Stars Can Tell: A Robust Method to Defend against GPS Spoofing Attacks using Off-the-shelf Chipset

Shinan Liu*, Xiang Cheng*†, Hanchao Yang†, Yuanchao Shu‡,
Xiaoran Weng§, Ping Guo¶, Kexiong (Curtis) Zeng††, Gang Wang‡‡,
Yaling Yang†
University of Chicago, †Virginia Tech, ‡Microsoft Research,
§University of Electronic Science and Technology of China,
¶City University of Hong Kong, ††Facebook, ‡‡University of Illinois at Urbana-Champaign
shinanliu@uchicago.edu, {xiangcheng, hcyang, yyang8}@vt.edu, yuanchao.shu@microsoft.com,
xiaoranweng@outlook.com, pingguo5-c@my.cityu.edu.hk, curtiszeng@fb.com, gangw@illinois.edu

Abstract

The GPS has empowered billions of users and various critical infrastructures with its positioning and time services. However, GPS spoofing attacks also become a growing threat to GPS-dependent systems. Existing detection methods either require expensive hardware modifications to current GPS devices or lack the basic robustness against sophisticated attacks, hurting their adoption and usage in practice.

In this paper, we propose a novel GPS spoofing detection framework that works with off-the-shelf GPS chipsets. Our basic idea is to rotate a one-side-blocked GPS receiver to derive the angle-of-arrival (AoAs) of received signals and compare them with the GPS constellation (consists of tens of GPS satellites). We first demonstrate the effectiveness of this idea by implementing a smartphone prototype and evaluating it against a real spoofer in various field experiments (in both open air and urban canyon environments). Our method achieves a high accuracy (95%–100%) in 5 seconds. Then we implement an adaptive attack, assuming the attacker becomes aware of our defense method and actively modulates the spoofing signals accordingly. We study this adaptive attack and propose enhancement methods (using the rotation speed as the "secret key") to fortify the defense. Further experiments are conducted to validate the effectiveness of the enhanced defense.

1 Introduction

The Global Positioning System (GPS) is a satellite-based system that provides geolocation and time information to GPS receivers anywhere on or near the Earth [5]. In addition to military usage, GPS also supports a wide range of civilian applications that require positioning services such as vehicle navigation, drone/boat operation, cargo tracking, and farm automation. Critical infrastructures such as cellular networks, financial systems, and power grids also rely on civilian GPS's time service to obtain globally synchronized time information.

Unfortunately, civilian GPS is known to be vulnerable to spoofing attacks [50, 53]. Adversaries can generate and transmit falsified GPS signals to take control of the victim's GPS device, producing the wrong location and time information to affect the dependent systems. Existing works have demonstrated GPS spoofing attacks in various applications, including diverting a luxury yacht from Monaco to Greece [1, 6], attacking the road navigation system [31, 56], and manipulating sensor-fusion algorithms on self-driving cars [44].

In recent years, there is a growing concern about GPS spoofing threat, considering the increasing number of devices (e.g., IoT devices, robots, autonomous vehicles) that are equipped with GPS sensors. Meanwhile, the software and hardware tools needed to launch the attack are becoming increasingly accessible. For example, software-defined radio platforms [10] have significantly reduced the cost of generating GPS signals. Recent studies show that a portable and programmable spoofer only costs about $200 [26, 56].

GPS Spoofing Defense. To address the threat of GPS spoofing, various solutions are proposed. Unfortunately, few are adopted in practice. Existing techniques either require significant modifications to the current GPS devices or need specialized hardware (i.e., *high deployment cost*), or are *not robust* against sophisticated attackers. For example, one solution is to introduce encryption and authentication mechanisms to civilian GPS [16, 42]. However, the estimated cost can be multi-billion dollars given the need to modify the satellites and existing GPS receivers. Alternatively, researchers have proposed to collect advanced measurement data about GPS signals to detect anomalies [25, 39, 40]. Due to the need for special hardware (e.g., antenna-array), these methods can only be realized on software-defined radio platforms or a limited set of GPS receivers equipped with enhanced chipsets.

To reduce the cost, other software-based methods aim to detect sudden changes of the GPS signals [14]. However, recent works show that advanced attackers can use a

*Both authors contributed equally to the project.

"smooth-takeover" method to avoid sudden signal changes during spoofing [14, 56]. Researchers also propose to cross-check GPS signals with other information sources such as WiFi/Cellular access points, and other Global Navigation Satellite System (GNSS) such as Galileo and GLONASS [13]. The problem is these alternative information channels can also be manipulated [47], and the ground infrastructures such as cellular towers are not dense enough for cross-validation.

Our Proposal. In this paper, we investigate new anti-spoofing techniques aiming to achieve both *high robustness* and *low cost*. We propose software-based methods to detect spoofing attacks that work for off-the-shelf GPS chipsets. The key idea is to measure and analyze GPS signals to derive the angle-of-arrival (AoA), based on the intuitions that attackers cannot (easily) emulate the physical angle-of-arrival of GPS signals from tens of GPS satellites around Earth simultaneously. Unlike traditional methods to derive AoA (which require expensive hardware such as large antenna-arrays [8, 25, 40]), our idea is to place a signal-blocking shield on one side of the GPS receiver while rotating the GPS receiver with the shield. Experimentally, we show that the physical rotation could simulate the effect of a directional antenna to estimate AoA for spoofing detection.

Based on these ideas, we first design defense methods by deriving and analyzing the AoAs across different GPS satellites. These methods are experimentally validated to be effective against *basic attackers* who are not aware of the presence of the defense. To explore to what extend the *adaptive attackers* can mimic the legitimate GPS signals when they are aware of our defenses, we further implemented an adaptive attack. We find that adaptive attackers can modulate the spoofing signals to eliminate many of the AoA artifacts. However, this adaptation is highly dependent on key information about the victim GPS device such as the rotation speed and the facing angle. Based on this observation, we then develop advanced defense methods by using the rotation speed and the facing angle as the "secret key". Fundamentally, the defender has full control of the rotation speed and can even change it in real-time. This makes the defense more robust because (1) the attacker has low visibility of the receiver's precise rotation speed and real-time facing angle, and (2) it is extremely difficult to adapt the spoofing modulation in real-time.

Implementation and Evaluation. We implemented our defense methods in a smartphone app. For the evaluation, we also built a programmable GPS spoofer using software-defined radios which supported both the basic attack and the adaptive attack. We performed real-world experiments with the spoofer and the prototype mobile app while complying with ethical and legal guidelines (see Section 6.2). We tested human body and metal sheet as the signal blocking materials (for different deployment scenarios), and confirmed that both materials are effective. Our experiments showed that the defense methods could detect the *basic* spoofing attacks with 100% accuracy within 5 seconds in "open air" and 20 seconds in "urban canyon", respectively. Against adaptive attackers, our advanced methods also demonstrated effectiveness (with slightly longer detection time) with detection accuracy of 95% in "open air" and 80% in "urban canyon".

Contributions: We make the following contributions:

- First, we proposed a new method for GPS spoofing detection that works on off-the-shelf GPS chipsets. The method leverages the idea of rotation and partial blockage to emulate the function of a directional antenna to facilitate spoofing detection.
- Second, we explored both basic attacks and adaptive attacks (i.e., adversaries are aware of our defense), and introduced additional measures to fortify the defense.
- Third, we implemented proposed methods (as a mobile app) and the adaptive attacks (using software-defined radios). Field experiments were conducted under various conditions to validate the effectiveness of our defenses.

To facilitate future research, we release code of our defense prototypes and analytical tools [1].

2 Background and Related Work

2.1 GPS Spoofing Attack

GPS is one of the Global Navigation Satellite Systems (GNSS). Today's GPS contains 31 satellites in medium Earth orbit, each equipped with a synchronized atomic clock. The satellites continuously broadcast GPS information using Coarse/Acquisition (C/A) code on the L1 band at 1575.42 MHz and encrypted precision (P/Y) code on the L2 band at 1227.60 MHz with 50 bps data rate. The GPS receiver can use the received information to calculate its own longitude, latitude, and altitude. Note that only authorized U.S. military receivers can use the P(Y) code. Civilian receivers can only get access to C/A code which is not encrypted.

Civilian GPS equipment is known to be vulnerable to spoofing attacks [24, 53]. In a spoofing attack, the attacker first lures the victim GPS receiver to migrate from the legitimate signal to the spoofing signal. This takeover phase can be either "brute-forced" or "smooth". In a brute-force attack, the false signals are transmitted at high power, causing the victim to lose track of the satellites and locking onto the stronger spoofing signals during the signal reacquisition process. Brute-force takeover is easy to implement but will cause abnormal jumps in the received signal strength or the computed clocks. In comparison, a smooth takeover is more stealthy. It begins by transmitting signals synchronized with the legitimate signals and then gradually overpowering the

[1] https://github.com/shinan6/robust-gps-antispoofing

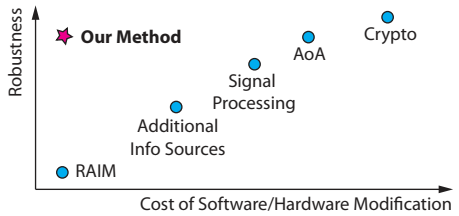


Figure 1: A comparison with existing anti-spoofing methods.

legitimate signals to cause the migration. Smooth takeover requires specialized hardware to perform real-time tracking and synchronization with the legitimate signals, which is more costly [18, 33]. Once the receiver locks on the spoofing signal, the attacker can manipulate the GPS receiver by either shifting the signals' arrival time or modifying the navigation messages [33, 34, 37].

Existing works have demonstrated GPS spoofing attacks in various applications. Examples include diverting a luxury yacht from Monaco to Greece [1, 6], misleading road navigation systems [31, 56], and manipulating sensor-fusion algorithms on self-driving cars [44]. In addition to location spoofing, the timing service provided by GPS is also vulnerable. For instance, in 2012, a single GPS spoofer manipulated timestamps of Phasor Measurement Units (PMU) in power grids [45]. In 2008, the London Stock Exchange lost 10 min of timing information due to a GPS jamming attack [53].

2.2 Existing Anti-spoofing Methods

We use Figure 1 to discuss existing GPS anti-spoofing methods from two dimensions: the cost of modifying existing software and hardware stacks in GPS equipment, and the robustness in detecting attacks.

Receiver Autonomous Integrity Monitoring. Receiver Autonomous Integrity Monitoring (RAIM) is designed for integrity checks on GPS signals. It handles non-adversarial errors caused by natural signal propagation disturbance such as ionospheric dispersion. However, RAIM cannot detect advanced GPS attacks (e.g., smooth takeover attacks) [37].

Checking Additional Info Sources. Researchers have proposed to cross-check GPS readings with additional information sources including Inertial Measurement Units (IMUs) [9, 17, 48] and Inertial Navigation System (INS) [49]. However, IMU and INS systems suffer from significant drift and deviation errors [7, 54] and hence are ineffective in detecting spoofing attacks that gradually deviate from true locations.

Other works propose to check external information offered by wireless infrastructures such as Network Time Protocol [12], Precision Time Protocol, WiFi, Cellular, Bluetooth, Bands of GPS L2 or L1 P(Y) [35], and other GNSS systems (Galileo, Beidou, GLONASS) [13]. However, not all the chipsets on commodity devices (e.g., smartphones) can receive multi-source information. Also, advanced adversaries can still launch attacks on these wireless channels to alter the location/timing information or simply jam channels [45–47]. For instance, for cross-constellation comparison based methods, multi-frequency, multi-constellation spoofers can overcome their defenses. Such spoofers can be realized in low-cost SDR [13] and are also commercialized [3]. Finally, many of these methods also require a dense deployment of the wireless infrastructures on the ground, which limits their coverage and usability in practice (e.g., in rural areas). Recent works propose specialized defenses for aircraft (or a group of coordinated aerial vehicles) by cross-checking the satellite imagery [55] or checking with other peers in the group [21, 22]. These methods are specialized for (and thus limited to) aircraft and/or multi-receivers.

Signal-processing-based Defenses. This line of defense aims to extract features from real and spoofed signals to detect spoofing. For example, one direction is to detect overpowered spoofing signals by examining Automatic Gain Control (AGC) and Carrier-to-Noise-Density (CN0) measurements [28, 41]. The Auxiliary peak tracking method tracks all GPS signals in the environment to detect incoherence between spoofed and legitimate signals [39]. The fingerprinting method [11] detects fingerprint differences between legitimate signals and spoofing signals.

While these approaches only need a single antenna, they must access low-level hardware information that is not traditionally accessible through software in GPS receivers. As such, these methods can only be realized on software-defined radio platforms or a limited set of GPS receivers equipped with enhanced GNSS chipsets (i.e., not widely deployable).

AoA-based Defenses. Angle of Arrival (AoA) based defenses leverage multi-receiver or specialized antennas (e.g., arrays, dual-polarization) to estimate the direction of GPS signals. These methods detect spoofing attacks by identifying abnormal AoA estimations [8, 25, 29, 40] or abnormal carrier phase changes during motion [38]. AoA is recognized as a robust defense method [30], but the high costs of specialized hardware become the barrier to their adoption in practice. For example, specialized lab-built antennas (such as GALANT [27] and Stanford PCB Dual Polarization Antenna [19, 25]) are not readily available. Similarly, the method described in [38] also requires special hardware (i.e., USRP and a patch antenna) to access highly accurate phase information. Thus, it is not supported by most off-the-shelf GPS chips. Commercial phased antenna arrays (with GPS band) could cost thousands or tens of thousands of dollars [4].

Cryptography-based Defenses. Crypto-based solution is to introduce encryption and authentication schemes to civilian GPS [16, 42, 52]. However, this is also the most costly approach (estimated cost of multi-billion dollars) since it demands changes in both the satellites and existing GPS receivers. More importantly, this approach is not backward-compatible with existing billions of GPS chipsets.

Our Method. We seek to design spoofing detection methods for GPS devices with off-the-shelf chipsets. The goal is to strike the balance between cost and robustness.

3 Threat Model

Before describing our defense methods, we first introduce the threat model. The goal of the attacker is to stealthily manipulate the location computation of a target GPS receiver (victim) by generating spoofing GPS signals. We assume the attacker owns a powerful state-of-the-art spoofer that can launch "smooth takeover" without causing anomalies during the takeover phase. Like most spoofing attacks, we assume the attacker has no physical access to the GPS receiver and cannot impose any physical alteration, hardware mounting, configuration change, or malware installation on the victim device. The attack can only be launched remotely by transmitting wireless signals on the GPS channel.

In this paper, we assume that the attacker uses a single spoofer to generate GPS signals for practical reasons. While multiple spoofers can generate signals from different angles, these spoofers will require specialized hardware to facilitate precise coordination [18]. Otherwise, the spoofing signals can be easily exposed due to a lack of synchronization. Increasing the number of spoofers will also make it harder to conceal the physical presence of the spoofers. While a multi-spoofer coordination attack is theoretically possible, we do not consider this setup in this paper.

Under this threat model, we consider two types of attacks.

- **Basic Attack:** We assume that the attacker is not aware of the presence of any defense method when launching the attack. We will design and evaluate our defense methods against this basic attack in Section 4, 5 and 6.
- **Adaptive Attack:** We assume the attacker is aware of our defense methods and tries to bypass them. We will describe the details of this adaptive attack in Section 7, and our designs to harden the detection methods in Section 8. Evaluation is presented in Section 9

4 GPS Spoofing Detection: Design Intuitions

We start by describing the key intuitions behind our defense methods. Among the defense methods shown in Figure 1, the Angle of Arrival (AoA) method is widely considered as a robust way to detect spoofed signals [37]. However, AoA measurement requires specialized hardware (e.g., antenna arrays) which incurs a high cost. Our idea is to conduct AoA measurements with off-the-shelf chipsets that are widely available on GPS devices such as smartphones. These chipsets usually only have an omnidirectional GPS antenna, making it challenging to derive AoA directly.

Rotational Blockage Effect. We solve the above problem based on an intuitive idea. Given a GPS receiver, if we place a signal-blocking material close to one side of the receiver, it in effect turns the receiver from omnidirectional to directional. An example radiation pattern is shown in Figure 21 in the Appendix. Considering the frequency bands of GPS signals (1.1 GHz ~1.6 GHz), it is easy to find signal blocking materials. For example, human body, a piece of foil paper, a metal plate, or a tin can are all qualified blocking materials.

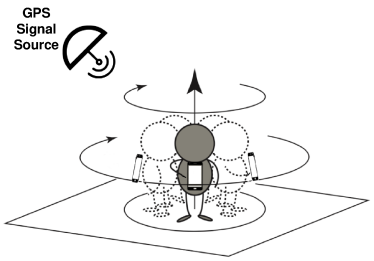
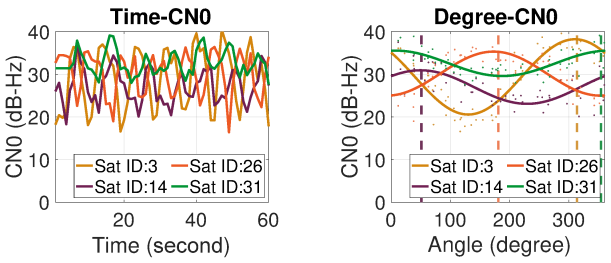


Figure 2: An illustration of how to create blockage effect using human body as an obstacle. GPS signal sources can be either legitimate satellites or a GPS spoofer.

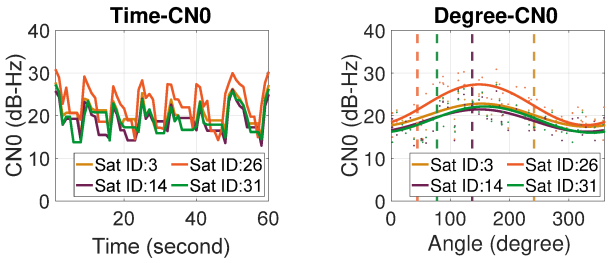
If we rotate the GPS receiver along with the blocking material, the received signal strength will fluctuate during the rotation process due to the different receiver gains at different signal arrival angles. The fluctuation patterns provide information regarding the AoA of the GPS signal.

Figure 2 illustrates an example of such a rotation mechanism. A person holds a smartphone (i.e., GPS receiver) over the chest and spins locally. During the spinning process, when the person along with the phone is facing a particular GPS signal source, this signal will be received without any blocking. When the person along with the phone is back facing the signal source, the human body will cause significant attenuation to the signal, which leads to a reduction in the received signal strength (RSS). By analyzing the fluctuation in RSS, we can estimate the AoA of the signal.

Spoofer Implementation and Experiment Setups. To demonstrate the rotational blockage effect, we implement the *basic* GPS spoofing attack by modifying open-source SDR-based GPS projects [10, 26, 36]. This spoofer contains four components: a HackRF One, a Raspberry Pi, a portable power bank, and an antenna. The size of the spoofer is small enough to be placed inside a lunchbox. HackRF One is a Software Defined Radio (SDR) platform, which is used to transmit the spoofing GPS signals. It comes with an SMA-interface omnidirectional antenna with a frequency range of 700–2700 MHZ that covers the civilian GPS band (1575.42 MHz). We use a 10000 mAh power bank as an energy source for the spoofer. A Raspberry Pi 3B (Quad-Core 1.2GHz Broadcom BCM2837 64bit CPU, 1GB RAM) runs our core software for the spoofing attack. This HackRF based spoofer is sufficient for our attack implementation and experiments. While a more sophisticated (and expensive) spoofer might make the takeover more seamless, it does not add much value



(a) Non-spoofing



(b) Basic spoofing attack

Figure 3: CN0 pattern under non-spoofing and *basic spoofing attacks*. Vertical dash lines represent AoEs (the ground-truth angle) of satellites at the time of experiments.

to our experiment since our detection scheme does not rely on take-over anomalies.

As a quick experiment, we set the spoofer 5 meters away from the target smartphone GPS receiver. To measure GPS signals, we developed a prototype Android app which continuously collects GNSS measurements as well as sensors data (accelerometer, gyroscope, and magnetometer). One researcher holds a XIAOMI MIX2 (Android 8.0, Snapdragon 835, supporting GPS L1 Band) over his chest and spins locally to collect GPS signals. We will provide more details for the smartphone prototype, experiment setups, and the ethical considerations of the experiments in Section 6.1.

Initial Measurements. Figure 3 illustrates the different characteristics between spoofing and non-spoofing cases. The results are collected from an open-space CN0 field measurement. We report carrier-to-noise-density ratio (CN0), which is the ratio of received signal power to noise density (a standard metric for signal strength). The unit of CN0 is dB-Hz, and a higher value often results in more precise positioning.

During the non-spoofing experiment, we collect GPS signals from 10 legitimate satellites. For simplicity, we plot Figure 3a using four satellites (ID: 3, 14, 26, 31) whose lines do not overlap with each other. The left figure shows CN0 over time, in which we see periodical changes of signal strength due to the rotation of GPS receiver and blockage. The right figure shows CN0 measurements over different rotation angles, which are derived from IMU sensors in the phone. The colored dots are measured CN0 values while solid curves are fitted curves of measurement results. Dashed vertical lines are the Angle of Ephemeris (AoE) of these GPS satellites, which correspond to the ground-truth angle of satellites and are publicly available at [32]. We observe that (1) different satellites are located at different directions with respect to the receiver; and (2) fitted curves reach the peak value when facing satellites. Also, the results confirm that the blocking effect exists across satellites despite different elevation angles. For instance, during the time of the experiment, satellites ID-3 and ID-14 had an elevation angle of 27.2 degrees and 69.0 degrees, respectively.

Figure 3b shows the results when the GPS signals are generated by a spoofer (basic attack), which have different patterns. We again select satellites whose lines are not completely overlapped. In the left figure, we observe that the spoofed signals from different "simulated satellites" are almost synchronized over time. In the right figure, we observe that the peak of the signal strength is not well aligned with the AoE of the real GPS satellites. Fundamentally, the spoofer is detectable because the diverse AoA of different satellites are difficult to simulate by a single spoofer, especially when the target GPS receiver is rotating (with blockage material) under an unknown/uneven rotational speed. In the following, we will develop spoofing detection methods based on the anomalies in the AoA measurements.

5 Detection Methods for Basic Attack

Based on these intuitions, we next introduce our defense methods against *basic attacks* (where the attacker is not aware of the presence of any defense). There are several challenges to address to detect GPS spoofing signals. First, we need to overcome the noisy CN0 measurements of GPS signals (particularly when there are signal reflections from the nearby environment). Second, the detection needs to be efficient, considering most off-the-shelf chips have a low refresh rate to measure CN0. Below, we introduce three methods with different design trade-offs, namely *AoA-Diff*, *AoA-Dev*, and *CN0-Corr*. Key notations are listed in Table 1.

5.1 AoA-Diff Detection

The most intuitive detection method is to compare GPS signals' Angle-of-Arrival (AoA) with the satellite ground-truth angles calculated from the Ephemeris Dataset (AoE). We called this method as AoA-Diff.

While intuitive, AoA-Diff has some practical challenges. First, it is difficult to always estimate AoA accurately in practice because GPS signals may be reflected by buildings and other surrounding surfaces. Second, to obtain the ground-truth AoE, the receiver needs to provide at least some coarse time and location in order to query the Ephemeris Dataset. While the time information can be obtained from the receiver's internal clock, the location information may be more

Symbol	Definition
G	The GNSS measurements that are being processed
T	A predefined threshold
N	Number of samples in the log file of G
M	Number of satellites
s_i	Satellite ID for the ith satellite
S	Set of satellite IDs, $S = \{s_i \mid i = 1,...,M\}$
AoA	Set of GPS signals' angle-of-arrivals (AoAs), obtained from our measurement algorithm: $AoA = \{aoa_{s_i} \mid s_i \in S\}$
AoE	Set of ground-truth satellite angles calculated from the Ephmeris dataset: $AoE = \{aoe_{s_i} \mid s_i \in S\}$
$CN0$	Carrier-to-Noise-density ratio of the GPS signal
C_{s_i}	Time sequence of CN0 measurements for satellite s_i, $C_{s_i} = [c1_{s_i}, c2_{s_i}, ..., cN_{s_i}]$
A	Time sequence of Azimuth of the GPS receiver, $A = [a_1, a_2, ..., a_N]$
R	Correlation matrix of CN0 of different satellites.
r	Combined correlation coefficient of CN0 sequence of all satellites.
δ_{AoA}	The standard deviation of AoA

Table 1: Notation and definition.

challenging to obtain (given the device is under a spoofing attack). We assume a coarse-grained location (e.g., at the city level) is available.

Considering these challenges, we only treat AoA-Diff as a naive baseline. More specifically, given a satellite s_i, we first compute its AoA (i.e., aoa_{s_i}) based on CN0 measurements and then query the ground-truth satellite angle aoe_{s_i}. We then put all the satellites' AoA and AoE into two separate vectors and calculate their Euclidean distance. If the difference is greater than a threshold T_{diff}, we determine the GPS receiver is under spoofing. Later in Section 6.3, we will evaluate the performance of AoA-Diff in comparison with other proposed methods.

5.2 AoA-Dev Detection

Considering the limitations of AoA-Diff, we next design an *AoA-Dev* method that does not require accurate AoA estimation or precise AoE as the ground-truth. AoA-Dev is short for "AoA standard deviations". The idea is based on the intuition that legitimate signals' AoAs from different satellites are more widespread compared with spoofed signals. Even if there are reflections by nearby objects and buildings, the spoofed signals (from a single spoofer) are likely to be reflected towards similar directions. As such, analyzing the deviations of AoA can overcome the influence of environmental signal reflections.

As shown in Algorithm 1, we first estimate the AoAs of the received signals in lines 2–7. Given a satellite s_i, we take the CN0 measurement sequence C_{s_i} and its corresponding receiver's orientation angles A. This creates a CN0-to-Angle scatter plot (similar to the right figures in Figure 3). We then fit these points into a *Sine* wave curve. We consider the peak

ALGORITHM 1: AoA-Dev Algorithm

Input: G, T_{dev}
Output: AoA, $SpoofFlag$, δ_{AoA}
1: Initialization: $AoA \leftarrow \emptyset$
2: Preprocessing: Obtain $S = \{s_1, s_2, ..., s_M\}$, $C_{s_i} = [c1_{s_i}, c2_{s_i}, ..., cN_{s_i}]$ and $A = [a_1, a_2, ..., a_N]$ from GNSS measurements G
3: **for** *each satellite* s_i **do**
4: Fit CN0-Angle sequence into sine wave curve: $SW_i = fit(A, C_{s_i})$
5: Get angle that resides peak of SW_i: $aoa_{s_i} = getPeakAngle(SW_i)$
6: Append aoa_{s_i} into set AoA: $AoA = append(AoA, aoa_{s_i})$
7: **end for**
8: Compute the mean of aoa_{s_i} in $[0, 2\pi)$: $\overline{AoA} = mean(AoA)$
9: Derive standard deviation: $\delta_{AoA} = \sqrt{\sum_{i=1}^{N} (aoa_{s_i} - \overline{AoA})^2 / (N-1)}$
10: **if** $\delta_{AoA} > T_{dev}$ **then**
11: $SpoofFlag = False$
12: **else**
13: $SpoofFlag = True$
14: **end if**
15: **return** AoA, $SpoofFlag$, δ_{AoA}

of the curve as the AoA of the GPS signal, denoted as aoa_{s_i}. Noted that the rotation angles A are measured in real time by the receiver's IMU sensors. Thus, our algorithm does not require the GPS receiver to rotate at a constant speed.

Given a set of estimated AoAs $\{aoa_{s_i}\}$, we then compute their standard deviation δ_{AoA} in line 8–9. Considering the elements in the set are angles, we need to compute the *circular* standard deviation [20]. For example, the difference between 1° and 359° should be 2° instead of 358°. Here, we map the elements in $\{aoa_{s_i}\}$ onto a unit circle and identify the minimal sized circular curve that covers all the AoAs in the set. Then we map each AoA to their corresponding position in $[0, 2\pi)$. After that, we can compute the standard deviation value, denoted as δ_{AoA}. If δ_{AoA} is below the threshold T_{dev}, we determine the receiver is under spoofing attacks.

Since AoA-Dev is based on the standard deviation of AoAs, it is less sensitive to the inaccuracy of AoA estimations. Also, using standard deviation makes AoA-Dev less sensitive to the sensor biases/noises that may affect the measured rotation angles (i.e., azimuth A).

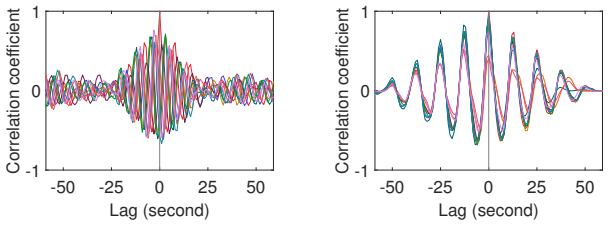
5.3 CN0-Corr Detection

The above method still needs to infer the AoAs of the received GPS signals, which requires the GPS receiver to rotate at least a full circle. The next method, called *CN0-Corr*, could potentially reduce the required CN0 measurements. CN0-Corr is based on the observation that CN0 measurements of spoofed signals from different satellites are more synchronized in time domain (see Figure 3b) compared to non-spoofing cases. We can capture this pattern by running a cross-correlation between the received signals as shown in Figure 4.

As shown in Algorithm 2, we compute the correlation coefficient of every pair of satellites' CN0 time sequences.

ALGORITHM 2: CN0-Corr Algorithm

Input: G, T_{corr}
Output: $SpoofFlag$
Initialization: $R \leftarrow \emptyset$, $timewindow = \{1, 2, ..., N\}$
Preprocessing: Obtain $S = \{s_1, s_2, ..., s_M\}, C_{s_i} = [c_{1s_i}, c_{2s_i}, ..., c_{Ns_i}]$ from GNSS measurements G
for $i, j <= M$ **do**
    Measure normalized cross correlation between s_i and s_j, $R_{i,j} = XCorr(C_{s_i}, C_{s_j}^T)$
end for
Calculate combined correlation coefficient, $r = \frac{1}{M}(\sum_{i=1}^{M}\sum_{j=1}^{M} R_{i,j} - \sum_{i=1}^{M} R_{i,i})$
if $r > T_{corr}$ **then**
    $SpoofFlag = True$
else
    $SpoofFlag = False$
end
return $SpoofFlag$



(a) Nonspoofing (b) Spoofing

Figure 4: Correlation coefficients between normalized CN0-Time sequences from different satellites in spoofing and non-spoofing scenarios. Lines of different colors represent different satellites. Note that 4b has the same number of lines as 4a but the lines are overlapped with each other.

Then we map the pair-wised values into a matrix R, where each element $R_{i,j}$ indicates the similarity between signal s_i and signal s_j. To estimate the overall similarity, we add up all the $R_{i,j}$ except for the diagonal element $R_{i,i}$, and normalize the summation by the number of satellite M to get the combined correlation coefficient r. If r is larger than the threshold T_{corr}, we regard the receiver is under spoofing attacks.

The amount of data required for CN0-Corr is small because this method does not require users to rotate one or multiple full circles to estimate the correlation. In addition, this method does not need sensors to report rotation angle, and thus is less susceptible to sensor noises.

6 Evaluation: Basic Spoofing Detection

To evaluate the proposed methods, we first implemented the prototype of the defense methods as a smartphone app. Then we used the app to perform real-world field experiments with the spoofer we built in Section 4.

6.1 Smartphone Prototype

We implemented the detection schemes in an Android app. The app is used as a proof-of-concept for evaluation — it is possible to implement the defenses in other ways (more discussions are in Section 10). We implemented the data collection and AoA analysis parts based on an open-source GnssLogger framework [15]. The app has been tested over multiple phone models, including Xiaomi MIX2 (Android 8.0, Snapdragon 835, supporting GPS L1 Band), Xiaomi MI8 (Android 8.1, Broadcom BCM47755 chip, supporting GPS L1 and L5 bands), and Xiaomi Redmi Note 7 (Android 9.0, Snapdragon 660, supporting GPS L1 Band).

The app collects both GNSS measurements from GPS sensors and position sensor data from the accelerometer, gyroscope and magnetic sensors using the system APIs. We first filter out invalid GNSS measurements by verifying their tracking state (i.e., the signals must be locked, and TOW decoded). Then we extract AoE and CN0 readings for each satellite. The phone's azimuth readings are derived from position sensor data and are paired with CN0 readings according to the timestamp.

6.2 Experiment Setup

In the field experiment, a victim phone was placed at a fixed location to perform rotation. The defense app was running on the phone to collect the GNSS measurements and azimuth reading. The rotation speed of the victim phone was about 12 seconds per rotation cycle (on average). Recall that our algorithms do not require the receiver to rotate at a constant speed, and thus the rotation speed does not need to be perfectly controlled. The rotation duration for each experiment was set to at least 30 seconds (rotating about 2.5 cycles). Empirically, this is more than sufficient for AoA analysis to converge—we rotated a bit longer than needed to gather extra data points to experiment with different parameters. The distance between the victim phone and the spoofer ranged from 1 to 15 meters, and the elevation angle of the spoofer ranged from 10 to 30 degrees. By default, the spoofer set the victim's spoofing location to a nearby town, which was about 11 km away from the true location.

Blocking Materials. To evaluate the impact of different blocking materials, we consider two different types:

- *Human Body*: The phone is held by a researcher in front of the researcher's chest. The human body is acting as the blocking material.
- *Metal*: We use a $33cm \times 36cm$ 176 layered aluminum foil sheet and attach it to one side of the phone.

The reason for testing metal-based shield is to set up the groundwork for implementing the defense for other systems beyond smartphones (e.g., automobiles and ships). The material should have enough blockage effect on GPS signals, and

should not affect the magnetic field if a compass is used to obtain orientation data. Aluminum foil satisfies both requirements. To make the metal spin together with the phone, we use a bookend as the support and place them together on a plastic turntable. A picture for the metal blockage setup is shown in Figure 20 in the Appendix.

In both cases, the back of the smartphone is attached to the shield with the screen facing out.

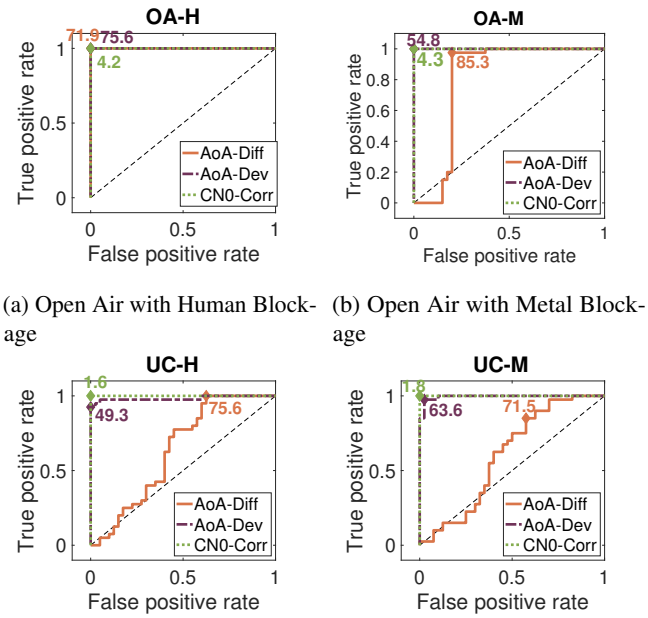
Testing Environments. To evaluate the methods in different scenarios, we have two testing environments:

- *Open Air (OA)*: An outdoor place with no building or obstacle nearby. The GPS signals are strong which are not blocked or reflected by surrounding buildings.
- *Urban Canyon (UC)*: An outdoor place surrounded by tall buildings, where the GPS signals are relatively weaker than those in open air. The signals are easily blocked or reflected by the wall.

Ethics and Legal Considerations. Considering the potential harms of GPS spoofing, *outdoor experiments with real spoofers* are strictly prohibited by the authors' institution as well as the FCC (Federal Communications Commission) regulations. To ensure the experiment ethics and legality, we have the following setups. First, only the *non-spoofing* experiments were conducted outdoor in the true "open air" and "urban canyon" environments. Second, for the spoofing experiments where we run the actual spoofer, we created *indoor* environments that have similar radio propagation features for "open air" and "urban canyon". More specifically, the spoofing experiments for open air setting took place in an anechoic chamber which is a room where RF absorbers are attached to the wall. These absorbers can significantly reduce the signal reflections from the environment, which enable us to simulate an open air environment far away from buildings [38, 51]. A picture of the chamber room is shown in Figure 19 in the Appendix. All spoofing experiments in the urban canyon setting were conducted in a large underground room with multiple large metal panels to emulate the strong multipath effect on GPS signals.

6.3 Evaluation Results

We conducted the experiments in four different settings: open air with a human body as the shield (OA-H), urban canyon with a human body as the shield (UC-H), open air with the metal shield (OA-M), and urban canyon with the metal shield (UC-M). In each setting, we collect data under 40 repeated non-spoofing experiments and 40 repeated simple-spoofing experiments. As stated above, the non-spoofing cases were set up in outdoor environments, and the spoofing cases were set up in lab-created indoor environments (for ethical and legal reasons).



(a) Open Air with Human Blockage

(b) Open Air with Metal Blockage

(c) Urban Canyon with Human Blockage

(d) Urban Canyon with Metal Blockage

Figure 5: ROC curve for the detection of basic spoofing attacks. The best performance points are marked out with a diamond sign. The corresponding threshold values are marked out in the figures.

Detection Accuracy. The main experiment results are presented in Figure 5. We plot the receiver operating characteristic (ROC) curve where the x-axis shows the false positive rate (FPR) and the y-axis shows the true positive rate (TPR). Here, we treat the spoofing cases as the "positive" cases. The ROC curve shows the trade-off between FPR and TPR under different threshold values of the detection algorithms.

First, we observe that AoA-Dev and CN0-Corr can accurately detect spoofing signals for all the four settings. Specifically, CN0-Corr can achieve a 1.0 true positive rate and 0 false positive rate in all settings. AoA-Dev can obtain the same performance in the open air environment, in Urban canyon environment, it can achieve a true positive rate of 0.90.

Second, we find AoA-Diff does not perform well, especially in the Urban Canyon (UC) environment. The problem is AoA-diff suffers from the multipath effect in Urban Canyon. More specifically, AoA-Diff requires accurately estimating the angle of arrival. In Urban Canyon, the GPS signals are either blocked or reflected by buildings. The multipath effect changes the legitimate GPS signals' AoAs, which leads to a higher false positive rate. On contrary, AoA-Dev and CN0-Corr rely on statistical comparisons among different satellites, which are more robust against the multipath effect.

Given the experiment results (and AoA-Diff's requirement for obtaining AoEs, see Section 5.1), we will no longer con-

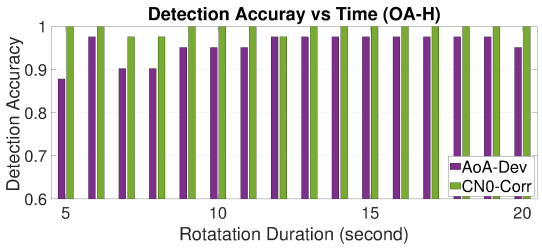


Figure 6: Spoofer detection accuracy (OA-H) within different rotation duration. Same configuration as that in Figure 5a.

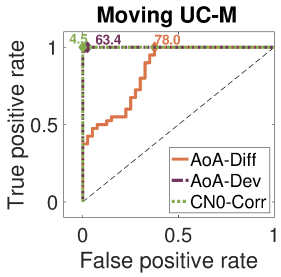


Figure 7: ROC curve for moving spoofer detection (UC-M).

sider AoA-Diff as a viable defense method. For the rest of the paper, we will focus on AoA-Dev and CN0-Corr.

Detection Speed. Next, we compare the detection speed of different algorithms. Given an experiment setting, we examine the detection accuracy by setting different data collection time periods. Due to the space limit, we only show the results for OA-H in Figure 6. The conclusion is the same for other settings. We find that both AoA-Dev and CN0-Corr can converge to a steady detection accuracy within 20 seconds. In particular, the CN0-Corr method can achieve a 100% accuracy within only 5 seconds. The reason is CN0-Corr directly computes the correlation of CN0 sequences, which only requires data from a partial rotation cycle.

Impact of Relative Movements. We run an additional experiment to examine the potential impact of relative movements between the spoofer and the victim. We take the UC-M setting, and dynamically change the distance between the spoofer and the victim phone during the experiment. We do so by fixing the victim location while moving the spoofer around the victim within 1–15 meters. We run 40 spoofing experiments and 40 non-spoofing experiments (30 seconds per experiment), and the detection results are shown in Figure 7. The results confirm that our methods remain effective. Intuitively, even when the spoofer is moving, the CN0 measurements from different spoofed signals are still synchronized with each other in both time and degree domains, which makes them detectable.

7 Adaptive Attack

So far, we show that our defense methods are effective on the basic attack. Next, we investigate the adaptive attack given the attacker is aware of the defense methods.

At the high level, the basic attack is detectable because the spoofed GPS signals have the same AoA. This not only allows us to detect the spoofing attack, but also potentially reveals the direction of the spoofer (to localize the spoofer or null the spoofing signal). To mitigate this artifact, the attacker can actively modulate the spoofing signals to mimic those of different satellites during the receiver rotation process, i.e., running an adaptive attack. Our threat model for the *adaptive attack* is similar to before: the attacker operates a single spoofer and has no direct access or visibility to the internal software/hardware of the GPS receiver.

Figure 8 gives an example of how the adaptive attack could modulate the spoofing signals to mimic the legitimate ones. The modulation requires knowing the GPS satellites' AoEs, the exact rotation speed, and the initial facing angle of the target GPS receiver. Among these parameters, the satellites' AoEs are easily available given they are public knowledge. However, the attacker will need to guess the rotation speed and the initial facing angle of GPS device. In practice, the defender can arbitrarily set the rotation speed and the initial facing angle. More importantly, the defender can even change the rotation speed in real-time. Given the attacker has no physical access to the GPS device, it is difficult for the attacker to know the precise speed of rotation and adapt the modulation in real time when the rotation speed changes. Essentially, the rotation speed can serve as a "secret key" that the attacker needs to guess.

Given a satellite, the attacker needs to first compute the shape of the CN0 curve (i.e., the *Sine* curve) based on the angle between the target GPS receiver and the satellite (AoE). The attacker needs to know the initial facing angle of the receiver to set the starting phase of the curve. Then the attacker sets the frequency of the curve based on the rotation speed. In addition to the phases, the power amplitudes of different satellites' signals should be different. A satellite with a higher elevation φ_{el} would have a lower amplitude variance (given the GPS receiver spins horizontally). This is because the shield will have a weaker blocking effect on their signals. Finally, the distance between the satellite and the receiver also matters. A higher distance leads to a lower peak CN0 value.

With the above consideration, an attacker can mimic the GPS signal for satellite s_i. We denote this spoofed signal's strength as $S_i(t)$ which needs to be changed with t during the rotation process. More specifically:

$$S_i(t) = [A_i \cdot cos(\omega_i \cdot t + \gamma_i) + D_i]L_i(t), \quad (1)$$

where $L_i(t)$ is the basic spoofing signal strength. Symbols ω_i, γ_i, A_i and D_i are the frequency, phase, amplitude, and the mean signal strength that the adaptive spoofer uses to control

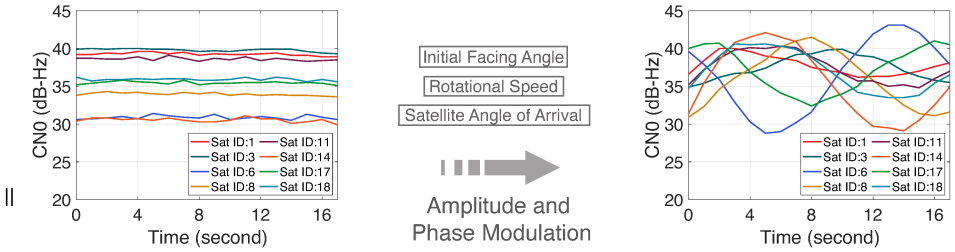


Figure 8: An example of the adaptive spoofing attack. To produce the modulated spoofing signals, the amplitude and phase are altered through time based on the attacker's knowledge of the receiver's initial facing angle, rotational speed, and also satellite angle of arrival (AoE). Each line represents the spoofed signal from one satellite.

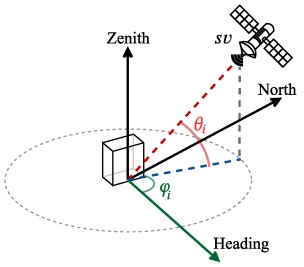
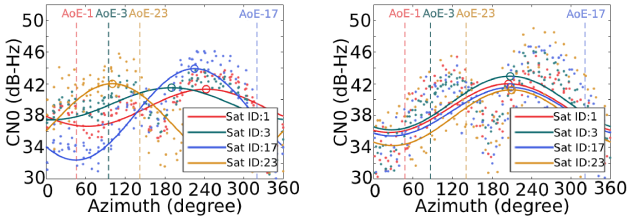


Figure 9: Illustration of angle of arrival of GPS signal.

the spoofing signal patterns. To mimic the legitimate signals, the spoofer can make A_i, γ_i, and D_i functions of satellite elevation, azimuth angle, and distance to the GPS receiver.

We implement this adaptive GPS spoofing attack by modifying the software prototype from Section 4. Instead of generating signals with constant power, the adaptive spoofer changes the signal power in real time according to pre-specified initial facing angle, rotational speed, and the satellites' positions. It renews power every 0.02 seconds (higher than the GPS receiver's 10Hz sampling rate). The goal is to make sure the received signals by the GPS receiver remain smooth without abrupt transitions.



(a) Correct guess of rotation speed and initial phase.

(b) Incorrect guess of rotation speed and initial phase.

Figure 10: CN0-Azimuth signal patterns for adaptive attack aggregated over multiple rotation cycles.

8 Detection Methods for Adaptive Attack

To detect the adaptive attack, in this section, we develop *advanced* detection methods based on the observation that the rotation speed of a GPS receiver is inherently a secret that can be dynamically changed by the defender. Incorrect guesses of the rotation speed or the receiver's real-time facing angle could lead to inauthentic signal patterns. Based on this intuition, we designed two methods to derive signals' AoAs from CN0 measurements to detect the adaptive attack.

8.1 Method 1: Average over Rotation Cycles

The first method is to simply take the CNO measurements at different rotation angles and average them over *a large number of rotation rounds*. As shown in Figure 10b, the spoofing signals from the adaptive spoofer will be eventually "synchronized", leading to similar fitting curves. This might be counter-intuitive considering the adaptive attacker is already simulating different spoofing signals for different satellites (e.g., with different amplitude and phase modulations). Below, we first explain the insights from the high-level and then demonstrate the reasoning mathematically.

High-level Intuitions. Recall that we assume the attacker needs to guess the rotation speed of the GPS receiver to modulate the GPS signal for each satellite. An incorrect guess means the modulated signal will not fully synchronize with the rotation process. The received GPS signal is essentially the product of the rotation effect and the attacker's modulation. For a given angle, we can measure the GPS signal strength (CN0) during each rotation round. Because the modulation is not fully synchronized with the rotation, at each round, the receiver will end up sampling at (slightly) different phases of the modulated signal. As a result, after sampling over a large number of rotation rounds, the modulation effect will be canceled out. In this case, the final fitting curve will be dominated by the frequency of the rotation, which is the same curve for all the satellites. By examining the final curves, we can reveal the true AoA of the spoofing signals.

Figure 10 shows an example. Figure 10b shows that if the attacker incorrectly guesses the rotation speed, the final fittings of different satellites will be "synchronized" due to the phase cancellation over a large number of rounds. Even though the CN0 measurements are not in a perfect *Sine* shape, using *Sine* curves to fit these dots will get synchronized results. Figure 10a shows an unrealistic case where the attacker knows the exact rotating speed and the initial facing phase. The attacker modulates the GPS signals that perfectly synchronizes with the rotation. In this case, the CN0 measurement at each angle will always be sampled from a particular modulated phase. Without the cancellation effect, the final fitting

curves will be different for each satellite (like legitimate GPS signals).

Mathematical Proof. We denote the GPS receiver's gain for a GPS signal s_i as $G_i(\theta_i, \varphi_i)$, where φ_i is the angle between orientation of the receiver and the satellite; θ_i is the satellite s_i's elevation angle in the sky as illustrated in Figure 9. Note that $G_i(\theta_i, \varphi_i)$ is a periodic function, which can be mathematically expressed as

$$G_i(\theta_i, \varphi_i) = G_i(\theta_i, \varphi_i + 2\pi k) \text{ for any integer } k \geq 0. \quad (2)$$

Consider that we rotate the GPS receiver horizontally so that only azimuth angle changes with time during the rotation process. If the GPS receiver receives legitimate GPS signals, the signal strength of a GPS signal i denoted as A_i, can be expressed as:

$$A_i(\Delta) = G_i(\theta_i, \varphi_i + \Delta)L_i, \quad (3)$$

where Δ is the change in rotation angle and is a function of time t. L_i is the GPS signal's strength at the receiver's position, and we assume L_i is stable during the rotation process. After n rounds of rotation, divide A_i measurements into n sections of length 2π based on the corresponding rotation angle. The average of A_i over a rotation angle Δ across all these sections, denoted as $\bar{A}_i$, has the following property:

$$\bar{A}_i(\Delta) = \frac{\sum_{k=0}^{n} A_i(\Delta + 2\pi k)}{n} = G_i(\theta_i, \varphi_i + \Delta)L_i, \forall \Delta \in [0, 2\pi), \quad (4)$$

It can be observed that for legitimate GPS signals, since a GPS satellite i's signal comes from a different angle compared with GPS satellite j's signal, φ_i are different from φ_j, which results in a different G_i variation pattern.

Now consider the case when the GPS receiver is rotating horizontally under adaptive attack by a single spoofer at a position θ. The ith spoofing signal's received signal strength, denoted as A'_i, can be expressed as:

$$A'_i(\Delta) = G_i(\theta, \varphi + \Delta)S_i(t_\Delta), \quad (5)$$

where $S_i(t)$ is the spoofing signal at time t and t_Δ is the time when the receiver rotates Δ angle. Note that since it is a single spoofer case, θ and φ are the same across all spoofed signals.

After n rounds of rotation, divide A'_i measurements into n sections of length 2π by the corresponding rotation angle. According to (1) and (5), the average of A'_i for a particular Δ across all these sections, denoted as $\bar{A}'_i(\Delta)$, can be expressed by:

$$\begin{aligned} \bar{A}'_i(\Delta) &= \frac{\sum_{k=0}^{n} A'_i(\Delta + 2\pi k)}{n} \quad (6) \\ &= \frac{G_i(\theta, \varphi + \Delta)A_i L_i}{n} \sum_{k=0}^{n} cos(\omega_i t_{\Delta + 2\pi k} + \gamma_i) \\ &\quad + G_i(\theta, \varphi + \Delta)D_i L_i, \forall \Delta \in [0, 2\pi) \quad (7) \end{aligned}$$

Consider the GPS receiver's rotation speed as υ, then $t_{\Delta+2\pi k} = \Delta/\upsilon + 2\pi k/\upsilon$. As long as ω_i/υ does not equal an integer, $\frac{\sum_{k=0}^{n} cos(\omega_i t_{\Delta+2\pi k} + \gamma_i)}{n} \overset{n\to\infty}{\approx} 0$. Thus, (7) can be approximated to:

$$\bar{A}'_i(\Delta) \overset{n\to\infty}{\approx} G_i(\theta, \varphi + \Delta)D_i L_i \quad (8)$$

Comparison of the mathematical expressions (4) and (8) reveals two facts. First, as long as the adversary cannot perfectly synchronize its modulation frequency ω with the true rotation speed of the GPS receiver, the mean received signal strength at a particular rotation angle (i.e., $\bar{A}'_i$) over a large enough number of rotation rounds (i.e., a large n) becomes independent of the spoofer's modulation on the phase and amplitude of the spoofed signal. Second, the variations of $\bar{A}'_i$ of different spoofed signals are highly synchronized because they have the same θ and φ. For legitimate GPS signals, the different satellite position results in a different φ_i, which leads to different variation patterns in $\bar{A}_i$ for different satellites.

Spoofing Detection. Based on the above reasoning, the detection method works as follows. We first map the CN0 measurements over multiple rotation rounds to the corresponding angles. Then we fit the *Sine* curve to derive AoAs (see Figure 10b). With AoAs, we can simply apply the AoA-Dev method developed in Section 5 for spoofing detection.

8.2 Method 2: Spectrum Analysis

The second method is to directly perform a spectrum analysis on the CN0 measurements. The intuition is that, given the attacker cannot perfectly guess the rotation speed and the initial facing angle of the GPS receiver, it means the modulated signal and the rotation will have two different frequencies. As a result, the received signal will be the product of these two, and thus exhibits *multiple peaks* in the spectrum domain.

More specifically, according to Equ. (5), the received signal A'_i is the multiplication of two signal $G_i(\theta, \varphi + \Delta)$ and $S_i(t_\Delta)$. Since our experiment results in Figure 3 has shown that CN0 measurements during rotation falls on a sinusoidal wave. We can approximate G_i by

$$G \approx M\cos(\upsilon t + \varphi_i) + C, \quad (9)$$

where υ is the rotation speed, φ_i is the angle between the orientation of the receiver and the satellite s_i. M is the amplitude and C is the mean of G. The value of M is set based on the material's blockage effect. The better the blockage effect is, the higher M will be. For legitimate signals, combining (9) with (1) and (3), we have:

$$A_i(\Delta) = L_i M[\cos(\upsilon t + \varphi_i) + C], \quad (10)$$

which has only one peak at frequency υ.

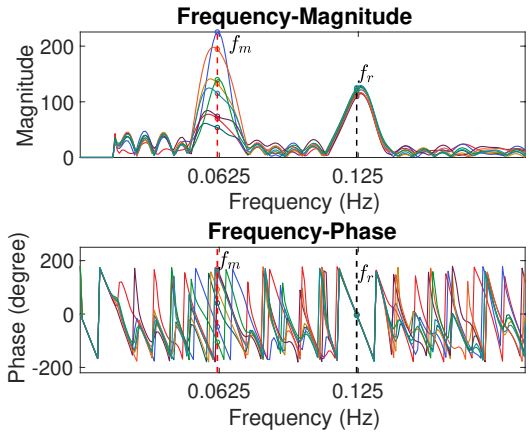


Figure 11: Spectrum analysis over spoofed signals. Each line represents the spoofed signal from one satellite.

For the adaptive spoofing signals, combining (9) with (1) and (5), we have

$$A_i'(\Delta) = (M\cos(\upsilon t+\varphi_i)+C)[A_i\cdot cos(\omega_i\cdot t+\gamma_i)+D_i]L_i(t)$$
$$= L_i\big[\frac{MA_i}{2}\cos((\upsilon+\omega_i)t+\varphi_i+\gamma_i)+\frac{MA_i}{2}\cos((\upsilon-\omega_i)t$$
$$+\varphi_i-\gamma_i)+MD_i\cos(\upsilon t+\varphi_i)+CA_i\cos(\omega_i t+\gamma_i)+CD_i\big] \quad (11)$$

The above expression reveals two insights. First, a spectrum analysis on A_i' will reveal four peaks at four frequencies: f_r+f_m, f_r-f_m, f_r and f_m (we denote the rotation frequency $f_r := \frac{\upsilon}{2\pi}$ and modulation frequency $f_m := \frac{\omega_i}{2\pi}$). Second, the spectrum analysis will also reveal phases at these four frequencies. Among them, phase φ at frequency f_r is especially critical since φ_i is the initial angle between the azimuth of the receiver and the satellite i. The AoA of the signal can be obtained by $a_1-\varphi_i$, where a_1 is the GPS receiver's initial facing angle recorded by the smartphone's IMU sensors.

Figure 11 shows an example of the spectrum analysis results (Spectrum Magnitude and Phase) on CN0 field measurements for an adaptive attack. In this example, the rotation frequency f_r is 0.125 Hz and the adversary's modulation frequency f_m is 0.0625 Hz. Since f_r-f_m and f_m both happen to be 0.0625 Hz, the peak at f_r-f_m overlaps with the peak at f_m in the figure. Another peak at frequency f_r is very visible in the figure. Note that the peak supposed to be present at f_r+f_m (0.1875 Hz) is not obvious because the coefficient $\frac{MA_i}{2}$ at f_r+f_m is approximately $\frac{1}{16}$ of the coefficient MD_i at f_r due to our A_i, D_i, M, C parameter settings in this experiment.

Nevertheless, the takeaway is that spoofing signals will produce multiple peaks in the spectrum domain in addition to the peak at the rotation frequency f_r. More importantly, Figure 11 shows the phases of different satellites' signals at the rotation frequency f_r are the same (i.e., φ), indicating that these signals share the same AoA (hence they are spoofed). We can use the initial facing angle (a_1) to obtain AoA as $a_1-\varphi_i$. Then we can apply the AoA-Dev method developed in Section 5 for spoofing detection. The detailed algorithm is shown in Algorithm 3 in the Appendix.

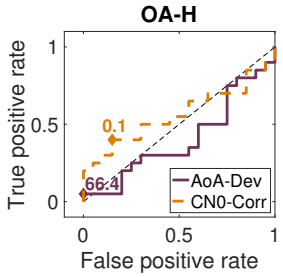


Figure 12: Applying basic detection methods on adaptive spoofing signals (OA-H) within 8 seconds. The threshold values for the best performing points are marked out.

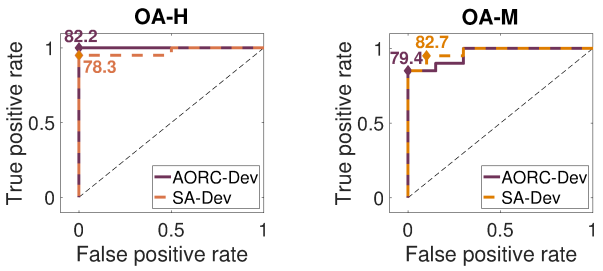
9 Evaluation: Adaptive Spoofing Detection

In the following, we first evaluate the adaptive spoofer against the basic defense methods developed in Section 5 to show the effectiveness of adaptive attacks. Then we apply the advanced defense methods proposed in Section 8 and examine their performance against adaptive attacks. If not otherwise stated, the rotation speed of the receiver is around 0.1 Hz. As mentioned before, we do not need to perfectly control the rotation speed since our detection algorithms do not depend on it. In this experiment, the *guessed* rotation speed by the adaptive spoofer is 0.125 Hz. We also feed the adaptive spoofer with the correct initial facing angle.

Adaptive Spoofer vs. Basic Detection. Recall that in Section 6.3, we have shown that the basic methods AoA-Dev and CN0-Corr can accurately detect the basic spoofing signals within one rotation cycle (8 seconds). Here, we further test AoA-Dev and CN0-Corr against the adaptive spoofers that customize the modulation of the signals for each satellite. The result is presented in Figure 12. Due to space limit, we only show the result for the OA-H setting. Other settings have similar outcomes and thus results are omitted for brevity.

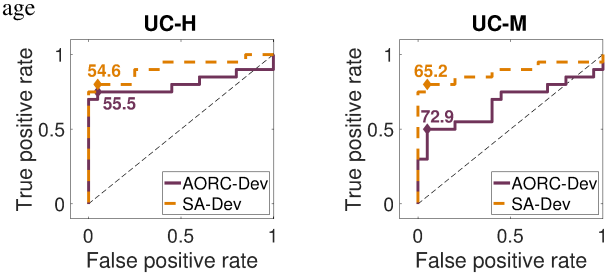
The result in Figure 12 confirms that the basic methods are no longer effective against the adaptive spoofer. The area under the ROC is close to 0.5, which means the detection is close to a random guess. This result confirms the effective implementation of adaptive spoofing.

Adaptive Spoofer vs. Advanced Detection. Next, we evaluate the advanced methods against adaptive spoofer. Recall both Average Over Rotation Cycles (AORC) and the Spectrum Analysis (SA) are used to estimate the spoofer's AoA. With the AoAs, we then run AoA-Dev to perform the detection. We call the two methods "AORC-Dev" and "SA-Dev" respectively. We perform the experiments under both open air (OA) and urban canyon (UC) environments using two different types of shields. Since the advanced methods, especially AORC, need more time to compute AoAs by



(a) Open air with Human blockage

(b) Open air with Metal blockage



(c) Urban Canyon with Human blockage

(d) Urban Canyon with Metal blockage

Figure 13: ROC curves for our countermeasures against adaptive spoofing under different environments. The threshold values for the best performing points are marked out.

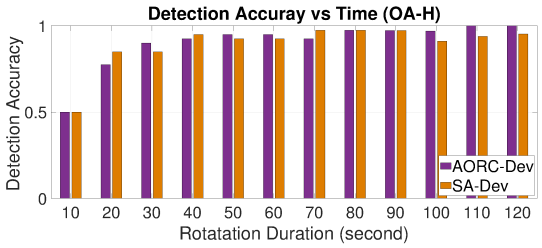
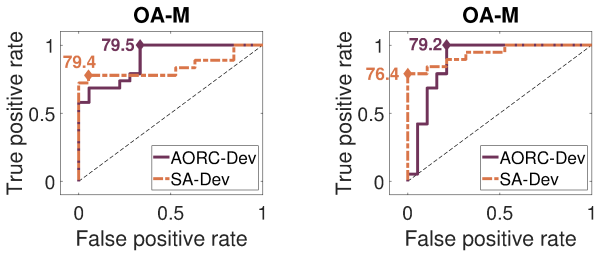


Figure 14: Detection accuracy (OA-H) within different durations. Same configuration as that of Figure 13a.

design, we run the measurements for at least 2 minutes per setting. The results are shown in Figure 13a.

We have two key observations. First, both AORC-Dev and SA-Dev methods work well in an open air (OA) environment. When the human body is used as the shield, AORC-Dev and SA-Dev can achieve true positive rates of 1.0 and 0.95, respectively, with 0 false positives. Comparing the two types of shields, we observe the metal sheet has a slight performance decline. This is likely due to the fact that the metal sheet is thinner and smaller than a human body.

Second, SA-Dev works better than AORC-Dev in the Urban Canyon (UC) environment. SA-Dev derives AoA from spectrum analysis based on phase, which is likely more robust against the multipath effect compared with the direct AoA estimation in the time domain (AORC-Dev). In the urban canyon, SA-Dev's performance is still acceptable with a true positive rate of 0.8 and a false positive rate of 0.05.



(a) $\Delta_f \in [0.01, 0.05)$

(b) $\Delta_f \in [0.05, 0.09)$

Figure 15: Impact of different guessing errors of the adaptive spoofer (Δ_f). The modulation frequency is 0.125 Hz in OA-M, 60 seconds of measurements. The threshold values for the best performing points are marked out.

Detection Speed. For AORC-Dev and SA-Dev methods, we further analyze their detection speed, by setting different measurement rotation duration. The results are shown in Figure 14. We find that both methods need about 70 seconds to converge to a steady accuracy. The time required to detect the adaptive attack is longer than that of the basic attack (in comparison with Figure 6). The reason is that we need to rotate the device for enough cycles to derive AoAs.

Sensitivity to Guessing Errors. Finally, we briefly evaluate the impact of the attacker's guessing errors. Recall that the attacker needs to guess the rotation speed of the GPS spoofer (even with the correct initial facing angle). Here, we examine the impact of guessing errors. Guessing error Δ_f is the difference between the GPS receiver's real rotation speed and the guessed one by the spoofer (measured in Hz). In this experiment, we configure the attacker-guessed value (modulation frequency) as 0.125 Hz. Then we change the rotation speed of the GPS spoofer dynamically. Figure 15 shows the impact of Δ_f on the detection performance. We find that when the guessing error is above 0.05 Hz, the detection accuracy remains high for both methods. Even if the attackers have guessed the rotation speed accurately (e.g., with an error between 0.01Hz – 0.05Hz), the detection performance only has a small decline. Overall, our detection methods are not very sensitive to the guessing errors of the adaptive spoofer.

10 Discussion

10.1 Spoofer Localization

Given our methods can provide a rough estimation of AoAs (both basic and advanced methods), the information can be further used to localize the spoofer. For example, the defender can conduct AoA measurements at two different locations and then perform triangulation to obtain the spoofer's location. However, this method may suffer from AoA estimation errors. Another idea is to perform AoA-guided navigation to locate the spoofer via multiple steps. Due to space limits, we

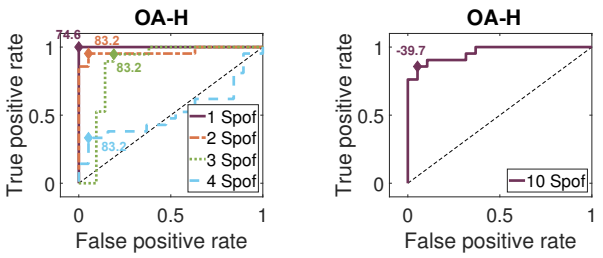
presented our experiments in Appendix A. The experiments shown promising results (e.g., we are able to localize the spoofer within 10 meters).

10.2 Multi-spoofer Scenario

Our threat model assumes the attacker has only one spoofer to transmit signals from one direction. For dedicated attackers, in theory, they can position multiple spoofers at *exactly the same angles* of all the available satellites (one spoofer per satellite), to potentially disrupt our defense methods. However, such an attack is difficult to realize in practice. First, the attacker needs to purchase a large number of spoofers (close to the number of available satellites). Second, all the spoofers need to be precisely synchronized (e.g., at the nanosecond level [2]) to avoid discrepancies in their signal time. Third, the spoofers also need to constantly adjust their positions to align the angles when the victim GPS device is on the move (which is expected during the navigation scenarios for vehicles, drones, and ships). This further complicates the attack given the difficulty of coordinating the precise movements of multiple (often more than 10) spoofers in real-time while ensuring they remain stealthy.

Attack Setup. To understand the multi-spoofer attack, we present a supplemental experiment. This experiment is based on a *trace-driven simulation* rather than real-world multi-spoofer deployments. This is again due to the ethical and legal constraints (as discussed in Section 6.2) that prevent us from running real spoofers in an open space. Our anechoic chamber is not big enough to support experimenting with multiple real spoofers (e.g., 10 spoofers). We relax some of the constraints and emulate a more practical multi-spoofer attack. We assume the attacker owns n spoofers. Instead of coordinating their precise positions and movements in real-time, we assume the attacker randomly position these spoofers on a circle around the target GPS device. We simulate this attack under the OA-H setting, based on the real-world GPS traces collected from both spoofing and non-spoofing experiments (Section 6). We keep the non-spoofing traces unchanged; For the spoofing traces, we shift the azimuth value (i.e., angle) of the single-directional spoofing signals to n random values, which creates/emulates n spoofers. We use the 30-second traces and evaluate the multi-spoofer attack against our detection methods.

Observations. We find that the multi-spoofer attack is indeed stronger than a single-spoofer attack. Figure 16a shows the performance of the AoA-Dev method. We observe that the AoA-Dev method can sustain at most $n = 3$ spoofers. When n is increased to 4, the detection accuracy is significantly decreased. This is expected, since AoA-Dev detects spoofing based on the standard deviation of AoAs of different satellites. When multiple spoofers are physically positioned at different



(a) AoA-Dev method (b) AoA-Combo method

Figure 16: Multi-spoofer simulation results under OA-H. "n Spof" means n spoofers are used in the simulation.

angles, the standard deviation of AoAs will be significantly increased (which misleads the detector).

In the meantime, we also evaluate the multi-spoofer attack against an improved version of AoA-Dev. The idea is to combine the AoA-Dev algorithm (Section 5.2) with the AoA-Diff algorithm (Section 5.1). We call this method as "AoA-Combo". Intuitively, while multi-spoofer attack may increase the standard deviation of AoAs, the difference between AoAs and AoEs would still exist. Note that both AoA-Dev and AoA-Diff produce an angular value (in degrees) as the output. AoA-Combo simply takes the output angle of AoA-Diff subtracting the output angle of AoA-dev (i.e., AoA-Diff - AoA-Dev). The detection result of AoA-Combo is shown in Figure 16b. The performance of AoA-Combo is better, with a true positive rate of 0.86, and a false positive rate of 0.05 under 10 spoofers.

The results show that our methods have some level of resilience against multiple spoofers. We leave more in-depth studies of multi-spoofer attacks to future work.

10.3 Applicable Scenarios and Limitations

Working with other GNSS. Our methods are mainly evaluated against GPS spoofing attacks. The same idea can be extended to the civilian bands of other Global Navigation Satellite Systems such as GLONASS, Beidou, and Galileo. Other wireless communication techniques that require multiple over-the-air sources (such as the transition zone of cellular networks) can leverage this idea to detect spoofing too.

Possible Deployment Scenarios. Our smartphone implementation is primarily used to examine the idea's feasibility. We have not fully explored the design space yet. For example, one of our prototypes relies on human body as the shield. This prototype can be further improved, e.g., by taking advantage of the GPS sensors in *wearable devices* such as smartwatches and smart necklaces. With wearable devices, we may leverage the blocking effect caused by *natural human movements*.

The experiments with the metal shield (Section 6.3) also suggest other design possibilities. For example, we may build a mechanical gadget that automatically rotates a GPS receiver along with a metal plank. Such a gadget can be used in

moving vehicles or stationary infrastructures that need GPS services. The rotation motion of the gadget can be powered either electrically or through natural forces (e.g., wind force propelling a pinwheel-like structure). We defer the design of such mechanical gadgets to future work.

Applicability to More Advanced GPS Chipsets. The smartphones we used all have a refresh rate of 1Hz for the GPS reading. Such a low refresh rate limits our speed of detection as it takes time to collect CN0 measurements. Note that many GPS chipsets in the market can have a refresh rate of 10Hz. We expect that our scheme can detect spoofing attacks even faster for these more advanced GPS chipsets.

Other Adaptive Attack Strategies. In addition to the adaptive attack method discussed in Section 7, attackers may come up with other strategies. For example, attackers may choose to spoof a subset of satellites instead of all of them. The idea is to let the victim device receive both spoofed and legitimate GPS signals, and thus disrupt our detection scheme (e.g., AoA-Dev). This adaptive strategy, however, is difficult to realize in practice. First, to avoid any suspicion caused by abrupt changes in GPS time estimation, spoofers must maintain both precise time synchronization and phase coordination between the spoofed and real signals. Then, even if this challenging requirement is met, the attacker would face two situations: (1) If the attacker lets the legitimate signals dominate, the victim will no longer calculate the desired fake location. This is because GPS devices typically have satellite selection algorithms that automatically exclude "outliers". Such algorithms are implemented differently among vendors (i.e., it is difficult to engineer a universally effective attack). (2) If the attacker lets spoofed signals dominate, our detection method can still work since the AoAs of the majority of the satellites would still be clustered around similar angles.

Other Limitations. Our experiment setups also have limitations. Due to FCC rules and regulations, we only conducted non-spoofing experiments in the outdoor environments and limited our spoofing experiments to indoor. It is possible the indoor setup cannot perfectly mimic the open air and urban canyon environments.

11 Conclusion

In this paper, we propose a GPS anti-spoofing framework for off-the-shelf GPS chipsets. This allows our spoofing detection methods to be backward compatible with a large number of existing GPS devices. By rotating the GPS receiver, we create a blocking effect that allows us to estimate the signals' angle-of-arrival (AoA) to facilitate spoofing detection. we demonstrate the robustness and the efficiency of the detection schemes under both basic and adaptive spoofing attacks. We also discuss other potential application scenarios of the detection methods beyond our current prototypes.

Acknowledgment

We thank our shepherd Aanjhan Ranganathan and anonymous reviewers for their constructive comments and suggestions. This work was supported in part by NSF grants CNS-1547366, CNS-1824494, CNS-2030521, and CNS-1717028.

References

[1] Spoofing a super yacht at sea. UT News, 2013. https://news.utexas.edu/2013/07/30/spoofing-a-superyacht-at-sea/.

[2] Gp2015 datasheet. DigChip.com, 2015. https://www.digchip.com/datasheets/parts/datasheet/537/GP2015.php.

[3] Labsat 3 wideband. labsat.co.uk, 2017. https://www.labsat.co.uk/index.php/en/products/labsat-3-wideband.

[4] Internet for the masses not a focus for kymeta, phasor, 2018. https://spacenews.com/internet-for-the-masses-not-a-focus-for-kymeta-phasor/.

[5] Global positioning system (gps), 2020. https://www.gps.gov/.

[6] Jahshan Bhatti and Todd E Humphreys. Hostile control of ships via false gps signals: Demonstration and detection. *NAVIGATION: Journal of the Institute of Navigation*, 2017.

[7] Kenneth R Britting. Inertial navigation systems analysis. *Wiley-Interscience*, 1971.

[8] Ali Broumandan, T Lin, A Moghaddam, D Lu, J Nielsen, and G Lachapelle. Direction of arrival estimation of gnss signals based on synthetic antenna array. In *ION GNSS+*, 2007.

[9] Hongjun Choi, Wen-Chuan Lee, Yousra Aafer, Fan Fei, Zhan Tu, Xiangyu Zhang, Dongyan Xu, and Xinyan Deng. Detecting attacks against robotic vehicles: A control invariant approach. In *ACM CCS*, 2018.

[10] crescentvenus. *WALB (Wireless Attack Launch Box)*, 2017. https://github.com/crescentvenus/WALB.

[11] Mahsa Foruhandeh, Abdullah Z. Mohammed, Gregor Kildow, Paul Berges, and Ryan Gerdes. Spotr: Gps spoofing detection via device fingerprinting. In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2020.

[12] Alessia Garofalo, Cesario Di Sarno, Luigi Coppolino, and Salvatore D'Antonio. A gps spoofing resilient wams for smart grid. In *EWDC*, 2013.

[13] Inside GNSS. In feasibility of multi-frequency spoofing, 2018. https://insidegnss.com/infeasibility-of-multi-frequency-spoofing/.

[14] G. Goavec-Merou, J.-M Friedt, and F. Meyer. Gps spoofing using software defined radio. In *FOSDEM*, 2019.

[15] Google. *Raw GNSS Measurements*, 2020. https://developer.android.com/guide/topics/sensors/gnss/.

[16] Christoph Günther. A survey of spoofing and counter-measures. *NAVIGATION: Journal of The Institute of Navigation*, 2014.

[17] Pinyao Guo, Hunmin Kim, Nurali Virani, Jun Xu, Minghui Zhu, and Peng Liu. Roboads: Anomaly detection against sensor and actuator misbehaviors in mobile robots. In *DSN*, 2018.

[18] Todd E Humphreys, Brent M Ledvina, Mark L Psiaki, Brady W O'Hanlon, and Paul M Kintner. Assessing the spoofing threat: Development of a portable gps civilian spoofer. In *Radio Navigation Laboratory Conference Proceedings*, 2008.

[19] Hridu Jain, Sherman Lo, Yu Hsuan Chen, Fabian Rothmaier, and J David Powell. Accommodating direction ambiguities in direction of arrival based gnss spoof detection. In *ION Pacific PNT*, 2019.

[20] S Rao Jammalamadaka and Ambar Sengupta. Topics in circular statistics. *World Scientific*, 2001.

[21] Kai Jansen, Matthias Schäfer, Daniel Moser, Vincent Lenders, Christina Pöpper, and Jens Schmitt. Crowd-gps-sec: Leveraging crowdsourcing to detect and localize gps spoofing attacks. In *IEEE SP*, 2018.

[22] Kai Jansen, Nils Ole Tippenhauer, and Christina Pöpper. Multi-receiver gps spoofing detection: Error models and realization. In *ACSAC*, 2016.

[23] Wonho Kang and Youngnam Han. Smartpdr: Smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors Journal*, 2015.

[24] Andrew J Kerns, Daniel P Shepard, Jahshan A Bhatti, and Todd E Humphreys. Unmanned aircraft capture and control via gps spoofing. *JFR*, 2014.

[25] Sherman C. Lo and Yu Hsuan Chen. Robust gnss spoof detection using direction of arrival: Methods and practice. In *ION GNSS+*, 2018.

[26] Steve Markgraf. *osmo-fl2k: Using cheap USB 3.0 VGA adapters as SDR transmitter*, 2019. https://osmocom.org/projects/osmo-fl2k/wiki/Osmo-fl2k.

[27] Michael Meurer, Andriy Konovaltsev, Manuel Appel, and Manuel Cuntz. Direction-of-arrival assisted sequential spoofing detection and mitigation. In *ION GNSS+*, 2016.

[28] Damian Miralles, Nathan Levigne, Dennis M Akos, Juan Blanch, and Sherman Lo. Android raw gnss measurements as the new anti-spoofing and anti-jamming solution. In *ION GNSS+*, 2018.

[29] Paul Y Montgomery. Receiver-autonomous spoofing detection: Experimental results of a multi-antenna receiver defense against a portable civil gps spoofer. In *Radio Navigation Laboratory Conference Proceedings*, 2011.

[30] Ruben Morales-Ferre, Philipp Richter, Emanuela Falletti, Alberto de la Fuente, and Elena Simona Lohan. A survey on coping with intentional interference in satellite navigation for manned and unmanned aircraft. *IEEE Communications Surveys & Tutorials*, 2019.

[31] Sashank Narain, Aanjhan Ranganathan, and Guevara Noubir. Security of gps/ins based on-road location tracking systems. In *IEEE SP*, 2019.

[32] NASA. Broadcast ephemeris data, 2020. https://cddis.nasa.gov/Data_and_Derived_Products/GNSS/broadcast_ephemeris_data.html.

[33] Tyler Nighswander, Brent Ledvina, Jonathan Diamond, Robert Brumley, and David Brumley. Gps software attacks. In *ACM CCS*, 2012.

[34] Juhwan Noh, Yujin Kwon, Yunmok Son, Hocheol Shin, Dohyun Kim, Jaeyeong Choi, and Yongdae Kim. Tractor beam: Safe-hijacking of consumer drones with adaptive gps spoofing. *ACM TOPS*, 2019.

[35] Brady W O'Hanlon, Mark L Psiaki, Jahshan A Bhatti, Daniel P Shepard, and Todd E Humphreys. Real-time gps spoofing detection via correlation of encrypted signals. *NAVIGATION: Journal of The Institute of Navigation*, 2013.

[36] osqzss. *gps-sdr-sim*, 2016. https://github.com/osqzss/gps-sdr-sim/.

[37] Mark L. Psiaki and Todd E. Humphreys. Protecting gps from spoofers is critical to the future of navigation. *IEEE Spectrum*, 2016.

[38] Mark L Psiaki, Steven P Powell, and Brady W O'Hanlon. Gnss spoofing detection using high-frequency antenna motion and carrier-phase data. In *ION GNSS+*, 2013.

[39] Aanjhan Ranganathan, Hildur Ólafsdóttir, and Srdjan Capkun. Spree: a spoofing resistant gps receiver. In *MobiCom*, 2016.

[40] Fabian Rothmaier, Yu-Hsuan Chen, and Sherman Lo. Improvements to steady state spoof detection with experimental validation using a dual polarization antenna. In *ION GNSS+*, 2019.

[41] 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.

[42] Erick Schmidt, Zachary Ruble, David Akopian, and Daniel J Pack. Software-defined radio gnss instrumentation for spoofing mitigation: A review and a case study. *IEEE TIM*, 2018.

[43] Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. Spinloc: Spin once to know your location. In *HotMobile*, 2012.

[44] Junjie Shen, Jun Yeon Won, Zeyuan Chen, and Qi Alfred Chen. Drift with devil: Security of multi-sensor fusion based localization in high-level autonomous driving under gps spoofing. In *USENIX Security*, 2020.

[45] Daniel P Shepard, Todd E Humphreys, and Aaron A Fansler. Evaluation of the vulnerability of phasor measurement units to gps spoofing attacks. *IJCIP*, 2012.

[46] Yubo Song, Kan Zhou, and Xi Chen. Fake bts attacks of gsm system on software radio platform. *Journal of Networks*, 2012.

[47] Ivan Studnia, Vincent Nicomette, Eric Alata, Yves Deswarte, Mohamed Kaâniche, and Youssef Laarouchi. Survey on security threats and protection mechanisms in embedded automotive networks. In *DSN-W*, 2013.

[48] Peter F Swaszek, Scott A Pratz, Benjamin N Arocho, Kelly C Seals, and Richard J Hartnett. Gnss spoof detection using shipboard imu measurements. In *ION GNSS+*, 2014.

[49] Çağatay Tanıl, Samer Khanafseh, Mathieu Joerger, and Boris Pervan. An ins monitor to detect gnss spoofers capable of tracking vehicle position. *IEEE TAES*, 2018.

[50] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. On the requirements for successful gps spoofing attacks. In *ACM CCS*, 2011.

[51] Bertold Van den Bergh and Sofie Pollin. Keeping uavs under control during gps jamming. *IEEE Systems Journal*, 2018.

[52] Kyle D Wesson, Jason N Gross, Todd E Humphreys, and Brian L Evans. Gnss signal authentication via power and distortion monitoring. *IEEE TAES*, 2018.

[53] Tegg Westbrook. The global positioning system and military jamming. *Journal of Strategic Security*, 2019.

[54] Yuan Wu, Hai-Bing Zhu, Qing-Xiu Du, and Shu-Ming Tang. A survey of the research status of pedestrian dead reckoning systems based on inertial sensors. *IJAC*, 2019.

[55] Nian Xue, Liang Niu, Xianbin Hong, Zhen Li, Larissa Hoffaeller, and Christina Pöpper. Deepsim: Gps spoofing detection on uavs using satellite imagery matching. In *ACSAC*, 2020.

[56] Kexiong Curtis Zeng, Shinan Liu, Yuanchao Shu, Dong Wang, Haoyu Li, Yanzhi Dou, Gang Wang, and Yaling Yang. All your GPS are belong to us: Towards stealthy manipulation of road navigation systems. In *USENIX Security*, 2018.

[57] Zengbin Zhang, Xia Zhou, Weile Zhang, Yuanyang Zhang, Gang Wang, Ben Y Zhao, and Haitao Zheng. I am the antenna: accurate outdoor ap location using smartphones. In *MobiCom*, 2011.

A Appendix: Spoofer Localization

With the ability to estimate AoA of GPS signals, we can further infer the location of the spoofer with additional analysis. A naive approach is to perform rotation from at least two different locations. Then we use the estimated AoAs to locate the spoofer by simple triangulation. However, this naive approach is highly dependent on the accurate AoA estimation, which can be error-prone especially under spoofing conditions. Instead, we perform AoA-guided navigation (an adaptation of [43, 57]) to locate the spoofer.

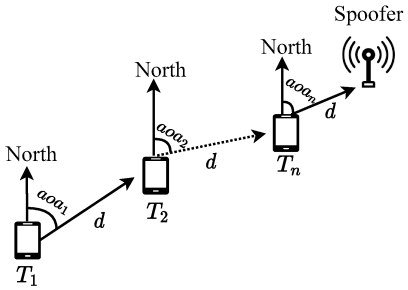


Figure 17: Illustration of the spoofer localization algorithm.

The basic idea is demonstrated in Figure 17. At an initial location T_1, the GPS receiver spins locally to obtain an AoA aoa_1, then we move toward the AoA direction for a certain distance d and arrive at T_2. After that, the receiver spins again and repeat the above steps until the spoofer is within view. During the process, we leverage the build-in IMU sensors in the smartphone to measure the moving distance d based on an existing method [23].

	MIN	MAX	AVERAGE
Basic Attack - Fitting	1.4°	21.4°	10.5°
Basic Attack - SA	3.0°	31.7°	15.2°
Adaptive Attack - Fitting	6.9°	66.1°	29.6°
Adaptive Attack - SA	12.6°	152.6°	62.3°

Table 2: Minimum, maximum, and average values of AoA error from different methods in open-air with human blockage.

Evaluation: Direction Derivation. The localization depends on an accurate estimation of AoAs. We first evaluate the accuracy of AoA inference. We considered both the basic attack and the adaptive attack. We examine two methods to derive AoAs: sine-wave fitting (Section 5.1) and frequency analysis (Section 8.2). The derived AoAs are used to compare with ground-truth AoAs to calculate the AoA error (i.e., the absolute difference between the two angles). Table 2 shows the AoA errors for the OAH setting. These observations suggest that a simple triangulation of AoAs cannot accurately locate the spoofer.

From Table 2, we observe that (1) the sine-wave fitting method work betters than the FA method on deriving AoAs; (2) the sine-wave fitting method is able to get basic spoofing signals' AoAs accurately with an average error of 10.5° but it is more difficult to estimate AoAs under adaptive attacks (with an average error of 29.6°).

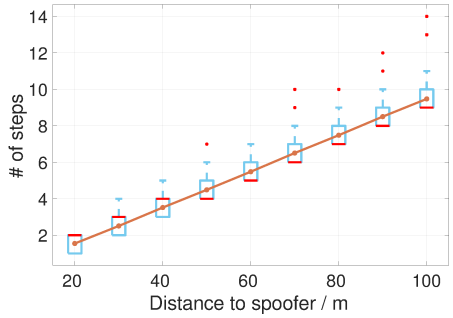


Figure 18: Simulation Result of gradient-based localization Method, moving step $d = 10m$, stop threshold $c = 10m$

Evaluation: Localization Accuracy. To evaluate our proposed method, we build a simulation framework (in MATLAB), which allows us to analyze the localization results without performing spoofing in a large outdoor space. In this framework, a GPS spoofer is randomly assigned at a location which is L meters away from the GPS receiver. Once our system detects the spoofing signal, it will use the sine-wave

fitting method to derive the AoA and repeat steps in the *AoA-guided* navigation method to locate the spoofer. Once the GPS receiver is close enough to the spoofer (less than c meters), the simulation is terminated as the spoofer is within view. We record the number of steps it takes to locate the spoofer. In an ideal situation where AoA error is 0, the number of steps should be closed to L/d (where d is the moving distance per step). Both AoA error and walking distance measurement errors are modeled by a normal distribution (the mean and standard deviation is set based on our measured data).

The experiment results are shown in Figure 18. We set the spoofing signals to be generated by a basic attack and the sine-wave fitting method will be used to derive AoA. Then we set the moving distance d to 10m, the threshold distance for stopping searching c to 10m, and changes the L from 20m to 100m. For each setting, 1000 simulations are performed and the distribution of required steps is plotted as blue box plots in Figure 18. The orange line connects the average value of steps in each setting. Our observation is that as L changes from 20m to 100m, the value of required steps for locating the spoofer centralized in the range of $[L/d-1, L/d+1]$, which is closed to the results from the ideal situation where AoA error is 0. These results suggest that the multi-step navigation helps to rectify the AoA inference errors and converge to the spoofer location.

B Appendix: Other Supporting Materials

We put other supplementary materials in this section. Figure 19 and Figure 20 are photos of our experiment setups. Figure 21 shows the Radiation Pattern of an omnidirectional dipole antenna with a metal plate as the blocking material. With the blocking material, the Radiation Pattern can effectively mimic that of a directional antenna. Algorithm 3 shows the detailed process of the Spectrum Analysis (SA) based spoofing detection method. The SA method is used to detect adaptive spoofing attacks.

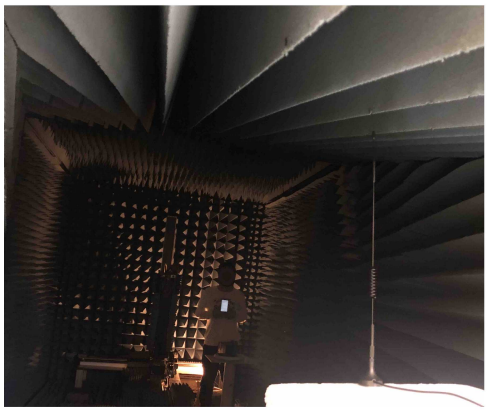


Figure 19: Anechoic chamber used for testing

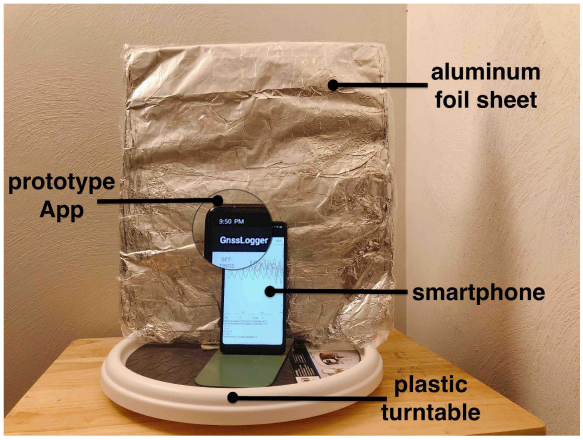
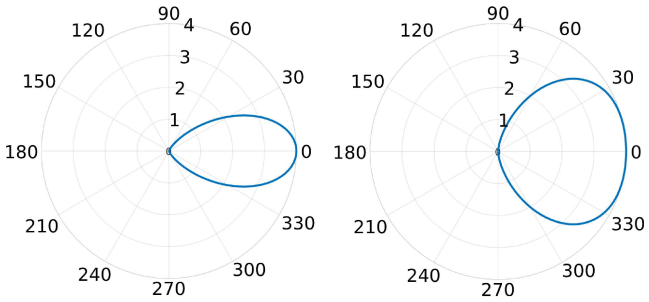


Figure 20: Metal blockage experiment setup



(a) Side view (b) Top view

Figure 21: Radiation Pattern of an omnidirectional dipole antenna that is placed $< \lambda/4$ away from a metal plate, where λ is the wavelength and is around 19cm for GPS signal.

ALGORITHM 3: Spectrum Analysis

Input: G
Output: AoA
Initialization: $AoA \leftarrow \emptyset$
$timewindow = \{1, 2, ..., N\}$
Preprocessing: Obtain $f_r, S = \{s_1, s_2, ..., s_M\}, C_{s_i} = [c_{1s_i}, c_{2s_i}, ..., c_{Ns_i}]$ and $A = [a_1, a_2, ..., a_N]$ from GNSS measurements G
for *each satellite* s_i **do**
    $X_{s_i}(f) = FFT(C_{s_i})$
    Get phase from the rotation frequency f_r: $phase_{s_i} = getAngle(X_{s_i}(f_r))$
    $aoa_{s_i} = a_1 - phase_{s_i}$
    $AoA = append(AoA, aoa_{s_i})$
end for
return AoA