A Spoof-Proof GPS Receiver[*]

Manuel Eichelberger
ETH Zurich
manuelei@ethz.ch

Ferdinand von Hagen
ETH Zurich
vhagenf@ethz.ch

Roger Wattenhofer
ETH Zurich
wattenhofer@ethz.ch

ABSTRACT

We present a precise and robust GPS spoofing mitigation method, which is based on a position likelihood distribution. Compared to existing spoofing mitigation methods, this maximum likelihood method is less prone to selecting the wrong satellite signals when (spoofed) duplicates are received. The presented method operates from GPS signal snapshots as short as one millisecond and detects all likely receiver positions. Even with spoofing signals stronger than the authentic satellite signals, the actual receiver position is still found through iterative dampening of the strongest signals. The spoofing mitigation capability is evaluated on the de facto standard *TEXBAT* spoofing dataset. We reach median position errors introduced by spoofing below 19 m and keep the maximum error below 222 m for all TEXBAT scenarios. This is six times more accurate than the best previous work, which only *detects* spoofing attacks, but does not *mitigate* them.

CCS CONCEPTS

• **Information systems → Global positioning systems**; • **Security and privacy → Domain-specific security and privacy architectures**.

KEYWORDS

aircraft, anti-spoofing, attack, collective detection, defend, implementation, maximum likelihood, low power, ocean, plane, precise, probability distribution, robust, ship, sea, snapshot

1 INTRODUCTION

1.1 Motivation

Today, countless applications rely on the Global Positioning System (GPS). GPS is not only used to get an accurate position, but also to get an accurate time. As such, GPS receivers are becoming prime targets for attacks. Wrong information in time or space can have severe consequences, as we highlight in the following examples.

Aircraft Navigation. Air traffic control is partially transitioning from radar to GPS. Each aircraft determines its own location using an on-board GPS receiver. Each aircraft transmits its current GPS location twice per second, through so-called *ADS-B* messages. This system is already mandatory for most airliners in Europe and in the US [9, 17]. If a wrong location is estimated by the on-board GPS receiver, for instance due to signal spoofing, this may have fatal consequences. For instance, wrong routing instructions might be given due to a wrong reported aircraft location, leading to collisions with the ground or other aircraft.

Ship Navigation. Like aircraft, apart from GPS, ships may have few reference points to localize themselves. Trusting a wrong location indication can strand a ship or alter its course. A recent GPS spoofing incident had several ships placed inland although they were actually on the Black Sea, showing that spoofing attacks against ships already happen in the wild [5].

Car/Truck Navigation. Drivers increasingly rely on GPS navigation alone rather than orienting themselves. Too often, directions given by car navigation systems are not validated but followed blindly. This emerging dependence on GPS is dangerous: Even without spoofers being present, people get stuck in remote places. This may be due to errors in the given directions or simply because of typing errors. In some cases, consequences are fatal [32]. Attackers can use this combined weakness of GPS and car drivers to reroute cars and cause traffic chaos, for instance.

Train Control. Emerging train control systems such as the ETCS may employ GPS localization for each train instead of placing numerous balises along tracks [33]. Wrong location estimates could wreak havoc: Collisions between trains might not be anticipated early enough or barriers may not be lowered in time. Also track switches could be triggered while a train is passing through.

While the examples above are in the domain of *location* spoofing, an attacker can also try to change a GPS receiver's perceived *time*.

Cellular Network Synchronization. Cellular networks rely on accurate time synchronization for exchanging communication data packets between ground antennas and mobile handsets in the same network cell. Also, *all* neighboring cells of the network need to be time-synchronized for seamless call handoffs of handsets switching cells and for coordinating data transmissions in overlapping coverage areas [1, 22]. Because most cellular ground stations get their timing information from GPS, a signal-spoofing attacker could decouple cells from the common network time. Overlapping cells might then send data at the same time and frequencies, leading to message collisions and losses [1]. Failing communication networks can disrupt emergency services, as people in need of help lose the means of requesting assistance. Also, businesses relying on mobile phones to coordinate their work with customers, like taxi services and transport companies, could not carry out their work.

Stock Market Synchronization. Audit rules mandate that financial markets record trading activities with accurate timestamps [10]. Such timing is often accomplished through GPS receivers on the roof of those market places [29]. Timestamps help revealing illegal trading activities, which can sometimes be detected by trading discontinuities, arising for instance when market orders are not executed immediately. Also, with too coarse timestamps, it is possible to observe new market orders and then place one's own orders

[*] The authors of this paper are alphabetically ordered.

"concurrently", so that the latter might be executed before the former [28].

Power Grid Synchronization. The operation of power grid assets is coordinated with GPS-based precision timing. Also, grid operators use GPS-synchronized observations for disturbance monitoring and fault localization, to maintain grid stability [13]. For many nations, a power grid outage is one of the worst threats. Problems include water pumps that stop working and food and medicine which cannot be delivered due to failing communication [12].

These threats and weaknesses show that there is the potential for large damages as a result of forged GPS signals. So what is GPS spoofing?

1.2 Spoofing

A GPS receiver computing its location incorrectly or even failing to estimate any location at all can have different causes. Wrong localization solutions come from (i) a low signal-to-noise ratio (SNR) of the signal, for instance when inside a building or in urban canyons, (ii) overlapping reflected signals in multipath scenarios or (iii) deliberately jammed or (iv) spoofed signals. The first two cases are challenging, but various ideas help in mitigating their effects:

In case (i), low SNR, it is possible to use a longer recorded signal in order to increase the total received signal energy. There are some challenges associated with this, for instance phase changes in the signal due to data modulation on top of the carrier signal. Also, the latency of the localization solution increases because the amount of signal used can be on the order of minutes.

In case (ii), multipath signals can often be discarded by selecting only the strongest signals and those which are consistent in the sense that the localization solution fits well with all chosen signals.

In case (iii), an attacker simply *jams* the frequency band of the GPS signals with strong random signals, increasing the noise level at receivers. Jamming is the least sophisticated kind of attack and has a result equivalent to case (i) above: a low SNR at the receiver. Therefore, longer signal recordings also help against jamming. Apart from taking measures against special types of jamming attacks, like using directional antennas to exclude ground-based jammers, jamming is easily detectable. On the other hand, jamming is basically impossible to mitigate [35].

Signal spoofing (iv) is the most difficult case. An attacker can freely choose the signal powers and delays for each satellite individually. So, the attacker will be located nearby a GPS receiver, and simply transmit a competing signal s'. At its current location, the GPS receiver should receive a combined satellite signal s. However, at another location and/or time, a combined signal s' would be expected. Such spoofed signals can nowadays be generated with only a few hundred dollars worth of hardware.

If the attacker uses less power than what is received from the legitimate GPS satellites (that is, "$s' \ll s$"), the signal s' essentially plays the role of interference, lowering the SNR as in case (i). If on the other hand the attacker uses significantly more power ("$s' \gg s$"), a GPS receiver may easily notice that an attack is going on, as the received signal power is unusually high. So, the most interesting case is if the attacking power is natural, in the order of the regular satellites ("$s' \approx s$").

These threats are well acknowledged. A recent US government study concluded that critical infrastructure relies on GPS, but is not prepared for signal disruptions [16].

While information about the internals of commercial receivers is scarce, to the best of our knowledge, consumer products currently have at most simple spoofing mitigation integrated. Sometimes, commercial receivers may detect inconsistencies in the received signals and simply cease operation [39]. But it is even possible to mislead commercial receivers whose output is combined with other sensors such as an altimeter, a magnetometer and an inertial measurement unit (accelerometer, gyrometer, compass) [40]. Military receivers use symmetrically encrypted GPS signals which are not available to the public. In this case, the signals are unknown to attackers in advance. Still, an attacker could replay even those encrypted signals with a small delay to confuse receivers. Academically, some anti-spoofing methods have been studied (see next section for details), but the spatial resolution of those methods is hundreds of meters, which means that attacks spoofing a close-by location cannot be detected. Our method achieves median errors under 19 m on the TEXBAT dataset, which is the de facto reference dataset for testing GPS anti-spoofing algorithms [37, 46].

1.3 Collective Detection

In this work, we not only *detect* spoofing attacks, but also *mitigate* them. We present a robust spoofing mitigation algorithm based on the *collective detection* maximum likelihood localization approach [2]. Our method can differentiate closer distances between correct and spoofed locations than previously known approaches.

A specialty of our method is that it uses only a few milliseconds worth of raw GPS signals, so-called *snapshots*, for each location fix. This enables the offloading of the computation into the cloud, which allows the combination of knowledge of observed attacks. Measurements from enough receivers may even permit finding spoofers' locations. Cloud offloading also makes our technique suitable for energy-constrained sensors. A novel class of low-power GPS sensors are *snapshot receivers* [11, 26]. In contrast to classic GPS receivers, which are "always on", snapshot receivers capture only a few millisecond of the satellite signals for each localization. More details about snapshot receivers follow in Section 3.2. Existing spoofing mitigation methods require a constant stream of the GPS signals and track these signals over time. Since snapshot receivers only have access to extracted signal segments, classic GPS anti-spoofing techniques are not applicable to snapshot receivers. To our knowledge, our GPS anti-spoofing work is the first for snapshot receivers and can also be used with conventional receivers. Generally, spoofing mitigation is computationally more demanding than normal localization, since fake signals have to be detected. In order to remove spoofed locations, different potential location solutions need to be compared. Therefore, spoofing mitigation is a computational challenge on smartphones.

2 RELATED WORK

Three tracks of research are most relevant to our work, maximum likelihood GPS localization, GPS spoofing mitigation algorithms and successive signal interference cancellation.

2.1 Maximum Likelihood Localization

Our work is based on *collective detection (CD)*, which is a maximum likelihood GPS localization technique. Maximum likelihood GPS localization was already proposed in 1996 [41], but was computationally infeasible at that time. CD has first been implemented by Axelrad et al. in 2011 [2]. Due to search spaces containing millions or more location hypotheses that have to be searched through, subsequent work focused on reducing the computational burden through heuristics [7, 24]. Recently, a branch-and-bound algorithm has been proposed that finds the optimal solution within some ten seconds running on a single CPU thread [3]. Our method is an adaptation of this branch-and-bound algorithm to mitigate GPS signal spoofing attacks. Another maximum likelihood approach by Closas et al. models the signal observations as a function of the receiver state [8]. Due to a high-dimensional and non-linear cost function, it remains unclear how the optimal receiver location can efficiently be computed in that framework.

2.2 Spoofing Mitigation

GPS spoofing defenses have intensively been studied. However, while most research focuses on *detecting* spoofing attacks, there is a lack of ideas for spoofing *mitigation* and *recovering* from successful attacks by finding and authenticating the correct signals [36]. Our work helps in this area, as the technique presented in this paper inherently mitigates spoofing attacks.

A lot of research focuses on tracking multiple signals per satellite instead of at most one [6, 37]. This is a useful approach for *detecting* spoofing attacks. However, given multiple signals per satellite, it is a challenge to select the correct signal from each satellite. In addition, the *SPREE* receiver implements plausibility checks for less elaborate attacks such as modified satellite orbit parameters [37]. Another method for detecting spoofing attacks is hypothesis testing [48].

Whether sophisticated spoofing attacks are practical is subject to debate [39]. Still, spoofing hardware performing a relatively challenging *seamless satellite-lock takeover attack* has already been built, although it has only been tested in a lab environment [20]. Challenges associated with spoofing are for instance matching the spoofed and authentic signals' amplitudes at the receiver, which might not be in line of sight and moving [38]. Despite that, it is even practically feasible for a spoofer to erase the authentic signals with signals at a 180° phase offset [36]. This is one of the strongest attacks and can only be detected with multiple receiver antennas or by a moving receiver [36]. Thus, a cooperative victim, like a convicted criminal with an ankle monitor, could use this technique to deceive authorities [36, 38]. For signal erasure to be feasible, the spoofer needs to know the receiver location more accurately than the GPS L1 wavelength, which is 19 cm. Receivers with only a single antenna cannot withstand such an erasure attack. Our method targets single-antenna receivers and we therefore do not deal with signal erasure. In basically all other types of spoofing attacks (cf. Section 4), including signal replay and even spoofers with multiple transmission antennas, the original signals are still present and our algorithm remains robust.

Due to the limitations of receivers with a single antenna, some research focuses on receivers with multiple antennas or even multiple receivers combining their information [27]. Coordinated spoofing attacks with multiple antennas can circumvent *some* defenses using multiple receiver antennas like detecting signal timing inconsistencies [42]. Also, size requirements and a high price sensitivity for consumer GPS receivers make multi-antenna receivers impractical for many applications. Single-antenna receivers cannot differentiate between spoofing signals sent from one or more locations. Our algorithm is aimed at those single-antenna receivers and is therefore indifferent to multi-antenna attackers.

One approach against erasing spoofers with a single transmitting antenna focuses on moving receivers [4]. Signals are classified into spoofed and non-spoofed signals by moving the receiver around and observing the spatial correlation of signals sent from a single source. The method does not cover stationary applications like the introductory time synchronization examples and time periods during which a mobile receiver is not moving.

The GPS anti-spoofing work most relevant to this paper is that based on joint processing of satellite signals and maximum likelihood localization. One method is able to *mitigate* a limited number of spoofed signals by the vector tracking of all satellite signals [23]. A similar technique is shown to be relatively robust against jamming and signal replay [34]. Another idea is to combine all satellite signals in a Bayesian estimation algorithm [25]. Compared to our snapshot receiver, this technique uses a continuous stream of received signals for the sequential parameter estimation. Extensions of aforesaid maximum likelihood method by Closas [8] for countering spoofing have also been proposed. One assumes a spoofer which sends unsynchronized spoofing signals that do not consistently point to a spoofed location [45] and the other tries to solve the global convergence problem with an initial grid search and subsequent iterative refinement [47]. Our method can tolerate consistent spoofing signals, even in case the spoofing signal is already present when the receiver starts.

We could not find any anti-spoofing methods for GPS snapshot receivers. Since our method yields robust location fixes from signal snapshots, there is no need for recovery like in classic receivers. The latter may lock onto spoofed signals without noticing a drift from the authentic satellite signals over time.

2.3 Successive Interference Cancellation

Our iterative signal dampening technique to deal with spoofing signals is similar to successive interference cancellation (SIC). SIC removes the strongest received signals one by one in order to find weaker signals and has been used with GPS signals before [30, 31]. That work is based on a classic receiver architecture which only keeps a signal's timing, amplitude and phase. Our receiver is based on CD, which directly operates in the localization domain and does not identify individual signals in an intermediate stage. As it is impossible to differentiate between authentic and spoofed signals a priori, we do not remove signals from the received sampled data. Otherwise, the localization algorithm might lose the information from authentic signals. Instead, we dampen strong signals in order to reveal weaker signals. This can reveal localization solutions with lower CD likelihood.

3 GPS LOCALIZATION

The Global Positioning System (GPS) is a Global Navigation Satellite System (GNSS) operated by the United States Air Force. It provides location and time information to receivers anywhere on Earth where signals from at least four satellites can be received. The GPS satellites are located in a non-stationary medium Earth orbit and circle the Earth about twice a day.

GPS satellites transmit multiple signals in different frequency bands. Some of the signals are encrypted and reserved for military use. We focus on the signal most commonly used in civilian receivers, which is located in the L1 frequency band at 1.57542 GHz. To distinguish the satellites, *code division multiple access (CDMA)* is used. The employed Gold codes, one for each satellite, with 1023 bits length, achieve good correlation and cross-correlation properties [21]. Those signals are also called *pseudo-random noise (PRN)* sequences due to their noise-like nature. Sent with a data rate of 1.023 MHz, the codes repeat every millisecond. The satellites further transmit navigation data. The navigation data contains satellite orbit information, called *ephemeris*, and transmission timestamps, which allow calculating the exact location of the satellites at the time of signal transmission. The data is modulo-2 added to the Gold codes at a rate of 50 bit/s. Hence, each data bit is transmitted through 20 subsequent Gold codes. The generated signal is sent with *binary phase shift keying (BPSK)* on the L1 frequency band.

3.1 Localization

For localization, GPS receivers measure the time of flight of received satellite signals. Using an orbit model whose parameters are received with the navigation messages of the previous step, the location of the satellites at the time of signal transmission is determined. Unlike the satellites, the receiver does not carry an atomic clock and is thus not synchronized with the satellites. Therefore, the localization problem has four unknowns, namely three spatial coordinates and the receiver's time offset from the GPS system time. The classic way of computing a solution to the localization problem, a so-called *fix*, consists of setting up a system of equations from the measured satellite distances and solving it in a least-squares sense.

Classic GPS receivers consist of three stages, acquisition of the satellite signals, decoding of the satellite data and finally, calculation of a location solution based on the received data.

The acquisition finds the visible satellites and detects the code phase of the Gold codes and the Doppler shifts of the signals. A strong correlation marks the code phase of the Gold code for a given satellite. An example can be seen in Figure 1. The code phase is determined by the time of flight of the signal between the satellite and the receiver and therefore by their distance. The relative speed between satellite and receiver introduces a significant Doppler shift to the carrier frequency. This Doppler shift has to be found during acquisition to allow decoding of the signal.

Classic receivers use the information gathered during acquisition and start using a feedback loop to track the satellite signals to decode the contained navigation message. After a receiver obtains that information from at least four satellites, the receiver can compute its location.

3.2 Snapshot Receivers

Assisted GPS (A-GPS) addresses a weakness of the basic GPS system: Due to a low data rate, limited by the large satellite distance and therefore weak received signal power, satellite orbit parameters are only transmitted every 30 seconds. Thus, the latency of a first fix after turning on a receiver, the so-called *time to first fix (TTFF)*, can be high. With A-GPS instead, these orbit parameters are fetched over the Internet, for instance via a cellular network, which reduces the data transmission time, and thus the TTFF, drastically [44].

While the satellite orbit parameters are usually valid for two hours, classic receivers also need to receive a current timestamp. Timestamps are transmitted from satellites every six seconds. Threrefore, receiving timestamps still causes a relatively high latency and high energy consumption in GPS receivers. Snapshot GPS receiver techniques such as *Coarse-Time Navigation (CTN)* or *collective detection (CD)* allow computing the receiver location even if no timestamp is received. GPS signals repeat every millisecond and the signals propagate 300 km during that time. Therefore, only the remainder of the satellite distances modulo 300 km can be measured without receiving a timestamp. If the initial estimate of the receiver's location and time has an error equivalent to less than 150 km, the measurement's full-millisecond ambiguity vanishes. For this purpose, an offset of one second is roughly equivalent to an error of 1 km since the satellites' relative speed to an observer on the Earth surface is about 1 km/s. With such an approximate initial receiver state, one can estimate the satellite locations and signal times of flight and the localization can be executed [44]. With longer code periods, such as Galileo's 4 ms long signals, the receiver state estimate's required accuracy is relaxed proportionately.

With this insight, snapshot receivers are able to compute their location from as little as one millisecond of data if the signal quality is good. However, the influence of noise is often too large to make localization viable from 1 ms of signal only. Combining several milliseconds of signal is more robust [26]. Due to only a few milliseconds activation to receive enough signal power for a localization, snapshot receivers use low power. Snapshot receivers are even suitable for multi-year tracking of battery-powered sensors [11]. In comparison, classic GPS receivers drain a smartphone battery in a few hours. Thus, it can be expected that snapshot receivers will be deployed extensively in the future. However, snapshot receivers cannot be protected by existing GPS anti-spoofing methods that track signals over time. Our present work is designed for signal snapshots, and therefore helps in protecting snapshot GPS receivers.

3.3 Collective Detection

In recent years, maximum likelihood (ML) localization methods have been proposed, promising more robust localization. That is, ML methods are more tolerant to low SNR, multipath effects and spoofing than the classic least-squares localization methods. Since the arrival time of a satellite signal cannot always be determined with certainty, a wrong signal time of flight (ToF) might be estimated. This renders the system of equations unsolvable. The information from the rest of the satellites could still be enough to compute the location fix, but eliminating "bad" measurements is not always easy. ML methods [8] and in particular collective detection

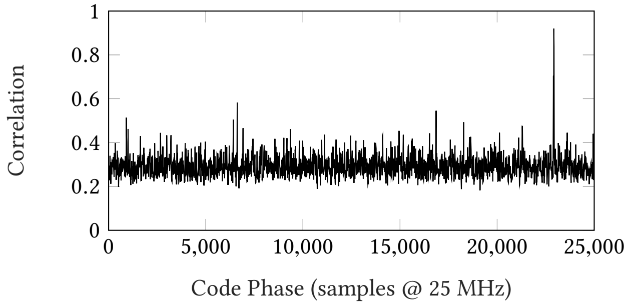


Figure 1: Acquisition result for a satellite signal with average SNR. The (single) correlation peak indicates the signal's receive time. The length of the correlation vector with 25,000 samples corresponds to 1 ms of signal.

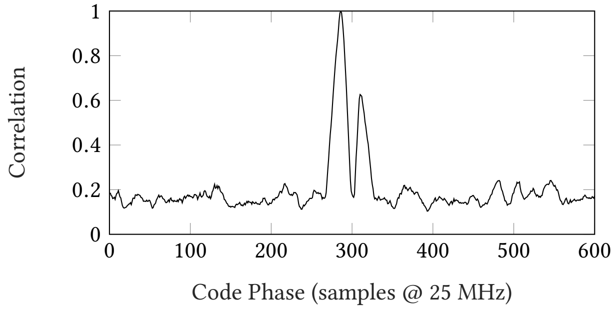


Figure 2: Acquisition result for a satellite signal with good SNR but two matching signals. The two peaks are 24 samples apart, which corresponds to a measured distance difference of 288 m. Two possible interpretations are that the first signal is the authentic signal and the second is a signal reflection (multipath) or that one of the signals is spoofed. Only the relevant part of the acquisition vector is shown. The full vector is 25,000 samples or 1 ms long.

(CD) [2, 3, 7, 24] do not pick an arrival time for each satellite signal, but rather combine all the available information and take a decision only at the end of the computation. This uses more computation power, but is less prone to errors than solving a system of equations in the least-squares sense like in CTN and classic GPS localization.

Since GPS localization is based on satellite signal ToF measurements, the main challenge is determining the signal arrival times despite low received signal power. In the methods presented so far, the arrival times are detected based on the amplitude in the correlation with the corresponding satellite's PRN sequence. This requires the presence of a clear peak in the correlation vector. With bad signal conditions, for instance under a tree, in an urban canyon or even indoors, there may be several or no such correlation peaks. The problem is particularly pronounced when only a few milliseconds of signal are used as in CTN, because the received signal power is less than with multiple seconds of signal. To mitigate this problem, CD does not only "accumulate" the captured signal over time, but also over all available satellites. Combined, the signal energy of multiple satellites gives a higher chance to detect the signal arrival times correctly. The gain in the signal-to-noise ratio (SNR) of CD compared to CTN means that CD is more robust to noise. Therefore, CD is suited for bad signal conditions such as in spoofing scenarios.

Our method is based on an efficient implementation of CD [3]. A given four-dimensional (location and time) search space is discretized as a regular grid of solution hypotheses. The expected distance between satellite and receiver, and thus the expected code phase of the received signal, is calculated for each grid point. The satellite signal correlation vectors from the acquisition are aligned by the expected code phase and a pseudo-likelihood of the point is calculated. By searching over all possible solutions in the grid, the algorithm is guaranteed to deliver the most likely location given the observed signals. A branch-and-bound implementation delivers the same result with reduced computational effort.

4 GPS SIGNAL ATTACKS

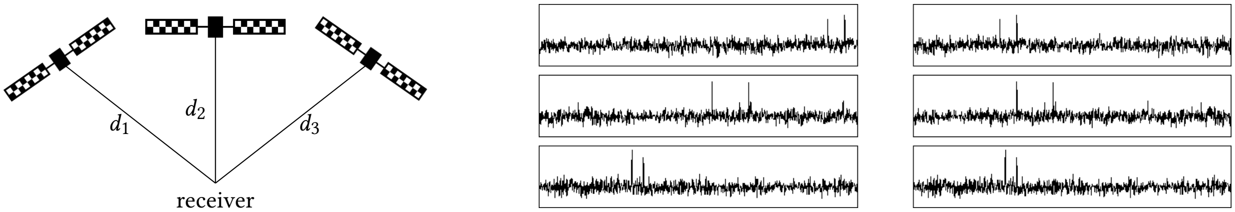
The easiest way to prevent a GPS receiver from finding its location is jamming the GPS frequency band. GPS signals are weak and require sophisticated processing to be found. Jamming considerably worsens the signal-to-noise ratio (SNR) of the satellite signal acquisition. CD algorithms achieve a better SNR than classic receivers and are thus able to tolerate more noise or stronger jamming [2].

A jammed receiver is also less likely to detect spoofing, since the original signals cannot be found. The receiver tries to acquire any satellite signals it can find. Thus, the attacker only needs to send a set of valid GPS satellite signals stronger than the noise floor, without any synchronization with the authentic signals.

As jamming is detectable by observing the noise floor, in-band power level and loss of satellite signal lock, a more subtle attack may be performed. The spoofer can send the set of satellite signals with adjusted power levels and synchronized to the authentic signals to successfully spoof the receiver.

Seamless Satellite-Lock Takeover. The most powerful attack is a seamless satellite-lock takeover. In such an attack, the original and counterfeit signals are nearly identical with respect to the satellite code, navigation data, code phase, transmission frequency and received power. This requires the attacker to know the location of the spoofed device, so that time of flight and power losses over distance can be factored in. After matching the spoofed signals with the authentic ones, the spoofer can send its own signals with a small power advantage to trick the receiver into tracking those instead of the authentic signals. When the spoofer tries to draw the receiver away from the authentic signal, a correlation vector might look like the one in Figure 2. A classic receiver without spoofing countermeasures, such as tracking multiple peaks, is unable to mitigate or detect this attack, as there is no indicative interruption of the receiver's signal tracking.

Navigation Data Modification. An attacker basically has two attack vectors: modifying the signal's code phase or altering the



(a) Satellite distances for a given receiver location hypothesis.

(b) Satellite signals as received (left) and shifted circularly according to the satellite distances d_i (right). The aligned peaks on the right show that this is a location hypothesis with a coherent set of signals and thus with high likelihood. As visible, there are two peaks for each satellite, meaning that a spoofed set of signals is likely present in addition to the authentic satellite signals.

Figure 3: Collective detection: The (pseudo-)likelihood of a point is computed by 1) shifting the signal correlation vector for each satellite according to that point and satellite's distance and then 2) pointwisely adding all satellites' vectors and selecting the maximum value of the result as the likelihood.

navigation data. Misaligning the code phase leads to changes in the signal arrival time measurements. And by changing the navigation data, the attacker displaces the perceived satellite locations. Both methods influence the calculated receiver location. In comparison to classic receivers, assisted or snapshot GPS receivers like CTN and CD are not vulnerable to navigation data changes in the satellite signals as they fetch that information from other sources like the Internet. An attacker could tamper with such data sources, but that shall not be our concern in this paper. Rather, we deal with modified, wireless GPS signals.

5 ALGORITHM DESIGN

Our method is aimed at single-antenna receivers. Therefore, we do not deal with signal erasure attacks (cf. Section 2.2). Instead, given a mix of authentic and spoofed signals observed at a receiver, such as shown in Figures 2 and 3b, our goal is to identify all likely locations. Pseudocode for our method is given in Algorithm 1. The basic idea is to run collective detection (CD) localization repeatedly and find one plausible receiver location in each iteration. Specifically, we use a fast branch-and-bound CD version (BBCD) [3].

Collective Detection. CD is a good choice for several reasons: 1) CD has improved noise tolerance compared to classic receivers, 2) CD is not susceptible to navigation data modifications, 3) CD is suitable for snapshot receivers, and 4) CD computes a location likelihood distribution which can reveal all likely receiver locations including the true location, independent of the number of spoofed and multipath signals. Actually, spoofing and multipath signals are similar from a receiver's perspective. Both result in several observed signals from the same satellite, like shown in Figures 2 and 3b. The difference is that multipath signals' delay depends on the environment while spoofing signals can be crafted to yield a consistent localization solution at the receiver.

In order to detect spoofing and multipath signals, classic GPS receivers, such as SPREE (cf. Section 2.2), can be modified to track an arbitrary number of signals per satellite, instead of only one [37]. In a classic receiver, the set of authentic signals—one signal from each satellite—would have to be correctly identified for a successful localization. Any selection of signals can be checked for consistency by verifying that the resulting localization solution's residual error is small. Consistent solutions are either the actual receiver location or a spoofed one. However, already finding sets of signals which are consistent for one receiver location, is combinatorially difficult: For n satellites and m transmitted sets of spoofed signals, there are $(m+1)^n$ possibilities for the receiver to select a set of signals. Only $m+1$ of those will result in a consistent localization solution, namely the actual location and m spoofed locations. Even if running a least-squares optimization for each signal combination may be feasible, in practice multipath signals enlarge the search. Therefore, tracking multiple signals per satellite helps detecting spoofing and multipath events by raising a warning if multiple signals per satellite are received, but it is impractical to mitigate spoofing.

CD avoids this signal selection problem by combining all signals into a location likelihood distribution, as explained in Figure 3. CD only shows consistent signals, since location hypotheses leading to few aligned signals accumulate an insignificant likelihood. In the CD likelihood measure, the plausible receiver locations—given the observed signals—have high likelihoods. However, finding those likely locations efficiently is challenging. The basic version of CD computes a likelihood for *all* location hypotheses in a given discretized search region. A 2D example is given in Figure 4. Since the search is usually performed in four dimensions, space and time, computing the whole likelihood distribution is impractical. The branch-and-bound CD version, which we use, improves the computation performance, but only yields the most likely location.

Spoofing Mitigation. Among all location hypotheses, the basic CD algorithm selects the most likely location. However, as shown in Figure 4, that may be a spoofed location, so it is necessary to also consider less likely locations to be sure that the true receiver location is included in the results. But simply picking all points with a likelihood above some threshold does not work well. For close location hypotheses, the satellite distances and thus the shifts of the signal vectors differ by only a few entries (cf. Fig. 3). For two reasons, such small shift differences result in only marginally differing likelihoods: 1) the correlation peaks form triangular shapes

Algorithm 1 Spoof-Proof Branch-and-Bound Collective Detection

```
$n$: number of satellites
$l$: length of recording in samples
$S$: $n \times l$-matrix of received satellite signals
$SNR_{min}$: minimum solution SNR

$\boldsymbol{solutions} \leftarrow ()$
while true do
    $(\boldsymbol{location}, \text{SNR}, \boldsymbol{strongest\_signal\_indices}) \leftarrow BBCD(\boldsymbol{S})$
    if $SNR < SNR_{min}$ then
        return $\boldsymbol{solutions}$
    end if
    $\boldsymbol{solutions} \leftarrow (\boldsymbol{solutions}, \boldsymbol{location})$
    $S \leftarrow dampen(\boldsymbol{S}, \boldsymbol{strongest\_signal\_indices})$
end while
```

due to usual oversampling of the received signal, and 2) small timing estimation errors between signals from different satellites may misalign the correlation vectors by a few entries. Therefore, locations close to local maxima all have high likelihoods. Thus, a thresholding approach will yield uselessly many points, clustered around points with local likelihood maxima. That effect can be observed in Figure 4. Therefore, we would ideally only pick such local maxima as potential localization solutions. This could be done with some clustering technique. Instead, our proposed method finds local maxima through iterative signal dampening.

5.1 Iterative Algorithm

Rather than computing the whole location likelihood distribution, branch-and-bound CD finds the most likely location orders of magnitude faster [3]. As discussed above, we want to find *all* local likelihood maxima with an SNR above some threshold, since we assume that the receiver observes the spoofed *and* authentic signals. To achieve this, we run the branch and bound algorithm repeatedly. In each iteration, we extract the next likely location.

Signal Erasure. The most likely location is formed by high peaks in the individual satellite signal vectors, as shown in Figure 3. In order to remove the most likely location, one could therefore try erasing the highest peak for each satellite before proceeding to the next iteration. However, it need not be *the*, but only *some*, of the highest peaks forming the maximum likelihood. For instance, for some satellites, the highest peak may result from an authentic signal, while for other satellites, the highest peak may be from a spoofed signal. That might be the case when the attacker sends the spoofed signals with different power levels in order to thwart our strategy. In such a case, the most likely location may be the result of a combination of authentic and spoofed signals. If the highest peaks are removed, then also some authentic signals are removed and the actual receiver location may not be found in any later algorithm iteration. In essence, this is the same problem that classic receivers face: If multiple peaks per satellite are present, it is unclear a priori which peaks belong together, that is, which peaks yield a consistent localization solution.

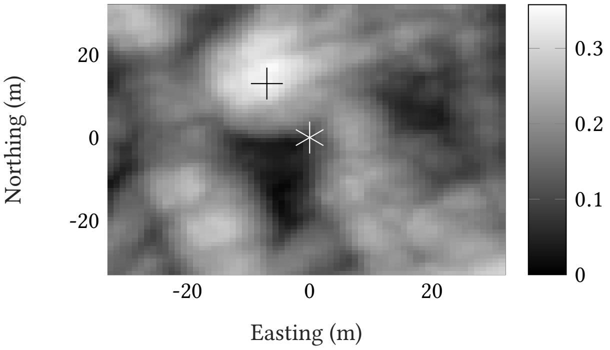


Figure 4: Two-dimensional likelihood distribution computed from a signal snapshot of a spoofing scenario. Higher values indicate higher likelihood of the point being the receiver location. The actual receiver location in the middle (∗) is invisible while the spoofed location slightly northwest (+) dominates the likelihood distribution. This means that the first iteration of our algorithm would find this spoofed location. Other points with high likelihood result from a combination of spoofed and authentic signals.

Signal Dampening. Instead of completely removing the highest signal peaks for each satellite, we exploit CD's advantage that we do not need to take a hard decision whether those peaks are authentic or not. Instead, in every algorithm iteration, we attenuate each satellite's strongest peak by some factor. Like this, that peak has less influence on the next iteration, but it can still aggregate with signals from other satellites. For instance, if the peak is formed by an authentic signal, it will still reinforce the likelihood of the correct location. Also, not completely removing signals prevents the problem from becoming underdetermined due to too few signals being available. Generally, signals from at least four satellites are needed to resolve the location and time of the receiver. (For simplicity, we just write *location* in this paper, but actually mean *location and time.*) In the end, the peak dampening emulates the selection of local maxima in the complete likelihood distribution, as outlined above. By dampening the tallest peaks iteratively, the highest local maxima vanish eventually, letting other local maxima stand out and be found subsequently.

Loop Condition. The algorithm iterations terminate when the likelihood of the computed location sinks into the noise floor of the likelihood distribution, that is, when the SNR falls below a threshold. That threshold can be chosen based on the search space's median or average likelihood, determined through random sampling.

If the (maximum) number of spoofed signals per satellite is known, as is the case for the TEXBAT scenarios, the number of algorithm iterations can also be fixed (or limited).

5.2 Solution Selection

A GPS receiver which receives multiple sets of consistent localization signals is unable to choose the authentic location based on those signals themselves. An exception might be when the spoofed and authentic signals could be discerned by different signal characteristics, such as the shape or timing of rising and falling signal edges. GPS receivers are unlikely to detect such characteristics as the GPS signals can only be detected after correlating with known satellite signals, which means that those signal characteristics are already mingled through that processing. Therefore, our method just outputs all plausible receiver locations. Based on external knowledge, the receiver can then decide which of the found locations must be correct. For instance, using sensor data from an accelerometer, a motion model can be matched with sequences of likely locations. Or only smooth receiver paths can be accepted, based on the receiver's maximum de- and acceleration. Further, location hypotheses can be reconciled with a map, for instance eliminating locations not on a road. In practice, a combination of external information sources may be used. For example, restricting the receiver location to a road still gives an attacker the possibility to reroute the computed position at any junction. To mitigate such an attack, 1) an accelerometer reading can limit the accepted turn rate; 2) a map of known WiFi or LTE transmitters and expected signal power can help deciding which street section is correct; or 3) visual clues from a camera or a user, such as street signs, traffic lights and distinctive buildings can eliminate location candidates. The visual technique may especially be useful when an attacker's fake positions suggest a gradual slowdown or speedup, but stay on the receiver's actual track.

6 IMPLEMENTATION

Our implementation follows a branch-and-bound algorithm for collective detection (CD) [3] with our modifications to find several likely points in iterations, as described in the previous section. We implemented the algorithm in CUDA, leveraging the parallelism of GPUs. The selection of minimal data types further increases the computation performance and reduces memory usage. For instance, 16-bit indices for the location hypotheses grid enable more cached hypotheses. To improve the localization accuracy, we account for the received signals' atmospheric delays. And we found that more consistent results can be achieved when scaling the raw input signals and satellite acquisition vectors to the value range $[-1, 1]$.

6.1 Acquisition

In the acquisition stage, which separates the received signal into the satellite components, two important parameters are the bin width for the Doppler shift frequency search and the correlation length. A bin width of 500 to 667 Hz suffices for most applications [15]. Since the run time of the acquisition is negligible in our implementation, we use a fine resolution of 200 Hz. For the correlation, at least one millisecond is required to fit a whole code period. Longer correlations can significantly increase the SNR. Unfortunately, we cannot choose an arbitrarily long correlation: Every 20 ms, a navigation bit flip can happen, which degrades the correlation. To make sure that at least one millisecond of data without a bit flip is available for each satellite, CTN receivers capture at least two consecutive milliseconds of signal [26]. In our experiments (cf. Sec. 7), a correlation length of 3 ms provides the best results. The resulting correlation vectors are reduced to a single vector with one millisecond length for each satellite: Point by point, we add 1 ms long vector snippets within each frequency bin to form one millisecond long vectors. Then, we combine all frequency bins by selecting the maximum value at each vector index. We found that this yields consistent results with good SNR.

6.2 Signal Dampening

To save recomputing the acquisition in each algorithm iteration, we do the peak dampening directly in the correlation vectors instead of in the raw received signal. For each satellite, we reduce the maximum peak by 60 %. We experimentally determined this value to work well with the TEXBAT dataset. The triangular peak shapes resulting from oversampling (see Fig. 2) are reconstructed and then subtracted from the correlation vectors. When peaks overlap, similar to Figure 2, only the non-overlapping part of the peak is dampened. Before using the modified vectors for the next algorithm iteration, the vectors are normalized to the range $[-1, 1]$ again.

7 EXPERIMENTS

Ideally, anti-spoofing methods would be tested using a receiver in the real world and an attacking transmitter. But since transmitting GPS signals in the real world may interfere with receivers outside the experimental setup, this is not a good idea. Indeed, in all countries that we know of, it is forbidden to transmit GPS signals. Instead of resorting to a full simulation, we evaluate our method on the TEXBAT dataset, which is a hybrid consisting of generated attacking signals which are physically added over a wire to legitimate signals collected in the real world.

TEXBAT. In 2012, Humphreys et al. presented *TEXBAT*, the first public dataset of GPS scenarios with spoofing attacks [19]. So far, it has been the de facto standard for any GPS spoofing research [37, 46]. TEXBAT contains two "clean" signal recordings without any spoofing and 8 spoofing scenario recordings. The spoofing scenarios are constructed based on the clean recordings. The first clean recording is in a stationary setting with an antenna placed on top of an university building. The second clean recording is a dynamic recording from an antenna mounted on a car driving across a city. The spoofing scenarios are produced by replaying one of the clean datasets and adding counterfeit signals from a signal generator. Those combined signals are recorded using signal capture hardware. The counterfeit signals are generated with appropriate characteristics to be as representative as possible for all currently known attack techniques. The TEXBAT dataset contains complex samples with 16-bit quantization, sampled at a rate of 25 MHz [19]. Table 1 gives an overview of the TEXBAT spoofing scenarios. While in Scenarios 4 and 6 a *location* error of approximately 600 m to the north is introduced, all other scenarios introduce a *time* error of about 2 µs. Scenarios 5 and 6 are most difficult, as environmental effects like multipath will vary during the recording. Such effects could modify authentic signals in a way that they might be mistaken for spoofed signals. Also, for those dynamic scenarios, no ground truth is available, so we evaluate against the track computed from the clean dataset.

For our experiments, we extract snapshots from the TEXBAT scenarios. For every second of a recording, five windows of 9 ms length are extracted and the localization results are averaged over those five windows. So, each localization uses a total of 45 ms of signal data. Average, median, and maximum errors as well as the variance of the location estimates, compared to the respective

Table 1: TEXBAT Spoofing Scenarios

	time-push	location-push
stationary	Scenarios 1,2,3,7,8	Scenario 4
moving	Scenario 5	Scenario 6

clean scenarios, are summarized in Table 2. As the TEXBAT dataset contains at most one set of authentic satellite signals and at most one set of spoofed signals, we show the results of two algorithm iterations. The first iteration is equivalent to a run of the basic branch-and-bound CD algorithm. The second iteration uses the modified signals with the dampened high-power signal components. So, since at most two sets of signals are present, either our first or second algorithm iteration should find the correct receiver location. Which iteration that is, depends on the relative signal power of the authentic versus the spoofed signals. The first iteration finds the location pertaining to the stronger set of signals.

Attacks on classic GPS receivers usually require higher received signal power of the spoofed signals versus the authentic signals. Otherwise, the receivers will likely not switch from tracking the authentic signals to tracking the spoofed signals. Therefore, with our algorithm, we would expect that the first iteration finds the spoofed location and the second iteration finds the true receiver location. However, it does not matter which iteration performs better. For some datasets, e.g. Scenario 4, the first iteration is more accurate, that is, it finds the correct receiver location, while for other Scenarios, e.g. Scenario 6, the second iteration finds the correct receiver location. First, note that the interpretation of the two signals – authentic and spoofed – present in each dataset could be exchanged, in which case the results of the two iterations would be swapped. In the TEXBAT scenarios, the power differences between the authentic and spoofed signals are fixed. So, for each scenario, the correct and spoofed locations are always found in the same algorithm iteration. However, in practice, an attacker could vary the transmit power of the spoofed signals, such that for some location computations, the correct location is found in the first iteration and for other location computations, the correct location is found in the second iteration. Also, the received power of the authentic and spoofed signals can vary for each location computation due to obstacles blocking the line of sight (LOS) path between the receiver and some satellites or the attacker. Especially, since the attacker may be located on the ground, its signals may inadvertently be blocked by terrain or buildings. Another option for the spoofed signals resulting in a lower likelihood in CD than the authentic signals, despite being received with more power, is that the attacker does not model atmospheric delays accurately and thus the spoofed signals arrive with inconsistent delays at the receiver. In essence, a receiver has to decide for each location computation, which of the found positions in all iterations is the correct one. In Section 5.2, we discuss how to identify correct locations using GPS-external data.

7.1 TEXBAT Time-Push Scenarios

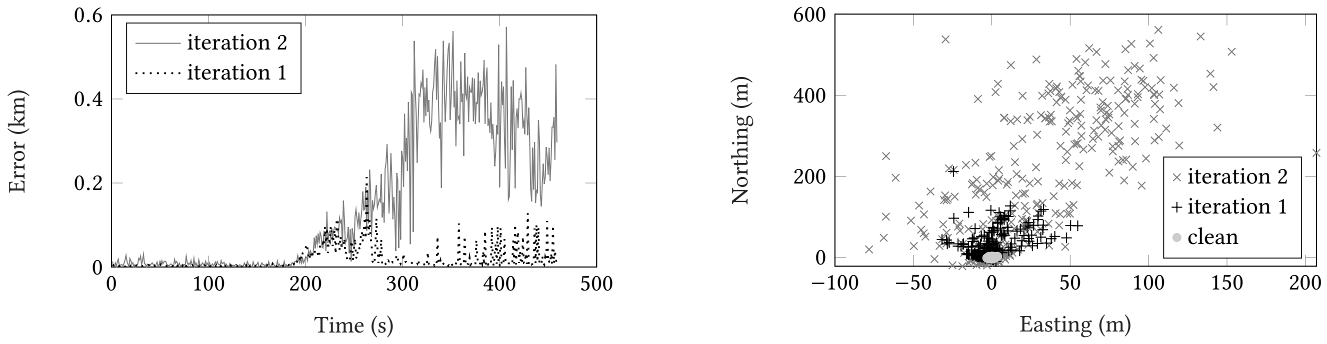
Scenario 1 contains a *switch* attack in which the original signals are switched for counterfeit signals. In that scenario, while it might be possible to detect whether spoofing is happening or not by analyzing the raw data, it is impossible to recover the original signals as they are not present once the spoofing starts. Scenarios 2 and 5 contain *overpowered* attacks in which the adversary adds the spoofing signals with a 10 dB power advantage over the authentic signals. Such an attack could be detected by a sudden in-band power increase. In Scenarios 3, 7 and 8, the spoofer attempts to match the authentic signals' power [18]. In those *matched-power* attacks, the adversary signals only have a 1.3 dB power advantage.

Table 2: Median, average and maximum errors and variance of our localizations computed with two algorithm iterations on each TEXBAT scenario. Units are in meters and for the variance in m^2. Location-push scenarios are marked in bold.

Sce-	Iteration 1				Iteration 2			
nario	med	avg	max	var	med	avg	max	var
1	4	5	32	14	10	13	182	165
2	3	3	9	3	3	4	25	12
3	5	8	78	104	19	51	552	5135
4	**9**	**23**	**214**	**911**	**155**	**190**	**572**	**28443**
5	7	21	348	1659	7	14	156	548
6	**405**	**385**	**559**	**21142**	**19**	**34**	**222**	**1315**
7	3	4	10	4	15	46	438	5138
8	4	5	45	30	19	63	630	10596

The time-push scenarios from the TEXBAT dataset are of limited use for testing our algorithm. Compared to the classic least-squares GPS localization method, the CD algorithm is less sensitive to time shifts. In the least-squares approach, a satellite's distance is measured through the arrival time of a message transmit timestamp encoded in the satellite signal. A timing error is therefore linked to a distance error by the signal speed, which is the speed of light. In CD, the message timestamps are ignored. Rather, satellite distances at a given time are computed from the known satellite orbit data. The relative speed of a satellite to an observer on Earth is less than 1 m/ms [43]. Thus, our algorithm is neither substantially affected by nor does it detect the induced time errors of up to 2 μs. Still, time errors result in a slight shift of the satellites' signal peaks, misaligning them in the CD algorithm (cf. Fig. 3b). Thus, the receiver location's likelihood and the localization's signal-to-noise ratio (SNR) are lowered. Therefore, with time errors, we can expect a higher localization variance than with a clean signal.

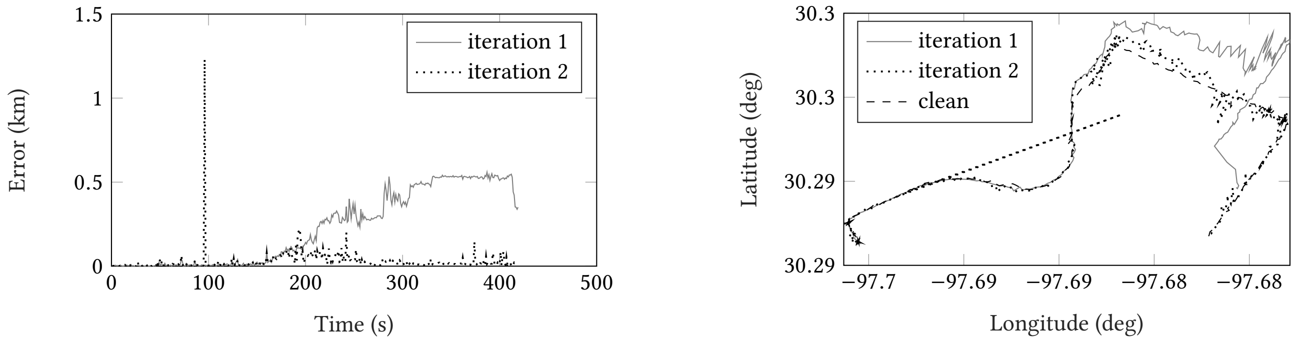
Table 2 shows that our algorithm produces stable results and finds the correct location with high accuracy. The average and median errors for the static time-push scenarios (Sc. 1,2,3,7,8) stay below 7.6 m for the locations from all first algorithm iterations, which is even better than the results achieved with the original branch-and-bound algorithm for benign scenarios [3]. The difference could be due to a more sensitive receiver being used to produce the TEXBAT dataset. The second iteration of our algorithm produces worse results, with the maximum error increasing significantly. This is expected: The authentic and spoofed signals are separated by up to 2 μs and thus point to locations only a few millimeters apart (see above). With the dampening of the stronger signals, the weaker signals with a lower SNR remain for the second iteration. Interestingly, in Scenario 5 the second iteration's result is better. One possible explanation for this is the influence of environmental effects, such as multipath from nearby cars.



(a) Location differences between clean and spoofed datasets.

(b) Computed ground locations.

Figure 5: Results for the TEXBAT Scenario 4. *Static matched-power location-push* scenario with 0.4 dB spoofing power advantage. At 200 to 300 s, the takeover attack has a negative impact on the accuracy. Afterwards, the first iteration finds the authentic location with little error. In this scenario, the second iteration tracks the spoofed location as the spoofer only has a temporary power advantage.



(a) Location differences between clean and spoofed datasets.

(b) Computed ground tracks.

Figure 6: Results for the TEXBAT Scenario 6. *Dynamic matched-power location-push* scenario comparable to Scenario 4 but with 0.8 dB spoofing power advantage and based on the dynamic dataset. Iteration 1 tracks the spoofed location while iteration 2 tracks the authentic location. The high error at 99 seconds is likely a data artifact from the TEXBAT dataset and vanishes with other correlation lengths. This erroneous localization leads to the straight dashed line in the right-hand plot.

7.2 TEXBAT Location-Push Scenarios

As CD algorithms such as ours are indifferent to small time offsets, the location-push scenarios are more interesting. Only two TEXBAT scenarios contain location spoofing. Scenario 4 contains a *matched-power* attack with a spoofing power advantage of 0.4 dB and frequency locking of the spoofed to the authentic signals. Scenario 6 is similar, but based on the dynamic dataset.

For the first 180 s of the static datasets and first 100 s of the dynamic datasets, no spoofing signals are present. This allows classic receivers to start tracking the authentic signals. With these benign signals, our location estimates are accurate (see Figures 5a and 6). When the spoofer gradually starts introducing the location error, the satellite signal peaks get broader, which increases location error and variance. This happens approximately in between 180 s and 280 s for Scenario 4 (Fig. 5) and in between 150 s and 250 s for Scenario 6 (Fig. 6). Once the spoofed and true locations differ enough for the counterfeit and authentic satellite signals to become visible individually, the location estimates start to diverge. One algorithm iteration finds the spoofed location while the other iteration recovers the true location. Due to the reduced SNR, the true location is not recovered perfectly, but the error remains small.

Table 2 lists the error statistics for both location estimates compared to the clean recordings. A maximum location error of 222.4 m and a median error of 18.8 m are not exceeded. This assumes that the receiver selects the localization solutions from the correct iterations (cf. Sec. 5.2). SPREE, the best anti-spoofing work that we are aware of, can only *detect* spoofing attacks, but may not find the actual receiver location and has a maximum location offset evading detection greater than 1.5 km [37].

It can be observed that the results from our algorithm's first iteration are far better for Scenario 4. But a spoofer with constant power advantage should lead to worse results for the first iteration compared to the second, since the spoofed location is found first.

Figure 19 of [19] shows indeed that the spoofer has a power advantage only for roughly the first 60 s of its attack lasting about 240 s. A classic receiver whose tracking loops have already been acquired by the spoofer would continue tracking the spoofed signals whereas our algorithm falls back to the stronger, authentic signals. Scenario 6 does not lead to such behavior as the spoofer has a continuous power advantage.

7.3 Correlation Length

For the presented results, a correlation length of 3 ms length is used. Figure 7 shows the average, median and maximum error of Scenario 6 for different correlation lengths. The best results are achieved with a correlation length of 3 ms. Foucras et al. present that 5 ms should be the ideal trade-off between long correlation length for high SNR and low probability of bit flips [14]. In our case, the accuracy is similar with 3 and 5 ms. One reason for the good performance of 3 vs. 5 ms might be that longer correlation lengths increase the absolute SNR advantage of the spoofed signals, because those are slightly stronger than the authentic signals. This could worsen the results for the authentic location found in the second iteration of our algorithm.

7.4 Computation Time

Currently, the algorithm is optimized for robustness rather than speed. However, depending on the required update rate it is possible to use the algorithm in real-time applications. The computation speed is mainly dependent on the size of the search grid, the number of visible satellites in the signal, the sampling rate of the recording and its SNR. The SNR influences the efficiency of the branch-and-bound technique, as a higher SNR allows earlier detection of uninteresting search regions, allowing discarding more location hypotheses without evaluating them.

Currently, two limitations impact the performance directly. Computing the likelihood of each point is bounded by memory speed. This means that doubling the sampling frequency, and thus the amount of data, also doubles the computation time. Currently, at least half of the computation time is used for computing the likelihoods. The second performance limitation is the calculation of grid points and code phases. This accounts for about one third of the computation time. Experiments with pre-calculated satellite orbits show that this could be reduced significantly. Sorting and filtering the points consumes the remaining computation time. Calculating the satellite acquisition results is negligible.

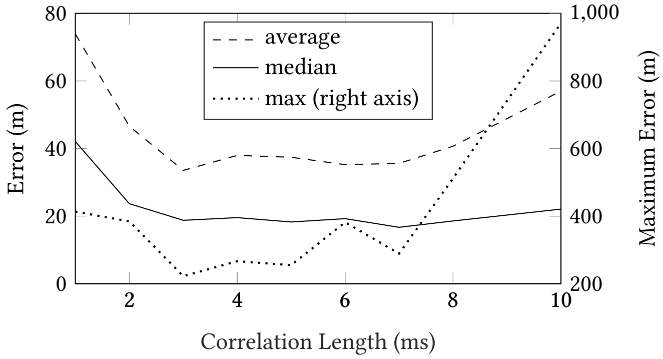


Figure 7: Average, median and maximum error for different correlation lengths in Scenario 6. The algorithm shows best results with 3 ms correlation length. Good results are also achieved with 4 and 5 ms correlation length. Note that the maximum error is shown at a different scale.

A tracking feature, which feeds back the previous location estimate, allows the receiver to reduce the search space. But due to branch and bound, the computation time is reduced by only 30 % in the tested scenarios, although the search space is two magnitudes smaller. In our setup, which primarily runs our algorithm implementation on an NVIDIA GTX 1080 graphics card, tracking allows us to perform an algorithm iteration in about 1.0 s. As we run two iterations, the computation time per localization is 2.0 s. We calculated 2349 localization for the static scenarios and 2090 localizations for the dynamic scenarios, taking roughly 80 min and 70 min, respectively. Such a two-second delay should be acceptable for real-time applications like car, ship and pedestrian navigation. Even for aircraft navigation and routing, this might suffice, as long as the aircraft do not change their course, such that short-time location extrapolations are possible. For fast maneuvers, a higher location update frequency may be required. Code optimizations and more powerful computing hardware should allow covering such scenarios, too. However, note that the signal correlation vectors with mainly random entries and one or a few sharp peaks (cf. Figures 1 and 2) lead to a non-convex CD likelihood distribution which is not amenable to classic, fast optimization algorithms.

8 CONCLUSION

GPS spoofing is a broad topic and many methods have been proposed to detect and mitigate spoofing. Most research focuses on the detection of spoofing attacks. Methods for spoofing mitigation are often specialized or only work for certain scenarios.

Our implementation and evaluation shows that with some modifications, the robustness of collective detection can be exploited to mitigate spoofing attacks. We show that multiple locations, including the actual one, can be recovered from scenarios in which several signals are present. Experiments based on the TEXBAT dataset show that a wide variety of attacks can be mitigated. In the TEXBAT scenarios, an attacker can introduce a maximum error of 222 m and a median error under 19 m. This is less than a sixth of the maximum unnoticed location offset reported for SPREE, that only *detects* spoofing attacks [37]. Compared to SPREE, which is the most accurate spoofing *detection* method known to us, our method also *mitigates* attacks by finding the correct receiver location. Further, our method is based on CD, a completely different approach from classic GPS receivers, and as such has even more potential.

Since our method does not track signals, but works with signal snapshots, our spoofing mitigation method is suitable for snapshot receivers, which are a new class of low-power GPS receivers [11, 26]. To date, no snapshot receiver method with anti-spoofing capabilities has been presented. Our method is also compatible with classic receivers by splitting the data stream into windows with a duration of a few milliseconds. Thus, our spoofing mitigation method can be used with any GPS receiver and does not require special hardware.

ACKNOWLEDGMENTS

We sincerely thank Penina Axelrad and the anonymous reviewers for their valuable suggestions and feedback.

REFERENCES

[1] Anonymous. 2014. *Timing and Synchronization for LTE-TDD and LTE-Advanced Mobile Networks.* Technical Report. Microsemi. https://www.microsemi.com/document-portal/doc_download/133615-timing-sync-for-lte-tdd-lte-a-mobile-networks.

[2] Penina Axelrad, Ben K Bradley, James Donna, Megan Mitchell, and Shan Mohiuddin. 2011. Collective Detection and Direct Positioning Using Multiple GNSS Satellites. *Navigation* 58 (2011).

[3] Pascal Bissig, Manuel Eichelberger, and Roger Wattenhofer. 2017. Fast and Robust GPS Fix Using One Millisecond of Data. In *16th International Conference on Information Processing in Sensor Networks (IPSN).* ACM/IEEE.

[4] Ali Broumandan, Ali Jafarnia-Jahromi, and Gérard Lachapelle. 2015. Spoofing detection, classification and cancelation (SDCC) receiver architecture for a moving GNSS receiver. *GPS Solutions* 19 (2015).

[5] Matt Burgess. 2017. *When a tanker vanishes, all the evidence points to Russia.* https://www.wired.co.uk/article/black-sea-ship-hacking-russia

[6] Antonio Cavaleri, Beatrice Motella, Marco Pini, and Maurizio Fantino. 2010. Detection of spoofed GPS signals at code and carrier tracking level. In *5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC).* IEEE.

[7] Joon Wayn Cheong, Jinghui Wu, Andrew G. Dempster, and Chris Rizos. 2011. Efficient Implementation of Collective Detection. In *IGNSS Symposium.* International Global Navigation Satellite Systems Society.

[8] Pau Closas, Carles Fernández-Prades, and Juan A Fernández-Rubio. 2007. Maximum likelihood estimation of position in GNSS. *IEEE Signal Processing Letters* 14 (2007).

[9] European Commission. 2011. Commission Implementing Regulation (EU) No 1207/2011. *Official Journal of the European Union* 305 (2011). https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2011:305:0035:0052:EN:PDF.

[10] European Commission. 2017. Commission Delegated Regulation (EU) 2017/574 of 7 June 2016 supplementing Directive 2014/65/EU of the European Parliament and of the Council with regard to regulatory technical standards for the level of accuracy of business clocks. *Official Journal of the European Union* L87 (2017).

[11] Manuel Eichelberger, Ferdinand von Hagen, and Roger Wattenhofer. 2019. Multi-Year GPS Tracking Using a Coin Cell. In *20th International Workshop on Mobile Computing Systems and Applications (HotMobile).* ACM.

[12] Marc Elsberg. 2012. *BLACKOUT-Morgen ist es zu spät: Roman.* Blanvalet Verlag.

[13] NASPI Time Synchronization Task Force. 2017. *Time Synchronization in the Electric Power System.* Technical Report. North American Synchrophasor Initiative. https://www.naspi.org/sites/default/files/reference_documents/tstf_electric_power_system_report_pnnl_26331_march_2017_0.pdf

[14] Myriam Foucras, Bertrand Ekambi, Fayaz Bacard, Olivier Julien, and Christophe Macabiau. 2014. Optimal GNSS Acquisition Parameters when Considering Bit Transitions. In *Position, Location and Navigation Symposium (PLANS).* IEEE/ION.

[15] Bernhard C Geiger and Christian Vogel. 2013. Influence of Doppler Bin Width on GPS Acquisition Probabilities. *IEEE Trans. Aerospace Electron. Systems* 49 (2013).

[16] Mark Goldstein, Joseph Kirschbaum, Sally Moino, Glenn Davis, Eli Albagli, Melissa Bodeau, Katherine Davis, Richard Hung, Bert Japikse, SaraAnn Moessbauer, Josh Ormond, Nalylee Padilla, and Daniel Rodriguez. 2013. *GPS DISRUPTIONS – Efforts to Assess Risks to Critical Infrastructure and Coordinate Agency Actions Should Be Enhanced.* Technical Report. United States Government Accountability Office.

[17] US Federal Government. 2018. Title 14 – Aeronautics and Space. *Code of Federal Regulations* (2018).

[18] Todd Humphreys. 2016. *TEXBAT DATA SETS 7 AND 8.* Technical Report. University of Texas at Austin.

[19] Todd E. Humphreys, Jahshan a. Bhatti, Daniel P. Shepard, and Kyle Douglas Wesson. 2012. The Texas Spoofing Test Battery: Toward a Standard for Evaluating GPS Signal Authentication Techniques. *25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS)* (2012).

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

[21] Global Positioning Systems Directorate Systems Engineering & Integration. 2013. *Interface Specification IS-GPS-200: Navstar GPS Space Segment/Navigation User Interfaces, Revision H.* Technical Report. http://www.gps.gov/technical/icwg/IS-GPS-200H.pdf

[22] William Jackson. 2013. The serious side of GPS, where timing is everything. https://gcn.com/articles/2013/11/12/gps-timing-position.aspx.

[23] Ali Jafarnia-Jahromi, Tao Lin, Ali Broumandan, John Nielsen, and Gérard Lachapelle. 2012. Detection and mitigation of spoofing attacks on a vector-based tracking GPS receiver. *ION ITM* (2012).

[24] Zhengxuan Jia. 2016. A type of collective detection scheme with improved pigeon-inspired optimization. *International Journal of Intelligent Computing and Cybernetics* 9 (2016).

[25] Bernhard Krach, Michael Lentmaier, and Patrick Robertson. 2008. Joint Bayesian positioning and multipath mitigation in GNSS. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on.* IEEE.

[26] Jie Liu, Bodhi Priyantha, Ted Hart, Yuzhe Jin, Woosuk Lee, Vijay Raghunathan, Heitor S. Ramos, and Qiang Wang. 2016. CO-GPS: Energy Efficient GPS Sensing with Cloud Offloading. *IEEE Transactions on Mobile Computing* (2016).

[27] Sherman Lo, David De Lorenzo, Per Enge, Dennis Akos, and Paul Bradley. 2009. Signal authentication: A secure civil GNSS for today. *Inside GNSS* 4 (2009).

[28] Michael Lombardi. 2015. *Delivering NIST Time to Financial Markets Via Common-View GPS Measurements.* https://www.gps.gov/cgsic/meetings/2015/lombardi.pdf

[29] Michael Lombardi. 2017. Time and Frequency Traceability in Emerging Technologies: Synchronizing Financial Markets. https://tf.nist.gov/sim/2017_Seminar/SIM_2017_Time_in_Financial_Markets.pptx.

[30] Gustavo López-Risueño and Gonzalo Seco-Granados. 2005. CN/sub 0/estimation and near-far mitigation for GNSS indoor receivers. In *61st Vehicular Technology Conference.* IEEE.

[31] Premala H Madhani, Penina Axelrad, Kent Krumvieda, and John Thomas. 2003. Application of Successive Interference Cancellation to the GPS Pseudolite Near-Far Problem. *IEEE Trans. Aerospace Electron. Systems* 39 (2003).

[32] Greg Milner. 2016. *Death by GPS.* Ars Technica. https://arstechnica.com/cars/2016/05/death-by-gps/

[33] A Neri, F Rispoli, and P Salvatori. 2015. An analytical assessment of a GNSS-based train integrity solution in typical ERTMS level 3 scenarios. In *European Navigation Conference (ENC).*

[34] Yuting Ng and Grace Xingxin Gao. 2016. Mitigating jamming and meaconing attacks using direct GPS positioning. In *Position, Location and Navigation Symposium (PLANS).* IEEE/ION.

[35] Aron Pinker and Charles Smith. 1999. Vulnerability of the GPS Signal to Jamming. *GPS Solutions* 3 (1999).

[36] Mark L Psiaki and Todd E Humphreys. 2016. GNSS Spoofing and Detection. *Proc. IEEE* 104 (June 2016).

[37] Aanjhan Ranganathan, Hildur Ólafsdóttir, and Srdjan Capkun. 2016. Spree: A spoofing resistant gps receiver. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking.* ACM.

[38] Desmond Schmidt, Kenneth Radke, Seyit Camtepe, Ernest Foo, and Michał Ren. 2016. A survey and analysis of the GNSS spoofing threat and countermeasures. *ACM Computing Surveys (CSUR)* 48 (2016).

[39] Seong-Hun Seo, Byung-Hyun Lee, Sung-Hyuck Im, and Gyu-In Jee. 2015. Effect of Spoofing on Unmanned Aerial Vehicle using Counterfeited GPS Signal. *Journal of Positioning, Navigation, and Timing* 4 (2015).

[40] Daniel P Shepard, Jahshan A Bhatti, Todd E Humphreys, and Aaron A Fansler. 2012. Evaluation of smart grid and civilian UAV vulnerability to GPS spoofing attacks. In *Radionavigation Laboratory Conference Proceedings.*

[41] James J. Spilker. 1996. Fundamentals of Signal Tracking Theory. *Progress in Astronautics and Aeronautics* 163 (1996). Chapter 7 in *Global Positioning System Theory and Applications*, Volume 1, Parkinson, B. et al.

[42] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. 2011. On the requirements for successful GPS spoofing attacks. In *Conference on Computer and Communications Security (CCS).* ACM.

[43] James Bao-Yen Tsui. 2000. *Fundamentals of Global Positioning System Receivers.* Wiley-Interscience New York, NY, USA.

[44] Frank Stephen Tromp Van Diggelen. 2009. *A-GPS: Assisted GPS, GNSS, and SBAS.* Artech House.

[45] Fei Wang, Hong Li, and Mingquan Lu. 2015. ARPSO-MLE based GNSS anti-spoofing method. In *Signal Processing, Communications and Computing (ICSPCC), 2015 IEEE International Conference on.* IEEE.

[46] Kyle Douglas Wesson. 2014. *Secure Navigation and Timing Without Local Storage of Secret Keys.* Ph.D. Dissertation. University of Texas at Austin.

[47] Gao Yan and Li Qing. 2013. Closely spaced multipath mitigation in GNSS receiver based on maximum likelihood estimation. In *2013 International Conference on Wireless Communications and Signal Processing (WCSP).* IEEE.

[48] Dingbo Yuan, Hong Li, and Mingquan Lu. 2014. A method for GNSS spoofing detection based on sequential probability ratio test. In *Position, Location and Navigation Symposium (PLANS).* IEEE/ION.