Location-independent GNSS Relay Attacks: A Lazy Attacker's Guide to Bypassing Navigation Message Authentication

Maryam Motallebighomi
Northeastern University
Boston, MA, USA

Harshad Sathaye
Northeastern University
Boston, MA, USA

Mridula Singh
CISPA Helmholtz Center for Information Security
Saarbrucken, Germany

Aanjhan Ranganathan
Northeastern University
Boston, MA, USA

ABSTRACT

In this work, we demonstrate the possibility of spoofing a GNSS receiver to arbitrary locations without modifying the navigation messages. Due to increasing spoofing threats, Galileo and GPS are evaluating broadcast authentication techniques to validate the integrity of navigation messages. Prior work required an adversary to record the GNSS signals at the intended spoofed location and relay them to the victim receiver. Our attack demonstrates the ability of an adversary to receive signals close to the victim receiver and in real-time generate spoofing signals for an arbitrary location *without modifying* the navigation message contents. We exploit the essential *common reception* and *transmission time* method used to estimate pseudorange in GNSS receivers, thereby potentially rendering any cryptographic authentication useless. We build a proof-of-concept real-time spoofer capable of receiving authenticated GNSS signals and generating spoofing signals for any arbitrary location and motion without requiring any high-speed communication networks or modifying the message contents. Our evaluations show that it is possible to spoof a victim receiver to locations as far as 4000 km away from the actual location and with any dynamic motion path. This work further highlights the fundamental limitations in securing a broadcast signaling-based localization system even if all communications are cryptographically protected.

CCS CONCEPTS

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

KEYWORDS

Navigation Message Authentication, Relay Attack, Delay

ACM Reference Format:
Maryam Motallebighomi, Harshad Sathaye, Mridula Singh, and Aanjhan Ranganathan. 2023. Location-independent GNSS Relay Attacks: A Lazy Attacker's Guide to Bypassing Navigation Message Authentication. In *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '23), May 29-June 1, 2023, Guildford, United Kingdom.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3558482.3590186

WiSec '23, May 29-June 1, 2023, Guildford, United Kingdom

1 INTRODUCTION

Global Navigation Satellite Systems (GNSS) such as Galileo [2], GPS [5], and GLONASS [3] are critical to a wide variety of applications ranging from navigation and tracking to modern communication and networking systems. It is well-known that civilian GNSS is vulnerable to signal spoofing attacks with increasing spoofing incidents observed in the wild [11]. In a GNSS spoofing attack, an adversary transmits radio-frequency signals that imitate legitimate satellite signals specifically crafted to force a receiver to compute a false location. With the widespread availability of low-cost software-defined radio and public repositories [30], the cost to spoof GPS signals has been significantly lowered (less than $100). Prior work has shown the possibility of changing the course of autonomous aerial [46], terrestrial [12], and aquatic [8] vehicles by simply spoofing GNSS signals. Moreover, there are an increasing number of GPS signal interference and spoofing incidents [11] being reported. For example, thousands of ships and GPS devices in Shanghai were suspected to be affected by GPS spoofing. It is also suspected that GPS spoofing resulted in several boats transmitting signals indicating they were sailing in circles off the California coast [13]. In reality, they were thousands of miles away. The lack of message authentication permits the generation of fake satellite signals that can falsify a receiver's location. To this extent, several countermeasures based on cryptographic authentication [56] to protect against attackers generating spoofing signals are being proposed. For example, the recently launched Galileo's Open Service Navigation Message Authentication (OSNMA) [15] authenticates the navigation message contents based on the TESLA protocol [49] and one-way hash functions. To modernize next-generation GPS, the United States Department of Defense is also exploring using Chips Message Robust Authentication (CHIMERA) [22]. Galileo's OSNMA and CHIMERA digitally sign the navigation message contents and include the MAC within the message itself. The above countermeasures aim to protect the integrity of the navigation message contents. Since in GNSS, the user's location is computed based on both the navigation message contents *and* its time of arrival, such localization is still vulnerable to signal relay/replay attacks.

In this work, we analyze the security guarantees of authenticated GNSS signals and show that attackers can spoof receivers to any location independent of the cryptographic primitive implemented. Through this work, we aim to raise awareness of the fundamental limitations of the proposed architectures and drive the research community to address these drawbacks in time for broad public

deployment. Prior works [38, 48, 52] showed the possibility of relaying GNSS signals (meaconing attacks) to spoof the victim receiver's location. However, most of these works are limited to spoofing the victim receiver's location to where the legitimate signals were originally recorded [38]. In this work, we demonstrate the ability to spoof both arbitrary static and dynamic GNSS locations in real time without modifying the contents of the navigation message and discuss its effectiveness in potentially bypassing navigation message authentication schemes. Our proposed scheme has three vital components; i) NAVMSG streamer, ii) delay estimator and iii) spoofing signal synthesizer. These three components are collectively responsible for extracting the navigation bits and synthesizing the spoofing signal using the extracted bits. Conventional receivers output navigation message contents every 6 s, making it harder to circumvent the time-binding of navigation message authentication primitives. So generating the spoofing signals without decoding the entire satellite navigation message is necessary. To overcome this challenge, we designed NAVMSG streamer to output the navigation message bits (note there is no encryption but only authentication) as it gets decoded every 20 ms. Next, the delay estimator exploits the concept of *relative offsets* fundamental to *common reception time* and *common transmission time* methods of calculating pseudoranges to calculate necessary delays required to achieve relative offsets corresponding to the desired spoofed location and time. These delays are calculated in *real-time* and are represented as code-phase offsets. Since the satellites are continuously in motion, our delay estimator is designed to update these relative offsets continuously. Finally, the spoofing signal synthesizer uses the PRN codes, relative offsets calculated by the delay estimator, and other physical layer parameters like Doppler shift, carrier phase, and amplitude to re-modulate the navigation bits provided by the NAVMSG streamer. The three aforementioned components enable an attacker to spoof a victim receiver to a location even hundreds of kilometers from its true location by temporally manipulating satellite signals. Additionally, our attacker setup can generate and spoof dynamic motion paths independent of its location in real-time and without physically moving. Our real-time setup has a processing delay of ≈4.3 ms [1] to generate spoofing signals from legitimate satellite signals, making the current delayed key-disclosure schemes incapable of detecting the attack. Specifically, we make the following contributions:

- We designed and developed a real-time location spoofer using available software-defined radio platforms (less than $1500). Our setup can receive legitimate GNSS signals and generate spoofing signals for any arbitrary location and motion in real-time.
- We successfully tested our attack on a commercial receiver (ublox M8N [17], ublox ZED-F9 [19], Septentrio mosaic-go [7]) and a software-defined GNSS receiver (GNSS-SDR). We showed that it is possible to spoof a victim receiver without requiring any high-speed relay network or manipulating the message contents. We demonstrated our real-time setup in a video [2].
- In this paper, we demonstrate the ability to spoof both arbitrary static and dynamic GNSS locations in real time without modifying the contents of the navigation message, thereby rendering any cryptographic authentication useless. As a proof of concept for spoofing dynamic motion, we generate a 2.43 km dynamic motion path around a water reservoir 5.5 km away from the true location (also where the legitimate signals were recorded).
- We show that the generated signals pass satellite augmentation systems-based integrity checks implemented in commercial receivers [19]. It is important to note that even though cryptographic signatures are transmitted as part of the navigation messages, as of today, commercial receivers lack the necessary infrastructure to validate and verify the signatures.
- We evaluate the effect of different parameters on the accuracy and performance of the attack, e.g., the attacker's sampling rate, satellite constellation, and orbital motion.
- We discuss the limitations and possible countermeasures.
- Our implementation is publicly available[3] to the community.

2 BACKGROUND

2.1 GNSS Spoofing Attacks

A GNSS signal spoofing attack is a physical-layer attack where an attacker transmits specially-crafted radio signals identical to legitimate satellite signals to force a victim receiver to compute a wrong location and/or time. GPS, Galileo, GLONASS, and Beidou are all vulnerable to spoofing attacks due to the lack of signal authentication and publicly available pseudorandom codes, signal modulation schemes, and data-frame formats. Today, commercial signal generators [6] can even transmit multiple GNSS signals simultaneously, and low-cost software-defined radio platforms [1] and open-source GPS signal generation software [9] make it possible to execute a signal spoofing attack with less than $100 of hardware equipment. Adversaries can transmit static or entire trajectories, allowing them to spoof a stationary receiver's location as moving several kilometers away from its actual location.

2.2 Cryptographic Countermeasures

Several cryptographic countermeasures [61] were proposed to prevent spoofing attacks, broadly classified into Navigation message authentication (NMA) and Spreading Code Authentication (SCA). NMA uses digital signatures [27, 39] to authenticate navigation messages, while SCA punctures the public spreading sequence with random symbols (watermarks), later verified by the receiver. The increasing GNSS spoofing threat has forced GNSS operators to upgrade their existing infrastructure, as Europe's Galileo publicly testing its open service navigation message authentication (OSNMA) [15]. The US DoD is also exploring the use of chips message robust authentication (Chimera) to improve GPS security [22].

Galileo's OSNMA is based on an adaption of the original timed efficient stream loss-tolerant authentication (TESLA) protocol [49]. The navigation message is digitally signed and includes the message authentication code using a set of *40 reserved bits*. The key to verifying the MAC is released after a delay, and the key itself can be verified using a previous key generated in the TESLA protocol's one-way chain. An adversary cannot generate the key chain as the root key is kept secret. The keys are released after a delay meaning that the key used in the MAC generation procedure is not released until

[1] 317.485 μs of average processing time and 4 ms for sending the bits to the attacker's transmitter

[2] A video demonstration of this attack is available at https://youtu.be/ylTpEsTCczs

[3] https://www.gnssrelayattack.com/

after the message and MAC are already received. Since OSNMA is based on a delayed key disclosure scheme, loose time synchronization at the receiver is critical and affects the scheme's effectiveness. OSNMA-enabled receivers verify the integrity of the navigation data once the satellite releases the corresponding TESLA chain key. This requires the receiver to be synchronized with a given accuracy to the Galileo system time. According to OSNMA specifications [14, 15, 28], the receiver is required to be in synchronization requirements at the receiver can range from 18 sec to 5 min. The US Air Force Research Laboratory also plans to test Chimera by launching Navigation Technology Satellite–3 (NTS-3) in 2023 [32]. One of the modes of operation in GPS Chimera which will be tested is "NMA-only" mode, specifically designed to enhance tracking performance at the receivers [32].

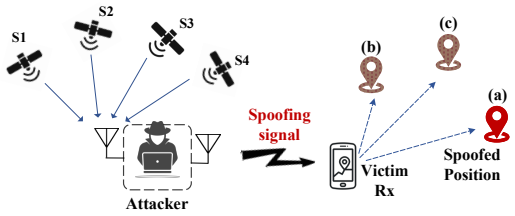

Figure 1: Attack Overview. Attacker receives signals at a fixed location and synthesizes a new signal by applying appropriate delays to the received navigation messages enabling the attacker to spoof the victim receiver to an *arbitrary* location (a, b, or c) or a *trajectory* without any physical movement.

3 TIME-SHIFTING AND RELAYING OF GNSS SIGNALS

3.1 Attacker and System Assumptions

The attacker's overall goal is to manipulate the estimated position at a victim receiver. For example, the attacker is close to the victim receiver and intends to spoof the receiver to a target location l_T, as shown in Figure 1. The attacker's proximity to the victim receiver ensures that they both have a majority of the same visible satellites. Although not necessary, using the same satellites to conduct the attack increases stealth. We assume that the attacker has access to all public information, such as pseudorandom codes, signal modulation schemes, and data-frame formats, and can receive legitimate GNSS signals, even if OSNMA and Chimera are enabled, as the spreading codes are public knowledge. The attacker can also transmit GNSS signals using the modulation and frame format expected at the victim receiver. Finally, we assume that the attacker has sufficient transmission power to overshadow the legitimate signals[4]. We emphasize that the attacker *cannot* modify the contents of the navigation message to maintain integrity and avoid detection by the victim receiver. Hence, unlike today's spoofing attacks, our adversary cannot generate navigation messages in advance.

We assume that the victim's receiver is a standard GNSS receiver capable of decoding GNSS messages and validating the authenticity of the message content (e.g., Galileo OSNMA or GPS Chimera). The victim receiver can access confidential out-of-band information (e.g., Chimera's fast channel mode) and is loosely synchronized as required by the respective authentication scheme. However, we assume that the victim receiver does not implement other non-cryptographic spoofing detection mechanisms, e.g., physical-layer-based spoofing detection techniques [41, 53, 60].

3.2 Attack Overview

First, it is crucial to understand how GNSS receivers process the satellite signals to determine their position. After pre-processing the received signal, the receiver searches for visible satellite messages by correlating its own replica of the pseudorandom code corresponding to each satellite. Once a satellite signal is detected, the receiver switches to tracking and demodulating the navigation message data for that specific satellite. The decoded data estimates the receiver's range from each visible satellite. Note that the satellite clocks are tightly synchronized, and the receiver's clock (not using atomic clocks) contains errors and biases; therefore, we refer to the estimated ranges as pseudoranges. The receiver needs at least four pseudoranges to estimate its position.

To determine a pseudorange, the receiver needs the satellite signal's transmission and reception time. While the transmission time of each subframe is included in the navigational message, reception time estimation [50] is a complex process. As shown in Figure 2a, signal transmission from the satellites is synchronized, but due to varying propagation delays (Figure 2b), they arrive at the receiver at different times. Since the receiver does not have a high-accuracy reference clock as the satellites, the receiver uses the earliest arriving signal as the reference and computes the relative time difference of the remaining satellite signals. The result of this approach is not an absolute range for each satellite but a pseudorange relative to the first arriving reference satellite. Absolute pseudoranges are then estimated assuming a minimal travel time for the reference satellite based on known satellite orbits and typical user altitudes (e.g., for GPS, this is 65 to 85 ms). Such an estimation of pseudoranges is fundamental to all GNSS receivers, and our attack exploits this.

In a signal spoofing attack, an adversary can manipulate position estimation by either modifying the content of the navigation messages or the propagation delay. Since the message content is authenticated, we design our attack strategy to manipulate the reception time estimation method used for the pseudorange estimation. Suppose an adversary records and replays the GNSS signals as shown in prior work [57], i.e., delays all the satellite signals by the same amount. The victim receiver's spoofed location is limited to where the adversary recorded the signal.

Given a set of satellite signals, our attacker continuously calculates and applies appropriate delays to spoof the victim to a specific location. It is important to note that acquiring the legitimate signal and selecting delay values for each satellite signal is time-constrained when using OSNMA and Chimera. Recall that the victim considers the navigation message invalid and discards them once the keys for authentication are released. Furthermore, the satellites are in continuous motion. Therefore, the frequency of delay estimation directly impacts the satellite signals selected for temporal manipulation, and the achieved spoofed location accuracy. The key modules that we design enable the attacker to overcome

[4]GPS's signal strength on the ground is typically -127.5 dBm

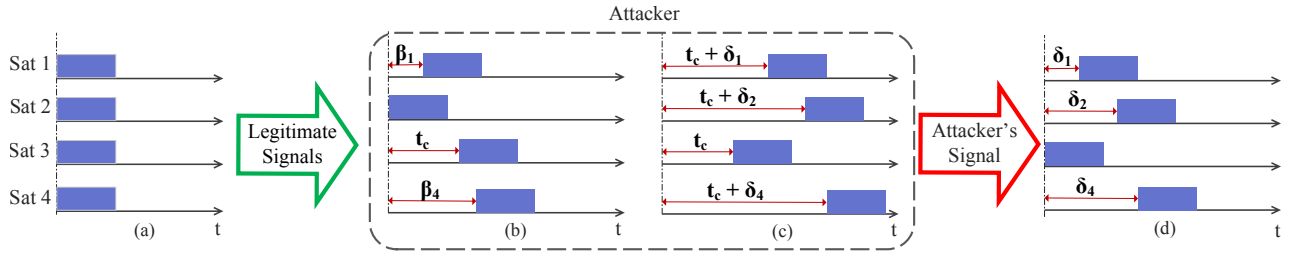

Figure 2: Our attack exploiting common reception time: (a) GNSS satellite transmission time (b) relative time of arrival at the attacker (c) relative times at attacker TX after attacker modifications (d) relative time of arrival at the victim.

these challenges by providing access to navigation messages within the time constraints set by cryptographic countermeasures.

3.3 Key Components of the Attack

Our attack comprises three key components as shown in Figure 3: i) NAVMSG streamer, ii) Delay Estimator, and iii) Spoofing Signal Synthesizer. Authentication mechanisms like OSNMA and Chimera enforce timing constraints, meaning navigation messages arriving after key disclosure will be discarded by the receiver. To achieve this, the NAVMSG streamer exploits the non-necessity of decoding the entire content of the navigation messages, as the attack does not manipulate the navigation message data in any way. The delay estimator module calculates the necessary delays for each visible satellite signal to spoof the victim receiver to a target location. The spoofing signal synthesizer module applies the delays to the satellite signals forwarded by the NAVMSG streamer. It carefully selects the satellite signals to apply the delays and combines them before spoofing the victim receiver during the synthesis process.

NAVMSG streamer: The NAVMSG streamer is responsible for detecting visible satellite navigation messages and streaming them to the spoofing signal synthesizer in real-time. In conventional receiver designs, the navigation message is output as a *receiver observable* after the entire sub-frame is decoded, i.e., the signal has gone through the signal acquisition, demodulation, and decoding process, which takes 6 s for GPS and 30 s for Galileo [35]. In our attack, it is necessary to detect and forward the navigation message signals as fast as possible for temporal manipulation. The goal is to hit the victim receiver with the spoofing signal *before* the revelation of the appropriate authentication key. We design the NAVMSG streamer to directly output the navigation message symbol from the receiver's tracking stage. GNSS receivers perform correlation to identify visible satellite signals and synchronize before decoding the navigation message contents. Our design uses the correlator output directly and streams the value as a single navigation message bit to the spoofed signal synthesizer. This process eliminates the delays caused by other GNSS signal processing blocks.

The NAVMSG streamer can output a single navigation message bit every 20 ms for GPS and every 4 ms for Galileo. GPS messages have a bitrate of 50 bps and 250 bps for Galileo. At this rate, a receiver needs 20 ms and 4 ms to decode an individual navigation bit. Also, the NAVMSG streamer separates each satellite signal using its unique pseudorandom spreading codes to allow the spoofing

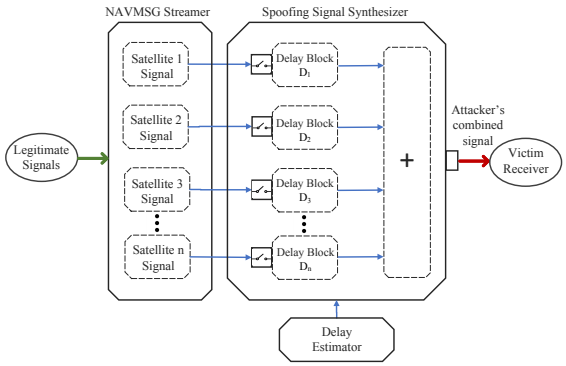

Figure 3: Signal processing pipeline. i) NAVMSG streamer, ii) the delay estimator, and iii) signal synthesizer module.

signal synthesizer to manipulate each satellite signal temporally. For civilian-GNSS signals, the pseudorandom spreading codes are publicly known, allowing the possibility of acquiring each satellite's signals individually. If codes are not publicly available, signals originating from the different satellites can be separated using spatial methods like high gain antennas or antenna arrays [20, 42, 63].

Delay Estimator: The delay estimator calculates the delays to introduce in each satellite's signal received by the attacker at location l_A such that when the spoofing signal is transmitted to the victim receiver, it computes the spoofed target location l_T. To do that, we require the location coordinates $\{x_i, y_i, z_i\}$ of a satellite S_i at time t, which can be assumed to be public knowledge as it is part of the navigation message. Using this information, we can estimate the distance of location l_A and l_T from the satellite S_i as:

$$r_A^i = \sqrt{(x_s^i - x_A)^2 + (y_s^i - y_A)^2 + (z_s^i - z_A)^2} \quad (1)$$

$$r_T^i = \sqrt{(x_s^i - x_T)^2 + (y_s^i - y_T)^2 + (z_s^i - z_T)^2}$$

where $l_A(x_A, y_A, z_A)$ and $l_T(x_T, y_T, z_T)$ denote location coordinates for the attacker receiver and spoofed location. To spoof the location l_T, the attacker delays the signal originating from satellite S_i and received at location l_A by $\Delta\tau_t^i$.

$$\Delta\tau_t^i = (r_T^i - r_A^i - c * t_p)/c \quad (2)$$

where t_p is the attacker's processing delay in receiving the legitimate signal, temporally manipulating, and transmitting it to the victim receiver. We note that t_p is constant for specific attacker hardware and is thus known to the attacker. We emphasize that the calculated delay is independent of the distance between the attacker and victim, as delays only affect relative time offsets (Figure 2). As satellites are continuously moving, the attacker must continually update the calculated delays as they directly impact the victim's obtained position and velocity. Section 5 evaluates the impact of these factors on the victim's estimated location.

Spoofing Signal Synthesizer: The spoofing signal synthesizer combines the individual satellite signals after applying the necessary delays computed by the delay estimator module. In other words, the synthesizer generates the spoofing signal to be transmitted to the victim receiver for the attack. A key function of the spoofing signal synthesizer is to comb through the available satellite signals and select the best satellites to include in the spoofing signal. In Section 5.3, we demonstrate that satellite selection significantly impacts the spoofer's accuracy and performance. For instance, there are more than six visible GNSS satellites at any given time and location. However, using all the satellite signals will limit the maximum spoofable distance from the true location l_A. It is also important to choose a subset of satellite signals that offer the *lowest* geometric dilution of precision (GDOP). GDOP is the geometry of the visible satellites in the sky and is low for a satellite constellation that is more spread out in the sky. Remember that the adversary knows the true location l_A and the spoofed target location (l_T) apriori and, therefore, the visible satellite signals to manipulate temporally. Hence, the attacker can also compute the best subset of satellites with the highest GDOP in advance and use it during the attack. The signal synthesizer also takes care of sanitizing the calculated delays. For instance, the attacker should avoid transmission of the signal where estimated delay $\Delta\tau_t^i < 0$. If the calculated delay $\Delta\tau_t^i$ for some of the chosen satellites is negative, the attacker picks the lowest negative delay value and adds it to all the other delays. The updated delay values will be as follows:

$$\hat{\Delta\tau}_t^i = \Delta\tau_t^i + t_c \tag{3}$$

where t_c is the common code phase offset and is equal to the lowest negative delay. Opting for satellites that are closer to l_A than l_T will lead to smaller values of t_c (more in Section 5).

4 ATTACKER IMPLEMENTATION

In this section, we present the design and implementation challenges and describe how we implement the proposed attacker components that enable us to realize the attacks in real-time and with delays that are well within those described in the standard [30].

The primary requirement of our attack is the ability to manipulate code phase offsets, i.e., temporally shift individual satellite signals to spoof a certain location. This requires the ability to apply time delays to separated satellite signals. To check the feasibility of such an attack, we designed a proof-of-concept where we generate individual satellite signals using a modified version of *gps-sdr-sim*. Next, we calculate the required relative delays and applied them to each satellite signal by leveraging the *delay* block in GNURadio. This proof-of-concept implementation confirms the feasibility of our attack; however, realizing this attack in real time has the following challenges: i) Separating satellite signals, ii) keeping track of satellites' motion relative to the spoofed location, and iii) applying delays and transmission of temporally modified signals within time constraints of the implemented authentication scheme. As mentioned in prior works [57], an attacker can use directional antennas to separate satellite signals at the physical layer. Since GNSS signals are time-sensitive, this requires the attacker's receivers to be tightly synchronized, adding to the attack's cost and complexity. To follow the minimum requirements of our attacker model, in our design, we extract raw navigation bits from the receiver and re-modulate the *unmodified* navigation message bits with appropriate code phase and Doppler shift, thus eliminating the need to separate the satellite signals physically. It is important to clarify our decision regarding the source of bit extraction. While commercial receivers like uBlox provide the necessary data, this process adds additional delay as we need to wait for the receiver to receive an entire sub-frame in the case of GPS or each nominal page in the case of Galileo. We solve this issue by leveraging GNSS-SDR [30], a popular open-source software-defined GNSS receiver. Specifically, we extract the navigation symbols directly from the tracking loop. This proves to be the fastest way to extract the navigation bits in a cost-efficient manner.

4.1 NAVMSG Streamer

In Section 3.3, we discussed the importance of detecting a satellite's navigation message within the time constraints set by the message authentication scheme, which is 1.5 sec or 3 min for GPS Chimera (based on the mode of operation) and 30 sec for Galileo. While commercial GPS receivers like uBlox provide access to raw navigation messages the wait time for each subframe is too long (6 s for GPS and 30 s for Galileo), making it unsuitable for our attack. In our work, we leveraged the design of GNSS-SDR and implemented the NAVMSG streamer as part of its telemetry decoder module. The telemetry decoder module provides access to raw navigation message symbols directly at the correlator output after the receiver detects the presence of a specific satellite. Additionally, we timestamp the navigation symbols and stream them to the spoofing signal synthesizer. Each NAVMSG streamer message includes the navigation message, our timestamp, and the corresponding satellite identification PRN. Furthermore, the NAVMSG streamer separates the satellite signals before streaming them to the spoofing signal synthesizer. Here, we note that prior works [20, 42, 63] have demonstrated the ability to separate satellite signals using directional antennas even if the pseudorandom codes are kept secret.

4.2 Delay Estimator and Signal Synthesizer

The required time delays for each satellite signal are estimated based on the location where the attacker records the legitimate signals, the target location to spoof, and the satellite's orbital status. The satellite orbital information is typically public knowledge or predicted based on previously decoded navigation messages. The method for calculating the required delays is described in Section 3.3. The spoofing signal synthesizer accepts a location, satellite ephemeris, and received raw navigation bits and generates the required spoofing signal. The architecture of the signal synthesizer is depicted in Figure 4. Based on the provided location and current

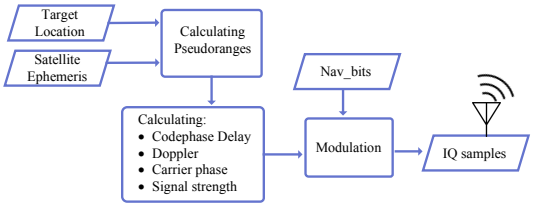

Figure 4: A flowgraph showing the implementation of the signal synthesizer. Desired pseudoranges and necessary signal parameters like carrier phase and code-phase delay are calculated using satellite ephemeris and the target location.

satellite positions, we first calculate the distances from all the visible satellites. Next, the calculated range is used to obtain the code phase delay and the carrier phase measurements. The necessary Doppler shift is calculated from the rate of change of pseudorange and the wavelength of the carrier frequency. These calculated parameters, along with the PRN code, are used to modulate the bits received from the NAVMSG streamer. It is important to note that the signal synthesizer must perform all these calculations periodically to account for the satellite's motion over time.

The spoofing signal synthesizer was configured to accept calculated delays at run-time, enabling complete control of the spoofing target location (Figure 4). The synthesizer receives the satellite signals from the NAVMSG streamer and selects a subset of the satellites (Figure 3). The estimated delays are applied to each satellite signal, combined, and transmitted to the victim receiver.

5 EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of our attack. First, we describe the evaluation setup and metrics used to measure our attack performance. Then, we provide an example spoofing scenario that we tested using the experimental setup for both GPS and Galileo signals. Finally, we discuss the results of our experiments.

5.1 Experiment Setup

We test our attack on both a commercial receiver (ublox M8N) and GNSS-SDR [30], an open-source software-defined GNSS receiver based on GNU Radio [4] capable of detecting, synchronizing, demodulating, and decoding the navigation messages originating from the constellations like GPS, Galileo, GLONASS, and BeiDou. It can process raw GNSS signals from a file source or from SDRs such as USRP [1] and enables us to gain deep insights into the attack performance. Figure 5 shows our setup that is capable of manipulating live GPS signals [5]. An active GNSS antenna feeds live GPS signals to GNSS-SDR that uses a USRP B210 as its RF front-end. A streamer client connects to GNSS-SDR and streams the decoded navigation bits. Based on the spoofed location, the signal synthesizer modulates the received bits as described in Section 4. It interfaces with another USRP B210 to transmit the generated signal which is then fed to a uBlox GPS receiver. In addition, to live signals, we use

[5]A video demonstration of this attack is available at https://youtu.be/ylTpEsTCczs

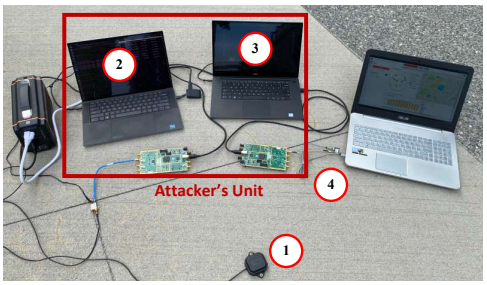

Figure 5: Experimental setup showcasing the real-time relay system: 1) Active GNSS antenna with a 5V bias-tee to capture live signals, 2) NAVMSG streamer, 3) signal synthesizer and 4) ublox M8N GNSS receiver.

a real-time GPS signal generator that generates a continuous IQ sample stream that is transmitted using another SDR.

5.2 Evaluation Scenarios

Based on the setup we described above, we show that our attacker can generate spoofing signals for a target location far away from the victim's (and the attacker)'s true location. We evaluated our attack for both static (stationary locations) and dynamic scenarios.
Static scenarios: To verify the feasibility of our attack on static scenarios, we picked spoofing locations at various distances away from the receiver's true location. We evaluated the accuracy of the proposed attack by measuring the offset between the spoofing location and the obtained location at the victim receiver. We were able to spoof the victim receivers (both ublox and GNSS-SDR) to our arbitrary locations, proving the attack's success. The results of our experiments are shown in Figure 6. Specifically, we continuously run real-time experiments for each location for 10 minutes to prove the stability of the obtained results, although the satellite's constellation keeps changing over time. We thoroughly examine the impact of satellite orbital motion in section 5.3. We once again emphasize that the contents of the navigation messages remained *unchanged* throughout the attack.
Dynamic motion scenario: We also evaluate the ability to generate spoofing signals that deceive the receiver into believing it is in motion at a specific location away based on the legitimate signals received at the attacker's location. To spoof such a motion, the attacker must manipulate the physical layer properties of the spoofed signal to reflect the updated position as per the desired trajectory. Our implementation calculates pseudoranges, rate of change of pseudoranges, carrier phase, code phase delay, and carrier frequency offsets to replicate the Doppler shifts for the specified position. To enable the dynamic motion, we use a sequence of latitude and longitude values that reflect the target path such that target speed = $distance(p_t, p_{t+1})/dt$ where p_t and p_{t+1} are sequential positions of the trajectory as a real-time input to the signal synthesizer. The signal synthesizer then calculates the required parameters and modulates the incoming bits to generate the necessary spoofing signal. To evaluate our strategy and its implementation, we generate and transmit a signal that forces the target into believing that it is moving at a speed of 1.98 m/s (7.13 km/h) along a pre-determined

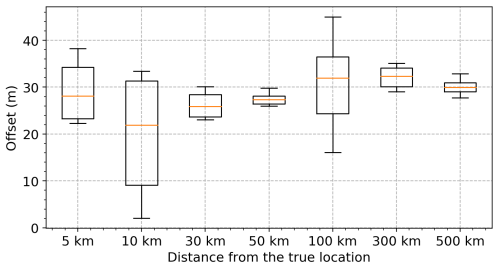

Figure 6: Spoofed location accuracy vs distance from the attacker's location. The offset is the distance between the spoofed and the victim receiver's estimated location.

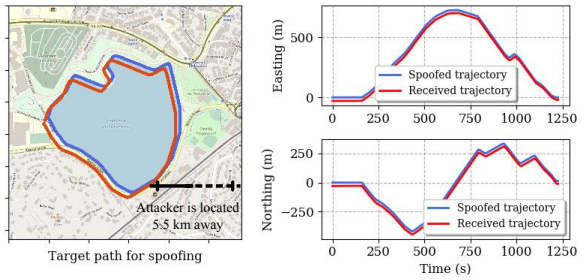

Figure 7: Spoofed trajectory vs victim receiver's estimates. The attacker generates signals corresponding to a 2.43 km trajectory almost 5.5 km away from the true location.

path which is 2.43 km and in an area that is 5.5 km away from the victim's original location. In this experiment, we only assume that the victim is within the radio range of the attacker, and the victim can be either stationary or in motion. It is worth noting that the attacker doesn't need to have prior knowledge of the victim's true location for the successful execution of the dynamic motion scenario. Similar to static location spoofing, without manipulating the legitimate navigation message contents, we spoof dynamic motion by introducing appropriate temporal and Doppler changes to the legitimate signals. Figure 7 shows a comparison of the spoofed trajectory and the received location estimates.

Proof-of-Concept Attack on Galileo Signals: For testing the attack's success against Galileo, we generated signals corresponding to a specific location using NCS TITAN GNSS simulator [44]. Then, we configured the attacker to generate spoofing signals for a location 100 km away. Knowing the satellite ephemeris data, we calculated the delays in advance for the spoofed location. The victim receivers were successfully spoofed to the desired location with an offset of ≈90 m. We used Septentrio [7], a commercial GNSS receiver capable of receiving and validating OSNMA messages, for further analysis. As we mentioned earlier, Galileo open service navigation message authentication (OSNMA) has started its test phase recently. But due to the limited availability of the OSNMA signals in space [45], we implemented our real-time setup based on GPS signals. Additionally, powerful GPS simulators provide the required flexibility to investigate the different aspects of the attack. Since our attack works by manipulating the time of arrival without modifying the navigation message, techniques like OSNMA, which rely on the integrity of navigation messages to verify signal authenticity are potentially vulnerable to our proposed attack.

Proof-of-Concept Attack on Augmentation Systems Aided Integrity Checks: Satellite-Based Augmentation Systems (SBAS) can provide additional data to GNSS users furthest location an attacker can to improve accuracy, reliability, and availability. The United States WAAS, the European EGNOS, and the Japanese QZSS are all examples of SBAS. SBAS data is generated based on measurements from a ground network and transmitted to geostationary satellites (GEO satellites). The SBAS GEO can provide information to the user's equipment via one of the GNSS receiver channels, as they use the same frequencies and signal structure as GNSS signals.

Commercial receivers, like ublox ZED-F9P [19], which we tested in our experiments, can use the SBAS messages to check the integrity of the received GPS data. When SBAS integrity checks are enabled, the navigation engine uses data from only those satellites whose integrity is verified by comparing it with SBAS data [19]. Thus, if a particular satellite doesn't pass this integrity check, it is ignored, and data from that satellite is not used in PVT calculation. If enough satellites fail the check, the receiver won't obtain a position fix. To test our attack against such integrity checks, we conducted an experiment where we first combine our synthesized signal with live SBAS signals. We feed the combined signal to a ublox ZED-F9P [19]. The receiver tracks SBAS signals as well as our signals and successfully obtains a 3-dimensional position fix with integrity verified. This confirms that the real-time signals we generate pass the applied integrity checks. Besides the integrity check advantage, augmentation systems provide corrections, such as ionospheric delay, that are used to improve the accuracy of the calculated location. The information provided by augmentation systems may end up improving the accuracy of the spoofed location.

5.3 Attack Performance Analysis

In this section, we assess and evaluate the factors that impact the performance of the attack. We evaluate accuracy as the difference in the location spoofed by the attacker and the location estimated by the victim receiver. Coverage is the attacker's furthest location to spoof the victim receiver *from where the legitimate signals were received*. The accuracy of the spoofed location, i.e., the difference between the spoofed target location and the location estimated by the victim receiver, depends on three main factors: i) attacker's sampling rate, ii) geometric dilution of precision (GDOP) of the spoofed satellites, and iii) satellite's orbital motion. We assume that the victim receiver and the adversary are in close proximity.

Impact of Attacker's Sampling Rate: The attack premise is to introduce specific delays to individual satellite signals by temporally shifting the raw signal (i.e., IQ samples) appropriately. Thus, the accuracy of the spoofed location directly relies on the attacker's ability to achieve precise sample delays, which is influenced by the sampling rate. As expected, we observe that the accuracy of the spoofed target location increases with the sampling rate (Figure 8). For instance, given a sampling rate of 4 MHz, delaying each sample

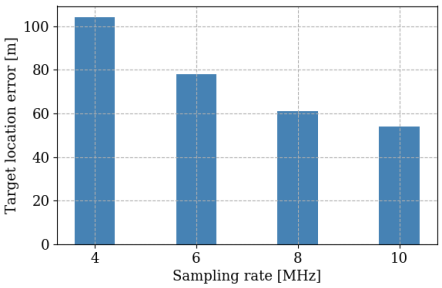

Figure 8: The accuracy of the spoofed location varies with the attacker's sampling rate.

is equal to 75 m change in pseudorange[6]. At 10 MHz sampling rate, the attacker can manipulate the pseudorange value of each satellite with a resolution of 30 m. Figure 8 shows the final offsets from the target location as estimated by the victim receiver.

Effect of GDOP: An important factor that directly affects the accuracy of the position estimates in any GNSS is the constellation of satellites' signals used to compute the location. The accuracy depends on the number of visible satellites and their elevations in the sky, i.e., the spread of satellites. The GDOP is low for a satellite constellation spread apart and high for a constellation with satellites clustered in a single direction. Figure 9 shows an example of constellations with good and poor GDOP values. The same principle applies to the attacker's spoofing signals. While choosing the satellite signals to manipulate temporally, it is essential to select satellite signals that have a low GDOP. To determine the best constellation for a receiver, we calculate the GDOP of several constellations and choose a satellite constellation with the appropriate GDOP. Our experiments confirm our hypothesis that selecting the correct subset of satellites provides better control to the attacker regarding spoofing positioning accuracy.

Impact of Satellite Orbital Motion: Recall that the pseudoranges are calculated based on the distance between the satellite and the receiver on the ground. The satellite orbits are configured to have a certain number of satellites visible to any part of the earth. For example, GPS has its satellites orbit the Earth along six orbital planes, and Galileo's satellites orbit the Earth along three orbital planes. As a result, the estimated pseudorange changes over time with a rate dependent on the location and time. Based on the satellite's velocity and position in the ephemeris data, the adversary can estimate the delay update rate and the required delay. Consequently, the attacker's choice of satellites directly impacts how often the adversary needs to update the delay calculation. Figure 10 shows the estimation location drift if the adversary does not recompute the delays, given a set of satellite signals spoofed. For example, for satellite sets 1 and 4, the drift in the spoofed location is less than 100 m even after 5 minutes. However, for satellite sets 2 and 3, the location drifts more than 500 m within the first minute. Spoofing a high GDOP satellite set leads to faster location drifts. Thus, the choice of satellites plays a critical role in the attack's performance.

[6]assuming the signal travels at the speed of light

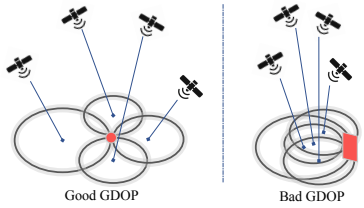

Figure 9: A spread-out satellite constellation has a lower dilution of precision than a clustered one.

Satellite Signal Coverage: In a typical GNSS spoofing attack, the attacker can spoof its target to any location on the earth as the attacker can generate navigation messages for any satellite. However, cryptographic signatures prevent attackers from generating navigation messages, limiting their spoofing capabilities. For example, prior work [37] showed the feasibility of spoofing the victim's location to where the signals were originally received. To verify our attacker's advantage, we evaluate the furthest distance an adversary can spoof a victim, assuming proximity between the attacker and the victim receiver. Given a set of satellites, the main factor that affects the ability of an attacker to spoof a specific location is the coverage of the set. In other words, an attacker can spoof a location as long as there are at least four common satellites visible at the attacker's location and the desired spoofed location. To analyze potential spoofing locations, we assume the attacker is located on the east coast of the United States and divide North America into hexagons with sides measuring 1100 km. Then, we select locations on the edges of the hexagons and identify the overlapping satellites between each of these locations and the attacker's location. Our results (Figure 11) indicate there are at least 7-8 common satellites across North America at any given time. Using the satellites visible at the attacker's location, we generate signals for our desired location in San Francisco and successfully spoof the receiver to a location $\approx$ 4000 km away from the attacker's location, demonstrating the wide coverage of the attack.

We highlight that an adversary can also use multiple receiver stations located strategically to receive the entire constellation and use it to spoof arbitrary locations on Earth. Our analysis showed that with two receivers positioned along the equator, we could observe at least four overlapping satellites at various corners of the world. This means that an attacker can generate spoofing signals to most of the earth's locations with two receivers carefully placed and connected to the spoofer positioned close to the victim receiver.

6 DISCUSSION

Capture Effect: When multiple signals following a similar signal structure arrive simultaneously, a typical wireless receiver automatically starts tracking and demodulating the stronger signal. In a spoofing attack, the attacker's signals are often strong enough to bury the legitimate signals under the noise floor, preventing the receiver from tracking them. If the target receiver is undergoing a cold start, the receiver tracks the attacker's signal. However, if the receiver is already tracking the legitimate signals, the overshadow attack causes the tracking loop to lose a lock, and the receiver restarts acquisition. This time, it acquires and tracks the stronger

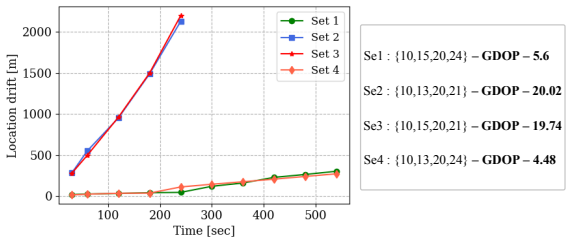

Figure 10: Impact of satellite constellation on spoofing accuracy indicating victim receiver's location drifting over time.

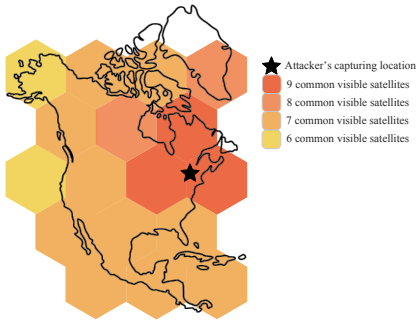

Figure 11: Number of shared satellites between each region and the set of satellites visible at the attacker's location.

signal, i.e., the attacker's signals. A GPS receiver will not experience this capture effect if mechanisms that detect an overshadow attack are present. Given the low transmission power of legitimate satellites, an attacker can overshadow the legitimate signals and force the receiver to experience the capture effect. Several works [34, 46, 55] have already demonstrated the feasibility of overshadowing GNSS signals. Overshadow attacks have also been shown on other wireless systems such as 4G LTE [36, 62]. More advanced attacks such as a seamless takeover of a target receiver that is already locked onto authentic satellite signals without the receiver noticing any disruption or loss of navigation data have also been demonstrated. Tippenhauer et al. [58] analyzed the requirements for a successful seamless takeover attack. The main challenge to execute a seamless takeover in our attack would be the need for strong synchronization between the victim and the adversary as temporally manipulating the messages inherently introduces delays. The feasibility of temporal shifting and relaying while performing a seamless takeover attack remains to be explored.

Time to Fix: Time to fix depends on several factors, with the victim's GPS receiver playing a significant role based on, e.g., the number of satellites it can process simultaneously, the validity/availability of ephemeris and almanac data, and so on. During our experiments, we compared the time to first fix in both legitimate (no spoofing) and spoofing scenarios. Our 10 runs of static spoofing showed no difference between the mean time to fix 32.39 s for a legitimate scenario and the mean time to fix 31.5 s for a spoofing scenario.

Loss of Lock: If the receiver is already locked to the legitimate satellite signals, the receiver will experience a loss of lock for a brief period of time. This is a common occurrence even in non-adversarial settings (hence hard to detect) such as tall buildings, tunnels, thick tree cover, etc. However, once the victim locked on to our spoofing signals, we did not experience any loss of lock even for a spoofing duration of more than 20 minutes (Figure 7).

Time Jump: The GPS receiver calculates the date and time using the TOW and Week# present in the navigation message. Since we do not manipulate the navigation message contents, the time jump would only be dependent on the attacker's processing time delay (4 ms in our case) and the delays introduced to spoof the new location (order of tens of ns). Ideally, an attacker could simply tap out the satellite signals directly in the acquisition phase in hardware which would reduce the delay to $\approx 1\ ms$ (correlation time). **Velocity Evaluation:** During our evaluation of the dynamic motion scenario, we tested our attack considering different spoofing velocities. In our implementation, the spoofing location in real-time needs to be provided by the attacker. As a result, the velocity does not impact the outcome. We tested different motion scenarios with variable velocities. An attacker can trivially control the spoofed velocity as a function of manipulated delays, which directly affect pseudorange calculation. The signal synthesizer is programmed to modify the physical properties of the spoofed signal, like the Doppler shift and the carrier phase, as a function of rate of change of the pseudorange. Through such a setup, the attacker can configure the velocity through strategic manipulation of spoofed location. Civilian receivers have hard-coded upper bounds in terms of velocity and altitude. Hence, through fine-grain control over spoofed velocity, we can carefully manipulate the velocity to avoid triggers.

7 COUNTERMEASURES

For complete spoofing resilience in satellite-based navigation systems, it is necessary to protect both the received signals' time of arrival (consequently the pseudorange computation) and the navigation message contents. Although GPS Chimera and Galileo OSNMA aim to protect the integrity of the navigation message data, they fail to prevent an attacker from manipulating the navigation messages' time of arrival and controlling the pseudoranges computed at the receiver, as shown in this paper. Our attack strategy allows for arbitrary location spoofing without modifying navigation messages, making even SBAS-aided integrity checks ineffective (Section 5.2). This eliminates the possibility of using navigation message contents to detect the attack. Adding selective delays and re-modulating extracted legitimate navigation bits introduces a collective delay providing an opportunity to detect spoofing by observing jumps in GNSS-derived time. For example, our setup adds a processing delay of ≈4 ms (Section 6). This would require receivers to have an external time reference source capable of detecting 4 ms clock jumps. Note that most systems directly rely on GNSS time as a reference and use it to generate timestamps. Prior work [47] shows that even NTP servers with the same time reference can have time differences of ≈200 μs. Such inaccuracies may impede GNSS-independent timekeeping and are susceptible to false positives and false negatives. Additionally, an attacker can leverage hardware-acceleration to reduce the processing delay, thereby further tightening the external clock accuracy needed for robust spoofing detection.

Our attack is unique in that it can spoof a victim receiver hundreds of kilometers away from its true location using the satellite signals received at the true location. This is in contrast to existing attacks that can only spoof the victim to the location where the signals were originally recorded. Techniques that leverage partially unknown spreading codes with hidden markers that are disclosed after a time delay [51], can limit the attack as the exact spreading code used by the legitimate signals is unknown to the attacker. This makes it difficult to separate and temporally manipulate individual satellite signals. However, the delayed disclosure mechanism requires the receiver to synchronize with the satellites. The frequency at which the hidden markers are renewed and revealed dictates the tightness of the synchronization. A high rate of change of hidden markers enforces tight synchronization requirements, and revealing the hidden markers after a long period will delay the integrity check at the receiver, making it unsuitable for many applications [32]. Attackers can also leverage codeless tracking techniques [23] that enumerate codes at run-time without waiting for the satellite to reveal the hidden markers or the unknown part of the spreading code. Directional antennas and spatial multiplexing techniques [59, 63] can also be used to separate satellite signals at the RF level. An attacker can then add temporal shifts to these individual satellite signals and re-transmit them without spreading and modulation. Many countermeasures [21, 40, 53, 54] aim to detect and mitigate spoofing attacks against unauthenticated GNSS signals. They mainly rely on identifying anomalies in the signal's physical layer characteristics and can be circumvented by carefully crafting the spoofing signal. Fundamentally, reliance on unidirectional broadcast communication makes them susceptible to signal spoofing attacks. This work further emphasizes the need to explore alternatives [16, 18] to satellite-based position, navigation, and timing estimation.

8 RELATED WORK

Cryptographic solutions may not always be sufficient to protect a system, especially for satellite navigation systems that rely on one-way communications. Past research has demonstrated the ineffectiveness of cryptographic solutions [26]. Our work is closest to Lenhart *et al.* [38], however with the following differences. In our attack, the attacker is not restricted to spoofing the location where it is located, and the spoofed dynamic trajectory is not limited to how the attacker's receiver moves. Our real-time code-phase manipulation technique can spoof any arbitrary locations and trajectories even 4000 km away without compromising the integrity of the navigation message content, and without physically moving from its location [7](Section 5). We also comprehensively studied all the parameters that affect the attack's performance allowing the attacker to choose the best configuration for the attack. For example, the attacker can select the optimal set of satellites based on their GDOP. We manipulated commercial receivers that utilize satellite augmentation systems for checking the integrity of the signals. Our attacker's main advantage is its 317.485 μs processing delay, which is critical when attacking NMA-enabled receivers since it has been assumed that they have a loose time synchronization. We also do not require mobile or high-speed networks as we generate spoofing signals directly from the live signals received close to the victim. In [26], the authors studied the theoretical aspects of meaconing attacks on GNSS signals. Spreading code estimation replay (SCER) attacks were introduced in [33], where a statistical estimate of the current bit at transmission time is determined based on the previously received samples. However, this attack is limited in terms of spoofable locations, and estimating parameters like chip length and power level can be complex. Follow up work [25] improved the effectiveness of SCER attack by optimizing system parameters and proposed a countermeasure based on the assumption that the victim knows the attacker's strategy. Other works [48, 65] discuss the threat landscape and basic countermeasures for cryptographically secure and unsecured GNSS signals. In [64], the authors demonstrated distance-decreasing attacks against GNSS authentication techniques. Works like [29, 57] discuss various replay attack approaches against authenticated GNSS signals.

In [57], the authors propose a delay control method that is capable of delaying signals by an integer multiple of the sampling period as well as a fractional multiple of the sampling period. They focused on how to add delays to individual signals. In [58], the authors referred to relative time offsets and studied the effect of spoofing in a specific scenario where the attacker has more than one transmission antenna, and he can send the spoofing signals using two or more omnidirectional antennas with some appropriate delays in order to covert satellite-lock takeover. In [24], the authors focused on studying the self-spoofing attacks on GNSS Signals with Message Authentication. In self-spoofing, the GNSS receiving equipment is under the control of the adversary. There has been limited work on detecting selective delay attacks. Most notably [31], where the authors demonstrate an approach based on machine learning to detect a SCER attack using a set of features extracted from the receiver search phase. Several other works like [10, 43, 53] describe the use of physical layer characteristics, multiple antennas, and crowd-sourced networks to provide spoofing attack detection.

9 CONCLUSION

In this work, we designed and developed an attack that allows spoofing a victim receiver's location or motion without modifying the legitimate signal's navigation message contents. Specifically, we demonstrated how an attacker can temporally manipulate legitimate satellite signals received at a victim's true location in real-time to generate signals that correspond to arbitrary locations and motions far away from the victim's actual position. This is in contrast to prior work that required an attacker to be present and record legitimate satellite signals at the location they intend to spoof the victim's receiver. We analyzed the effect of factors like sampling rate, satellite constellation, and orbits on the accuracy of the spoofed location and discussed the effectiveness of existing spoofing detection and mitigation techniques countermeasures against the proposed attack. We make our implementation publicly available[8] to the community for further research and development.

ACKNOWLEDGMENTS

The work was partially supported by NSF grant 2144914.

[7]The attacker is static throughout the attack, hence, the lazy attacker

[8]https://www.gnssrelayattack.com/

REFERENCES

[1] [n. d.]. *Ettus Research.* https://www.ettus.com/products/.
[2] [n. d.]. *Galileo.* https://galileognss.eu/.
[3] [n. d.]. *GLONASS.* https://www.glonass-iac.ru/en/about_glonass/.
[4] [n. d.]. *GNU Radio.* https://www.gnuradio.org/.
[5] [n. d.]. *GPS.* https://www.gps.gov/.
[6] [n. d.]. *LabSat GPS Simulator.* https://www.labsat.co.uk/.
[7] [n. d.]. *Septentrio.* https://www.septentrio.com/en/products/gnss-receivers/receivers-module/mosaic-go-clas-evaluation-kit.
[8] 2013. *UT Austin Researchers Successfully Spoof an $80 million Yacht at Sea.* https://news.utexas.edu/2013/07/29/ut-austin-researchers-successfully-spoof-an-80-million-yacht-at-sea/.
[9] 2015. *Software-Defined GPS Signal Simulator.* https://github.com/osqzss/gps-sdr-sim.
[10] 2018. Crowd-gps-sec: Leveraging crowdsourcing to detect and localize gps spoofing attacks, author=Jansen, Kai and Schäfer, Matthias and Moser, Daniel and Lenders, Vincent and Pöpper, Christina and Schmitt, Jens. In *IEEE Symposium on Security and Privacy (SP)*.
[11] 2019. *Ghost ships, crop circles, and soft gold: A GPS mystery in Shanghai.* https://www.technologyreview.com/s/614689/ghost-ships-crop-circles-and-soft-gold-a-gps-mystery-in-shanghai/.
[12] 2019. *How Hackers Can Take Over Your Car's GPS.* https://www.bloomberg.com/news/articles/2019-06-19/threat-of-gps-spoofing-for-autonomous-cars-seen-as-overblown.
[13] 2020. *Mysterious GPS signals reveal GHOST SHIPS sailing in wide circles off the coast of San Francisco, despite tracking data saying they should actually be thousands of miles away.* https://www.dailymail.co.uk/sciencetech/article-8400397/Mysterious-GPS-signals-reveal-GHOST-SHIPS-sailing-wide-circles-coast-San-Francisco.
[14] 2021. *Galileo OSNMA Information Note.* https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OSNMA_Info_Note.pdf.
[15] 2021. *Guidelines for Test Phase v1.0.* https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OSNMA_Receiver_Guidelines_for_Test_Phase_v1.0.pdf.
[16] 2022. *America Needs GPS Backup.* https://www.forbes.com/sites/dianafurchtgott-roth/2022/03/10/america-needs-gps-backup/.
[17] 2022. *M8N, Integration manual.* https://content.u-blox.com/sites/default/files/products/documents/u-blox8-M8_ReceiverDescrProtSpec_UBX-13003221.pdf.
[18] 2022. We Need a Backup for GPS . https://www.defenseone.com/ideas/2020/12/we-need-backup-gps-actually-we-need-several-them/170391/.
[19] 2022. *ZED-F9P, Integration manual.* https://content.u-blox.com/sites/default/files/ZED-F9P_IntegrationManual_UBX-18010802.pdf.
[20] Dennis Akos, Stephan Esterhuizen, Alexander Mitelman, R Eric Phelts, and Per Enge. 2004. High gain antenna measurements and signal characterization of the GPS satellites. In *Proceedings of the 17th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS)*.
[21] Dennis M Akos. 2012. Who's afraid of the spoofer? GPS/GNSS spoofing detection via automatic gain control (AGC). *NAVIGATION, Journal of the Institute of Navigation*.
[22] Jon M Anderson, Katherine L Carroll, Nathan P DeVilbiss, James T Gillis, Joanna C Hinks, Brady W O'Hanlon, Joseph J Rushanan, Logan Scott, and Renee A Yazdi. 2017. Chips-message robust authentication (Chimera) for GPS civilian signals. In *Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+)*.
[23] D Borio. 2011. Squaring and cross-correlation codeless tracking: analysis and generalisation. *IET radar, sonar & navigation*.
[24] Gianluca Caparra, Silvia Ceccato, Nicola Laurenti, and Justan Cramer. 2017. Feasibility and limitations of self-spoofing attacks on GNSS signals with message authentication. In *Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+)*.
[25] Gianluca Caparra, Nicola Laurenti, Rigas T Ioannides, and Massimo Crisci. 2014. Improving secure code estimate-replay attacks and their detection on gnss signals. *Proceedings of NAVITEC*.
[26] Maxandre Coulon, Alexandre Chabory, Axel Garcia-Pena, Jérémy Vezinet, Christophe Macabiau, Philippe Estival, Pierre Ladoux, and Benoit Roturier. 2020. Characterization of meaconing and its impact on GNSS receivers. In *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+)*.
[27] James T Curran and Cillian O'Driscoll. 2017. Message authentication as an anti-spoofing mechanism. *Working Paper*.
[28] I Fernández, V Rijmen, T Ashur, P Walker, G Seco, J Simón, C Sarto, D Burkey, and O Pozzobon. 2016. Galileo Navigation Message Authentication Specification for Signal-In-Space Testing–v1. 0. *European Commission*.
[29] Ignacio Fernández-Hernández and Gonzalo Seco-Granados. 2016. Galileo NMA signal unpredictability and anti-replay protection. In *International Conference on Localization and GNSS (ICL-GNSS)*.
[30] Carles Fernandez-Prades, Javier Arribas, Pau Closas, Carlos Aviles, and Luis Esteve. 2011. GNSS-SDR: An open source tool for researchers and developers. In *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS)*.
[31] Francisco Gallardo and Antonio Pérez Yuste. 2020. SCER Spoofing Attacks on the Galileo Open Service and Machine Learning Techniques for End-User Protection. *IEEE Access*.
[32] Joanna Hinks, James T Gillis, Perry Loveridge, Greg Myer, Joseph J Rushanan, Steve Stoyanov, et al. 2021. Signal and Data Authentication Experiments on NTS-3. In *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+)*.
[33] Todd E Humphreys. 2013. Detection strategy for cryptographic GNSS anti-spoofing. *IEEE Trans. Aerospace Electron. Systems*.
[34] Todd E Humphreys, Brent M Ledvina, Mark L Psiaki, Brady W O'Hanlon, Paul M Kintner, et al. 2008. Assessing the spoofing threat: Development of a portable GPS civilian spoofer. In *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS)*.
[35] JM. Juan Zornoza J. Sanz Subirana and M. Hernandez-Pajares. 2011. Galileo Navigation Message. https://gssc.esa.int/navipedia/index.php/Galileo_Navigation_Message.
[36] Martin Kotuliak, Simon Erni, Patrick Leu, Marc Roeschlin, and Srdjan Čapkun. 2022. {LTrack}: Stealthy tracking of mobile phones in {LTE}. In *31st USENIX Security Symposium*.
[37] Malte Lenhart, Marco Spanghero, and Panagiotis Papadimitratos. 2021. Relay/replay attacks on GNSS signals. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*.
[38] Malte Lenhart, Marco Spanghero, and Panos Papadimitratos. 2022. Distributed and Mobile Message Level Relaying/Replaying of GNSS Signals. In *International Technical Meeting of The Institute of Navigation*.
[39] Davide Margaria, Beatrice Motella, Marco Anghileri, Jean-Jacques Floch, Ignacio Fernandez-Hernandez, and Matteo Paonni. 2017. Signal structure-based authentication for civil GNSSs: Recent solutions and perspectives. *IEEE signal processing magazine*.
[40] Charles E McDowell. 2007. GPS spoofer and repeater mitigation system using digital spatial nulling. US Patent 7,250,903.
[41] Emily McMilin, David S De Lorenzo, Thomas Lee, Per Enge, et al. 2015. GPS anti-jam: A simple method of single antenna null-steering for aerial applications. In *Proceedings of the ION Pacific PNT Meeting*.
[42] J Merwe, Sascha M Bartl, Cillian O'Driscoll, Alexander Rügamer, Frank Förster, Philipp Berglez, Alexander Popugaev, and Wolfgang Felber. 2020. GNSS Sequence Extraction and Reuse for Navigation. In *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+)*.
[43] Paul Y Montgomery. 2011. Receiver-autonomous spoofing detection: Experimental results of a multi-antenna receiver defense against a portable civil GPS spoofer. In *Radionavigation Laboratory Conference Proceedings*.
[44] NCSTITAN [n. d.]. *NCS TITAN GNSS Simulator.* https://www.ifen.com/products/ncs-titan-gnss-simulator/.
[45] Mario Nicola, Beatrice Motella, Marco Pini, and Emanuela Falletti. 2022. Galileo OSNMA Public Observation Phase: Signal Testing and Validation. *IEEE Access*.
[46] Juhwan Noh, Yujin Kwon, Yunmok Son, Hocheol Shin, Dohyun Kim, Jaeyeong Choi, and Yongdae Kim. 2019. Tractor beam: Safe-hijacking of consumer drones with adaptive GPS spoofing. *ACM Transactions on Privacy and Security (TOPS)*.
[47] Andrew N Novick and Michael A Lombardi. 2017. A comparison of NTP servers connected to the same reference clock and the same network. In *Proceedings of the 48th Annual Precise Time and Time Interval Systems and Applications Meeting*.
[48] Panagiotis Papadimitratos and Aleksandar Jovanovic. 2008. Protection and fundamental vulnerability of GNSS. In *IEEE International Workshop on Satellite and Space Communications*.
[49] Adrian Perrig, Ran Canetti, J Doug Tygar, and Dawn Song. 2002. The TESLA broadcast authentication protocol. *Rsa Cryptobytes*.
[50] Marco Pini, Gianluca Falco, and Letizia Lo Presti. 2012. Estimation of satellite-user ranges through GNSS code phase measurements. *Global Navigation Satellite Systems: Signal, Theory and Applications*.
[51] Anna Poltronieri, Gianluca Caparra, and Nicola Laurenti. 2018. Analysis of the Chimera Time-Binding Scheme for Authenticating GPS L1C. In *ESA Workshop on Satellite NavigationTechnologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*.
[52] Mark L Psiaki and Todd E Humphreys. 2016. GNSS spoofing and detection. *Proc. IEEE*.
[53] 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*.
[54] Harshad Sathaye, Gerald LaMountain, Pau Closas, and Aanjhan Ranganathan. 2022. SemperFi: Anti-Spoofing GPS Receiver for UAVs. In *Network and Distributed Systems Security (NDSS) Symposium*.
[55] Harshad Sathaye, Martin Strohmeier, Vincent Lenders, and Aanjhan Ranganathan. 2022. An Experimental Study of {GPS} Spoofing and Takeover Attacks on {UAVs}. In *31st USENIX Security Symposium*.

[56] Logan Scott. 2003. Anti-spoofing & authenticated signal architectures for civil navigation systems. In *Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS)*.
[57] Shunshun Shang, Hong Li, Yimin Wei, and Mingquan Lu. 2020. A Flexible Replay Delay Control Method for GNSS Direct Meaconing Signal. In *Proceedings of the 2020 International Technical Meeting of The Institute of Navigation*.
[58] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. 2011. On the requirements for successful GPS spoofing attacks. In *Proceedings of the 18th ACM conference on Computer and communications security*.
[59] J Rossouw Van der Merwe, S Bartl, Cillian O'Driscoll, Alexander Rügamer, Frank Förster, Philipp Berglez, Alexander Popugaev, and Wolfgang Felber. 2020. GNSS Sequence Extraction and Reuse for Navigation. In *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+)*.
[60] Jon S Warner and Roger G Johnston. 2003. GPS spoofing countermeasures. *Homeland Security Journal*.
[61] Kyle Wesson, Mark Rothlisberger, and Todd Humphreys. 2012. Practical cryptographic civil GPS signal authentication. *NAVIGATION: Journal of the Institute of Navigation*.
[62] Hojoon Yang, Sangwook Bae, Mincheol Son, Hongil Kim, Song Min Kim, and Yongdae Kim. 2019. Hiding in Plain Signal: Physical Signal Overshadowing Attack on LTE.. In *USENIX Security Symposium*.
[63] Jiaqi Zhang, Xiaowei Cui, Hailong Xu, Sihao Zhao, and Mingquan Lu. 2018. Efficient signal separation method based on antenna arrays for GNSS meaconing. *Tsinghua Science and Technology*.
[64] Kewei Zhang and Panos Papadimitratos. 2019. On the effects of distance-decreasing attacks on cryptographically protected GNSS signals. In *International Technical Meeting of The Institute of Navigation*.
[65] Kewei Zhang and Panos Papadimitratos. 2019. Safeguarding nma enhanced galileo os signals from distance-decreasing attacks. In *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+)*.

A CONFIRMING THE INTEGRITY OF NAVIGATION MESSAGES

To validate our setup and to verify that the navigation bits stay untouched throughout our attack, we compare the bits received by the attacker and the bits received by the victim receiver using the cross-correlation function (Figure 12) for a single sub-frame (300 bits). It is important to note that the signal generation methods that we used do not affect the feasibility of the attack on signals with message authentication since the navigation message contents remain untouched by the attacker.

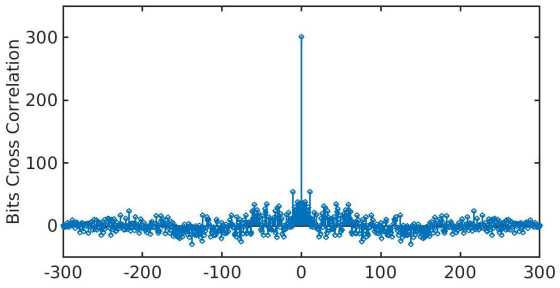

Figure 12: Correlation coefficient peak with a value of 300 showing that 300 bits of a single sub-frame received by the attacker and by the victim are the same.