All Your GPS Are Belong To Us: Towards Stealthy Manipulation of Road Navigation Systems

Kexiong (Curtis) Zeng[†], Shinan Liu[‡], Yuanchao Shu[§], Dong Wang[†]
Haoyu Li[†], Yanzhi Dou[†], Gang Wang[†], Yaling Yang[†]
[†]*Virginia Tech;* [‡]*University of Electronic Science and Technology of China;* [§]*Microsoft Research*
{*kexiong6, dong0125, haoyu7, yzdou, gangwang, yyang8*}*@vt.edu; liushinan63@163.com; yuanchao.shu@microsoft.com*

Abstract

Mobile navigation services are used by billions of users around 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 paper, we explore the feasibility of a stealthy manipulation attack against road navigation systems. 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. We use the results to discuss countermeasures moving forward.

1 Introduction

Billions of users around globe are relying on mobile navigation services today [45]. 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 [11].

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 [55]. 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 [28]. In 2013, a luxury yacht was intentionally diverted from Monaco to Greece by spoofing its receiving GPS signals [46].

To understand the risks of GPS spoofing attacks, researchers have explored to build GPS spoofers to spoof drones, ships and wearable devices [25,26,61]. However, these works mainly focus on simple attacks by setting random locations in the target device [25, 26, 61]. Other works have examined GPS spoofing attacks on systems in the open environment (*e.g.,* open air/water) such as drones and ships [28, 46] 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 paper, 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 algorithms using trace-driving simulations (600 taxi trips in total) from Manhattan [5] and Boston [1]. 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.

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

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 paper 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 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) [26, 41]. Next, in the second step, the attacker can manipulate the GPS receiver by either shifting the signals' arrival time or modifying the navigation messages [41, 46].

2.1 Threat Model

In this paper, 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 [8, 25, 28, 46, 61]., 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).

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 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) [6] running on Raspberry Pi. The Raspberry Pi has a cellular network connection and supports remote access through SSH (Se-

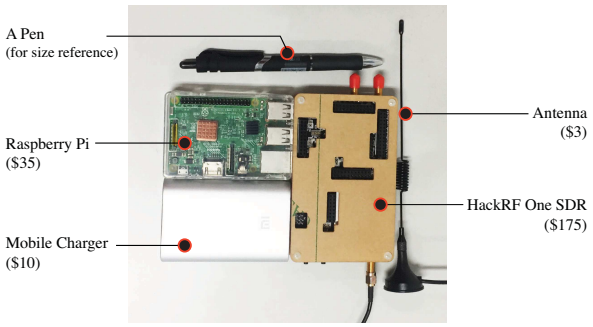
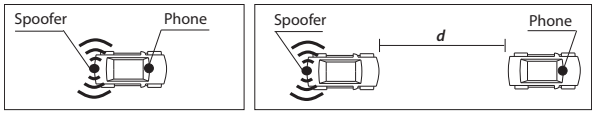


Figure 1: A low-cost portable GPS spoofer.



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

Figure 2: Measurement setups.

cure 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. 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). 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 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.

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

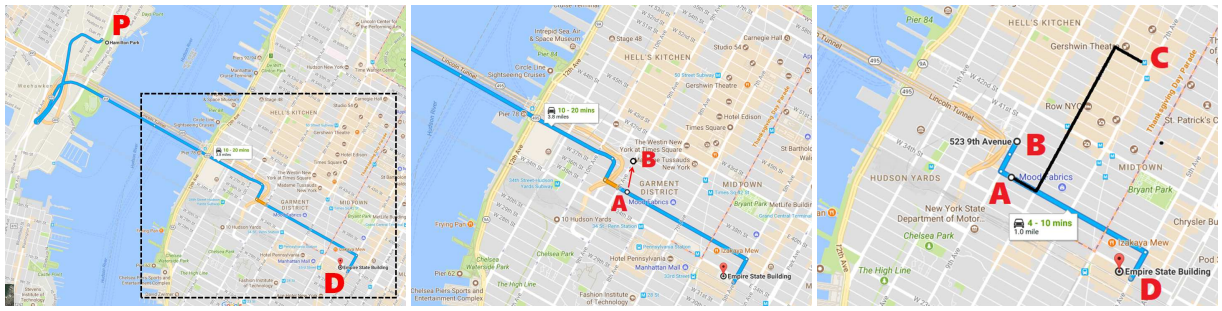
Table 1: Average takeover time and the failure rate.

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 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.

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



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

Figure 3: An attack example: the victim's original navigation route is $P \rightarrow D$; At location A, the spoofer sets the GPS to a ghost location B which forces the navigation system to generate a new route $B \rightarrow D$. Following the turn-by-turn navigation, the victim actually travels from A to C in the physical world.

the next turn, which is likely to increase the attacker's chance of success.

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 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 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 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 $B \rightarrow D$ route is similar with that of the $A \rightarrow C$ 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 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: Notation and definition.

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.

Road Model. As shown in Figure 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 [4]. $\phi_i > 0$ and $\phi_i < 0$ indicate a left and right turn respectively.

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

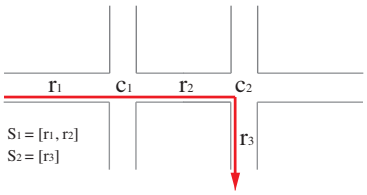


Figure 4: Road model example.

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 location 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.

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.

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 [46].

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

Input: $G, D, Loc_a, \Omega_{driftDis}, \Omega_{speed}$
Output: $O = \{o_1, o_2, ..., o_K\}$, $o_i = (\Gamma_v, \Gamma_g, Loc_g)_i$

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

ALGORITHM 1: Basic attack algorithm

Input: $G, D, \Omega_{driftDis}, \Omega_{speed}$, O_0, I, attack goal
Output: 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

ALGORITHM 2: Iterative attack algorithm

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.

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.

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.

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.

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 [39] 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 [39].

Driving Trace Dataset. To examine the attack performance on realistic driving trips, we obtain taxi trip datasets from NYC Taxi and Limousine Commission (TLC) [5] and the Boston taxi trace dataset used by MIT Challenge [1]. 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). 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 $\Omega_{driftDis}$ = 400m. A measurement study shows that a GPS drift of less than 400m is common during active driving [10]. In addition, given the speed limits in the two cities are 25 to 30 mph, we set Ω_{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 [7] 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.

6.2 Evaluation Results

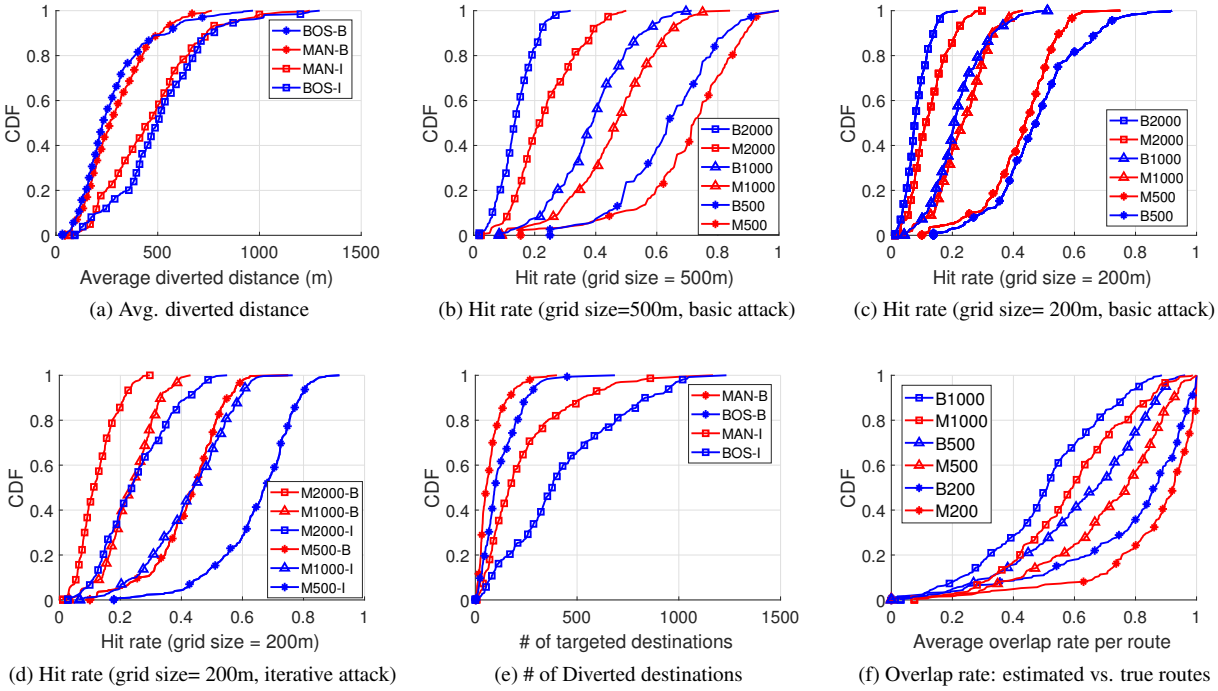
The performance metric depends on the specific goal of the attacker. Recall in our threat model (Section 2.1), we defined three types of attacks which need different evaluation metrics. Below, our metrics are all based on each of the taxi trips (per-trip metric).

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 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-B.

Figure 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



(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 5: Attack results in Manhattan (MAN) and Boston (BOS). B = Basic Attack; I = Iterative Attack; M500 = Manhattan with a 500m grid size; B500 = Boston with a 500m grid size.

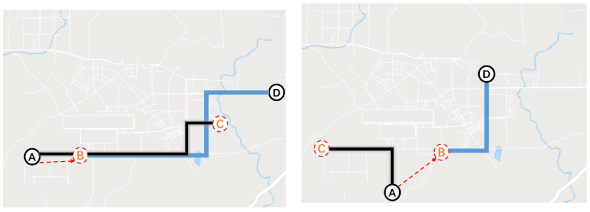
r= 500m, 1000m, and 2000m respectively. This indicates that even a randomly selected destination grid is highly likely to be reachable. No surprisingly, victim routes get sparser when it is further away from the original route. Note that even with 20% hit rate in 2000m range, if the attacker provides three candidate target destination grids, the success rate will be higher $1-(1-0.2)^3 = 48.8\%$. Comparing Figure 5b and Figure 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 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 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 5b and Figure 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 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 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 di-



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

Figure 6: The original routes and victim routes in the real-world driving tests.

vert 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.

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 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 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 predefined location 2.1 kilometers away from the original destination. In the second case (Figure 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.

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).

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

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

Figure 7: User study setups; The ETS II Game View is comparable to the Google Street View at the same location.

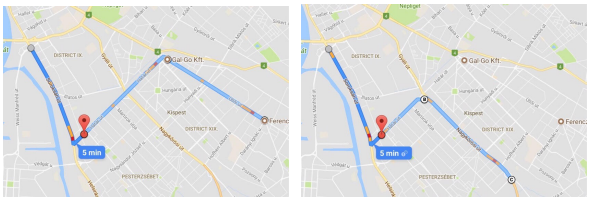
usability of our simulation software. We debrief users after the driving test to obtain the informed consent. The study takes about 50 minutes and we compensate each participant $10.

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) [2]. 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 7b and Figure 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 [3], 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 7a. For ethical and

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



(a) Original Route (b) Victim Route

Figure 8: The original and victim route for the user study.

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 8 shows the driving routes used in our user study. Figure 8a shows the original route that the participant is supposed to take. Figure 8b shows the route to which the attacker aims to detour the participants. This route is chosen because it 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 partic-

ipants 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.

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 successful 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

Mechanism		$ Cost	Deploy. Overhead	Effectiveness	Robustness
Modif.-based	Encryption & authentication [29, 64]	High	High	High	High
	Ground infrastructures [12, 27, 36, 49, 50]	High	High	High	High
	GPS receiver hardware [24, 31, 35, 40, 47, 73]	Medium	High	High	High
	GPS receiver software [32, 35, 47, 48, 55, 63, 65]	Low	Low	Low	Low
Modif.-free	External location verification [23, 70]	Low	Low	Low	Low
	Internal sensor fusion [19, 57]	Low	Low	Low	Low
	Computer vision [13, 42, 69]	Low	Low	Medium	Unknown

Table 3: Comparison of different countermeasures.

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 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 paper.

8 Discussion and 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) [38]. In the following, we discuss key directions of countermeasures.

In Table 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, 64]. 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 [12, 49], multilateration [50], multi-receiver crowdsourcing [27] and physical-layer feature checks [36]. 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 [35], or special moving antenna [47]. 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 [32, 63, 65]. In addition, the GPS receiver can also lock on additional satellites [48] or synchronize with other GPS receivers [55] to identify spoofing. However, these methods often suffer from the multi-path effect and are vulnerable to smooth takeovers [26].

Modification-Free Approaches. First, location verification can leverage existing GNSS signals (*e.g.*, Galileo, GLONASS, Beidou) [23], and wireless network signals [70]. 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 [26]. Similarly, the network location is based on the MAC address of the WiFi or cell tower ID, which can also be jammed or spoofed [43, 56].

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) [19, 57]. 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 [13, 42]. SLAM (Simultaneous Localization And Mapping) can also localize images based on geo-referenced street view databases [69].

What remains unknown is the *robustness* of vision-based methods against adversarial manipulations. Recent works [18, 67] 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 [22,33,66]. Further research is needed to understand the feasibility of vision-based location verification.

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 [9], 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).

9 Related Work

GPS spoofing attack was first systematically discussed in [59]. To date, researchers and hackers have successfully spoofed GPS devices in moving trucks [62], ships [46], drones [28] and mobile platforms [25,61] using off-the-shelf GPS signal simulator [62] or software defined radios [25, 28, 46, 61]. Humphreys *et al.* have demonstrated seamless GPS takeover on a moving yacht with a portable receiver-spoofer [26]. Later, an attachable miniature version one called "limpet spoofer" was proposed in [16]. Similar technical concepts were also used in [37, 41] to develop spoofing devices. In [55], 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 [60]. Narain *et al.* proposed a route matching algorithm to infer user movement traces based on motion sensor data [39]. 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. [71] preliminarily formulated the route spoofing problem. Compared to [71], 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 [14, 15, 17, 20, 21, 30, 34, 44, 52, 54, 58, 72]. Some of the attacks specifically target (autonomous) vehicles to disrupt their ultra-sonic sensor, millimeter-wave radar, LIDAR, and wheel speed sensor [51,53,68]. 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.

10 Conclusion

In this paper, we explored the feasibility of real-time stealthy GPS spoofing attacks targeting road navigation systems. Real-world driving tests, 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.

Acknowledgments

We would like to thank anonymous reviewers for their helpful comments. This project was supported by NSF grants CNS-1750101, CNS-1717028, CNS-1547366, and CNS-1527239. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any funding agencies.

References

[1] City of Boston taxi dataset. MIT Big Data Challenge. http://bigdata.csail.mit.edu/challenge.

[2] Ets2 telemetry web server 3.2.5 + mobile dashboard. https://github.com/Funbit/ets2-telemetry-server.

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

[4] The measurement of angles. The Oxford Math Center. http://www.oxfordmathcenter.com/drupal7/node/489.

[5] NYC taxi & limousine commission trip record data. NYC.gov. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml.

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

[7] Waze documentation. https://wiki.waze.com/wiki/How_Waze_determines_turn_/_keep_/_exit_maneuvers.

[8] 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.

[9] 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.

[10] BO, C., LI, X.-Y., JUNG, T., MAO, X., TAO, Y., AND YAO, L. Smartloc: push the limit of the inertial sensor based metropolitan localization using smartphone. In *MobiCom* (2013).

[11] BOUDETTE, N. 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.

[12] BRANDS, S., AND CHAUM, D. Distance-bounding protocols. In *Advances in Cryptology-Eurocrypt* (1993).

[13] BRUBAKER, M. A., GEIGER, A., AND URTASUN, R. Lost! leveraging the crowd for probabilistic visual self-localization. In *CVPR* (2013).

[14] DAVIDSON, D., WU, H., JELLINEK, R., SINGH, V., AND RISTENPART, T. Controlling UAVs with sensor input spoofing attacks. In *WOOT* (2016).

[15] DESHOTELS, L. Inaudible sound as a covert channel in mobile devices. In *WOOT* (2014).

[16] DOVIS, F. *GNSS Interference Threats and Countermeasures*. Artech House, 2015.

[17] DUC, N. M., AND MINH, B. Q. Your face is not your password face authentication bypassing lenovo–asus–toshiba. *BlackHat* (2009).

[18] EVTIMOV, I., EYKHOLT, K., FERNANDES, E., KOHNO, T., LI, B., PRAKASH, A., RAHMATI, A., AND SONG, D. Robust physical-world attacks on machine learning models. *arXiv abs/1707.08945* (2017).

[19] FARRELL, J., AND BARTH, M. *The global positioning system and inertial navigation*, vol. 61. McGraw-Hill New York, NY, USA:, 1999.

[20] FARSHTEINDIKER, B., HASIDIM, N., GROSZ, A., AND OREN, Y. How to phone home with someone else's phone: Information exfiltration using intentional sound noise on gyroscopic sensors. In *WOOT* (2016).

[21] GALBALLY, J., CAPPELLI, R., LUMINI, A., MALTONI, D., AND FIERREZ, J. Fake fingertip generation from a minutiae template. In *ICPR* (2008).

[22] HE, W., WEI, J., CHEN, X., CARLINI, N., AND SONG, D. Adversarial example defenses: Ensembles of weak defenses are not strong. *arXiv abs/1706.04701* (2017).

[23] HOFMANN-WELLENHOF, B., LICHTENEGGER, H., AND WASLE, E. *GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer Science & Business Media, 2007.

[24] HU, L., AND EVANS, D. Using directional antennas to prevent wormhole attacks. In *NDSS* (2004).

[25] HUANG, L., AND YANG, Q. Low-Cost GPS Simulator GPS Spoofing by SDR. DEFCON, 2015.

[26] HUMPHREYS, T. E., LEDVINA, B. M., PSIAKI, M. L., OHANLON, B. W., AND KINTNER JR, P. M. Assessing the spoofing threat: Development of a portable GPS civilian spoofer. In *ION GNSS* (2008).

[27] JANSEN, K., SCHÄFER, M., MOSER, D., LENDERS, V., PÖPPER, C., AND SCHMITT, J. Crowd-GPS-Sec: Leveraging crowdsourcing to detect and localize GPS spoofing attacks. In *IEEE SP* (2018).

[28] KERNS, A. J., SHEPARD, D. P., BHATTI, J. A., AND HUMPHREYS, T. E. Unmanned aircraft capture and control via GPS spoofing. *Journal of Field Robotics 31*, 4 (2014), 617–636.

[29] KUHN, M. G. An asymmetric security mechanism for navigation signals. In *Information Hiding* (2004).

[30] KUNE, D. F., BACKES, J., CLARK, S. S., KRAMER, D., REYNOLDS, M., FU, K., KIM, Y., AND XU, W. Ghost talk: Mitigating EMI signal injection attacks against analog sensors. In *IEEE SP* (2013).

[31] LAZOS, L., POOVENDRAN, R., AND ČAPKUN, S. Rope: robust position estimation in wireless sensor networks. In *IPSN* (2005).

[32] LEDVINA, B. M., BENCZE, W. J., GALUSHA, B., AND MILLER, I. An in-line anti-spoofing device for legacy civil GPS receivers. In *ION ITM* (2001).

[33] MADRY, A., MAKELOV, A., SCHMIDT, L., TSIPRAS, D., AND VLADU, A. Towards deep learning models resistant to adversarial attacks. *arXiv abs/1706.06083* (2017).

[34] MICHALEVSKY, Y., BONEH, D., AND NAKIBLY, G. Gyrophone: Recognizing speech from gyroscope signals. In *USENIX Security* (2014).

[35] MONTGOMERY, P. Y., HUMPHREYS, T. E., AND LEDVINA, B. M. Receiver-autonomous spoofing detection: Experimental results of a multi-antenna receiver defense against a portable civil GPS spoofer. In *ION ITM* (2009).

[36] MOSER, D., LEU, P., LENDERS, V., RANGANATHAN, A., RICCIATO, F., AND CAPKUN, S. Investigation of multi-device location spoofing attacks on air traffic control and possible countermeasures. In *MobiCom* (2016).

[37] MOTELLA, B., PINI, M., FANTINO, M., MULASSANO, P., NICOLA, M., FORTUNY-GUASCH, J., WILDEMEERSCH, M., AND SYMEONIDIS, D. Performance assessment of low cost GPS receivers under civilian spoofing attacks. In *NAVITEC* (2010).

[38] MUOIO, D. 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/`.

[39] NARAIN, S., VO-HUU, T. D., BLOCK, K., AND NOUBIR, G. Inferring user routes and locations using zero-permission mobile sensors. In *IEEE SP* (2016).

[40] NIELSEN, J., BROUMANDAN, A., AND LACHAPELLE, G. GNSS spoofing detection for single antenna handheld receivers. *Navigation 58*, 4 (2011), 335–344.

[41] NIGHSWANDER, T., LEDVINA, B., DIAMOND, J., BRUMLEY, R., AND BRUMLEY, D. GPS software attacks. In *CCS* (2012).

[42] NISTÉR, D., NARODITSKY, O., AND BERGEN, J. Visual odometry. In *CVPR* (2004).

[43] PAGET, C. Practical cellphone spying. DEFCON, 2010.

[44] PARK, Y.-S., SON, Y., SHIN, H., KIM, D., AND KIM, Y. This ain't your dose: Sensor spoofing attack on medical infusion pump. In *WOOT* (2016).

[45] POPPER, B. 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`.

[46] PSIAKI, M. L., AND HUMPHREYS, T. E. 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`.

[47] PSIAKI, M. L., POWELL, S. P., AND OHANLON, B. W. GNSS spoofing detection using high-frequency antenna motion and carrier-phase data. In *ION GNSS* (2013).

[48] RANGANATHAN, A., ÓLAFSDÓTTIR, H., AND CAPKUN, S. SPREE: A spoofing resistant GPS receiver. In *MobiCom* (2016).

[49] RASMUSSEN, K. B., AND CAPKUN, S. Realization of RF distance bounding. In *USENIX Security* (2010).

[50] SCHÄFER, M., LENDERS, V., AND SCHMITT, J. Secure track verification. In *IEEE SP* (2015).

[51] SHIN, H., KIM, D., KWON, Y., AND KIM, Y. Illusion and dazzle: Adversarial optical channel exploits against Lidars for automotive applications. In *CHES* (2017).

[52] SHIN, H., SON, Y., PARK, Y.-S., KWON, Y., AND KIM, Y. Sampling race: Bypassing timing-based analog active sensor spoofing detection on analog-digital systems. In *WOOT* (2016).

[53] SHOUKRY, Y., MARTIN, P., TABUADA, P., AND SRIVASTAVA, M. Non-invasive spoofing attacks for anti-lock braking systems. In *CHES* (2013).

[54] SON, Y., SHIN, H., KIM, D., PARK, Y.-S., NOH, J., CHOI, K., CHOI, J., KIM, Y., ET AL. Rocking drones with intentional sound noise on gyroscopic sensors. In *USENIX Security* (2015).

[55] TIPPENHAUER, N. O., PÖPPER, C., RASMUSSEN, K. B., AND CAPKUN, S. On the requirements for successful GPS spoofing attacks. In *CCS* (2011).

[56] TIPPENHAUER, N. O., RASMUSSEN, K. B., PÖPPER, C., AND ČAPKUN, S. Attacks on public WLAN-based positioning systems. In *MobiSys* (2009).

[57] TITTERTON, D., AND WESTON, J. L. *Strapdown inertial navigation technology*, vol. 17. IET, 2004.

[58] TRIPPEL, T., WEISSE, O., XU, W., HONEYMAN, P., AND FU, K. Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks. In *EuroS&P* (2017).

[59] VOLPE, J. Vulnerability assessment of the transportation infrastructure relying on the global positioning system. *Technical Report* (2001).

[60] WANG, G., WANG, B., WANG, T., NIKA, A., ZHENG, H., AND ZHAO, B. Y. Defending against sybil devices in crowdsourced mapping services. In *MobiSys* (2016).

[61] WANG, K., CHEN, S., AND PAN, A. Time and Position Spoofing with Open Source Projects. BlackHat, 2015.

[62] WARNER, J. S., AND JOHNSTON, R. G. A simple demonstration that the global positioning system (GPS) is vulnerable to spoofing. *Journal of Security Administration 25*, 2 (2002), 19–27.

[63] WARNER, J. S., AND JOHNSTON, R. G. GPS spoofing countermeasures. *Homeland Security Journal 25*, 2 (2003), 19–27.

[64] WESSON, K., ROTHLISBERGER, M., AND HUMPHREYS, T. Practical cryptographic civil GPS signal authentication. *Navigation 59*, 3 (2012), 177–193.

[65] WESSON, K. D., SHEPARD, D. P., BHATTI, J. A., AND HUMPHREYS, T. E. An evaluation of the vestigial signal defense for civil GPS anti-spoofing. In *ION GNSS* (2011).

[66] XU, W., EVANS, D., AND QI, Y. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv abs/1704.01155* (2017).

[67] XU, W., QI, Y., AND EVANS, D. Automatically evading classifiers. In *NDSS* (2016).

[68] YAN, C., XU, W., AND LIU, J. Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. In *DEFCON* (2016).

[69] ZAMIR, A. R., AND SHAH, M. Accurate image localization based on google maps street view. In *ECCV* (2010).

[70] ZANDBERGEN, P. A. Accuracy of iPhone locations: A comparison of assisted GPS, WiFi and cellular positioning. *Transactions in GIS 13*, s1 (2009), 5–25.

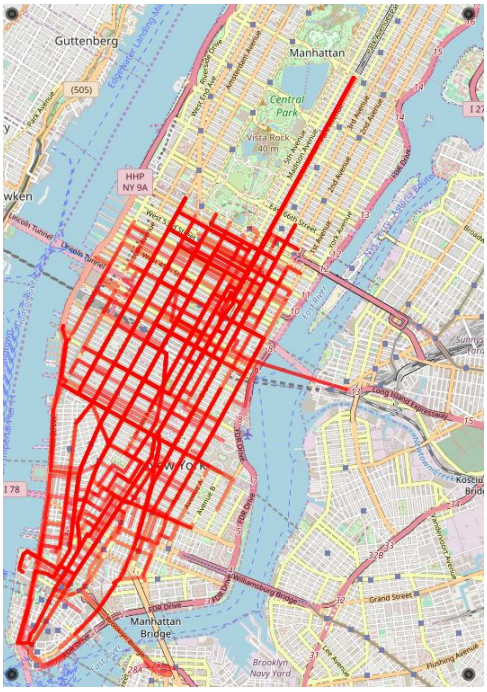
[71] ZENG, K. C., SHU, Y., LIU, S., DOU, Y., AND YANG, Y. A practical GPS location spoofing attack in road navigation scenario. In *HotMobile Workshop* (2017).

[72] ZHANG, G., YAN, C., JI, X., ZHANG, T., ZHANG, T., AND XU, W. Dolphinattack: Inaudible voice commands. In *CCS* (2017).

[73] ZHANG, Z., TRINKLE, M., QIAN, L., AND LI, H. Quickest detection of GPS spoofing attack. In *MILCOM* (2012).

Appendix-A: Taxi Route Visualization

Figure 9 visualizes 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$\times$7.38 km with a latitude range (40.7003, 40.7959) and a longitude range (-74.0180, -73.9308). The considered experiment area in Boston is 8.52km$\times$10.60km with a latitude range (42.3134, 42.3885) and a longitude range (-71.1435, -71.0149). As shown in Figure 9, the taxi routes are concentrated in the downtown areas in both respective maps.



(a) 300 taxi routes in Manhattan.

(b) 300 taxi routes in Boston.

Figure 9: Visualization of taxi routes in Manhattan and Boston.

Appendix-B: 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 10 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.

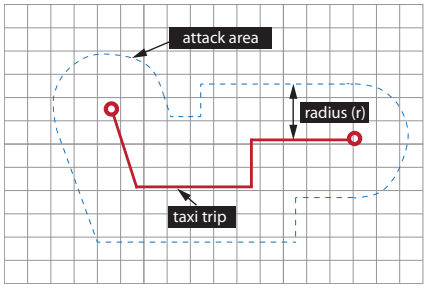


Figure 10: Illustration of the attack area and grids.