

GPS spoofing detection via crowd-sourced information for connected vehicles

Gabriele Oligeri^{a,*}, Savio Sciancalepore^b, Omar Adel Ibrahim^a, Roberto Di Pietro^a

^a Division of Information and Computing Technology, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar

^b Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands

ARTICLE INFO

Keywords:

GPS spoofing detection
Crowd-sourcing
Connected vehicles
Mobile IoT
Security

ABSTRACT

Modern vehicular systems rely on the Global Positioning System (GPS) technology to provide accurate and timely services. However, the GPS has been proved to be characterized by an intrinsic insecure design, thus being subject to several security attacks. Current solutions can reliably detect GPS spoofing attacks leveraging the physical features of the received GPS signals or resorting to multiple antennas. However, these techniques cannot be deployed when the physical properties of the received signals cannot be accessed, which is the most general case for commercial GPS receivers. Alternative solutions in the literature rely on the cross-check of the received signal with information coming from additional sources. However, such proposals are typically limited to a single source, are rarely supported by experimental results, and do not provide insights on the impact of several parameters, such as detection accuracy, time, false-positives, and robustness to malicious information.

To overcome the cited limitations, in this paper, we propose an innovative approach, resorting to combined crowd-sourced information from the mobile cellular infrastructure and the WiFi networks to detect GPS spoofing attacks. Our analysis leverages an extensive experimental dataset, available online for the research community, gathered by driving around a car in urban, suburban, and rural scenarios, for around 5 h and covering more than 196 km. Our solution allows for a tunable tradeoff between detection delay and false positive; for instance, we can detect an attack in approximately 6 s, when leveraging the information coming from only the WiFi, while the delay increases to 30 s when using the information from the mobile cellular network, still achieving a false positive probability strictly less than 0.01. We also show the limitations and trade-offs of our approach, in terms of minimum detection accuracy, time, and robustness to malicious information.

The data adopted in this work are publicly released to allow results replicability and foster further research in the highlighted directions.

1. Introduction

The Global Positioning System (GPS) is nowadays the most used technology to achieve fine positioning of users and systems, as well as to provide enhanced navigation solutions to connected and autonomous vehicles [1–3]. Tiny GPS receivers are installed into smartphones, handheld devices, cars, and Unmanned Aerial Vehicles (UAVs) to enable a wide range of services, including the suggestion of the optimal path according to pre-defined requirements, location-based services, and emergency management [4]. Thus, the availability and reliability of the GPS technology is instrumental to enable dependable applications, as well as the highest levels of Quality of Service (QoS) for end-users [5–7].

However, designed in the 70s with the only requirement of service availability, the GPS is insecure by design. Thus, attacks such as *jamming* or *GPS spoofing* could be easily launched, leading to significant threats and safety issues for their users [8]. In fact, contrary to the military GPS infrastructure, the civilian GPS signals are neither encrypted nor authenticated; therefore, they can be easily spoofed using dedicated equipment [9]. Specifically, commercially available Software Defined Radios (SDRs), available at very affordable prices, can be easily tuned on the same frequency of the GPS communication technology, and configured via freely-available open-source tools to transmit messages that are indistinguishable from authentic GPS signals emitted by legitimate satellites. Moreover, since the legitimate signals are very weak at the

* Corresponding author.

E-mail addresses: goligeri@hbku.edu.qa (G. Oligeri), s.sciancalepore@tue.nl (S. Sciancalepore), oaibrahim@hbku.edu.qa (O.A. Ibrahim), rdipietro@hbku.edu.qa (R. Di Pietro).

<https://doi.org/10.1016/j.comnet.2022.109230>

Received 31 January 2022; Received in revised form 11 July 2022; Accepted 23 July 2022

Available online 29 July 2022

1389-1286/© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

ground level, the malicious GPS signals can be easily super-imposed to the legitimate ones, thus leading the GPS receivers to estimate a different location than the actual one, and hence deviate vehicles relying on such a technology from their intended path [10,11]. As an example, semi-autonomous assisted-driving vehicles could be diverted from their intended track, e.g., to exhaust fuel resources, to steal such vehicles, or to crash them [12,13].

In the last years, several techniques have been proposed to both detect and mitigate GPS spoofing attacks (see Section 2 for a comprehensive overview). Overall, these solutions can tell apart genuine signals from malicious ones by looking at the physical properties of the wireless signals, e.g., by estimating the Doppler effect or the direction of arrival of the messages, to name a few.

Despite the ones cited above are valuable and reliable approaches, these solutions require the access to the raw signals received by a MT, i.e., the I/Q samples, whose availability generally depends on the host systems. Also, the available solutions for GPS spoofing detection, often require the deployment of multiple receiving antennas either on-board or along the intended path, making this approach impractical due to space or budget requirements. Although some approaches overcome such limitations by using information from additional sources, such techniques are limited to a single source, rarely provide experimental results, and they neither provide insights on the impact of several parameters, such as detection accuracy, time, false-positives, and robustness to malicious information.

Contribution. In this paper, we propose and investigate the effectiveness of a new GPS spoofing detection and mitigation technique, exploiting the signals received by the neighboring mobile cellular base stations and WiFi networks. Our solution compares the information received from the GPS infrastructure with the ones coming from the BS belonging to the mobile cellular network and the AP of the neighboring local WiFi networks, as their positions are publicly available via online crowdsourced platforms such as *OpenCellID* and *WiGLE* [14,15]. To evaluate the effectiveness of our solution and obtain more insights into its behavior, we conducted a large measurement campaign, driving for an overall 5 h (about 196 km), while gathering data from real GSM infrastructure and local WiFi networks. These data, released at [16], have been used to build four analytic models including: (i) the number of in-range BSs and APs; (ii) the distance between the user and the anchors; (iii) the RSS at the user side; and, finally, (iv) the location estimation error, further combined and used to measure the performance of the proposed technique against GPS spoofing attacks.

We also evaluated the tight relationship existing between several parameters of our solution, including the selected operational mode (WiFi-only, GSM-only, and WiFi+GSM), the detection threshold for the anomaly detection, the detection delay, and the False Positives probability, providing the configuration of all these parameters to achieve specific target objectives. All the reported results have been evaluated both via extensive simulations and real spoofing attacks, performed using commercial SDRs. Finally, we also evaluated the performance of our solution in the presence of a powerful adversary, controlling a few mobile cellular BSs or WiFi APs and aware of the path of the connected vehicle in advance. We prove that defeating our solution would require a significant amount of transmitting sources (especially in urban areas), with implications on the required operating space and budget.

We highlight that this manuscript is a thorough extension of our previous contribution published in [17]. Compared to the above manuscript, the following novel contributions are provided in this paper:

- **Extension to the WiFi technology.** In this new version of our work, we have extended our methodology, now including also WiFi networks to help to detect ongoing GPS spoofing attacks. To this aim, we leveraged an open-source crowd-sourcing database, namely *WiGLE*, that provides information about the WiFi networks available at a certain location. In addition, we extended our

protocol to deal with three operating modes: GSM-only, combined GSM+WiFi, and WiFi-only respectively (CFR. Section 5, Section 6, and Section 7).

- **New Experimental Data Acquisition Campaign.** For this new version of our work, we collected brand new measurements related not only to the WiFi networks but also to GSM cellular networks available in the city of Doha, Qatar. Our experimental measurement campaign has been carried out by driving around a car for about 196 km, and collecting GSM and WiFi information using an in-house developed android application. The data have been collected every $T = 133$ ms, considering GSM information from two different cellular network operators in Qatar — i.e. Vodafone-Qatar and Ooredoo (see Section 5).
- **Study of False Positives and Decision Threshold.** While in our conference paper we only considered the configuration of the GPS Spoofing Detection algorithm guaranteeing 0% false positives, in this extended version of our work we studied the relationship between the decision threshold, the time (i.e., the duration of the bursts), and the false positives probability that our algorithm can achieve. We demonstrated that our algorithm can be finely tuned to raise a GPS spoofing detection alarm in a significantly reduced amount of time, while also guaranteeing an acceptable amount of false positives (see Section 6).
- **Performance evaluation with malicious GSM and WiFi transmitters.** In this extended version of our work, we considered a more powerful adversary, able to spoof not only the GPS signals but also ad-hoc GSM base stations and WiFi hotspots. We demonstrated that our GPS spoofing detection algorithm can reject such an adversary in several configuration scenarios. Given that the adversary does not know in advance the real path of the user, the only way for the adversary to be successful would be to over-provision GSM and WiFi malicious emitters — a hard to implement attack, due to realistic scenario constraints and physical limitations (see Section 8).

Paper organization. The rest of this paper is organized as follows: Section 2 reviews the most important related work, Section 3 introduces the scenario, the adversary model, and the equipment adopted in our work, Section 4 provides a big picture of our proposed spoofing detection scheme, Section 5 describes the data collection process and the models of the WiFi and the cellular network deployment, used in the following Section 6 to estimate the spoofing detection performance according to the fine-tuning of its system parameters. Real spoofing attacks are introduced in Section 7, where we also assess the validity of our solution to thwart them, while Section 8 provides an overview of the performance of our scheme when multiple BSs and APs are controlled by the adversary. Section 9 summarizes the most important features of our techniques, and finally, Section 10 concludes our work.

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 [18].

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 a higher number of receivers, i.e., to location diversity schemes. To provide a few examples, authors in [19] 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 [20], 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 received by other receivers to detect spoofing attacks. Similarly, in the context of vehicular communications, authors in [21] 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 [22], the authors proposed a system based on two different antennas, 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 set up. Another approach exploiting the raw signals has been proposed in [23], providing an algorithm based on maximum likelihood (ML) estimation. Multiple correlators are iteratively used to detect and remove the counterfeit signals, to allow the GPS receiver to lock to the authentic signals. However, this method requires access to physical information; even if this is achievable now on Linux-based and Android-based connected vehicles (as shown in [24,25]), such a method is not generalizable to any connected vehicle. When desert and open areas are involved, [26] recently proposed a location verification scheme using meteor burst communications to detect GPS spoofing attacks. Unfortunately, this method requires a dedicated infrastructure to be set up, and could not be used for urban scenarios.

In the context of avionics communications, authors in [27] 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 [28] 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 [29] 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 access to the physical GPS signals, rarely available in regular receivers.

Concerning the specific constraints of Internet of Things (IoT) devices, authors in [30] 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 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 access to the on-board oscillator, not enabled in regular GPS receivers.

Recently, the authors in [31] presented AuthGPS, a system that can identify spoofing attacks performing a verification of the validity of the GPS satellite information via LTE base stations' location information, employing a 6-digit one-time password-based authentication system.

This system adds a 6-digit code in the field of LTE broadcast messages and does not need to communicate with a dedicated server. Despite being valuable, such a system would require the modification of the standard LTE message — an invasive and not always viable solution.

The use of information provided by other auxiliary networks as validation or a backup to the legitimate GPS infrastructure has been investigated by only a few solutions. Some contributions are mainly focused on the localization task, and aims either at improving the accuracy of the location estimation in indoor scenarios or to provide a rough localization when the GPS is not available. For instance, the authors in [32] proposed a novel method to detect GPS-spoofing based on the monocular camera and Inertial Measurement Unit (IMU) sensor of a UAV. Regarding the use of information coming from the cellular network, the authors in [33] 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 mainstream approach of using only three or four information transmitting users or anchors (base stations). [34] 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 [35] proposed an accurate and calibration-free mobile device localization algorithm in cellular networks, exploiting the mutual 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 (e.g., see [36]), spoofing detection techniques are designed to work aside with the GPS, raising alarms and providing corrections only in hazardous situations, where inconsistencies are detected.

In the context of Global Navigation Satellite System (GNSS) spoofing detection, the authors in [37], similarly to our previous conference paper [17], used the broadcast information emitted by the base station of the neighboring cell of the mobile cellular network to cross-check the information obtained from the GNSS technologies. However, being based on the mobile cellular network only, this technique is affected by quite large errors, acceptable only in rural scenarios, where no other source can be used to cross-check GNSS information. At the same time, the authors did not investigate the trade-offs existing between the performance of the GNSS detection strategy, the detection distance, the number of available anchors, and the detection time. In [38], the same authors indicate some expedients and cross-checks that do not require modifications to the hardware of the device, allowing to enhance the security of a smartphone to GNSS spoofing attacks at a minimum cost. The analysis considered only static sources, and require the integration of specific sensors, not available in all connected vehicles. The authors in [39] proposed a spoofing detection technique leveraging the combination of raw GNSS measurements and raw IMU measurements, improving the detection of spoofing attacks compared to the use of GNSS information only. When considered in the use-case of connected vehicles, Inertial Navigation System (INS) units are hardly available (they could be available in autonomous vehicles, that are different from connected vehicles), preventing the application of the above technique. In addition, practical works such as [40] demonstrated the susceptibility of these INS-based approaches to environmental factors, such as wind. Finally, we highlight the thesis work presented in [41], where both WiFi and cellular networks were used to detect GNSS spoofing. Despite the logic of our work and this thesis could appear similar, the thesis work provided only very preliminary results, and obtained considering only static targets, thus hardly applying to connected moving vehicles. In addition, all the experiments included in the cited thesis report results available for a single location only, with very little general significance. In our case, conversely, all the findings have been obtained on data gathered from a moving target and validated

over an experimental campaign lasting several hours, across different environmental conditions. Furthermore, conversely to the above-cited contributions, our method is characterized by advanced flexibility, as it allows the network administrator to trade-off detection delay and detection threshold with the desired false-positive rate. At the same time, our contribution applies to a variety of application scenarios. Finally, our method does not rely on a persistent Internet connection to detect the presence of a GNSS spoofing attack.

Finally, interested readers can notice that the strategy adopted in our paper could be similar to the one adopted by other assisting technologies, such as Assisted-GPS (A-GPS). Assisted GPS (A-GPS) is a technology that augments the standard satellite-based localization by resorting to an Internet connection [42]. The position is calculated by a remote server, and then, sent to the smartphone using standard data traffic. Conversely, our solution does not require the aforementioned infrastructure (neither a remote server nor an Internet connection, as the list of GSM and Wi-Fi anchors can be downloaded before the travel), while all the computations are performed on the client-side by only leveraging unencrypted broadcast GSM messages. Although our solution is *assisted* by the GPS – and by the Wi-Fi – to detect a spoofing attack, A-GPS refers to a different infrastructure with a completely different purpose, i.e., either increasing the satellite acquisition time or off-board device position computation. Our solution does not improve the effectiveness of satellite positioning, but simply validates it. Moreover, we stress that, differently from A-GPS, we combine GSM information with the ones collected from Wi-Fi networks in the neighborhood of the smartphone.

Overall, we notice that the usage of information from additional communication technologies, such as the Mobile Cellular Network and the Wi-Fi, to cross-check the received GNSS location is not new. Nonetheless, the novel contributions of our work in this research area are manifold. First, to the best of our knowledge, we are the first to consider, at the same time, the combination of the information coming from the Mobile Cellular Network and the Wi-Fi information. Moreover, to the best of our knowledge, our contribution is the first to apply GNSS spoofing detection on multiple time-related measurements, and to evaluate the tight relationship existing between the spoofing detection accuracy and other critical parameters such as the detection time, the minimum distance at which spoofing can be detected, the density of the network elements deployed in the scenario, and the overall probability of false-positives. We provided quantitative relationships among the previously mentioned parameters through an extensive real measurement campaign, showing for the first time in the literature that effective spoofing detection can be achieved for distances as short as a few meters by increasing the detection time. As an additional contribution, we also highlighted the limitations of such crowd-sourced detection strategy in the presence of a malicious adversary, able to deploy malicious anchors. Our contributions, together with the related results and the generated data (publicly released), are not only novel but can also have practical impact on the understanding of the effectiveness and limitations of crowd-sourced spoofing detection strategies, as well as to foster further research. We will provide more details in the next sections.

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

Fig. 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 connected 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 car or a truck (as depicted in Fig. 1) from a source to a destination, leveraging the GPS infrastructure for the turn-by-turn navigation. Indeed, modern cars and trucks come with turn-by-turn navigation embedded in the on-board entertainment system, and to work properly, they should be connected to the Internet (contributing to the current definition of *connected vehicles*) [43]. The MT is able also to receive information from the mobile cellular infrastructure, i.e., GSM, 3G or LTE, and the nearby Wi-Fi APs, 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 Wi-Fi and cellular networks), our solution does not fit scenarios involving either airplanes or ships, that already leverage dedicated techniques [27,44].

We highlight that we refer to *crowd-sourcing* as the practice to opportunistically obtain information from sources in the neighborhood, i.e., in our case, the anchors, being either base stations or Wi-Fi APs [45]. We also highlight that our study aims to investigate the feasibility of detecting GPS spoofing attacks in connected vehicles, implicitly characterized by dynamic movements. At the same time, we recall that identifying GNSS spoofing attacks on static targets when a user is involved is very easy, as the user could easily notice a movement of the position reported by the GPS, without any movement involved on the physical perspective. Instead, realizing the presence of GPS spoofing attacks when a vehicle is moving could be not easy to be realized from the user perspective, especially if the user does not have previous knowledge of the path.

Note that, despite our solution can be considered a plausibility check based on the GPS position and the estimated ranges of all visible base stations and APs, there are significant challenges to address. Information sources are very heterogeneous (involving both GSM base stations and Wi-Fi APs) and they are originated from wireless technologies, i.e., characterized by severe service discontinuity. The opportunistic nature of the collected data introduces gaps in the system reliability (as it happens for all the wireless services), and therefore, requiring a strong statistical characterization, a careful configuration of the parameters, and finally, a thorough analysis of the results.

Table 1 summarizes the notation used in the paper.

The following definitions will be also considered:

Definition. We define *Anchor* an entity emitting messages according to any mobile cellular communication technology (GSM, 3G, 4G, and so on) or Wi-Fi technology (IEEE 802.11a, IEEE 802.11b, IEEE 802.11n, IEEE 802.11ac, to name a few), whose position is well-known or publicly available from online sources.

Definition. We define *GPS 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 Wi-Fi networks and the mobile cellular network infrastructure. The MT exploits the in-range cellular BSs and the Wi-Fi anchors in its proximity 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 originally intended path. One of the cheapest ways for an adversary to reach the above goal consists in resorting to the usage of a SDR and a GPS spoofing software. Note that

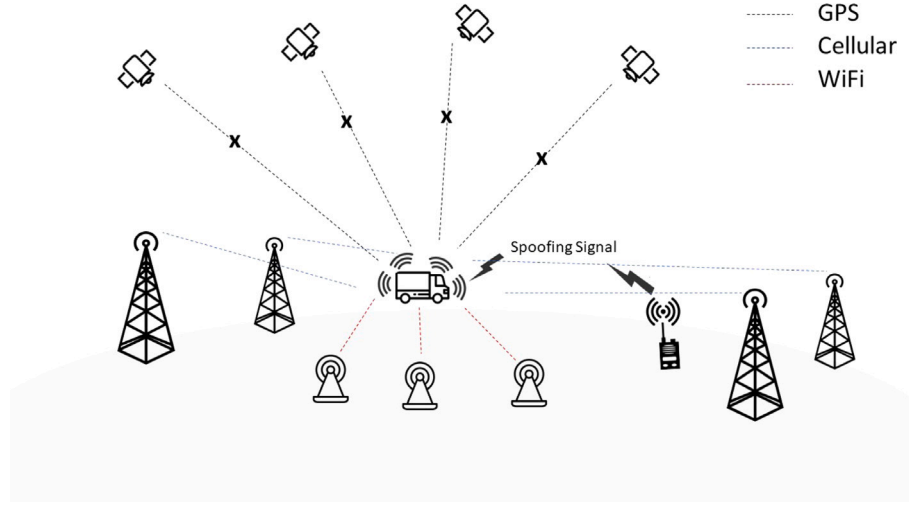


Fig. 1. The communication scenario assumed in our work. A Mobile Terminal (a truck in the figure) receives spoofed GPS signals, super-imposed over legitimate GPS signals. At the same time, the MT can also receive signals from neighboring Mobile Cellular Base Stations and WiFi Access Points.

Table 1

Notation summary.

BS	Base Station of the mobile cellular network infrastructure.
AP	Access Point of any local WiFi network.
MT	Mobile Terminal: the entity moving from a source to a destination position.
X_{GPS}	Current GPS position of the MT.
$lat_{GPS}(t)$	Current Latitude GPS coordinate of the MT at time instant t .
$lon_{GPS}(t)$	Current Longitude GPS coordinate of the MT at time instant t .
$Y_{est,cell}$	Estimated position from the cellular network.
$Y_{est,WiFi}$	Estimated position from the WiFi network
Y_{est}	Combined crowdsourced estimated position from the WiFi and cellular networks.
$RSS_i(t)$	Received signal strength from the i th BS at time t .
$RSS_m(t)$	Received signal strength from the m th AP at time t .
d_e	Error distance between X_{GPS} and Y_{est} .
Φ	Threshold to consider d_e as an anomaly.

this set-up makes the attacker very 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 at the receivers, 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.

Our solution aims to leverage the messages received from the nearby WiFi networks and 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.

While we assume that the MT is a trusted device, we also assume that the adversary can either compromise or deploy ex-novo a given number of anchors. In Section 8, we will consider the deployment of malicious anchors and we will evaluate how many of them have to be deployed to experience a given probability to defeat our solution,

as well as the intuition about practical and physical challenges in deploying such an attack. Moreover, as it will be clear in the following, the spoofing of many GSM/WiFi anchors is highly impractical. Indeed, for each spoofed anchor, the adversary should add an entry to a publicly-available database (that is checked by the MT during the GPS validation), and while this might be feasible (and proven as useless in the remainder of this paper) for a few anchors, the cited technique becomes impractical and easily detectable when scaled up to many anchors.

Fig. 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 GPS 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 set up by the adversary such that the MT still assumes to be moving towards the destination point, while it has 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 crowdsourced WiFi and cellular networks; a difference between the two positions could be leveraged to declare the MT being the victim of a spoofing attack.

Finally, we highlight that jamming of the WiFi could be easily detected by our algorithm, as the target would notice that just a few WiFi APs are visible in locations where many were expected. Overall, this attack would cause our location estimation algorithm to operate only using mobile-cellular networks. Jamming also the signals from the mobile cellular network would make the scenario perfectly matchable with a remote location, where our solution could not work. However, we remark that jamming, either WiFi or GSM, is out of the scope of this manuscript, and it would require the design of different ad-hoc techniques (jamming mitigation).

4. GPS validation via crowd-sourced information

A core component of our spoofing detection algorithm is the MT location estimation procedure, which leverages both the WiFi networks and the mobile cellular network infrastructure.

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

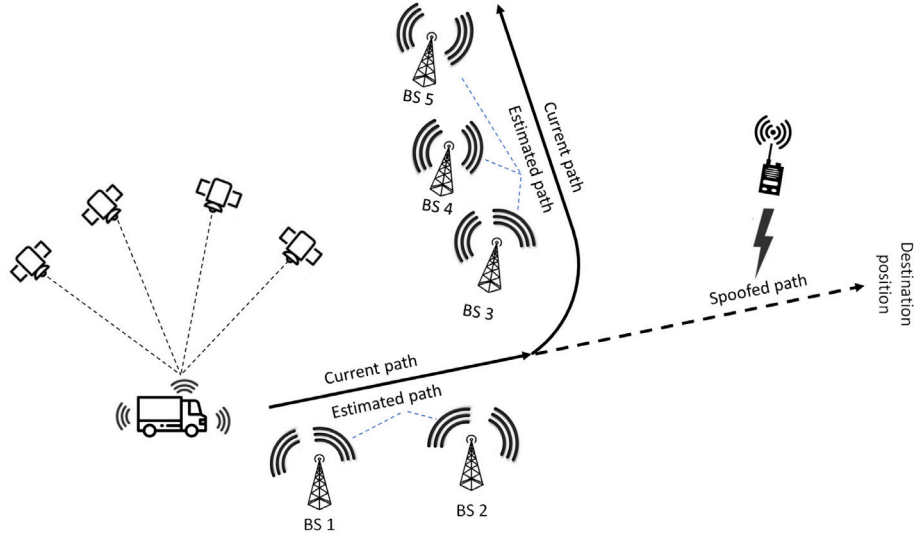


Fig. 2. Adversary model, spoofing attack, and crowd-sourced estimated position.

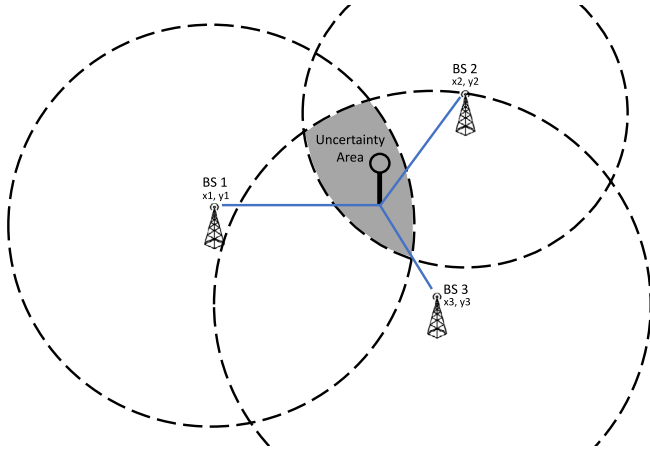


Fig. 3. Rough location estimation: three BSs (B1, B2, and B3) uniquely identify an area (uncertainty area).

Three or more mobile cellular network BSs and some WiFi APs 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 anchors (they can be either pre-loaded or dynamically acquired via the Internet). Note that the anchors can be uniquely identified, and that their positions (x_i, y_i) are publicly available, via dedicated crowd-sourced platforms such as *OpenCellID* and *WiGLE* [14,15]. Therefore, the MT might estimate its approximated position as a function of the in-range anchors, e.g., by averaging their geographical positions.

Note that the usage of *OpenCellID* and *WiGLE* for crawling information is only an example, and it does not limit the applicability of our proposal to other databases, such as the private ones managed by Google and Apple, to name a few. Also, other open-source solutions could be used, such as *OpenStreetMap*. The fundamental requirements for such databases is that they fulfill requirements such as completeness and freshness.

We highlight that, unlike other solutions in the literature, our MT location estimation technique does not resort to the RSS for the estimation of the distance between the MT and the anchor. Instead, the RSS is used only as a weight for the computation of an approximate position of the MT.

Moreover, we highlight that our method uses the well-known centroid method to compute the location of the mobile target starting from the information available from the cellular networks and the WiFi. Note that this is specifically intended for two reasons: (i) limit our solution to use only position information, so as to be integrable as a simple software application/update for any connected vehicle; and (ii) provide a worst-case scenario for our method, so as to identify the lowest bound on the performance achievable by our strategy. Other solutions, using other types of information, could possibly provide better results, but never performance worse than the ones reported in this paper.

Algorithm 1 provides the pseudo-code of our proposed solution. Our algorithm is triggered on a tunable time basis, e.g., every $T = 166$ ms in the measurements we have collected for this work. Note that the value of T we used is a technological limit of the device used for acquisition, and can be further reduced based on the capabilities of the equipment used for collecting measurements. Every T seconds, the algorithm processes the following information: (i) the GPS position of the MT, i.e. $X_{GPS}(t) = [lat_{GPS}(t), lon_{GPS}(t)]$; (ii) a unique identifier $BSID_i$ of the i th BS, constituted by the Cell ID (CID), the Location Area Code (LAC), and the Mobile Network Code (MNC); (iii) $RSS_i(t)$, that is the received signal strength associated to each signal received from a given BS, (iv) a unique identifier AP_m of the m th WiFi anchor, constituted by the MAC Address (MAC) of the transmitting device, and, finally, (v) $RSS_m(t)$, that is the received signal strength associated to each signal received from a given WiFi anchor.

Overall, our algorithm relies on a seven-step procedure:

1. Leveraging the identifiers $BSID_i$ to retrieve the BSs position;
2. Leveraging the identifiers AP_m to retrieve the position of the WiFi anchors;
3. Estimate MT position as a function of the distances to the BSs;
4. Estimate MT position as a function of the distance to the WiFi Access Points;
5. Combine the two positions in a single crowdsourced location estimation;
6. Detect possible anomalies; and,
7. Decide whether to declare a GPS spoofing attack.

The details of these steps are provided in the following.

Retrieving BSs positions from the Base Station ID. For each $BSID_i$, the MT retrieves the actual position of the i th BS. Note that the BS position could be retrieved from the Internet or looked for in a pre-loaded data structure.

Retrieving WiFi positions from the WiFi MAC Addresses. For each AP_m , the MT retrieves the actual position of the m th WiFi anchor.

```

1  /* Online computations at time t = KT, with 0 ≤ K < ∞ */
2  let  $X_{GPS}(t) = [lat_{GPS}(t), lon_{GPS}(t)]$  be the GPS position at the time t.
3  let  $N(t)$  be the number of in-range BS at time t.
4  let  $BSID_i$  Unique identifier of the i-th BS.
5  let  $X_{BS_i} = [lat_{BS_i}, lon_{BS_i}]$  be the GPS position of the i-th BS.
6  let  $RSS_i(t)$  be the Received Signal Strength of the signal received from the i-th BS at the time t.
7  let  $M(t)$  be the number of in-range WiFi anchors at time t.
8  let  $AP_m$  Unique identifier of the m-th WiFi anchor.
9  let  $X_{AP_m} = [lat_{AP_m}, lon_{AP_m}]$  be the GPS position of the m-th WiFi anchor.
10 let  $RSS_m(t)$  be the Received Signal Strength of the signal received from the m-th WiFi anchor at the time t.
11 let spoof_vector be the vector logging the detected anomalies.
12
13 while true do
14   /* Retrieving BSs and WiFi coordinates. */
15   MT retrieves BSs and WiFi anchors coordinates from the Internet exploiting the received  $BSID_i$  and MAC Addresses;
16   /* Generate a weights' vector for both the Mobile Cellular Network and the WiFi, to take into account stronger signals as more reliable. */
17   Compute the weights  $w_{cell} = [w_{cell,1}, w_{cell,2}, \dots, w_{cell,N}]$  for each of the  $N$  BSs and  $w_{WiFi} = [w_{WiFi,1}, w_{WiFi,2}, \dots, w_{WiFi,M}]$  for each of the  $M$  WiFi anchors;
18   /* Estimate the MT position combining the BS distances. */
19   Compute the weighted centroid  $Y_{est,cell} = [lat_{Y_{est,cell}}, lon_{Y_{est,cell}}]$  of the positions of the  $N$  BSs;
20   /* Estimate the MT position combining the distances of the WiFi anchors. */
21   Compute the weighted centroid  $Y_{est,WiFi} = [lat_{Y_{est,WiFi}}, lon_{Y_{est,WiFi}}]$  of the positions of the  $M$  WiFi anchors;
22   /* Combine Cellular and WiFi location estimates */
23   Compute the final position estimation  $Y_{est}$  by combining the position  $Y_{est,cell}$  obtained from the cellular network and the position  $Y_{est,WiFi}$  obtained from the WiFi, via a weighting factor  $w$ ;
24   /* Compute the error distance  $d_e$  */
25    $d_e(t) = Y_{est} - X_{GPS}(t)$ ;
26   /* Validate  $X_{GPS}$  with  $Y_{est}$  against a pre-defined threshold. */
27   if  $d_e(t) \geq \Phi$  then
28     /* Anomaly detected. */
29     spoof_vector[i] = 1;
30   else
31     /* Anomaly not detected. */
32     spoof_vector[i] = 0;
33   end
34   if discriminate_spoofing_from_anomalies(spoof_vector) then
35     Raise Spoofing Alarm;
36   end
37   else
38     Spoofing Not Detected;
39   end
40 end

```

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

Note that, as in the previous step, the position of the WiFi anchors could be retrieved either from the Internet or from a pre-loaded database.

Estimate MT position as a function of the BSs distances. We adopt a weighted-centroid computation technique to estimate the position of the MT according to the mobile cellular network. The rationale is to weigh the BSs 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).

$$y = 1 - f(x, \mu) = 1 - \frac{1}{\mu} e^{-\frac{x}{\mu}}, \quad (1)$$

where x are the normalized and sorted RSS values, and μ is the mean parameter. Eventually, the weights $w_{cell,i}$ are computed as the normalization of the elements in y , as $w_{cell,i} = \frac{y_i}{\sum_{i=1}^N y_i}$. These weights are then used to compute a weighted centroid $Y_{est,cell}$, as shown by Eq. (2).

$$Y_{est,cell} = [lat_{Y_{est,cell}}, lon_{Y_{est,cell}}] = \left[\sum_{i=1}^N lat_{BS_i} \cdot w_i, \sum_{i=1}^N lon_{BS_i} \cdot w_i \right]. \quad (2)$$

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

Estimate MT position as a function of the WiFi distances. Similarly to the previous step, to estimate the position of the MT according to the WiFi infrastructure, we adopt a weighted-centroid computation. The exponential distribution function reported in Eq. (1) is used to weigh each WiFi anchor (with index m) as a function of its RSS, and then, the weights $w_{WiFi,m}$ are computed as the normalization of the elements in y , as $w_{WiFi,m} = \frac{y_m}{\sum_{m=1}^M y_m}$. These weights are then used to compute the weighted centroid $Y_{est,WiFi}$, as shown by Eq. (3).

$$Y_{est,WiFi} = [lat_{Y_{est,WiFi}}, lon_{Y_{est,WiFi}}] = \left[\sum_{m=1}^M lat_{BS_m} \cdot w_{WiFi,m}, \sum_{m=1}^M lon_{BS_m} \cdot w_{WiFi,m} \right]. \quad (3)$$

Note that the considerations made when estimating the position exploiting the BSs distances apply also for this step, regarding the choice of the value μ . If μ is close to the value 0, the WiFi APs reporting the strongest RSSs are more influential in the computation of the position $Y_{est,WiFi}$.

Combine the two positions for estimating the final crowd-sourced location. Starting from the location estimated from the mobile cellular network, namely $Y_{est,cell}$, and the location estimated from the WiFi, namely $Y_{est,WiFi}$, a unique location estimation is computed, namely Y_{est} , by appropriately weighting the two input location estimations with a weight factor w . Specifically, the final location estimate is computed according to Eq. (4).

$$Y_{est} = [lat_{Y_{est}}, lon_{Y_{est}}] = \left[(1-w) \cdot lat_{Y_{est,WiFi}} + w \cdot lat_{Y_{est,cell}}, (1-w) \cdot lon_{Y_{est,WiFi}} + w \cdot lon_{Y_{est,cell}} \right]. \quad (4)$$

It is worth noting that when $w = 0$, only the WiFi communication technology is adopted, while when $w = 1$ only the mobile cellular network is considered. When w takes any other value, i.e. $w \in]0, 1[$, the information coming from both the sources are used to obtain the location estimation.

Detecting anomalies. When the error distance d_e , obtained as the distance between the GPS position X_{GPS} and the position Y_{est} estimated by combining the cellular and WiFi information, is greater than a given threshold Φ , i.e., $d_e(t) \geq \Phi$, an anomaly event is detected and a counter, namely (*anomaly*), is incremented. Note that an anomaly does not lead directly to a spoofing attack. Note that we do not take into account position errors due to the accuracy of the GPS, as they are considerably smaller than the location estimation error obtained by weighting GSM and WiFi broadcasts.

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 and WiFi, 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 consecutive anomalies are experienced by the MT (as it will be clear in the following Sections). This procedure, implemented by the *discriminate_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 modeling

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.



Fig. 4. Equipment set-up for the measurement and for the spoofing attack.

5.1. Equipment and tools

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

The details of our equipment are reported in the following:

- **Smartphone.** 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 1.3 GHz, 8 GB of ROM memory and 1 GB of RAM. 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 [46] 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 [47]. 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 spoofing either fixed positions or moving ones, creating very effective GPS spoofing attacks.
- **Android application.** We developed a dedicated in-house Android application, dedicated to the collection of information coming from the cellular network, the WiFi, 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 `getAllCellInfo()`), the information from the in-range WiFi (using the Android methods `wifiManager.startScan()` and `wifiManager.getScanResults()`), and the current location of the smartphone (MT) obtained via the GPS (using the Android methods `requestLocationUpdates` and `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 R2019a©. 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 $T = 166$ ms, collecting information from the neighboring WiFi APs and from the two different cellular network operators in our country (Qatar), including both Vodafone-Qatar and Ooredoo. Recall that $T = 166$ ms is only a technological limit of the device used for acquisition, and can be further reduced based on the capabilities of the equipment used for collecting measurements.

Each BS is specified as a unique combination of the CID, the LAC, and the MNC, while each WiFi AP is specified through its MAC address. As for the position of the BSs, we retrieved them from the Internet, using the online services Unwiredlabs and Opencellid [14,48]. As for the positions of the WiFi APs, we collaborated with the online service WiGLE to provide data and obtain location estimates [15]. It is worth noting that the map of both the BSs and the APs could be also built in advance, especially for recurrent paths, simply navigating the path while, at the same time, logging the BSs and APs identifying data.

Finally, as for the mobile cellular network, we highlight that all our measurements have been collected by using the 2G cellular network technology. Note that this is a worst-case condition. Indeed, due to their reduced coverage, the amount of 3G and 4G base stations is significantly higher than the 2G ones, especially in urban settings, possibly enhancing and strengthening our results. 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 several different measurements in the city of Doha (Qatar), as depicted in Fig. 5. Solid red lines show our path, red 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 previously mentioned websites. Similar considerations apply for the WiFi APs, depicted using blue dots. The measurement playground is a rectangle of about $20.25 \text{ Km} \times 21.55 \text{ Km}$, for a total area of 436.388 Km^2 . The BSs are mainly deployed along the streets, with a higher (unsurprising) concentration in densely populated areas (e.g. upper left corner of Fig. 5). At the same time, we highlight that the blue dots representing WiFi APs are concentrated very close to our path, since the reception range of the WiFi communication technology is usually limited to a few hundred meters.

Overall, our measurement campaign resulted in a rich dataset, available for download at [16], consisting of an overall number of events equal to 387,193, gathered by driving around a car for a total distance of about 196.474 Km, collected in about 5.5 h, moving at an average speed of 9.7 m/s (approximately 34.92 Km/h).

5.3. Measurements, statistics and modeling

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 and APs (hereby referred also as *anchors*);
- Distance between anchors and MT; and,
- RSS estimated by the MT for each anchor.

Number of in-range Anchors. Fig. 6 shows the complemented Cumulative Distribution Function (1-CDF) associated to the number of in-range anchors experienced by the MT, where the GSM anchors are reported in blue and the WiFi anchors are reported using the red color. Since we adopted an MT equipped with two SIM cards, a maximum of 14 different GSM anchors can be logged, at the same time.

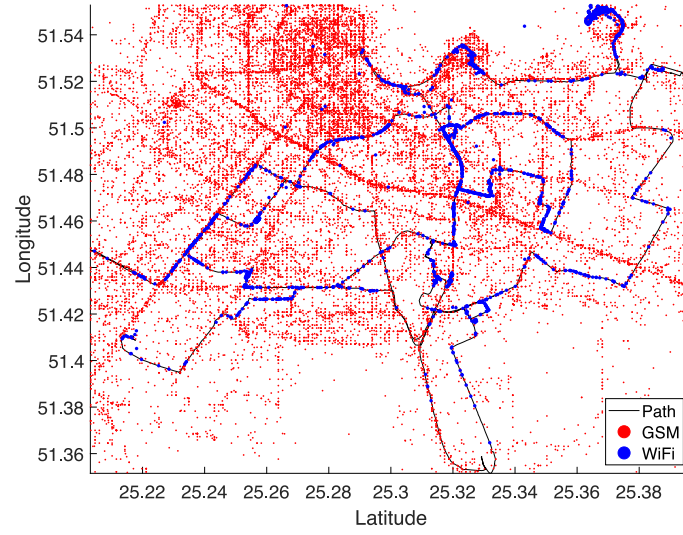


Fig. 5. Geo-located paths, APs, and BSs positions.

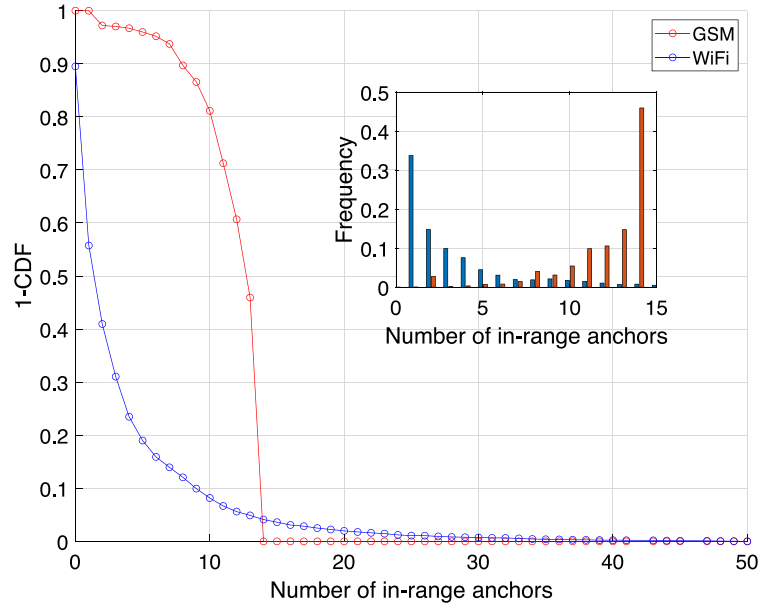


Fig. 6. Number of in-range anchors: Probability to experience at least a given number of GSM and WiFi anchors. The inset figure represents the probability density function associated to the number of in-range GSM and WiFi anchors.

To provide a reference example, we notice that the probability that the MT experiences at least 11 in-range GSM anchors is about 0.87, while this probability drops to 0.1 considering the WiFi anchors. Overall, this means that, in a given time instant, the MT will experience a higher number of GSM anchors than WiFi anchors.

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 for the GSM, with a median value equal to 13. For the WiFi, instead, we notice that the median value is 2, while the number of in-range anchors increases up to 40, corresponding to measurements in more populated areas. This means that in more populated areas the MT experiences an increased number of WiFi anchors, and such a number is definitively higher than the maximum number of GSM anchors that the MT ever experiences.

Distance between Anchors and MT. Fig. 7 shows the PDF associated to the distances among MT and all the in-range anchors.

Focusing on the GSM anchors (red dots), we observe that the median value is about 754 m, with the quantile 0.9 being about 1.91 Km. For the WiFi (blue dots), instead, the median value is 47.7 meters and the quantile 0.9 is about 355 m. These data are indeed consistent, given that the transmission/reception range of the GSM is higher than the WiFi one.

RSS estimated by the MT. Fig. 8 shows the PDF associated to all the RSSs values collected for all the traces, for both the GSM and the WiFi.

Focusing on the GSM communication technology, we observe that RSS values span between -110 dBm and 10 dBm, with a quantile 0.5 value (median) of about -49 dBm. For the WiFi, instead, the domain size is shorter: the RSS values span between -91 and -48 dBm, with a median value of about -80 dBm.

We used the distribution of the data reported in Figs. 6–8 to generate realistic synthetic data, where the following part of our study has been performed.

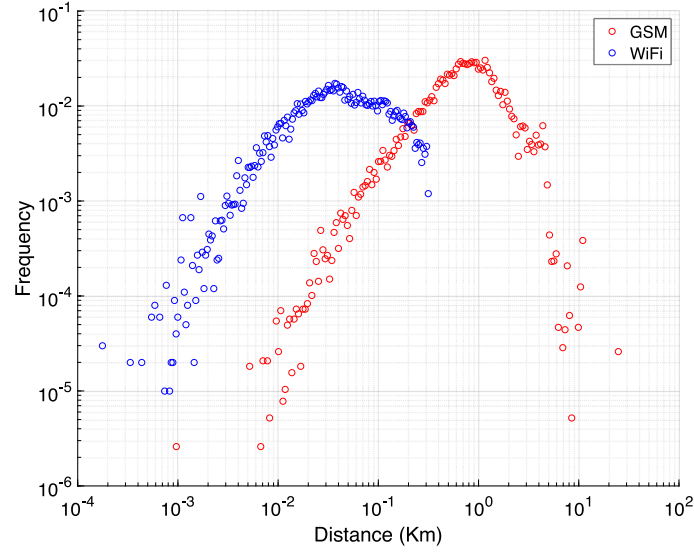


Fig. 7. Probability Distribution Function of the distance between MT and the anchors.

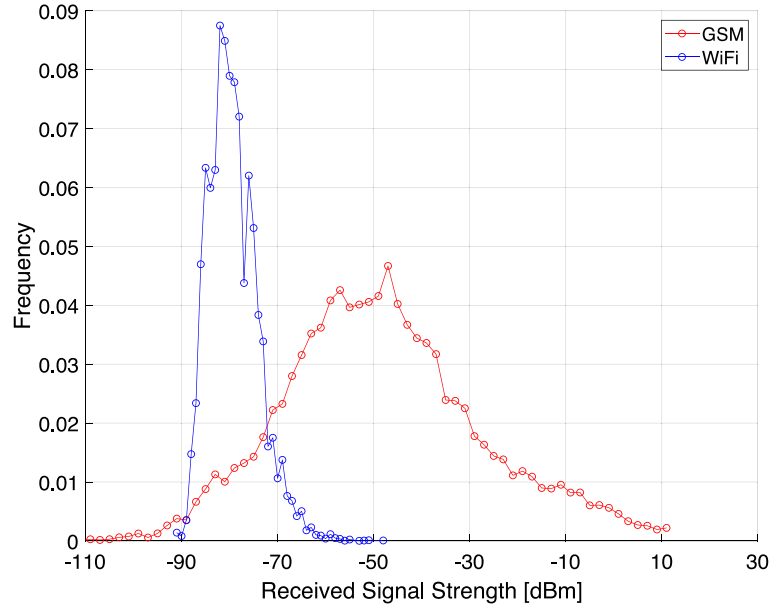


Fig. 8. Received Signal Strength (RSS) estimated from all the BSs for all the paths.

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}) computed from the crowdsourced information (WiFi and GSM anchors). As already discussed before, combining the distances between the anchors and the mobile terminal does not provide a precise location, but only an uncertainty area (recall Fig. 3). To estimate the size of the aforementioned area, we combine all the traces, and, for each time slot ($T = 166$ ms), we compute the distance error d_e between X_{GPS} and Y_{est} , by using Eq. (4). Fig. 9 depicts the statistical analysis associated to d_e : we consider the Cumulative Distribution Function (CDF) associated to d_e , while varying the weight w between the WiFi and the GSM location estimation (recall Eq. (4)). We recall that when $w = 0$, only the location estimation provided by the WiFi is considered, while when $w = 1$ only the location estimation provided by the GSM is taken into account. As a result, when $w = 0.5$, the location estimations provided by the two communication technologies are weighted in the same way.

The inset figure provides the details about the quantile 0.9 of the position estimation error while increasing the value of w .

We anticipate that, despite the quantile 0.9 of the position estimation error appears to be quite high, our solution leverages multiple consecutive position estimations, thus significantly reducing such a single-shot error. More details on this aspect will be provided in the following sections.

The final objective of our analysis is to define a decision threshold (Φ), useful to discriminate between anomalies and consistent estimations for the current location.

Definition. We define *Decision Threshold* (Φ) the maximum acceptable error between the current GPS position (X_{GPS}) and the estimated one (Y_{est}) to declare the two positions as consistent, i.e., $d_e < \Phi$.

We consider such a threshold (Φ — line 20 of Algorithm 1) as the quantile 0.9 of the error as depicted by the inset figure of Fig. 9. When $w = 0$ (i.e., only the information of the WiFi anchors are used), the threshold Φ is equal to 424 m. The value of the threshold, i.e., the

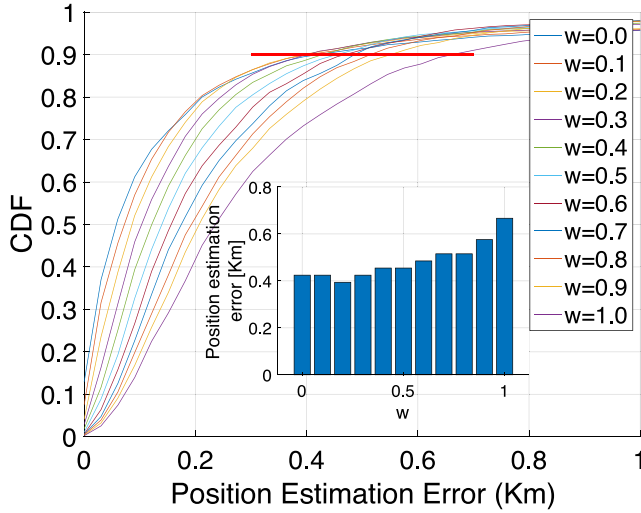


Fig. 9. Position estimation error: Cumulative Distribution Function (CDF) of the position estimation error d_e . The inset figure shows the quantile 0.9 of the position estimation error, with increasing values for w .

quantile 0.9 associated to the position estimation error, increases as w increases. In fact, without loss of generality, since GSM has a higher transmission range than WiFi, the location estimate provided by the GSM will be affected by an error d_e that is higher than the one of the WiFi, leading the final location estimate to be characterized by a larger error. When $w = 1$ (i.e., only the GSM is considered), the position estimation error is the highest, i.e., 666 meters far from the MT position.

Therefore, we will assume the GPS position as trusted if its distance d_e from the one estimated leveraging the GSM and WiFi anchors is less than Φ . From now on, our analysis will consider only three reference cases:

- *WiFi-only*, being equal to the case $w = 0$;
- *WiFi+GSM*, being equal to the case $w = 0.5$;
- *GSM-only*, being equal to the case $w = 1$.

In general, these cases correspond to a densely-populated area, a medium populated area, and a rural area. However, these scenarios are only a reference, and they do not restrict the applicability range of our solution. Indeed, the heterogeneity of the collected data and the related configuration parameters, which have been calibrated accordingly, assure that our solution works in the extreme scenarios of both rural and densely populated areas — as well as in all the possible configurations included in that range that a standard user can experience.

The PDF of the distance estimation error d_e for the three aforementioned configuration are provided in Fig. 10.

The three scenarios exhibit a common behavior. As shown by Fig. 10, there are a few location estimates characterized by either a very small or a very high error, while there are peaks located at 0.015, 0.090, and 0.135 meters for the WiFi, WiFi+GSM, and GSM scenarios, respectively.

We fitted the empirical data in Fig. 10 according to the MLE criterion, obtaining the following continuous probability distribution functions:

- 1. Generalized Extreme Values Distribution, for the case of $w = 0$ and $w = 1$, yielding Eq. (5).

$$GEQ(\mu, \sigma, \epsilon) = e^{-(1+\epsilon \cdot s)^{\frac{1}{\epsilon}}}, \quad (5)$$

where $\epsilon = 1.068$ (0.354) is the shape parameter for $w = 0$ ($w = 1$), $\mu = 0.042$ (0.193) is the location parameter for $w = 0$ ($w = 1$), $\sigma = 0.052$ (0.141) is the scale parameter for $w = 0$ ($w = 1$), and $\epsilon = \frac{(x-\mu)}{\sigma}$ is the standardized variable [49].

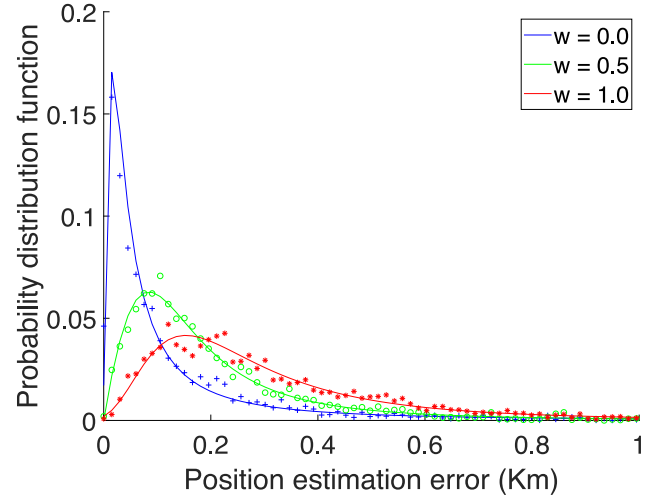


Fig. 10. Position Estimation Error of the three reference configurations of our spoofing detection algorithm, and best-fit discrete functions, according to the MLE criterion.

- 2. Log Logistic Distribution, for the case of $w = 0.5$, yielding Eq. (6).

$$LLD(\alpha, \beta) = \frac{x^\beta}{\alpha^\beta + x^\beta}, \quad (6)$$

where $\alpha = 0.521$ and $\beta = -1.872$ are the scale and shape parameters, respectively.

These continuous probability functions will be adopted in the remainder of this paper to model the position estimation errors.

We recall that the single event $d_e > \Phi$ is not enough to declare a spoofing attack. Indeed, looking at Fig. 9, we observe the presence of many points crossing the Φ threshold (represented by the red straight horizontal line). These transients can be filtered out resorting to the already introduced *discriminate_spoofing_from_anomalies()* function (recall lines 26–31 in Algorithm 1). More details will be provided in Section 6.

Finally, the last parameter to be considered is μ , introduced by Eq. (1). It represents the mean of the exponential distributions adopted to weigh the contributions of the anchors to the centroid computation for either the GSM or the WiFi communication technology. Indeed, we recall that each anchor contributes via its Received Signal Strength (RSSs). Specifically, the value of μ in Eq. (1) provides the relative weight of the anchor with the strongest RSS compared to the others (weaker). The smallest (i.e., close to 0) the value of μ , the most the measurements coming from close anchors will be considered for the location estimation compared to the contributions coming from the far anchors. Fig. 11 shows the error (d_e) as a function of μ , with $1 \leq \mu \leq 100$, considering all the collected traces considering the two communication technologies (GSM and WiFi).

We observe that the value of μ slightly affects the final error. This phenomenon affects more the GSM technology, while for the WiFi the error is almost constant and independent of the choice of μ .

In the remainder of this paper, we set $\mu = 20$, since it represents an acceptable trade-off (minimum error) for all the collected traces.

Final considerations should be done concerning the morphological distribution of the anchors. Indeed, the considered playground (either rural or urban) might significantly affect the performance of our detection solution. Fig. 12 shows the estimated position error (computed with respect to the actual one) as a function of the number of in-range anchors, where we considered the median values of the error associated to the vehicle position estimation in all the considered trajectories. Finally, we interpolated the median values with a second-grade polynomial function (red and blue lines), to highlight the trend.

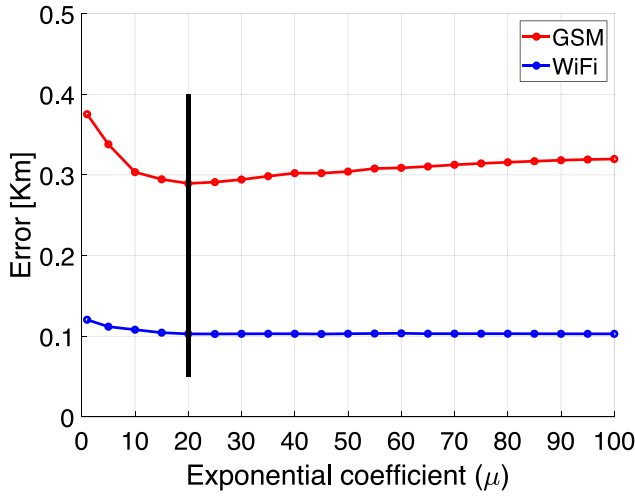


Fig. 11. Location Estimation error for the GSM-only and WiFi-only configurations when varying the value of the coefficient μ in Eq. (1).

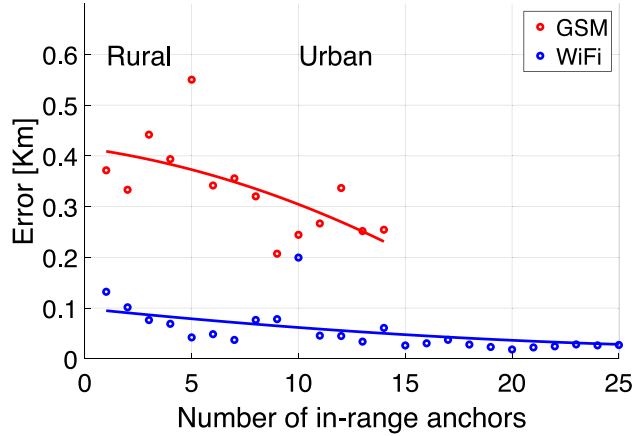


Fig. 12. Location estimation error as a function of the number of in-range anchors, considering either GSM or WiFi anchors.

The results confirm that the adoption of the GSM-only configuration provides less accurate location estimations compared to the WiFi-only configuration. Nevertheless, our analysis highlights that increasing the number of in-range anchors, i.e., moving from rural to urban scenarios, helps reducing the localization error. As an example, when the vehicle is in a rural scenario, the localization error can only be mitigated by the presence of a WiFi anchor (and actually with its position), otherwise the GSM technology cannot reduce the error to less than about half a kilometer. Conversely, when moving to more anchor-dense areas (urban scenarios), the GSM can help reducing the localization error (less than 300 meters), while the WiFi technology can reduce the error up to few tens of meters.

To the best of our knowledge, a strict categorization of an area as either rural or urban as a function of the number of in-range anchors is not available in the literature, and it would be also inconsistent with the actual deployment strategies, depending on a lot of factors (density of people in the area, time of the day, period of the year, to name a few). Nevertheless, it is important to highlight the impact of the number of anchors in the detection process. Therefore, we associated the label *rural* to a “few” in-range anchors, independently from the specific technology (either GSM or WiFi), while we refer to *urban* scenarios, when the number of in-range anchors is “high”. Without loss of generality, we leave to the readers the decision of labeling a given number of anchors as either *rural* or *urban*.

6. Spoofing detection

In this section, we measure the performance of our solution in terms of *Detection delay*, i.e., the time to detect an ongoing GPS spoofing attack, by assuming different crowd-sourced scenarios: WiFi only, WiFi + GSM, and finally, GSM only. We start our analysis by considering a benign scenario, with no ongoing spoofing attacks (Section 6.1), and we evaluate the best configuration of the parameters to guarantee the minimum number of false spoofing alarms. Next, we show the performance of our proposed solution against both emulated and real GPS spoofing attacks (see Section 6.2).

6.1. System calibration in a benign scenario

We recall that our spoofing detection solution mainly consists of comparing the GPS position with the one coming from the crowd-sourcing, i.e., WiFi and GSM. As discussed in the previous sections, the crowd-sourced position estimation is affected by an error that depends on the environment, i.e., the number of deployed in-range anchors, technology (WiFi, WiFi+GSM, GSM), and Received Signal Strength (RSS). Moreover, the vehicle speed affects the overall process; indeed, the density of both GSM and WiFi fluctuated during the time, and, eventually, this affects the precision of the crowd-sourced position estimation.

Definition. We define *anomaly* an arbitrary long sequence of consecutive events, such that $d_e(t) \geq \Phi$, i.e., the distance between the crowd-sourced position (Y_{est}), and the GPS one (X_{GPS}) at time t , is greater or equal to Φ .

Depending on the selection of the decision threshold, the anomalies exhibit a different pattern. In the following, we evaluate their statistical significance. Specifically, Fig. 13 shows the longest duration of the anomalies (worst cases) as a function of the decision threshold under benign (no-spoofing) conditions. Firstly, we observe that a small decision threshold (e.g., 10 m) involves a large anomaly duration (roughly between 30 s, for the case of the WiFi-only configuration, and 800 s, for the case of the GSM-only configuration). This confirms that, when the accepted error (Φ) between the GPS position and the crowd-sourced one is small, the duration of the anomaly is likely to be larger. Conversely, when $\Phi \approx 1$ Km, the duration of the anomaly is always less than 1 s. We recall that we sampled both the GSM and the WiFi technologies every 0.16 s (on average), and therefore, we collected about 6 samples per second for each technology.

The values reported in Fig. 13 for the three operation modes of our spoofing detection algorithm are key to understand its performance, in particular concerning False Positives occurrences. Being aware of the maximum length of an anomaly in a benign scenario, it is possible to guarantee a false positives probability ($FP = 0$) by setting the detection delay to 10 s, 30 s, and 800 s for the WiFi, WiFi + GSM, and GSM-only configuration, respectively.

However, fixing such thresholds could lead to an overwhelming detection delay, especially when the GSM communication technology is considered. By setting the detection delay to smaller values, the end-user would have to accept the presence of non-zero false positives, that would cause the generation of false alarms by the system. We studied the relationship between the decision threshold, the detection delay, and the false positives probability for each of the three configurations of our scheme, and the results are reported in Figs. 14(a), (b), and (c), respectively.

Overall, by fixing the decision threshold, increasing the anomaly duration will lead to reduced false positives. In fact, the longer the duration of the anomaly for a given decision threshold, the lower the probability to observe a sequence of consecutive events such that $d_e \geq \Phi$. As previously mentioned, given that location estimates performed via WiFi tend to be closer to the MT position, the duration of the

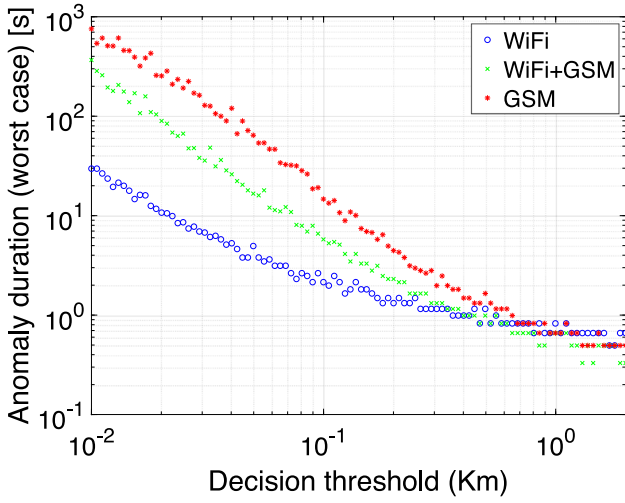


Fig. 13. Detection Delay: Time to detect a GPS spoofing attack, assuming as decision threshold the maximum burst length (sequence of anomalies) in benign conditions.

anomaly increases when considering the GSM, both in combination with the WiFi or in the GSM-only configuration.

Therefore, we observe that selecting the desired maximum level of false positives, a limited choice of the pairs (decision threshold, anomaly duration) are available to satisfy this requirement. The lower bound can be set to the selection $FP < 0.01$, i.e., the pairs (decision threshold, anomaly duration) that can guarantee a maximum percentage of False Positives strictly less than 1%, represented by the blue lines in each of the sub-figures of Fig. 14. Fig. 15 shows the regions achieving such a lower bound, where (A), (B), and (C) represent the regions for the WiFi, WiFi+GSM, and GSM technology, respectively.

As an example, let us consider the GSM-only configuration. Selecting the pairs (decision threshold, anomaly duration) above the red line, e.g., $\Phi = 0.08$ Km and an anomaly duration of 80 s, actually provides $FP < 0.01$, achieving very high reliability. If we select a shorter anomaly duration, by keeping the same decision threshold, more false positives will appear (region (C) with $FP > 0.01$), and the system will become less reliable; for instance, an anomaly duration equal to 40 s cannot guarantee $FP < 0.01$ with $\Phi = 0.08$ Km.

Similar considerations apply also for the WiFi+GSM and WiFi-only configuration. Nevertheless, we can consider smaller values for both the anomaly duration and the decision threshold when considering different technologies (WiFi+GSM and WiFi), while being able to guarantee $FP < 0.01$. For instance, assuming a detection threshold Φ of 0.07 km, while the lower bound value of the anomaly duration for the GSM-only configuration is 79.68 s, it decreases to 23.4 s for the WiFi+GSM configuration and 6.86 s for the WiFi-only, still guaranteeing $FP < 0.01$.

Overall, the selection of the decision threshold affects the minimum delay to detect the spoofing attack, i.e., the *reaction speed* of the system. More details will be provided in Section 6.2.

6.2. Detecting GPS spoofing attacks

In this section, we study the performance of our solution when the adversary performs a GPS spoofing attack.

We recall that during a spoofing attack the position of the MT X_{GPS} will be very different from the estimated position Y_{est} computed via crowd-sourced information. Thus, a single anomaly event will be generated at each time slot t . However, as already discussed in the previous sections, a single anomaly is not enough to declare a GPS spoofing attack, given that sequence of anomalies (bursts) are also present in benign scenarios. GPS spoofing events can be efficiently

and effectively detected by observing both the temporal sequence of anomalies and the anomaly duration. Thus, the maximum anomaly duration has a direct impact on the capability of the system to timely detect the attack.

To investigate the spoofing detection delay, we spoofed the position of a static MT, by emulating an attack that smoothly changes the GPS position of the MT from the actual one. Fig. 16 shows the spoofing detection delay when the probability of false positives is set as per Fig. 15, i.e., $FP \leq 0.01$, while changing the value of the decision threshold, in each of the three different configurations of the algorithm, i.e., GSM, GSM+WiFi, and WiFi.

Considering the GSM-only configuration, when the decision threshold is small, e.g., 0.02 Km, a time of 295.68 s is needed to raise an alarm, while guaranteeing $FP \leq 0.01$. Such a large time is due to the duration of the anomalies that have been observed in a benign scenario for the same detection threshold, leading to an increased observation time before declaring the spoofing attack. By increasing the decision threshold, e.g., $\Phi = 0.08$ km, the spoofing detection delay can be reduced to about 35.31 s, leading to a more *timely* detection delay. However, the aforementioned decision threshold also affects the dimension of the area where the spoofing is detected. Indeed, the attacker will have to divert the MT for a larger distance than the previous case to let the system detect the attack. Assuming a smaller decision threshold, when the GSM-only configuration is the only possible choice (e.g., in a rural area where no WiFi networks are available), the only possible option to reduce the detection delay is to accept false-positive events, i.e., by recalling Fig. 15, moving within the region (C). For instance, selecting the decision threshold equal to $\Phi = 0.07$ Km, and taking a decision after an anomaly duration of 60 s, the system would be characterized by $FP > 0.01$. Where WiFi networks are available, it could be possible to reduce the decision threshold and the anomaly duration, and guaranteeing high reliability, i.e., $FP < 0.01$. For instance, by using the WiFi+GSM configuration, by setting a decision threshold of $\Phi = 0.07$ Km, it could be possible to detect a spoofing attack in 15.675 s, while still achieving $FP < 0.01$. The reactivity of the scheme further improves when using the WiFi-only configuration, where a GPS spoofing can be detected in approximately 7.425 s with $FP < 0.01$.

Finally, note that similar considerations apply to the WiFi+GSM and the WiFi-only configuration: if larger false positives can be accepted ($FP > 0.01$), both the detection delay and the decision threshold could be made arbitrarily small. At the same time, the opposite is also true: lower false positives (less than $FP = 0.01$) can be achieved, at the expense of an increase in the decision threshold and detection delay.

Overall, the shortest detection delay that achieves $FP < 0.01$ can be obtained with a decision threshold equal to 0.04 Km, and it allows the system to detect a GPS spoofing attack in approximately 6.27 s. Assuming a shorter decision threshold is desired, e.g., $\Phi = 0.01$ Km, the spoofing detection delay increases to 13.53 s, to still achieve $FP < 0.01$.

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 three different pre-defined paths, as depicted in Fig. 17. Our experiments are based on the following multi-step 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 [50], making the path consistent with actual roads, intersections, and turns.
2. **Data format conversion.** The output of Google Earth Pro [50] (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 [51] to convert the fake path to the standard NMEA GGA stream format [52].

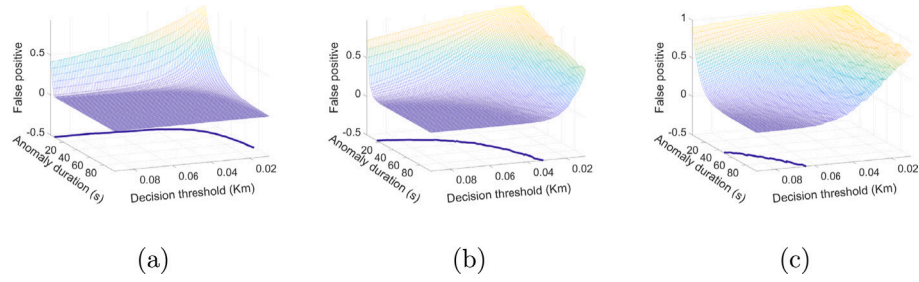


Fig. 14. Relationships between Decision Threshold, Anomaly duration, and Probability of False Positives in the three scenarios: (a) WiFi, (b) WiFi + GSM, and (c) GSM.

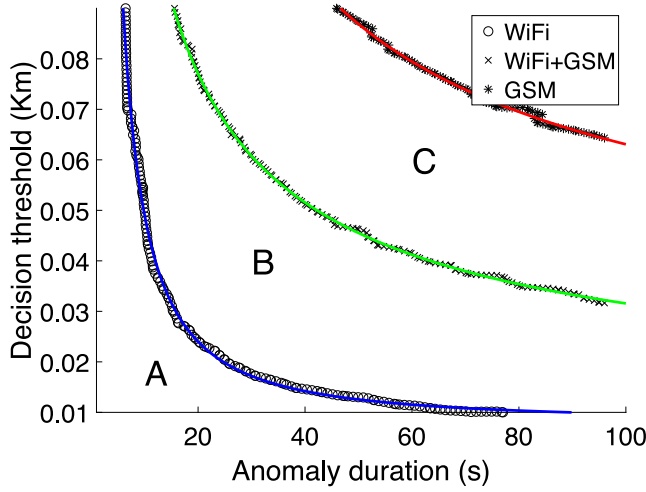


Fig. 15. Decision boundaries as a function of the anomaly duration, $FP = 0.01$. Areas A, B and C represent the pairs (Decision Threshold - Anomaly duration) with $FP < 0.01$.

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, the GSM BSs, and the WiFi APs into a file.

Note that spoofing a static target is equivalent to spoofing a mobile target. Indeed, since our spoofing detection solution is based on the distance $d(o)$ between the actual position p and the estimated one p' , the following Eq. (7) yields:

$$d(p + \bar{\Delta}, p') = d(p, p' + \bar{\Delta}), \quad (7)$$

where $\bar{\Delta}$ is a random movement. We observe that the aforementioned relation yields independently of the movement affecting either the actual (p) or the estimated point (p'), and therefore, it is not affected by the fact that p is either static or moving.

Fig. 17 shows the details of our real spoofing attack. We spoofed the position of the MT (red diamond in Fig. 17) along three different paths (represented with black circles, crosses, and squares, respectively), overlapping only in the initial MT position. It is possible to notice that, after some time, the spoofed GPS position moves away from the real one.

Overall, the performance of our spoofing detection algorithm in detecting the three spoofing attacks depends on the particular configuration of the scheme, concerning the decision threshold (Φ), false positives, and detection delay desired by the end-user.

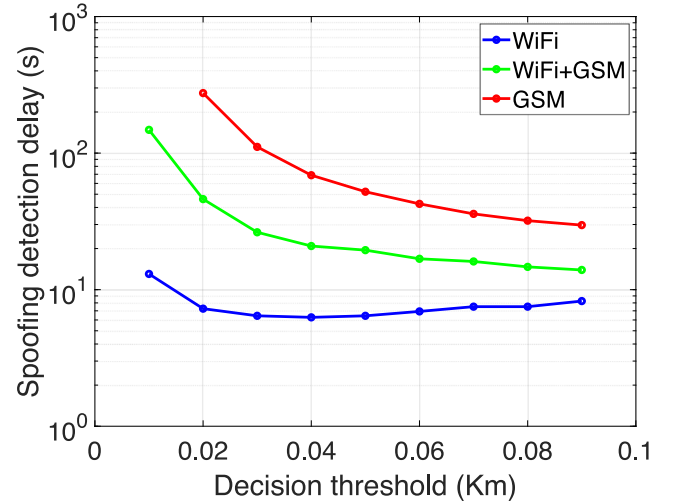


Fig. 16. Spoofing Detection Delay while varying the Decision Threshold. Assuming $FP < 0.01$, the shorter the decision threshold, the higher will be the number of consecutive anomalies, and therefore, the duration of the anomalies to be observed to declare a GPS spoofing attack.

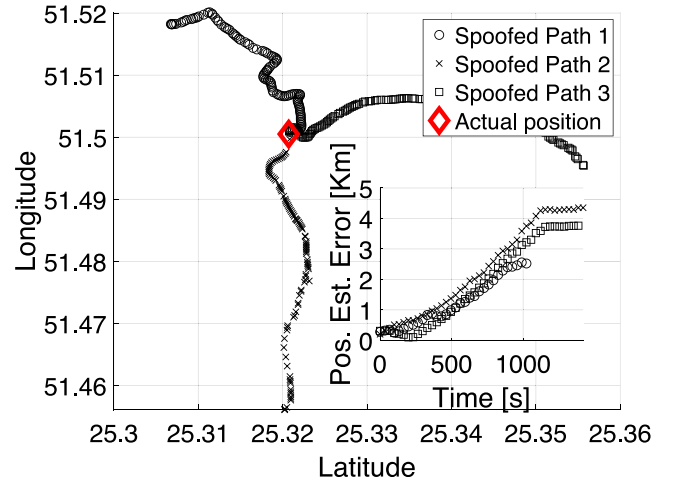


Fig. 17. Real GPS spoofing attack. We spoofed the real MT position (red diamond) according to three different paths, represented by black circles, crosses, and squares, respectively. The inset figure shows the position estimation error, i.e., the difference between the position indicated by the (spoofed) GPS on board of the MT and the real MT position.

When WiFi networks are available, the spoofing detection algorithm can be configured with $w = 0$, to achieve smaller detection delays and shorter detection thresholds, while still maintaining high reliability (see Section 6). When there are few WiFi networks (or none), the GSM can still be used to achieve GPS spoofing detection, but with

increasing detection delays and, generally, higher detection threshold. Better performance (in terms of the aforementioned parameters) can still be obtained but at the expense of an increased false positives probability.

Despite the WiFi-only configuration is far more efficient than the WiFi+GSM and the GSM-only configurations, the choice of the adopted configuration by the end-user should be carried out smartly, by considering environmental and contextual factors. While urban areas could be very dense of WiFi networks, thus resulting in a reliable location estimation, suburban and rural areas usually include few (or none) WiFi networks, and thus resorting to the WiFi+GSM and GSM-only configuration could be a better strategy.

8. Malicious GSM and WiFi anchors

In the following, we consider an adversary that, being aware of our solution, can add a certain number of malicious anchors (WiFi + cellular GSM base stations) to the pool of the trusted ones, e.g., through the attack illustrated in [53,54], to name a few. The intuition is that, being able to deploy or spoof its anchors according to the spoofed path, the adversary might increase its chances to successfully perform the spoofing attack, using tools such as SkyLift [55]. We consider one or more anchors as malicious if they collude with the spoofed GPS signal. Indeed, the GPS spoofing attack is successful if, by deploying a given number of malicious anchors, the location estimate provided by our solution is close enough to the (spoofed) position, i.e., the distance between the GPS position and the location estimate is less than the detection threshold.

First, we remark that this attack likely requires the adversary to be aware of the path of the target Mobile Terminal (MT). Indeed, even if the adversary generates fake WiFi anchors on-the-fly, such anchors will not be included in the database that the MT downloaded before the travel, making any real-time attack ineffective. In addition, the adversary should be able either to *poison* the online databases, by inserting fake anchors exactly in the desired location, or to record the existing WiFi and/or GSM signals at the target (spoofed) path and replay them to the moving vehicle, during the path. We recall that our measurements show an average number of in-range WiFi and GSM base stations equal to 4 and 12, respectively. In addition, we stress that each spoofed anchor should be previously added to the database and it should be validated by other users, e.g., by majority voting criteria, and therefore, to perform a massive injection, the adversary has to scale up the number of fake injecting sources. In addition, this malicious behavior can be easily detected at the server-side, by exploiting the community of the actual users, and therefore, malicious accounts performing the injections can be easily detected and banned. Thus, these requirements are usually very hard to achieve, unless the adversary is in full control of the databases.

In addition, we assume as impractical to deploy the malicious anchors along the path of the mobile node. Indeed, a realistic deployment for an adversary spoofing multiple GSM and WiFi transmitters (at the same time) involves hiding a malicious device on-board of the vehicle. In principle, the tools to generate such an attack are already available. The adversary can use a device featuring an SDR and a Raspberry Pi, spoofing both WiFi and GSM, at the same time, forging fake SSID as beacons by spoofing the legitimate ones, and dynamically overpowering surrounding APs, as previously introduced by [56].

Now, as a worst-case assumption, let us assume that the adversary is aware of both of the above requirements. Therefore, we assume the adversary deploys an increasing number of on-board malicious anchors (through a single device), and we estimate the probability of a successful spoofing attack. For all the collected measurements, we emulate the presence of (a given number of) malicious anchors, colluding with the GPS spoofed position and trying to change the vehicle path from its intended one. Moreover, within the scope of our investigation, we assume that the adversary does not know in advance the number of

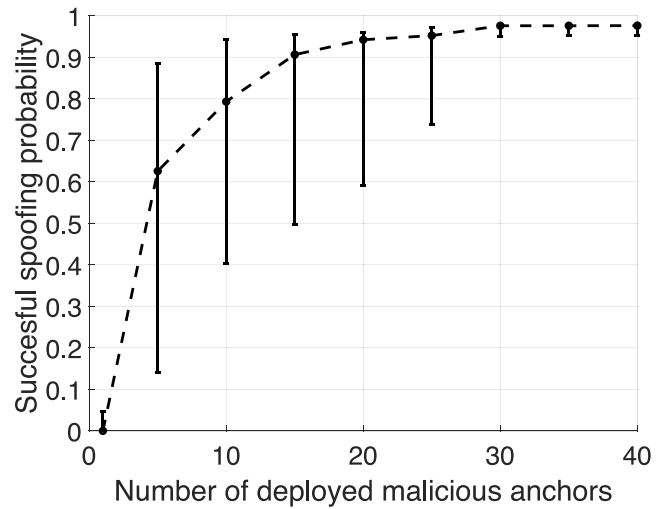


Fig. 18. Probability of a successful spoofing attack, while varying the overall number of malicious $N/2$ (GSM) + $N/2$ (WiFi) anchors, for a total of N anchors.

(trusted) in-range anchors, and therefore, it deploys a budget of N increasing malicious anchors that can be dynamically allocated to act as either WiFi or GSM. Indeed, the vehicle will experience a different number of WiFi and GSM anchors when moving along its path, and therefore, without loss of generality, we assume the adversary allocates $N/2$ malicious anchors to the WiFi technology and $N/2$ to the GSM, for a total of N . The target device will experience two main (inconsistent) positions, i.e., one estimated from the malicious clique of anchors and, the other one, from the trusted anchors. Although this scenario (two or more disjoint cliques) might be enough to infer on the presence of an attack, in the following, we consider a majority voting decision, and therefore, the spoofed device will decide to trust the position supported by the bigger clique. Fig. 18 shows the results of our experimentation, in terms of successful spoofing probability as a function of the increasing number of deployed malicious anchors. As suggested by the intuition, the higher the number of malicious anchors deployed by the adversary, the higher the probability that the spoofing attack will be successful. Indeed, given the number of malicious anchors deployed by the adversary, the probability of a successful attack depends on the number of anchors already available in the scenario and the selected operation mode, i.e., WiFi, WiFi+GSM, or GSM. Thus, to maximize the chances of a successful attack, the attacker has to deploy an increasing number of anchors, to guarantee that the signals emitted by these anchors are dominant compared to the legitimate ones. Overall, we observe that, even through a single device, the adversary should deploy a total number of malicious anchors strictly greater than 27 to have at least a 90% probability ($p \geq 0.9$) to successfully spoof the target device without being detected. On the one hand, this is a limitation of our methodology, which cannot be overcome, due to the untrusted nature of the adopted communication technologies. Nevertheless, we notice that deploying such a huge number of malicious anchors through a single device can be very expensive. Indeed, while cheap devices such as the Raspberry Pi may allow spoofing multiple technologies at once, spoofing 27 WiFi access points and GSM base stations at the same time by resorting to a single tiny device (or a tiny SDR) might require a significant amount of resources. The SDR should continuously switch among the different technologies without the possibility to perform a fully parallel attack. Moreover, as we mentioned in the manuscript, an effective attack should take into account multiple anchors, and therefore, we should assume the SDR being able to emulate tens of anchors in parallel. To the best of our knowledge, this requires either a more bulky hardware (hardly to be hidden) or a state-level attacker.

9. Discussion

In the following, we discuss our solution in terms of performance, effectiveness, efficiency, trade-offs, adaptability, and robustness.

Performance. As highlighted by our analysis in Matlab, our solution guarantees the detection of a GPS spoofing attack in approximately 6 seconds (while requiring false positives rate to be smaller than 0.01, and assuming the use of the WiFi-only configuration).

However, it is worth noting that our results are affected by the anchors' distribution, and therefore, different deployment of the anchors 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 *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 our results would still hold, as they are, in a variety of scenarios.

Effectiveness. We observe that, assuming an average car trip distance of 20 Km [57], the detection error (the ratio between the length of the diverted path and the overall trip) is less than 0.8%. Such a value becomes significantly smaller when assuming the case of a truck, driving a standard trip of 1062 Km (96 Km/h for 11 hours — duty limit in the US): in such a case, the detection error is about 0.015%.

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 the WiFi APs and, therefore, only minor processing is required to implement our solution. Moreover, our solution can be directly integrated into the vast majority of the smart navigation systems, since it does not require any special hardware and it resorts only on already available information: GPS coordinates, data traffic from the WiFi APs, and data traffic from the BSs, the latter ones being a requirement for all the modern connected navigation systems.

Trade-offs. Our solution can be further optimized based on different requirements, regarding the scenario and the end-user side. Indeed, while the optimal setup of the algorithm guarantees quasi-zero false positives ($FP < 0.01$), a much smaller detection delay can be provided relaxing the previous requirement and accepting more false positives. This can be achieved by considering a smaller value for the decision threshold Φ , or shorter burst lengths.

Adaptability. Our solution can be configured in order to adapt to any scenario, being it *rural*, *suburban*, or *urban*. In an urban scenario, dense of WiFi networks, the WiFi-only configuration can be used to obtain minimum detection delays and a reduced decision threshold. In a suburban scenario, where the number of WiFi networks is less than the previous one, the WiFi+GSM configuration can be used to provide increased redundancy, while still achieving small detection delays and decision threshold. Finally, in a rural scenario, where WiFi networks are very unlikely, mobile cellular networks BSs can be adopted in the GSM-only configuration to still provide GPS spoofing detection, although with higher detection delays. We highlight that Doha (and Qatar in general) is a very heterogeneous place, representing one of the most challenging (and interesting) locations for performing such measurements. Indeed, the number of in-range anchors spans between 0 and 50, as depicted in Fig. 6, due to the very heterogeneous areas considered in our driving, e.g., downtown, sub-urban, and desert. To fully leverage these features, we drove a car for 5.5 hours and 196.474 km, moving across different districts of the city and experiencing very different *measurement conditions*, either characterized by a very large number of anchors (i.e., mobile cellular base-stations or WiFi access points) or by limited anchors availability (desert). Hence, while on the one hand, we recognize that our results could be affected by a small bias, on the other we believe that our measurements take into account a very heterogeneous measurement scenario that can fit several real use-cases.

Further, we would like to highlight that our paper aims to propose a methodology to use untrusted sources, such as the mobile cellular

network and the WiFi access point locations, as a means to identify GPS spoofing attacks. Therefore, in a real use-case scenario, any network administrator can apply our logic *without any change*, to any new set of data, and inferring performance indexes specific to the intended deployment region.

Robustness. Our solution is very robust against attackers that, being aware of its deployment, can deploy malicious anchors. As shown by the analysis in Section 8, to guarantee a successful attack, the adversary would have to know the path of the MT in advance and to deploy an overwhelming amount of malicious anchors, that must overcome, in numbers, the number of legitimate in-range anchors. Even in this unlikely scenario, the adversary will be successful if and only if it deploys in advance an overwhelming set of malicious anchors, to perform a continuous spoofing over both the time and space. Overall, this would require the adversary to register its anchors to online crowd-sourcing platforms, and deploy many anchors (see Fig. 18). This would require not just a skilled adversary, but likely also physical room on board of the vehicle following the target and a significant budget.

Ease of Integration. When the objective of the adversary is to move the path of the target vehicle by just a few meters (i.e., less than 5 meters), our solution might not be able to detect ongoing GPS spoofing attacks. However, we do believe that those scenarios are limited, in practice. This is acknowledged by the authors in [58], reporting that GPS *effective spoofing range* is 40–50 meters. In all these cases of practical applicability, our solution can be effective to detect ongoing GPS spoofing attacks. Indeed, as shown by our thorough investigation in Section 6.2 of the manuscript, through an appropriate selection of the detection time, our solution can detect GPS spoofing attacks moving the vehicle in a range of only 5 meters, with a false-positive probability less than 0.01.

Considering that our solution is mostly software-based, it is easily deployable in most connected vehicles independently from the sensors available on-board. Moreover, when sensors are available, it can be easily integrated with techniques based on sensor fusion such as the ones presented by the authors in [58], to boost GPS spoofing detection effectiveness on autonomous vehicles.

10. Conclusion

In this work, we have introduced and evaluated the effectiveness and limitations of a novel technique to detect and mitigate GPS spoofing attacks, leveraging existing broadcast transmissions from mobile-cellular base stations and WiFi access points. Although our reference scenario assumes connected vehicles, our results are general, and they might be applied to any entity (including drones) moving in any area covered by any density of anchors, thus being deployable in either rural or densely populated regions.

All the obtained results are supported by an extensive experimental campaign. In particular, we have generated a substantial dataset by driving around a car for 196 km and more than 5 h. The results show that our solution is effective in detecting an ongoing GPS spoofing attacks, while requiring little overhead and incurring in a very low false positive rate. We achieved a false positive rate of less than 0.01, while experiencing a detection delay of about 6 s when using only signals from the WiFi networks. Our solution can be finely tuned, e.g., trading-off a slight increase in false positives with a reduced detection delay, or taking into account signals from the mobile cellular networks when the number of WiFi is small or null. The novelty of the proposed approach, the excellent achieved results, the extent of the expected impact, the availability to the research community of the collected data, and the discussed possible extensions of this work, do pave the way for further research in this field.

CRediT authorship contribution statement

Gabriele Oliveri: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Savio Sciancalepore:** Conception and design of study, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Omar Adel Ibrahim:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data. **Roberto Di Pietro:** Conception and design of study, Analysis and/or interpretation of data, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We have released the data at <https://github.com/cri-lab-hbku/gps-spoofing-detection-cellular>

Acknowledgments

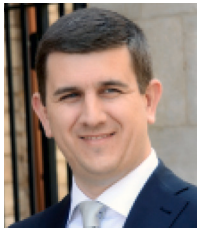
The authors would like to thank the anonymous reviewers that, with their comments, helped improving the quality of the manuscript.

This publication was partially supported by awards NPRP12S-0125-190013, NPRP-S-11-0109-180242, NPRP12C-0814-190012 and GSRA6-1-0528-19046, from the QNRF-Qatar National Research Fund, a member of The Qatar Foundation. This work has been partially supported also by the INTERSECT project, Grant No. NWA.1162.18.301, funded by Netherlands Organisation for Scientific Research (NWO). 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

- [1] J. Kwak, Y. Sung, Autonomous UAV flight control for GPS-based navigation, *IEEE Access* 6 (2018) 37947–37955.
- [2] S. Kuutti, S. Fallah, K. Katsaros, M. Dianati, F. McCullough, A. Mouzakitis, A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications, *IEEE Internet Things J.* 5 (2) (2018) 829–846.
- [3] L. Hsu, Y. Gu, S. Kamijo, Intelligent viaduct recognition and driving altitude determination using GPS data, *IEEE Trans. Intell. Veh.* 2 (3) (2017) 175–184.
- [4] L. Van Huynh, J. den Hartog, L. Zannone, Security and privacy for innovative automotive applications: A survey, *Comput. Commun.* 132 (2018) 17–41.
- [5] A. Koubaa, B. Qureshi, DroneTrack: cloud-based real-time object tracking using unmanned aerial vehicles over the internet, *IEEE Access* 6 (2018) 13810–13824.
- [6] D. He, Y. Qiao, S. Chan, N. Guizani, Flight security and safety of drones in airborne fog computing systems, *IEEE Commun. Mag.* 56 (5) (2018) 66–71.
- [7] B. Sheehan, F. Murphy, M. Mullins, C. Ryan, Connected and autonomous vehicles: A cyber-risk classification framework, *Transp. Res. A* (2018).
- [8] J. Petit, S.E. Shladover, Potential cyberattacks on automated vehicles, *IEEE Trans. Intell. Transp. Syst.* 16 (2) (2015) 546–556.
- [9] N. Tippenhauer, C. Pöpper, K. Rasmussen, S. Capkun, On the requirements for successful GPS spoofing attacks, in: *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS '11*, 2011, pp. 75–86.
- [10] L. Huang, Q. Yang, Low-cost GPS simulator – GPS spoofing by SDR, in: *DEFCON '15*, 2015.
- [11] G.D.L. Torre, P. Rad, K.R. Choo, Driverless vehicle security: Challenges and future research opportunities, *Future Gener. Comput. Syst.* (2018).
- [12] D. He, S. Chan, M. Guizani, Communication security of unmanned aerial vehicles, *IEEE Wirel. Commun.* 24 (4) (2016) 134–139.
- [13] Cyware Labs, 2022, <https://medium.com/cyber-journal/seven-car-manufacturers-hit-by-gps-spoofing-attacks-bc0f400f49ec>.
- [14] Opencellid, 2022, <https://opencellid.com>.
- [15] WiGLE, 2022, <https://wigle.net/>.
- [16] Dataset, 2022, Link 1: <https://cri-lab.net/drive-me-not/>, Link 2: <https://github.com/cri-lab-hbku/gps-spoofing-detection-cellular>.
- [17] G. Oliveri, S. Sciancalepore, O. Ibrahim, R. Di Pietro, Drive me not: GPS spoofing detection via cellular network, in: *Proceedings of the 12th ACM Conference on Security & Privacy in Wireless and Mobile Networks, WiSec '19*, 2019.
- [18] K. Zeng, Y. Shu, S. Liu, Y. Dou, Y. Yang, A practical GPS location spoofing attack in road navigation scenario, in: *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications, HotMobile '17*, 2017, pp. 85–90.
- [19] K. Jansen, N. Tippenhauer, C. Pöpper, Multi-receiver GPS spoofing detection: error models and realization, in: *Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC '16*, 2016, pp. 237–250.
- [20] L. Heng, D.B. Work, G.X. Gao, GPS signal authentication from cooperative peers, *IEEE Trans. Intell. Transp. Syst.* 16 (4) (2015) 1794–1805.
- [21] F.A. Milaat, H. Liu, Decentralized detection of GPS spoofing in vehicular ad hoc networks, *IEEE Commun. Lett.* 22 (6) (2018) 1256–1259.
- [22] Z. Zhang, M. Trinkle, L. Qian, H. Li, Quickest detection of GPS spoofing attack, in: *MILCOM 2012 - 2012 IEEE Military Communications Conference*, 2012, pp. 1–6.
- [23] Y. Guo, L. Miao, X. Zhang, Spoofing detection and mitigation in a multi-correlator GPS receiver based on the maximum likelihood principle, *Sensors* 19 (1) (2018).
- [24] D.-K. Lee, N. Spens, B. Gattis, D. Akos, AGC on android devices for GNSS, in: *Proceedings of the 2021 International Technical Meeting of the Institute of Navigation*, 2021, pp. 33–41.
- [25] S. Liu, X. Cheng, H. Yang, Y. Shu, X. Weng, P. Guo, K.C. Zeng, G. Wang, Y. Yang, Stars can tell: a robust method to defend against {GPS} spoofing attacks using off-the-shelf chipset, in: *30th USENIX Security Symposium (USENIX Security '21)*, 2021, pp. 3935–3952.
- [26] S. Sciancalepore, G. Oliveri, R. Di Pietro, Shooting to the stars: secure location verification via meteor burst communications, in: *2018 IEEE Conference on Communications and Network Security, CNS*, 2018, pp. 1–9.
- [27] K. Jansen, M. Schäfer, D. Moser, V. Lenders, C. Pöpper, J. Schmitt, Crowd-GPS-sec: leveraging crowdsourcing to detect and localize GPS spoofing attacks, in: *2018 IEEE Symposium on Security and Privacy, SP*, Vol. 00, 2018, pp. 189–202.
- [28] K. Jansen, M. Schäfer, V. Lenders, C. Pöpper, J. Schmitt, POSTER: localization of spoofing devices using a large-scale air traffic surveillance system, in: *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Asia CCS '17*, 2017, pp. 914–916.
- [29] A. Ranganathan, H. Ólafsdóttir, S. Capkun, SPREE: a spoofing resistant GPS receiver, in: *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, MobiCom '16*, 2016, pp. 348–360.
- [30] M. Arafin, D. Anand, G. Qu, A low-cost GPS spoofing detector design for internet of things (IoT) applications, in: *Proceedings of the on Great Lakes Symposium on VLSI 2017, GLSVLSI '17*, 2017, pp. 161–166.
- [31] S. Tariq, H. Kim, J. Ryoo, AuthGPS: lightweight GPS authentication against GPS and LTE spoofing (poster), in: *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys '19*, 2019, pp. 547–548.
- [32] Y. Qiao, Y. Zhang, X. Du, A vision-based GPS-spoofing detection method for small UAVs, in: *2017 13th International Conference on Computational Intelligence and Security, CIS*, 2017, pp. 312–316.
- [33] T. Mismar, J. Kim, M. Alam, Indoor antispoofing cooperative localization in cellular networks, *IEEE Trans. Aerosp. Electron. Syst.* 51 (4) (2015) 2823–2833.
- [34] D. Zhang, S. Zhao, L.T. Yang, M. Chen, Y. Wang, H. Liu, NextMe: localization using cellular traces in internet of things, *IEEE Trans. Ind. Inf.* 11 (2) (2015) 302–312.
- [35] S. Fang, Y. Hsu, Y. Shiao, F. Sung, An enhanced device localization approach using mutual signal strength in cellular networks, *IEEE Internet Things J.* 2 (6) (2015) 596–603.
- [36] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, C. Fischione, A survey of enabling technologies for network localization, tracking, and navigation, *IEEE Commun. Surv. Tutor.* 20 (4) (2018) 3607–3644.
- [37] F. Formaggio, S. Ceccato, F. Basana, N. Laurenti, S. Tomasin, GNSS spoofing detection techniques by cellular network cross-check in smartphones, in: *Proceedings of the 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2019*, 2019, pp. 3904–3916.
- [38] S. Ceccato, F. Formaggio, G. Caparra, N. Laurenti, S. Tomasin, Exploiting side-information for resilient GNSS positioning in mobile phones, in: *Proceedings of IEEE/ION PLANS*, 2018, pp. 1515–1524.
- [39] Y. Wei, H. Li, S. Shang, M. Lu, GNSS spoofing detection using raw IMU measurements and pre-integration, in: *Proceedings of the 2020 International Technical Meeting of the Institute of Navigation*, 2020, pp. 1515–1524.
- [40] C. Tanil, S. Khanafseh, B. Pervan, Impact of wind gusts on detectability of GPS spoofing attacks using RAIM with INS coupling, in: *Proceedings of the ION 2015 Pacific PNT Meeting*, 2015, pp. 674–686.
- [41] J. Shokouh, Detecting GNSS Attacks on Smartphones (Master thesis), 2013.
- [42] Cisco, 2022, <https://www.cisco.com/c/en/us/td/docs/routers/access/800/829/software/gps/Assisted-gps.html>.

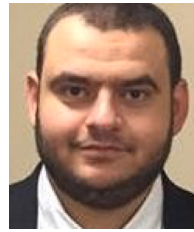
- [43] N. Lu, N. Cheng, N. Zhang, X. Shen, J.W. Mark, Connected vehicles: solutions and challenges, *IEEE Internet Things J.* 1 (4) (2014) 289–299.
- [44] G. Oligeri, S. Sciancalepore, R. Di Pietro, GNSS spoofing detection via opportunistic IRIDUM signals, in: *Proceedings of the 13th Conference on Security and Privacy in Wireless and Mobile Networks, WiSec '20*, 2020.
- [45] C.L. Leca, P. Ciotirnae, C.I. Rincu, I. Nicolaescu, Characteristics of crowdsourcing for outdoor radio fingerprinting positioning, in: *International Conference on Electronics, Computers and Artificial Intelligence, ECAI*, 2017, pp. 1–4.
- [46] HackRF One, 2022, <https://greatscottgadgets.com/hackrf>.
- [47] GPS-SDR-SIM, 2022, <https://github.com/osqzss/gps-sdr-sim>.
- [48] Unwiredlabs, 2022, <https://unwiredlabs.com>.
- [49] M. Leadbetter, G. Lindgren, H. Rootzén, *Extremes and Related Properties of Random Sequences and Processes*, Springer-Verlag, 1983.
- [50] Google Earth Pro, 2022, <https://www.google.com/earth>.
- [51] LabSat SatGen, 2022, <https://www.labsat.co.uk/index.php/en/products/satgen-simulator-software>.
- [52] SiRF Technology, NMEA Reference Manual, Technical Report, 2005.
- [53] H. Yang, S. Bae, M. Son, H. Kim, S.M. Kim, Y. Kim, Hiding in plain signal: physical signal overshadowing attack on {ITE}, in: *28th {USENIX} Security Symposium, {USENIX} Security 19*, 2019, pp. 55–72.
- [54] N.O. Tippenhauer, K.B. Rasmussen, C. Pöpper, S. Capkun, iPhone and iPod Location Spoofing: Attacks on Public WLAN-based Positioning Systems, Technical Report/ETH Zurich, Vol. 599, Department of Computer Science, 2012.
- [55] A. Harvey, 2020, URL: <https://github.com/adamhrv/skylift>.
- [56] K. Wang, S. Chen, A. Pan, Time and position spoofing with open source projects, *Black Hat Europe 148* (2015) 1–8.
- [57] Statista, 2022, <https://www.statista.com/statistics/697120/car-trip-average-distance-europe-by-purpose>.
- [58] K.C. Zeng, S. Liu, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, Y. Yang, All your GPS are belong to us: towards stealthy manipulation of road navigation systems, in: *27th USENIX Security Symposium, USENIX Security 18*, 2018, pp. 1527–1544.



Gabriele Oligeri is Assistant Professor at HBKU-CSE. Previously, he was Assistant Research Professor at KINDI Center for Computing Research at Qatar University. He received his M.Sc. Degree in Computer Engineering from the University of Pisa and the Ph.D. degree in Computer Engineering from the same university. His research interests are in the area of security and privacy for cyber physical systems.



Savio Sciancalepore is currently Assistant Professor in IoT security at Eindhoven University of Technology (TU/e), Eindhoven, Netherlands. He received his Master's degree in Telecommunications Engineering in 2013 and the Ph.D. in 2017 in Electric and Information Engineering, both from the Politecnico di Bari, Italy. He received the prestigious award from the ERCIM Security, Trust, and Management (STM) Working Group for the best Ph.D. Thesis in Information and Network Security in 2018. From 2017 to 2020, he was Post Doc Researcher at HBKU-CSE-ICT, Doha, Qatar. From January 1, 2021, he is with TU/e. His major research interests include wireless and network security issues in Internet of Things (IoT) systems and Cyber-Physical Systems, including UAV networks, avionics systems, and mobile networks.



Omar Adel Ibrahim is currently a Computer Science and Engineering Ph.D. candidate at HBKU-CSE-ICT, Doha, Qatar. He received his bachelor's degree in computer engineering from Qatar University in 2017 and the Master of Cybersecurity degree from HBKU-CSE-ICT in 2019. His main research interests cover security issues in Cyber-Physical Systems (CPS), including Drones, GPS, and USB devices security.



Roberto Di Pietro, ACM Distinguished Scientist, is Full Professor in Cybersecurity at HBKU-CSE. Previously, he was in the capacity of Global Head Security Research at Nokia Bell Labs, and Associate Professor (with tenure) of Computer Science at University of Padova, Italy. He also served 10+ years as senior military technical officer. Overall, he has been working in the cybersecurity field for 23+ years, leading both technology-oriented and research-focused teams in the private sector, government, and academia (MoD, United Nations HQ, EUROJUST, IAEA, WIPO). His main research interests include security and privacy for wired and wireless distributed systems (e.g. Blockchain technology, Cloud, IoT, On-line Social Networks), virtualization security, applied cryptography, computer forensics, and data science. In 2011–2012 he was awarded a Chair of Excellence from University Carlos III, Madrid. In 2020 he received the Jean-Claude Laprie Award for having significantly influenced the theory and practice of Dependable Computing.