SemperFi: Anti-spoofing GPS Receiver for UAVs

Harshad Sathaye, Gerald LaMountain, Pau Closas, Aanjhan Ranganathan
{sathaye.h, lamountain.g, pau.closas, aanjhan.ranganathan}@northeastern.edu
Northeastern University, Boston USA

***Abstract*—It is well-known that GPS is vulnerable to signal spoofing attacks. Although several spoofing detection techniques exist, they are incapable of mitigation and recovery from stealthy attackers. In this work, we present SemperFi, a single antenna GPS receiver capable of tracking legitimate GPS satellite signals and estimating the true location even against strong adversaries. Our design leverages a combination of the Extended Kalman Filter based GPS failsafe mechanism built into majority of UAVs and a custom designed legitimate signal retriever module to detect and autonomously recover from majority of spoofing attacks. We develop algorithms to carefully synthesize recovery signals and extend the successive interference cancellation technique to preserve the legitimate signal's ToA, while eliminating the attacker's signal. For strong adversaries capable of stealthy and seamless takeover, SemperFi uses brief maneuvers designed to exploit the short-term stability of inertial sensors and identify stealthy spoofing attacks. We implement SemperFi in GNSS-SDR, an open-source software-defined GNSS receiver, and evaluate its performance using UAV simulators, real drones, a variety of real-world GPS datasets, as well as on various embedded platforms. Our evaluation results indicate that in many scenarios, SemperFi can identify adversarial peaks by executing flight patterns less than 100 m long and recover the true location within 0.54 s (Jetson Xavier). We show that our receiver is secure against both naive and stealthy spoofers who exploit inertial sensor errors and execute seamless takeover attacks. Furthermore, we design SemperFi as a pluggable module capable of generating a spoofer-free GPS signal for processing on any commercial-off-the-shelf GPS receiver available today. Finally, we release our implementation to the community for usage and further research.**

I. Introduction

Today, Global Positioning System (GPS) is critical to a wide variety of safety and security-critical applications. The use of GPS is so ubiquitous that it plays an enabling role in 14 out of 16 industries classified as critical infrastructure [14] by the US Department of Homeland Security. Due to the lack of authentication in civilian navigation messages, GPS is vulnerable to signal spoofing attacks. In a GPS signal spoofing attack, the attacker transmits specially crafted signals that imitate satellite signals with power high enough to overshadow the legitimate signals [21]. Several researchers have shown that it is possible to modify the course of ships [16], unmanned aerial vehicles [69], and self-driving cars [20] by simply spoofing GPS signals. There is also an increase in GPS spoofing incidents [18] reported from around the world. For example, there are reports of thousands of ships in Shanghai falling victim to GPS spoofing [19]. There are also reports [18] of state actors using GPS spoofing and jamming in several countries to disrupt everyday affairs. With the widespread availability of software-defined radio and public GPS signal generator repositories [17], it is now possible to spoof GPS signals with less than $100 of hardware equipment.

Proposed countermeasures are either cryptographic solutions or leverage physical-layer signal properties. Countermeasures that use some form of cryptographic authentication [29], [46], [48], [77] prevent attackers from generating arbitrary false GPS signals. The recently launched Galileo's *Open Service Navigation Message Authentication* [34] service is based on the TESLA protocol and one-way hash functions that provide navigation message authentication service. However, they do not protect against attackers capable of recording and replaying legitimate GPS signals. The receiver's location and time are estimated using the GPS signal's time-of-arrival and not just the navigation message content. Other countermeasures that do not require cryptographic authentication rely on detecting anomalies in the received GPS signal's physical characteristics, such as received signal strength [75], noise levels, direction or angle of arrival [53], and other data that are readily available as receiver observables on many COTS GPS receivers. Some countermeasures [65] exploit the difficulty in canceling out legitimate GPS signals completely to detect stealthy, seamless takeover attackers. A few countermeasures propose the use of additional sensors [42] and receivers [54], [73] to detect spoofing attacks. The majority of the above schemes only detect a GPS spoofing attack, i.e., raise the alarm in case of a spoofing attack and often require manual intervention. Moreover, existing spoofing mitigation techniques are ineffective against strong adversaries capable of completely overshadowing legitimate signals and stealthy attackers, e.g., seamless takeover [73] of victim's GPS location without any signal disruption, despite having redundant fail-safe sensors [56]. In summary, today's GPS receivers, specifically those implemented on UAVs, are incapable of uninterrupted operation during a spoofing attack.

In this work, we present SemperFi, a single-antenna GPS receiver for UAVs that autonomously recovers and continues to output legitimate location even against strong adversaries capable of stealthy and seamless takeover. SemperFi is comprised of two main building blocks: i) Adversarial Peak Identifier (API), and ii) Legitimate Signal Retriever (LSR). The API is responsible for detecting a spoofing attack *and* distinguishing the attacker's signal from the legitimate GPS signals. Based on the information provided by the API, the LSR synthesizes an appropriate recovery signal that eliminates the spoofing signal using a successive interference cancellation (SIC) technique. With SemperFi, we make the following contributions:

- We design a spoofing mitigation technique that leverages

Network and Distributed Systems Security (NDSS) Symposium 2022
27 February - 3 March 2022, San Diego, CA, USA
ISBN 1-891562-74-6
https://dx.doi.org/10.14722/ndss.2022.23071
www.ndss-symposium.org

inertial sensors and Extended Kalman Filter (EKF) methodology that is commonly part of majority of UAV's built-in GPS fail-safe mechanisms [50], and integrate this technique with SemperFi's adversarial peak identifier module. In combination with SemperFi's legitimate signal retriever, we show that the receiver can detect majority of GPS spoofing attacks present in literature and autonomously recover its true location.

- We introduce active spoofing verification that forces the UAV to execute a maneuver in a scenario where an adversary is capable of gradually introducing location offsets [80]. This is done without triggering any detection mechanisms.We rely on the auxiliary peak tracking technique [65] that has been shown to be highly effective against stealthy seamless takeover adversaries to initiate the maneuver.
- We model the maneuvers as a series of velocity vectors subject to time-varying acceleration with the goal of minimizing the time-to-trigger of the GPS failsafe in case of an attack. As a result, SemperFi overcomes prior works limitations of being unable to detect or recover from a stealthy adversary capable of seamless takeover attack. Prior spoofing detection techniques based on inertial sensors have been shown to be vulnerable to adversaries deviating the UAV's path at a rate that is within the EKF estimation bounds. Techniques that can detect such attacks were unable to distinguish the spoofing signal from legitimate ones–a key requirement for autonomous recovery.
- Traditional wireless communication systems have successfully applied SIC to recover message contents. However, in the case of GPS it is essential to preserve the ToA of the satellite signal itself in addition to the data contained within the navigation messages. To address this unique challenge present in eliminating GPS spoofing signals, we develop algorithms to estimate the various physical characteristics such as amplitude, phase, and ToA of both the legitimate and spoofing signals without significant changes to the receiver's signal flow and overall architecture.
- We implement SemperFi using GNSS-SDR [35] and evaluate its performance against both synthetically generated as well as real-world GPS signals using consumer drones like DJI Flamewheel F450 [5] and Holybro S500 [10]. We also evaluate the performance of SemperFi on various embedded systems commonly used as UAV flight controllers. Furthermore, we evaluate the effectiveness of SemperFi against TEXBAT [41], a public dataset of GPS spoofing traces. Our evaluation shows that, in the majority of scenarios, SemperFi can recover from an attack by executing flight patterns less than 100 m in length, recovering the true location within 0.54 s with an accuracy of less than 20 m(in majority of the cases identification maneuver is not required) on popular embedded platforms such as Jetson Nano and Xavier
- We designed SemperFi as a pluggable module that outputs spoofer-free GPS signals identical to legitimate satellite signals. Therefore, SemperFi allows an unmodified COTS GPS receiver to process and generate location and time estimates without any disruption.
- Finally, we open source[1] our design, implementation, and evaluation datasets to the community for further research.

[1] https://semperfi-gps.com/

II. GPS and Spoofing Attacks

A. GPS Overview

Global Positioning System (GPS) is the most widely used Global Navigation Satellite System (GNSS) that uses the L1 frequency band. GPS consists of 31 operational satellites at an altitude of approximately 20,220 km[2]. Each satellite continuously transmits navigation messages containing timing information, satellites' ephemeris data, and other necessary information that enables the receiver on the ground to localize itself. The navigation messages are spread using a coarse-acquisition (C/A) code unique for each satellite and transmitted using a 1575.42 MHz carrier. The C/A code is public and contains 1023 bits (also referred to as *chips*) repeated every 1 ms. Military GPS signals use a longer and a secret spreading code. This paper focuses on civilian GPS signals as they are widely used even in security-critical applications [38], [71]. The navigation data comprises of five subframes. Each subframe contains 1500 bits transmitted at 50 bps [25]. These subframes contain satellite clock and orbital information. The ephemeris data is updated every 2 hrs and is valid for 4 hrs [30].

A typical GPS receiver consists of four main components, i) RF front end, ii) Acquisition module, iii) Tracking module, and iv) Position Velocity Time (PVT) module.

RF front-end receives raw RF signals and converts the raw signal to an intermediate frequency for efficient processing. Each satellite is assigned a "channel". This channel is analogous to a hardware pipeline for processing a single satellite.

Acquisition module performs a two-dimensional search for visible satellites in the received signal by correlating the received signal with a locally generated replica of each satellite's C/A code. The two-dimensional search is a time- and frequency-domain search to account for code phase delays and Doppler shifts that arise because of the satellite's and the receiver's motion. If the code and Doppler searches result in a correlation peak above a certain threshold, the receiver then switches to tracking and demodulating the navigation message.

Tracking module is responsible for tracking the code phase and the Doppler shift provided by the acquisition module. It also demodulates the navigation messages and passes them on to the PVT module.

Position Velocity Time Estimation (PVT) module decodes raw navigation bits and calculates the pseudorange between the satellite and the receiver. A receiver requires information from at least four satellites for accurately calculating position, velocity, and time. The PVT module is the last block of the GPS receiver and implements algorithms to compute navigation solutions and delivers information in appropriate formats (e.g., RINEX, UBX, NMEA [8]) for further processing.

B. Attacker goals and assumptions

In a GPS spoofing attack, an adversary transmits specially-crafted radio signals identical to authentic GPS satellite signals. The spoofing signals are generated for an attacker-defined trajectory or a static position and transmitted typically using

[2] As of January 2021. [13]

a software-defined radio hardware platform. All the necessary information for generating GPS signals like modulation schemes, message formats, and spreading codes is publicly available. The goal of an attacker can be to i) force the user to calculate a wrong geographic location, ii) forge timing information, or iii) execute a denial of service attack by causing interference. During a spoofing attack, the GPS receiver locks onto the stronger signal i.e., the attacker's signals, ignoring the weaker legitimate satellite signals. This results in the receiver computing a false position, velocity, and time-based on the spoofing signals. Note that the received GPS signal power on the ground is typically around -127.5 dBm and, therefore, trivial for an attacker to overshadow the legitimate signal with the spoofing signal.

In this work, we focus on an attacker that forces the user to calculate a wrong geographic location. We do not consider an attacker whose goal is to cause a denial of service attack by transmitting jamming signals. An attacker can manipulate the calculated PVT solution as follows: i) manipulate ToA of messages and/or ii) manipulate navigation message contents (e.g., satellite location, transmission time). We base the attacker model on work done in [73] and drone hijacking strategies proposed in [60]. We assume the following about the attacker. The attacker can be equipped with an omni-directional or a directional antenna and is allowed to spoof any number of satellites. Our threat model includes attackers with little know-how as well as sophisticated seamless-takeover attackers [73]. Attackers with little know-how typically use GPS signal generators (both hardware [12] or software [17]) to execute the spoofing attack due to their low complexity, portability and ease of use. Such an attack will result in sudden loss-of-lock and abrupt jumps in the location estimates. In contrast, during a seamless takeover attack, the receiver does not undergo abrupt loss of signal reception or lock. The attacker keeps the navigation message content identical to the legitimate GPS signals and gradually increases the power of the spoofing signals while carefully introducing offsets in the code phase delays or modify navigation message contents; thereby affecting the pseudorange calculations. The requirements to execute such a seamless takeover attack has been explored in [73]. We do not restrict the position of the attacker and assume that the attacker is well-aware of SemperFi. Furthermore we assume that the attacker has access to the trajectory of the drone and the attacker can track the drone in real-time. Factors that can affect attacker's success are explained further in Section V. We also assume that the attacker has neither compromised the onboard sensors and that these sensors provide valid, unadulterated data, nor the attacker has access to the inertial sensor measurements.

In this paper, we show that our proposed GPS receiver, SemperFi, can counteract all the types of attackers mentioned above. In our analyses, we give special focus to stealthy seamless takeover attacks and, in general, attacks that are hard to not only detect but pose challenges to realizing a fully-autonomous GPS receiver capable of uninterrupted true location estimates even in an adversarial setting.

III. Design of SemperFi

SemperFi is a single-antenna GPS receiver capable of providing uninterrupted location estimates even when subjected to a GPS spoofing attack. In this section, we present the design of SemperFi and the challenges that follow.

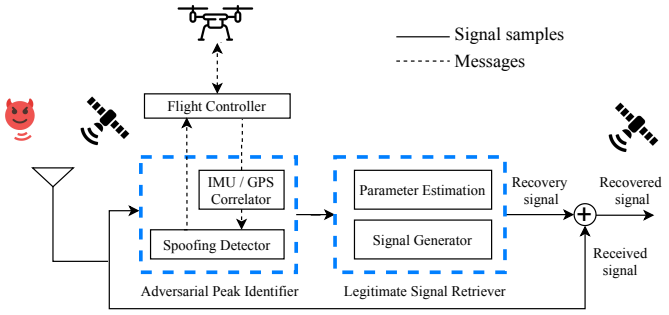


Figure 1: High-level overview depicting essential components of SemperFi.

A. Challenges

For the GPS receiver to operate autonomously in an adversarial setting, the receiver must continuously perform the following actions. First, it is necessary to detect an ongoing spoofing attack reliably. Then, the receiver must be capable of identifying or distinguishing between spoofing signal and the legitimate signal. Finally, after identifying the spoofing signal, the receiver has to eliminate or reduce the spoofing signal's effect on the final estimated location. To the best of our knowledge, there is no receiver design so far in prior work that addresses all the above three challenges. Unlike typical wireless communication systems where it is sufficient to recover the signals' data, GPS receivers require both the signal's data and its ToA. Moreover, GPS receivers are not tolerant to received sample losses. Continuous tracking of the satellite signals is necessary to estimate code and carrier phase delays that directly affect the PVT estimation. Finally, in the case of a spoofing attack that injects fake dynamic motion pattern (e.g., diverting the course of a ship or force a drone to deviate from its flight path), the attacker dynamically manipulates ToA of the spoofing signal and the data contained within the navigation messages. Therefore, traditional interference cancellation and mitigation techniques need to be modified or extended in order to handle this kind of attack.

B. High-level Overview

SemperFi provides fully-autonomous spoofing resistance through the combined effort of two modules: i) the Adversarial Peak Identifier (API), and the ii) Legitimate Signal Retriever (LSR). API is responsible for detecting and identifying the adversarial signals and LSR is responsible for signal recovery. A block diagram of SemperFi's various components is shown in Figure 1.

API relies on a widely adopted extended Kalman filter (EKF) based sensor fusion algorithm for UAVs and the spoofing detection methodology based on prior work [65] that demonstrated the ability to detect even a strong, seamless takeover attack [73] for providing reliable spoofing detection and signal identification. In most cases, SemperFi can detect spoofing and identify the adversarial signals by monitoring the position and velocity variance in the innovations from

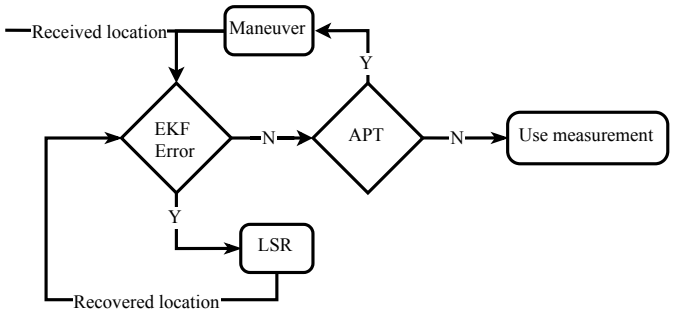


Figure 2: Flowchart depicting the operations of SemperFi and crucial decisions it makes to execute the recovery process.

the EKF algorithm. The receiver can consider the currently tracked peaks as adversarial if the EKF error is triggered, because, position *and* velocity variance error will be raised only for inconsistent or spoofed GPS locations (more details in Section V). However, it is shown in prior works [45], [80] that it is possible to stealthily spoof GPS coordinates without alerting the EKF algorithm. Such a sophisticated attacker, that is capable of performing a seamless takeover attack of GPS as well as EKF will be detected by the auxiliary peak tracking technique [65]. Even though such a test can confirm the presence of an adversary, it is unable to identify the adversarial signal. To distinguish between the legitimate and adversary signals, we introduce an *active spoofing verification* component, where we instruct the UAV to perform a maneuver unknown to the attacker. This maneuver introduces a perturbation in the drone's movement. Following the perturbation, the flight controller monitors the position and velocity variance and determines if the receiver is tracking an adversarial signal i.e., if the maneuver causes the variance to rise above a pre-determined threshold, then SemperFi determines the tracked signal to be adversarial. It is important to note that the UAV needs to perform the maneuver only when SemperFi is unable to identify the spoofing signal. Figure 2 shows the flow of operations for SemperFi. For example, consider a naive position push spoofer where the attacker adds offsets to the GPS position calculated by the UAV. The EKF variance check will be triggered as the GPS position jump will not match the accelerometer values and SemperFi considers the currently tracked peak as adversarial and proceeds to eliminate it. Since EKF was successful in detecting and identifying the adversarial signal, auxiliary peak detection and the identification maneuver is not required.

Once API identifies the signal, it sends an "adversarial/non-adversarial" message to the LSR. An "adversarial" message means that the currently tracked peaks are adversarial. Then, the LSR generates a replica of the adversarial signal and performs SIC to recover the legitimate signal.

SIC [63] is a well-known technique used for canceling out interference caused by stronger signals. The noiseless GPS signal from a single satellite can be modelled as

$$S_R = a[k]\tilde{s}_T[k-\tau(k)]e^{j2\pi f_D[k]T_s k+\phi[k]} \quad (1)$$

where $s_T(k)$ is the baseband signal (k number of samples per C/A code), and $a[k], \tau(k), f_D[k], \phi[k]$ are the amplitude, time-varying code delay, Doppler shift and carrier phase shift respectively. In presence of an adversary, received signal is

$$S_R = S_L + S_{AT} \quad (2)$$

where S_L is the legitimate signal and S_{AT} is the attacker's signal. In a GPS spoofing attack, the attacker overpowers the legitimate signal. Thus, $a_{AT} > a_L$ and as result, the GPS receiver tracks S_{AT}. The LSR module uses the spoofing detector's tracking parameters τ_{AT}, Doppler shift f_{AT}, to track the adversary signal for a specific duration and extracts the baseband data s_{AT}. The amplitude a_{AT} and carrier phase shift ϕ_{AT} of the adversarial signal are then estimated and used in combination with the baseband data to generate the recovery signal $S^{'}_{AT}$, a close replica of the estimated adversary signal. Using the above information, S_L can be obtained as follows:

$$S_L = S_R - S^{'}_{AT} \quad (3)$$

The replica is fed back to perform SIC and undergoes re-acquisition. If necessary, SemperFi repeats this process until the spoofing detector does not raise an alarm. At this stage, the spoofing signal is eliminated or significantly attenuated, and therefore the receiver starts tracking the legitimate signals. There are scenarios where, despite a successful recovery, either due to the spoofing signal's strength or synchronization concerning the legitimate signals, the navigation message content and arrival time are hard to decode and introduce ambiguities in the PVT estimates. We developed a pseudorange rectifier for such specific scenarios that can recover from an attack with decreased accuracy. Finally, we designed SemperFi as a plugin module that can be configured to act as a spoofing signal filter, where the filtered signal is fed directly to any commercial GPS receiver for PVT estimation. This prevents significant hardware design changes to existing deployments.

C. Adversarial Peak Identifier (API)

The API leverages the measurements obtained from the EKF implemented in the UAV. Precise estimation of the time-varying position of a vehicle is required for the purposes of autonomous navigation and control. Kalman filtering has been the gold standard for performing dynamic state estimation. The Kalman filter in all its forms (e.g. linear Kalman filter, extended Kalman filter, etc.) operates by performing a repeating sequence of prediction, observation, and correction according to a set of equations based on a hidden Markov model (HMM). In this way, the algorithm provides statistically optimal estimates of the unknowns required by the vehicle controller for accurate navigation. The UAV observes its own position and velocity through the use of on-board sensors. The EKF algorithm implemented in the UAV monitors the position and the velocity variance and triggers a failsafe if the variances exceed a pre-determined threshold. As mentioned earlier, only inconsistent or faulty GPS measurements can trigger this failsafe.

The presence of a valid satellite signal is determined by a peak that forms as a result of the correlation operation performed by the acquisition module. Malicious signals result in additional correlation peaks which may be misidentified as legitimate GPS signals. The adversarial peak identifier (API) is responsible for identifying such malicious peaks. Similar to any wireless receiver, the GPS receiver also experiences

capture effect [26] and by default locks on to the strongest signal and tracks it. Thus, even in scenarios where the receiver receives both adversarial and legitimate signals, it calculates the stronger GPS signals' PVT solution. SemperFi then attempts to attenuate adversarial signals to recover from the spoofing attack. This is not, however, a simple matter of attenuating the signal producing the strongest peak. An attacker aware of this strategy can transmit signals with a power lower than the received signal in specific attack scenarios. Even though the attacker's signal is weaker, it will still be visible in the acquisition plot. As a result, the spoofing detector will raise a spoofing alarm. If the stronger peak is assumed to be the adversarial peak, SemperFi will eliminate the legitimate peak as the legitimate signal is stronger than the adversarial signal. Therefore, for SemperFi to successfully attenuate adversarial signals and recover the location, it is essential to ensure that the peak currently being tracked is the adversarial signal and account for the above described scenarios. As described in Section III-B, for majority of spoofing attacks, EKF will raise an error and identify the signal to eliminate. However, it is possible for an attacker to spoof GPS signals without raising this error. In such a scenario, presence of multiple peaks is an indication of a spoofing attack. To verify whether the currently tracked signal is an adversarial signal, SemperFi performs a controlled maneuver. This maneuver will be performed only in a scenario where SemperFi is unable to correctly identify the adversarial signal. SemperFi sends a series of velocity vectors with varying acceleration that are independent of GPS measurements. If SemperFi was tracking the spoofing signal, the unplanned controlled maneuver results in inconsistencies in the GPS measurements as the attacker will be unaware of this maneuver; thereby triggering the EKF error.

D. Legitimate Signal Retriever (LSR)

LSR is responsible for generating the corresponding replica signal i.e., the recovery signal for every spoofed satellite. LSR requires; i) Amplitude, ii) code phase delay, iii) Doppler shift, iv) carrier phase, and v) navigation bit of the attacker's signal for generating the recovery signal. LSR obtains the code phase delay and the Doppler shift from the acquisition module. The replica signal is aligned with the received spoofing signal in the time domain using the code phase delay and the frequency domain using the Doppler shift. The LSR consists of a minimal tracking module that extracts the navigation bits and the carrier phase information of the adversarial spoofing signal. Each of the required components except the signal amplitude is readily available through the basic acquisition and tracking components in any standard receiver architecture. We devised an amplitude estimation technique that relies on the correlation coefficient of the attacker's peak.

Amplitude Estimation: The amplitude of the acquired signal can be estimated from the magnitude of the corresponding peak in the two-dimensional function of code phase delay and the Doppler shift called the cross-ambiguity function (CAF). Recall that the input to the acquisition block is a set of K observations of a modulated GNSS signal. The sampled baseband signal can be modeled as

$$x_{IN}[k] = a[k]\tilde{s}_T[k-\tau_k]e^{j(2\pi f_D[k]kT_s+\phi[k])} + n(t) \quad (4)$$

where $a[k]$ is the sampled signal amplitude, $\tilde{s}_T[k]$ is a filtered and sampled version of the complex baseband GNSS signal, $f_D[k]$ is the time-varying Doppler shift, $\phi[k]$ is the time-varying phase shift, and $n[k]$ is additive noise. Computation of the correlations which comprise the sampled CAF in the acquisition block is typically done in the Fourier domain after carrier wipe-off.

$$x[k] = x_{IN}[k] \cdot e^{-j2\pi \check{f}_D kT_s} \quad (5)$$

At the peak of the CAF, the parameters $\check{f}_D[k], \check{\tau}[k], \check{\phi}[k]$ correspond to the maximum likelihood estimate of the "true" parameter values, and the discrete Fourier domain representation of the signal after wipe-off simplifies to

$$X[k] = \text{FFT}_K\{x[k]\} = a[k] * S[k]W_K^{\tau} \quad (6)$$

where $*$ is the convolution operator, $S[k]$ is the discrete Fourier transform of $\tilde{s}_T[k]$ and W_K^{τ} is the term associated with the signal delay. Applying the FFT of the local code replica $D[k]$ is performed by multiplication in Fourier domain

$$Y[k] = X[k] \cdot D[k] = a[k] * S[t]D[k]W_K^{\tau} \quad (7)$$

The final step in computing the CAF is taking the inverse FFT

$$R_{xd}(f_D, \tau) = \text{IFFT}_K\{Y[k]\} = a[k] \sum_{n=0}^{K-1} s[n]d[k-n] \quad (8)$$

The "peak metric" for a given local replica is found by maximizing the squared magnitude of the correlation grid. At the peak where the signal component $s[k]$ and the local replica are identical, this ideally reduces to

$$S_{\text{MAX}} = |R_{xd}(f_D, \tau)|^2\big|_{f_D \approx \check{f}_D, \tau \approx \check{\tau}} = |a|^2|K|^2 \quad (9)$$

where S_{MAX} is the maximum peak and $R_{xd}(f_D, \tau)$ is the search grid. Rearranging this, we find an expression for the amplitude of the input signal in terms of the peak metric

$$|a| = \frac{\sqrt{S_{\text{MAX}}}}{K} \quad (10)$$

Equipped with all the above information, the recovery signal is generated. LSR performs this iterative cancellation process for all the satellites.

Pseudorange Rectifier: Specific attack scenarios, such as adversary introducing extreme interference or the spoofing signal's code phase and doppler are in close proximity to the legitimate signal can result in the navigation bits of the legitimate signal getting corrupted. In such a scenario, even if SemperFi can recover the legitimate peak, it won't be able to successfully track and decode navigation bits, leading to incorrect calculation of true location. In SemperFi, we design the pseudorange rectifier module to correct these ambiguities and aid in the recovery of the true location. Use of pseudorange rectifier() is optional. It is designed to be used in a very specific scenario where it is not possible to track the legitimate peaks and the attacker manipulates the location by changing the ToA of the signals without changing the navigation messages, i.e., legitimate and adversarial messages are the same.

Commercial GPS receivers use a common reception time technique [66] to calculate pseudorange to the satellite, an essential component in PVT calculation. In this technique, a common reception time, which is usually 65-85 ms [66], is set across all the channels as the propagation time of the closest

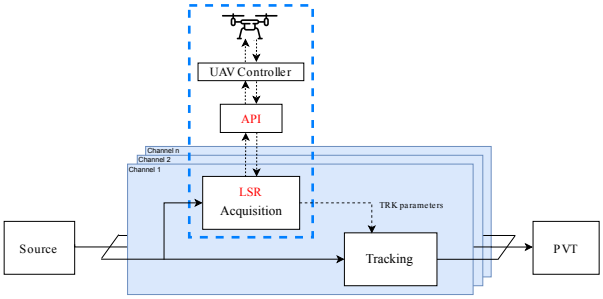


Figure 3: A schematic showing the implementation of SemperFi. The LSR is implemented as part of GNSS-SDR and API is implemented as part of the UAV flight controller.

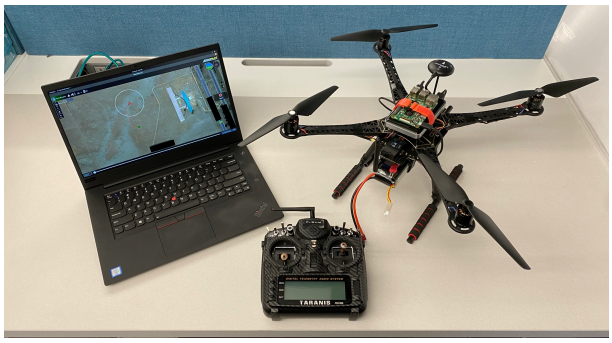


Figure 4: Hardware setup showcasing the Holybro S500 drone, the radio controller, and the ground control station.

satellite's signal. The receiver calculates the propagation time of signals from other satellites relative to this reference. Modern GPS receivers maintain a sample counter for accurate time measurement. According to the common reception time technique, pseudorange is calculated as follows:

$$P^i = c(t_{ref} + t_{rx} + \tau^i) \tag{11}$$

where P^i is the pseudorange measurement for i^{th} satellite, c is the speed of light, t_{ref} is the initial reference time (usually 65-85 ms [66]), t_{rx} is the receiver time maintained by a sample counter, and τ^i is the code phase delay of i^{th} satellite.

SemperFi attenuates the adversarial peak and obtains tracking parameters of the legitimate peak. However, it doesn't track the legitimate peak. Instead, it starts tracking the adversarial peak and obtains adversarial navigation messages. A stealthy attacker will keep navigation messages the same and change only the signals' ToA. It offsets the sample counters by $\tau^i_{at} - \tau^i_l$ where τ^i_{at} is the code phase delay of i^{th} satellite of the attacker and τ^i_l is the code phase delay of i^{th} legitimate satellite obtained during the peak recovery.

$$P^i_l = c(t_{ref} + t_{rx} + \tau^i_{at} - \Delta\tau^i) \tag{12}$$

$$\Delta\tau^i = \tau^i_{at} - \tau^i_l \tag{13}$$

Substituting (13) in (12) we get (11). In this way, SemperFi can obtain legitimate pseudoranges (P^i_l) by rectifying ToA of adversarial signals.

SemperFi is designed to protect against sophisticated seamless-takeover attacks as well as naive hard-spoofing attacks. In hard-spoofing attacks, the adversarial signals are not synchronized with the legitimate satellite signals and may contain different navigation messages. The attacker transmits with excessive power, and as a result, the receiver experiences a sudden loss of lock. A typical receiver is configured to restart the acquisition process if there is a loss of lock. Restarting the acquisition process triggers SemperFi. If spoofing is detected, SemperFi will initiate the recovery process as mentioned.

IV. Implementation

The two sub-systems which make up SemperFi are implemented independent of one another: the API is implemented at the flight controller level while the LSR along with the spoofing detector is implemented in GNSS-SDR as part of the acquisition block. These two components interact with each other over a TCP socket. We implemented the LSR module of SemperFi in GNSS-SDR [35] an open-source software-defined GNSS receiver written in C++. We implemented the API module using consumer drones. Refer to Figure 3 for a schematic of the implementation. GNSS-SDR follows GNU-Radio architecture and supports the processing of pre-recorded signals from a file source and software-defined RF-frontends like a USRP [4]. GNSS-SDR follows a hardware receiver's design as described in Section II, except all the components are implemented in software. Signals from individual satellites are processed by individual *channels*. Each channel is like a hardware pipeline of various GPS signal processing blocks, including acquisition, tracking, and PVT calculation. At runtime, the GNSS-SDR builds the receiver using these blocks based on specifications from a user-defined configuration file. This allows loosely coupled operations. In our implementation and evaluation, we use software-defined radio hardware platforms manufactured by Ettus Research [4], specifically, USRP B210 and N210 with SBX-40 daughterboard, for recording and providing raw data.

A. Adversarial Peak Identifier (API)

In SemperFi, API is implemented as an independent module that interacts with the LSR. This was implemented on an unmanned aerial vehicle in a simulated environment as well as on a DJI Flamewheel F450 and a Holybro S500. These drones were specifically chosen as they support Pixhawk 4 [9], an advanced autopilot system and ArduCopter [2] firmware. Refer to Figure 4 for the hardware setup. When the position and velocity variance crosses a set threshold, the flight controller activates an *EKF Variance* error alongwith a *GPS Glitch* error. As a response to these errors, the UAV executes the programmed fail-safes. By default, ArduCopter switches to *LAND* mode and lands at the current location. To prevent this, we temporarily disabled EKF and GPS failsafes by manipulating the *FS_OPTIOINS* parameter. During the identification maneuver the UAV undergoes rapid acceleration / deceleration in an unpredictable direction defined by NED velocity vector

relative to its own body frame. To achieve this, we used the *SET_POSITION_TARGET_LOCAL_NED* MAVLink message type to instruct the drone to move according to the specified velocity vector. In our implementation, we used *DroneKit* [3] to generate the maneuver and instruct the flight controller to execute it. A specific sequence of these messages then carries out the entire maneuver. Once the UAV completes the maneuver, API performs the correlation operation as described in Section III-C and notifies LSR over a TCP socket.

B. Legitimate Signal Retriever (LSR)

In SemperFi, we implement LSR as a part of GNSS-SDR's acquisition module. As mentioned earlier, we use auxiliary peak based spoofing detection technique (detects seamless takeover attack) and implement several navigation message sanity checks (detects changes to navigation message contents) to detect an attack as proposed in [65]. We modified the acquisition block such that spoofing detection module is enabled every time the acquisition block is activated. This allows SemperFi to recover from hard spoofing attacks during which the receiver loses lock (stops tracking the satellite signals) and initiates re-acquisition due to the abrupt change in received GPS signals. Positive detection of an adversarial signal triggers further processing that includes peak identification, recovery signal generation, and signal recovery. GNSS-SDR allows external communications using TCP sockets as outlined here [36]. This enables GNSS-SDR to interact with the UAV's flight controller responsible for performing peak identification maneuvers. Once the API validates spoofing and provides peak information, LSR enters the cancellation and recovery state. At this stage, LSR has the peak information and a rough estimate of the Doppler and the code phase delay of the satellite signal. The accuracy of parameter estimates is directly related to the degree to which the adversarial peaks may be attenuated; SemperFi performs re-acquisition using a more refined grid search to obtain more precise estimates. After performing a narrow search, LSR generates a replica of the satellite signal using the tracking parameters estimated in the two-step acquisition process. LSR also estimates the satellite signal's amplitude using the method described in Section III-D. We use the Vector-Optimized Library of Kernels [79] function to perform vector operations. These functions provide a significant boost to performance and reduce computation time. Once the signal is regenerated, it undergoes phase correction and cycles through phase shifts to determine maximum attenuation. In certain cases, due to inaccuracies in the amplitude, Doppler, and the code phase delay estimates, a single attempt at recovery will not entirely attenuate the adversarial peak. SemperFi iterates the entire acquisition and recovery process until the legitimate signal is stronger than the adversarial signal.

C. Pseudorange Rectifier:

This module is implemented as an optional component in the tracking module and is disabled by default. The receiver enables pseudorange rectifier if the navigation message decoder fails to detect a preamble even after tracking the correct peak. Even if the navigation message decoder can find preamble and decode the navigation bits, there is a possibility that adversarial peak interferes with correct PVT estimation. In these cases, the receiver will activate pseudorange rectifier. Pseudorange Rectifier can also be activated manually by setting a flag in the receiver configuration file. When pseudorange rectifier is activated, the tracking module tracks the adversarial peak instead of the legitimate peak. It, however, still obtains tracking parameters of the legitimate peak. It uses legitimate and adversarial code phase information to calculate $\Delta\tau^i$. Code phase information and subframe start pointer determined by preamble position in a buffer of samples are used to determine the ToA of satellite signals. A sample counter accurately maintains this information. $\Delta\tau^i$ is used to offset sample counters appropriately. The receiver still decodes adversarial navigation messages; however, it uses the ToA of legitimate signals for pseudorange calculation to calculate the correct PVT solution in those specific scenarios where the attacker spoofs a location by manipulating ToA of signals and keeps the navigation messages same.

D. Integration for Real-Time Operations

For SemperFi to be operational, we must integrate all the functions such that they operate as a single unit. There are specific engineering challenges related to the design architecture of GNSS-SDR that limits us from integrating all the modules. However, we note that these challenges are independent of the proposed techniques and do not exist when implemented directly in hardware (e.g., FPGA). The main challenge is integrating GNSS-SDR with the UAV's flight controller. For SemperFi to operate with RF-frontends, it requires modifications to GNSS-SDR architecture which includes implementation of a particular type of asynchronous data structure that can tag and pool signal samples. One issue is that RF-frontends strictly require synchronous access to the signal samples i.e., the producer and the consumer operate in real-time. Pausing the consumption of samples breaks the connection to the RF-frontend and this is an essential requirement for SIC to operate. A solution to this challenge is to modify the underlying GNSS-SDR and GNURadio framework to add a controlled *null sink* to continue sample consumption even if the flowgraph is temporarily paused. Alternately, we can implement a tracking loop that can converge and successfully track the carrier signal even after the delay introduced by the cancellation process such as the ones proposed in [67]. Another challenge is power consumption; GNSS-SDR is a tool designed for research and development; it provides an avenue for developing proof-of-concept systems. However, it has high resource usage and hence is not the best solution for small, low-powered embedded systems. This work focuses on the implementation of GPS signal processing required to provide a robust GPS spoofing mitigation solution and to that extent, we have implemented provisions that can allow the individual components to communicate and operate as a single system. This indeed is a limitation of SemperFi's implementation in its current state.

V. Security and Performance Evaluation

A. Theoretical Security Evaluation of Identification Maneuver

In this section we conduct a theoretical evaluation of the peak identification from an attacker's perspective. We establish the fundamental property that ensures success of the identification maneuver. The drone adopts a discrete-time linear

kinematic model for its own behavior, which is represented by the general process model

$$\mathbf{x}_{n+1} = \mathbf{F}_n \mathbf{x}_n + \mathbf{G}_n \mathbf{u}_n + \omega_n \tag{14}$$

The drone tracks the time-evolution of own state, which in its most basic form is comprised of three-dimensional position, velocity and acceleration $\mathbf{x}_n = [x_n, \dot{x_n}, \ddot{x_n}, y_n, \dot{y_n}, \ddot{y_n}, z_n, \dot{z_n}, \ddot{z_n}]^\top$. To this, it incorporates $\mathbf{F}_n$ and $\mathbf{G}_n$, which represent the matrix forms of the kinematic equations and controller action respectively, along with the controller input $\mathbf{u}_n$, which contains additional information about the controlled acceleration of the aircraft. Through the use of on-board sensors, the drone observes its own position and velocity. These observations are incorporated into the model by way of the measurement equation

$$\mathbf{y}_n = \mathbf{H}_n \mathbf{x}_n + \nu_n \tag{15}$$

where the observation vector $\mathbf{y}_n$ is comprised of the three-dimensional position and velocity of the craft $\mathbf{y}_n = [x_n, \dot{x}_n, y_n, \dot{y}_n, z_n, \dot{z}_n]^\top$. The sensor fusion algorithm implemented in the UAV monitors the position and the velocity variance and triggers a fail-safe if the variances exceed a pre-determined threshold. Thus, the attacker has to generate the spoofing signal such that this failsafe is not triggered. To achieve this the attacker can also adopt a discrete-time linear kinematic model for the behavior of its target. This model is similar to the model used by the drone to estimate its own position, but there are key differences. In particular, we consider a process model of the form

$$\mathbf{x}^A_{n+1} = \mathbf{F}^A_n \mathbf{x}^A_n + \omega^A_n \tag{16}$$

In comparing this to the model used by the drone, we observe that the attacker has no knowledge of the input $\mathbf{u}_n$ imposed by the drone's controller. Additionally, since the attacker has no access to the internal sensors of the drone, the attacker model differs in the observations available. In general, the attacker relies entirely on positional observations from radar or imaging systems to perform its tracking. Hence, the attacker observation vector $\mathbf{y}^A_n$ is comprised of only the three-dimensional position $\mathbf{y}^A_n = [x_n, y_n, z_n]^\top$. The discrepancy between the models used by the drone's own tracking and those used by the attacker results in positioning inconsistencies which are reflected in the spoofed position observed by the drone's GPS. This discrepancy and the resulting inconsistencies result in high position and velocity variance, which can be leveraged to detect interference by an attacker. By increasing the magnitude of the input $\mathbf{u}_n$ which is known only to the drone's own internal tracking, the effect of the model discrepancy can be exacerbated, thus increasing the rate at which the inconsistency in positioning will grow and consequently *decreasing* the amount of time required to detect a seamless takeover attack.

Critically, the attacker in this scenario has no access to information that is internal to the target (e.g. IMU measurements, guidance information, controller information). Of particular interest is the controller input: if the target induces an input to the system model by way of a control input, the attacker model will be *mismatched* with respect to the true model of the target. Over time this discrepancy will result in an accumulation of positioning errors, which can be detected by the target. To demonstrate this, we must analyze the probabilistic basis of the EKF used by the attacker to track the position of its target. The objective of the Kalman filter in general is to recursively determine the Gaussian posterior distribution given a set of sequential observations. The predictive and posterior densities can be approximated by a Gaussian filter as [68]

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \mathcal{N}\left(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t-1}, \mathbf{\Sigma}_{t|t-1}\right) \tag{17}$$

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \mathcal{N}\left(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t}, \mathbf{\Sigma}_{t|t}\right) \tag{18}$$

Computation of the posterior density is done by a two-stage procedure of prediction and update. Prediction is performed by propagating the mean and posterior of the previous posterior estimate characterized by $\mathbf{x}^A_{t-1|t-1}$ and $\mathbf{\Sigma}_{t-1|t-1}$ through the process model given in (16).

$$\mathbf{x}^A_{t|t-1} = \int \hat{\mathbf{f}}(\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \tag{19}$$

$$\mathbf{\Sigma}_{t|t-1} = \int \hat{\mathbf{f}}^2(\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} - \mathbf{x}^{A2}_{t|t-1} + \hat{\mathbf{Q}}_{t-1}$$

Due to the aforementioned attacker limitations, including a lack of knowledge of the input provided by the target drone's controller, both the previous posterior $p(\mathbf{x}_{t-1}|\mathbf{y}_{t-1})$ and the predictive model $\mathbf{f}(\mathbf{x}_{t-1})$ will differ from the true behavior of the target, resulting in a predictive distribution which is increasingly uncharacteristic of the true position of the target. Even if the subsequent update step proceeds without issue, the resulting posterior estimate characterized by $\mathbf{x}^A_{t|t}$ and $\mathbf{\Sigma}_{t|t}$ will be increasingly diverged from the true position of the target with each iteration. If this estimate is then transmitted to the target in the form of a spoofed GPS signal, it will result in an observation $\mathbf{z}$, which is compared by the target against its own prediction, which was made based on the fully-informative process model given by (14) according to the Kalman innovation equation

$$\boldsymbol{y} = \boldsymbol{z}_k - \boldsymbol{x}^A_{k|k-1} \tag{20}$$

Since the observations $\mathbf{z}_k$ do not come from the true distribution, but are instead the product of spoofing, the resulting innovation y will increase, and consequently so will the computed innovation covariance. Over subsequent iterations, this innovation covariance will continue to increase until it eventually exceeds the threshold set in the drone configuration.

Despite the lack of knowledge about the UAV's true motion, the attacker can guess the acceleration or the controller input $\mathbf{u}_n$ to track the unknown identification maneuver performed by the UAV. With prior knowledge of UAV's configuration, the attacker can narrow down each step of the maneuver to a finite set of possible instantaneous acceleration values. The attacker guesses the change in acceleration value from a set of possible values.

$$A = \{\boldsymbol{x} \in \mathbb{R} | -\boldsymbol{j} < \boldsymbol{x} < \boldsymbol{j}, |\boldsymbol{x}_n - \boldsymbol{x}_{n-1}| = r\} \tag{21}$$

where $\boldsymbol{j}$ is the maximum possible instantaneous jerk and r is the resolution of the accelerometer. The maximum possible change in acceleration is defined as the maximum instantaneous jerk. Thus, the maximum instantaneous acceleration is directly proportional to the maximum *jerk* that the UAV supports. These values are then integrated to estimate respective velocity and position components. Attacker's probability of guessing the correct value is $P(A) = 1/|A|$. However, the attacker has to correctly guess the values for each step of the maneuver. The probability of attacker's success $P(S)$ is

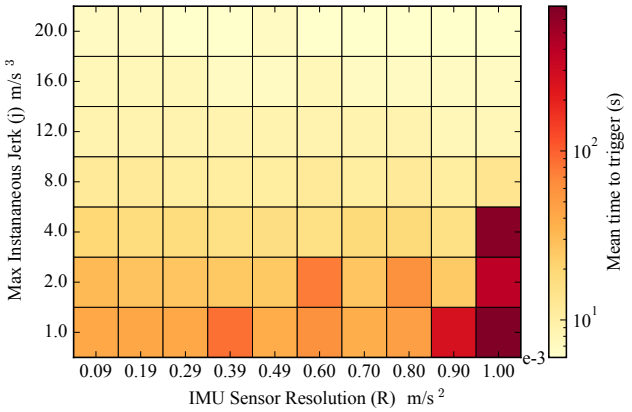


Figure 5: This plot shows the effect of sensor resolution and maximum possible instantaneous jerk on time to trigger. An attacker can stay undetected for more than 100 s when attacking a slow moving UAV and with a low resolution sensor. However, if the UAV is capable of a rapid motion it can detect an attack in less than 10 s.

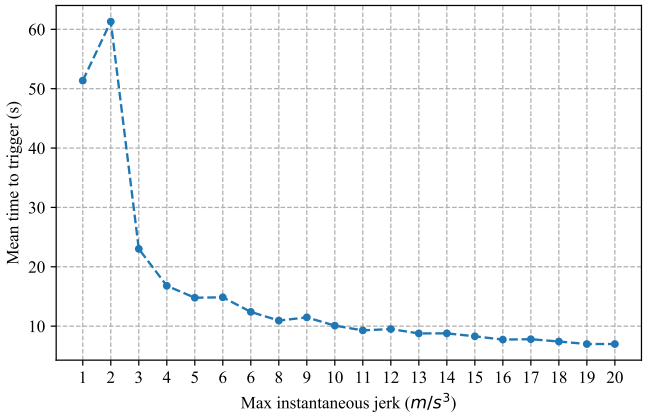


Figure 6: A plot showing the mean time to trigger for various **j** values for a UAV that uses an IM-20689 inertial sensor. An attacker will be detected in less than 10 s if the UAV's rate of change of acceleration is atleast 10 m/s^3.

given as $P(A)^{\mathbf{n}}$ where $\mathbf{n}$ is the total number of steps in the maneuver. In this *guessing-game*, the attacker's success depends on the $\boldsymbol{j}$ that the UAV is capable of and the r of the on-board sensors. To study the effect of these properties on attacker success we evaluated the time taken to trigger the error. We built a simulation that updates at 400 Hz[3] where the UAV spontaneously performs a maneuver that is unknown to the attacker and the attacker uses its knowledge of the UAV to guess the maneuver. We follow a threshold mechanism similar to ArduCopter. Specifically, we calculate the mean square error of the position and velocity obtained from the attacker's guesses and compare it against a threshold value obtained from ArduCopter's implementation. To account for randomness, we performed over 110,000 simulations, these results of which are summarized in Figure 5. An attacker can stay undetected for more than 100 s when it is attacking a UAV with a low resolution sensor and is incapable of rapid motion. In case of a lower $\boldsymbol{j}$ value and a lower r value, the accumulation of errors as a result of discrepancy in the UAV's true position and velocity and the spoofed position and velocity is smaller and slower. Hence, the UAV will take longer time to trigger the error. Through our experiments we observed that the mean time to trigger depends majorly on the UAV's acceleration capabilities. Thus, even if the UAV is using a low resolution sensor, it can force trigger the error by being fast enough. The resolution of a MEMS sensor depends on the resolution of the ADC used in the sensor. ADC's resolution is represented as *bits*. A typical inertial sensor like IM-20689[4] [7] with a 16-bit ADC and acceleration range of $\pm 2\,\mathrm{g}$ has a resolution of 0.000598 m/s^2. Figure 6 shows the mean time to trigger for a UAV that uses this sensor. An attacker will be detected in less than 10 s if the UAV is capable of changing it's acceleration at atleast 10 m/s^3.

[3]Update rate of a typical UAV flight controller

[4]UAVs that are a part of our evaluation use this type of IMU sensor.

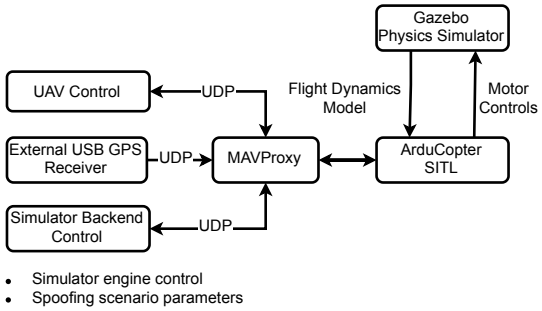


Figure 7: Our experimental setup consists of an ArduCopter software-in the loop simulator which uses Gazebo as a physics simulator. UAV Control, GPS receiver and Simulator Backend Control are implemented using *dronekit* and interact with the simulator through MAVProxy.

B. Experimental Evaluation of Identification Maneuver

In this section, we analyze the security and performance of identification maneuvers by evaluating it in a simulated environment using Gazebo [6] as well as in a practical real-world setting using real drones. We used ArduCopter [2] for both cases. Figure 7 offers a schematic view of our experimental setup and the implementation of spoofing scenario simulator. All the values presented in this analysis are specific to the UAV that we tested. These values are heavily dependent on the physical capabilities of the UAV. For this analysis, we assume a scenario in which the attacker is successful in executing a seamless takeover attack. The auxiliary peak detection will raise an alarm and instruct the flight controller to initiate the maneuver. This will occur when the peak separation is more than 500 ns as described in [65]. We evaluate the peak identification strategy by studying the time required for the drone to forcefully trigger the EKF variance error in case of a GPS spoofing attack. We consider the following scenarios:

1. Static non-adaptive signal spoofing attack: We assume

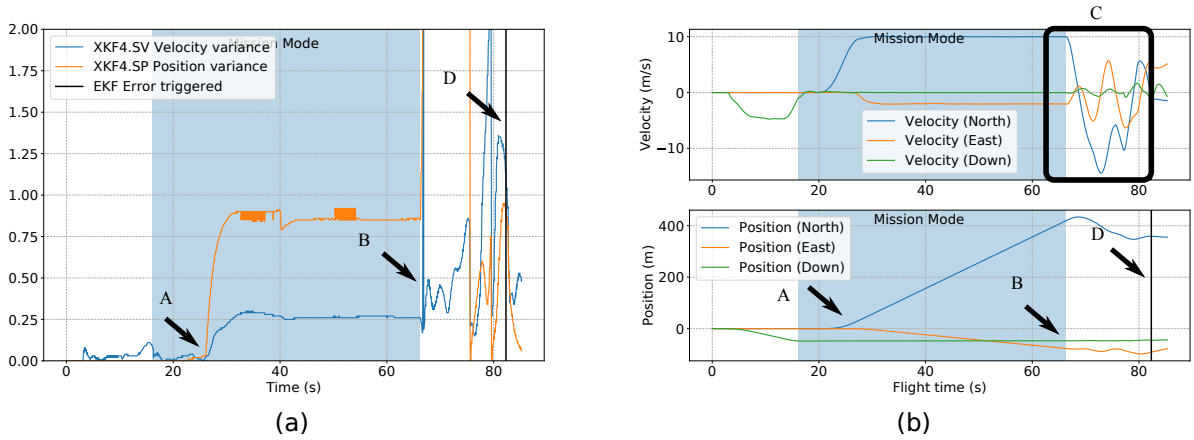


Figure 8: A plot showing (a) EKF innovations variance and (b) position data for the moving target adaptive GPS takeover scenario. The spoofer starts introducing offsets at marker A. This is evident from rise in position variance. At marker B, SemperFi kicks in, pauses the mission and instructs the UAV to perform a maneuver that is marked at C, the maneuver initiates a series of rapid changes in velocity components that increase the velocity variance. Finally at marker D the EKF variance is triggered.

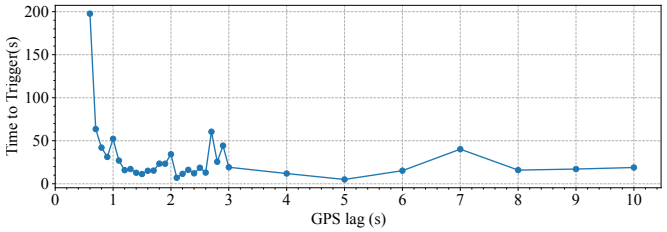


Figure 9: The effect of GPS lag on time to trigger the EKF variance.

that the UAV is hovering when the attacker starts spoofing a static location. This will force the drone to slowly start drifting because of IMU error accumulation. For this scenario we performed two tests. First, we let the UAV drift freely to observe its behavior. In this case the UAV drifted for 47.01 s before a GPS glitch was detected and the EKF error was raised. During this process, the drone drifted 1.160 km away from its initial position. In the second test we instructed the UAV to perform an identification maneuver which consisted of a series of velocity vectors to induce acceleration as described in Section V. As a result of this maneuver, the EKF error was raised in 5.74 s.

2. Stationary target adaptive GPS takeover: In this scenario, the attacker performs an adaptive GPS takeover attack on a stationary UAV that is instructed to hover at the current location. The goal of the attacker is to move the UAV to an arbitrary location of the attacker's choosing. The attacker has managed to perform a seamless takeover attack and now the attacker starts inserting offsets to the spoofed GPS positions. The UAV starts correcting itself according to the GPS positions it receives. If SemperFi does not intervene the UAV will keep drifting as guided by the attacker. In this scenario, when the peak separation is more than 500 ns the UAV performs the maneuver and is able to trigger the error in 11.94 s.

3. Moving target adaptive GPS takeover: In this scenario, the attacker performs an adaptive GPS takeover attack on a moving UAV that is traveling from point A to point B. We assume that the attacker is aware of the UAV's path. Just like scenario 2, the attacker deviates the UAV by inserting offsets to the spoofed GPS position and has managed to perform a seamless takeover attack. The UAV starts correcting itself according to the GPS positions it receives. As soon as auxiliary peak is detected, the UAV performs the identification maneuver and is able to trigger the error in 18.001 s. Refer to Figure 8 for a timeline of events. Similar to scenario 2, the UAV will keep following the spoofed locations until any failsafe is activated.

To evade identification after the maneuver, the attacker needs to take the time lag into account that the UAV is going to experience between its true position and the spoofed position. The maximum tolerable GPS lag t for a particular UAV is given by

$$t = \frac{\Delta V_{min}}{a_{max}} \tag{22}$$

where ΔV_{min} is the minimum error in velocity that triggers the variance error and a_{max} is the maximum acceleration of the UAV. To study the effect of GPS lag induced by an attacker and the underlying tracking technology, we performed multiple simulations where we purposefully added a delay to the GPS emulator component of the physics simulator. Refer to Figure 9 for the results of the experiment. From these simulations, we observed that the UAV was able to trigger the EKF error with 100% certainty for a lag of 600 ms and above. For values less than 600 ms, we observed that out of 110 flights with 500 ms GPS lag, with the maneuver, EKF error was raised just 35% times. Based on these simulations we set the lower bound at 600 ms. As evident from Equation (22), it is important to note that this lower bound is specific to a particular model of the UAV as it depends on the overall capabilities of the UAV.

1) Maneuver Design Consideration: In this section we consider the process of designing a maneuver which is specifically designed to force position and velocity variance. In

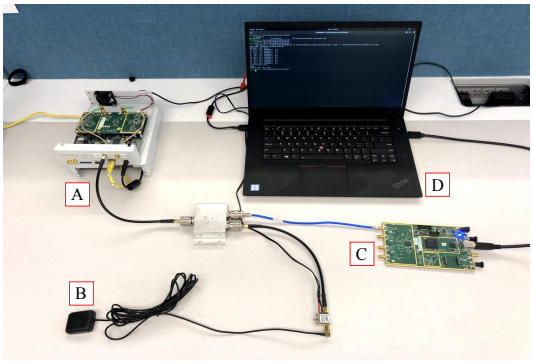


Figure 10: Signal recording setup A) GPS signal RX (USRP N210), B) ANT-555 active GPS antenna with a 5V bias-tee, C) GPS signal TX and D) GPS simulator control unit.

our implementation, our maneuver design is specific to ArduCopter. With that said, the method is highly configurable based on EKF implementation and capabilities of a specific UAV, lending itself to implementation on other platforms. In ArduCopter the EKF algorithm first raises a glitch error if the calculated GPS position is outside the configured GPS radial uncertainty region. By default this is set to 25 m. This radial uncertainty radius and the lag in GPS position is used to calculate the velocity required to exceed the position variance. In our implementation we set the attacker's delay to 600 ms based on our experiments. Possible maneuvers are limited by the capabilities of the UAV.

2) False Positive Analysis: We performed multiple flights with aggressive maneuvers in a non-adversarial setting. The objective was to observe the position and velocity variance. We were able to perform maneuvers where we took the UAV to its maximum capability in terms of acceleration (5 m/s^2) and jerk 20 m/s^3 without triggering the EKF variance error that we mentioned above. Furthermore, since the maneuver is triggered only in a situation where the EKF variance error is in check in spite of the presence of auxiliary peaks, the likelihood of a false positive is low.

C. Experimental Performance Evaluation

In this section we present SemperFi's experimental performance in recovering legitimate GPS signals under various adversarial scenarios. The chosen metrics for evaluating the recovery process are amplitude estimation accuracy, the accuracy of the recovered location, and the time required to perform recovery. We also analyze the effect of the attacker's synchronization and its power advantage over the legitimate signals on the recovered location's accuracy. Finally, we discuss and evaluate the effect of jamming attacks on the drones.

1) Evaluation Traces: We use three different datasets that contain both spoofing and legitimate signals: i) Synthetic GPS signals generated using COTS GPS simulators, ii) a public repository of GPS spoofing signals (TEXBAT) [41], and iii) recorded real-world GPS signals.

a) GPS Simulator: We performed most of our evaluation on synthetic signal traces generated locally using GPS-SDR-SIM [17], an open-source tool for generating GPS signals. This provides granular control over signal properties such as power levels, temporal delays, and Doppler shifts; thus enabling us to generate a variety of spoofing scenarios. We evaluated SemperFi against both static (stationary locations) and dynamic scenarios (motion trajectories). These signals were transmitted using two USRP B210s, one each for the legitimate and attacker signal. We recorded the signals using a USRP N210 at a rate of 10 MSa/s. We wired all RF-frontends to prevent signal leakage as it is illegal and hazardous to transmit GPS signals. For static and dynamic scenarios, we picked locations in downtown San Francisco. We generated the attacker's signal such that the obtained location is at a specific offset from the legitimate location. We picked locations with the offset increasing in steps of 500 m up to a maximum spoofed offset of 3500 m.

b) Texas Spoofing Test Battery (TEXBAT): TEXBAT is a set of civilian GPS spoofing scenarios that are a standard for evaluating spoofing countermeasures. The repository consists of spoofing signals traces that include both position and time push scenarios. TEXBAT also provides scenarios where the attacker's signals and the legitimate signals are synchronized, similar to the strong seamless-takeover attack. We evaluate the effectiveness of SemperFi against both static and dynamic position push. These signal traces were recorded at 25 MSa/s. The traces are 7 mins long, and the attacker starts spoofing roughly $90 - 100$ s into the signal trace.

c) Live GPS Recordings: We also evaluated SemperFi against a combination of live legitimate GPS signals and attacker signals. This scenario covers the real-world spoofing scenario where the attacker transmits spoofing signals while the receiver is locked on to legitimate signals. We recorded a set of real-world GPS signal traces through extensive war-driving in Boston. We recorded the legitimate GPS signals using the setup shown in Figure 10. We captured the GPS signals using an ANT-555 antenna supplied with a 5 V DC power supply. We combined the received signal with the attacker signals using a combiner and used GPS-SDR-SIM to generate attacker's signals. The spoofed location was set to 4.1 km away from the original location. Hard-wiring the attacker allowed us to test in a best-case scenario for the attacker as they have a clear channel to the victim receiver and evaluate its performance in eliminating the spoofing signal.

2) Amplitude Estimation: It plays a vital role in successful signal recovery. In SemperFi, we leverage the max CAF value or the correlation coefficient value to estimate the original signal's amplitude. In this strategy, the estimate's accuracy is susceptible to various factors like interference caused by signals from other satellites, the presence of adversarial signals, and artifacts introduced by a wireless channel. For evaluating the accuracy, we conducted an experiment where we executed amplitude estimation in four cases. The accuracy of amplitude increases as the attacker's power advantage increases. SemperFi compensates for the inaccuracies in amplitude estimation caused by Doppler shifts, clock skews, and phase shifts by executing multiple iterations of the signal recovery process and successfully attenuates the adversarial signal. Refer to Figure 11 for results.

3) Recovered Location Accuracy: We evaluate SemperFi's effectiveness in eliminating the spoofing signal by determining the location's accuracy after passing through the various blocks of SemperFi. We use the Universal Transverse Mercator

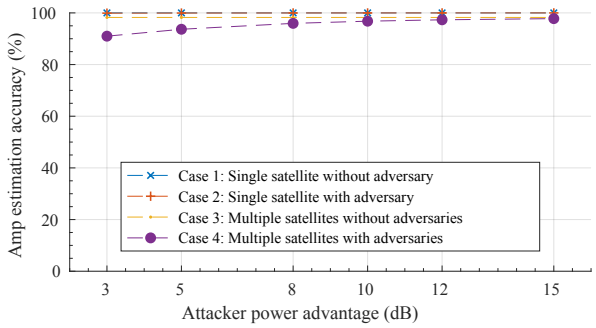


Figure 11: Amplitude estimation accuracy in various scenarios. Attacker power advantage does not apply to cases 1 and 3. In case 4, each satellite is spoofed. Power advantage refers to the advantage that the attacker's signal has over the legitimate signal.

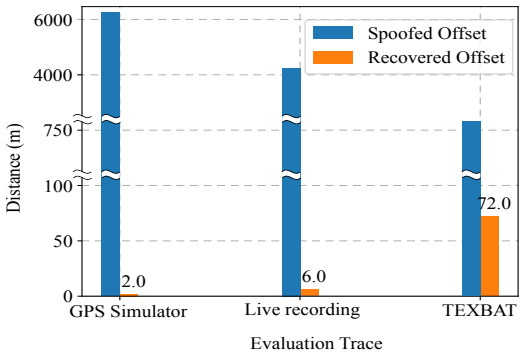


Figure 12: The spoofed offset and the recovered offset for three scenarios.

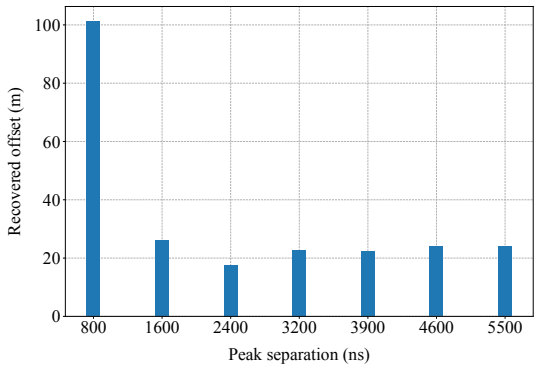


Figure 13: The effect of peak separation on accuracy of the recovered location. The closer the peaks, the harder it gets to accurately track them. Power advantage is 3 dB which is strong enough to takeover the receiver and yet not strong enough to bury the signals under noise. It also allows us to evaluate the effects of signal synchronization on signal recovery.

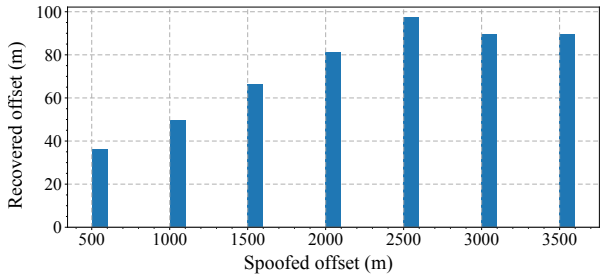


Figure 14: Spoofed offset vs offset in recovered location for attacker with 15 dB power advantage. SemperFi uses Pseudorange Rectifier for recovery. For locations refer to Section V-C1.

(UTM) [15] system to present our location accuracy results. We evaluated the performance of SemperFi against both static and dynamic scenario spoofing attacks present in the datasets described in Section V-C1.

First, we evaluate the performance of SemperFi against the dataset generated using GPS signal generators. The UTM plots depict the variations in locations and a timeline of events. Figure 12 shows the recovery operation results on static scenarios across all three datasets. GPS simulator traces where the spoofed offset is 6.2 km with recovered offset of 2 m. Live recording with a recovered offset of 6 m. TEXBAT's power matched position push scenario where the attacker spoofs only in Z plane. Figure 12 shows varrying recovered offset as a result of attacker signals' synchronization with the legitimate signals. More details are present in the following section.

4) Attacker synchronization: One major factor that affects recovered locations is attacker synchronization with legitimate signals. In other words, the effectiveness of eliminating spoofing signals depends on the temporal shifts in the ToA of legitimate and spoofing satellite navigation messages. The closer the synchronization, the harder it is to recover entirely without additional processing. We evaluated the effects of attacker synchronization by generating spoofing scenarios where the attacker spoofs locations with an offset in the increments of 500 m from the original position. This results in a corresponding temporal shift between the attacker's spoofing signal and the legitimate signal. The minimum peak separation was 800 ns at 500 m, and the maximum peak separation 5500 ns at 3500 m. Note that this peak separation depends on the satellite constellation at any point in time. Figure 13 shows the results of this experiment. Peak separation directly affects how the attacker's signals interact with legitimate signals as peaks that are too close (e.g., less than $1\mu s$) poses a challenge to the tracking loops, and as a result, the tracking loops undergo signal cross-over and the tracking loop starts tracking the wrong signal. This is evident from the higher recovered location offset for the scenario with peak separation of 800 ns.

5) Effect of Attacker's Power Advantage: We evaluate the performance of SemperFi against attackers with varying power levels up to 15 dB. Note that in seamless takeover attacks, the maximum power difference required to execute the attack successfully is not more than $2-3$ dB [41], [73]. TEXBAT repository's seamless takeover attack data-trace has a power difference of not more than 10 dB. We created spoofing scenarios where the attacker has a power advantage of 3 to 15 dB. SemperFi can attenuate stronger peaks and make the suppressed weaker legitimate peaks visible in the acquisition

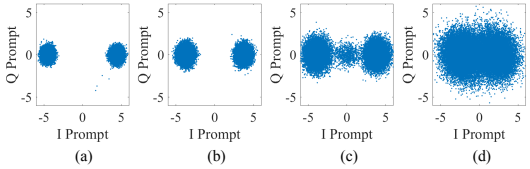


Figure 15: Discrete time scatter plot of recovered nav message where attacker has (a) 3 dB, (b) 5 dB, (c) 10 dB, and (d) 15 dB power advantage. A powerful attacker adds noise and hence distorts legitimate navigation messages.

plot. Figure 16 shows a multi-stage attenuation process for an adversary with 15 dB power advantage. However, as seen in the discrete-time scatter plot in Figure 15(d), in the case of an attacker with a 15 dB power advantage, the adversarial signal introduces much noise, which distorts the navigation bits. In such a scenario, despite the reduced accuracy SemperFi can enable our pseudorange rectifier and recover the correct location by rectifying pseudoranges. Figure 14 shows the results of signal recovery in the presence of an attacker with a 15 dB power advantage. A typical drone flies at an altitude of 50 m and the antenna is installed pointing upwards. For a standard GNSS antenna, gain below the horizon starts dropping below -15 dB at 0° [1] which means, an attacker who is on ground is already has a disadvantage of 15 dB. Moreover, an attacker trying to compensate for this power disadvantage can be easily localized thus making it easier for detection.

6) Real-time performance: We evaluate the SemperFi's performance by deploying and executing it on the following embedded platforms: i) NVIDIA Jetson Nano, ii) NVIDIA Jetson Xavier, iii) Intel Core i7[5], and iv) Intel Xeon E5-2630[6]. These systems are some of the standard systems used as flight controllers onboard UAVs. We use signal traces as described in Section V-C1 for evaluating the performance. The sampling rate of 10 MSa/s plays a significant role in determining the performance of SemperFi as it is directly related to the processing overhead. Our primary evaluation metric is the time required per iteration of cancellation. It is important to note that GNSS-SDR is itself a resource-demanding application. Refer to Table I for a comparison showing each system's performance. We executed SemperFi over 2000 times on various datasets to investigate the number of iterations required for successful recovery. According to our experiments, on an average each execution required 2.33 iterations to complete the recovery process. Standard deviation and variance are 1.65 and 2.73 iterations respectively. The number of iterations depends on the attacker's synchronization and the power advantage over legitimate signals. It is important to note that our implementation of SIC is sensitive to missed samples and sample losses result in more iterations. In some cases SemperFi required just two iterations to recover the signal. Thus, complete signal recovery may add delay to the calculation of the PVT solution; in the case of Jetson Xavier, for example, by 0.54 s which is sufficient in most cases as the identification maneuver is required only in certain cases. It is important to note that these values are from a sub-optimized version of SemperFi. It is possible to improve the performance by optimizing SemperFi for a specific system that leverages its unique characteristics and features. For example, SemperFi can be re-programmed to use CUDA cores available on NVIDIA Jetson Nano and Xavier. In general, it is best to deploy SemperFi on an FPGA as it will significantly improve the performance.

[5]https://www.dji.com/manifold-2

[6]not currently used in any UAV platform and ported only for comparison

Table I: A comparison of time required by the corresponding system to perform one iteration of signal cancellation.

Model	Processing time
Jetson Nano	0.8 s/itr
Jetson Xavier	0.23 s/itr
Intel Core i7	0.2 s/itr
Intel Xeon	0.11 s/itr

VI. Discussion

A. Flexible design

SemperFi is designed to be flexible and versatile. In addition to integrating SemperFi into the acquisition module as shown in Section IV, we can use SemperFi as a plug-gable module that can filter out adversarial signals and pass on legitimate signals to a conventional receiver. This mode of operation requires minimal modifications to the existing receivers. Furthermore, SemperFi's can be adopted for other satellite navigation systems like GALILEO as they follow a similar operating principle of code division multiple access using spreading codes and computation of pseudoranges.

B. Limitations and Future Work

An attacker capable of predicting the maneuver and generating appropriate spoofing signals in real-time to defeat SemperFi may use several techniques like acoustic sensors, ultra-wideband scanners, visual sensors, and directional RF antennas to track and localize drones [11], [27], [28], [32], [39], [70]. However, these works are restrictive in terms of coverage area, tracking precision, and latency. Drone localization and tracking system that uses acoustic sensors are effective only up to 300 m while the system that relies on radio telemetry transmissions has an update rate of just 1 Hz. It is important to note that the attacker also needs to generate and transmit the spoofing signals. This requirement makes such an attack extremely challenging.

Another limitation of SemperFi's current implementation is that tracking legitimate signals fails if the attacker has a power advantage of more than 15 dB. We note that this 15 dB limit is a limit defined by our signal processing hardware and peripherals like multiple directional antennas and receivers can further increase the 15 dB limit. Moreover, an attacker transmitting with more than 15 dB of power advantage can easily be detected and localized by the receiver.

An attacker can also cause a denial of service attack by transmitting multiple signals that can overload the system. Even though SemperFi can handle multiple peaks through an

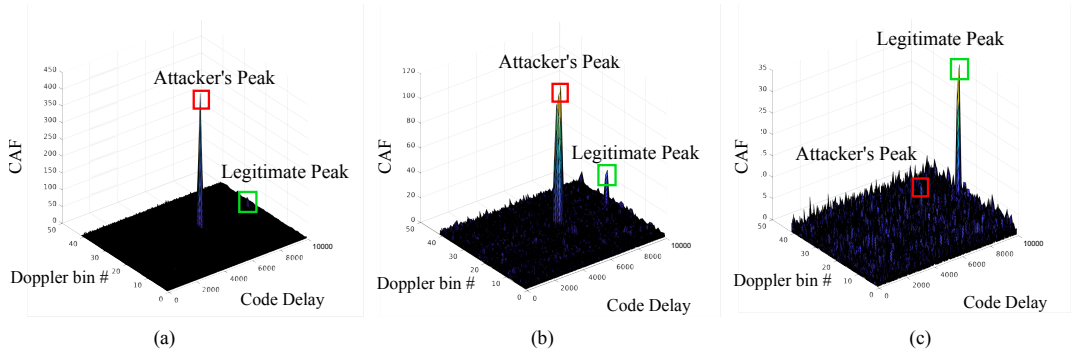


Figure 16: Two step signal attenuation of a strong adversarial signal. (a) shows the original acquisition plot, (b) shows acquisition plot where legitimate peak is slightly visible and (c) final acquisition plot with fully suppressed adversarial peak.

iterative cancellation process, it is prone to resource exhaustion as each iterative cancellation increases process overhead. As future work, we plan to investigate techniques to quickly identify the legitimate signal amongst several spoofing signals and amplify it.

Finally, the proposed maneuver technique works well for UAVs and it is challenging to design these maneuvers for terrestrial vehicles because of mobility constraints. It is important to note that, in [56], the authors show that an attacker can exploit the short-term stability of IMU sensors due to predictable maneuvers in an urban setting. The problem is similar when a UAV is flying between obstacles. However in that case, the attacker also has similar constraints to force the drone onto a different path successfully. Even if the drone is operable, frequently changing weather conditions, especially wind vectors, can affect the drone's maneuverability, especially high-velocity crosswinds. However, the algorithm can be modified to work with crosswinds by determining the force and velocity of wind as proposed in [58] or by equipping the drone with solid-state anemometers.

VII. Related Work

Several GPS spoofing countermeasures have been proposed in the past. Majority of these works focused on building spoofing detection techniques and do not address the challenge of neutralizing the attacker's spoofing signals. In this paper, we present SemperFi, a single antenna GPS receiver specifically designed for UAVs, that is capable of tracking legitimate GPS satellite signals and estimating the true location even during a spoofing attack. The work that comes closest to ours is the spoof-proof GPS receiver [31] and the in-line GPS spoofing mitigation technique [47]. In [31], the receiver uses maximum likelihood estimates after dampening the attacker signal to estimate the correct location. The in-line GPS spoofing mitigation technique [47] implements an extended RAIM method to filter outliers and correlation peak distortion techniques to detect spoofing signals. Both these works are incapable of distinguishing adversarial peaks and fail against strong adversaries such as a seamless takeover attacker. Signal cancellation has been explored as a method to attack GPS receivers in [55] by attenuating a specific satellite. Furthermore, successive interference cancellation has been used to eliminate the near-far problem associated with pseudolites [49]. The authors treat overpowering pseudolites as interference because despite the signal being legitimate, it is so powerful that signals from GPS satellites are buried under the noise-floor. In other words, there was no ambiguity in determining the exact signal to be canceled. Some proposals [51], [52] explored the use of null steering to reduce the effect of the attacker's signal. Such solutions require additional hardware and fail in a multi-spoofer setup, as described in [73]. Borio *et al.* [24] provide an interference cancellation technique for recovering from GPS jammers. This work statistically models GPS jamming signals, which aides in jamming signal removal.

Several cryptographic solutions have been proposed for securing navigation messages. In [29], [46], the authors propose an asymmetric and hidden marker approach for securing civilian GPS signals from signal-synthesis attacks. In [77], the authors propose an authentication scheme by incorporating digital signatures. All these cryptographic solutions, although they prevent signal spoofing attack, requires key distribution and management. It is important to note that GPS is a public service used by millions of devices worldwide. Deployment of these solutions requires serious modifications to existing GPS infrastructure, which is impractical. Furthermore, cryptographic solutions do not prevent record and replay attack [62].

Several spoofing detection schemes require extra peripherals like multiple antennas [22], [53], [54], which detect discrepancies in the angle of arrival of GPS signals. GPS signals and location estimates are correlated with data from extra IMU sensors [33], [42], [74], [76] for detecting GPS spoofing attacks using vector-based tracking. Extensive work is present that focuses on the use of EKF to aid in recovering from GPS glitches [40], [72]. ArduPilot has one such implementation. Our experiments found that a spoofer can avoid detection by controlling the introduced error in the positions. In [57], [80], authors show how an attacker can create signals to defeat Kalman filter-based detection algorithms and inject false sensor data. However, GPS/IMU sensor-fusion based navigation [56] has been recently shown to be vulnerable to attacks against on-road navigation systems. Several works [44], [73] propose using multiple receivers to detect spoofing signals by comparing reported positions of several GPS receivers with their deployed constellation.

Researchers have also proposed spoofing detection schemes

that correlate civilian GPS signals with military signals [64], cross-validation of PVT solutions across multiple navigation systems [59] e.g., GPS, GLONASS, Galileo, etc. Just like military signals, researchers have developed spoofing detection techniques that use opportunistic IRIDIUM signals [61] In [43], the authors leverage a crowdsourced network to detect GPS spoofing attacks. In [23], the authors discuss the use of deep learning schemes for spoofing detection and propose a detection approach based on machine learning. Another reliable GPS spoofing detection technique involves the use *device fingerprinting* technology [37] to detect GPS spoofing attacks by identifying legitimate satellites. Works like SPREE [65] and vestigial signal detection [78] provides a spoofing detection approach based on identifying auxiliary peaks. All the above countermeasures only perform spoofing detection and are incapable of autonomous recovery during the spoofing attack. To the best of our knowledge, SemperFi is the first receiver design in the open literature that reliably detects, identifies, and recovers from a majority of GPS spoofing attacks.

VIII. Conclusion

In this paper, we presented SemperFi, a single-antenna spoofer signal eliminating GPS receiver that is capable of providing uninterrupted legitimate locations even in the presence of a strong adversary. We designed, implemented SemperFi in GNSS-SDR capable of real-time operations and evaluated it using various GPS signal traces, real drones and popular embedded platforms. We showed that SemperFi is capable of identifying adversarial peaks by executing flight patterns less than 100 m long and recover the true location in under 0.54 s for most scenarios as identification maneuver is not required for all scenarios. Finally, we release the design and implementation of SemperFi to the community for usage and further research.

IX. Acknowledgements

This work was partially supported by NSF grant 1850264 and ECCS-1845833.

References

[1] Antennas. https://gssc.esa.int/navipedia/index.php/Antennas#Antenna_Power.

[2] Arducopter. https://ardupilot.org/copter/.

[3] Dronekit: Developer tools for drones. https://dronekit.io/.

[4] Ettus Research. https://www.ettus.com/products/.

[5] Flame Wheel ARF KIT. https://www.dji.com/flame-wheel-arf.

[6] Gazebo. http://gazebosim.org/.

[7] High Performance 6-Axis MEMS MotionTracking™ Device - Datasheet. https://3cfeqx1hf82y3xcoull08ihx-wpengine.netdna-ssl.com/wp-content/uploads/2021/03/DS-000143-ICM-20689-TYP-v1.1.pdf.

[8] Interfaces and Protocols. https://gssc.esa.int/navipedia/index.php/Interfaces_and_Protocols.

[9] Pixhawk 4. https://docs.px4.io/master/en/flight_controller/pixhawk4.html.

[10] Pixhawk4 S500 KIT. http://www.holybro.com/product/pixhawk4-s500-kit/.

[11] RFeye DroneDefense. https://www.crfs.com/drone-detection/.

[12] Simulators. https://www.navtechgps.com/departments/simulators/.

[13] Space segment. https://www.gps.gov/systems/gps/space/.

[14] The World Economy Runs on GPS. It Needs a Backup Plan. https://www.bloomberg.com/news/features/2018-07-25/the-world-economy-runs-on-gps-it-needs-a-backup-plan.

[15] What does the term UTM mean? https://www.usgs.gov/faqs/what-does-term-utm-mean-utm-better-or-more-accurate-latitudelongitude.

[16] (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/.

[17] (2015) Software-Defined GPS Signal Simulator. https://github.com/osqzss/gps-sdr-sim.

[18] (2019) Above Us Only Stars: Exposing GPS Spoofing in Russia and Syria. https://www.c4reports.org/aboveusonlystars.

[19] (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/.

[20] (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.

[21] M. G. Amin, P. Closas, A. Broumandan, and J. L. Volakis, "Vulnerabilities, threats, and authentication in satellite-based navigation systems [scanning the issue]," *Proceedings of the IEEE*, 2016.

[22] S. Bhamidipati, K. J. Kim, H. Sun, and P. V. Orlik, "GPS spoofing detection and mitigation in PMUs using distributed multiple directional antennas," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, 2019.

[23] P. Borhani-Darian, H. Li, P. Wu, and P. Closas, "Deep Neural Network Approach to Detect GNSS Spoofing Attacks," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 2020.

[24] D. Borio and P. Closas, "A fresh look at GNSS anti-jamming," *Inside GNSS*, 2017.

[25] K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen, *A software-defined GPS and Galileo receiver: a single-frequency approach*. Springer Science & Business Media, 2007.

[26] A. Bottcher and M. Dippold, "The Capture Effect in Multiaccess Communications-The Rayleigh and Landmobile Satellite Channels," *IEEE transactions on communications*, 1993.

[27] J. Busset, F. Perrodin, P. Wellig, B. Ott, K. Heutschi, T. Rühl, and T. Nussbaumer, "Detection and tracking of drones using advanced acoustic cameras," in *Unmanned/Unattended Sensors and Sensor Networks XI; and Advanced Free-Space Optical Communication Techniques and Applications*, 2015.

[28] X. Chang, C. Yang, J. Wu, X. Shi, and Z. Shi, "A surveillance system for drone localization and tracking using acoustic arrays," in *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2018.

[29] X.-j. Cheng, J.-n. Xu, K.-j. Cao, and J. Wang, "An authenticity verification scheme based on hidden messages for current civilian GPS signals," in *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*, 2009.

[30] M. J. Dunn, "Global positioning systems directorate systems engineering & integration interface specification," IS-GPS-200G, Tech. Rep., 2012, http://www.gps.gov/technical/icwg/IS-GPS-200G.pdf.

[31] M. Eichelberger, F. von Hagen, and R. Wattenhofer, "A Spoof-Proof GPS Receiver," in *19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Sydney, Australia*, 2020.

[32] G. Fang, J. Yi, X. Wan, Y. Liu, and H. Ke, "Experimental research of multistatic passive radar with a single antenna for drone detection," *IEEE Access*, 2018.

[33] J. Farrell and M. Barth, *The global positioning system and inertial navigation*, 1999.

[34] I. Fernández-Hernández, V. Rijmen, G. Seco-Granados, J. Simon, I. Rodríguez, and J. D. Calle, "A navigation message authentication proposal for the galileo open service," *NAVIGATION, Journal of the Institute of Navigation*, 2016.

[35] C. Fernandez-Prades, J. Arribas, P. Closas, C. Aviles, and L. Esteve, "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 2011)*, 2011.

[36] C. Fernández-Prades, J. Arribas, L. Esteve, D. Pubill, and P. Closas, "An open source Galileo E1 software receiver," in *2012 6th ESA Workshop on Satellite Navigation Technologies (Navitec 2012) & European Workshop on GNSS Signals and Signal Processing*, 2012.

[37] M. Foruhandeh, A. Z. Mohammed, G. Kildow, P. Berges, and R. Gerdes, "Spotr: GPS spoofing detection via device fingerprinting," in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2020.

[38] M. Goldstein, J. Kirschbaum *et al.*, "GPS disruptions: efforts to assess risks to critical infrastructure and coordinate agency actions should be enhanced." United States. Government Accountability Office, Tech. Rep., 2013.

[39] İ. Güvenç, O. Ozdemir, Y. Yapici, H. Mehrpouyan, and D. Matolak, "Detection, localization, and tracking of unauthorized uas and jammers," in *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, 2017.

[40] C. Hajiyev and H. E. Soken, "Robust adaptive Kalman filter for estimation of UAV dynamics in the presence of sensor/actuator faults," *Aerospace Science and Technology*, 2013.

[41] T. E. Humphreys, J. A. Bhatti, D. Shepard, and K. Wesson, "The Texas spoofing test battery: Toward a standard for evaluating GPS signal authentication techniques," in *Radionavigation Laboratory Conference Proceedings*, 2012.

[42] A. Jafarnia-Jahromi, T. Lin, A. Broumandan, J. Nielsen, and G. Lachapelle, "Detection and mitigation of spoofing attacks on a vector-based tracking GPS receiver," *Proc. ION ITM*, 2012.

[43] K. Jansen, M. Schäfer, D. Moser, V. Lenders, C. Pöpper, and J. Schmitt, "Crowd-GPS-Sec: Leveraging crowdsourcing to detect and localize GPS spoofing attacks," in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018.

[44] K. Jansen, N. O. Tippenhauer, and C. Pöpper, "Multi-receiver GPS spoofing detection: error models and realization," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, 2016.

[45] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *Journal of Field Robotics*, 2014.

[46] M. G. Kuhn, "An asymmetric security mechanism for navigation signals," in *International Workshop on Information Hiding*, 2004.

[47] B. M. Ledvina, W. J. Bencze, B. Galusha, and I. Miller, "An in-line anti-spoofing device for legacy civil GPS receivers," in *Proceedings of the 2010 international technical meeting of the Institute of Navigation*, 2001.

[48] S. C. Lo and P. K. Enge, "Authenticating aviation augmentation system broadcasts," in *IEEE/ION Position, Location and Navigation Symposium*, 2010.

[49] P. H. Madhani, P. Axelrad, K. Krumvieda, and J. Thomas, "Application of successive interference cancellation to the GPS pseudolite near-far problem," *IEEE Transactions on Aerospace and Electronic Systems*, 2003.

[50] G. Mao, S. Drake, and B. D. Anderson, "Design of an extended Kalman filter for UAV localization," in *2007 Information, Decision and Control*, 2007.

[51] C. E. McDowell, "GPS spoofer and repeater mitigation system using digital spatial nulling," 2007, uS Patent 7,250,903.

[52] E. McMilin, D. S. De Lorenzo, T. Lee, P. Enge *et al.*, "GPS anti-jam: A simple method of single antenna null-steering for aerial applications," in *Proceedings of the ION 2015 Pacific PNT Meeting*, 2015.

[53] M. Meurer, A. Konovaltsev, M. Appel, and M. Cuntz, "Direction-of-arrival assisted sequential spoofing detection and mitigation."

[54] P. Y. Montgomery, "Receiver-autonomous spoofing detection: Experimental results of a multi-antenna receiver defense against a portable civil GPS spoofer," in *Radionavigation Laboratory Conference Proceedings*, 2011.

[55] D. Moser, V. Lenders, and S. Capkun, "Digital radio signal cancellation attacks: An experimental evaluation," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, 2019.

[56] S. Narain, A. Ranganathan, and G. Noubir, "Security of GPS/INS based on-road location tracking systems," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.

[57] S. Nashimoto, D. Suzuki, T. Sugawara, and K. Sakiyama, "Sensor CONFusion: Defeating Kalman filter in signal injection attack," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 2018.

[58] P. P. Neumann and M. Bartholmai, "Real-time wind estimation on a micro unmanned aerial vehicle using its inertial measurement unit," *Sensors and Actuators A: Physical*, 2015.

[59] T. Nighswander, B. Ledvina, J. Diamond, R. Brumley, and D. Brumley, "GPS software attacks," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, 2012.

[60] J. Noh, Y. Kwon, Y. Son, H. Shin, D. Kim, J. Choi, and Y. Kim, "Tractor beam: Safe-hijacking of consumer drones with adaptive GPS spoofing," *ACM Transactions on Privacy and Security (TOPS)*, 2019.

[61] G. Oligeri, S. Sciancalepore, and R. Di Pietro, "GNSS spoofing detection via opportunistic IRIDIUM signals," in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2020.

[62] P. Papadimitratos and A. Jovanovic, "GNSS-based positioning: Attacks and countermeasures," in *MILCOM 2008-2008 IEEE Military Communications Conference*, 2008.

[63] P. Patel and J. Holtzman, "Analysis of a simple successive interference cancellation scheme in a DS/CDMA system," *IEEE journal on selected areas in communications*, 1994.

[64] M. L. Psiaki, B. W. O'Hanlon, J. A. Bhatti, D. P. Shepard, and T. E. Humphreys, "Civilian gps spoofing detection based on dualreceiver correlation of military signals," in *Radionavigation Laboratory Conference Proceedings*, 2011.

[65] A. Ranganathan, H. Ólafsdóttir, and S. Capkun, "SPREE: A spoofing resistant GPS receiver," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 2016.

[66] M. Rao and G. Falco, "How can pseudorange measurements be generated from code tracking," *Inside GNSS Mag*, 2012.

[67] P. A. Roncagliolo, J. G. García, and C. H. Muravchik, "Optimized carrier tracking loop design for real-time high-dynamics GNSS receivers," *International Journal of Navigation and Observation*, 2012.

[68] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[69] D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Drone hack: Spoofing attack demonstration on a civilian unmanned aerial vehicle," 2012.

[70] J.-M. Shin, Y.-S. Kim, T.-W. Ban, S. Choi, K.-M. Kang, and J.-Y. Ryu, "Position tracking techniques using multiple receivers for anti-drone systems," *Sensors*, 2021.

[71] A. Silverstein, "Electric Power Systems and GPS," *Civil GPS Service Interface Committee, North American Synchrophasor Initiative*, 2016.

[72] Ç. Tanıl, S. Khanafseh, M. Joerger, and B. Pervan, "Kalman filter-based INS monitor to detect GNSS spoofers capable of tracking aircraft position," in *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2016.

[73] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful GPS spoofing attacks," in *Proceedings of the 18th ACM conference on Computer and communications security*, 2011.

[74] D. H. Titterton and J. L. Weston, "Strapdown inertial navigation technology. 2nd," *London: Institution of Electrical Engineers*, 2004.

[75] J. S. Warner and R. G. Johnston, "GPS spoofing countermeasures," *Homeland Security Journal*, 2003.

[76] J. Wendel, O. Meister, C. Schlaile, and G. F. Trommer, "An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter," *Aerospace Science and Technology*, 2006.

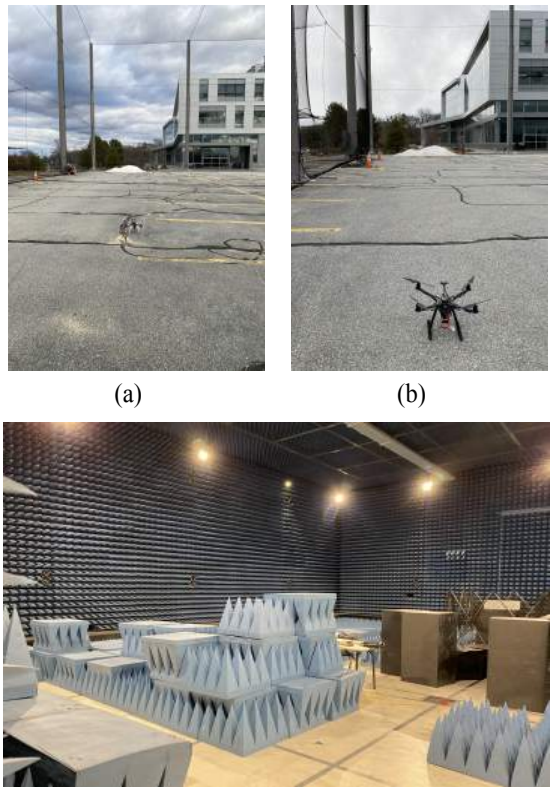
[77] K. Wesson, M. Rothlisberger, and T. Humphreys, "Practical cryptographic civil GPS signal authentication," *NAVIGATION: Journal of the Institute of Navigation*, 2012.

[78] K. D. Wesson, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "An evaluation of the vestigial signal defense for civil GPS anti-spoofing," in *Radionavigation Laboratory Conference Proceedings*, 2011.

[79] N. West, "Vector optimized library of kernels," *Internet: http://libvolk.org/,[Sep. 10, 2018]*, 2016.

[80] Z. Zhang, L. Zhou, and P. Tokekar, "Strategies to design signals to spoof Kalman filter," in *2018 Annual American Control Conference (ACC)*, 2018.

APPENDIX



(a) (b)

(c)

The drone testing and evaluation setup. (a) and (b) show the UAVs that we used (DJI Flamewheel F450 and Holybro S500) in the outdoor UAS testing facility. (c) shows the anechoic chamber used for GPS spoofing and jamming experiments.