Drive Me Not: GPS Spoofing Detection via Cellular Network
(Architectures, Models, and Experiments)

Gabriele Oligeri, Savio Sciancalepore, Omar Adel Ibrahim, Roberto Di Pietro
Division of Information and Computing Technology
College of Science and Engineering, Hamad Bin Khalifa University
Doha, Qatar

ABSTRACT
The Global Positioning System (GPS) has been proved to be exposed to several cybersecurity attacks, due to its intrinsic insecure design. GPS spoofing is one of the most easiest, cheap, and dreadful attacks that can be delivered: fake GPS signals can be sent to a target device and make it moving according to a pre-computed path.

Although some proposals exist to discriminate between legitimate and rogue GPS signals, those solutions are still difficult to deploy, since they resort to special hardware capable of identifying physical properties of original GPS signals.

In this paper, we propose a brand new approach, exploiting the broadcast signals transmitted by the mobile cellular network infrastructure to validate the position received by the GPS infrastructure. In detail, we provide several contributions: (i) the architecture of our solution; (ii) the analytic models related to the GSM infrastructure, including the number of in-range base stations, the distance to the base stations, and the received signal strength; and, (iii) the results achieved via an extensive measurement campaign, carried out by first collecting GPS signals while driving for more than 158 Km, and then using these data to build an experimental model for the evaluation of the performance of our technique in the detection of a wide number of emulated spoofing attacks.

Finally, we also tested our solution against a real GPS spoofing attack. We proved it being able to guarantee 0% false positive and 100% detection with an almost negligible delay—all the system parameters being finely tunable, allowing for a wide range of possible trade-offs.

ACM Reference format:

1 INTRODUCTION
Systems and technologies rely more and more on the use of positioning and navigation technologies, such as the Global Positioning System (GPS), to provide enhanced services to their users [4]. Information derived from the GPS technology are used for a wide range of applications, from location based services up to the suggestion of the optimal path to reach a specific destination. All the GPS receivers get the signals from dedicated satellites, and derive their location based on the estimation of the signal time-of-arrival. Thus, the availability and reliability of those signals is of critical importance to guarantee high levels of Quality of Service (QoS), as well as effective positioning and path decisions [6, 12, 24].

This is especially true for smart navigation systems, heavily relying on GPS information to travel from a given source to destination, and to select the most appropriate path according to traffic information [30]. Indeed, smart navigation systems strongly rely on GPS infrastructure and either jamming or GPS spoofing could significantly affect their effectiveness, as well as the safety of their users [17]. Indeed, differently from military-use GPS systems, civilian-use GPS systems are neither encrypted nor authenticated—hence being prone to cybersecurity attacks [27]. The threats to GPS are also exacerbated by the wide availability of cheap Software Defined Radios (SDRs). SDRs can easily forge GPS signals and deviate the target device from its intended path [19]. Since real GPS signals are very weak in power (as they are received from very far satellites), fake signals can be easily superimposed and let the receiver/user devices deviating from the real position, causing severe threats to the security and safety of the users [8, 28]. For instance, in the context of semi-autonomous assisted-driving vehicles (such as transportation trucks) assumed throughout this paper, malicious adversaries can inject fake GPS signals with the objective to deviate selected vehicles from their intended truck, e.g. to exhaust fuel resources, to steal such vehicles, or to crash them.

Over the last years, several solutions have been introduced to cope with the above threats (see Sec. 2 for a comprehensive overview), but they either rely on the access to the physical properties of the GPS signals (Doppler effect, direction of arrival, and so on), or they require additional modifications to the underlying hardware components, e.g., resorting to phased antennas arrays. Thus, they all require specific hardware that, in the vast majority of the cases, cannot be easily integrated with the already existing ones. In addition, while existing solutions are effective at only detecting an ongoing attack, they do not provide any backup localization/navigation solution.

Contribution. We propose a GPS spoofing detection and mitigation technique exploiting the signals received by the mobile cellular network. Our solution compares the information received from the GPS infrastructure with that ones coming from the Base Station (BS) belonging to the mobile cellular network, its position

ACM, 2019. This is personal copy of the authors. It is posted here by permission of ACM for your personal use. Not for redistribution.
The definitive version of the paper will be published soon through the ACM Digital Library on https://dl.acm.org.
WiSec’19, Miami, USA
© 2019 ACM 978-1-4503-xxxx-xxxx/X/Y/M. . . $15.00
DOI 10.1145/mmmmm.mmmmm
being static and publicly available. To evaluate the effectiveness of our solution, we conducted a wide measurement campaign driving more than 150km for about 10 hours, while gathering data from a real GSM infrastructure. These data have been used to build three analytic models including: (i) the number of in-range BSs, (ii) the distance between the user and the BSs, and finally, (iii) the Received Signal Strength (RSS) at the user side, further used to test the performance of our technique against GPS spoofing attacks.

Finally, we provide the details and the results of a real GPS spoofing attack and related detection. We prove that our solution can detect all the spoofing attacks while experiencing zero false positives and a detection delay of less than 115 seconds—that is, experiencing a maximal deviation from the planned path of just 1.75 Miles when traveling at 55Mph.

Paper organization. The rest of this paper is organized as follows: Sec. 2 provides an overview of the related work. Sec. 3 introduces the scenario, the adversary model, and the equipment adopted in our work, while Sec. 4 introduces a high level overview of our proposed spoofing detection scheme. Sec. 5 describes the data collection strategy and the analytic models of the cellular network deployment, adopted in the subsequent Sec. 6 to provide the performance of our solution against several emulated spoofing attacks. Real spoofing attacks are introduced in Sec. 7, where we also assess the validity of our solution to thwart them. Finally, Sec. 9 tightens conclusions and draws our future research activities.

2 RELATED WORK

The vulnerabilities of GPS to spoofing attacks are well-known in the current literature and, especially in the context of turn-by-turn navigation, new threats are arising, as demonstrated by recent attacks such as [31].

Indeed, in the very recent period, there have been several contributions focusing on methods to detect GPS spoofing attacks. One of the relevant directions to detect spoofed GPS signals is to recur to an higher number of receivers, i.e., to location diversity schemes. To provide a few examples, authors in [11] focused on the technique of multi-receiver GPS spoofing detection, aiming at detecting malicious spoofing signals by exploiting positions from several GPS receivers deployed in a fixed constellation. Specifically, the authors investigated how previous models can be improved due to the correlation of errors at co-located receiver positions, and they concluded that the receivers could be either located very close to each other, improving the overall applicability of the countermeasure. The use of many GPS receivers is exploited also by [7], introducing a signal authentication architecture based on a network of cooperative GPS receivers. A receiver belonging to the network correlates its received signal with those ones received by other receivers to detect spoofing attacks. Similarly, in the context of vehicular communications, authors in [13] proposed a decentralized scheme for the detection of GPS spoofing. In this scheme, vehicles exchange their measured GPS code pseudo-ranges with neighboring vehicles using dedicated short-range communications. The vehicles then perform linear operations on the exchanged GPS data and derive independent statistics that are related to the measurements of each neighbor. Using these statistics, a vehicle implements a cumulative sum procedure to locally detect high correlations in the time of arrival of spoofed GPS signals. However, this technique cannot be applied when the vehicle cannot communicate with other vehicles in the range. Also in [33], the authors proposed a system based on two different antennas, in order to evaluate and compare the arriving direction of the GPS signals. Thanks to the redundant design, the system can detect GPS spoofed signals as they arrive all from the same directions, while this is not true for the genuine GPS sources. However, all of the above schemes can be applied only if a certain number of receiver antennas can be deployed. If a single receiver is available, they cannot be setup. With reference to desert and open areas, [23] recently proposed a location verification scheme using meteor burst communications to detect GPS spoofing attacks. Unfortunately, this method requires a dedicated infrastructure to be setup, and could not be used for urban scenarios.

In the context of avionics communications, authors in [10] proposed Crowd-GPS-Sec, a spoofing detection mechanism that neither requires any updates of the GPS infrastructure nor of the airborne GPS receivers. In contrast, Crowd-GPS-Sec leverages crowdsourcing to monitor the air traffic from GPS-derived position advertisements that aircraft periodically broadcast for air traffic control purposes. Specifically, spoofing attacks are detected and localized by an independent infrastructure on the ground which continuously analyzes the contents and the times of arrival of these advertisements. In the same context, the contribution in [9] detects and localizes spoofing devices by utilizing the information provided by a large-scale air traffic surveillance system such as Opensky-network, dedicated to the monitoring of the air traffic.

Authors in [21] introduced SPREE, a spoofing detection mechanism using a technique called auxiliary peak tracking. SPREE does not rely on GPS signal authentication and therefore can be used to detect both civilian and military GPS spoofing attacks. Despite being designed to be standalone, without depending on other hardware such as antennas, additional sensors or alternative sources of location information (like maps or inertial navigation systems), it requires the access to the physical GPS signals, rarely available in regular receivers.

With reference to the specific constraints of Internet of Things (IoT) devices, authors in [1] proposed a novel GPS spoofing detection scheme based on hardware oscillators. The scheme depends on measuring the frequency drift and offset of a free-running crystal oscillator with respect to the GPS signals. The receiver only trusts the on-board free running local oscillator, and the intrinsic properties of these oscillators exhibit a strong correlation with the authentic GPS signals. However, it requires the access the on-board oscillator, not enabled in regular GPS receivers.

The use of information provided by other auxiliary networks as a validation or a backup to the legitimate GPS infrastructure has been investigated by only a few solutions. However, these contributions are mainly focused on the localization task, and aims either at increasing the accuracy of the location estimation in indoor scenarios or to provide a rough localization when the GPS system is not available. For instance, the authors in [20] proposed a novel method to detect GPS-spoofing based on monocular camera and IMU sensor of a UAV. With reference to the use of information coming from the cellular network, the authors in [14] shows that a rough localization of an indoor user can be achieved by processing information.
from seven or more cooperative localization users instead of the main stream approach of using only three or four information transmitting users or anchors (base stations). [32] recently proposed a novel localization scheme called NextMe, which is based on cellular phone traces, leveraging the fact that mobile call patterns are strongly correlated with the co-locate patterns. Such correlation is extracted as social interplay from cellular calls, and used for location prediction from temporal and spatial perspectives. Similarly, the authors in [3] proposed an accurate and calibration-free mobile device localization algorithm in cellular networks, exploiting the mutual Received Signal Strength (RSS) between base stations.

However, despite being strictly related, localization and spoofing detection are two separate research topics and require different system design choices. While localization solutions are willing to replace GPS or provide rough location estimation when GPS signals are not available, spoofing detection techniques are designed to work aside with the GPS, raising alarms and providing corrections only in hazardous situations, where inconsistencies are detected.

3 SCENARIO AND ADVERSARY MODEL

In this section we introduce the scenario tackled in the paper, the adversary model, and the equipment used for performing the measurements and assessing the effectiveness of our solution.

3.1 Scenario

Figure 1 shows the system and adversary model adopted in this work.

Our solution perfectly fits every scenario involving an entity that resorts to the GPS infrastructure to move from a source to a destination position. Some examples include, but are not limited to: (i) a tourist walking in a city; (ii) car/motorcycle sharing services; (iii) a truck pulling a trailer of goods; and, (iv) a flying autonomous drone. Without loss of generality, in the following we consider a pretty standard yet relevant scenario where a user, provided with a Mobile Terminal (MT), resorts to a semi-assisted navigation system to drive a truck (as depicted in Fig. 1) from a source to a destination, leveraging the GPS infrastructure for the turn-by-turn navigation. The MT is able also to receive information from the cellular infrastructure, i.e., GSM, 3G or LTE, and to leverage these information to validate the position obtained from the GPS. Since the moving entity should be close to the ground (given the presence of a cellular network), our solution does not fit scenarios involving either airplanes or ships, that already leverage dedicated techniques [10]. Table 1 summarizes the notation used in the paper.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS</td>
<td>Base Station of the mobile cellular network.</td>
</tr>
<tr>
<td>MT</td>
<td>Mobile Terminal: the entity moving from a source to a destination position.</td>
</tr>
<tr>
<td>X_GPS</td>
<td>Current GPS position of the MT.</td>
</tr>
<tr>
<td>lat_GPS(t)</td>
<td>Current Latitude GPS coordinate of the MT at time instant t.</td>
</tr>
<tr>
<td>lon_GPS(t)</td>
<td>Current Longitude GPS coordinate of the MT at time instant t.</td>
</tr>
<tr>
<td>X_est</td>
<td>Estimated position from the cellular network.</td>
</tr>
<tr>
<td>RSS(t)</td>
<td>Received signal strength from the ith BS at time t.</td>
</tr>
<tr>
<td>d_e</td>
<td>Error distance between X_GPS and X_est.</td>
</tr>
<tr>
<td>Φ</td>
<td>Threshold to consider d_e as an anomaly.</td>
</tr>
</tbody>
</table>

The following definitions will be also considered:

**Definition.** We define Current position the position of the MT estimated from the GPS network infrastructure. This position can be made inconsistent (i.e., not real) when the adversary spoofs the GPS infrastructure by transmitting fake signals to the mobile node.

**Definition.** We define Estimated position the position computed by the MT by exploiting the mobile cellular network infrastructure. The MT exploits the in-range BSs to define a plausibility area for its position.

3.2 Adversary model

Our adversary model involves a malicious user willing to divert the moving entity MT from the original intended path. One of the cheapest way for an adversary to reach the above goal consists in resorting to the usage of a SDR and a GPS spoofing software. Note that this set-up makes the attacker really powerful, mobile, and difficult to identify. Indeed, the adversary tunes the SDR on the GPS frequency (1.575.42 MHz) and it starts spoofing the actual GPS, emitting messages with the same identifiers and the same format of legitimate satellites.

Given that the legitimate GPS signals are characterized by relatively low power levels, the spoofed signals can be easily superimposed and the MT, not being able to discriminate between the real and the spoofed signals, will lock to the spoofed signals, deviating from its intended trajectory.

The aim of our solution is to leverage the messages received from the mobile cellular network to detect the ongoing GPS spoofing attack, as well as to assess the real position of the MT. It is worth noting that, once spoofing is detected, further measures could be taken, such as raising an alarm about the ongoing attack.

We assume the adversary is not able to either compromise any of the BS belonging to the cellular network or to compromise the MT, which is assumed to be a trusted device.

Figure 2 wraps up on the adversarial model and the envisaged spoofing attack. When the MT (a truck in the figure) is moving in a benign scenario, the current position (path) and the estimated one are consistent. Of course, in a benign scenario, the current position is consistent with the one retrieved via the GPS (black straight line). At a certain time, the adversary performs its attack spoofing the position of the MT. The spoofed position is setup by...
the adversary such that the MT still assumes to be moving towards the destination point, while it has actually been diverted towards a completely different direction (black dashed line). However, the MT might compare the spoofed position coming from the GPS with the estimated position retrieved from the cellular network; a difference between the two positions could be leveraged to declare being victim of a spoofing attack.

A core component of our spoofing detection algorithm is the MT averagin\textsuperscript{1}g the geographical positions of the BSs. approximated position as a function of the in-range BSs, e.g., by that the BSs can be uniquely identified, and that their positions be either pre-loaded or dynamically acquired via Internet). Note

Figure 3: Rough location estimation: three BSs (B1, B2, and B3) uniquely identify an area (uncertainty area).

Three or more BSs keep transmitting beacon messages, that define the uncertainty area, i.e., the area where messages are received by a mobile receiver. We assume the MT belongs to that area, and it can retrieve the GPS coordinates of the transmitting BSs (they can be either pre-loaded or dynamically acquired via Internet). Note that the BSs can be uniquely identified, and that their positions \((x_i, y_i)\) are publicly available. Therefore, the MT might estimate its approximated position as a function of the in-range BSs, e.g., by averaging the geographical positions of the BSs.

We highlight that, unlike other solutions in the literature, our MT location estimation technique does not resort to the Received Signal Strength (RSS) for the estimation of the MT-BS distance.

Algorithm 1: Pseudo-code of the GPS spoofing detection algorithm.

```
/* Online computations at time \(t = KT\), with \(0 \leq K < \infty\).*/
let \(X_{GPS}(t) = [\text{lat}_{GPS}(t), \text{lon}_{GPS}(t)]\) be the GPS position at the time \(t\).
let \(N(t)\) be the number of in-range BS at time \(t\).
let \(BSID_i\) Unique identifier of the \(i\)-th BS.
let \(X_{BS_i} = [\text{lat}_{BS_i}, \text{lon}_{BS_i}]\) be the GPS position of the \(i\)-th BS.
let \(RSS_i(t)\) be the Received Signal Strength of the signal received from the \(i\)-th BS at the time \(t\).
let \(\text{spoof}_{vector}\) be the vector logging the detected anomalies.

while true do
    /* Retrieving BSs coordinates
    MT retrieves BSs’ coordinates from the Internet exploiting the received BSID
    */
    /* Remove outliers
    Remove (potential) BSs whose distances from the other BS are more than three scaled Median Absolute Deviation (MAD) away from the median;
    Generate a weights’ vector to take into account stronger signals as more reliable.
    */
    Compute the weights \(w = [w_1, w_2, ..., w_{N-1}]\) for each of the \(N - 1\) remaining BSs;
    /* Estimate the MT position combining the BS distances.
    Compute the weighted centroid \(Y_{est} = [\text{lat}_{Y_{est}}, \text{lon}_{Y_{est}}]\) of the positions of the remaining \(N - 1\) BSs;
    Compute the error distance \(d_{e}(t) = Y_{est} - X_{GPS}(t)\);
    Validate \(X_{GPS}\) with \(Y_{est}\) against a pre-defined threshold.
    */
    if \(d_{e}(t) \geq 0\) then
        /* Anomaly not detected. */
        spoof_vector[\(i\)] = 0;
    else
        /* Anomaly detected. */
        spoof_vector[\(i\)] = 1;
    end
end
```

4 GPS VALIDATION VIA MOBILE CELLULAR NETWORK

A core component of our spoofing detection algorithm is the MT location estimation procedure, that leverages the mobile cellular network infrastructure.

Our solution exploits the beaconing messages transmitted in broadcast from the BSs to compute a rough localization, whose aim is only to validate the position provided by the GPS infrastructure, as depicted in Fig. 3.

Figure 2: Adversary model, spoofing attack, and BSs estimated position.
in a pre-loaded data structure.

**Removing outlier BS.** When the MT moves around, it might receive and collect data from anchors that are far away with respect to its current position. Such anchors negatively affect the aforementioned uncertainty area (that is, enlarging it) due to the high error in the distance between the MT and the BS. In each time-slot, we select and discard all the BS with a distance, measured with respect to the other BSs in the same set, greater than three times the Median Absolute Deviation (MAD)—this latter value computed over the median of the distances.

**Estimate MT’s position as a function of the BSs distances.** We adopt a weighted-centroid computation technique to estimate the position of the MT. The rationale is to weight the BS as a function of their RSS values, such that BSs that are closer to the MT, having higher RSS, are considered more reliable. Therefore, we consider the exponential distribution function in Eq. 1

\[
y = 1 - f(x, \mu) = 1 - \frac{1}{\mu} e^{-\frac{x}{\mu}}, \tag{1}
\]

where \(x\) are the normalized and sorted RSS values, and \(\mu\) is the mean parameter. Eventually, the weights \(w_i\) are computed as the normalization of the elements in \(y\), as \(w_i = \frac{\mu}{\sum_i y_i}\). These weights are then used to compute a weighted centroid \(Y_{est}\), as shown by Eq. 2.

\[
Y_{est} = [\text{lat}_{Y_{est}}, \text{lon}_{Y_{est}}] = \left[ \sum_{i=1}^{N} \text{lat}_{BS_i} \cdot w_i, \sum_{i=1}^{N} \text{lon}_{BS_i} \cdot w_i \right]. \tag{2}
\]

It is worth noting that the value of the mean \(\mu\) in the exponential Probability Distribution Function (PDF) (Eq. 1) influences the relative difference between the weights. If \(\mu\) is close to the value 0, the BSs reporting the strongest RSSs are more influential in the computation of the position \(Y_{est}\). Conversely, if \(\mu\) has an higher value, the weights will be more homogeneous, and thus, the RSSs will have a minor influence on the final centroid estimation.

**Detecting anomalies.** When the error distance \(d_e\), obtained as the distance between the GPS position \(X_{GPS}\) and the position estimated as the centroid of all the BS-MT distances, is greater than a given threshold \(\Phi\), i.e., \(d_e(t) \geq \Phi\), an anomaly event is detected and a counter, namely \(\text{anomaly}\), is incremented. Note that an anomaly does not lead directly to a spoofing attack.

**Decide on spoofing event.** As it will be clear in the following, a spoofing attack cannot be declared by evaluating only one sample or, equivalently, a single anomaly—the false alarm rate would be unbearable. Indeed, the estimated position by the BSs, i.e., \(Y_{est}\), might be affected by a significant error that, in turn, might raise a lot of false positive alarms. Therefore, we consider a temporal sequence of events, and we decide to declare the MT being subject to a spoofing attack when a pre-defined number of anomalies is experienced by the MT (as it will be clear in the following Sections). This procedure, implemented by the `detect_spooﬁng_from_anomalies` function, allows us to filter out spurious events (false positives), while enabling the detection of a real spoofing attack.

### 5 EQUIPMENT, DATA COLLECTION, AND MODELLING

In this section we introduce the details of our measurement campaign, including the software and tools used for data acquisition, as well as the theoretical models we designed to fit our measurements.

#### 5.1 Equipment and tools

Figure 4 shows the equipment used for our measurements and for the spoofing attack.

![Figure 4: Equipment set-up for the measurement and for the spoofing attack.](image)

The details of our equipment are reported in the following:

- **Smart-phone.** We adopted a smartphone running the Android Operating System version 8.1.0, kernel version 4.4.95+, equipped with a MT6739 Quad Core processor running at 1.3Ghz, 8GB of ROM memory and 1GB of RAM memory. The smartphone features two Subscriber Identification Module (SIM) cards, thus being able to receive messages from two different operators at the same time.

- **Software Defined Radio (SDR).** We adopted the HackRF One [15] as Software Defined Radio to perform the spoofing attack. HackRF One is an open source hardware platform that can be used as a USB peripheral or programmed for stand-alone operation for either transmitting or receiving radio signals in the range from 1 MHz to 6 GHz. The HackRF One has been equipped with a Temperature Compensated Crystal Oscillator (TCXO), used to provide much higher levels of temperature stability than the ones that are possible to achieve with the default crystal oscillator.

- **GPS Spoofing application.** To carry out the GPS spoofing attack, we consider the publicly available tool GPS-SDR-SIM [5]. GPS-SDR-SIM generates GPS baseband signal data streams, which can be converted to RF using the HackRF One. In addition, GPS-SDR-SIM allows to spoof either fixed positions or moving ones, creating very effective GPS spoofing attacks.

- **Android application.** We developed a dedicated Android application to collect information from both the cellular network and the GPS infrastructure at the same time. The user can specify a sampling period \(T\); then, every \(T\) seconds, the application logs the information from the in-range BSs.
(by calling the Android method \textit{getAllCellInfo}) and the current location of the smartphone (MT) obtained via the GPS (using the Android methods \textit{requestLocationUpdates} and \textit{getLastKnownLocation} of the LocationManager library). Then, it generates a log-file with all the above information.

- **Spoofing detector software.** The spoofing detector has been implemented in Matlab R2018b©. It takes the log file from the smartphone as input, and it provides several statistics, as well as the spoofing detection decision for each considered time frame.

Our experimental measurement campaign has been carried out by driving around a car and collecting information with the in-house developed android application. Data are acquired every $T=100\text{ms}$, collecting information from two different cellular network operators in our country (Qatar), including both Vodafone-Qatar and Ooredoo. Each BS is specified as a unique combination of the CID, the LAC, and the MNC. As for the position of the BSs, we retrieved them from the Internet [29][16]. It is worth noting that the map of the BSs could be also built in advance, especially for recurrent paths, simply navigating the path while, at the same time, logging the BSs identifying data.

Finally, all our measurements have been collected by using the 2G cellular network technology. Although our smartphone is able to log information from both 3G and 4G, the number of deployed anchors for such technologies was significantly less than the number observed for 2G. Since our solution is significantly affected by the density of the BSs—but not by the underlying technology, be it 3G, 4G or 5G—, we choose to focus on 2G BSs.

### 5.2 Measurements description

We collected 10 different traces (paths) in the city of Doha (Qatar) as depicted in Fig. 5. Solid lines show the different paths, while black dots represent the GSM BSs identified combining the CID, LAC, and MNC captured by our measurements and cross checked against the data extracted from the previous mentioned websites. The measurement playground is a rectangle of about $20.36\text{Km} \times 15.71\text{Km}$, for a total area of $319.95\text{Km}^2$. The BSs are mainly deployed along the streets, with an higher (unsurprisingly) concentration in densely populated areas (e.g. upper left corner of Fig. 5).

Table 2 provides a high level description of the collected traces. For each path, we report the distance (in meters), the duration (in seconds), and the average speed (in meters per seconds). Path lengths span between about $9\text{Km}$ to about $25\text{Km}$, with different average speed depending on the traffic conditions. We took particular care on the choice of the paths, in order to capture the most different and heterogeneous phenomena related to the anchors deployment. We generated a rich dataset, that will be available for download at [2], consisting of an overall distance of about $158\text{Km}$ collected in about 10 hours.

### 5.3 Measurements, statistics and modelling

In the following, we design the statistical models that will be used later on for emulating the spoofing attacks, based on the real measurements discussed above. Our statistical models capture the following mobile cellular network patterns:

- Number of in-range BSs (hereby referred also as anchors);

![Figure 5: Geo-located paths and BSs positions.](image)

<table>
<thead>
<tr>
<th>ID</th>
<th>Dist. (m)</th>
<th>Durat. (s)</th>
<th>Speed (m/s)</th>
<th>N. Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9984.23</td>
<td>1205.65</td>
<td>8.28</td>
<td>6319</td>
</tr>
<tr>
<td>2</td>
<td>6034.85</td>
<td>628.10</td>
<td>9.61</td>
<td>6282</td>
</tr>
<tr>
<td>3</td>
<td>6708.56</td>
<td>921.50</td>
<td>7.28</td>
<td>9216</td>
</tr>
<tr>
<td>4</td>
<td>10199.55</td>
<td>1284.76</td>
<td>7.94</td>
<td>9362</td>
</tr>
<tr>
<td>5</td>
<td>23165.00</td>
<td>2066.38</td>
<td>11.21</td>
<td>20057</td>
</tr>
<tr>
<td>6</td>
<td>25462.06</td>
<td>1895.77</td>
<td>13.43</td>
<td>18059</td>
</tr>
<tr>
<td>7</td>
<td>22636.34</td>
<td>1866.93</td>
<td>12.12</td>
<td>16802</td>
</tr>
<tr>
<td>8</td>
<td>14138.03</td>
<td>1273.80</td>
<td>11.10</td>
<td>11767</td>
</tr>
<tr>
<td>9</td>
<td>24067.50</td>
<td>1955.80</td>
<td>12.31</td>
<td>18632</td>
</tr>
<tr>
<td>10</td>
<td>16286.16</td>
<td>2002.35</td>
<td>8.13</td>
<td>19072</td>
</tr>
</tbody>
</table>

Total 158682.28 15101.05 10.14 135569

- Distance between BSs and MT; and,
- RSS estimated by the MT.

**Number of in-range BSs.** Figure 6 shows the complemented Cumulative Distribution Function (1-CDF) associated with the number of in-range anchors experienced by the MT. For instance, the probability that the MT experiences at least 8 in-range BS is about 0.32. The inset figure represents the associated PDF, i.e., the probability for the MT to experience exactly a given number of (in-range) anchors. We observe that the number of in-range anchors spans between 2 and 14 with a median value equal to 7.

The solid blue line in Fig. 6 represents the best fit distribution according to the Maximum Likelihood Estimate (MLE) technique. We found that the best fit is a negative binomial distribution, having parameters $r = 16.83$ and $p = 0.70$, respectively. Thus, the following Eq. 3 yields:

$$P(x; r, p) = \left(\frac{r + x - 1}{x}\right)p^x(1 - p)^r,$$

with $x$ spanning between 2 and 14, i.e., $2 \leq x \leq 14$.

**Distance between BSs and MT.** Figure 7 shows the PDF associated with the distances among MT and all the in-range anchors. We observe that the median value is about 790 meters, while the quantile 0.9 is about 2.11 Km. The inset figure, instead, represents
the Cumulative Distribution Function (CDF) associated to the distance between the MT and the anchors. It is worth noting that the probability that the distance MT-BS is less than 2Km is about 0.9. Finally, we performed a best fit analysis on the PDF according to the MLE technique. We found out that the PDF can be approximated by a Gamma distribution function, with parameters $a = 1.61$ and $b = 0.64$. Thus, the following Eq. 4 yields:

$$P(x; \alpha, \beta) = \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}}, \quad (4)$$

where $\Gamma(\alpha)$ is the Gamma function, being equal to $\Gamma(\alpha) = (\alpha-1)!$, while $x$ spans between 0 and 5.5 Km.

![Figure 6: Probability to experience at least a given number of anchors. The inset figure represents the probability density function associated with the number of in-range anchors, while the blue solid line is the best-fit distribution.](image6)

![Figure 7: Probability Distribution Function associated to the distance between MT and the anchors. The inset figure shows the CDF computed on the same data.](image7)

**RSS estimated by the MT.** Fig. 8 shows the PDF associated to all the RSSs values collected for all the traces.

We observe that RSS values span between -110 dBm and 10 dBm, with a quantile 0.5 value (median) of about -39 dBm. As for the previous cases, we performed a best fit analysis on the PDF according to the MLE technique, and we found the best fit to be equal to a normal distribution with parameters $\mu = -38.61$ and $\sigma = 19.74$, yielding the following Eq. 5:

$$P(x; \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (5)$$

with $x$ spanning between -110 and 10, i.e., $-110 \leq x \leq 10$.

**5.4 Error bounds for BS-based location estimation**

In the following we discuss the validation of the GPS position ($X_{GPS}$ from Algorithm 1) exploiting the estimated position ($Y_{est}$) provided by the BSs. As already discussed before, combining the distances among the base stations BSs and the mobile terminal MT does not provide a precise location, but only an uncertainty area (recall Fig. 3). In order to estimate the size of the aforementioned uncertainty area, we combine all the traces and, for each time slot ($T = 100\text{ms}$), we compute the distance error $d_e$ between $X_{GPS}$ and $Y_{est}$. Figure 9 depicts the statistical analysis associated with $d_e$: we consider both the PDF and the temporal analysis associated to $d_e$ (inset figure). The final objective of this analysis is to define a decision threshold ($\Phi$), useful to discriminate between anomalies and consistent estimates.

![Figure 8: Received Signal Strength (RSS) estimated from all the BSs for all the paths.](image8)

![Figure 9: Position estimation error: Probability Distribution Function and time serie analysis.](image9)
We consider such a threshold (Φ — line 15 of Algorithm 1) as the quantile 0.9 of the error, being equal to about 600 meters. Such an assumption implies that all GPS positions are considered trusted if their distance $d_e$ from the one computed leveraging the BSs is less than $\Phi = 600$ meters. We recall that the single event $d_e > \Phi$ is not enough to declare a spoofing attack. Indeed, looking at the inset of Fig. 9, we observe the presence of many points crossing the Φ threshold. These transients can be filtered out resorting to the already introduced detecting_spoofing_from_anomalies() function (recall lines 21–26 in Algorithm 1).

Finally, the last parameter to be considered is $\mu$, introduced in Eq. 1. It represents the mean of the exponential distribution adopted to weight the BS contributions to the centroid computation. Indeed, we recall that each BS contributes via its Received Signal Strength (RSSs). Fig. 10 shows the error ($d_e$) as a function of $\mu$, with $1 \leq \mu \leq 100$, considering all the collected traces from Table 2. We recall that the value of $\mu$ in Eq. 5 provides the relative weight of BSs with strongest RSSs with respect to other (less strong) BSs. The smallest (i.e., close to 0) the value of $\mu$, the most the measurements coming from close BSs will be considered for the location estimation with respect to measurements coming from far BSs.

In the remainder of this paper, we set $\mu = 25$, as we believe it represents an acceptable trade-off (minimum error) for all the collected traces.

![Figure 10: Position estimation error as a function of the exponential coefficient $\mu$ adopted in the computation of the RSS-weighted centroid for approximating the node position.](image)

6 SPOOFING DETECTION

In this section we show the performance of our solution by emulating several spoofing attacks. Indeed, we considered the traces summarized in Tab. 2 and we randomly choose: (i) the time at which the spoofing event starts; and, (ii) the trajectory of the spoofed path. We considered a diverting trajectory (as already introduced by Fig. 2), characterized by the same speed of the target as before the spoofing event. In addition, we emulated the reception of the BSs signals according to the models previously obtained through eqs. 3, 4, and 5. Therefore, the diverted path looks fully genuine according to speed, number of in-range BSs, distance to the BSs, and finally, Received Signal Strength (RSSs). However, a critical difference could still be spotted: the anchors experienced in the diverted path will be different with respect to the ones in the genuine, expected trajectory; therefore, our algorithm (leveraging the position computed from the in-range BSs) will be able to raise an alarm for the ongoing spoofing attack.

6.1 Baseline example

In this section, we consider only one path (trace ID 1), and we provide some insights on the logic of our solution.

Fig. 11 shows an excerpt from trace T1, where the MT (black dots) is moving towards a certain destination. The circles show the overall in-range BSs, while the crosses correspond to all the estimations of the MT positions using the weighted centroid introduced by Eq. 2.

![Figure 11: Track example (T1). Circles depict in-range BSs, black dots show the MT track, while crosses correspond to the BS estimations of the MT position.](image)

As expected, estimated positions (crosses) are affected by an error $d_e$ that, on average, is less than $\Phi = 600$ meters (as previously discussed for Fig. 10). $\Phi$ represents the minimum distance from the real position of the target to declare the current position as an anomaly event (recall Fig. 9). As previously introduced, an anomaly event is not considered directly as a spoofing attack, since we require multiple, sequential anomalies in order to declare a spoofing attack. Firstly, we recall the definition of spoofing_vector introduced with Algo. 1, as a Boolean array taking on either 1 or 0 as a function of the presence of an anomaly event. To declare a spoofing attack, we rely on the analysis of the sample distribution (anomalies) inside the aforementioned vector in order to discriminate false positives from true spoofing events. We observe that, when the MT is on a diverted path due to a spoofing attack (as in Fig. 2), the in-range anchors will let the MT to compute an estimated position that will be sensibly different from the experienced (spoofed) one. Once the MT is on a spoofed path, we expect the number of anomalies to constantly grow and, in the following, we refer to a consistent, temporal sequence of anomalies, as a burst. We perform the analysis of the distribution of the anomalies inside the spoofing_vector: Fig. 12 shows the burst length for both the not-spoofed path and the spoofed one. We observe that the maximum burst length for the benign scenario is constituted by 297 subsequent anomalies—that,
in this case, are just false positives. Conversely, the adversarial scenario experiences bursts of length 131, 132, and 1609 respectively. Finally, it is worth noting that any threshold strictly greater than 297 events will introduce zero false positives and the correct identification of the spoofing attack.

In particular, for Trace 1, the time requested to identify the aforementioned spoofing attack sums up to 1345 + 297 events = 1642 (164.2 seconds), where 1345 is the number of events after the beginning of the spoofing and before the long burst (1609) started.

### 6.2 Detection delay and algorithm performance

In this section we test all the system parameters over the collected traces from Table 2. We consider 100 spoofing attacks for each of the traces in Table 2, for an overall 1000 emulated GPS spoofing attacks: both the target spoofing destination and the starting time of the spoofing attack have been randomly set at each run. After the spoofing attack starts, we emulate the number of in-range anchors, the distance between the MT and the anchors (BSs), and finally, the Received Signal Strength (RSS) according to eqs. 3, 4, and 5, respectively. Then, for each round and according to Algo. 1, we compare the estimated position from the BSs with the one provided by: (i) the MT’s GPS receiver (benign scenario); and, (ii) the spoofing attacker (adversarial scenario).

Assuming a benign scenario and as sampling period $T = 100$ms, the solid green line in Fig. 13 shows the number of bursts as a function of their duration. Each burst element (anomaly) has been classified as such according to the quantile 0.9 of all the errors estimated for the specific trace (recall Section 5.4). The burst distribution of the benign scenario allows us to compute the lower bound (solid red line) for the spoofing decision: since the maximum burst length in the benign scenario is constituted by 1145 consecutive anomalies, we define a spoofing event as a burst constituted by at least 1146 anomalies. We highlight that the previous definition guarantees zero false positives. It is worth noting that it could be possible to sensibly decrease such a burst length in order to catch spoofing attack much earlier, the price being an increase in the number of false positives.

The solid blue line in Fig. 13 shows the number of bursts as a function of their duration, assuming the adversarial scenario above introduced. Given the previous threshold definition (a burst of 1146 anomalies to declare a spoofing attack), we can compute the detection delay experienced before declaring a spoofing attack: recalling that the sampling period $T$ is equal to 100ms, the detection delay sums up to 114.6 seconds.

### 7 DETECTING A REAL SPOOFING ATTACK

In the following we test the effectiveness of our algorithm against a real spoofing attack. We consider a standing still MT, and we spoof its position by mimicking a pre-defined path as depicted in Fig. 14. Our experiments are based on the following multi-steps procedure:

1. **Fake path generation.** We generated a fake path from the actual MT position to a random destination by using the software Google Earth Pro [18], making the path consistent with actual roads, intersections, and turns.

2. **Data format conversion.** The output of Google Earth Pro [18] (KML file format) is not suitable to be used directly with the adopted GPS spoofing software GPS-SDR-SIM. Therefore, we resort to Labsat SatGen [22] to convert the fake path to the standard NMEA GGA stream format [25].

3. **GPS signal file generation.** We generated the signal file (gpssim.bin) using the GPS-SDR-SIM tool, adopting the default RINEX navigation file for ephemerides (brdc3540.14n), and 8 bits for the I/Q data format.

4. **GPS signal transmission.** We used the hackrf_transfer software to transmit the generated signal at the GPS frequency 1.57542 GHz, using a sampling frequency of 2.6 MHz.

5. **MT logging.** During our tests, we used our Android app, run by the MT, to log the GPS coordinates into a file.

Fig. 14 shows how the estimated positions from the BSs (circles) are close to the initial position of the path (the actual position of the MT). After some time, the spoofed position (black dots trajectory) moves away from the estimated one (crosses), and therefore, our algorithm will eventually detect the ongoing spoofing attack.
We set the anomaly threshold as the distance between the position estimated via the BSs, and the one received from the GPS, i.e., $\Phi = 78m$, as depicted by the red circle in Fig. 14. Then, Alg. 1 is used to declare a spoofing attack. Since all the distances are considered as anomalies—being greater than $\Phi = 78m$, the spoofing alarm is raised after 114.6 seconds (recall the analysis of Section 6.2). We consider two models: (i) a car moving to an average speed of 48Km/h; and, (ii) a track moving to an average speed of 96Km/h. We observe that the spoofing attack on the car is detected after 372 meters (green circle), while the spoofing attack on the track is detected after 750 meters (blue circle). For both the scenarios, we consider the results from Section 6.2, and we set the spoofing alarm to be raised when a sequence of 1146 anomalies have been detected (114.6 seconds). The showed results are consistent, as the burst of anomalies required to declare a spoofing attack (equal in the two cases) translates in different distances, since the two vehicles travel at different speed—the higher the speed, the higher the distance from the correct path when the spoofing attack is detected.

8 DISCUSSION

In the following we discuss the quality of our solution in terms of performance, effectiveness, and efficiency.

Performance. Our solution guarantees the detection of a GPS spoofing attack in less than 115 seconds (with the requirement of zero false positives). However, it is worth noting that our results are affected by the BSs distribution, and therefore, different BSs deployment might affect the algorithm performance. Nevertheless, we paid particular attention to the measurement collection and we drove the car in different urban areas such as the downtown, characterized by skyscrapers and dense population, suburban, with villas, single family homes and services, and finally, rural, characterized by open fields with no obstructions and minimum population density. As such, we are confident that the reported data would still hold, as is, in a variety of scenario. Should the scenario vary, the system parameter could be tuned accordingly.

Effectiveness. We observe that, assuming an average car trip distance of 20Km [26], the detection error (the ratio between the length of the diverted path and the overall trip) is less than 6%. Such a value becomes significantly smaller when assuming the case of a truck, driving a standard trip of 1062Km (96Km/h for 11 hours—duty limit in US): in such a case, the detection error is about 0.1%.

Efficiency. Our solution does not introduce any major overhead to the tasks already carried out by the MT. Indeed, the MT already receives the broadcast messages from the BSs and, therefore, only minor processing is required to implement our solution. Moreover, our solution can be directly integrated in the vast majority of the smart navigation systems, since it does not require any special hardware and it resorts only to already available information: GPS coordinates and data traffic from the BSs, the latter ones being a requirement for all the modern autonomous navigation systems.

Trade-offs. Our solution can be further optimized. Indeed, we recall that the current configuration guarantees zero false positives (FP=0), while a much smaller detection delay can be provided relaxing the previous equation. This can be achieved by considering a smaller value for $\Phi$, or shorter burst lengths. The aforementioned analysis will be part of our future work.

9 CONCLUSION

In this work we have introduced a novel technique to detect and mitigate GPS spoofing attacks, leveraging existing broadcast transmissions from base stations belonging to the mobile cellular network. Although our reference scenario assumes land vehicles, our results can be considered as very general, and they might be applied to any entity (including drones) moving in an area covered by a pervasive mobile cellular network.

We have detailed our solution and we have collected real world measurements that have been put at use in testing our solution via both simulations and on an extensive experimental campaign. Results show that our solution is effective in detecting an on-going spoofing attack, while requiring little overhead and incurring in a very low false positive rate. We achieved a 0% false positive while incurring a detection delay of some 115 sec. These parameters, as well as all the ones describing our systems, can be finely tuned, for instance trading-off a slight increase in false positives with a reduced detection delay. The novelty of the proposed approach, the excellent achieved preliminary results, the extent of the expected impact, and the discussed possible extensions of this work, do pave the way for further research in this field.

Further research activities will be devoted to the investigation of the robustness of our technique to fake BSs setup by a more powerful adversary, as well as to the evaluation of different advanced metrics for a more effective detection of ongoing spoofing attacks.

ACKNOWLEDGEMENTS

This publication was partially supported by awards NPRP-S-11-0109-180242, UREP23-065-1-014, and NPRP X-063-1-014 from the QNRF-Qatar National Research Fund, a member of The Qatar Foundation. The information and views set out in this publication are those of the authors and do not necessarily reflect the official opinion of the QNRF.
REFERENCES


