A Practical GPS Location Spoofing Attack in Road Navigation Scenario

Kexiong (Curtis) Zeng[1], Yuanchao Shu[2], Shinan Liu[3], Yanzhi Dou[1], Yaling Yang[1]
[1]Virginia Tech; [2]Microsoft Research; [3]University of Electronic Science and Technology of China
[1]{kexiong6, yzdou, yyang8}@vt.edu; [2]yuanchao.shu@microsoft.com; [3]liushinan63@163.com;

ABSTRACT

High value of GPS location information and easy availability of portable GPS signal spoofing devices incentivize attackers to launch GPS spoofing attacks against location-based applications. In this paper, we propose an attack model in road navigation scenario, and develop a complete framework to analyze, simulate and evaluate the spoofing attacks under practical constraints. To launch an attack, the framework first constructs a road network, and then searches for an attack route that smoothly diverts a victim without his awareness. In extensive data-driven simulations in College Point, New York City, we managed to navigate a victim to locations 1km away from his original destination.

CCS Concepts

•**Security and privacy → Mobile and wireless security;**

Keywords

GPS spoofing; Navigation; Route planning

1. INTRODUCTION

Location-based applications are rapidly becoming "killer applications" as location-enabled mobile devices proliferate [1]. In addition to an increasing number of GPS navigation users, location-based applications (e.g., Waze, Pokemon Go) and emerging self-driving cars rely heavily on GPS navigation. In Intelligent Transport Systems, vehicles broadcast their real-time locations along with other critical vehicle information to avoid collisions and congestions. Taxi-hailing applications like Uber and Didi also dispatch jobs, navigate drivers and calculate fares using GPS traces provided by in-vehicle mobile devices.

Along with the booming of GPS-based navigation, attackers have more incentives to launch GPS spoofing attacks. For example, through manipulating locations in navigation systems, attackers can divert valuable vehicles or target persons to unsafe areas [2, 3]. In 2012, Australian police rescued tourists who were directed by erroneous Apple Maps navigation application to a life-threatening desert with no water supply and extremely high temperature [4]. Missouri armed robbers trapped unwitting Pokemon Go players at isolated locations on their way navigated to Pokestops. Besides life-threatening issues, GPS spoofing attacks can also cause havoc to other location-based applications. For taxi-hailing applications, adversaries can cause chaos to job dispatch systems and manipulate drivers' taximeter by manipulating GPS locations of the targeted in-vehicle mobile devices. For crowdsourcing services, attackers can spoil the system by creating fake real-time events (e.g., traffic congestions) in the same way.

Launching GPS spoofing attacks on civilian GPS signals without cryptographic authentication mechanisms is not hard[1]. Due to the ubiquitousness of unencrypted GPS signals, GPS-based navigation systems are inherently vulnerable to signal spoofing attacks. Moreover, with the development of programmable radio platforms such as USRP, HackRF and bladeRF, it has become much easier to build low-cost portable GPS spoofers [5, 6, 7]. For example, WALB is a Raspberry Pi and HackRF based lunch box sized low-cost portable spoofer, which is able to achieve real-time GPS signal generation and location manipulation [8]. With these spoofers, researchers and hackers have extensively demonstrated successful GPS spoofing attacks on a variety of navigation systems for vehicles, drones, ships and wearable devices [9, 6, 7].

However, the majority of existing works are focused on GPS signal spoofing at physical layer. Without explicit goals, the attackers mostly demonstrate the capability of arbitrary location manipulation [5, 6, 7]. Moreover, there are no existing practical attack models in road navigation scenario. For example, in a yacht spoofing attack [2], what the attackers did is to simply spoof the yacht's GPS location 200m away from the navigation route. However, to smoothly deviate vehicles or people moving on roads in a real-time manner is far more complicated due to the constraints of road maps and variations of moving speeds.

To fill this gap, in this paper, we propose an attack model in road navigation scenario, and devise a GPS spoofing framework under practical constraints. The framework consists of three components: (1) road network construction by extracting and parsing public geographic data from OpenStreetMap, (2) attack route searching algorithm based on an input of an navigation route and corresponding constraints, (3) GPS spoofing attack simulator and performance evaluation tool.

Contributions of this paper include:

• A practical GPS spoofing attack model in road navigation scenario. To the best of our knowledge, this is the first GPS spoofing attack model with efficient attack route searching algorithm.

• A simulator based on real world road network to evaluate the proposed GPS spoofing attack. Valuable insights are provided by

HotMobile '17, February 21-22, 2017, Sonoma, CA, USA
© 2017 ACM. ISBN 978-1-4503-4907-9/17/02. . . $15.00
DOI: http://dx.doi.org/10.1145/3032970.3032983

[1]In this paper, GPS refers to civilian GPS by default.

extensive data-driven simulations in College Point, New York City. The results show that a victim can be finally diverted to a location around 1km away from his original destination.

• An implementation of a low-cost GPS spoofer using HackRF. Successful real-time GPS spoofing attacks are launched against various location-based applications on multiple devices.

2. GPS SPOOFING ATTACK

We first introduce the background of GPS spoofing attacks and then describe our attack model and problem formulation.

2.1 Background

Global Positioning System (GPS) is a global navigation satellite system that provides location and time information. It consists of 24 to 32 satellites in medium Earth orbit. These satellites are equipped with stable atomic clocks that are synchronized with one another. Each satellite continuously broadcasts GPS information on L1 C/A (1575.42 MHz) and L2 P/Y (1227.60MHz) band with 50bps data rate. GPS information includes satellite coordinates and message transmission time that can be used by receivers to calculate their longitude, latitude and altitude. While transmitting GPS signals on the same frequency, each satellite modulates signal with distinct CDMA code for receivers to distinguish individual satellites from each other. There are two different sets of CDMA codes – the precise (P(Y)) code and the coarse/acquisition (C/A) code. While P(Y) is used for authorized US military receivers only, C/A code is not encrypted for civilian access. As the vast majority of commercial GPS receivers, even those providing location and time information to critical infrastructures, are working on civilian GPS signals, spoofing attacks can be launched by transmitting counterfeit signals or replay prerecorded authentic signals.

2.2 Attack Model

We consider an external attacker who possesses a lunch box sized portable spoofer and launches GPS spoofing attacks remotely. He has knowledge about the victim's destination. The goal is to navigate the victim's vehicle or himself to unsafe areas by feeding false GPS locations to location-based applications. The attack scenarios include but not limited to (1) The attacker sets a destination in the application as a normal Uber passenger and launches attack from the back seat. (2) The attacker tailgates the victim by driving or walking. (3) The attacker can even control a spoofer-carrying drone to follow the victim.

Assume that an attacker starts to take over the victim's GPS from somewhere. He first creates a "ghost" on the map to misrepresent the victim's current location[2]. This way, the attacker fools the application to recalculate a ghost navigation route (e.g., from the "ghost's" location to the original destination). By following the ghost route, the victim will eventually travel along a different path and end up at another unexpected location.

To better illustrate the attack, we present a simple example from New York City. As shown in Figure 1(a), a person is driving from Lincoln Tunnel to New York University. Assume that an attacker takes over the victim's GPS receiver at the end of the tunnel. When the victim arrives at the black circle (Figure 1(b)), the attacker modified spoofed GPS signals to create a ghost position drifted to a nearby location (red circle). Hence, the navigation application will recalculate a new route (ghost navigation route) to the destination. Seemingly, as the victim follows navigation instructions, he perfectly follows the ghost navigation route and reaches the destination from the navigation application's perspective. However, the victim is diverted to traverse a different route (the black route) that ended up in another place. Similarly, the attacker can also place the ghost location on the original route, which is also able to divert the victim (as shown in Figure 1(c)).

[2]By manipulating the counterfeit GPS signals transmitted by the spoofer, the "ghost" can be anywhere under full control of the attacker.

2.3 Problem Formulation

Now we formulate the problem and propose the attack strategy design in this section. First of all, terminology and notations for our problem description are introduced. Table 1 lists the notations used in this section.

Symbol	Definition
G	Any given geographic area
R	Set of road segments in the geographic area
r_i	The ith road segment, $r_i \in R$
C	Set of connections in the geographic area
c_i	The ith connection, $c_i \in C$ and $c_i = (r_j, r_k) \vert r_j, r_k \in R$
L	The set of road segment length
l_i	The length of the ith road segment
Φ	The set of connection turn angle
ϕ_i or $\phi(r_j, r_k)$	The ith connection turn angle
P, D, Γ	Starting point, destination point and the corresponding navigation route
S_i	The ith longest continuous segment, $S_i \in \Gamma$
$\Gamma_o, \Gamma_g, \Gamma_v$	The original navigation route, the ghost navigation route, the diverted victim route
Loc_{ac}	The victim's actual current location
Loc_{gc}	The victim's ghost current location, which is under full control of the attacker
$\Omega_{driftDis}$	The upper bound of the instant drifted distance between Loc_{gc} and Loc_{ac}
Ω_{maxDis}	The maximum spoofed distance between Γ_g and Γ_v
Ω_{speed}	The speed scale factor that satisfies $(v_g - v_a)/v_a \leq \Omega_{speed}$
v_g, v_a	The ghost's speed that is under full control of the attacker, and the victim's actual speed

Table 1: Notation

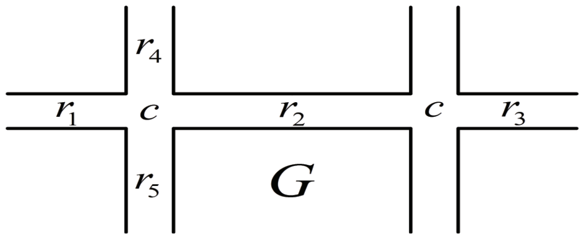
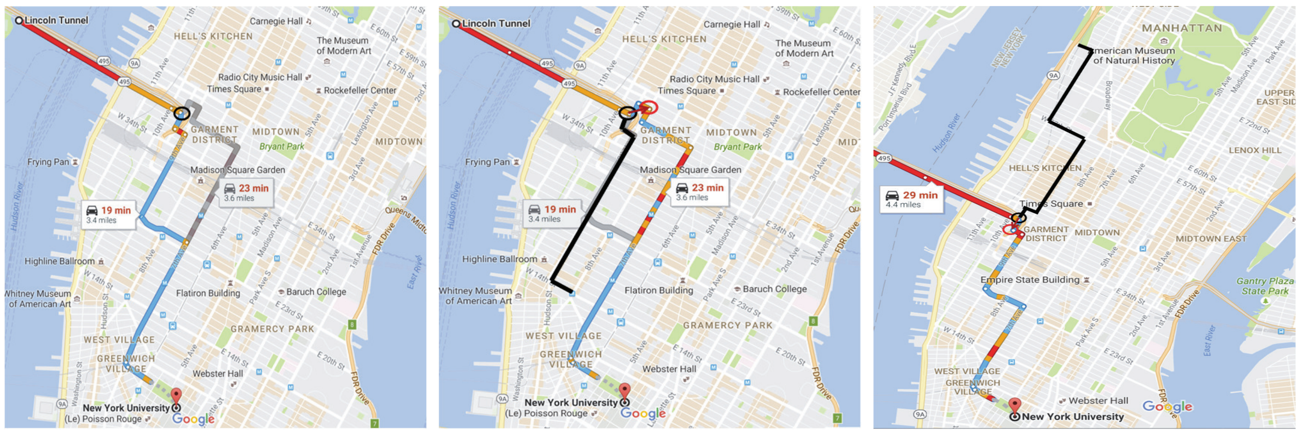


Figure 2: Illustration of a geographic area

As shown in Figure 2, for a given geographic area, it can be uniquely represented by a set of road segments and connections. Every road segment is the specific representation of a portion of a road with uniform characteristics, whose end points are determined by connections. Connections are created when a road intersects with other roads, turns into a different direction and bifurcates, etc. Therefore, a geographic area G can be represented as $G = (R, C, L, \Phi)$, where R is the road segment set, $C = \{c_i = (r_j, r_k) \vert r_j, r_k \in R\}$ is the connection set and the ordered pair (r_j, r_k) represents the connection between two road segments if the connection exists. L stores the road segment length for R, such that $l_i = length(r_i)$. Φ stores the turn angle for C, such that



(a) Original navigation route (b) Off-route spoofed location (c) On-route spoofed location

Figure 1: An attack example

$\phi_i = angle(c_i)$ which is the angle between two corresponding vectors r_j and r_k. A counterclockwise convention is adopted for angle representation, such that $\phi_i > 0$ and $\phi_i < 0$ indicates a left turn and a right turn, respectively. The absolute value of ϕ_i does not exceed $180°$. All information for R, C, L and Φ are extracted from or calculated based on public geographic data.

As shown in Figure 1, given a starting point P (e.g., Lincoln Tunnel) and a destination D (e.g., New York University), a navigation route Γ (e.g., the blue route in Figure 1(a)) is usually the shortest path from P to D. In road segment representation, $\Gamma = (r_{a_1}, r_{a_2}, ..., r_{a_n})$, where $(r_{a_i}, r_{a_{i+1}}, ..., r_{a_{i+j}}) \in R$ and $(r_{a_i}, r_{a_{i+1}}) \in C$. However, in real world, the granularity of a navigation route identified by system or human is not at road segment level, but at longest continuous segment level. For example, if the turn angle at the connection $(r_{a_i}, r_{a_{i+1}})$ is below a certain threshold θ (e.g., $\theta = 10°$), these two road segments are considered continuous and often belong to the same road (in Figure 2, r_4, r_5 are continuous road segments and belong to the same road). Road navigation system tells people to keep on the road for a maximal distance before a turn is required. Therefore, such continuous road segments should be aggregated to one longest continuous segment $S = (r_{a_i}, r_{a_{i+1}}, ..., r_{a_{i+j}})$, where $\phi(r_{a_k}, r_{a_{k+1}}) \leq \theta, k = i, i+1, ..., i+j-1$, $\phi(r_{a_{i-1}}, r_{a_i}) > \theta$ and $\phi(r_{a_{i+j}}, r_{a_{i+j+1}}) > \theta$. In this way, the navigation route can be rewritten as $\Gamma = (S_1, S_2, ..., S_m)$. The turn angle between two adjacent longest continuous segments $\phi(S_i, S_{i+1})$ is defined as $\phi(r_{last}, r_{first})$, where r_{last} and r_{first} are the last road segment of S_i and the first road segment of S_{i+1}, respectively.

Consider a victim is following an original navigation route Γ_o to a destination D. At some point, an attacker launches location spoofing attack and spoofs the victim's actual current location Loc_{ac} (e.g., the black circle) to a nearby location called ghost current location Loc_{gc} (e.g., the red circle). This will fool the navigation system and the victim to find a new navigation route from Loc_{gc} to D, which is called ghost navigation route $\Gamma_g = (S_{g_1}, S_{g_2}, ..., S_{g_m})$ (e.g., the blue routes) . Consequently, the victim will follow navigation instructions for Γ_g and could end up traversing a different victim route $\Gamma_v = (S_{v_1}, S_{v_2}, ..., S_{v_m})$ (the black routes), where Γ_v matches Γ_g in terms of navigation characteristics (e.g., travel distance and turn maneuvers). A formal definition of the attacker's objective is as follows.

Attack Objective O. Given a geographic area G, a victim's actual current location Loc_{ac} and his destination D, the attack A outputs all possible diverted victim routes along with associated ghost current location Loc_{gc} and ghost navigation route Γ_g. In mathematical representation, it is $O = A(G, D, Loc_{ac}) = \{o_1, o_2, ..., o_K\}$, where o_i is a three-element tuple $(\Gamma_v, \Gamma_g, Loc_{gc})_i$ and Γ_v matches Γ_g elementwisely.

In order to make the attack practical in road navigation scenario, constraints should be imposed by physical layer information, such as victim's intuition, road speed limit and possible countermeasures. Specifically, we define three constraints as follows.

Practical Constraints Ω. (1) Location drift constraint $\Omega_{driftDis}$ that is defined as the upper bound of the instant drifted distance between ghost current location Loc_{gc} and actual current location Loc_{ac}, i.e., $||Loc_{gc} - Loc_{ac}|| \leq \Omega_{driftDis}$. (2) Maximum spoofed distance constraint Ω_{maxDis} between ghost navigation route and victim route, i.e., $||S_{g_i} - S_{v_i}|| \leq \Omega_{maxDis}$ for $i = 1, 2, ..., m$. (3) Speed scale factor constraint Ω_{speed} that limits the ghost speed v_g within a reasonable factor of actual speed v_a, i.e., $(v_g - v_a)/v_a \leq \Omega_{speed}$.

We are very interested in statistical results of the attack success rate, the number of possible diverted victim routes and final diverted destinations under such constraints.

3. DESIGN

Our design consists of the four components as follows. (1) *Initialization:* For any target geographic area, the road network is constructed by extracting and parsing the information from some public database. The initialization is a one-time operation and can be reused for unlimited times as long as the geographic information stays the same. (2) *Attack inputs and settings:* Input the required information (e.g., the victim's starting point P and the destination point D) and adjust the settings (e.g., the practical constraints Ω and the attacker's interested area) in a specific attack. (3) *Search:* Based on previous attack inputs and settings, the search algorithm is run for the target geographic area. (4) The final results are a list of possible attack-launching positions along with their corresponding attack objective O.

3.1 Road Network Construction

The raw geographic data downloaded from OpenStreetMap database is called OSM XML file, whose basic data structure consists of node, way and relation that are used to represent various physical features (e.g., roads, buildings, boundaries, etc.). What we are interested is the road network for a given area. Hence, we parse the raw OpenStreetMap data and construct a directed graph $G = (V, E)$ to represent the road network. Every node $v \in V$ is a location node with unique ID, coordinates and the associated road ID. Adjacent nodes are connected by a directed edge $e \in E$. We find the intersection nodes in the graph to determine road segments and calculate turn angles based on their endpoints' coordinates. Given any starting point and destination point, the navigation route Γ can be obtained by either offline calculation using Dijkstra algorithm or online query through some API. As presented in Section 2.3, a navigation route should be represented in longest continuous segment form as $\Gamma = (S_1, S_2, ..., S_m)$.

3.2 Attack Algorithm Design

Given a graph G, a victim's actual current location Loc_{ac} where the attacker wants to begin spoofing, the victim's destination D and along with practical constraints Ω, the attack algorithm A first begins his search by picking up a ghost current location Loc_{gc} within the distance bound $\Omega_{driftDis}$ from Loc_{ac}. Then, a ghost navigation route $\Gamma_g = (S_{g_1}, S_{g_2}, ..., S_{g_m})$ from Loc_{gc} to D is calculated. In order to find any possible victim routes originated from Loc_{ac}, a search is performed by traversing the graph from Loc_{ac} and keeping candidate routes that have passed the search criteria at every step. Consider that the $\Gamma_v = (S_{v_1}, S_{v_2}, ..., S_{v_m})$ and $\Gamma_g = (S_{g_1}, S_{g_2}, ..., S_{g_m})$ should match elementwisely under constraints Ω, three criteria have been developed. The first criterion is *segment length match*. Given a speed scale factor constraint Ω_{speed}, during the same period, the distance the ghost can travel is within $(1 \pm \Omega_{speed})$ times the victim's actual travel distance. Therefore, $(1 - \Omega_{speed}) \cdot S_{v_i} \leq S_{g_i} \leq (1 + \Omega_{speed}) \cdot S_{v_i}$, where $i = 1, 2, ..., m$. The second criterion is *maneuver instruction match*. Since navigation systems give turn-by-turn maneuver instructions (e.g., keep on, turn left/right, U-turn left/right, etc.), the ghost navigation maneuver instructions should also work out on the victim route, otherwise the attack will fail due to a conflict between navigation instruction and physical environment (e.g., turn instructions vs. a deadend). So, it is required that $\phi(S_{v_i}, S_{v_{i+1}})$ and $\phi(S_{g_i}, S_{g_{i+1}}) \in$ same maneuver instruction category, where $i = 1, 2, ..., m - 1$. The third criterion is *maximum spoofed distance constraint*. With this constraint, the algorithm has to filter out routes that do not satisfy the condition $||S_{g_i} - S_{v_i}|| \leq \Omega_{maxDis}$ for $i = 1, 2, ..., m$.

Based on the above discussions, our attack algorithm (Algorithm 1) performs a breadth-first-search and maintains a list of candidate victim routes that have passed the search criteria at every step. The final output is a list of tuples, i.e., $O = \{o_1, o_2, ..., o_K\}$ indicating possible victim routes, corresponding ghost current location and ghost navigation routes. In more details, the algorithm starts with attack input and initialization (line 1). Then, the preprocessing operation (line 2) determines N candidate ghost current locations, where N equals the number of road segments within the drift constraint. For every candidate ghost current location, we perform victim route search as follows (line 4-19). First of all, the navigation route Γ_g from the ghost current location to the destination is obtained through an API getNavigationRoute (line 4). This API can be acquired from the location-based applications (e.g., Uber, Waze, Google Maps, etc.) that the victims are actively using. Then, the search starting point Loc_{ac} and m containers $U_1, U_2, ..., U_m$ storing the candidate victim routes at each step are initialized (line 5-6). Next, originating from the actual current location, the algorithm performs an m-depth breadth-first search (line 7-19). At each step, the end point (v) of each candidate victim route (u) stored in last step (U_{j-1}) is obtained. Only the segments that start from the endpoint and have passed the criteria can be added to the candidate victim routes for current step (U_j). Note that if no qualified segments can be found at some intermediate step, the search terminates right away (line 8-10). Finally, the search results U_m (if there is any) are appended to output O. After all N candidate ghost current locations have been checked, O is returned.

```
**Input:** $G, D, Loc_{ac}, \Omega_{driftDis}, \Omega_{maxDis}, \Omega_{speed}$
**Output:** $O = \{o_1, o_2, ..., o_K\}$, $o_i = (\Gamma_v, \Gamma_g, Loc_{gc})_i$
 1: Initialization: $O \leftarrow \emptyset$,
 2: Preprocessing: Find all candidate ghost current locations
    $\{Loc_{gc_1}, Loc_{gc_2}, ..., Loc_{gc_N}\}$ within $\Omega_{driftDis}$ distance
    from $Loc_{ac}$
 3: for $i$ = 1 to $N$ do
 4:     $\Gamma_g = (S_{g_1}, S_{g_2}, ..., S_{g_m})$, where $\Gamma_g$ is obtained through an
        API getNavigationRoute$(G, Loc_{gc_i}, D)$
 5:     $U_0 = \{[r_{ac}]\}$, where $Loc_{ac} \in r_{ac}$
 6:     $U_1, U_2, ..., U_m \leftarrow \emptyset$
 7:     for $j$ = 1 to $m$ do
 8:         if $U_{j-1} == \emptyset$ then
 9:             break
10:         end if
11:         for $u \in U_{j-1}$ do
12:             $v \leftarrow u.endpoint$
13:             for $s \in$ segments with starting point of $v$ do
14:                 if $s$ has passed the search criteria then
15:                     Append $u.append(s)$ to $U_j$
16:                 end if
17:             end for
18:         end for
19:     end for
20:     if $U_m \neq \emptyset$ then
21:         Append $(U_m, \Gamma_g, Loc_{gc_i})$ to $O$
22:     end if
23: end for
24: return $O$
```
Algorithm 1: Attack algorithm

The complexity of the proposed algorithm is worth an analysis. For time complexity, given the tremendous computing power of the commercial servers, the navigation route query can be responded in real time. For a small $\Omega_{driftDis}$, the number of candidate ghost current locations N is limited (e.g., in a dense grid city plan, N is usually around 16 for $\Omega_{driftDis} = 250m$). In each round for a specific candidate ghost current location, there can be as many as a^m (a is the number of adjacent intersection nodes of a search starting point) final victim routes if we do brute force enumeration. However, thanks to the search filter at each step, the number of candidate victim routes is often less than a small constant and the search usually terminates prematurely because no further matched segment can be found. Therefore, the search approximately runs in linear time $O(m)$ (e.g., the average search time in our experiment is around 10ms in MATLAB implementation). All in all, the algorithm runs in subsecond-level real time. Besides, an $O(N+m)$ space complexity indicates negligible memory overhead.

4. IMPLEMENTATION AND EVALUATION

In this section, we first implemented a spoofer on HackRF One and demonstrated successful takeover against various location-based applications on multiple devices. Then, we evaluated the proposed attack by data-driven simulations. The attack routes generated in simulations were in turn fed to the spoofer to launch real attacks against applications.

4.1 GPS Spoofing Attack Implementation

In order to understand our target location-based applications' practical behavior under GPS spoofing attacks, we implemented a low-cost spoofer on HackRF One platform. The HackRF One has a frequency range of 1MHz-6GHz and firmware version of 2015.07.2. The antenna has an SMA interface, 12dBi gain and frequency range of 700MHZ-2700MHZ, which covers the civilian GPS band L1 (1575.42MHz). The target devices are an iPhone 6 (with iOS 10.0.2) and a Samsung S7 Edge (with Android 6.0). The host laptop has 2.50GHz Intel quad-core i7-4710MQ CPU, 4GB RAM and runs on Ubuntu 14.04 LTS (64bit). Our GPS spoofer software is based on an open source Software-Defined GPS Signal Simulator [10]. The spoofed location can be set up by command line or a csv file containing the trajectory in Earth Centered Earth Fixed (ECEF) coordinate with a 10Hz update frequency.

In order to avoid interfering with legal GPS signals, we conduct all experiments in indoor environment with no one around. The distance between the spoofer and the target is more than 15m. With this spoofer, we have successfully launched real-time attacks against various location-based applications running on both devices with/without cellular and WiFi, such as Google Maps, Apple Maps, PokeMon Go, Waze, Uber, Lyft and Didi, etc. We have also created a ghost to travel some sample ghost navigation routes generated in our simulation, such that navigation applications give instructions completely based on the ghost movement. Using uptodate almanac and ephemeris data, the takeover time is usually around a typical warm start time (45s), but sometimes varies to two or three minutes. An interesting observation is that when there is a discrepancy between GPS location and network (cellular, WiFi) location, most of the applications trust GPS locations (except that Didi sometimes reverts to network locations in our experiments). This is because most of the location-based applications calls GPS location API whenever GPS location is available for its highest accuracy.

4.2 Preliminary Results

For data-driven simulations, we selected a square area (1.98km $\times$ 1.98km) in College Point, New York City as the geographic area G. As shown in Figure 3, the lower left corner with coordinates (40.778750,-73.857270) and the upper right corner with coordinates (40.797400,-73.833750) of G are marked as stars. We simulated attacks on 1129 distinct original navigation routes defined by (D, Loc_{ac}), which are randomly picked from G. These navigation routes' distance range from 200m to 2000m. We conservatively set the location drift constraint $\Omega_{driftDis} = 250m$, which is only two or three block distance in G and can often happen due to natural multipath effects. Considering the limited input map size, the maximum spoofed distance constraint is set as $\Omega_{maxDis} = 1000m$, which is the maximum

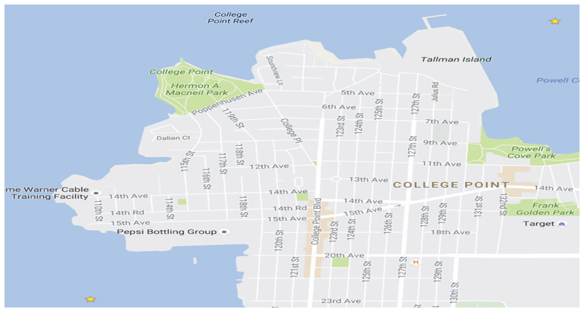


Figure 3: College Point, New York City on Google Maps

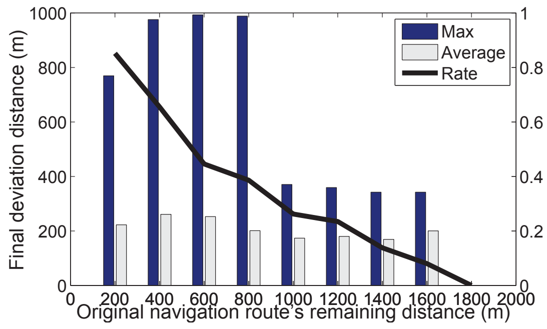


Figure 4: Preliminary results

spoofed distance measured based on a sophisticated software countermeasure SPREE [11]. Since the speed limit in G is 25 to 30 mph, the speed scale factor constraint is set as $\Omega_{speed} = 0.2$, which maps to a 5 or 6 mph speed offset that can hardly be noticed by a moving victim. Regarding maneuver instruction category, we refer to Waze's document [12] and define 5 categories in Table 2.

Turn angle	$[-10°, 10°]$	$\pm(10°, 170°]$	$\pm(170°, 180°]$
Instruction	Continue onto	left/right turn	left/right U-turn

Table 2: Maneuver instruction category

The preliminary results are presented in Figure 4, which are grouped by the original navigation route's remaining distance (i.e., from Loc_{ac} to D) with a 200m interval. For example, the bar at 200 is for all under-attack original navigation routes with remaining distance in $(200m, 400m]$. we present the final deviation distance and attack success rate by left and right y-axes, respectively. The final deviation distance is defined as the distance between the diverted victim routes' destinations and their original destinations. The attack success rate is defined as the ratio between the number of attacks that find at least one victim route and the number of total attack trials. As you can see, the attack success rate keeps decreasing as the original navigation route's remaining distance increases, which is intuitive because it is less likely to find a victim route that matches a long ghost navigation route. Additionally, no victim routes can be found when original navigation route's remaining distance $> 1800m$. Regarding the maximum final deviation distance, it first increases, reaches the maximum spoofed distance constraint and then decreases. This is because if the victim is attacked near the destination, the ghost navigation route is relatively short due to $\Omega_{driftDis}$ constraint, which is not able to divert the victim to a very far place. As the original route's remaining distance increases, the ghost navigation route becomes longer so that victim can be diverted to a further place. However, as the original navigation route's remaining distance keeps increasing, a victim route could exist only if the ghost navigation route and the original navigation route are parallel but differs in each segment's distance. Therefore, as the victim follows the ghost navigation route instructions, he will travel along the right direction but end up at some location that is near the original destination. Note that a deviation distance more than 975m can be obtained when original navigation route's remaining distance falls in $(400m, 1000m]$. This indicates that an attacker could launch an attack at somewhere that is $(400m, 1000m]$ navigation distance away from the destination, such that he might be able to divert a victim to somewhere 1km away from the original destination. This is more than enough for diverting to unsafe areas in cities, since the safety conditions can vary a lot even between neighboring communities. Moreover, if the maximum spoofed distance constraint can be relaxed and the input map size is large enough, a even larger final deviation distance can be obtained. Although no victim route can be found when the original route distance $> 1800m$, there are some routes that can partially match the ghost navigation route. Based on this, an advanced attacker can launch attack iteratively to find new victim routes when the previous partial victim route is about to end.

An important insight brought by our attack simulations is that some safety features should be added on current navigation systems. For example, candidate navigation routes can be ordered in terms of safety score, which is measured based on if the navigation route passes through any unsafe areas and if any victim routes exist when under attack. This could help prevent people from becoming victims of potential crimes.

5. RELATED WORK

GPS spoofing was first identified as a serious vulnerability in [13], which inspired the study of practical GPS spoofing attacks. Researchers and hackers have successfully spoofed GPS receivers of moving trucks, drones and smartphones, etc. with off-the-shelf GPS signal simulator or software defined radios [14, 6, 7]. Humphreys el al. have demonstrated seamless takeover GPS spoofing attack on a moving yacht with their portable receiver-spoofer [5]. [15] provided in-depth analysis about the requirements for such seamless takeover. However, no practical attack model for specific applications has been proposed.

Existing countermeasures for GPS spoofing attacks can be classified into two categories – *Infrastructure based* and *Infrastructure free*. The first category requires modifications on satellites (encryption), deployment of secure location verification infrastructures, or modifications on receiver antennas. The second category modifies software to run self-check algorithms or crosscheck with other location sources [16]. However, due to various practical limitations, no effective countermeasure has been implemented in civilian GPS receivers yet.

6. DISCUSSIONS AND FUTURE WORK

We first list several practical constraints that will influence the attack performance. First, the attacker might lose line of sight with the victim's receiver. However, in our experiments in an underground garage, it is shown that the takeover still works well when the the line of sight is blocked by a wall or cars. Second, it is much more difficult to find a valid ghost route with non-grid-like city plans. To overcome this issue, based on the basic attack algorithm that only creates a one-time location drift from Loc_{ac} to Loc_{gc}, an advanced attack algorithm that iteratively apply the basic attack and create multiple location drifts along the victim route can be devised. Such advanced attack can not only significantly improve attack success rate, but also improve the likelihood for diverting the victim to some specific location. Therefore, advanced attack gives the attacker more capability on diverted location control.

As one simple countermeasure, victims who are familiar with the area can manually crosscheck street signs and surrounding environmental information to avoid being navigated to random places. Also, crosschecking GPS location with network location (WiFi and cellular) could be an effective solution. However, network location is also vulnerable to jamming (e.g., all-in-one jammers [17]) and spoofing attacks (e.g., USRP forging WiFi access points [18] or cell towers [19]) as it is essentially based on unprotected wireless signals. As billions of unprotected civilian GPS receivers are being widely used in various systems, it also brings huge overhead to implement countermeasures that require infrastructure/hardware/software modifications. In order to circumvent the intrinsic vulnerability of unprotected wireless localization, robust location verification mechanisms can be developed by examining the correlations between location features (e.g., intersection, stop sign, traffic light and Google Street View etc.) and wired sensor hints (e.g., inertial sensor readings, camera images, etc.) [20].

7. CONCLUSION

In this paper, we propose a practical GPS spoofing attack in road navigation scenario. Successful real-time GPS spoofing attacks are launched against various location-based applications on multiple devices.

Acknowledgment

We are grateful to NSF for funding this research with grants CNS-1228903, CNS-1054697 and CNS 1547366.

8. REFERENCES

[1] Stefan Saroiu and Alec Wolman. Enabling new mobile applications with location proofs. In *ACM HotMobile*, 2009.
[2] Mark L. Psiaki and Todd E. Humphreys. Protecting GPS From Spoofers Is Critical to the Future of Navigation. IEEE Spectrum, 2016.
[3] Parmy Olson. Hacking A Phone's GPS May Have Just Got Easier. Forbes, 2016.
[4] Nick Hide. Apple Maps gets drivers lost in Australian outback, police warn. CNET, 2012.
[5] Todd E Humphreys, Brent M Ledvina, Mark L Psiaki, Brady W OH́anlon, and Paul M Kintner Jr. Assessing the spoofing threat: Development of a portable gps civilian spoofer. In *ION GNSS*, 2008.
[6] Ling Huang and Qing Yang. Low-Cost GPS Simulator âĂŞ GPS Spoofing by SDR. DEFCON, 2015.
[7] Kang Wang, Shuhua Chen, and Aimin Pan. Time and Position Spoofing with Open Source Projects. BlackHat, 2015.
[8] Keiichi Horiai and Kazuhisa Shirakami. WALB (WIRELESS ATTACK LAUNCH BOX). BlackHat, 2016.
[9] UT Austin Researchers Successfully Spoof an $80 million Yacht at Sea. UTNews, 2013.
[10] https://github.com/osqzss/gps-sdr-sim.
[11] Aanjhan Ranganathan, Hildur Ólafsdóttir, and Srdjan Capkun. Spree: A spoofing resistant gps receiver. In *ACM MobiCom*, 2016.
[12] https://wiki.waze.com/wiki/How_Waze_determines_turn_/_keep_/_exit_maneuvers.
[13] J Volpe. Vulnerability assessment of the transportation infrastructure relying on the global positioning system. 2001.
[14] Jon S Warner and Roger G Johnston. A simple demonstration that the global positioning system (gps) is vulnerable to spoofing. *Journal of Security Administration*, 2002.
[15] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. On the requirements for successful gps spoofing attacks. In *ACM CCS*, 2011.
[16] Desmond Schmidt, Kenneth Radke, Seyit Camtepe, Ernest Foo, and Michał Ren. A survey and analysis of the gnss spoofing threat and countermeasures. *ACM Computing Surveys (CSUR)*, 2016.
[17] http://www.cell-jammers.com.
[18] Nils Ole Tippenhauer, Kasper Bonne Rasmussen, Christina Pöpper, and Srdjan Čapkun. Attacks on public wlan-based positioning systems. In *ACM MobiSys*, 2009.
[19] Chris Paget. Practical Cellphone Spying. DEFCON, 2010.
[20] Kexiong Curtis Zeng, Yanzhi Dou, Yaling Yang, and Ranveer Chandra. Poster: Location verification and recovery for mobile in-vehicle applications. In *ACM MobiSys*, 2015.