, validation, and test datasets with ratios of 60%, 20%, and 20%. The number of routes, total driving time and distance of both normal and under-attack traces are shown in Table 2.4.

We adopt 11 features, as shown in Table 2.5, to cross-check GPS locations. Intuitively, these features examine the anomaly of GPS locations by comparing them with network locations

User	Route #	Total time (h)	Total distance (km)
M7	10 vs. 59	1.6 vs. 5.1	31.8 vs. 102.4
Nexus5	30 vs. 100	3.2 vs. 7.6	67.3 vs. 163.4
Nexus5X	9 vs. 58	1.3 vs. 4.5	25.3 vs. 90.4
S6	36 vs. 60	2.2 vs. 3.5	37.6 vs. 66.6

Table 2.4: Normal vs. under-attack sensor traces.

Table 2.5: The meaning of the feature vector.

Feature	Meaning
x_1	Indication of the existence (0 or 1) of the network (WiFi/cellular tower) location
x_2	Distance between network location and GPS location
x_3	Road curve and turn angles indicated by GPS
x_4	Road curve and turn angles indicated by gyroscope
x_5	Headings indicated by GPS
x_6	Headings indicated by magnetometer
x_7	Indication of GPS location jump with trigger condition *GPS speed > speed limit + 50 mph (22.352 m/s)*
x_8	Displacements indicated by GPS
x_9	Displacements indicated by accelerometer ($vt + \frac{1}{2}at^2$)
x_{10}	Speed limit of the road indicated by GPS
x_{11}	Instant speed indicated by GPS

and IMU data (*e.g.*, physical distance, road curve and turn angles, headings, location jumps, etc.). As this is a typical multivariate time series classification problem, the sensor data is sliced into chunks and averaged along the 11 feature dimensions based on a certain size time window. Under-attack and normal samples are labeled as 1 and 0, respectively. An SVM classifier is used because of its generally good performance on small dataset. For each user, we use sequential feature selection and tune SVM parameters (*e.g.*, kernel type, box constraint, etc.) on validation dataset and then apply them to the test dataset. The precision and recall are shown in Table 2.6 for different users. As you can see, the SVM classifier generally suffers from low precision, which indicates a high false alarm rate that renders the defense system unusable.

We have two insights. First, road-related features (*e.g.*, road curvature, heading, and speed limit) are not robust to the stealthy attack. This is because the stealthy attack algorithm essentially searches for ghost routes that are extremely similar to the real-driving routes in terms of the shape and speed limits. Second, contrary to what we expected, network location did not help much on the classification. In our driving dataset, the network location updates are very sparse on the road due to limited number of WiFi hot spots. In addition, cellular location has a large error margin up to a few kilometers, much higher than the location

User	M7	Nexus5	Nexus5X	S6
Precision	54.31%	84.47%	100%	56.84%
Recall	100%	82.73%	58.82%	93.10%

Table 2.6: Precision and recall for different users.

drift constraint (*e.g.*, 400m in our attack). Note that even if dense WiFi/cellular network presents along the victim's driving route, the attacker is also able to jam or spoof the signals [44, 45]. In conclusion, we believe that although sensor data and network location deliver a certain level of mitigation, reliable countermeasure (*e.g.*, integrating CV techniques) for stealthy location manipulation attack is still challenging and worth further investigation.

2.8.3 Computer Vision Based Defense

As mobile devices and self-driving cars are equipped with embedded cameras, it opens up opportunities for image-based localization to verify GPS locations. We conducted a preliminary experiment to confirm its feasibility.

Geo-Tagged Reference Image Database. In image-based localization problem, a query image has to be matched against a database of reference images with geo-tags. It is fairly important to build a reliable reference image database. Google Street View is a very good example of such a database, which are fine-grained spherical 360 degree view with a distance of about 12 meters between locations. All these panorama images have a decent resolution and an accurate GPS tag. Moreover, they have covered almost all main streets and roads in the US as well as a number of other countries. Additionally, Google Street View images can be freely downloaded via the publicly available API (Application Program Interface).

We use a crawler to automatically download the images for a specific area from Google Street View website. First of all, for a specific region, all coordinates are identified where

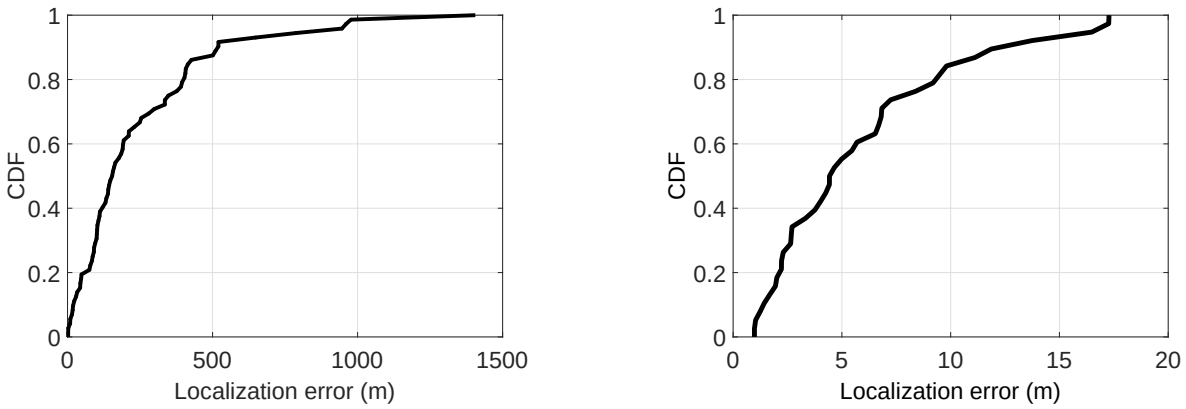
Google Street View panoramic images exist. Then, URL requests are sent with relevant parameters, such as latitude, longitude, pitch, yaw, heading, FOV (field of view) etc. to obtain corresponding images. The parameters are set to make the images that are similar to the view point of an in-vehicle driver. For every location, we cut each panoramic image to four rectilinear images with 90 degree separation by changing heading parameter. Finally, we tag and store every image with [latitude, longitude, heading].

Localization Using Image Feature Matching. We first explored image feature matching techniques, which are the most commonly used methods in past image-based localization methods [50, 56]. However, we discovered that these techniques are highly inaccurate for our target applications. We employ the method proposed by [50], which is based on nearest-neighbor tree search with pruning and smoothing process. First of all, SIFT (Scale Invariant Feature Transform) [57] features are extracted for all interest points detected in the reference images and organized into a tree using FLANN [58], which is a library for performing fast approximate nearest neighbor searches in high dimensional spaces. Then, for every query image, we compute its SIFT features and search for their nearest neighbors. We collect the votes for each reference image based on the number of retrieved nearest-neighbor vectors that belong to it. We prune the matches to filter out noisy ones and smooth the votes with a Gaussian kernel, which makes the vote peak that corresponds to the correct location more distinct.

We drove along streets in Blacksburg, VA with an iPhone 5 mounted on the windshield of the vehicle. The smartphone automatically took pictures of street view periodically, which generated a total number of 72 query images. GPS coordinates extracted from the EXIF (Exchangeable image file format) data of the images were used as the ground truth. The reference image database covers the entire map with 4028 Google Street View images for 1007 reference locations (each location has four side views). The localization performance

is shown in Figure 2.9a. As we can see, the median localization error is around 100 meters and the maximum localization error is around 1400 meters. Obviously, the feature-matching based method did not yield reliable location estimation. In order to improve the localization accuracy, we tried several methods including tuning parameters in the algorithm, changing smartphone models and camera orientation, even switching to other features like SURF (Speeded Up Robust Features) [59]. However, only trivial improvement could be made. With inspection and analysis on the returned location estimations and reference images, we found some intrinsic limitations of this algorithm. The first one is non-informative query images, which only contains trees, roads, traffic, sky etc. Most of interest points in these images are noisy and give misleading information for image matching. The second one is the difference of viewpoint, illumination and occlusion etc. between query images and Google Street View images. The localization results become poor if the difference is beyond the tolerance of SIFT. The third one is the repetitive patterns in our testing area. For example, many buildings are made of red bricks with the same pattern, which will divert the matching process for a query image that mainly contains such red brick building. In order to overcome such limitations, we also calculated the Confidence of Localization [50] to measure the reliability for each location estimation, but failed to find a strong correlation between them.

Localization Using Text Visual Landmarks. By looking into the images captured in our driving experiment, we notice that there is plenty of textual information appearing in shop signs, road signs, billboards and building walls along the road. This information can be leveraged as text visual landmarks for localization. For example, the device takes a picture of the street view and extract a text string "Blacksburg Taphouse". Then, it is compared with the pre-downloaded text visual landmarks with coordinates. Finally, a match is found and the coordinates of "Blacksburg Taphouse" are returned as the location estimation result.



(a) Image feature matching (b) Text visual landmark

Figure 2.9: (a) and (b) show the CDF of localization error for image feature matching method and text visual landmark method, respectively.

Textual information extraction from images can be achieved by optical character recognition (OCR) technique. OCR converts images of typewritten or printed text into machine-encoded text, which can be electronically edited, searched and stored compactly. There are many off-the-shelf OCR software for mobile platforms, such as Google Goggles [60]. After textual information extraction, we search the input string against the pre-stored text visual landmarks. If a match exists, the true location can be recovered as the coordinates of the matching text visual landmark. We used the same experimental settings as before. Figure 2.9b plots the results for localization using text visual landmarks. As we can see, for all query images with textual information that can be extracted by the OCR software, the median and maximum localization error are around 5m and 17m, which is sufficient for verifying GPS locations.

2.9 Discussions

Study Limitations. In this work, we optimize the GPS spoofing attack to be stealthy,

which has to compromise on other factors. First, the effectiveness of our attack will be decreased in suburb or rural area with sparse road structures. However, given that 54% of the world's population lives in urban areas [61], the attack can potentially impact many people. Second, the attack does not work on all users. We target users who travel in unfamiliar area since those users are more likely to rely on the GPS for navigation. We also argue that the increasingly popular auto-pilot systems would weaken the human-level checking in the long run.

Our user study has several limitations. First, to simulate traveling in an unfamiliar area, we choose a European city. It is possible that Hungarian street names are less understandable to Chinese/American. However, even in the US, many streets have Spanish street names. Second, due to the length and the depth of the user study, the study cannot reach a massive scale. There are biases in our user population (*e.g.*, people with a Computer Science background). We argue that the general population can be more susceptible compared to tech-savvy users. Third, our study only tested on one route, and the route does not contain wrong-ways or loops. In practice, once users enter the wrong way, they may recognize the attack (but already in danger).

2.10 Related Work

GPS spoofing attack was first systematically discussed in [62]. To date, researchers and hackers have successfully spoofed GPS devices in moving trucks [63], ships [13], drones [12] and mobile platforms [15, 16] using off-the-shelf GPS signal simulator [63] or software-defined radios [13, 12, 15, 16]. Humphreys *et al.* have demonstrated seamless GPS takeover on a moving yacht with a portable receiver-spoofer [14]. Later, an attachable miniature version one called "limpet spoofer" was proposed in [64]. Similar technical concepts were

also used in [19, 65] to develop spoofing devices. In [11], authors provided in-depth analysis and summarized requirements for seamless GPS takeover. However, above works focus on basic signal spoofing, making them unlike to succeed in road navigation scenarios.

Recently, a number of *privacy* attacks have been proposed in road navigation scenarios to infer user movements [66]. Narain *et al.* proposed a route matching algorithm to infer user movement traces based on motion sensor data [23]. Our work differs from them in terms of the attack goals and methods. Our goal is to stealthily manipulate/control the victim's navigation system by supplying fake inputs (*i.e.* GPS signals) at the right time. [4] preliminarily formulated the route spoofing problem. Compared to [4], we have made significant contributions by proposing new attack algorithms (*e.g.*, iterative attack, targeted diverting attack), and more importantly conducting real-world driving tests and user studies to validate the feasibility.

GPS spoofing belongs to the broad category of sensor manipulation. Researchers have examined attacks on other sensors such as camera, fingerprint sensor, medical infusion pump, analog sensors, and MEMS sensors [67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78]. Some of the attacks specifically target (autonomous) vehicles to disrupt their ultra-sonic sensor, millimeter-wave radar, LIDAR, and wheel speed sensor [79, 80, 81]. The unique contribution of our work is to demonstrate the feasibility of (GPS) sensor manipulation with both physical constraints (road networks) and human in the loop.

2.11 Chapter Summary

In this work, we explored the feasibility of real-time stealthy GPS spoofing attacks targeting road navigation systems. Physical measurements, taxi-trace evaluations, and human-in-the-loop user study results all confirmed high attack effectiveness and efficiency. We hope that

the results can motivate practical defense mechanisms to protect the massive GPS users and GPS-enabled autonomous systems. We also took an initial step towards studying location verification based on sensors including inertial, network, and camera sensors.

Chapter 3

Location Spoofing Attack and Its Countermeasures in Database-Driven Cognitive Radio Networks

In this chapter, we study GPS spoofing attack with frequency-agile radio and its countermeasures in database-driven cognitive radio networks.

3.1 Challenges and Contributions

Today's explosion of data communication needs is stretching the capacity limit of wireless networks. The principal limiting factor for the capacity of wireless networks is spectrum. Military communications, broadcast TV, WiFi, cellular systems and many such applications all compete for spectrum. Currently, the available spectrum is licensed to these different applications. However, some applications, like cellular systems, have grown much faster than others, such as broadcast radio and broadcast TV. This leads to overcrowding in some spec-

trum bands and underutilization of other spectrum bands. Cognitive radio networks (CRNs) based on dynamic spectrum access help to ease this imbalance in spectrum utilization. In CRNs, two kinds of users are defined - primary users (PUs) and secondary users (SUs). PUs always have the full access to the spectrum whenever they need it. SUs are permitted to use the spectrum only if they do not interfere with the PUs.

Two potential applications are in TV White Spaces (TVWS) and 3550-3650 MHz band (3.5 GHz Band). TVWS refers to the unused TV channels in any location. In November 2008, FCC issued a report that specifies the requirements for SUs to operate in licensed TV bands [82]. According to the requirements, a trusted geolocation database will be used to assign spectrum to SUs so that they will impose no interference to licensed PUs. In December 2012, FCC proposed a new Citizens Broadband Service in the 3.5 GHz Band [83]. The Citizens Broadband Service incorporates database-driven dynamic spectrum access and small cell technology[1] to enable more efficient use of radio spectrum.

With FCC laying stress on database-driven methods, it is imperative to examine the security threats of such methods. One critical security loophole is that this method relies on an SU to obtain its location information from GPS (Global Positioning System). GPS has been shown to be very vulnerable to spoofing attacks that make GPS receivers lock on spoofed GPS signals and compute false locations. With the development of programmable radio platforms such as USRPs (Universal Software Radio Peripheral), it has become quite easy to build GPS signal simulators to generate spoofed GPS signals with arbitrary date, time and location. Although there are some proposed countermeasures for GPS spoofing attacks, none of them have been implemented in commercial GPS receivers yet.

Since database-driven CRNs depend on GPS that is vulnerable to signal spoofing attacks, the

[1]Small cells are low-powered wireless base stations (also SUs) intended to cover small indoor or outdoor areas.

focus of this work is to understand the impact of GPS spoofing attacks on database-driven CRNs and explore several methods to defend against such attacks. Our study includes CRNs in both TV bands and 3.5 GHz band. We formulate various attack models and identify metrics for studying the impact of such attacks. Through simulation, we demonstrate that GPS spoofing attacks can not only create denial-of-service attacks, but also cause harmful interference to PUs. To detect and defend against GPS spoofing attacks, we propose three kinds of schemes: centralized detection scheme (CDS), environmental-radio-based location verification (ELV) and peer location verification (PLV). Moreover, we comprehensively analyze the effectiveness and limitations of the proposed countermeasures. In order to evaluate the proposed schemes, we practically implement the ELV and analyze the PLV by simulations. Experimental and simulation results show that the proposed countermeasures can effectively defend against GPS spoofing attacks on database-driven CRNs and mitigate each other's limitations if these three countermeasures are combined.

The contributions of this work are summarized as follows:

- To the best of our knowledge, this is the first work to examine the impact of GPS spoofing attacks in dynamic spectrum access, which used to be two unrelated topics.
- We formulate different attack models and analyze possible damage of such attacks. Based on the knowledge that a single attacker can easily spoof a group of SUs to an arbitrary location, we define random and optimal attack model in TV band CRNs. Simulation results show that even a simple random attack can cause PU interference in extremely sparse network with only 100 SUs in a $16km \times 16km$ cell. Additionally, we discuss attack models in 3.5 GHz CRNs, which can result in serious interference between SUs and critical DoD radar system.
- We propose various solutions for defending against GPS spoofing attacks. By thor-

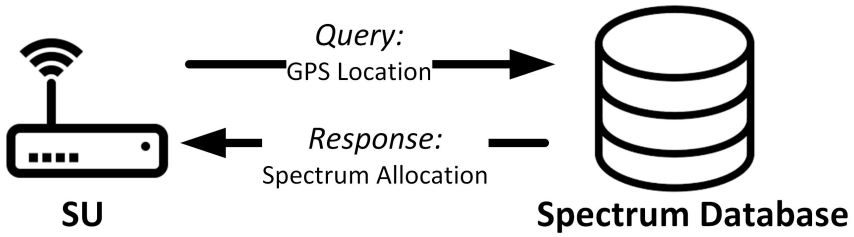


Figure 3.1: Overview of database-driven CRNs.

oughly discussing the effectiveness and limitations of each scheme, we demonstrate that the proposed countermeasures can significantly mitigate the threat of GPS spoofing attacks if a hybrid approach is taken.

- We evaluate the countermeasures by implementation, simulation and mathematical analysis. We implement the ELV by using a commercial spectrum analyzer to collect data from real world. The experiment results show that the ELV can effectively thwart GPS spoofing attacks. Besides, we present an analytical model for the PLV and use spectral analysis to derive an upper bound for convergence time. Finally, we evaluate the PLV by extensive simulations. Simulation results show that our mechanism can achieve nearly 0 false negative and false positive in most cases.

3.2 Background and Threat Model

In this section, we first present an overview of database-driven CRNs, and then introduce threat models for database-driven CRNs operating in TV band and 3.5 GHz respectively.

3.2.1 Overview of Database-Driven Cognitive Radio Networks

In cognitive radio networks, there exists primary users and secondary users. On the one hand, primary users have privileges of accessing dedicated spectrum all the time, such as military radars, satellites, and TV towers, etc.. On the other hand, secondary users can only opportunistically access available spectrum without interfering with primary users. In database-driven CRNs, a dedicated spectrum management server is in charge of coordinating PUs and SUs to achieve spectrum protection of PUs and coexistence of SUs. More specifically, as shown in Figure 3.1, an SU queries the server for available spectrum with its GPS location. Then, the server computes spectrum allocation and responds the SU with available channel list (e.g., TV white space CRNs) or a binary activation decision (e.g., 3.5 GHz CRNs). This unique mechanism causes a serious security loophole, which can be utilized by GPS spoofing attackers. For example, the attacker is able to manipulate SUs' spectrum allocation by setting their GPS locations. In what follows, we formulate attack model and present possible damage of GPS spoofing attacks in database-driven TV band CRNs and 3.5 GHz CRNs respectively, which are typical examples of different types of spectrum allocations (available channel list vs. binary activation decision).

3.2.2 Threat Model for Database-Driven TV Band CRNs

Attacker's Capability. We consider a single attacker that poses a GPS spoofing device setup at a fixed location (e.g., on a roof). As shown in Figure 3.2, with an omni-directional antenna, the attacker is able to spoof all SUs within his transmission range to an arbitrary location, say L'. For demonstration purpose, we assume the spoofing range to be 1 km. According to the report of the real spoofing incident in Moscow, Russia, the spoofing range reached around 3km. Therefore, a 1 km range is very conservative for any spoofing devices,

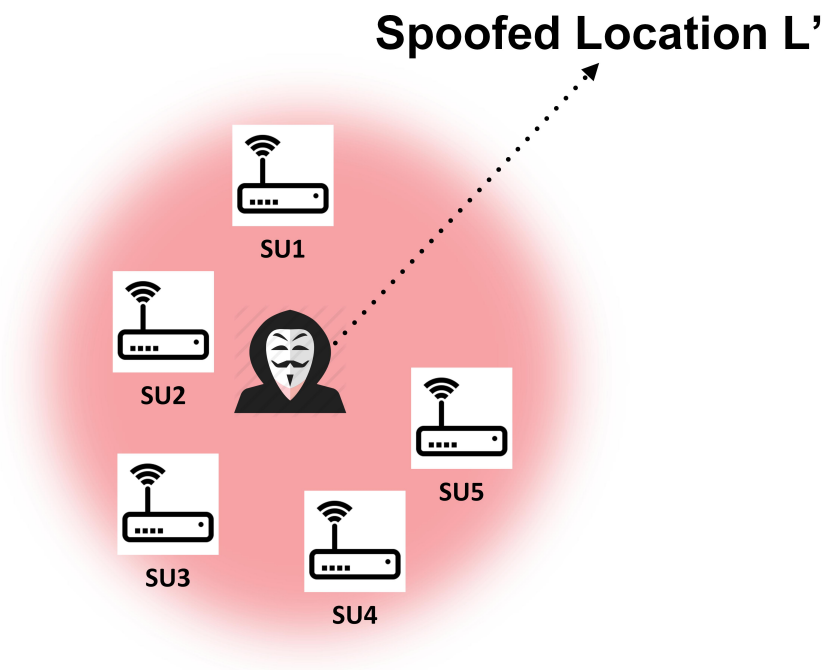


Figure 3.2: Attacker's capability.

which allows us to not impose any heavy restrictions on the attacker's capabilities.

Random and Optimal Attacks. A random attack is launched by an attacker that knows nothing about the spectrum database. The attacker just simply spoofs all SUs in his transmission range to a random location in the cell. We define an attack as optimal if the attacker has access to the database and knows complete geolocation information of all registered SUs. Based on that knowledge, he runs the same spectrum allocation algorithm on his own computer and uses brutal force algorithm to traverse all grids in the cell with different available channels to find the optimal solution, i.e. the spoofed location that maximizes the impact of his attack.

3.2.3 Threat Model for Database-Driven 3.5 GHz CRNs

Geographic Exclusion Zones. NTIA's Fast Track Report collects the information of critical DoD radars that operate from sea, land and airborne positions in or near 3.5 GHz band [84]. In order to prevent the potential interference between these radars and commercial WiMAX broadband technology, it determines "geographic exclusion zones" by calculating the separation distances between radar systems and a prospective outdoor WiMAX system. As illustrated in Figure 3.3, the geographic exclusion zones are imposed along the East, West, and Gulf Coasts. Exclusion zones are developed not only to protect incumbent DoD radar system but also to prevent interference from the high-powered radar operations to federal uses. Based on the exclusion zones in the Fast Track Report, FCC proposes similar geographic exclusion zones that considers low-powered small cell deployment in 3.5 GHz. In database-driven 3.5 GHz CRNs, low-powered wireless base stations are required to submit their accurate locations to query the database before operating. By checking the location with geographic exclusion zones, the database returns a binary decision that decides to activate the base station or not.

PU Interference and Denial-of-Service Attacks. Consider an attacker with the same capability as presented in Section 3.2.2, he is able to spoof a group of small cell base stations to an arbitrary location, which generally performs two kinds of attacks. One is spoofing the base stations from places inside the exclusion zones to a location outside, which causes all these base stations to interfere with critical DoD radars. The other attack causes SUs outside of exclusion zones to be identified as inside and hence results in denial-of-services to these SUs.

Figure 3.3: Geographic exclusion zones in 3.5 GHZ in NTIA's Fast Track Report.

3.3 Attack Evaluation

In this section, we evaluate the impact of the proposed GPS spoofing attacks in database-driven TV band CRNs.

3.3.1 Experiment Setup

In order to evaluate the impact of GPS spoofing attacks in database-driven TV band CRNs, we need to implement a prototype of such a network. We use WhiteSpaceFinder [85], which is a database that uses Longley Rice model with terrain data along with TV-tower information to predict the availability of white spaces at any location. We consider a single cell area with 16km-radius in Blacksburg, VA region and set the resolution of location queries in the area to $100m \times 100m$. A certain number of SUs are assumed to be randomly scattered around the area.

With the objective of maximizing spectrum utilization, we use round-robin scheduling and list-coloring based greedy algorithm proposed by Wang and Liu in [86] to assign available TV channels to SUs. In each time slot, we first sort all channels in ascending order of their geographic availability. A channel's geographic availability is defined as the number of grids where the channel is available. Then, we assign these channels to SUs following the sorted sequence. For each channel, there are a bunch of candidate SUs that can use the channel at their locations. The SU that has waited the longest for a channel will be assigned the channel. When a tie exists, the SU whose location has less available channels will be picked. We do not involve link degrees in the spectrum allocation algorithm, because we assume that there is no spatial reuse of channels in a single cell.

3.3.2 Attack Results

Random Attack. We first evaluate how a random attack may cause PU interference in TV band CRNs with different SU densities. We capture PU interference by the number of SUs who interfere with PUs in the simulated time. We run 50 runs of simulations under 10 different SU densities (i.e. 5 simulations for each density). The number of SUs in the cell increase from 100 to 1000 with a step of 100. Figure 3.4 shows the largest number of SUs interfering with PU for each SU density level. We can observe that as the SU density increases, PU interference becomes more serious. Also, even in an extremely sparse network with only 100 SUs in the 16km-radius range, we still observe PU interference.

We also examine the probability distribution of PU interference among 51 TV channels. As shown in Figure 3.5, most of the PU interference occur on TV channels 29, 33 and 21. This is due to the large variation in the availability of these 3 channels over the simulated geographic area. For a channel whose geographic availability varies a lot, it is more likely

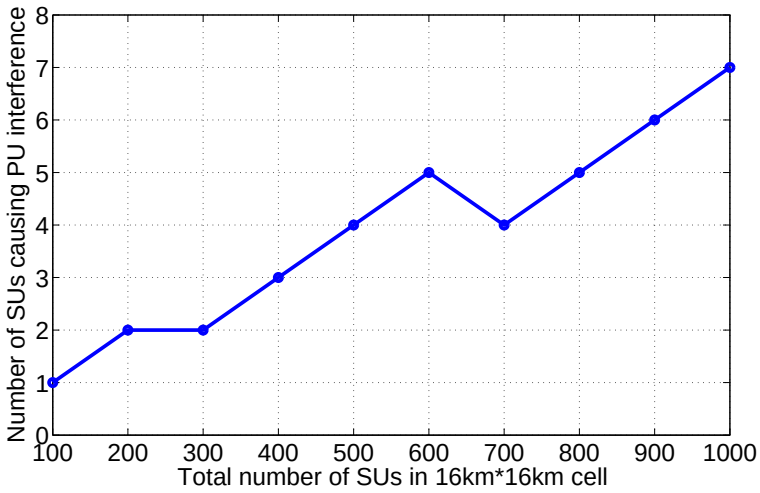


Figure 3.4: Variation in severity of PU interference with SU density.

that location spoofing attacks can cause interference.

Optimal Attack. In order to analyze the damage caused by the optimal attack, we run 30 individual simulations. In each case, we launch a random and optimal attack respectively. Figure 3.6 compares the performance of these two kinds of attacks. We average the number of SUs who interfere with PUs over simulated time and plot the cumulative distribution function (CDF). As we can see, the performance of the optimal attack is significantly better than the random attack. The optimal attack serves as the upper bound on the damage that a location spoofing attack can achieve.

3.4 Attack Detection and Countermeasures

In this section, we propose different solutions to detect and defend against GPS spoofing attacks, and analyze their effectiveness and limitations respectively. The defense mechanisms are designed based on two key insights of database-driven CRNs: (1) They are essentially

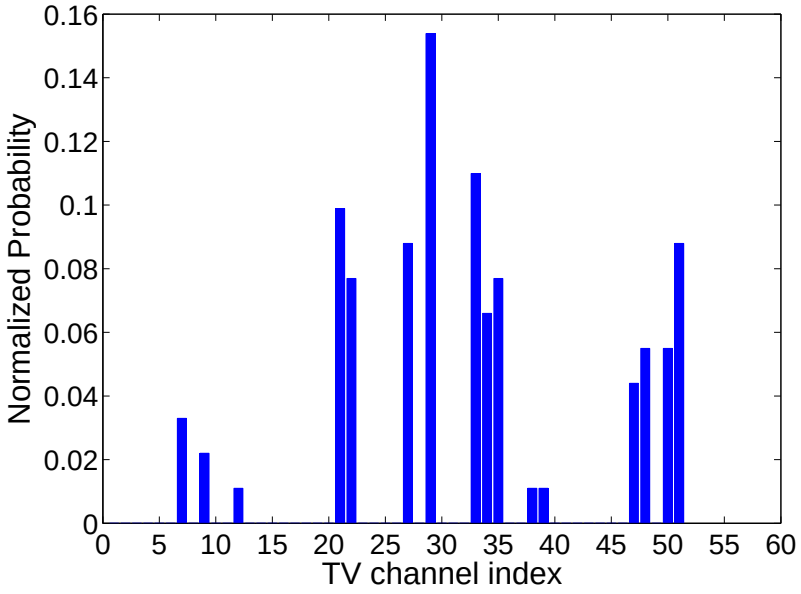


Figure 3.5: There are 21 available channels in total. In a 1000-SU network, 16 of them are interfered with at least once in 50 simulations.

centralized systems. (2) A cognitive radio usually can operate on a wide range of frequencies.

3.4.1 Centralized Detection Scheme

In database-driven CRNs, all SUs have to register their locations in the database, which can be utilized by the system to detect location spoofing attacks. For example, the system can ask every active SU to update his location information periodically. Based on the reported location and time stamp, the system maintains trace information for all SUs. By continuous running anomaly detection algorithms, once abnormal SU behaviors are detected, like a group of nearby SUs suddenly move to one place at tremendous speed, the system considers the network as under attack.

While the centralized detection scheme is fairly intuitive, it might result in false alarms. For example, thousands of people will crowd into a single stadium to watch a football game,

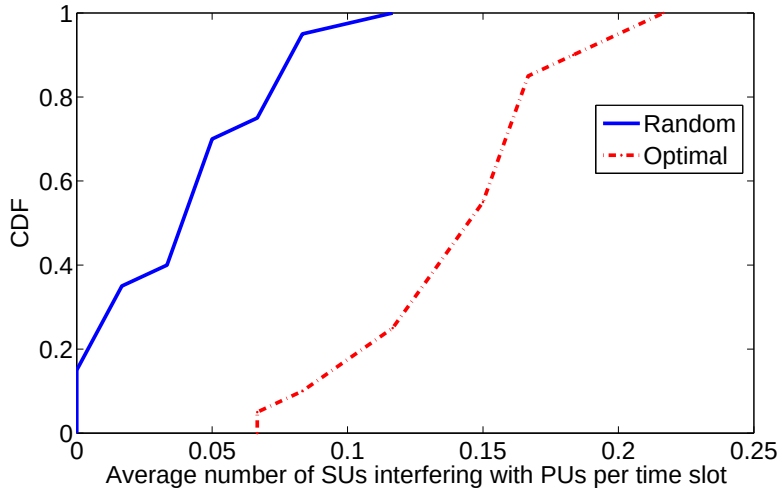


Figure 3.6: The performance of random and optimal attack in a 1260-SU network.

which can cause a lot of SUs simultaneously report the same location to the database. A centralized detection scheme that is not intelligent enough may mistake this as a result of location spoofing attacks. In addition, the database-based approach can only detect location spoofing attacks but cannot restore the normal network operation with correct location information. Finally, the tracking and analysis of every SU's location trace may impose a heavy overhead in the database system and also create potential violation to user privacy.

3.4.2 Environmental-Radio-Based Location Verification

We propose an Environmental-radio-based location verification (ELV) method to detect and thwart location spoofing attacks. Our method leverages the fact that an SU is often also a software defined radio (SDR), which can tune to different wireless systems operating at a wide range of spectrum bands. In addition, today's wireless devices often are equipped with multiple communication interfaces (e.g. WiFi, TV, FM and cellular) that work at different frequencies. Thus, in our scheme, an SU listens to the signals emitted by existing

infrastructures (i.e. WiFi access points, TV towers and FM towers). We assume that the SU pre-stores a Radio Environment Map (REM), which contains the fingerprints of ambient wireless signals at different locations [87]. By comparing the local signal fingerprints with the ones in the REM, an SU can estimate his location without relying on GPS. This estimation method can then be used to verify GPS location computation, detect GPS spoofing attacks and also be used as a backup localization mechanism when GPS is under attack.

Network-Based Location Verification

As Cellular towers and WiFi access points become increasingly prevalent, using cellular and WiFi signals for SUs to detect and defend against GPS spoofing attacks will be fairly effective, especially in urban areas. Cellular-based location usually comes with an accuracy of several kilometers, which should be sufficient for coarse-grained location verification (e.g., 3.5 GHz exclusion zone case). A WiFi access point's coverage radius is often less than 100m, an SU can verify if its GPS location is within the coverage of a specific WiFi access point and also use the WiFi access point's position as a backup location when under attack. Therefore, an SU only needs to narrow down its location to the coverage of a specific WiFi access point, whose information is stored in the REM. The accuracy of this WiFi localization scheme is more than enough to satisfy the location requirement for database-driven networks.

Methodology. In order to determine the surrounding cellular towers and WiFi access points, an SU first decodes the received GSM/IEEE 802.11 packets and records the received information like cellular tower ID, WiFi SSID and MAC address. Then, he looks up the local information in the REM to locate the corresponding cellular towers and WiFi access points. Finally, he checks if his GPS location matches the locations of surrounding cell towers and WiFi access points. If not, he considers himself as under attack and uses the backup location, i.e. cellular/WiFi positions.

Limitations. As discussed in Section 2.8, network-based localization mechanism is also vulnerable to jamming and spoofing attacks. Additionally, it is ineffective when an SU is at the blind spot of the REM, such as rural areas, where the fingerprints have not been collected yet.

TV-Signal-Based Location Verification

Recently, Rosum company has developed a chip called "Alloy" that can use TV broadcast signals to accurately localize people and objects. Rosum TV-positioning technology utilizes the time of arrival of TV broadcast signals and the locations of corresponding terrestrial broadcast television infrastructures to calculate location. With support for a variety of types of TV signals, Alloy is able to provide less than 150-meter accuracy even in the worst case [88]. Terrestrial broadcast TV signals are high-power, low-frequency signals that easily penetrate buildings and urban areas. Furthermore, TV signals have a quite large bandwidth (i.e. from 54 MHz to 890 MHz). Thus, TV signals are considered more robust to jamming and spoofing attacks.

Methodology. TV localization is a great auxiliary positioning method for small cell base stations in database-driven networks, especially in indoors or urban areas where GPS signals are unavailable. In addition, TV localization can detect and work as a backup plan against GPS spoofing attack. For example, the SUs equipped with the Alloy chips can use the location calculated by TV-positioning technology to verify the GPS location. We conservatively assume that the accuracy of GPS and TV localization are 10 m and 150 m respectively. If the error distance between these two estimated locations is 160 m or more, the SU consider himself as under attack and switch to the countermeasure, i.e. TV localization. Additionally, if the SU operates in TV bands, it neither needs extra radios or antennas for TV localization nor does it consume more energy.

Limitations. The "Alloy" chip is still under development and not commercially available in the market. Even if it is available, not every SU can afford the cost of the chip.

FM-Signal-Based Location Verification

FM signals are ubiquitous and widely-available across all environments - outdoor, indoor and urban. Therefore, we exploit FM-signal-based localization to detect and thwart GPS spoofing attacks.

Methodology. To evaluate the effectiveness of FM-based location verification, we develop a two-phase localization system using FM radio received signal strength indicator (RSSI). In the first phase (offline phase), we collect realistic FM RSSI fingerprints by a commercial spectrum analyzer and store them in the REM. In the second phase (online phase), the SU estimates his location by comparing the measured RSSI fingerprints and the pre-constructed model. Then, the SU checks if the error distance between the GPS location and the FM estimated location falls into a reasonable region. If not, a spoofing attack is detected and the SU activates the backup plan, i.e. FM radio localization. Our experiment results show that we can achieve 50-meter accuracy by 8 strongest FM channels and effectively thwart GPS spoofing attacks.

Limitations. The limitations of FM-signal-based radio localization are as the following: (1) The SU will need an extra FM antenna for signal reception. (2) Its performance highly depends on the number of local FM channels and the number of reference locations collected by the REM.

3.4.3 Peer Location Verification

Since environmental localization imposes requirements of hardware or supporting infrastructure, only limited number of SUs can use it to detect and thwart spoofing attacks. Therefore, we propose a distributed peer location verification (PLV) scheme, which propagates from a number of initial SUs to the whole network. We also describe an analytical model and study the efficiency of our mechanism by spectral analysis.

Methodology. At the very beginning of the PLV process, we assume that there are a certain number of SUs who can use the ELV to verify their GPS locations and receive location authentications. Hence, they become initial anchor nodes. Then, each anchor node transmits an r-radius beacon signal containing his position to surrounding SUs with probability β. Anchors transmit beacons in this way to avoid collisions among neighboring anchors. When an unverified SU hears a beacon signal from any anchor, he checks if his GPS location is within radius r of the anchor's location. If so, that SU trusts his GPS location and becomes a new anchor. Otherwise, the SU will infer that he is under GPS spoofing attack and remain silent. As this location verification propagates through the whole area, an equilibrium is achieved wherein no further nodes can be verified. The verification process ends at this point. If any SU fails the location verification, he indicates that he is the victim of a spoofing attack. The victim SUs can then counter the attack by estimating their locations through the average of the coordinates of surrounding anchor nodes.

Limitations. The PLV scheme also has its limitations. On the one hand, when SU distribution is sparse, it is possible that some victim SUs fail to hear any anchor node, *i.e.*, they are isolated. In such a case, they cannot positively identify whether they are under attack. This can lead to missed detection (*i.e.* false negative) of GPS spoofing attack. On the other hand, it is possible that some malicious SUs claim themselves as anchor nodes but

transmit beacon messages containing false location information to confuse other SUs. These malicious SUs can cause false alarms (*i.e.* false positive) of GPS spoofing attacks. We will discuss these two cases further in the evaluation section (Section 3.5).

Analytical Model. We propose an analytical model to study the convergence speed of PLV under no-attack condition, which is similar to virus spreading model in random geometric networks [89]. We assume that n SUs, $S_n = \{s_1, s_2, ..., s_n\}$, are located at random positions denoted as $L_n = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, where (x_i, y_i) are uniformly distributed in a 16×16 km^2 area. Each anchor has a beacon transmission range of r. We call two SUs, $s_i, s_j \in S_n$, correlated if and only if $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r$. We denote this random geometric graph by $G(L_n; r)$.

Our model divides time into discrete slots. During each time slot, an anchor node tries to verify its neighbors by transmitting a beacon signal with probability β. We denote the probability that an SU i is verified at time t as $p_{i,t}$. We assume that an SU i is verified at time t if

- i was already verified before t.
- i was not verified before t, but receives beacon signals from neighboring anchors and gets verified at t.

Hence, we define the probability of an SU i getting verified at time t to be

$$\begin{aligned} p_{i,t} &= p_{i,t-1} + (1 - p_{i,t-1})(1 - \prod_{j \in Neighbors\ of\ i} (1 - \beta p_{j,t-1})) \\ &= 1 - (1 - p_{i,t-1}) \prod_{j \in Neighbors\ of\ i} (1 - \beta p_{j,t-1}), \end{aligned} \tag{3.1}$$

where $i = 1, 2, ..., n$. Given a network topology and specific β value, we can numerically solve equation (3.1) and obtain time evolution of total number of verified SUs $\mathrm{N}_t = \sum_{i=1}^{n} p_{i,t}$.

Spectral Analysis of Convergence Time. The efficiency of the PLV mechanism highly depends on the convergence time of location verification process. Therefore, we use spectral analysis on our model to find an upper bound of convergence time. Assuming β is much smaller than 1, we can approximate (3.1) as:

$$p_{i,t} = p_{i,t-1} + \beta \sum_{j} p_{j,t-1}. \tag{3.2}$$

This uses the approximation $(1-\varepsilon)(1-\mu) \approx 1-\varepsilon-\mu$, where $\varepsilon \ll 1, \mu \ll 1$.

Using a column vector $\mathbf{P_t} = (p_{1,t}, p_{2,t}, ..., p_{n,t})^T$ to convert equation (3.2) to matrix form, we have

$$\mathbf{P_t} = (\mathbf{I} + \beta\mathbf{A})\mathbf{P_{t-1}} = \mathbf{B}\mathbf{P_{t-1}} = \mathbf{B}^t\mathbf{P}_0, \tag{3.3}$$

where $\mathbf{B} = (\mathbf{I} + \beta\mathbf{A})$ and $\mathbf{A}$ is the adjacency matrix of G.

$\mathbf{V}_{i,A}$ is the eigenvector of $\mathbf{A}$ corresponding to eigenvalue $\lambda_{i,A}$. By definition, we have $\mathbf{A}\mathbf{V}_{i,A} = \lambda_{i,A}\mathbf{V}_{i,A}$. So,

$$\mathbf{B}\mathbf{V}_{i,A} = (\mathbf{I} + \beta\mathbf{A})\mathbf{V}_{i,A} = \mathbf{V}_{i,A} + \beta\lambda_{i,A}\mathbf{V}_{i,A} = (1 + \beta\lambda_{i,A})\mathbf{V}_{i,A}.$$

Hence, $\mathbf{V}_{i,A}$ is also the eigenvector of $\mathbf{B}$ but corresponding to eigenvalue $\lambda_{i,B} = 1 + \beta\lambda_{i,A}$. Using spectral decomposition on real symmetric matrix $\mathbf{B}$, we have

$$\mathbf{B}^t = \sum_{i} {\lambda^t}_{i,B}\mathbf{V}_{i,B}{\mathbf{V}_{i,B}}^T. \tag{3.4}$$

Sorting the eigenvalues in non-increasing order such that $\lambda_{1,A} \geq \lambda_{2,A} \geq ... \geq \lambda_{n,A}$ and $\lambda_{1,B} \geq \lambda_{2,B} \geq ... \geq \lambda_{n,B}$. Substituting equation (3.4) into equation (3.3), we have

$$\mathbf{P_t} = \sum_i {\lambda^t}_{i,B} \mathbf{V}_{i,B} {\mathbf{V}_{i,B}}^T \mathbf{P}_0 = \sum_i {\lambda^t}_{i,B} \mathbf{C_i}, \tag{3.5}$$

where $\mathbf{C}_i$ are constant column vectors. Thus, time evolution of total number of verified SUs is

$$\mathrm{N}_t = \sum_{i=1}^{n} p_{i,t} = \sum_{i=1}^{n} ({\lambda^t}_{i,B} \sum_{j=1}^{n} c_{ij}),$$

where c_{ij} is the jth element of the constant matrix $\mathbf{C}_i$.

Furthermore, we can say that

$$\mathrm{N}_t > {\lambda^t}_{1,B} \sum_{j=1}^{n} c_{1j} \Rightarrow t < \log_{(1+\beta\lambda_{1,A})} \frac{N_t}{\sum_{j=1}^{n} c_{1j}}. \tag{3.6}$$

In equation (3.6), we see that the upper bound of convergence time is a log function with base $1 + \beta\lambda_{1,A}$ indicating that the convergence time scales very well with N_t and the system can converge fairly fast even with a large N_t.

In order to use simulation to evaluate the correctness of our theoretical analysis, we set up an initial anchor ratio $\gamma = \frac{Number\ of\ initial\ anchors}{Total\ SU\ number}$ and begin each simulation case with $\gamma \times n$ randomly chosen initial anchors. To calculate theoretical bound ${\lambda^t}_{1,B} \sum_{j=1}^{n} c_{1j}$, we obtain ${\lambda^t}_{1,B}$ and $\mathbf{V}_{i,B}$ from B and set $p_{i,0} = \gamma$ to match the initial anchor ratio with the simulation settings. Figure 3.7 plots time evolution of verified SUs in a simulation of 5000-SU network. We can see that the convergence time is indeed upper bounded by what indicates in equation (3.6). Since PLV converges very fast, we can reasonably assume that all SUs maintain a static topology in such a short time. Hence, we do not consider the model of SUs with mobility in this work.

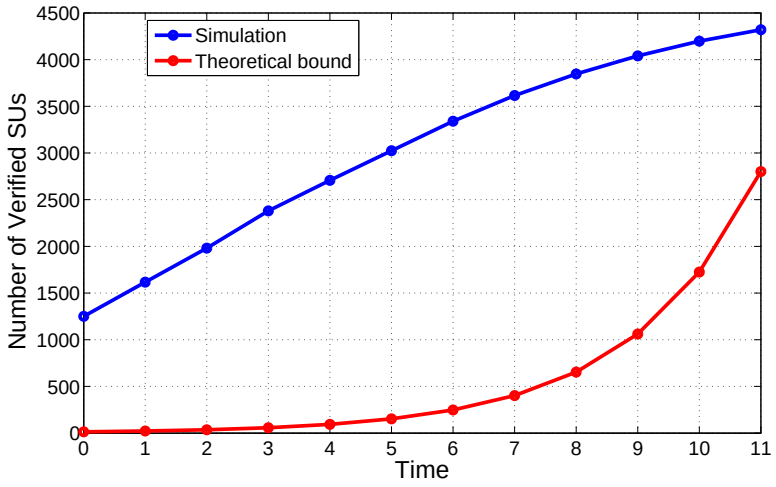


Figure 3.7: Simulation results are averaged over 30 individual runs in a 5000-SU network with $\gamma = 25\%$, $\beta = 0.03$ and $r = 500m$.

3.5 Implementation and Evaluation

In this section, we evaluate the performance of the ELV in real world and the effectiveness of the PLV by simulations.

3.5.1 Implementation of ELV

In order to evaluate the proposed ELV, we implement FM-signal-based location verification (we skip the implementation of network-based and TV-signal-based location verification. The reason is that for cellular-based location verification, it is already in widely use on smartphones. As for TV-signal-based location verification, the "Alloy" chip is not commercially available yet.). To implement the FM-signal-based localization, we drive around the Blacksburg, Virginia region to collect real RSSI fingerprints with a commercial spectrum analyzer and evaluate the performance by extensive experiments.

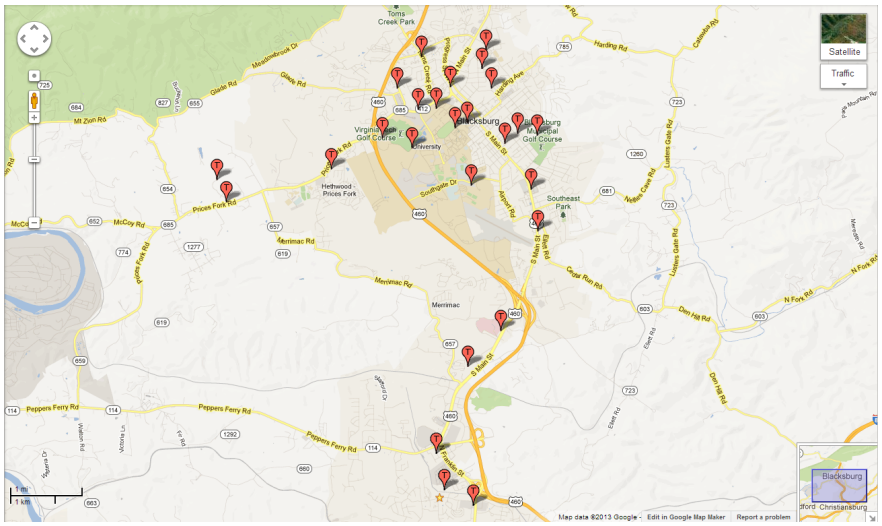


Figure 3.8: Red balloons indicate the locations of test points.

Data Collection. We perform the experiment in Blacksburg, Virginia region with an area of around 86.25 km^2. We select 26 locations shown as the red balloons in Figure 3.8. To measure real RSSI of FM channels, we use a Tektronix MDO4104-6 Mixed Domain Oscilloscope and a v-shaped "rabbit ear" FM antenna, as shown in Figure 3.9. We take measurements of 17 FM channels at each reference position and divide the measurements into two groups. The first group is called training group and the second group is called testing group. We use Gaussian regression to model the data in training group as a Gaussian distribution for each reference location. Then, we input the data in testing group to evaluate the performance of the FM-based location verification scheme.

Positioning Algorithm. The positioning algorithm we use is proposed by Fang et al. [90]. Our task is to estimate the locations from the pre-stored reference points. So we use likelihood of each reference position to calculate estimated location represented by numerical

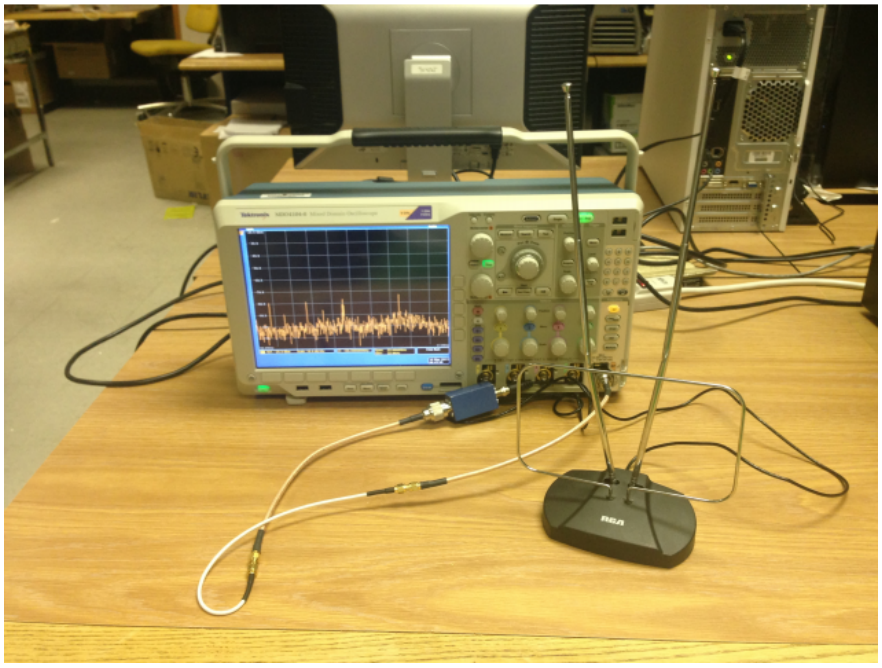


Figure 3.9: Tektronix MDO4104-6 Mixed Domain Oscilloscope and the FM antenna.

latitude and longitude. In this way, the localization problem can be formulated as follows:

$$\bar{L} = \sum_{k=1}^{n} L_k \cdot P_{norm}(\mathbf{C_k}|\mathbf{X}), \tag{3.7}$$

where

$$P_{norm}(\mathbf{C_k}|\mathbf{X}) = P(\mathbf{C_k}|\mathbf{X}) / \sum_{i=1}^{n} P(\mathbf{C}_i|\mathbf{X}). \tag{3.8}$$

$\overline{L}$ is the localization output represented by numerical latitude and longitude. L_k is the coordinate of kth reference position and n is the number of reference positions. $P_{norm}(\mathbf{C_k}|\mathbf{X})$ is the normalized likelihood that the observed RSSI vector $\mathbf{X}$ is measured in reference location $\mathbf{C_k}$.

Applying the Bayes' rule, we have

$$P(\mathbf{C}_i|\mathbf{X}) = P(\mathbf{X}|\mathbf{C}_i) \cdot P(\mathbf{C}_i)/P(\mathbf{X}), \tag{3.9}$$

where $i = 1, 2, ..., n$. In equation (3.9), the $P(\mathbf{C}_i)$ is the a priori probability of being at the reference position i. Since we have no priori knowledge of where the SU is, we assume a uniform distribution, i.e. $P(\mathbf{C}_i) = 1/n$.

Applying equation (3.9) to equation (3.8), we have

$$P_{norm}(\mathbf{C_k}|\mathbf{X}) = P(\mathbf{X}|\mathbf{C_k})/\sum_{i=1}^{n} P(\mathbf{X}|\mathbf{C}_i). \tag{3.10}$$

Since we already have the Gaussian distribution model for each reference point $\mathbf{C_k}$ and the observed RSSI vector $\mathbf{X}$, the likelihood $P(\mathbf{X}|\mathbf{C_k})$ is numerically computable.

3.5.2 Evaluation of ELV

ELV can only detect location spoofing attacks where the spoofed location is relatively far away from an SU's true location. If the distance between the spoofed location and the true location is smaller than the confidence range of ELV's localization scheme, the spoofing attack cannot be detected. To evaluate the performance of ELV, it is important to understand if these undetected attacks can impose a significant threat to CRNs.

We will focus on database-driven TV band CRNs rather than 3.5 GHz CRNs because the TV band CRNs have many channels with different availability distribution while in 3.5 GHz CRNs, it is only an on/off decision. Thus, TV band CRNs are more likely to have variations in channel availability in a small area than 3.5 GHz CRNs, such that undetected spoofing attacks have a higher chance to cause damage in TV band CRNs.

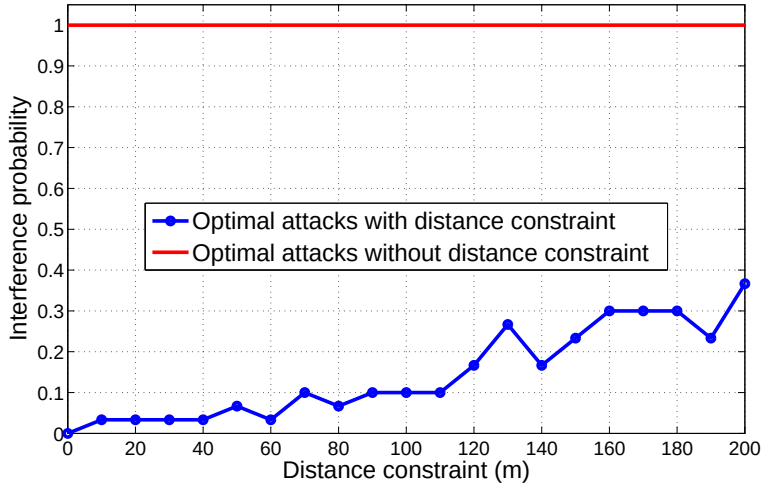


Figure 3.10: The probability is calculated by 30 individual simulations in an 84-SU network.

The undetected spoofing attack can be modeled as a location spoofing attack with a distance constraint so that the attacker cannot spoof an SU to a false location that is further than the constraint. Different ELV localization accuracy will impose different distance constraint. Our simulation settings are the same as described in Section 3.3.1. Figure 3.10 shows the probability that SU interferes with PU when an optimal spoofing attack is launched under the distance constraint. The interference probability is calculated as number of simulation runs that have PU interference over total number of simulation runs. It can be seen that the probability of interference caused by optimal attacks goes up as the distance constraint increases. Particularly, the probability is $\leq 10\%$ when constraint distance is $\leq 100m$. In comparison to the optimal attacks without distance constraint which causes PU interference with 100% probability, the PU interference threat is greatly mitigated.

Figure 3.11 shows the localization error distribution (i.e. mean plus standard deviation) under different number of FM channels. In order to obtain a better localization performance, we always select the strongest FM channels at each test position to calculate estimated location. It can be seen that the FM-signal-based localization can achieve 50m accuracy

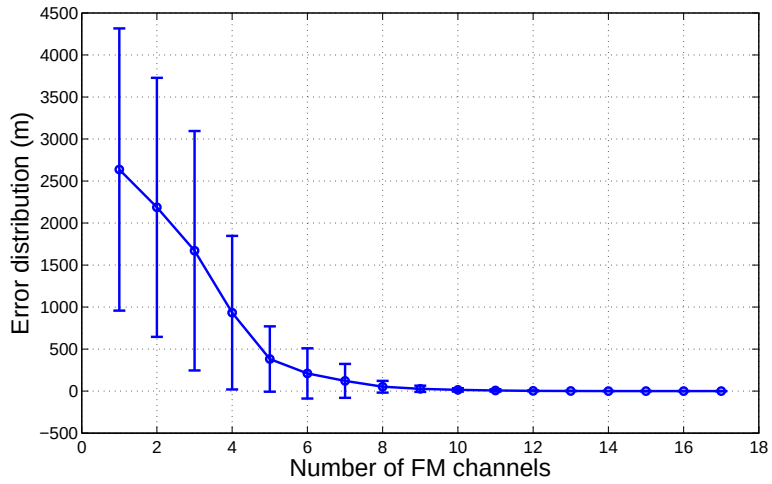


Figure 3.11: Localization performance using the strongest FM channels.

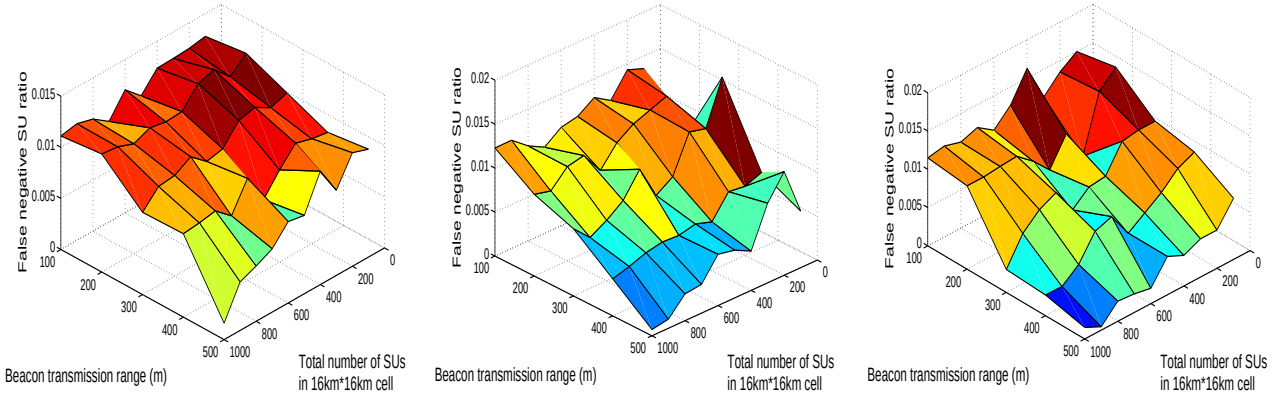
with 8 strongest FM channels. Thus, comparing with Figure 3.10, we can conclude that both WiFi and FM-based ELV can effectively limit the impact of spoofing attacks even when these attacks evade their detection. Hence, we can say our ELV mechanism can effectively countermeasure GPS spoofing attacks.

3.5.3 Evaluation of PLV

PLV scheme may experience missed detection (false negative) because of isolated SUs and false alarm (false positive) caused by malicious anchor nodes. In this section, we evaluate such situations by simulations. Each simulation plot is averaged over 100 individual runs.

Missed Detection. In Figure 3.12, we vary beacon transmission range r , SU density $n/(16 \times 16km^2)$ and initial anchor ratio γ to exam the performance in different scenarios. We define false negative SU ratio as the number of non-detected victim SUs over the total number of SUs who are not anchors.

As seen from the figure, the false negative SU ratio goes down as one of the three parameters



(a) $\gamma = 10\%$ (b) $\gamma = 30\%$ (c) $\gamma = 50\%$

Figure 3.12: False negative SU ratio varies with SU density, beacon transmission range and initial anchor ratio γ.

increases. As we state in section 3.4.3, missed detections are resulted from the isolated victim SUs, which cannot be positively identified by the system. Hence, with more initial anchors and larger beacon transmission range, the PLV can propagate more widely. It reduces the number of isolated SUs so that it is more likely to detect spoofing attacks. Additionally, as the density of SUs become larger, the graph is more connected, which also results in better detection performance. In Figure 3.12b, we can achieve $\leq 0.5\%$ false negative SU ratio with a beacon transmission range $r = 500m$ if SU density $\geq 600/(16 \times 16km^2)$. In Figure 3.12c, the initial anchor ratio increases to 50%. In order to achieve the same false negative SU ratio, we only need a $400m$ beacon transmission range for a 500-SU network. As the initial anchor ratio keeps increasing, the same false negative SU ratio can be achieved with even lower beacon transmission range for a sparser network. Therefore, we can adjust beacon transmission range according to initial anchor ratio and SU density. For example, it is more efficient to use a low beacon transmission range for urban area (high γ and SU density) and a higher beacon transmission range for suburb or rural area (low γ and SU density).

False Alarm. Malicious SUs can cause false alarms in PLV process. These malicious

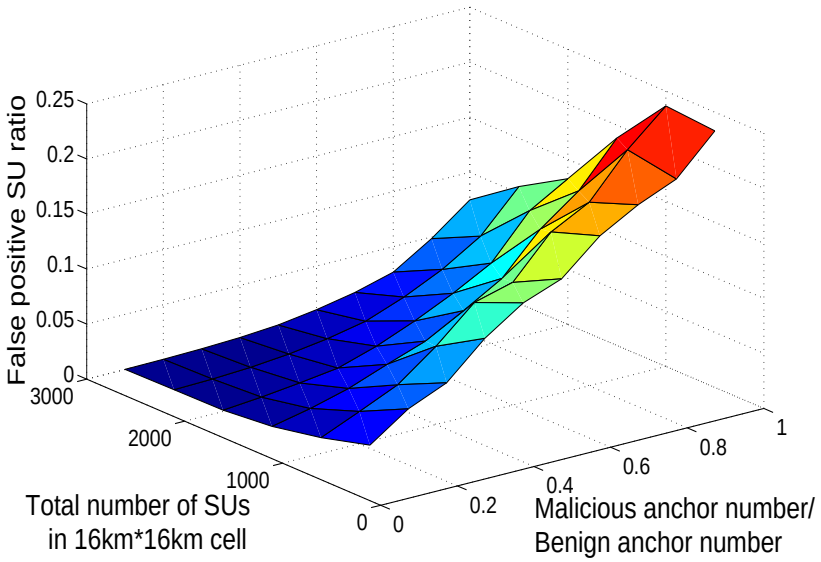


Figure 3.13: False positive SU ratio with $\gamma = 30\%$ and beacon transmission range $= 500m$.

SUs claim themselves as anchor nodes and transmit false location information to fool other SUs. To deal with such a problem, we adopt a majority voting mechanism. An unverified SU will always select the result (verified or spoofed) indicated by most of the surrounding anchor nodes. If there is a tie, the default assumption is that the SU's GPS location is verified.

We first fix the benign anchor ratio $\gamma = 30\%$. Then we vary the ratio of the number of malicious SUs over the number of benign anchor nodes and SU density to examine the performance of majority voting mechanism under different situations. We also define false positive SU ratio as the number of SUs who report false alarms over the total number of SUs who are not anchors. The malicious SUs are uniformly distributed in the $16 \times 16km^2$ area.

As shown in Figure 3.13, for a fixed malicious SU number, the false positive SU ratio decreases as the network density becomes larger. The reason is that for a sparse network, some unverified SUs can only hear one malicious anchor node and then report false alarms. The

majority voting mechanism is not effective in this case. With the SU distribution becomes denser, unverified SUs can hear more anchors, so that the majority voting becomes more reliable. For relatively dense SU situations, if the number of malicious anchor nodes is less than half of the number of benign anchor nodes, the false positive SU ratio is negligible. Our majority voting mechanism only cannot handle the case in which the network is extremely sparse and with a great portion of malicious SUs. However, it is very unlikely that a large portion of SUs are malicious in an extremely sparse network.

Discussions. The PLV mechanism requires a certain minimum initial anchor ratio and SU density to propagate, which causes an intrinsic limitation. It may have high rate of missed detection in extremely sparse SU situations with fairly low initial anchor ratio. Besides, it may also create false alarms in extremely sparse SU networks with high malicious SU ratio. However, in our experiments, we observe that attackers are less likely to launch GPS spoofing attacks on such a sparse network. The reason is that such an attack can hardly spoof enough SUs to cause PU interference. Hence, we conclude that our detection mechanism works well for relatively dense SU situations which are very likely for such a large cell.

3.6 Related Work

Our work is related to existing works in two areas: (1) GPS spoofing attacks and (2) security issues in database-driven CRNs. As the first area has been extensively discussed in Chapter 2, we focus on the research works in the second area.

The latest IETF draft discussed the potential attacks towards the querying progress between an SU and the database [91]. In such an attack, the adversary can track locations and identities of the SUs and respond the SUs with malicious spectrum information, which results

in PU interference. It also introduced Transport Layer Security (TLS) as a solution to mitigate the threats. There are also malwares that can compromise SUs to interfere with PUs [92]. Gao *et al.* [93] proposed location privacy attacks in database-driven CRNs. It pointed out that an adversary can infer an SU's location through the SU's channel usage and proposed corresponding countermeasure. Following up with that work, a series of privacy-preserving database-driven CRN frameworks have been proposed [94, 95, 96, 97, 98, 99]. To the best of our knowledge, location spoofing attacks in database-driven CRNs have not been systematically discussed in any existing work yet.

3.7 Chapter Summary

In this work, we identify GPS spoofing attacks in database-driven CRNs, which can result in interference between SUs and critical PUs and false denial-of-services to SUs. We also propose various solutions to defend against such attacks and analyze the effectiveness, efficiency and limitations of them. Through extensive experiments and simulations, we demonstrate that the proposed schemes can significantly mitigate the threat of GPS spoofing attacks in database-driven CRNs.

Chapter 4

Marinet: An Energy Harvesting Maritime Mesh Network

In this chapter, we develop a low-cost low-power white space router prototype based on frequency-agile platform. Combined with energy harvesting buoys, the white space routers can form a mesh network to provide connectivity on the ocean.

4.1 Challenges and Contributions

Broadband wireless communication technology, such as cellular and WiFi networks, has dramatically changed the modern society. This enables people to live a much more convenient life than ever. However, while people on land are taking the ubiquitous high-speed connections for granted, it is an entirely different story for people and industry on the ocean. Since cellular and WiFi connections are not extended to the ocean, they have to use expensive and/or low-speed alternative communication technologies. The existing maritime communication services can be basically summarized to three kinds. The first kind is satellite

communications. In order to use satellite communication service, a user first needs to buy a satellite terminal for more than $3000 [100]. Then, after installation, the user needs to buy a data plan for $10/MB with a speed only up to 100 Kbps [101]. In this case, there is no way for the user to smoothly browse basic web pages on Internet. The second kind is VHF radio, which only supports low-speed voice communication due to limited bandwidth (typically $<$ 38.4 Kbps [102]). The third kind is undersea fiber. Although this technology can provide high-speed connection to fixed sites on the ocean, the deployment cost is extremely high (around $33000/km [103]).

Obviously, the existing maritime communication services do not satisfy essential connectivity demand of various applications on the ocean. For example, ocean monitoring and rescue systems with UAVs (unmanned aerial vehicles) and AUVs (autonomous underwater vehicles) need to send back their real-time surveillance video and real-time sensor data. Workers on oil rigs need to search online to help resolve technical issues and make teleconference calls with technicians on land. Passengers on cruise want to be able to browse websites with lots of pictures, use mobile apps with heavy traffic demand (*e.g.*, Facebook, Instagram, etc.), and voice/video chat with family members and friends. Therefore, in all these application scenarios, low-cost high-speed connectivity is a must.

In order to provide low-cost high-speed connectivity on the ocean, we are facing three unique challenges as follows.

- Communication base stations on land are usually mounted on high places like mountains, tall buildings, and specialized communication towers. However, there are no such places on the ocean.
- While the power supply for communication infrastructures on land can be directly transmitted through power lines, such standard power supply does not exist on the

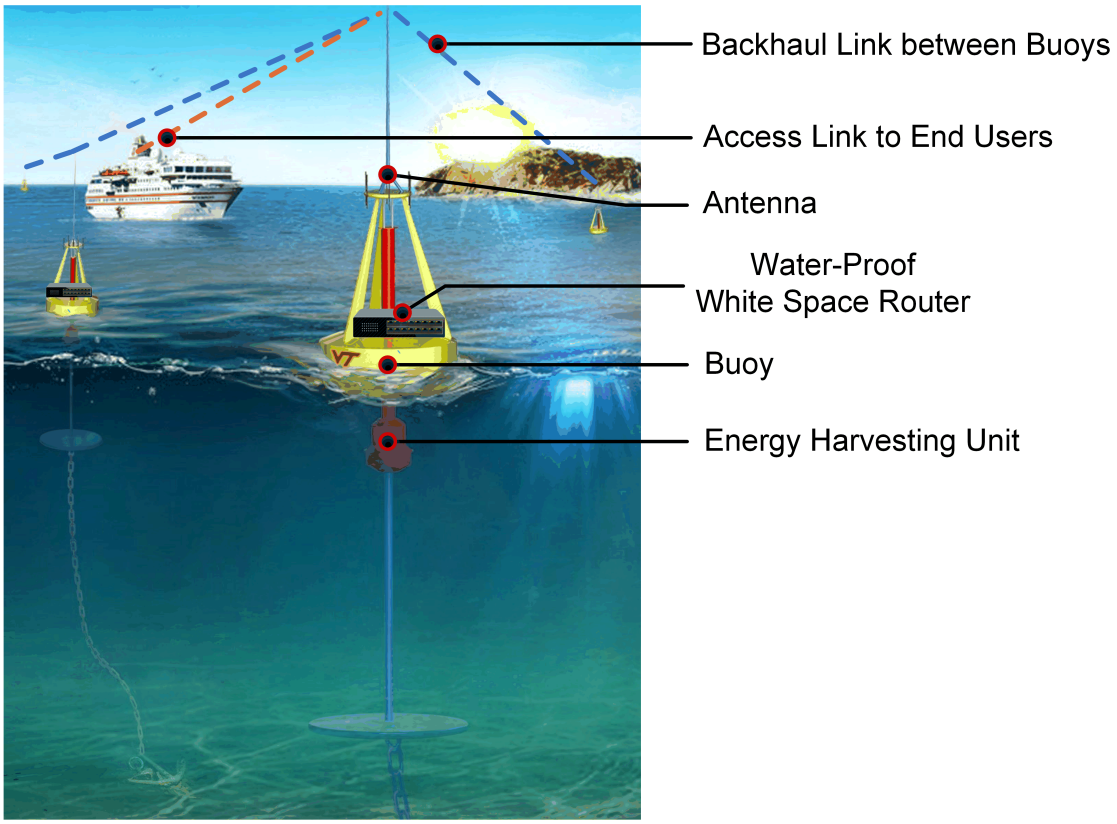


Figure 4.1: Illustration of the energy harvesting maritime mesh network.

ocean.

- On land, communication infrastructures are directly connected to the Internet via cables, but such cable connections are not available on the ocean.

To address the above challenges, we propose *Marinet*, an energy harvesting maritime mesh network. As shown in Figure 4.1, our system consists of two components, which are energy harvesting buoys and wireless mesh routers. The 1m-diameter energy harvesting buoy is compact, low-cost, and maintenance-free, which can be mass-produced and dropped into water with low deployment cost. Once in the ocean, the buoy can harvest energy from ocean wave and current, which provides energy source for wireless communication services. In this way, we solve the challenges of base station placement and power supply by using floating

base stations and energy harvested from the ocean. In the meanwhile, buoys maintain high-speed wireless communication links with neighboring buoys, which form a self-organized mesh network. In this way, the lack of cable connection is resolved by leveraging multi-hop wireless links. Therefore, combining the floating base stations, sustainable power harvested from the ocean, and multi-hop wireless links, our system can provide low-cost high-speed connectivity to various maritime applications.

4.2 Essential Components of *Marinet*

In this section, we present more details about two essential components of *Marinet*, which are energy harvesting buoy and wireless mesh router.

Energy Harvesting Buoy. The energy harvesting buoy is the cornerstone of the entire system. Traditionally, long cables are used for buoys to access land power, but this is extremely costly. With special energy harvesting units, our buoys can consistently harvest large enough amount of power to support communication services. As shown in Figure 4.1, the energy harvesting components consist of a floating buoy moving up and down with ocean waves, an energy harvesting unit in the middle, and a submerged body in certain depth of the ocean with relatively small wave amplitudes. The energy harvesting unit is composed of an ocean wave power takeoff system and an ocean current turbine. The wave power takeoff system is based on a mechanical motion rectifier that converts the bi-directional up and down motions of the floating buoy to uni-directional motions, which constantly drives the generator. The current turbine is integrated to drive the same generator from the lateral side through the ocean or tidal current. By measurement in ocean and wave tank, the buoy can continuously generate electrical power in tens or hundreds of watts from ocean wave and current (*e.g.*, 76 w–306 w for 0.5 m–1 m wave height that fall into sea state 2–3). Moreover,

compared to solar panels that are affected by light/weather condition and covering marine organisms (*e.g.*, barnacles, algae, etc.), wave and current energy is very stable. With such high and stable energy supply, the energy harvesting buoys can provide sustainable power to networking devices.

Wireless Mesh Router. The ideal wireless mesh router on the ocean should support high-speed links, have large coverage distance, and be low-cost and low-power. By analysis, we consider white space radio as a promising technology in maritime scenario for two reasons. (1) *High data rate.* Thanks to the large amount of available TV white space on the ocean, white space radio can easily provide a few to tens of Mbps data rate, which is critical to real-time applications such as voice/video chat that requires high-speed connection. (2) *Greater coverage distance.* TV white space signals can transmit in long distances because they operate at lower frequencies (*i.e.*, 470 – 698 MHz). Compared to 2.4 GHz WiFi signals, they can reach four or five times distance with the same transmit power. Tens-of-kilometer transmit range is achieved in practical white space network deployment on land [104]. With greater coverage distance, the number of nodes in the mesh network can be reduced. Therefore, a target area can be covered by less energy harvesting buoys and mesh routers, such that the manufacturing and deployment cost can be reduced. However, commercial off-the-shelf white space radios are expensive ($4000–$5000 for a base station and $1000–$2000 for a client [105]) and high-power (25 watts [106]). The high cost and power will significantly limit the scalability of the network and make the practical deployment unlikely. Moreover, since their hardware and software are proprietary, there is no way for us to customize the radios for maritime applications. Therefore, we decide to design and implement our own low-cost low-power white space router for maritime applications.

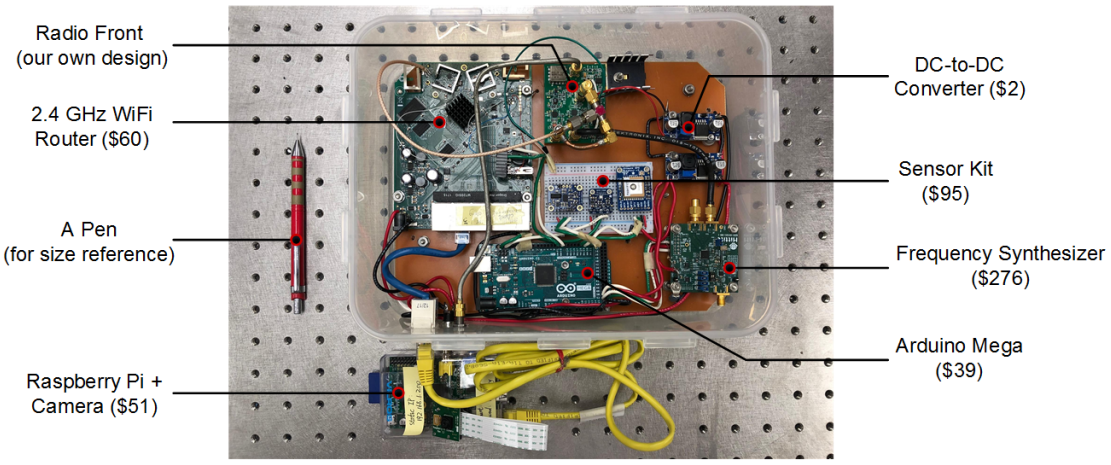


Figure 4.2: White space router prototype.

4.3 White Space Router Design and Implementation

In this section, we design and implement a low-cost low-power white space router prototype[1], which is customized for usage on the ocean.

Hardware. As shown in Figure 4.2, the design of our white space router hardware consists of six components, which are 2.4 GHz WiFi router, radio front designed on our own, Arduino Mega as an external microcontroller, frequency synthesizer, sensor kit, and Raspberry Pi with camera. For transmission, the WiFi router generates standard WiFi signals in 2.4 GHz band, which are fed into the radio front and down-converted to the target center frequency in TV band by mixing the signal generated by the frequency synthesizer. For receiving, vice versa, the radio front up-converts the incoming signal to 2.4 GHz band and sends it back to the WiFi router. The Arduino Mega receives commands from the router and controls the frequency synthesizer for dynamic channel selection in TV band. It also controls the sensors to send real-time streaming data back to the router for processing. The sensor kit includes a GPS receiver for location calculation and time synchronization, an inertial

[1]An FCC Program Experimental License is granted to our white space router.

measurement unit (accelerometer, gyroscope, and magnetometer) that calculates the real-time motion, and temperature, humidity, barometric pressure sensors for ocean environment monitoring. The Raspberry Pi with camera continuously streams surveillance video to the router via Ethernet cable. The total cost for the prototype is around $523, which is ten times cheaper compared with the commercial off-the-shelf white space routers. This low-cost white space router will significantly reduce the deployment cost of the proposed maritime mesh network. Moreover, our white space router supports high transmission power (currently set as 25 dBm) for long distance communication. By measurement, with current settings, our white space router only consumes 12 watts power in total, which is less than half of the power consumed by commercial off-the-shelf white space routers. It gives a large margin of power budget given the power harvested by the buoy (hundred-watt level). These results confirm the feasibility of our white space router for the energy harvesting mesh network application.

Software. We customize the operating system based on OpenWrt [107], which is an open-source Linux distribution for embedded devices. We successfully establish 802.11g mesh backhaul link between white space routers based on mac80211 framework embedded in Linux kernel. The operating system provides interface for flexible channel bandwidth configurations, which can appropriately fit into TV channels with 6 MHz bandwidth. For example, the channel bandwidth can be configured as 5 MHz, 10 MHz, and 20 MHz, which utilizes one or several contiguous TV channels as allowed by the white space spectrum database. It also provides low-level driver information and access to the serial interfaces (*e.g.*, UART), which facilitate the development of customized applications.

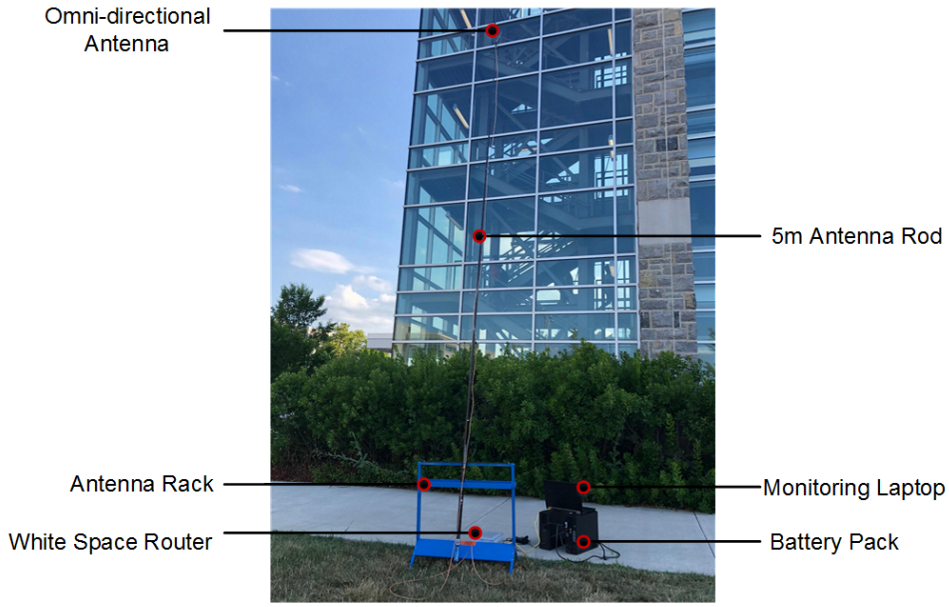


Figure 4.3: Experiment setup for field measurements.

4.4 Link Measurements

In this section, we perform link measurements on the field and use ocean-surface wave propagation simulation to confirm long-distance communication capability of our white space router.

Experiment Setups. We set up our equipment as shown in Figure 4.3. Two routers are put on two sides of a street with 500-meter line-of-sight separation distance. The two routers are configured to establish an 802.11g mesh backhaul link between them. As allowed by white space spectrum database, TV channel 14 with center frequency 473 MHz and a 5 MHz bandwidth are used for link measurement. The two routers both transmit at 25 dBm via a 5-meter omni-directional antenna with 2 dBi gain. In order to test the link quality with different receiving signal strength, we add controllable attenuation to reduce the signal strength to desired levels.

Table 4.1: Link quality results.

RSSI (dBm)	MCS	UDP Throughput	PER	RTT Delay (ms)
> -68	64-QAM 3/4 or 2/3	3.14–5.22 Mbps	0.044%–0.25%	1.57
-77–-85	QPSK 1/2 or BPSK 1/2	1.43–2.82 Mbps	0.24%–1.5%	2.44
< -87	DSSS	11.8–23.5 Kbps	33%–50%	N/A

Data Collection. We build a real-time monitoring system to display and log important data. The monitoring system is written in Shell and Python, which collects and processes low-level information from driver and sensors. The real-time data includes receiving signal strength indicator (RSSI) and modulation coding scheme (MCS), round-trip (RTT) delay, UDP throughput and packet error rate (PER), sensor data stream, and surveillance video stream. A fusion of these data enables the router to monitor link quality and ocean environment in real time, which are very useful for adaptive network control.

Link Quality Results. The link quality results are shown in Table 4.1. As we can see, our white space router can maintain a decent link quality with around 2 Mbps UDP throughput even when the receiving signal strength is as low as -85 dBm. Based on the link quality results, the communication distance of a white space router on the ocean can be estimated by using two-ray model to calculate ocean-surface path loss, which has been demonstrated by measurements for WiMAX at 5.8 GHz on the ocean [108]. For our experiment settings, if the white space router transmits at maximum power level (30 dBm), the simulator predicts a 5-kilometer coverage distance with -85 dBm receiving signal strength. This means one white space router can possibly cover more than 78.5 square kilometers area and provide a communication link with around 2 Mbps UDP throughput. It demonstrates the potential of our white space router to provide high-speed long-distance connection for real-time maritime applications with high data rate requirement.

4.5 Chapter Summary

In this chapter, we proposed an energy harvesting maritime mesh network, presented the design and implementation of a low-cost low-power white space router prototype. The preliminary link measurement and wave propagation simulation results confirmed its feasibility to provide high-speed long-distance connectivity on the ocean.

Chapter 5

Conclusions

In this dissertation, we conducted a series of research works through two threads—threat and application of frequency-agile radio systems. On the one hand, we studied the feasibility, impact, and countermeasures of GPS spoofing in road navigation systems and database-driven cognitive radio networks. On the other hand, we explored the feasibility of building white space radios to provide connectivity on the ocean.

In Chapter 2, a low-cost portable GPS spoofer was implemented and the feasibility of GPS spoofing targeting road navigation systems was confirmed by physical measurements and real-world driving tests. Novel stealthy attacking algorithms were proposed and evaluated on real-world taxi traces in Manhattan and Boston. We conducted deceptive user study to understand how human drivers react when under attack. The results showed that 95% of the participants followed the navigation instructions to wrong destinations without recognizing the attack. We summarized possible countermeasures and explored sensor fusion defense mechanisms.

In Chapter 3, a GPS location spoofing threat model was proposed for database-driven cog-

nitive radio networks, which can potentially create serious primary user interference and denial-of-service to secondary users. The impact of such an attack was examined in simulations with WhiteSpaceFinder spectrum database. Based on the characteristics of centralized system and the receiving capability of cognitive radios, a combination of three defense mechanisms was proposed to mitigate such threat.

In Chapter 4, we proposed an energy harvesting maritime mesh network, designed and implemented a white space router prototype. Link measurements on the field and ocean-surface propagation simulations were performed to confirm its capability of high-speed long-distance communications. The results demonstrated the potential of applying white space radios to provide connectivity on the ocean.

Chapter 6

Future Work

In this chapter, we introduce the future work along the directions of GPS spoofing threat and white space radio application on the ocean.

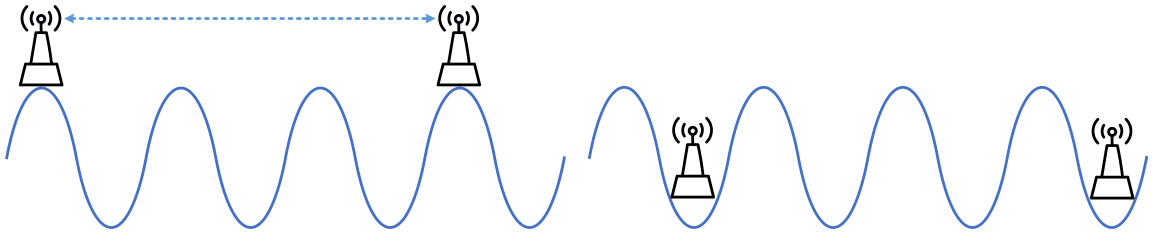
6.1 Practical Countermeasures for GPS Spoofing

As discussed in Section 2.8, existing countermeasures for GPS spoofing attack, such as signal encryption, ground infrastructure for location verification, and hardware/software modifications on GPS receivers, all suffer from high cost and/or long deployment cycle limitations. Therefore, we are exploring practical countermeasures that are modification free and can be easily deployed.

Countermeasures for Location Spoofing in Road Navigation Scenario. As demonstrated in Section 2.8, cross-checking surrounding environment with GPS location is a promising defense mechanism against GPS spoofing attack. Thanks to the powerful sensors equipped on vehicles, especially for autonomous cars, there is rich sensor data collected

for surrounding environment sensing. For example, camera and LIDAR sensors are used to understand, analyze the surrounding environment, and also refine coarse-grained GPS location to fine-grained location. However, it is still unknown that how would the localization algorithm behave under GPS spoofing attack that feeds specially-crafted false locations. Therefore, more attacking experiments targeting state-of-the-art autonomous driving systems need to be done. After this, we will develop a sensor fusion mechanism that is robust to GPS spoofing attack (and other sensor attacks). For example, location verification using textual information has been demonstrated as a promising way to defend against GPS spoofing attack. The challenges arise from applying it to long-term large-scale scenarios, such as differentiating permanent and temporary textual landmarks, making the system robust to bad light conditions, etc. More work should be done towards addressing such challenges.

Countermeasures for Generic Location/Time Spoofing. GPS spoofing can not only cause havoc to road navigation systems, but also critical infrastructures like power grids and financial markets. Therefore, we are also working on practical countermeasures for generic GPS spoofing attack including location and time spoofing. The key insight is to leverage the different angle-of-arrival for legitimate GPS satellites and counterfeit GPS spoofers set up by attackers. Specifically, due to the various angle-of-arrival that is consistent with the locations of GPS satellites in the space, the receiving signals should reveal predictable patterns. However, it is very hard for single or multiple spoofers to emulate such patterns. Based on this fact, an effective countermeasure can be developed for systematic GPS spoofing detection.



(a) Communication link is on. (b) Communication link is off.

Figure 6.1: Unstable communication link caused by dynamic ocean wave.

6.2 Mesh Networking on the Ocean

Existing research works have proposed various network designs and protocols for terrestrial mesh networks, whose feasibility has also been demonstrated by real-world deployments. Compared with the traditional terrestrial mesh networks, there are some unique challenges in maritime mesh networks brought by the intrinsic dynamics of ocean environment. One of the challenges is that the backhaul communication links will be affected by dynamic ocean wave. On land, the mesh nodes are usually mounted on the roof of tall buildings, which always maintain a stable line-of-sight communication link between each other. However, small-sized floating buoys cannot support such high antennas due to stability issues. Therefore, as the ocean becomes rough, the communication links are affected by the instant height of the transmitting and receiving nodes and the wave height between them. This might cause quality degradation or even blockage for the communication links. An example is shown in Figure 6.1, with a rough sea state, the communication link will be on and off when both buoys are on the crest and trough of ocean wave, respectively. In a mesh network, such unstable links can cause long packet delays and also affect the convergence of routing algorithms. Without carefully designed solutions, the entire mesh network might not work on rough sea states. Hence, our network design should be able to handle such dynamic sea states.

Ocean Wave Modeling. To understand the dynamics of ocean surface, the first step is to build an ocean wave model. We adopt a standard model in oceanography that represents naturally occurring wind-waves in the open ocean [109]. More specifically, it first creates an $L_x \times L_y$ rectangular ocean area. For a location $\mathbf{x} = (x, y)$, we denote its wave height at time t as $h(\mathbf{x}, \mathbf{t})$. It can be represented as the sum of sinusoids with complex and time-independent amplitudes $h(\mathbf{x}, t) = \text{Re}\{\sum_{\mathbf{k}} \widetilde{h}(\mathbf{k}, t) \exp(i\mathbf{kx})\}$, where $\mathbf{k}$ is a two-dimensional vector $\mathbf{k} = (\mathbf{k_x}, \mathbf{k_y})$, $k_x = \frac{2\pi n}{L_x}$, $k_y = \frac{2\pi m}{L_y}$, $-\frac{N}{2} \leq n \leq \frac{N}{2}$, and $-\frac{M}{2} \leq m \leq \frac{M}{2}$. Given $\widetilde{h}(\mathbf{k}, t)$, $h(\mathbf{x}, t)$ can be computed by inverse FFT (fast Fourier transform). Demonstrated by statistical analysis of real ocean monitoring data, $\widetilde{h}(\mathbf{k}, t)$ is nearly statistically stationary and independent Gaussian fluctuations. Mathematically, $\widetilde{h}(\mathbf{k}, t) = \widetilde{h}_0(\mathbf{k}) \exp(i\omega t) + \widetilde{h}_0^*(-\mathbf{k}) \exp(-i\omega t)$, where $\omega = \sqrt{g\frac{2\pi}{\lambda}}$ in deep water and λ is the wave length. The Fourier amplitudes of a wave height $\widetilde{h}_0(\mathbf{k}) = \frac{1}{\sqrt{2}}(\xi_r + i\xi_i)\sqrt{P(\mathbf{k})}$, where ξ_r and ξ_i are independent random numbers with Gaussian distribution $N(0, 1)$. $P(\mathbf{k}) = \frac{A}{|\mathbf{k}|^4} \exp(\frac{-g^2}{|\mathbf{k}|^2 V^4})(\frac{\mathbf{k}}{|\mathbf{k}|}\mathbf{d})$, where A is a scaling constant, g is the gravitational constant, V is the wind speed, and $\mathbf{d}$ is the wind direction. We implement a simulator that provides absolute value for ocean wave height by modifying the open-source code [110].

Link State Modeling and Prediction. For two buoys with a certain antenna height (*e.g.*, 5 meters) and separation distance (*e.g.* 5 kilometers), their line-of-sight is the dot line as shown in Figure 6.1a. With the ocean wave model and the earth curvature, the existence of a blockage can be determined by checking if there is any point in between with a wave height greater than the height of the line-of-sight. Moreover, based on the readings of accelerometer and gyroscope sensors equipped on the router, the real-time movement of ocean wave can be derived. Based on the history data of ocean wave movement, machine learning techniques can be applied to predict ocean wave height and buoy position in the next a few seconds, which can be converted to link state estimation.

Adaptive Network Protocol Design and Evaluation. With the link state prediction, adaptive routing algorithms can direct packets along links that are currently at a good state. Besides, link layer and transport layer protocols can adjust their timeout and retransmission policy to cope with links with bad quality. Beyond link-state awareness, the network protocol can also be energy-aware, because the amount of harvested energy is directly determined by ocean wave motion. This kind of information can be obtained from local energy harvesting history and weather forecast. Therefore, an energy-aware scheme would be very useful for avoiding network disconnection caused by power outage of individual nodes. For evaluation, we are building a maritime mesh network simulator in ns-3, which integrates ocean wave model and link state model. This will enable users to evaluate the network performance in various sea states. The field measurement results for white space link quality will also be plugged into the simulator for better approximation of real-world deployment.

Chapter 7

Bibliography

[1] B. Smith, "A rural broadband strategy: connecting rural America to new opportunities," Microsoft, 2017, https://blogs.microsoft.com/on-the-issues/2017/07/10/rural-broadband-strategy-connecting-rural-america-new-opportunities/.

[2] D. Meyer, "Microsoft and Facebook turn to white space broadband to connect Ghanaian students," GIGAOM, 2014, https://gigaom.com/2014/05/13/microsoft-and-facebook-turn-to-white-space-broadband-to-connect-ghanaian-students/.

[3] D. Hambling, "Ships fooled in GPS spoofing attack suggest Russian cyberweapon," New Scientist, 2017, https://www.newscientist.com/article/2143499-ships-fooled-in-gps-spoofing-attack-suggest-russian-cyberweapon/.

[4] K. C. Zeng, Y. Shu, S. Liu, Y. Dou, and Y. Yang, "A practical GPS location spoofing attack in road navigation scenario," in *HotMobile Workshop*, 2017.

[5] K. C. Zeng, S. Liu, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, and Y. Yang, "All Your GPS Are Belong To Us: Towards stealthy manipulation of road navigation systems," in *USENIX Security*, 2018.

[6] K. C. Zeng, Y. Dou, Y. Yang, and R. Chandra, "Poster: Location verification and recovery for mobile in-vehicle applications," in *MobiSys*, 2015.

[7] K. Zeng, S. K. Ramesh, and Y. Yang, "Location spoofing attack and its countermeasures in database-driven cognitive radio networks," in *CNS*, 2014.

[8] ——, "Location robustness in database-driven white spaces network," in *DYSPAN*, 2014.

[9] B. Popper, "Google announces over 2 billion monthly active devices on Android," The Verge, 2017, https://www.theverge.com/2017/5/17/15654454/android-reaches-2-billion-monthly-active-users.

[10] N. Boudette, "Building a Road Map for the Self-Driving Car," The New York Times, 2017, https://www.nytimes.com/2017/03/02/automobiles/wheels/self-driving-cars-gps-maps.html.

[11] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful GPS spoofing attacks," in *CCS*, 2011.

[12] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *Journal of Field Robotics*, vol. 31, no. 4, pp. 617–636, 2014.

[13] M. L. Psiaki and T. E. Humphreys, "Protecting GPS From Spoofers Is Critical to the Future of Navigation," IEEE Spectrum, 2016, https://spectrum.ieee.org/telecom/security/protecting-gps-from-spoofers-is-critical-to-the-future-of-navigation.

[14] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. OHanlon, and P. M. Kintner Jr, "Assessing the spoofing threat: Development of a portable GPS civilian spoofer," in *ION GNSS*, 2008.

[15] L. Huang and Q. Yang, "Low-cost GPS simulator: GPS spoofing by SDR," DEFCON, 2015.

[16] K. Wang, S. Chen, and A. Pan, "Time and position spoofing with open source projects," BlackHat, 2015.

[17] "NYC Taxi & Limousine Commission Trip Record Data," NYC.gov, http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.

[18] "City of Boston Taxi Dataset," MIT Big Data Challenge, http://bigdata.csail.mit.edu/challenge.

[19] T. Nighswander, B. Ledvina, J. Diamond, R. Brumley, and D. Brumley, "GPS software attacks," in *CCS*, 2012.

[20] "UT Austin Researchers Successfully Spoof an $80 million Yacht at Sea," UTNews, 2013, https://news.utexas.edu/2013/07/29/ut-austin-researchers-successfully-spoof-an-80-million-yacht-at-sea.

[21] "WALB (Wireless Attack Launch Box)," https://github.com/crescentvenus/WALB.

[22] "The Measurement of Angles," The Oxford Math Center, http://www.oxfordmathcenter.com/drupal7/node/489.

[23] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir, "Inferring user routes and locations using zero-permission mobile sensors," in *IEEE SP*, 2016.

[24] C. Bo, X.-Y. Li, T. Jung, X. Mao, Y. Tao, and L. Yao, "Smartloc: Push the limit of the inertial sensor based metropolitan localization using smartphone," in *MobiCom*, 2013.

[25] "How Waze determines turn / keep / exit maneuvers," Waze, https://wiki.waze.com/wiki/How_Waze_determines_turn_/_keep_/_exit_maneuvers.

[26] "ETS2 Telemetry Web Server 3.2.5 + Mobile Dashboard," https://github.com/Funbit/ets2-telemetry-server.

[27] "HUNGARY_MAP v0.9.28a [1.27]," https://forum.scssoft.com/viewtopic.php?t=24305.

[28] D. Muoio, "19 companies racing to put self-driving cars on the road by 2021," Business Insider, 2016, http://www.businessinsider.com/companies-making-driverless-cars-by-2020-2016-10/.

[29] M. G. Kuhn, "An asymmetric security mechanism for navigation signals," in *IH&MMSec*, 2004.

[30] K. Wesson, M. Rothlisberger, and T. Humphreys, "Practical cryptographic civil GPS signal authentication," *Navigation*, vol. 59, no. 3, pp. 177–193, 2012.

[31] S. Brands and D. Chaum, "Distance-bounding protocols," in *EUROCRYPT*, 1993.

[32] K. B. Rasmussen and S. Capkun, "Realization of RF distance bounding," in *USENIX Security*, 2010.

[33] M. Schäfer, V. Lenders, and J. Schmitt, "Secure track verification," in *IEEE SP*, 2015.

[34] K. Jansen, M. Schäfer, D. Moser, V. Lenders, C. Pöpper, and J. Schmitt, "Crowd-GPS-Sec: Leveraging crowdsourcing to detect and localize GPS spoofing attacks," in *IEEE SP*, 2018.

[35] D. Moser, P. Leu, V. Lenders, A. Ranganathan, F. Ricciato, and S. Capkun, "Investigation of multi-device location spoofing attacks on air traffic control and possible countermeasures," in *MobiCom*, 2016.

[36] P. Y. Montgomery, T. E. Humphreys, and B. M. Ledvina, "Receiver-autonomous spoofing detection: Experimental results of a multi-antenna receiver defense against a portable civil GPS spoofer," in *ION ITM*, 2009.

[37] M. L. Psiaki, S. P. Powell, and B. W. Ohanlon, "GNSS spoofing detection using high-frequency antenna motion and carrier-phase data," in *ION GNSS*, 2013.

[38] J. S. Warner and R. G. Johnston, "GPS spoofing countermeasures," *Homeland Security Journal*, vol. 25, no. 2, pp. 19–27, 2003.

[39] B. M. Ledvina, W. J. Bencze, B. Galusha, and I. Miller, "An in-line anti-spoofing device for legacy civil GPS receivers," in *ION ITM*, 2001.

[40] K. D. Wesson, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "An evaluation of the vestigial signal defense for civil GPS anti-spoofing," in *ION GNSS*, 2011.

[41] A. Ranganathan, H. Ólafsdóttir, and S. Capkun, "SPREE: A spoofing resistant GPS receiver," in *MobiCom*, 2016.

[42] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer Science & Business Media, 2007.

[43] P. A. Zandbergen, "Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning," *Transactions in GIS*, vol. 13, no. s1, pp. 5–25, 2009.

[44] N. O. Tippenhauer, K. B. Rasmussen, C. Pöpper, and S. Čapkun, "Attacks on public WLAN-based positioning systems," in *MobiSys*, 2009.

[45] C. Paget, "Practical cellphone spying," DEFCON, 2010.

[46] J. Farrell and M. Barth, *The global positioning system and inertial navigation*. McGraw-Hill New York, NY, USA:, 1999, vol. 61.

[47] D. Titterton and J. L. Weston, *Strapdown inertial navigation technology*. IET, 2004, vol. 17.

[48] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *CVPR*, 2004.

[49] M. A. Brubaker, A. Geiger, and R. Urtasun, "Lost! leveraging the crowd for probabilistic visual self-localization," in *CVPR*, 2013.

[50] A. R. Zamir and M. Shah, "Accurate image localization based on google maps street view," in *ECCV*, 2010.

[51] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on machine learning models," *arXiv*, vol. abs/1707.08945, 2017.

[52] W. Xu, Y. Qi, and D. Evans, "Automatically evading classifiers," in *NDSS*, 2016.

[53] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv*, vol. abs/1706.06083, 2017.

[54] W. He, J. Wei, X. Chen, N. Carlini, and D. Song, "Adversarial Example Defenses: Ensembles of weak defenses are not strong," *arXiv*, vol. abs/1706.04701, 2017.

[55] W. Xu, D. Evans, and Y. Qi, "Feature Squeezing: Detecting adversarial examples in deep neural networks," *arXiv*, vol. abs/1704.01155, 2017.

[56] C. Valgren and A. J. Lilienthal, "SIFT, SURF and Seasons: Long-term outdoor localization using local features," in *EMCR*, 2007.

[57] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999.

[58] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," *VISAPP (1)*, vol. 2, no. 331-340, p. 2, 2009.

[59] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *ECCV*, 2006.

[60] "Google Goggles," https://play.google.com/store/apps/details?id=com.google.android.apps.unveil&hl=en_US.

[61] "Worlds population increasingly urban with more than half living in urban areas," United Nations, 2014, http://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html.

[62] J. Volpe, "Vulnerability assessment of the transportation infrastructure relying on the global positioning system," *Technical Report*, 2001.

[63] J. S. Warner and R. G. Johnston, "A simple demonstration that the global positioning system (GPS) is vulnerable to spoofing," *Journal of Security Administration*, vol. 25, no. 2, pp. 19–27, 2002.

[64] F. Dovis, *GNSS Interference Threats and Countermeasures*. Artech House, 2015.

[65] B. Motella, M. Pini, M. Fantino, P. Mulassano, M. Nicola, J. Fortuny-Guasch, M. Wildemeersch, and D. Symeonidis, "Performance assessment of low cost GPS receivers under civilian spoofing attacks," in *NAVITEC*, 2010.

[66] G. Wang, B. Wang, T. Wang, A. Nika, H. Zheng, and B. Y. Zhao, "Defending against sybil devices in crowdsourced mapping services," in *MobiSys*, 2016.

[67] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu, "WALNUT: Waging doubt on the integrity of MEMS accelerometers with acoustic injection attacks," in *EuroS&P*, 2017.

[68] B. Farshteindiker, N. Hasidim, A. Grosz, and Y. Oren, "How to Phone Home with Someone Else's Phone: Information exfiltration using intentional sound noise on gyroscopic sensors." in *WOOT*, 2016.

[69] Y. Son, H. Shin, D. Kim, Y.-S. Park, J. Noh, K. Choi, J. Choi, Y. Kim *et al.*, "Rocking drones with intentional sound noise on gyroscopic sensors," in *USENIX Security*, 2015.

[70] Y. Michalevsky, D. Boneh, and G. Nakibly, "Gyrophone: Recognizing speech from gyroscope signals," in *USENIX Security*, 2014.

[71] L. Deshotels, "Inaudible sound as a covert channel in mobile devices," in *WOOT*, 2014.

[72] Y.-S. Park, Y. Son, H. Shin, D. Kim, and Y. Kim, "This Ain't Your Dose: Sensor spoofing attack on medical infusion pump," in *WOOT*, 2016.

[73] D. F. Kune, J. Backes, S. S. Clark, D. Kramer, M. Reynolds, K. Fu, Y. Kim, and W. Xu, "Ghost talk: Mitigating EMI signal injection attacks against analog sensors," in *IEEE SP*, 2013.

[74] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinatack: Inaudible voice commands," in *CCS*, 2017.

[75] D. Davidson, H. Wu, R. Jellinek, V. Singh, and T. Ristenpart, "Controlling UAVs with sensor input spoofing attacks," in *WOOT*, 2016.

[76] N. M. Duc and B. Q. Minh, "Your face is NOT your password: Face authentication bypassing Lenovo -Asus-Toshiba," *BlackHat*, 2009.

[77] J. Galbally, R. Cappelli, A. Lumini, D. Maltoni, and J. Fierrez, "Fake fingertip generation from a minutiae template," in *ICPR*, 2008.

[78] H. Shin, Y. Son, Y.-S. Park, Y. Kwon, and Y. Kim, "Sampling Race: Bypassing timing-based analog active sensor spoofing detection on analog-digital systems," in *WOOT*, 2016.

[79] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," in *DEFCON*, 2016.

[80] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and Dazzle: Adversarial optical channel exploits against Lidars for automotive applications," in *CHES*, 2017.

[81] Y. Shoukry, P. Martin, P. Tabuada, and M. Srivastava, "Non-invasive spoofing attacks for anti-lock braking systems," in *CHES*, 2013.

[82] "Unlicensed Operation in the TV Broadcast Bands (ET Docket No. 04-186)," FCC, 2012, https://apps.fcc.gov/edocs_public/attachmatch/FCC-12-36A1_Rcd.pdf.

[83] "Amendment of the Commissions Rules with Regard to Commercial Operations in the 3550-3650 MHz Band (GN Docket No. 12-354)," FCC, 2017, https://www.federalregister.gov/documents/2017/06/12/2017-12117/amendment-of-the-commissions-rules-with-regard-to-commercial-operations-in-the-3550-3650-mhz-

[84] G. Locke, L. E. Strickling, and A. Secretary, "An assessment of the near-term viability of accommodating wireless broadband systems in the 1675-1710 mhz, 1755-1780 mhz, 3500-3650 mhz, and 4200-4220 mhz, 4380-4400 mhz bands," *no. October*, pp. 3500–3650, 2010.

[85] "White spaces database," http://whitespaces.microsoftspectrum.com/.

[86] W. Wang and X. Liu, "List-coloring based channel allocation for open-spectrum wireless networks," in *VTC-Fall*, 2005.

[87] Y. Zhao, J. Gaeddert, K. K. Bae, and J. H. Reed, "Radio environment map enabled situation-aware cognitive radio learning algorithms," in *SDRF*, 2006.

[88] C. Ziegler, "Rosum's Alloy chip promises 'precise' location using TV signals," engadget, 2010, http://www.engadget.com/2010/03/01/rosums-alloy-chip-promises-precise-location-using-tv-signals/.

[89] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, "Epidemic spreading in real networks: An eigenvalue viewpoint," in *SRDS*, 2003.

[90] S.-H. Fang, J.-C. Chen, H.-R. Huang, and T.-N. Lin, "Metropolitan-scale location estimation using FM radio with analysis of measurements," in *IWCMC*, 2008.

[91] Y. Cui and Y. Wu, "Protocol to Access White Space Database: Security considerations," 2012.

[92] Y. Dou, K. C. Zeng, Y. Yang, and D. D. Yao, "MadeCR: Correlation-based malware detection for cognitive radio," in *INFOCOM*, 2015.

[93] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: Attacks and countermeasures," in *INFOCOM*, 2013.

[94] Y. Dou, K. C. Zeng, H. Li, Y. Yang, B. Gao, C. Guan, K. Ren, and S. Li, "P 2-SAS: preserving users' privacy in centralized dynamic spectrum access systems," in *MobiHoc*, 2016.

[95] Y. Dou, K. C. Zeng, and Y. Yang, "Poster: Privacy-preserving server-driven dynamic spectrum access system," in *MobiCom*, 2015.

[96] Y. Dou, K. C. Zeng, Y. Yang, and K. Ren, "Preserving incumbent users' privacy in exclusion-zone-based spectrum access systems: poster," in *MobiCom*, 2016.

[97] Y. Dou, H. Li, K. Zeng, J. Liu, Y. Yang, B. Gao, and K. Ren, "Preserving incumbent users' privacy in server-driven dynamic spectrum access systems," in *ICDCS*, 2016.

[98] Y. Dou, H. Li, K. C. Zeng, J. Liu, Y. Yang, B. Gao, and K. Ren, "Preserving incumbent users privacy in exclusion-zone-based spectrum access systems," in *ICDCS*, 2017.

[99] Y. Dou, K. Zeng, H. Li, Y. Yang, B. Gao, K. Ren, and S. Li, "P^2-SAS: Privacy-preserving centralized dynamic spectrum access system," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 1, pp. 173–187, 2017.

[100] "Addvalue Inmarsat Fleet One," http://www.globalmarinenet.com/product/addvalue-inmarsat-fleet-one-2/.

[101] "Monthly Inmarsat BGAN Plans," https://satellitephonestore.com/services/inmarsatBgan.

[102] "JAVAD VHF Radio," https://www.javad.com/jgnss/products/radios/vhf.html.

[103] T. Hornyak, "Google's 60Tbps Pacific cable welcomed in Japan," COMPUTERWORLD, 2015, https://www.computerworld.com/article/2939316/networking-hardware/googles-60tbps-pacific-cable-welcomed-in-japan.html.

[104] S. Roberts, P. Garnett, and R. Chandra, "Connecting Africa Using the TV White Spaces: From research to real world deployments," in *LANMAN*, 2015.

[105] A. Kumar, A. Karandikar, G. Naik, M. Khaturia, S. Saha, M. Arora, and J. Singh, "Toward enabling broadband for a billion plus population with TV white spaces," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 28–34, 2016.

[106] "GWS 4000 SERIES RURAL BROADBAND SOLUTION," http://queenstvws.com/GWS%204000%20Series%20Datasheet%20FCC%20certified%20July%202016-2.pdf.

[107] "Openwrt," https://openwrt.org/.

[108] J. Joe, S. Hazra, S. Toh, W. Tan, J. Shankar, V. D. Hoang, and M. Fujise, "Path loss measurements in sea port for WiMAX," in *WCNC*, 2007.

[109] J. Tessendorf *et al.*, "Simulating ocean water," *Simulating nature: realistic and interactive techniques. SIGGRAPH*, vol. 1, no. 2, p. 5, 2001.

[110] Hoki, "Ocean Simulator," https://www.dropbox.com/s/qow7cdf5z95t7hx/ocean_simulator.m?dl=0.

Appendix A

Visualization and Illustration

A.1 Taxi Route Visualization

Figure A.1 and Figure A.2 visualize the 600 taxi routes in Manhattan and Boston that are used for our evaluation. In our experiments, the considered area in Manhattan is 10.64 km×7.38 km with a latitude range (40.7003, 40.7959) and a longitude range (-74.0180, -73.9308). The considered experiment area in Boston is 8.52km×10.60km with a latitude range (42.3134, 42.3885) and a longitude range (-71.1435, -71.0149). As shown in Figure A.1 and Figure A.2, the taxi routes are concentrated in the downtown areas in both respective maps.

A.2 Attack Area and Grids

In the Targeted Deviating Attack, the attacker aims to divert the user to a pre-defined location. Our evaluation metric will focus on *hit rate*. In the following, we briefly explain

how to calculate the hit rate. For a given taxi trip, the hit rate reflects how likely a victim route can bypass the attacker-defined destination to achieve targeted diverting. Figure A.3 shows how we define the attack area, radius r and divide the grids. Given an attack area with the radius of r, the attacker can pick a grid inside the area as the target destination. Hit rate is the ratio of the grids that the victim can be diverted to over all the grids in the attack area.

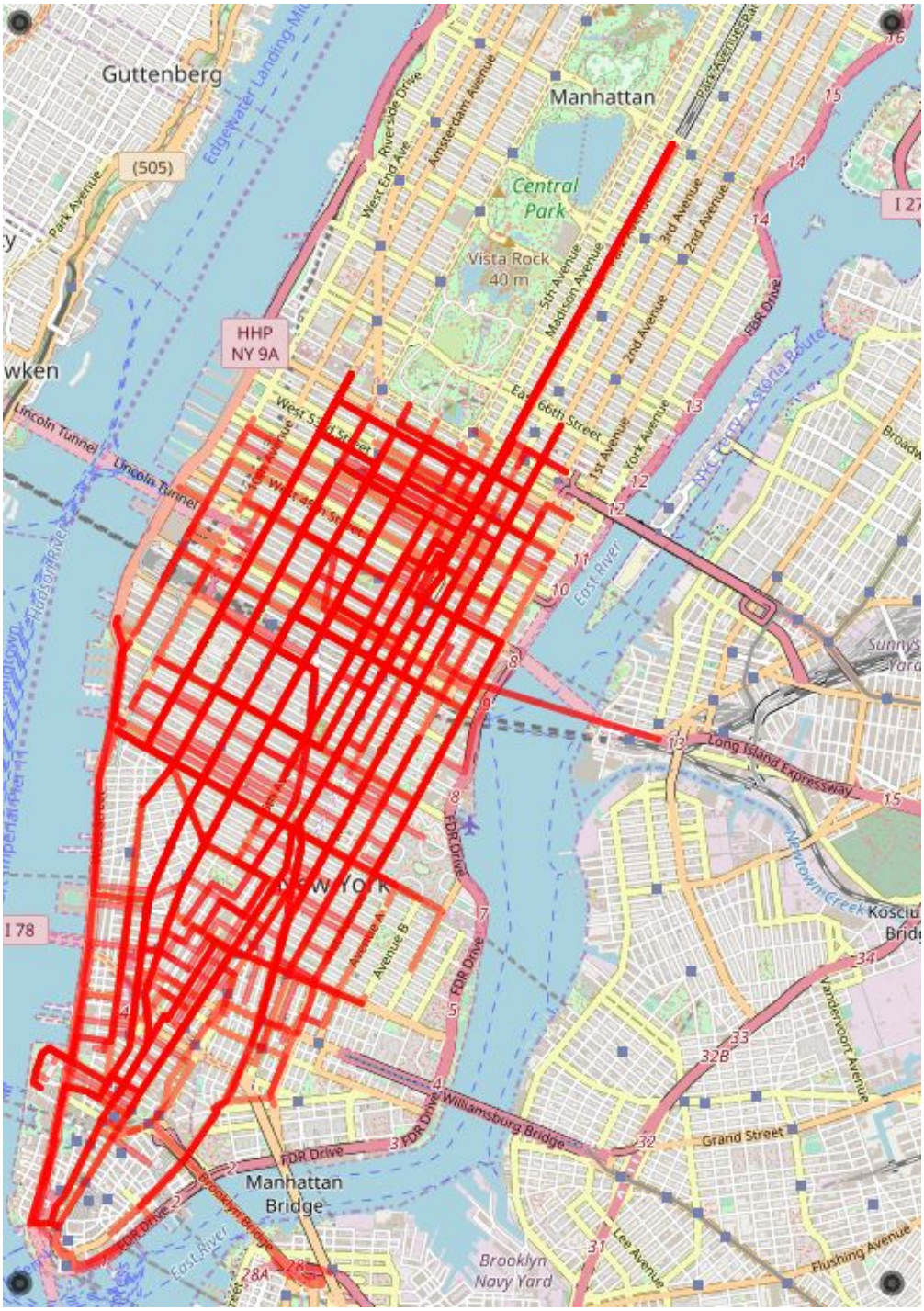


Figure A.1: Visualization of 300 taxi routes in Manhattan.

Figure A.2: Visualization of 300 taxi routes in Boston.

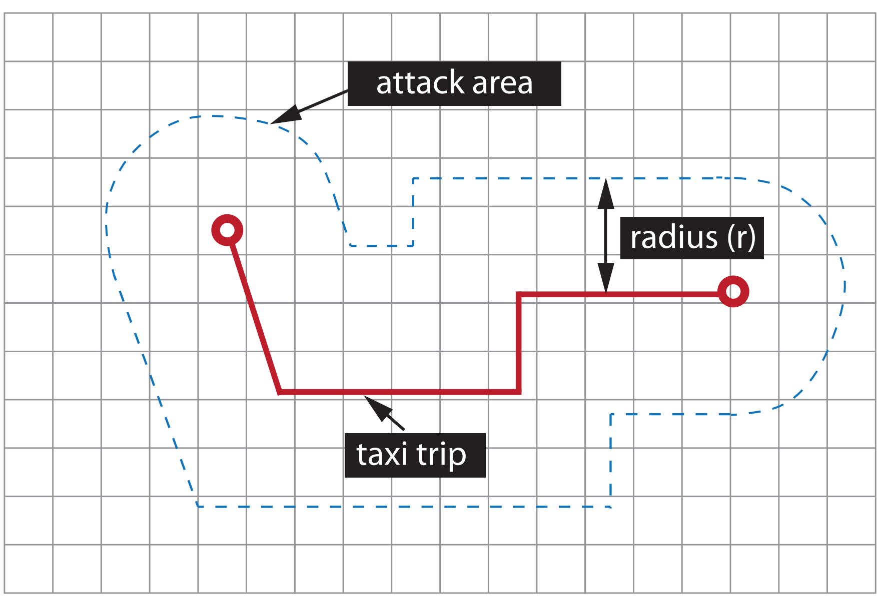


Figure A.3: Illustration of the attack area and grids.