Towards Secure Autonomous Aerial Vehicle Navigation

A dissertation presented in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

in the field of

Cybersecurity

by

Harshad Sathaye

Committee Members

Aanjhan Ranganathan, Northeastern University
Guevara Noubir, Northeastern University
Pau Closas, Northeastern University
Vincent Lenders, armasuisse S+T

Northeastern University
Khoury College of Computer Sciences
Boston, Massachusetts
April, 2023

Northeastern University
Khoury College of
Computer Sciences

PhD Thesis Approval

Thesis Title: Towards Secure Autonomous Aerial Vehicle Navigation

Author: Harshad Sathaye

PhD Program: ____ Computer Science ___✓___ Cybersecurity ____ Personal Health Informatics

PhD Thesis Approval to complete all degree requirements for the above PhD program.

Thesis Advisor	4/10/2023 *Date*
Thesis Advisor	4/10/2023 *Date*
Thesis Reader	4/10/2023 *Date*
Lenders Vincent NBMGX0 Digital unterschrieben von Lenders Vincent NBMGX0 Datum: 2023.04.10 16:20:58 +02'00' *Thesis Reader*	04/10/23 *Date*
Thesis Reader	*Date*

KHOURY COLLEGE APPROVAL:

Associate Dean for Graduate Programs	4/10/2023 *Date*

COPY RECEIVED BY GRADUATE STUDENT SERVICES:

Recipient's Signature	11 April 2023 *Date*

Distribution: Once completed, this form should be attached as page 2, immediately following the title page of the dissertation document. An electronic version of the document can then be uploaded to the Northeastern University-UMI Website.

I would like to express my sincere gratitude to my advisors, thesis committee, collaborators, friends, and family for their unwavering support, feedback, and encouragement throughout my academic journey.

Abstract

The modern aerial vehicle ecosystem relies heavily on various wireless communication and navigation technologies that are often unauthenticated and vulnerable to adversarial interference. This thesis evaluates the security of three critical components of automation systems in modern aerial vehicles: instrument landing system (ILS), aviation datalink applications like controller-pilot datalink communications (CPDLC), and satellite navigation systems such as global positioning system (GPS).

First, we demonstrate the feasibility of manipulating ILS instruments using commercially available software-defined radio (SDR) and a closed-loop ILS spoofer capable of manipulating spoofing signals based on the aircraft's position. In the second part, we propose a spoof-and-jam strategy to manipulate flight crew decision-making by implementing a reactive jammer for aviation datalink applications with a fast reaction time and high jamming success rate. The third part investigates the feasibility of controlling unmanned aerial vehicle (UAV) movements solely by GPS spoofing, highlighting the challenges of achieving a complete UAV takeover without causing crashes. We explore generic and UAV-specific strategies to control the UAV's speed and direction, successfully taking over consumer-grade UAVs from leading manufacturers. Finally, to address security issues surrounding satellite navigation, we design and implement SemperFi, a single antenna GPS receiver capable of tracking legitimate GPS signals and autonomously recovering from spoofing attacks. SemperFi leverages the extended Kalman filter (EKF) sensor-fusion mechanism built into most existing UAVs and a custom-designed legitimate signal retriever module to recover from attacks with high accuracy and a fast recovery time.

Overall, we address critical security concerns in modern aerial vehicles and pave the way for developing secure autonomous technologies by leveraging interconnected and tightly coupled sensors and individual systems.

Contents

List of Figures

List of Tables

List of Acronyms

AM amplitude modulation

ACARS aircraft communications, addressing, and reporting system

ADS-B automatic dependent surveillance-broadcast

ADS-C automatic dependent surveillance-contract

AMS ACARS message security

ARINC Aeronautical Radio Inc

ATC Air traffic controllers

ATN Aeronautical telecommunications network

ATSU air traffic services unit

CDI course deviation indicator

CSB carrier-plus-sidebands

CDA current data authority

CPDLC controller-pilot datalink communications

CRC cyclic redundancy checksum

DME distance-measuring equipment

DC direct current

DDM difference in the depth of modulation

D-ATIS digital automatic terminal information service

DY Doley-Yao

EKF extended Kalman filter

EICAS Engine Indication and Crew Alerting System

FLARM flight alarm

FAA Federal Aviation Administration

FMS flight management system

FANS future air navigation systems

GNSS global navigation satellite system

GPS global positioning system

HF high frequency

HMM hidden Markov model

IFR instrument flight rules

ICAO International Civil Aviation Organization

ILS instrument landing system

LDE Lateral deviation event

LRDE Level range deviation event

MSK minimum shift keying

NED north east down

NDB non-directional beacon

NDA next data authority

PID proportional–integral–derivative

PKI public-key infrastructure

PER packet encoding rules

PDC pre-departure clearance

PVT position velocity time

SATCOM satellite communication

SBO side bands only

SJR signal-to-jammer ratio

SDR software-defined radio

TCAS traffic collision avoidance system

UAV unmanned aerial vehicle

VOR very high frequency omnidirectional range

VRE Vertical rate change event

VHF very high frequency

WCE Waypoint change event

Chapter 1

Introduction

The aerial vehicles ecosystem is a complex and growing system encompassing many aircraft, technologies, regulations, and even humans! The aviation industry is at the core of this ecosystem and is responsible for transporting millions of passengers and tons of cargo daily. In the United States alone, the number of flights handled by the Federal Aviation Administration (FAA) was approximately 1.6 million in 2022, a number expected to grow in the coming years [35]. As the aerial vehicles ecosystem has grown, its complexity has also increased. The need to ensure safety and efficiency in the face of growing demands, new technologies and environmental challenges has driven this complexity. The aviation industry has responded by developing more advanced navigation, communication, and control systems, new automation technologies, and even new safety regulations and mandates.

Automation has been a major trend in recent years in the aviation industry, with many modern aircraft equipped with sophisticated autopilot systems that can interpret signals from conventional navigation systems and make on-the-fly decisions. These systems are designed to reduce the workload on pilots and improve safety by automating routine tasks and responding to unexpected events. For example, instrument landing system (ILS) enables the aircraft to land itself even in zero visibility conditions or technologies like automatic dependent surveillance-broadcast (ADS-B), controller-pilot datalink communications (CPDLC), and automatic dependent surveillance-contract (ADS-C) allow automatic surveillance and reporting traditionally performed manually over voice channels. Refer to Appendix A.1 for more details on automation systems in aerial vehicles. Automation has also given rise to a new class of unmanned aerial vehicle (UAV) capable of true human-independent operations. UAVs are aerial vehicles primarily controlled by sophisticated control systems that use a combination of sensors, such as global positioning system (GPS), cameras, and other instruments, to navigate and maintain altitude, speed, and direction. They come in various shapes and sizes, from small, handheld devices to large, fixed-wing aircraft capable

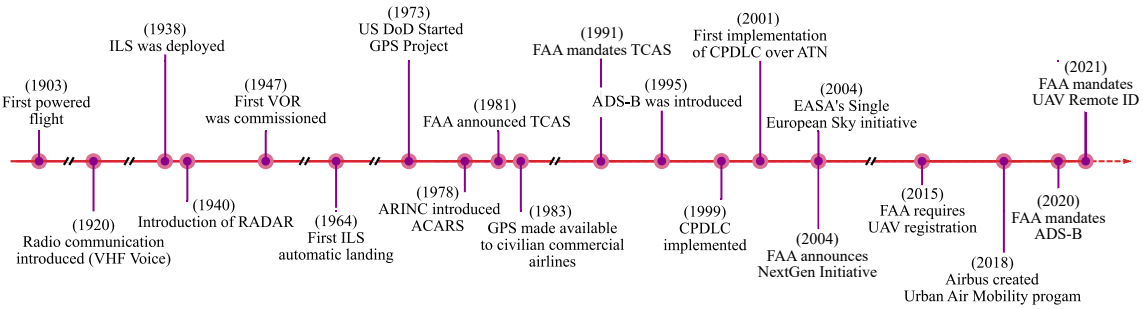


Figure 1.1: Developmental history and evolution of aerial vehicle systems and regulations

of carrying heavy payloads. The main advantage of UAVs is that they can be controlled remotely by humans or through automated flight plans and missions. UAVs are becoming an integral part of the aerial vehicles ecosystem as the industry constantly develops innovative applications that leverage micro aerial vehicles, a sub-class of UAVs, for commercial and recreational purposes. The use cases of UAVs are becoming increasingly diverse, with applications ranging from aerial photography and surveying to agriculture, delivery, and even search and rescue. For example, in 2020, UPS Flight Forward, a subsidiary of UPS, received the first-ever full Part 135 standard certification to operate a drone airline, allowing it to operate a drone delivery network in the United States [231]. The global UAV market size was valued at $14.18 billion in 2020 and is expected to reach $102.46 billion by 2030, growing at a CAGR of 19.6% from 2022 to 2030 [74]. According to the FAA, as of February 2023, there are about 872,248 registered drones and about 308,263 FAA-certified remote pilots in the United States alone [91].

Modern automation systems still rely on unauthenticated legacy communication and surveillance systems like ADS-B, CPDLC, and ADS-C and navigation systems like ILS and GPS. As evident from Figure 1.1, virtually every aerial vehicle relies on technology developed several decades ago. Due to the widespread availability of off-the-shelf radio equipment and open-source software, these systems are highly susceptible to wireless interference, which can compromise the automation systems that rely on them. Thus, there is a growing need to develop countermeasures to mitigate these risks and ensure the safe and responsible use of unmanned aerial systems. Overall, the aerial vehicles ecosystem is a constantly evolving, complex, and dynamic landscape. As new technologies and applications emerge, it is crucial to continue assessing and evaluating the threats and risks to different aerial vehicles, specifically autonomous systems, and develop strategies to ensure safe, secure, and efficient air travel. Simple design, lack of authentication, and availability of inexpensive off-the-shelf software-defined radio (SDR) make modern wireless systems

vulnerable to spoofing attacks.

Amongst the crucial technologies mentioned above, to the best of our knowledge, there exists a gap in the security evaluation of critical aviation datalink applications such as CPDLC and ADS-C and the precision approach system, ILS. Researchers in the past have extensively analyzed the security guarantees of aircraft surveillance technologies like ADS-B [224, 66, 205] and, to some extent, collision avoidance system [219]. Similarly, researchers have discovered profound vulnerabilities in GPS, which can be trivially exploited by a mere $100 worth of equipment. To some extent, they have demonstrated the ability to modify the course of ships [28], UAVs [212, 176, 141, 99] and self-driving cars [184] by transmitting malicious GPS signals.

Despite the above demonstrations and the rapidly growing importance of UAVs, there are only a limited number of studies on the feasibility of precisely controlling UAVs, specifically consumer UAVs, via GPS spoofing. Prior work was primarily focused on disrupting or altering the motion of the UAV in a non-specific direction or analyzing standalone GPS receivers. Importantly, no previous work in academic literature has examined and field-tested a controlled takeover of UAVs in a controlled real environment outside a simulator. This state of affairs severely limits the knowledge of the practicalities of GPS spoofing attacks on modern UAVs. Furthermore, even though GPS is at the core of modern automation systems, there is a lack of concrete countermeasures capable of autonomous recovery under sophisticated attacks like a seamless takeover [228].

Countermeasures that have been proposed so far rely either on cryptographic solutions or leverage physical-layer signal properties. Cryptographic countermeasures like [146, 240, 153, 65] prevent attackers from generating arbitrary false GPS signals. However, they are ineffective against signal replay attacks. Non-cryptographic countermeasures rely on detecting anomalies in the received GPS signal's physical characteristics, such as received signal strength [238], noise levels, direction or angle of arrival [165]. Some countermeasures can even detect stealthy seamless takeover attacks by detecting auxiliary signals [195]. A few countermeasures propose using additional sensors [135] and receivers [228, 166] to detect spoofing attacks. Most of the above schemes are limited to GPS spoofing attack *detection* and are incapable of autonomous legitimate location recovery. Moreover, existing mitigation techniques are ineffective against strong adversaries capable of completely overshadowing legitimate signals and stealthy attackers, e.g., seamless takeover of a victim's GPS location, despite having redundant fail-safe sensors [168]. This is evident in recent GPS spoofing incidents [105, 100] In summary, today's GPS receivers, specifically those implemented on UAVs, are incapable of uninterrupted operation during a spoofing attack.

There are two primary objectives of this thesis. First, to analyze the critical components of modern automation systems and shed light on the threat by answering three key ques-

tions: Can RF signals be spoofed to manipulate these instruments precisely? What are the requirements and limitations of such spoofing strategies? and how does manipulating a single component impact the entire vehicle? Second, to develop effective countermeasures that thwart most spoofing attacks by leveraging interconnected and tightly coupled sensors and individual systems. Thus ensuring minimal ecosystem changes.

1.1 Contributions

To achieve the stated objectives, first, we focus on evaluating the security guarantees of; i) ILS, the de-facto landing system used all over the world, ii) Aviation datalink applications such as CPDLC, and ADS-C, and iii) the use of GPS in autonomous navigation of modern unmanned aerial vehicles (UAVs). Determine the extent to which an attacker can exploit an autonomous vehicle using GPS and study the feasibility and requirements of such attacks. Second, based on the insights gained, we explore the possibilities of leveraging redundant systems and developing a countermeasure against GPS spoofing that can detect and recover from various signal spoofing attacks. Specifically, the contributions are as follows:

[Chapter 2] - Security Analysis of Instrument Landing System: To evaluate the impact of manipulating critical navigation automation components of landing systems, we analyze over-the-air ILS signals, show the fundamental design issue that makes the system vulnerable to spoofing attacks, and develop proof of concept attacks to manipulate ILS instruments. Specifically, we demonstrate the feasibility of spoofing ILS radio signals in real-time, causing last-minute "go around" decisions and even missing the landing zone in low-visibility scenarios. We evaluate the complete attack using a closed-loop ILS spoofer and systematically assess its performance against an FAA-certifiable flight simulator. Our results demonstrate a systematic success rate with offset touchdowns ranging from 18 to 50 meters.

[Chapter 3] - On the Implications of Spoofing and Jamming Aviation Datalink Applications: Like ILS, aviation datalink applications are vital in navigation automation, specifically route management and flight planning. We examine the security of aviation datalink applications critical to air traffic control, such as CPDLC and ADS-C. These applications are used to transmit important messages such as flight plans, altitude changes, and radio frequency assignments and can influence flight crew's decision-making. We demonstrate that signal spoofing and jamming attacks can be used to manipulate the flight crew's decisions, potentially leading to collisions and near-misses. We identify 48 vulnerable regions where an attacker has a 90% chance of encountering favorable conditions for coordinated

multi-aircraft attacks and implement a reactive jammer with a reaction time of 1.48 ms and 98.85% jamming success. We also discuss the possibility of integrated attacks and propose countermeasures.

[Chapter 4] - An Experimental Study of GPS Spoofing and Takeover Attacks on UAVs: To address the gap in understanding the practicalities of GPS spoofing attacks on modern UAVs, we assess the impact of GPS spoofing on navigation automation in consumer UAVs and evaluate their behavior under spoofing attacks. We design and implement a Real-time GPS spoofer that generates arbitrary trajectories in real-time for live over-the-air experiments. Through such a setup, we determine the feasibility and requirements for precise control of UAVs via GPS spoofing. We extract generic and UAV-specific strategies for complete maneuvering control and introduce a human-in-the-loop GPS spoofer that enables manual control of UAVs via GPS spoofing, including the execution of patterns such as 180-degree turns.

[Chapter 5] - SemperFi: Anti-spoofing GPS Receiver for UAVs: Given the vulnerabilities of GPS and its importance in automation, there is a dire need to develop robust countermeasures. As pointed out earlier, currently proposed countermeasures can only detect attacks or have partial mitigation capability, and most techniques are limited to naive attacks. To address these concerns, we present SemperFi, a GPS receiver capable of autonomous recovery and uninterrupted legitimate location output even in an adversarial setting. SemperFi utilizes the extended Kalman Filter-based GPS failsafe mechanism integrated into most UAVs. We implement custom-designed adversarial signal identification and legitimate signal retriever modules. These modules perform active spoofing verification and successive interference cancellation to detect and recover from GPS spoofing attacks, including a seamless-takeover. We implement SemperFi in GNSS-SDR, an open-source software-defined GNSS receiver, and evaluate its performance using UAV simulators, real drones, real-world GPS datasets, and embedded platforms. With these measures in place, SemperFi is able to achieve legitimate location recovery within 0.54 seconds and with an accuracy of up to 20 meters.

1.2 Publications

Parts of this thesis are based on the following peer-reviewed publications.

- Sathaye, H., Schepers, D., Ranganathan, A., & Noubir, G. (2019). **DEMO: Wireless Attacks on Aircraft Landing Systems**. *In Proceedings of the 12th ACM Conference*

on Security and Privacy in Wireless and Mobile Networks (WiSec '19)
Best Demo Award

- Sathaye, H., Schepers, D., Ranganathan, A., & Noubir, G. (2019). **Wireless Attacks on Aircraft Instrument Landing Systems**. *In 28th USENIX Security Symposium (USENIX Security '19)*

- Sathaye, H., & Ranganathan, A. (2019). **POSTER: SemperFi: A Spoofer Eliminating Standalone GPS Receiver**. *In Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '20)*
Best Poster Award

- Sathaye, H., Noubir, G., & Ranganathan, A. (2022). **On the Implications of Spoofing and Jamming Aviation Datalink Applications**. *In Proceedings of the 38th Annual Computer Security Applications Conference (ACSAC '22)*
Distinguished Paper Award Honorable Mention

- Sathaye, H., Strohmeier, M., Lenders, V., & Ranganathan, A. (2022). **An Experimental Study of GPS Spoofing and Takeover Attacks on UAVs**. *In 31st USENIX Security Symposium (USENIX Security '22)*

- Sathaye, H., LaMountain, G., Closas, P., & Ranganathan, A. (2022). **SemperFi: Anti-spoofing GPS receiver for UAVs**. *In Network and Distributed Systems Security Symposium (NDSS '22)*

In this process, I have co-authored the following publication.

- Motallebighomi, M., Singh, M., Sathaye, H., & Ranganathan, A. (2023). **Location-independent GNSS Relay Attacks: A Lazy Attacker's Guide to Breaking Navigation Message Authentication** (under review) *In Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '23)*

1.3 Outline

The thesis document is organized as follows. First, we describe our security analysis of the most widely used landing system (ILS) in Chapter 2. Next, in Chapter 3, we present a security evaluation and the implications of spoofing and jamming aviation datalink applications like ACARS, CPDLC, and ADS-C. This is followed by an experimental evaluation of UAV takeover strategies via GPS spoofing in Chapter 4. This work demonstrates the possibility of exerting post-takeover speed and direction control via GPS spoofing. Based

on the insights gained, in Chapter 5, we present our design and implementation of SemperFi, a single-antenna GPS receiver for UAVs that is capable of detecting and mitigating several GPS spoofing attacks. Finally, in Chapter 6, we conclude this thesis by stating the importance of leveraging interconnected systems and the need for testing integrated attack scenarios that synchronously target several vital systems. Thus paving the way for future research with the objective of securing autonomous aerial vehicle navigation.

Chapter 2

Security Analysis of Instrument Landing System

2.1 Introduction

Today, the aviation industry is experiencing significant growth in air traffic with more than 5400 flights [35] in the sky at any given time. It has become typical for air traffic control towers to handle more than a thousand takeoffs and landings every day. For example, Atlanta's Hartsfield-Jackson International airport handles around 2500 takeoffs and landings every day. Boston's Logan airport which is not one of the busiest airports in the world, managed an average of 1100 flights every day in August 2018. The modern aviation ecosystem heavily relies on a plethora of wireless technologies for safe and efficient operation. For instance, air traffic controllers verbally communicate with the pilots over the VHF (30 to 300 MHz) radio frequency channels. The airplanes continuously broadcast their position, velocity, callsigns, altitude, etc., using the automatic dependent surveillance-broadcast (ADS-B) wireless communication protocol. Primary and secondary surveillance radars enable aircraft localization and provide relevant target information to air traffic controllers. Traffic Alert and Collision Avoidance System (TCAS), an airborne wireless system independent of the air traffic controller, enables the aircraft to detect potential collisions and alert the pilots. Air traffic information, flight information, and other operational control messages between the aircraft and ground stations are transferred using the Aircraft Communications Addressing and Reporting System (ACARS), which uses the very high frequency (VHF) and high frequency (HF) radio frequency channels for communication. Similarly, many radio navigation aids such as GPS, very high frequency omnidirectional range (VOR), nondirectional beacon (NDB), distance-measuring equipment (DME), and instrument landing system (ILS) play crucial roles during different phases of an airplane's flight.

Many studies have already demonstrated that several of the above-mentioned aviation systems are vulnerable to attacks. For example, researchers [66] injected non-existing aircraft in the sky by merely spoofing ADS-B messages. Some other attacks [157] modified the route of an airplane by jamming and replacing the ADS-B signals of specific victim aircraft. ACARS, the data link communications system between aircraft and ground stations, was found to leak a significant amount of private data [218], e.g., passenger information, medical data, and sometimes even credit card details were transferred. GPS, one of the essential navigation aids, is also vulnerable to signal spoofing attacks [139]. Furthermore, an attacker can spoof TCAS messages [186], creating false resolution advisories and forcing the pilot to initiate avoidance maneuvers. Given the dependence on wireless technologies, the threats described above are real and show the importance of building secure aviation control, communication, and navigation systems.

One of the most critical phases of an airplane's flight plan is the final approach or landing phase, as the plane descends toward the ground, actively maneuvered by the pilot. For example, 59% of the fatal accidents documented by Boeing [50] occurred during descent, approach, and landing. Several technologies and systems, such as GPS, VOR, DME assist the pilot in landing the aircraft safely. The Instrument Landing System (ILS) [32] is today the de-facto approach system used by planes at most airports as it is the most precise system capable of providing accurate horizontal and vertical guidance. At Boston's Logan International Airport, 413,899 [2] flight plans were filed in 2022. Of these 413,899 flight plans, 95% were instrument flight rules (IFR) plans. Instrument flight rules are a set of instructions established by the FAA to govern flight under conditions in which flying by visual reference is either unsafe or just not allowed. Also, several European airports [26] prohibit aircraft from landing using visual flight rules at night. ILS incorporates radio technology to provide all-weather guidance to pilots, ensuring safe travel and any interference can be catastrophic.

In September 2018, the pilots of Air India flight AI-101 reported an ILS malfunction and were forced to do an emergency landing. Even worse, TCAS, ACARS, and most other systems that aid a smooth landing were unusable. Furthermore, NASA's Aviation Safety Reporting System [104] indicates over 300 ILS-related incidents where pilots reported the erratic behavior of the localizer and glideslope–two critical components of ILS. ILS also plays a significant role in autoland systems capable of landing aircraft even in the most adverse conditions without manual interference. Autoland systems have significantly advanced over the years since its first deployment in De Havilland's DH121 Trident, the first airliner to be fitted with an Autoland system [31]. However, several near-catastrophic events [19, 25, 29] have been reported due to the failure or erratic behavior of these autoland systems, with ILS interference being one of the principal causes. With increasing reliance

on auto-pilot systems and the widespread availability of low-cost software-defined radio hardware platforms, adversarial wireless interference to critical infrastructure systems such as the ILS cannot be ruled out.

In this first work on security guarantees of ILS, we demonstrate two types of wireless attacks: i) Overshadow and ii) single-tone attacks, and show that ILS is vulnerable to signal spoofing attacks. For both attacks, we generate specially crafted radio signals similar to the legitimate ILS signals using low-cost software-defined radio hardware platforms and successfully forced aviation-grade ILS receivers to lock and display arbitrary alignment to both horizontal and vertical approach paths. This means that an attacker with sufficient knowledge can precisely control the approach path of an aircraft without alerting the pilots, especially during low-visibility conditions. In the best case, such an attack can potentially result in several aborted landings causing air traffic disruption, and in the worst case aircraft will overshoot the landing zone or miss the runway entirely. Single-tone attack is a particular type of attack that significantly reduces the power and bandwidth requirements for the attacker. For example, overshadow attacks require the attacker to transmit two frequencies simultaneously. However, in the single-tone attack, it is sufficient for the attacker to transmit a single frequency tone that is loosely synchronized with the legitimate signal. We discuss potential countermeasures, including failsafe systems such as GPS, and show that these systems also do not provide sufficient security guarantees. We highlight that implementing cryptographic authentication on ILS signals is not enough, as the system would still be vulnerable to record and replay attacks. Therefore, this research highlights an open research challenge of building secure, scalable, and efficient aircraft landing systems.

2.2 Approach Systems in Aviation

Approach systems enable pilots to land airplanes even in extreme weather conditions. They are classified into non-precision and precision approach systems based on the accuracy and type of approach guidance provided to an aircraft. Non-precision approach systems provide only horizontal or lateral guidance (heading/bearing). Examples of non-precision approach systems are VOR [233], NDB [232], and satellite systems such as GPS. With the development of precision approach systems, non-precision approach systems such as VOR and NDB have significantly decreased today. Precision approach systems provide horizontal (heading/bearing) and vertical (glide path) guidance to an approaching aircraft. The Instrument Landing System (ILS) is today's most commonly deployed precision approach system. Other examples of precision approach systems include the Microwave Landing System (MLS), Transponder Landing System (TLS), Ground Based Augmentation Landing System (GLS), and Joint Precision Approach and Landing System (JPALS). It is important

to note that these alternate landing systems still use existing ILS concepts and equipment mostly in scenarios where ILS is unavailable. For example, TLS enables precision landing guidance in places where the terrain is uneven, and the ILS signal reflections off the ground cause undesirable needle deflections by *emulating* the ILS signals using only one base tower (in contrast to two for ILS) whose placement allows more flexibility. However, TLS still leverages the same fundamental concepts of ILS. In short, ILS plays a key, de-facto role in providing precision landing guidance at most airports today. It is essential to evaluate its resilience to modern-day cyber-physical attacks.

2.2.1 Instrument Landing System (ILS)

The first fully operational ILS was deployed in 1932 at the Berlin Tempelhof Central Airport, Germany. ILS enables the pilot to align the aircraft with the centerline of the runway and maintain a safe descent rate. ITU defines ILS [133] as "a radio navigation system which provides aircraft with horizontal and vertical guidance just before and during landing and at certain fixed points, indicates the distance to the reference point of landing". Autopilot systems on some modern aircraft [214] use ILS signals to execute a fully autonomous approach and landing, especially in low visibility settings. ILS (Figure 2.1) comprises three independent subsystems: i) localizer, (ii) glideslope, and iii) marker beacons. The localizer and the glideslope guide the aircraft in the horizontal and vertical plane, respectively. The marker beacons act as checkpoints that enable the pilot to determine the aircraft's distance to the runway. ILS has three operational categories: i) CAT I, ii) CAT II, and iii) CAT III. CAT III further has three sub-standards IIIa, IIIb, and IIIc. These operational categories are decided based on ILS installations at the airport [1]. They are independent of the receiver on the aircraft. With the advent of GPS and other localization technologies, marker beacons are less important today and increasingly obsolete. However, the localizer and the glideslope play a major role in an aircraft's safe landing today and are expected to remain so for many years.

ILS Signal Generation

ILS signals are generated and transmitted such that the waves form a specific radio frequency signal pattern in space to create guidance information related to horizontal and vertical positioning. ILS signal generators leverage *space modulation* i.e., use multiple antennas to transmit amplitude-modulated radio frequency signals with various powers and phases. The transmitted signals combine in the airspace to form signals with difference in the depth of modulation (DDM) at various points within the 3D airspace. Each DDM value

[1] Procedures for the Evaluation and Approval of Facilities for Special Authorization Category I Operations and All Category II and III Operations `http://fsims.faa.gov/wdocs/Orders/8400_13.htm`

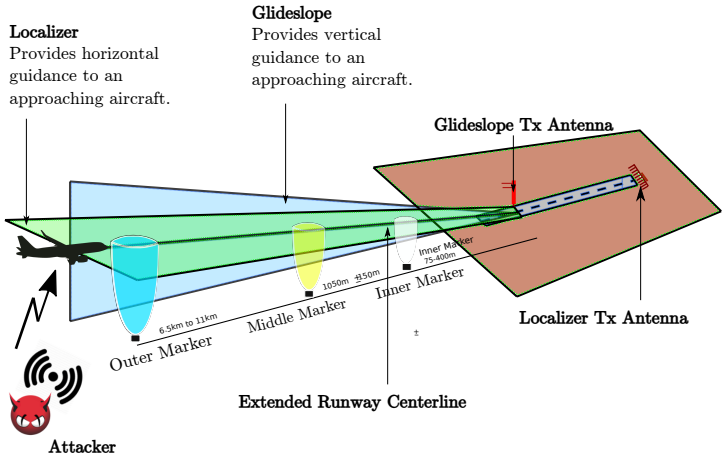


Figure 2.1: The ILS consists of three subsystems: i) Localizer, ii) glideslope, and (iii) marker beacons.

directly indicates a specific deviation of the aircraft from the correct touchdown position. For example, the signals combine in space to produce a signal with zero DDM along the runway's center line. It is important to note that, unlike traditional modulation techniques where the modulation occurs within the modulating hardware, in space modulation, the signals mix within the airspace.

The process of generating the localizer and glideslope signals (Figure 2.2) are similar, with differences mainly in the carrier frequency used and how they are combined in space to provide the relevant guidance information. The carrier signal is amplitude modulated with 90 Hz and 150 Hz tones to a certain modulation depth. The depth of modulation or modulation index measures the extent of amplitude variation about an un-modulated carrier. The depth of modulation is set at 20% and 40%, respectively, for localizer and glideslope signals. The outputs of the 90 Hz and the 150 Hz modulators are combined to yield two radio frequency signals: a carrier-plus-sidebands (CSB) and a side bands only (SBO) signal. The names of the signal directly reflect their spectral energy configuration, with the CSB containing both the sideband energy and the assigned carrier frequency, while in the SBO signal, the carrier frequency component is suppressed. The CSB and SBO signals are subjected to specific phase shifts before transmission. The phase shifts are carefully chosen such that when the CSB and SBO signals combine in space, the resulting signal enables the aircraft to determine its horizontal and vertical alignment with the approach path.

Localizer. The localizer subsystem consists of an array of multiple antennas that emit the CSB and SBO signals such that the 150 Hz modulation predominates to the right of

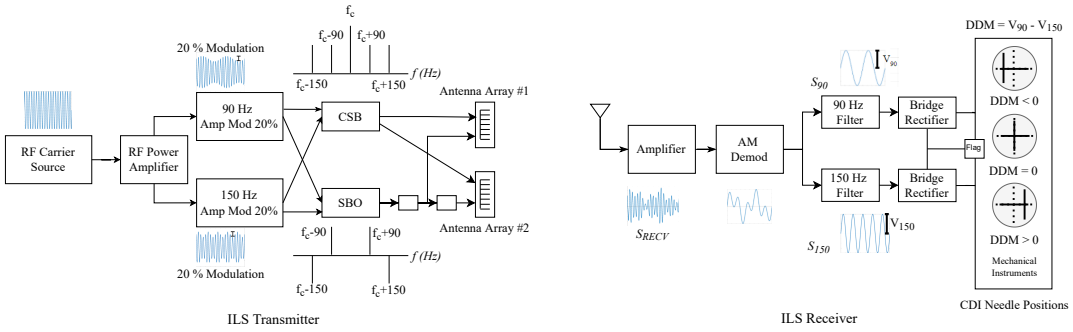


Figure 2.2: Block diagram of ILS transmitter and receiver describing the process of generation and reception of ILS signal along with waveforms at each stage.

the runway centerline and the 90 Hz signal prevails to the left. In other words, if the flight is aligned to the right of the runway during the approach, the 150 Hz dominant signal will indicate the pilot to steer left and vice versa. The antenna array of the localizer is located at the opposite end (from the approach side) of the runway. Each runway operates its localizer at a specific carrier frequency (between 108.1 MHz to 111.95 MHz), and the ILS receiver automatically tunes to this frequency as soon as the pilot inputs the runway identifier in the cockpit receiver module. Additionally, the runway identifier is transmitted using a 1020 Hz morse code signal over the localizer's carrier frequency.

Glideslope. The glideslope subsystem uses two antennas to create a signal pattern similar to that of the localizer except on a vertical plane. The two antennas are mounted on a tower at specific heights defined by the glide-path angle suitable for that particular airport's runway. In contrast to the localizer, the glideslope produces the signal pattern in the airspace based on the sum of the signals received from each antenna via the direct line-of-sight path and the reflected path. Mixing the CSB and SBO signals results in a pattern in which the 90 Hz component of the signal predominates in the region above the glide path while the 150 Hz prevails below the glide path. The glideslope uses carrier frequencies between 329.15 MHz and 335.0 MHz, and the antenna tower is located near the touchdown zone of the runway. Typically, the center of the glide slope defines a glide path angle of approximately 3°. The corresponding glideslope frequency is hardcoded for every localizer frequency, i.e., the localizer-glideslope frequencies occur in pairs and the instrument automatically tunes to the right glideslope frequency when the pilot tunes to a specific runway's localizer frequency.

ILS Receiver

The combined signals received at the aircraft are amplified, demodulated, and filtered to recover the 90 Hz and 150 Hz components. A bridge rectifier is used to convert the amplitude of the recovered tones to direct current (DC) voltage levels. The DC voltage output is directly proportional to the depth of the modulation of the 90 Hz and 150 Hz tones–a direct measure of the dominating frequency signal. The DC voltage causes the course deviation indicator (CDI) needle to deflect based on the difference in the depth of the modulation of the two tones, indicating the aircraft's lateral and vertical deviation from the approach path.

For example, an aircraft perfectly aligned on the centerline will receive 90 and 150 Hz signals with the same amplitude, i.e., equal depth of modulation, and will result in zero DDM and therefore cause no needle deflections. However, an aircraft that is off-course and not aligned with the approach path will receive signals with a non-zero difference in the depth of modulation resulting in a corresponding deflection of the needle. The instruments are calibrated to show full-scale deflection if $DDM > 0.155$ or $DDM < -0.155$ for localizer and if $DDM > 0.175$ or $DDM < -0.175$ for glideslope [61]. These values correspond to 2.5° offset on the left side of the runway, 2.5° offset on the right side of the runway, 0.7° offset above the glide path angle and 0.7° below the glide path angle respectively.

2.2.2 Typical Approach Sequence

Pilots use aeronautical charts containing vital information about the terrain, available facilities, and usage guidelines throughout a flight. Approach plates are a type of navigation chart used for flying based on instrument readings. Every pilot is required to abide by the routes and rules defined in an approach plate unless ordered otherwise by the air traffic controller. The approach plate contains information like the active localizer frequency of the runway, the runway identifier in Morse code, glideslope interception altitude, ATC tower frequencies, and other information crucial for a safe landing.

Once the pilot receives the clearance to land at an assigned runway, the pilot enters the localizer frequency associated with the designated runway and enters the course of the runway into the auto-pilot. Note that the localizer and glideslope frequencies occur in pairs; therefore, the pilot does not have to enter the corresponding glideslope frequency manually. The CDI needle is displayed on the cockpit when the pilot intercepts the localizer. The pilot then verifies whether the receiver is tuned to the right localizer by confirming the runway identifier transmitted as morse code on the localizer frequency. For example, for landing on runway 4R (Runway Ident - IBOS) at Logan International Airport, Boston, the pilot will tune to 110.3 MHz and will verify this by confirming the Morse code: .. / --...

/ --- / ... Based on the aircraft's deviation from the runway and the approach angle, the indicator will guide the pilot to maneuver the aircraft appropriately. Modern autopilot systems are capable of receiving inputs from ILS receivers and autonomously landing the aircraft without human intervention.

In fact, pilots are trained and instructed to trust the instruments more than their intuition. The pilots will fly right if the instruments ask them to fly right. This is true specifically when flying in weather conditions that force the pilots to follow the instruments. Detecting and recovering from any instrument failures during crucial landing procedures is one of the toughest challenges in modern aviation. Given the heavy reliance on ILS and instruments in general, malfunctions and adversarial interference can be catastrophic, especially in autonomous approaches and flights. In this work, we demonstrate the vulnerabilities of ILS and further raise awareness towards the challenges of building secure aircraft landing systems.

2.3 Wireless Attacks on ILS

We demonstrate two types of wireless attacks: i) Overshadow attack and ii) Single-tone attack. In the overshadow attack, the attacker transmits pre-crafted ILS signals of higher signal strength; thus overpowering the legitimate ILS signals. The single-tone attack is a special attack where it is sufficient for the attacker to transmit a single frequency tone signal at a specific signal strength (lower than the legitimate ILS signal strength) to interfere and control the deflections of the CDI needle.

Attacker model. We make the following assumptions regarding the attacker. Given that the technical details of ILS are in the public domain, we assume that the attacker has complete knowledge of the physical characteristics of ILS signals e.g., frequencies, modulation index etc. We also assume that the attacker is capable of transmitting these radio frequency signals over the air. The widespread availability of low-cost (less than a few hundred dollars) software-defined radio platforms has put radio transmitters and receivers in the hands of the masses. Although not a necessary condition, in the case of single-tone, the knowledge of the flight's approach path, the airplane's manufacturer, and the model will allow the attacker to significantly optimize their attack signal. We do not restrict the location of the attacker and discuss the pros and cons of both an onboard attacker as well as an attacker on the ground.

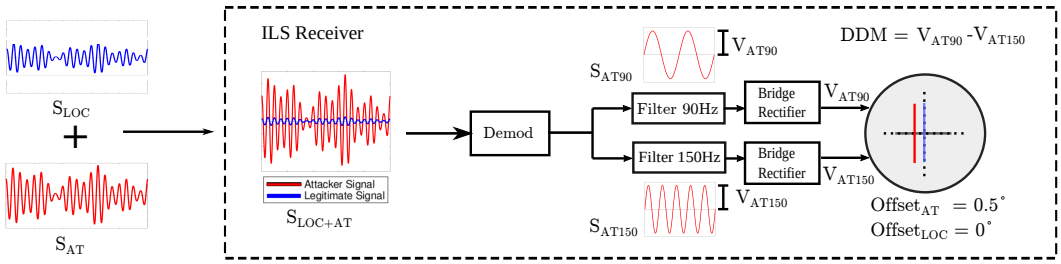


Figure 2.3: The attacker's signal has a preset DDM corresponding to 0.5° to the right of the runway. The attacker's signal overshadows the legitimate signal. The blue line represents the needle position without attack.

2.3.1 Overshadow attack

The overshadow attack is when the attacker transmits specially crafted ILS signals at a power level such that the legitimate signals get overpowered by the attacker's signal at the receiver. The main reason such an attack works is that the receivers "lock" and process only the strongest received signal. Figure 2.3 shows how the attacker's fake ILS signal completely overshadows the legitimate ILS signal resulting in the deflection of the CDI needle. We note that the attacker signal can be specially crafted to force the CDI needle to indicate a specific offset as demonstrated in Section 2.4.2.

Attack Signal Generation. Recall that the ILS receiver on board receives a mix of the transmitted CSB and SBO signals that contain the 90 and 150 Hz tones (Figure 2.2). The amplitude of received 90 and 150 Hz tones depends on the position of the aircraft relative to the runway and its approach path angle. For example, as shown in Figure 2.4, the 90 Hz tone will dominate if the aircraft is offset to the left of the runway and the 150 Hz dominates to the right. Similarly, for glideslope, the 90 Hz tone dominates glide angles steeper than the recommended angle, and the 150 Hz tone dominates otherwise. Both 90 and 150 Hz will have equal amplitudes for a perfectly aligned approach. Therefore, to execute an overshadow attack, it is sufficient to generate signals similar to the received legitimate ILS signals and transmit at a much higher power as compared to legitimate ILS signals. In other words, the attacker need not generate CSB and SBO signals separately; instead can directly transmit the combined signal with appropriate amplitude differences between the 90 and 150 Hz tones. The amplitude differences are calculated based on the offset the attacker intends to introduce at the aircraft. The attacker's signal (Figure 2.5) is generated as follows. There are two tone generators for generating the 90 and 150 Hz signals. Configuring each tone's amplitude is important to construct signals with a preset difference in the modulation depth corresponding to the required deviation to spoof. The tones are then added and amplitude modulated using the runway's specific localizer or glideslope frequency. Recall that the

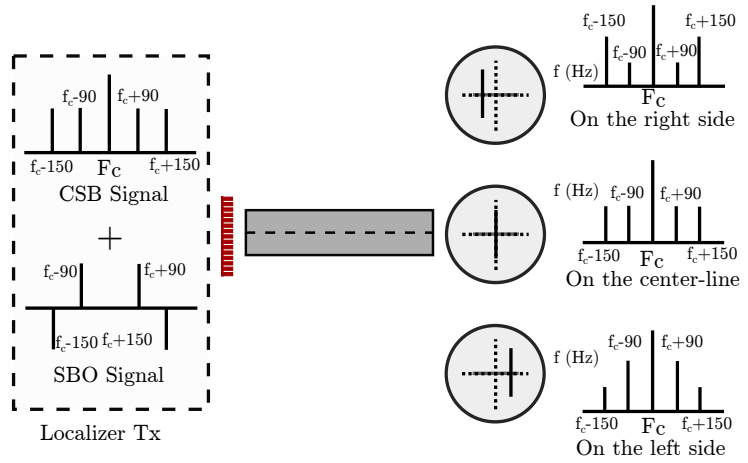


Figure 2.4: Frequency domain representation of the received signal showing the amplitudes of the sidebands as observed at various lateral offsets

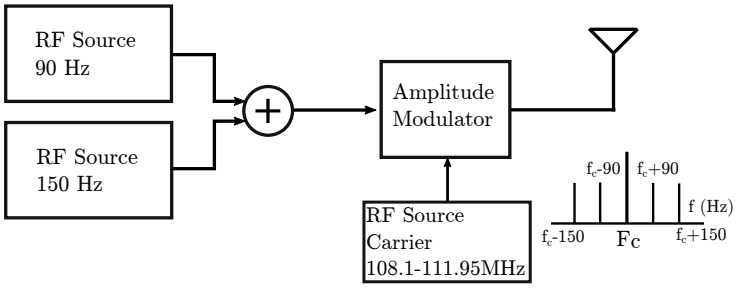


Figure 2.5: Signal generator used for generating the required attack signal with specific amplitudes of the 90 Hz and 150 Hz components

amplitude differences i.e., difference in depth of modulation (DDM) between the two tones, directly correspond to the required offset to spoof. In the absence of the adversarial signals, the estimated $DDM = V_{LOC90} - V_{LOC150}$. In the presence of the attacker's spoofing signals, the estimated $DDM = [V_{LOC90} + V_{AT90}] - [V_{LOC150} + V_{AT150}]$. Since $V_{AT90} >> V_{LOC90}$ and $V_{AT150} >> V_{LOC150}$, the resulting $DDM = V_{AT90} - V_{AT150}$. Thus by manipulating the amplitude *differences* between the transmitted 90 Hz and 150 Hz tones, the attacker can acquire precise control of the aircraft's CDI and the aircraft's approach path itself.

2.3.2 Single-tone attack

The single-tone attack leverages the properties of space modulation and depending on the intended CDI needle deflection, the attacker either performs a signal cancellation or signal amplification by transmitting only one of the sideband tones (either the 90 Hz or the

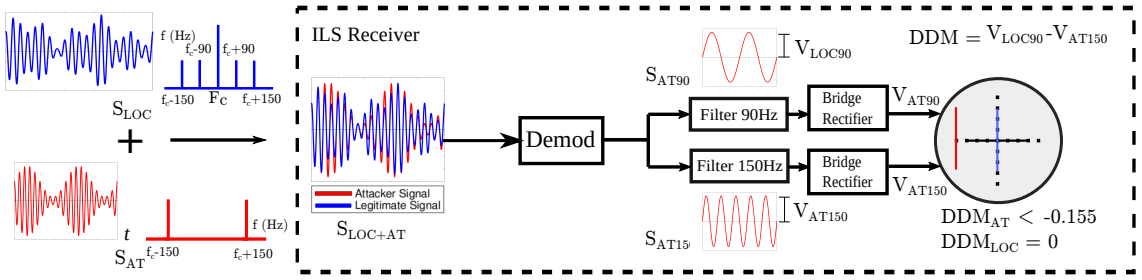


Figure 2.6: Schematic of the single-tone attack. The attacker constructs a DSB-SC signal without the 90 Hz component and the carrier. The blue line represents the needle position without the attack

150 Hz). In contrast to the overshadow attack, single-tone attack does not require high-powered spoofing signals. Recall that the aircraft's horizontal and vertical offset is estimated based on the difference in the depth of the modulation of the 90 Hz and the 150 Hz tones. As indicated in Figure 2.4, either frequency tone dominates depending on the offset. In the case of an overshadow attack, the spoofing signal was constructed with all the necessary frequency components. However, in the single-tone attack, the attacker aims to interfere with only one of the two sideband frequencies, directly affecting the estimated offset.

Attack Signal Generation. The working of the single-tone attack is shown in Figure 2.6. The legitimate localizer signal's spectrum contains the carrier and both the sideband tones of 90 Hz and 150 Hz. As described previously, the amplitudes of the sideband tones depend on the true offset of the aircraft. In a single-tone attack, the attacker generates only one of the two sideband tones i.e., $f_c \pm 90$ or $f_c \pm 150$ with appropriate amplitude levels depending on the spoofing offset (e.g., left or right off the runway) introduced at the aircraft. For example, consider the scenario where the attacker intends to force the aircraft to land on the left of the runway with an offset of 0.5°. The legitimate difference in depth of modulation will be zero as the aircraft is centered over the runway. To cause the aircraft to go left, the attacker must transmit signals that will spoof the current offset to be at the right side of the runway. As shown in Figure 2.4, the 150 Hz component dominates on the right side of the runway approach, and therefore, the attacker needs to transmit the $f_c \pm 150$ signal with an appropriate amplitude to force the aircraft to turn left. For the specific example of 0.5° offset, the amplitude of the $f_c \pm 150$ component should be such that the difference in the depth of modulation equals 0.03 [61].

Notice that the single-tone attack signal is similar to a double-sideband suppressed-carrier signal which is well-known to be spectrally more efficient than the normal amplitude modulation signal. Specifically, it is possible for the attacker to reduce the required power

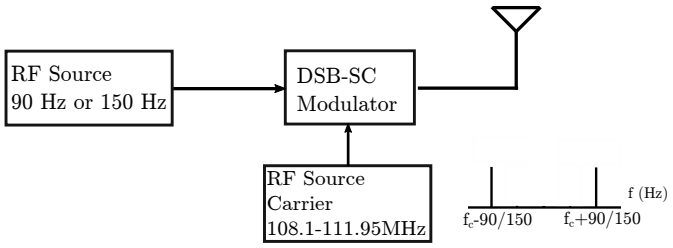


Figure 2.7: Single-tone attack signal generator with a DSB-SC modulator

to almost 50% of the overshadow attack as there is no need to transmit the carrier signal and one of the sideband signals. One of the important limitations of the single-tone is the effect of the attacker's synchronization with the legitimate signal. To precisely control the spoofing offset, the attacker needs to coarsely control the spoofing signal such that the phase difference between the attacker and the legitimate signals remains constant throughout the attack. We evaluate and show in Section 2.4.3 the effect of phase synchronization on this attack. Additionally, the spectral efficiency of the single-tone attack can be exploited to execute a low-power last-minute denial of service on the ILS system. This is specifically dangerous while an aircraft is executing an auto-pilot-assisted approach. The block diagram of the single-tone attack signal generator is shown in Figure 2.7.

2.4 Implementation and Evaluation of Attacks

In this section, we demonstrate the feasibility and evaluate the effectiveness of the attack with the help of both simulations and actual experiments conducted using commercial aviation-grade receivers and an advanced flight simulator qualified for FAA certification.

2.4.1 Experimental Setup

Our experimental setup is shown in Figure 2.8 and Figure 2.9. The setup consists of four main components: i) X-Plane 11 flight simulator, ii) attacker control unit, iii) software-defined radio hardware platforms (USRP B210s), and iv) commercial aviation-grade hand-held navigation receiver. We use X-Plane 11 flight simulator to test the effects of spoofing attacks on the ILS. X-Plane is a professional flight simulator capable of simulating several commercial, military, and other aircraft. X-Plane can also simulate various visibility conditions and implements advanced aerodynamic models to predict an aircraft's performance in abnormal conditions. It is important to note that X-Plane qualifies for FAA-certified flight training hours when used with computer systems that meet the FAA's minimum frame rate requirements. The certified versions of the software are used in numerous pilot training

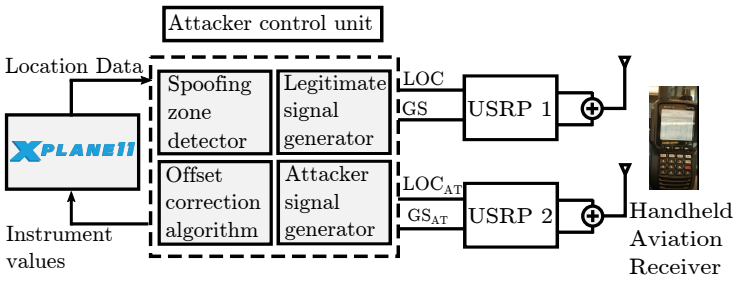


Figure 2.8: Schematic of the experimental setup used for evaluating the attacks on ILS. The attacker control unit interfaces with the simulator and USRP B210s. A flight yoke and throttle system is connected to the machine running X-Plane flight simulator software. The attacker control unit interfaces with the flight simulator over a UDP/IP network.

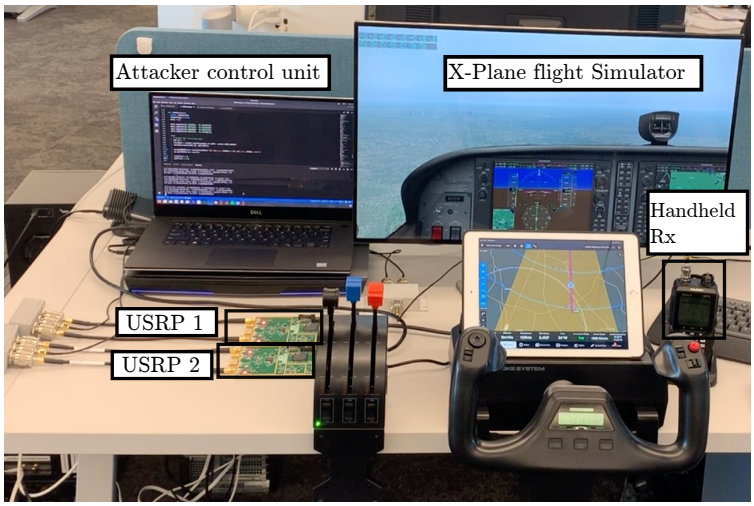


Figure 2.9: Photo of the experiment setup.

schools. X-Plane allows interaction with the simulator and instruments through a variety of mobile apps and UDP/IP networks. This feature allowed us to manipulate the instrument readings for evaluating our ILS attacks. Additionally, X-Plane has autopilot and AI-based autoland features which we leverage in our experiments. In other words, X-Plane contains all the features and flexibility to evaluate our proposed attacks in a close-to real-world setting. The second component of our setup is the attacker control unit module which takes the location of the aircraft as input from X-Plane and generates signals for the attack. The module is also responsible for manipulating X-Plane's instrument panel based on the effect of the spoofing signal on the receiver. The attacker control unit module is a laptop running Ubuntu and contains four submodules: spoofing zone detector, offset correction algorithm, legitimate signal generator, and attacker signal generator. The spoofing zone detector iden-

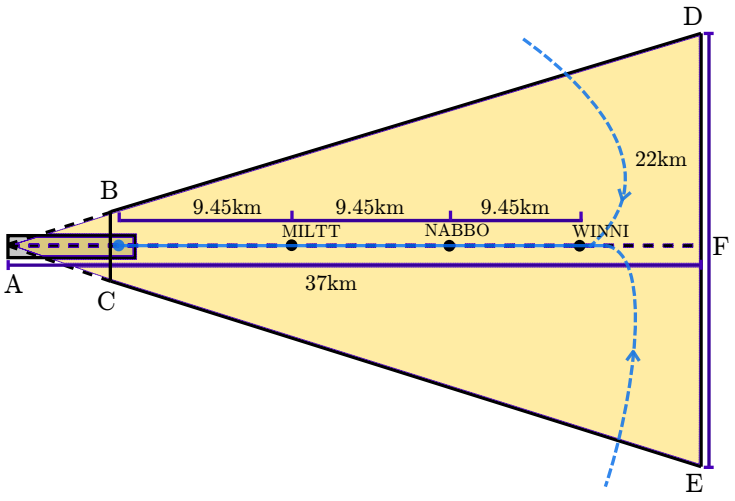


Figure 2.10: The spoofing zone is defined by points B, C, D, and E. WINNI, NABBO, and MILTT are the waypoints for the final approach as published for a mid-sized airport. The spoofing zone has a wide aperture as the air-traffic controller can vector in the aircraft onto the final approach in multiple ways.

tifies whether an aircraft is entering its first waypoint of the final approach and triggers the start of spoofing. The spoofing zone detector plays an important role in the timely starting of the spoofing attack so as to prevent any abrupt changes in the instrument panel and therefore avoid suspicion. The offset correction algorithm uses the current location of the aircraft to continuously correct its spoofing signals taking into consideration the aircraft's corrective actions. Note that the location data received from X-Plane can be analogous to receiving the location data through ADS-B signals [134] in the real world. The output of the offset correction algorithm is used to generate fake ILS signals. We also generate legitimate signals to evaluate the effect of the overshadow and the single-tone attacks. We use two USRP B210s [6], one each for transmitting legitimate ILS signals and attacker signals. We conducted the experiments in both wired and wireless settings. For the experiments conducted in wireless settings, the receiver was placed 2 m from the transmitter. Northeastern University has access to a Department of Homeland Security laboratory which provides RF shielding, thus preventing signal leakage. This is necessary as transmitting ILS signals over the air is illegal. We use two different ILS receivers, a Yaesu FTA-750L [27] and a Sporty's SP-400 Handheld NAV/COM Aviation Receiver [16], to evaluate the attacks.

Spoofing Zone Detection

The spoofing zone detection algorithm enables automated and timely triggering of the spoofing signal. One of the key requirements of the zone detector is to trigger the spoofing signals without causing any abrupt changes to the instrument readings, thereby avoiding detection

by the pilots. The spoofing region is shaped like a triangle following the coverage of the localizer and glideslope signals. For example, the localizer covers 17.5° on either side of the extended runway centerline and extends for about 35 km beyond the touchdown zone. Figure 2.10 shows the zone measurements. The attacker signals are triggered when the aircraft approaches the shaded region. The shaded region is decided based on the final approach patterns for a specific runway. We used an even-odd algorithm for detecting the presence of the aircraft within this spoofing zone [110]. The even-odd algorithm is extensively used in graphics software for region detection and has low computational overhead. Absolute locations cannot be used as aircraft enter the final approach path in many different ways based on their arrival direction and Air traffic controllers (ATC) instructions. The attacker automatically starts transmitting the signals as soon as the aircraft enters the spoofing region from the sides and the needle is yet to be centered. This prevents any sudden noticeable jumps, thus allowing a seamless takeover.

Offset correction algorithm

The attacker's signals are pre-crafted to cause the aircraft to land with a specific offset without being detected. The pilot or the autopilot system will perform course correction maneuvers to align with the runway centerline based on the instrument readings. At this point, the instruments will continuously indicate the spoofed offset irrespective of the aircraft's location and maneuvers raising suspicion of an instrument failure. To prevent this, we developed a real-time offset correction and signal generation algorithm that crafts the spoofing signals based on the aircraft's current location in real-time. The attacker can use the GPS coordinates if present inside the aircraft or leverage the ADS-B packets containing location information on the ground. We explain the offset correction algorithm using Figure 2.11. Consider an aircraft at point B, cleared to land and entering the spoofing zone. The air-traffic controller instructs the aircraft to intercept point C on the extended runway centerline. Assuming that the attacker's spoofing signal contains a pre-crafted offset to the left of the runway forcing the aircraft to follow path DA instead of CA. The offset correction module computes the current offset of the aircraft with respect to the centerline and subtracts the current offset from the spoofed offset to estimate the desired change in the course. Thus, the correction Δ required to be introduced is the difference between the required offset angle $\angle DAC$ and the current offset angle $\angle BAC$. Note that offsets to the left of the centerline are considered negative offsets and offsets to the right are considered positive offsets. The current offset θ can be estimated using $\theta = tan^{-}1[(m_{CA} - m_{BA})/(1 + m_{BA} * m_{CA})]$, where m is the slope. m_{CA} is typically hardcoded and is specific for each runway. m_{BA} can be estimated using the longitude and latitudes of the touchdown point and the aircraft's current location. Now, the correction Δ is converted to the respective difference in depth of

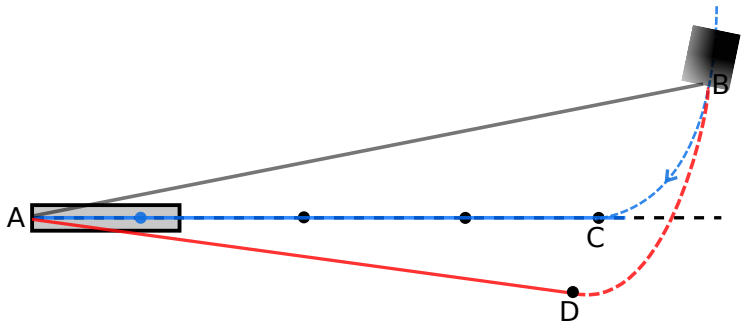


Figure 2.11: Offset correction algorithm takes into account the aircraft's current position to calculate the difference in the spoofed offset and the current offset.

modulation value using the formula $DDM = (DDM_{fullscale} * \Delta)/2.5$, where 2.5 is the angle that results in full-scale deviation and $DDM_{fullscale}$ is the difference in depth of modulation that causes full-scale deviation. The amplitude of the individual 90 and 150 Hz components is estimated using the formula $0.2 + (DDM/2)$ and sent to the signal generator module which then transmits the required signal. Note that the value 0.2 comes from the legitimate signal's depth of modulation. The algorithm was implemented on a laptop running Ubuntu and took less than 5 ms on average to compute the offsets. The complete algorithm is shown in Algorithm 1.

Algorithm 1 Offset correction algorithm.

procedure GETANGLEDIFFERENCE
    $\angle DAC \leftarrow TargetedLocalizerOffset$
    $\angle BAC \leftarrow GetAngle(location)$
    $difference \leftarrow \angle DAC - \angle BAC$
    return $difference$
procedure CALCULATEDDM
    $difference \leftarrow GetAngleDifference$
    $ddm \leftarrow (0.155 * difference)/2.5$
    $AT90 \leftarrow 0.2 + (ddm)/2$
    $AT150 \leftarrow 0.2 - (ddm)/2$
    $ChangeAmplitude(AT90, AT150)$

Setup Validation

We verified the working of our experimental setup as follows. First, we ensure consistency between the CDI needle displayed on the flight simulator and the handheld receiver. To this extent, we disabled the attacker signal and output only the legitimate signal to the

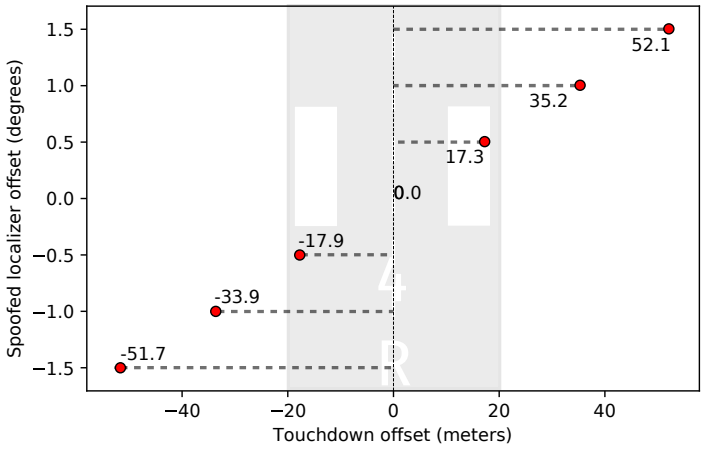


Figure 2.12: Results of localizer spoofing. 5 automated landings per spoofed localizer offset were executed and the touchdown offset in meters from the runway centerline was recorded.

handheld receiver based on the aircraft's location obtained from X-Plane. We manually validated that the alignment shown on the handheld receiver is the same as that of the flight simulator throughout the final approach. The uploaded attack demonstration video [2] also contains this validation for reference. We conducted the same experiment over the air in a controlled environment and verified consistency between the handheld receiver and the flight simulator cockpit. Second, we test our offset correction algorithm by maneuvering (swaying) the aircraft during its final approach. During this experiment, the offset correction algorithm should account for the maneuvers and generate corresponding ILS signals to the handheld receiver. We ensure the correctness of the algorithm by validating the consistency between the handheld receiver's CDI needle and the flight simulator cockpit. Note that we do not update the flight simulator's instrument readings for this experiment and the readings displayed in the simulator cockpit are only because of the simulator software engine. Finally, we validate the spoofing zone detector algorithm by entering the final approach from various directions and checking the trigger for beginning the spoofing attack. We are now ready to perform our attack evaluations.

2.4.2 Evaluation of Overshadow Attack

We evaluate the effectiveness of overshadow attack as follows. We leverage the autopilot and autoland feature of X-Plane to analyze the attack's effects avoiding any inconsistency that might arise due to human error. We configured X-Plane to land on the runway of a midsized airport in the US. This configuration is analogous to the pilot following approach instructions from the air-traffic controller. As soon as the aircraft entered the spoofing

[2]Video demonstration of the attack https://youtu.be/Wp4CpyxYJq4

zone, the spoofing signals were transmitted along with the legitimate signals. The spoofing signals were generated to fake various vertical and horizontal offsets. Note that the spoofing signals were generated in real-time based on the current position of the aircraft. For the localizer (horizontal offset), spoofing signals corresponding to 0.5, 1.0, and 1.5° offset on both sides of the runway were generated. The spoofing glideslope angles were between 2.8° and 3.3°. For each spoofing angle and offset, we performed five automated landings and the results are shown in Figure 2.12 and Figure 2.13. Throughout the attack, we continuously monitored the path of the aircraft using Foreflight [3], a popular app used both by aviation enthusiasts and commercial pilots as well as X-Plane's own interfaces. We did not observe any abrupt changes in the readings and observed a smooth takeover. The aircraft landed with an 18 m offset from the runway centerline for a spoofing offset of just 0.5°. Note that this is already close to the edge of the runway and potentially go undetected by both the air-traffic controllers as well as pilots onboard, especially in low visibility conditions. In the case of glideslope, a shift in the glide path angle by 0.1° i.e., 2.9° glide path angle instead of the recommended 3°, caused the aircraft to land almost 800 m beyond the safe touchdown zone of the runway. We have uploaded a video demonstration of the attack for reference (`https://youtu.be/Wp4CpyxYJq4`).

2.4.3 Evaluation of Single-tone Attack

We evaluate the effectiveness and feasibility of the proposed single-tone attack using the experimental setup described in Section 2.4.1. Recall that in the single-tone attack, the attacker transmits only one of the sideband tones (either the $f_c \pm 90$ or the $f_c \pm 150$ Hz) to causing deflections in the course deviation indicator needle. We implemented the attack by configuring one of the USRPs (attacker) to transmit the sideband signals and observed its effect on the handheld navigation receiver. We observed that the spoofing signal caused the needle to deflect to the configured offset. However, the needle was not as stable as in the overshadow attack and displayed minor oscillations. This is because the specific attack is sensitive to carrier phase oscillations and therefore must be accounted for to avoid detection. A significant advantage of this attack is the power required to cause needle deflections as the attacker only transmits one of the sideband components without the carrier. This gives an almost 50% increase in power efficiency and therefore can even act as a low-power last-minute denial of service attack in case the attacker is unable to establish full synchronization with the legitimate signal. In the following sections, we evaluate the effect of phase synchronization on the single-tone attack and develop a real-time amplitude scaling algorithm that can counter the phase oscillations.

[3]Advanced Flight Planner https://www.foreflight.com

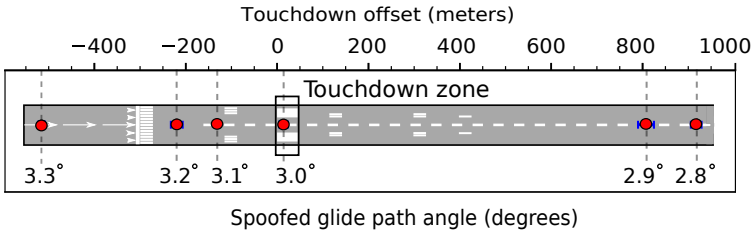


Figure 2.13: Results of glideslope spoofing. 5 automated landings per spoofed glideslope angle offset were executed and the touchdown offset in meters beyond the touchdown zone was recorded.

Effect of Phase Synchronization

Recall that the single-tone attack signal is similar to a conventional double-sideband suppressed-carrier (DSB-SC) signal. It is well known that one of the drawbacks of a DSB-SC communication system is the complexity of recovering the carrier signal during demodulation. If the carrier signal used at the receiver is not synchronized with the carrier wave used in the generation of the DSB-SC signal, the demodulated signal will be distorted. In the scenario of the single-tone attack, this distortion can potentially result in changes in the difference in the depth of modulation estimates causing the needle to oscillate. We simulated the effect of phase synchronization on the single-tone attack effectiveness and present our results in Figure 2.14 and Figure 2.15. We generated the single-tone attack signal to cause full-scale deviation i.e., $\gtrsim 2.5°$ for localizer and $\gtrsim 2.5°$ for the glideslope while perfectly in sync with the legitimate carrier signal. We observe that the phase difference causes the resultant offset to change. We also noted an uncertainty region around the 90° and 270° phase difference region. This is due to the dependency in a DSB-SC system [106] between the carrier phase difference ϕ and the resulting distortion at the output which is directly proportional to the $cos\phi$. Therefore, at angles around 90° and 270°, there is an uncertainty region for the resulting offset. However, in our experiments on the handheld receiver, we noticed that although the needle oscillated, it was not as pronounced as the simulation results indicate. One of the reasons is the rate at which the sensor measurements are being calculated and displayed on the screen. Additionally, the aircraft is in motion, therefore, causing the phase differences to cycle more rapidly than the display's refresh rate. A knowledgeable attacker can potentially leverage these properties to generate controlled spoofing signals and succeed with optimized transmission power.

Real-time Amplitude Scaling

In the following, we propose and evaluate a strategy to counter the effect of phase synchronization on the single-tone attack. The phase differences cause the output to be distorted.

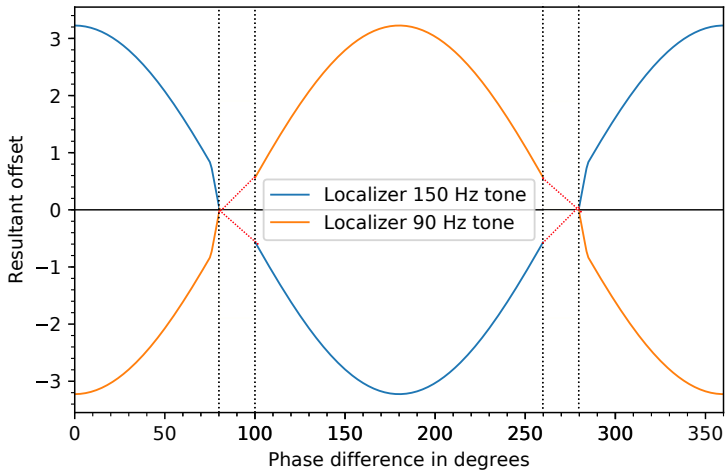


Figure 2.14: Comparison of calculated offset and the phase difference for localizer

Besides the uncertainty region around the 90° and 270°, it is possible to predict the phase given sufficient knowledge such as aircraft speed, current location, and antenna positions. We assume such a motivated attacker for the single-tone attack evaluation in this section. It is also well known that tightly controlling the phase of a signal is not trivial and therefore our algorithm proposes to manipulate the amplitude of the attacker signal instead of the phase. Changing the amplitude of the attacker signal will compensate for the effect of phase on the signal at the receiver and we call this the "real-time amplitude scaling" algorithm. The algorithm itself is inspired by prior works on amplitude scaling for DSB-SC systems [106]. We use the distance between the transmitter and the receiver to estimate the received phase of the signal by measuring complete and incomplete wave cycles. In the simulation, we then create an ILS signal with the necessary phase shift. We also create the attacker's signal and add it to the legitimate signal to estimate the DDM. This allows us to assess the impact of phase on the transmitted signal and use this information to calculate the amplitude that will be required to counter the effects of phase. For example, if the predicted phase offset is zero, then to spoof a certain offset, the attacker needs to reduce the amplitude of its signal. We present the results of our amplitude scaling experiment in Figure 2.16 and Figure 2.17.

2.4.4 Comparison of Power Requirements

One of the major advantages of the single-tone attack is the improvement over the power required to execute the attack, given sufficient knowledge and environmental conditions. In this section, we evaluate and compare the power requirements of the overshadow and the single-tone attacks. We note that the absolute power profiles are specific for the handheld

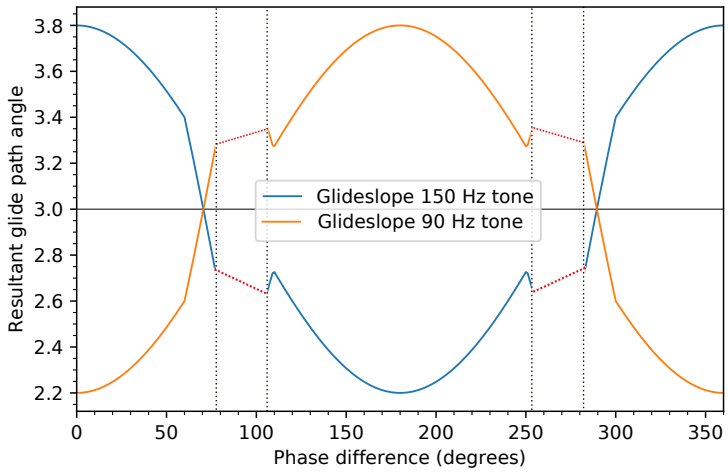


Figure 2.15: Comparison of calculated offset and the phase difference for glideslope

receivers used in the experiments. The goal of the power comparison is to verify whether there is an improvement in the attacker's required transmission power. We present our results in Figure 2.18 and Figure 2.19. Our evaluations show the required signal strength to successfully cause 0.5° and 0.1° deviation in localizer and glideslope, respectively. The received signal strength profile is shown in blue and acts as a reference for the attacker based on which the attacker can compute its required power to transmit the spoofing signals. We experimented by transmitting the signals to the handheld receiver and observing the attack's success (needle indicating the intended offset). The values result from over 400 trials with 95% confidence interval, and we find that, on average, the difference in power required reaches close to 20.53 dB and 27.47 dB for the localizer and the glideslope, respectively. Thus, given sufficient knowledge of the scenario, a motivated attacker can execute the single-tone attack successfully and with less power than the overshadow attack. As described previously, we acknowledge that the single-tone attack has drawbacks. However, given the low power requirements, an attacker can exploit the single-tone attack to cause a low-power denial of service attack. Such an attack can be disastrous, especially in an aircraft's final moments before landing.

2.5 Discussion

Receiving antenna characteristics and location of the attacker. The receiver hardware and its characteristics[4] vary depending on the type of aircraft it is mounted on. For

[4] `https://www.easa.europa.eu/certification-specifications/cs-23-normal-utility-aerobatic-and-commuter-aeroplanes`

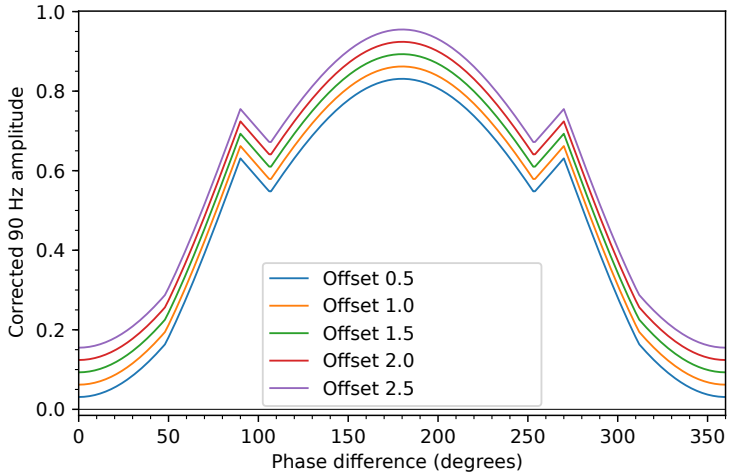


Figure 2.16: Amplitude scaling algorithm evaluation localizer. Amplitude required to compensate for the effect of phase

example, Cessna aircraft have their ILS antennas on the tail-fin or the vertical stabilizer. We note that the same antenna is typically used for a number of systems such as VOR, ILS, and DME; each signal arriving from a different direction. For commercial aircraft, the antennas are typically located on the nose of the plane with a forward-looking single broad lobe receiving beam pattern. In certain large aircraft, specifically those capable of landing with high nose attitude, the antennas are located either on the underside or on the landing gear of the aircraft itself [5]. The antenna equipment onboard plays an important role in determining the optimum location of the attacker to execute the attack. The ideal location of an on-ground attacker is at a point along the centerline of the runway that falls within the receiving lobe of the onboard antennas. Attackers inside the plane will have to deal with signal attenuation caused by the body of the aircraft itself and position the spoofing signal transmitter accordingly. A thorough investigation is required to fully understand the implications and feasibility of an on-board attacker and we intend to pursue the experiments as future work. The location of the attacker plays a more significant role in the scenario of the single-tone attacker since the attacker has to carefully predict the phase and accordingly manipulate the amplitude of the spoofing signal. The problem of identifying optimum locations for the attack is an open problem very similar to the group spoofing problem [228] proposed as a countermeasure for GPS spoofing attacks. In our context, the attacker has to identify locations on the ground such that the phase difference between the legitimate signal and the spoofing signal remains constant along the line of approach. Recall that in the single-tone attack, the offset indicated by the cockpit is sensitive to phase changes.

[5] https://www.casa.gov.au/sites/g/files/net351/f/_assets/main/pilots/download/ils.pdf

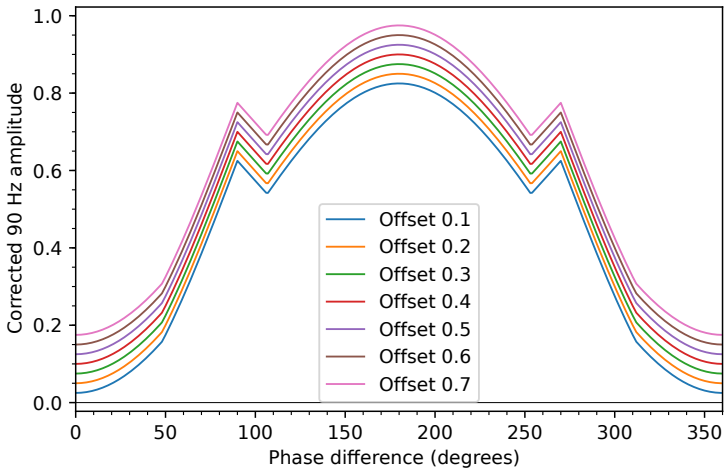


Figure 2.17: Amplitude scaling algorithm evaluation glideslope. Amplitude required to compensate for the effect of phase

Therefore locations that allow constant phase differences can result in a fixed spoofing offset and, therefore, minimal oscillations in the readings.

ILS Categories. The main advantage of ILS is that the pilot need not have visuals of the runway during the final approach as the ILS system is intended to guide the aircraft to a safe landing. The ILS categories are classified based on the maximum decision height at which a missed approach must be initiated if the pilot does not have a visual reference to continue the approach. In CAT I, the decision height is 60 m above the ground i.e., if the pilot does not have a visual reference at this height, a missed approach or go-around must be initiated. The decision height for CAT III is as low as 15 m above the ground. The demonstrated attacks can cause severe consequences in CAT III systems due to the low decision height. It might potentially be too late to execute a missed approach in case of an attack. The consequences of the attack on CAT I and CAT II systems are less catastrophic. However, they can still cause major air traffic disruptions. Note that CAT I approach is mostly used by smaller flights. Commercial flights typically fly a CAT II or CAT III approach.

Alternative technologies and potential countermeasures. Many navigation technologies, such as HF Omnidirectional Range, Non-directional Beacons, Distance Measurement Equipment, and GPS, provide guidance to the pilot during the different phases of an aircraft's flight. All the mentioned navigation aids use unauthenticated wireless signals and are, therefore, vulnerable to some form of a spoofing attack. Furthermore, it is worth mentioning that only ILS and GPS can provide precision guidance during the final approach.

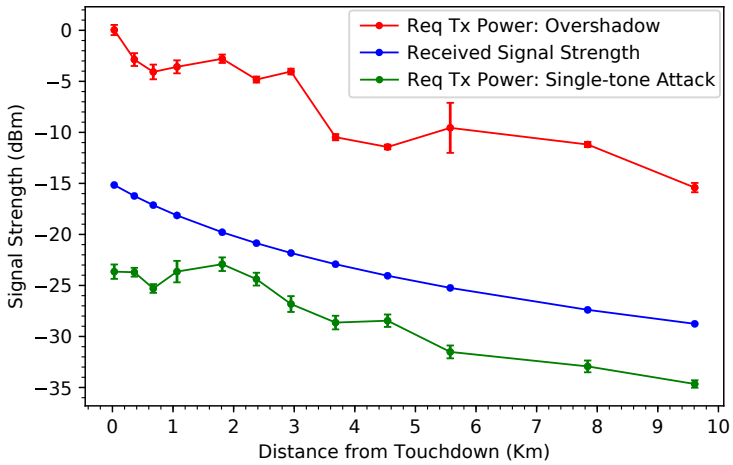


Figure 2.18: Comparison of required RSS (dB) for attack methodologies for the localizer

Also, ILS is the only technology today that provides both lateral and vertical approach guidance and is suitable for CAT III ILS approaches.

Most security issues faced by aviation technologies like ADS-B, ACARS and TCAS can be fixed by implementing cryptographic solutions [218] [223]. However, cryptographic solutions are not sufficient to prevent localization attacks. For example, cryptographically securing GPS signals [142, 92] similar to military navigation can only prevent spoofing attacks to an extent. It would still be possible for an attacker to relay the GPS signals with appropriate timing delays and succeeds in a GPS location or time spoofing attack. One can derive inspiration from existing literature on mitigating GPS spoofing attacks [195, 228, 137, 148, 144, 136] and build similar systems that are deployed at the receiver end. An alternative is implementing a wide-area secure localization system based on distance bounding [56] and secure proximity verification techniques [194]. However, this would require bidirectional communication and warrant further investigation with respect to scalability, deployability etc.

Experiment Limitations. Our experimental setup described in Section 2.4 was carefully constructed in consultation with aviation experts. Since we use an FAA-accredited flight simulator, we sent our configuration files and scripts to a licensed pilot for them to perform final approaches using the instruments and give us feedback. We were mainly concerned about whether there was any other indicator on the cockpit that raises suspicion about the attack. We conducted our attack evaluations in both wired and controlled wireless settings. Note that it is illegal to transmit ILS signals over the air in a public space. Effects due to the aircraft's motion such as Doppler shift do not affect the attacker signal as

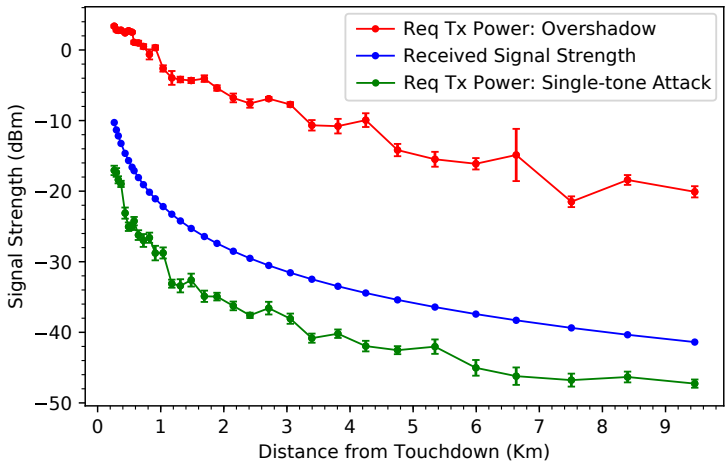


Figure 2.19: Comparison of required RSS (dB) for attack methodologies for glideslope

these are receiver end problems and the receiver hardware already accounts for such effects for the legitimate signal. Note that the attacker closely imitates the legitimate signals in frequency and amplitude. In short, we made the best effort to replicate a real-world approach. However, our setup has its limitations. We did not perform the experiments on a real aircraft which would give us more insights into the effects of aircraft's construction, antenna placements, cockpit display sensitivity, etc. One of the factors that will get affected is the power required by the attacker. Note that commercial ILS transmitters use a 25 watts transmitter for localizer signals and a 5 W power for the glideslope signals. To put things in perspective, a standard 12 V 10 Ah battery can power a 24 Watts amplifier for about 5 hours.

2.6 Related Work

Over the years, the aviation industry has largely invested and succeeded in making flying safer. Security was never considered by design as historically the ability to transmit and receive wireless signals required considerable resources and knowledge. However, the widespread availability of robust and low-cost software-defined radio platforms has altered the threat landscape. In fact, today the majority of wireless systems employed in modern aviation have been shown to be vulnerable to some form of cyber-physical attacks. In this section, we will briefly describe the various attacks demonstrated in prior work. Strohmeier et al. in [224] provides a comprehensive analysis of the vulnerabilities and attacks against the various wireless technologies that modern aviation depends on. Voice communication over VHF is primarily used to transfer information between the air traffic controller and the

aircraft. There have already been incidents [220] related to spoofed VHF communications and several efforts [88] to design a secure radio communication system. Primary surveillance radars have been shown to be vulnerable to signal jamming attacks [171]. Secondary surveillance radars [21] leverage the ability of the aircraft to respond to ground-based interrogations for aircraft localization. Due to the unauthenticated nature of these messages, it is possible for an attacker to use publicly available implementations for software-defined radio platforms to modify, inject and jam messages creating a false picture of the airspace. Such attacks were even demonstrated to be low-power, targeted, and stealthy against sophisticated wireless systems such as Wi-Fi [236], and WPA-Enterprise [62]. The ADS-B protocol used by aircraft to transmit key information such as position, velocity, and any emergency codes also faces the same challenges of active and passive attacks due to the unauthenticated nature of the signals. Several works have repeatedly demonstrated the vulnerabilities of ADS-B signals [201, 223, 47, 226, 66, 245, 222, 24, 205]. ACARS [20], the data link communications system between aircraft and ground stations was found to leak a significant amount of private data [227, 218, 149] e.g., passenger information, medical data, and sometimes even credit card details were transferred. Furthermore, an attacker can spoof TCAS messages [187] creating false resolution advisories and forcing the pilot to initiate avoidance maneuvers.

For navigation, the aviation industry relies on a number of systems such as ILS, GPS, VOR, and DME. Although the use of VOR and DME is rapidly decreasing, ILS and GPS will be in use for a very long time and are the only technologies available today for enabling autonomous landing. It is also well established that GPS is vulnerable to signal spoofing attacks [139, 28, 175, 228, 30, 246, 169]. Researchers have also demonstrated [189, 193] the feasibility of signal manipulation in the context of data communication systems. However, there has been no prior work on the security guarantees of ILS, and this work is in that direction. It is important to note that although many of the security issues in the aviation industry can be fixed by implementing some sort of cryptographic authentication, they are ineffective against the ILS attacks demonstrated in this chapter.

2.7 Conclusion

In this work, we presented the first security evaluation of an aircraft instrument landing system against wireless attacks. Through both simulations and experiments using aviation-grade commercial ILS receivers and FAA-recommended flight simulators, we showed that an attacker could precisely control the approach path of an aircraft without alerting the pilots, especially during low-visibility conditions. We discussed potential countermeasures, including failsafe systems such as GPS. We showed that these systems do not provide

sufficient security guarantees, and there are unique challenges to realizing a scalable and secure aircraft landing system.

Chapter 3

On the Implications of Spoofing and Jamming Aviation Datalink Applications

3.1 Introduction

Air traffic controllers (ATC) ensure the safe and efficient flow of air traffic on the ground and during flight. ATCs frequently provide instructions and receive reports from the aircraft present in their airspace. In 2021, air traffic controllers around the world handled more than 19 million flights [113] [1]. Traditionally, all instructions and reporting took place over voice communication channels. However, with the increasing air traffic density, voice communication channels are becoming a bottleneck. For example, according to Boeing [2], it can take up to 20 to 45 minutes to make a position report through the voice channel, i.e., waiting for one's turn to interact with the ATC in congested airspace directly impacting the flow of air traffic. Modifying previous instructions and noisy radio frequency channels further forces the aircraft to operate at sub-optimal altitudes and speeds. In fact, such drawbacks of voice communication have led to several accidents [183].

As a result, there are several ongoing efforts (e.g., NextGen in the US [33]) to modernize air transportation systems worldwide using new technologies to increase the safety, efficiency, and resiliency of global airspace. One of the technologies is the controller-pilot datalink communications (CPDLC) which enables the exchange of messages between the pilot and the ATC. The ATCs can issue flight plan updates, altitude, and speed changes, radio frequency channel assignments, etc., to the pilots using CPDLC. CPDLC also allows

[1] pre-covid traffic in 2019 was 38.9 million flights

[2] https://www.boeing.com/commercial/aeromagazine/aero_02/textonly/fo02txt.html

pilots to respond to messages, request new clearances, or report statuses back to the ATC. In general, CPDLC has been replacing voice communications, and today certain airspaces require the aircraft to be equipped with aviation datalink capability [22, 11]. Over the years, adopting this datalink has enabled the controllers to reduce aircraft separation by allowing more aircraft to share the airspace.

Given the importance and growing use of aviation datalink applications, testing the system's resilience to modern-day attacks is vital. As compared to popular aviation systems like instrument landing system (ILS) [204], automatic dependent surveillance-broadcast (ADS-B) [66, 205, 222], global navigation satellite system (GNSS) [237, 228], and collision avoidance systems [181, 219], only a few works have explored the security aspects of aviation datalink. Most notably, [216, 247] lay down strategies to exploit the lack of authentication. However, they do not elaborate on the feasibility and impact of the attack. Furthermore, the attacks proposed in the above works can be trivially detected, and the requirements of introducing stealthy CPDLC manipulation were not considered. Also, there exists no proof-of-concept implementation and evaluation of the attacks. In our work, we systematically analyze the security guarantees of aviation datalink applications and demonstrate their vulnerabilities at each phase of the flight. We consider the effect of spoofing aviation datalink applications such as CPDLC on other dependent systems (e.g., automatic dependent surveillance-contract (ADS-C) [114]) and present strategies to jam or craft appropriate messages to avoid raising suspicion selectively. Based on publicly available databases and other broadcast signals, we analyze and point out vulnerable regions in the US airspace and demonstrate strategies that can affect a single targeted aircraft as well as multiple aircraft.

Specifically, make the following contributions.

- First, we present a spoof and jam attack strategy to stealthily inject malicious messages to influence the flight crew's decision -making. We systematically analyze widely used mission-critical and often overlooked applications like CPDLC and ADS-C and outline the requirements and considerations for exploiting these applications.
- We show that an attacker can target a single aircraft and spoof CPDLC messages that provide the flight crew with incorrect information, including transponder codes, instrument calibration settings, departure clearances, and voice contact frequencies.
- Next, we propose a coordinated attack that targets multiple aircraft with intersecting trajectories. Specifically, we devise a strategy to craft CPDLC messages with appropriate altitude and optionally waypoint change instructions to force multiple aircraft to cross each other at the same altitude. Thus causing mid-air close contact.
- To facilitate these attacks we implement and evaluate an ACARS message spoofer and a reactive jammer. The reactive jammer can achieve a 1.48 ms reaction time

and 98.85% jamming success. Thus allowing the attacker to avoid detection. To the best of our knowledge, this is the first work presenting a reactive jammer for aviation datalink applications using the ACARS network.

- After analyzing air traffic data from 2021, we find a total of 592,224 instances where trajectories of two flights intersect in time and space[3], essentially 592,224 opportunities for an attacker to execute a coordinated attack. We further identify 48 regions with a 90% chance of spotting at least one such intersection in a day. These regions spread across the mainland US and amount to 1.49% of the total airspace.

Rest of the chapter is structured as follows. We first describe ACARS and provide a background of aviation datalink and its various components. In Section 3.3, we first present the attacker's goal and assumptions and outline the requirements for executing an attack. Next, we propose a spoof and jam strategy to stealthfully inject CPDLC messages. This is followed by a description of attack scenarios that target single aircraft and several strategies to execute coordinated attacks on multiple aircraft. In Section 3.4, we present the implementation of ACARS message spoofer and jammer, followed by Section 3.5, where we present the evaluation of various attacker components involved in the attack. Next, in Section 3.6, we discuss the need to consider the aircraft system as a whole for executing an effective attack and propose countermeasures. Finally, we provide an overview of related works and conclude this chapter.

3.2 Aviation Datalink Applications

Aviation datalink encompasses technologies that provide a direct controller-pilot datalink and enable safe, efficient, and accurate operations. Aviation datalink applications are based on two major implementations: i) Traditional aircraft communications, addressing, and reporting system (ACARS) network or ii) modern aeronautical telecommunications network developed by the International Civil Aviation Organization (ICAO) and adopted by Eurocontrol as the primary datalink infrastructure [23]. ACARS is a communication system that links aircraft with ground stations. Primarily, ACARS uses very high frequency (VHF) Datalink Mode A/0 physical layer technology to deliver messages. VHF radios strictly require line-of-sight operations. This limits VHF coverage to only up to 200 miles. To overcome the coverage limitation of VHF radios, an high frequency (HF) datalink, and satellite communication (SATCOM) links were added to the ACARS network.

Initially, ACARS was used to exchange data like weather information, aircraft maintenance reports, gate assignments, and passenger information with the airline's operational

[3]Aircraft maintain vertical separation while crossing

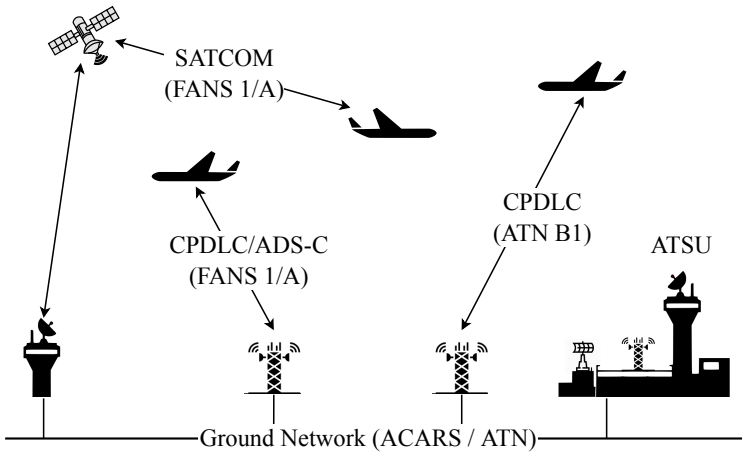


Figure 3.1: A graphic representation of aviation datalink ecosystem and its various components. The ground station radios use either the ACARS network or Aeronautical telecommunications network (ATN). However, there are systems that support both networks.

control. Support for datalink applications that provide air traffic control services were added under the future air navigation systems (FANS) structure conceptualized by ICAO; Boeing developed FANS 1, after which Airbus developed FANS A. These systems together are referred to as FANS 1/A [109]. FANS 1/A applications are designed to utilize existing ACARS networks. They support VHF Datalink Mode A/0 transmissions used by traditional ACARS applications for domestic and oceanic operations and newer VHF Datalink Mode 2.

Aeronautical telecommunications network (ATN) is a network of interconnected systems that efficiently facilitate ground-ground and air-ground sub-networks exchange of information between concerned entities. Unlike FANS 1/A, ATN baseline 1 (B1) provides a CPDLC implementation that supports only the VHF Datalink Mode 2 for message transmission. However, in the future, there are plans to adopt an IP-based network to improve message delivery and integrity further. As part of their Single European Sky initiative, Eurocontrol has adopted ATN B1 as the primary datalink implementation. ATN B1 operations are restricted to Europe. On the other hand, FANS 1/A is the primary datalink implementation adopted in America and most other places. Figure 3.1 provides a graphical representation of the aviation datalink ecosystem.

Both implementations are supported by two major datalink applications, CPDLC and ADS-C. Even though these networks are incompatible, avionics are designed to provide a seamless transition. These applications provide a direct communications link between a pilot and air traffic controllers (ATC) and enable the transmission of ATC commands and surveillance messages more efficiently. CPDLC is a data link that enables air traffic controllers and pilots to exchange air traffic service messages traditionally sent over voice channels such as

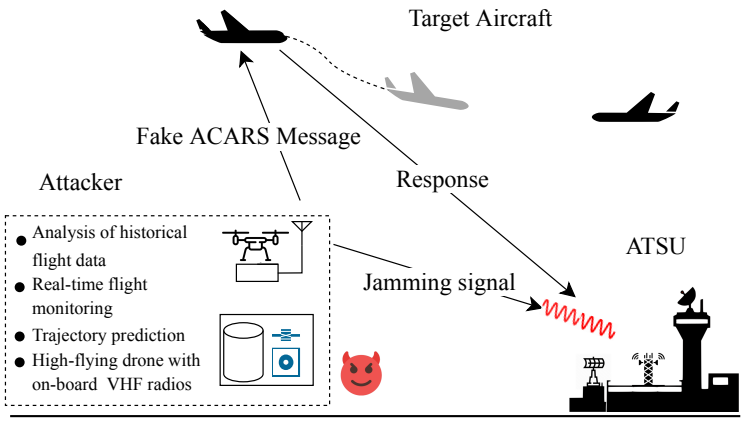


Figure 3.2: A graphical representation of the proposed attack. Real-time flight monitoring and trajectory prediction allow the attacker to time its transmission of fake ACARS messages to the target aircraft. The reactive jammer transmits a jamming signal to prevent the ATSU from receiving the response.

en-route services, flight plan amendments, departure clearances, and instructions to execute specific instructions maneuvers like changing the altitude. Not to be confused with ADS-B, ADS-C is a system where an aircraft initiates a contract with one or more air traffic services unit (ATSU) and periodically sends out various reports that contain the aircraft's position, speed, altitude, route predictions, airframe data, and meteorological data to one or more ATSU. It is important to note that ATN B1 does not support ADS-C application [115]. This work focuses on FANS 1/A applications and targets specific CPDLC and ADS-C messages exchanged using the ACARS network. ATN B1 and FANS 1/A support different message sets. However, these attacks can be used against ATN B1 applications with some changes.

3.3 Security Analysis of Aviation Datalink Applications

This section defines the attacker model, its assumptions, and its goals. Next, we analyze and outline the requirements for spoofing mission-critical CPDLC messages. We determine that along with spoofing CPDLC messages, an attacker should also jam appropriate transmissions. Finally, we describe specific attack scenarios that use the CPDLC message spoofing and reactive jamming to attack one or more aircraft.

3.3.1 Attacker Goal and Assumptions

In this work, we model our attacker based on the Doley-Yao (DY) attacker model. We consider an attacker capable of intercepting, injecting, and reactively jamming ACARS messages. Unlike the omnipotent attacker described in the DY model, in our case, the

attacker has certain physical layer restrictions like transmit power and radio coverage. VHF signal reception requires strict line-of-sight communication; this restricts the coverage of an attacker. For example, even though an attacker with a radio set located on the ground can communicate with a high flying plane that is 400 km away, being closer to the ground, it can only communicate with an air traffic controller 15 to 30 km away. However, the attacker can use high-flying drones to increase radio coverage. Furthermore, to better plan the attack and determine an optimal location, we consider that the attacker has access to historical and live air traffic data through services like Opensky network [206], FlightAware, or live over-the-air ADS-B transmissions. The attacker is equipped with a software-defined radio capable of transmitting and receiving VHF voice signals and ACARS signals that use airband frequencies from 118 MHz to 137 MHz. For example, commercial off-the-shelf software-defined radios like USRP B210, HackRF, and LimeSDR.

The attacker's goal is to influence the flight crew's decision-making by spoofing CPDLC messages and forcing them into making harmful decisions that jeopardize the aircraft's safety. For example, forcing the flight crew to change the flight path that brings their aircraft close to a passing aircraft or providing them with false information that facilitates incorrect instrument calibration. An attack is considered successful if the flight crew accepts the CPDLC messages and executes the spoofed instructions. Figure 3.2 provides a graphical representation of a generic attack scenario and various components required for a successful attack.

3.3.2 Attack Prerequisites

To spoof CPDLC messages, the attacker needs to know the identity of the aircraft's current data authority (CDA). In other words, the attacker should know which ATSU is currently authorized to send CPDLC commands to the aircraft. An aircraft will accept CPDLC commands only if issued by an ATSU, designated as the CDA. If it receives commands from any other ATSU, the onboard software will automatically reject the command, and the aircraft will automatically send a *DM63 - NOT CURRENT DATA AUTHORITY* message [116]. To execute specific multi-aircraft attacks, the attacker will also require route predictions and position estimates to calculate the closest point of approach necessary for determining the time of the attack.

CDA Identity:

An aircraft supports two connections at any time, one active and one inactive connection. It establishes an active connection after completing the logon and the connection procedure. Figure 3.3 shows the sequence of CPDLC messages exchanged during these procedures.

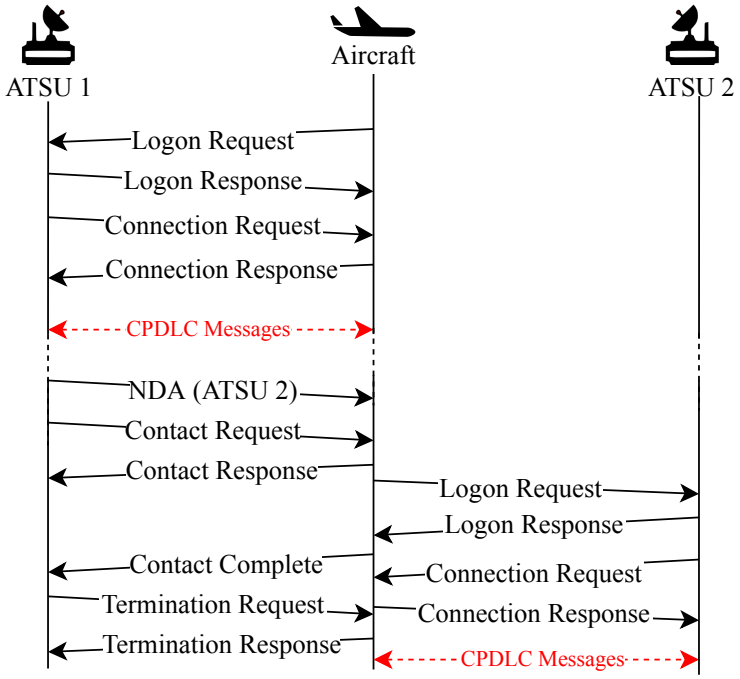


Figure 3.3: A sequence of CPDLC connection management messages exchanged between the aircraft and multiple ATSUs. The aircraft follows this sequence of messages until it reaches its destination.

Before departure, the flight crew initiates the logon procedure by entering the ATSU's ICAO identifier. After receiving the logon request, the ATSU correlates the flight plan and responds with a logon confirmation, followed by a connection request. The ATSU automatically rejects the logon request if the flight plan correlation fails [117]. If the aircraft doesn't have an active connection, the aircraft responds with a connection confirmation and establishes an active connection. The ATSU with which the aircraft establishes an active connection is called the CDA. If the aircraft already has an active connection and if the ATSU is the next data authority (NDA), i.e., the next ATSU according to the filed flight plan or as decided by the CDA, the aircraft establishes an inactive connection. However, if the ATSU is neither the CDA nor the NDA, FANS 1/A equipped aircraft will send a connection reject message followed by the identity of its CDA. ATN B1 aircraft, on the other hand, will only send a *DM107 NOT AUTHORIZED NEXT DATA AUTHORITY* message [118]. This way, an attacker can find out the identity of the CDA that it needs to impersonate to initiate CPDLC message exchanges.

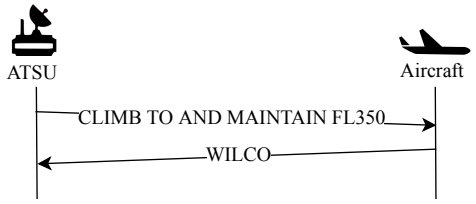


Figure 3.4: A sequence of CPDLC dialogue between ATSU and the aircraft. The ATSU instructs the aircraft to climb and maintain an altitude of 35000 ft. The flight crew confirms by sending *WILCO*, which means that the flight crew has accepted the instruction.

Route predictions:

An ATSU may initiate one or more ADS-C contracts with the aircraft by sending an ADS-C contract request. These contracts include a report group that contains route predictions. However, such a report is optional, and none of the ATSUs may have contracts that require this report. To receive route predictions, the attacker can initiate a periodic contract or request a one-time contract that contains the required route predictions and position estimates. It is important to note that all ADS-C functions, including signing up for contracts and reporting, are automated, i.e., they do not require flight crew intervention [119]. An aircraft may simultaneously have contracts to more than 1 ATSU. Moreover, the ATSUs are unaware of any other contract held by the aircraft. This allows the attacker to initiate contracts stealthily by impersonating an ATSU.

3.3.3 Spoofing CPDLC Messages

Depending on the intended effect of the attack, there are two ways to execute an attack: i) The attacker impersonates an ATSU and sends a malicious instruction to the aircraft, and jams the response to prevent the legitimate ATSU from receiving the message. This is useful when the injected messages contain commands that require the flight crew to execute maneuvers, e.g., when the attacker wants to modify the aircraft's trajectory. ii) The attacker first intercepts and jams a request from the aircraft preventing the legitimate ATSU from receiving the request. Next, the attacker sends a malicious response by impersonating the legitimate ATSU. Such a strategy is useful when the attacker wants to manipulate the ATSU's response. This is beneficial when the attacker wants to spoof route clearance or digital automatic terminal information service (D-ATIS) data.

Once the attacker impersonates the CDA, the aircraft's onboard computer will accept any CPDLC command that it receives from the attacker. Through CPDLC messages, controllers can provide mission-critical instructions to the flight crew, including commands to change speed, add a waypoint, change altitude, monitor and contact the controllers on

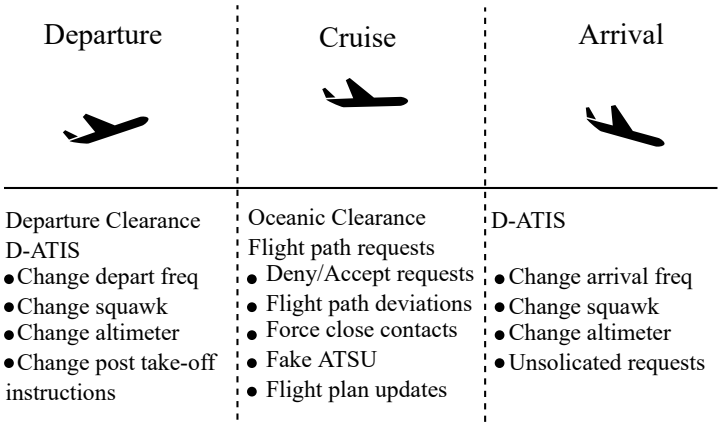


Figure 3.5: Using a combination of ACARS message injection and jamming, an attacker can modify instructions and procedures specific to a flight stage.

the specified VHF frequency, and assign squawk code, and provide the correct altimeter setting. Refer to Figure 3.4 for an example of a CPDLC dialogue between the controller and the flight crew. A typical flight goes through three distinct phases, i) the departure phase, ii) the cruise phase, and iii) the arrival phase. ACARS, in some capacity, is used in every flight phase for exchanging crucial information that the flight crew requires for maintaining safety. Figure 3.5 gives an overview of the data that the attacker can modify in each flight stage with malicious intent.

To execute a successful attack, the attacker should carefully craft messages with the appropriate label, message identifier, and content. Based on the source and the destination, CPDLC messages are categorized as: i) downlink (DMxx) and ii) uplink (UMxx). In Section 3.4.1 we provide more details on the structure of uplink and downlink CPDLC messages. Downlink messages are sent from the aircraft down to the ATSU, and uplink messages are sent from the ATSU up to the aircraft. Each uplink and downlink message contains a four-character message-id that enables the receiver to sequence and rearrange the messages. These messages follow a syntax similar to regular voice communications and are often used instead of analog voice channels, e.g., (go) *DIRECT TO [position]*. CPDLC provides 81 downlink and 183 uplink message types that have streamlined tower operations [120]. CPDLC messages are integrated into the flight management system (FMS), and uplink messages that amend flight plans are automatically loaded into the FMS upon flight crew approval [121]. This allows an attacker to send specific messages that can automatically amend the existing flight plan after flight crew approval. ACARS messages contain a 16-bit cyclic redundancy checksum (CRC) that allows error detection through a message integrity check. Furthermore, CPDLC messages also contain an application-level

CRC to verify the integrity of the transmitted CPDLC commands. If the CRC check fails at the ACARS level for a transmission initiated by the aircraft, the ground station will ignore the message. However, if the CRC check fails at the CPDLC level, the ground station will send a CPDLC error response [71]. CRC values are sensitive to bit flips, and an attacker can prevent the receiver from receiving a packet by corrupting a small portion of the message. This provides an opportunity for the attacker to jam the transmitted messages.

In the past, researchers have found that some airlines tend to use weak encryption to secure messages [217]. However, this is true only for airline-specific messages and not for operational messages between ATC and aircraft. All the essential messages used to manage the flight are unencrypted and are susceptible to adversarial manipulation.

3.3.4 Jamming CPDLC and ADS-C Messages

There are multiple redundancies available at the flight crew's disposal should need be. Even minor suspicion can cause the flight crew or the controllers to activate the fallback mechanism, which, in this case, is reverting to the VHF voice. Through an analysis of CPDLC and ADS-C messages, we learn that, despite the lack of security controls, as a result of strict message structure, limited request/response options, and mandatory response from the flight crew, simply spoofing ACARS messages to influence the flight crew's decision-making is not enough. Along with message spoofing, the attacker should also be able to jam appropriate messages. Continuous jamming will cause a denial of service type attack as the ATSU will not receive any messages. The flight crew and controllers will immediately revert to the VHF voice in this scenario. The attacker needs a reactive jammer capable of jamming only specific messages from particular flights to avoid such a situation. The attacker will have to reactively jam responses and ADS-C contracts to ensure attack success.

Response Jamming:

CPDLC message exchange follows a specific pattern and structure. The flight crew has to select from pre-determined and hardcoded request/response options. Available options are determined by the communication management unit based on the received request. A set of pre-determined responses ensures that the message structure remains intact and prevents the flight crew from transmitting inappropriate responses. The following options are considered as appropriate responses, i) *DM 0* WILCO - Will Comply, ii) *DM 1* UNABLE, iii) *DM 2* STANDBY, iv) *DM 3* ROGER, v) *DM 4* AFFIRM, and vi) *DM 5* NEGATIVE [122]. The spoofed instructions should be convincing, and even a slight suspicion can cause the flight crew to decline the commands and contact the ATSU over a voice channel. According to the operational guidelines, when the ATSU receives unexpected messages, inappropriate

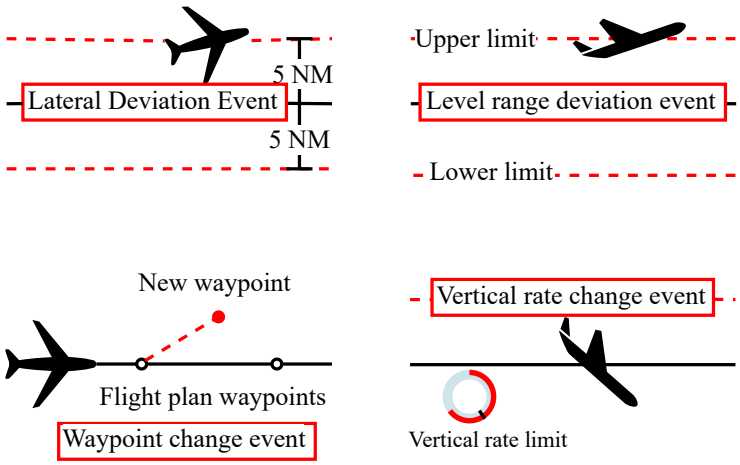


Figure 3.6: Graphical representation of various events that trigger ADS-C event reports.

responses, or even no response at all, the ATSU is required to follow up and investigate over a voice channel [123]. Furthermore, the ATSU may instruct the flight crew to disconnect CPDLC and continue solely over voice [124]. All messages are broadcast, and the ATSU will receive and respond to all the messages addressed to it. Hence the attacker should actively monitor ACARS communications and jam appropriate requests and responses to avoid detection.

ADS-C Report Jamming:

There are three types of contracts, i) periodic contract, which requires the aircraft to periodically send the specified information, ii) demand contract, where the ATSU requests a one-time report. And iii) event contract, where the aircraft automatically reports if the specified event occurs. The ATSU specifies the reporting interval from 64 to 4096 seconds for periodic contracts. These periodic reports contain aircraft position reports, waypoint predictions, airframe data, and meteorological information [125]. The attacker can leverage these predictions to coordinate attacks between multiple aircraft as described in Section 3.3.6.

When an aircraft enters into an event contract with an ATSU, the FMS automatically sends an event report to the ATSU when a particular event occurs. Based on airspace requirements, the ATSU must establish event contracts for the following events [126], i) Waypoint change event (WCE). Whenever the pilot enters a new waypoint, this event will be triggered, and the ATSU will be triggered notified of the change. ii) Level range deviation event (LRDE), the ATSU defines a lower and upper bound on the aircraft's altitude. Whenever the aircraft exceeds this limit, an LRDE report is transmitted. iii) Lateral deviation

event (LDE), similar to LRDE, the ATSU establishes a lateral limit. The FMS will transmit a message whenever the aircraft deviates beyond the set limit. iv) Vertical rate change event (VRE) report is sent when the aircraft descends or climbs at a rate outside of set limits [127]. Figure 3.6 shows a graphical representation of these events. These reports help the controllers avoid potential mid-air close contacts and maintain established separation minima in busy airspace with tight separation limits. It is essential for an attacker to know about active contracts agreed by the aircraft as these reports may indicate any maneuvers executed by the flight crew based on spoofed CPDLC messages.

Aircraft automatically sends out ADS-C reports, which reflect aircraft's movements, waypoint predictions, and various events. Depending on the airspace, some ATSUs may initiate event contracts. Whenever a specific event occurs, for example, the aircraft crossing the altitude range or the flight crew entering a custom waypoint, the aircraft automatically sends an event report to the ATSU. This may alert the air traffic controllers and can lead to attack detection. To prevent this, the attacker needs to monitor and jam event reports and any other ADS-C report that may indicate that the flight crew is executing unauthorized maneuvers.

3.3.5 Single Aircraft Attacks

Clearance Manipulation:

At the beginning of the departure phase, the flight crew requests information that helps them prepare the flight management system (FMS) for departure. The flight crew specifically requests the pre-departure clearance (PDC) and D-ATIS. Refer to Figure 3.7 for contents of the PDC[4]. PDC is a set of post-take-off instructions specific to a particular flight. Usually it includes the climb via waypoint, the filed flight plan, post-take-off altitude, and frequency for post-take-off voice contact. It also contains a 4-digit squawk code used to identify the aircraft on secondary surveillance RADAR screens uniquely. The flight crew is responsible for manually configuring their transponder to use the correct squawk code. D-ATIS contains information that is common to all aircraft. It includes the most recent weather report and other warnings that have potential obstructions around the airport. It also provides the current altimeter setting that allows the flight crew to calibrate the aircraft's altimeter to get a more accurate absolute altitude.

The attacker can jam PDC message requests and replace legitimate responses with malicious messages containing incorrect squawk code. Changing the squawk code can lead to confusion and disruption of situational awareness as ATCs will provide the right instructions to the wrong flight. In [224], a survey performed by the authors indicates that 54.82% of

[4]Contents redacted to preserve anonymity

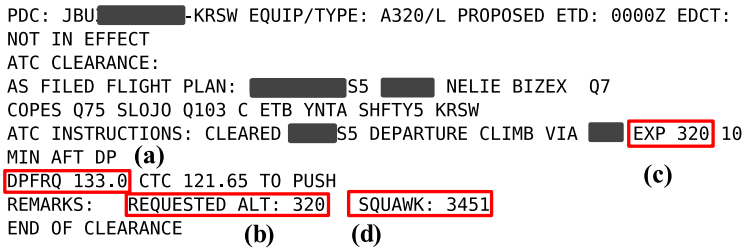


Figure 3.7: An example PDC that we intercepted. (a) mentions the departure and the ground frequency in (MHz) that the flight crew should contact, (b) and (c) shows the requested altitude and the approved altitude. (d) shows the assigned squawk code.

the participating controllers feel that incorrect labels on the RADAR screen can cause a major loss of situational awareness. The attacker can also change the ground control and the departure control frequencies to an unused frequency from the airband controlled by the attacker. Finally, the flight crew contacts the attacker for taxi and post-take-off instructions.

Altimeter Setting Manipulation:

According to a Boeing report [49], the arrival phase is the most accident-prone phase. The arrival phase starts when the flight crew receives instructions from the ATC to descend once it gets closer to the airport. The ATC assigns a certain runway, and the flight crew follows the stated approach procedures. During this phase, the flight crew puts in a request for arrival D-ATIS. This is similar to the D-ATIS it receives before take-off, except it has more information regarding the current approach and runways. Most importantly, it has the correct altimeter setting for the destination airport. To get the absolute altitude above ground level, it is important to re-calibrate the altimeter before landing [85] and the Federal Aviation Administration (FAA) requires it. An attacker can intercept and jam the flight crew's request to receive D-ATIS. The attacker then injects a malicious D-ATIS message with a modified altimeter setting. Refer to Figure 3.8 for the sequence of message exchanges for manipulating the altimeter setting. The altimeter setting is a four-digit number corresponding to inches of Hg e.g., 2992 means 29.92" Hg. A single-digit error results in a discrepancy of 10 ft in altitude. Flying with an incorrect altimeter can lead to a *controlled flight into terrain* event, which can have a catastrophic outcome [34].

Handoff Takeover:

When an aircraft transitions from one airspace to another, the CDA instructs the aircraft to logon to the next ATSU as per the flight plan, also called NDA. This process is called "handoff." The CDA first informs the aircraft about the NDA by sending a *UM160 NEXT*

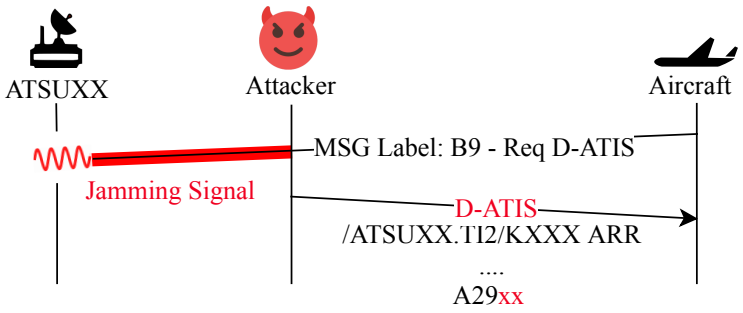


Figure 3.8: A sequence of message exchanges in an altimeter change attack forces the approaching aircraft to set the altimeter incorrectly. The approaching aircraft sends a request to receive ATIS information with a *B9* label. The attacker intercepts the message and jams it. The attacker then sends a fake ATIS message with a modified altimeter value.

DATA AUTHORITY message. As the aircraft gets closer to the boundary of the NDA, the CDA sends a contact request which triggers the aircraft to initiate a connection with the specified NDA. The aircraft follows the regular logon procedure (unlike initial logon, this procedure does not require pilot input). To exploit the handoff, an attacker can send a *UM160 NEXT DATA AUTHORITY* message with a fake ATSU. FANS 1/A aircraft will automatically overwrite its NDA entry after receiving a *UM160* message [128]. This will be followed by a sequence of contact request messages instructing the aircraft to initiate a connection with the attacker-specified ATSU. An attacker can choose the ATSU beyond the aircraft's radio range to make the attack more stealthy and hard to detect.

VHF Voice Man-in-The-Middle

Pilots and controllers will often revert to VHF voice for communicating time-sensitive and safety-critical information. Controllers will also revert to voice when they receive inappropriate or unexpected messages. Often, pilots are required to verbally check in with the controllers, especially when transferring from one controller region to another controller region. In case the controllers require the pilots to contact over voice channels, the controller will send a *UM120 MONITOR [ICAO address] [frequency]* [129]. CPDLC message that instructs the pilots to tune and monitor a specific frequency. An attacker can leverage this message type to execute a man-in-the-middle attack and establish a VHF voice relay between a controller and the flight crew. Figure 3.9 shows message exchange and the attack concept. The attacker sets up a rogue VHF voice transceiver station on an unused air-band frequency. If the controller decides to reach out to the flight crew, unaware of the rogue channel, the controllers will use their regular channel. The attacker then responds to the controller as the victim aircraft's flight crew. This attack requires the attacker to set up a

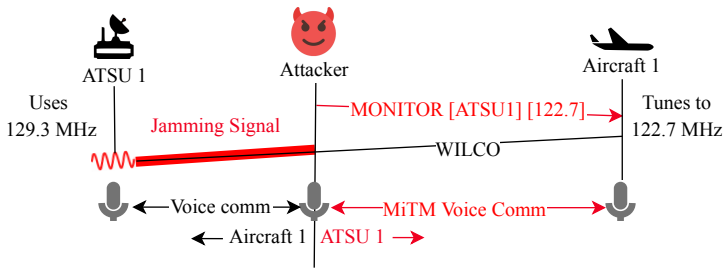


Figure 3.9: A sequence of message exchanges in a VHF voice man-in-the-middle attack. The attacker sets up a VHF voice station and instructs the aircraft to select the attacker's frequency instead of ATSU's. Next, the attacker relays manipulated voice communications between ATSU and the aircraft.

VHF voice station. Moreover, the attacker needs to be fluent in aviation phraseology and grammar.

3.3.6 Coordinated Multi-Aircraft Attacks

In this category of attacks, the attacker targets multiple aircraft and executes a coordinated flight path manipulation. As a result, the target aircraft come in close four-dimensional proximity. Flight routes are planned such that flights maintain lateral and vertical separation. However, flight paths frequently intersect. In this case, the flights cross each other while maintaining strict vertical separation. Through our analysis of flight data from the year 2021, we found that there were 592,224 instances of flight crossings with the lateral separation between the aircraft is <100 m. An opportunistic attacker can target these crossings and launch coordinated attacks instructing the flights to change their altitude to violate the vertical separation mandate. Alternatively, if the flight paths do not intersect, the attacker can add custom waypoints such that the flight paths cross at a point selected by the attacker.

The attacker prepares for the attack by finding a region that frequently sees intersections and positions itself to minimize jamming costs as described in Section 3.5. To send instructions to the flight crew through CPDLC messages, the attacker needs to impersonate the respective aircraft's current data authority. Depending on the location, each aircraft may be connected to a different ATSU. An attacker can determine the ATSU by monitoring CPDLC message exchanges or, as explained in Section 3.3.2. Once the attacker knows the CDA of both flights, the attacker can proceed with CPDLC message transmission and jamming.

The attacker's goal in this attack is to force multiple aircraft to cross at the same altitude while in close proximity to each other. This requires the attacker to estimate the closest

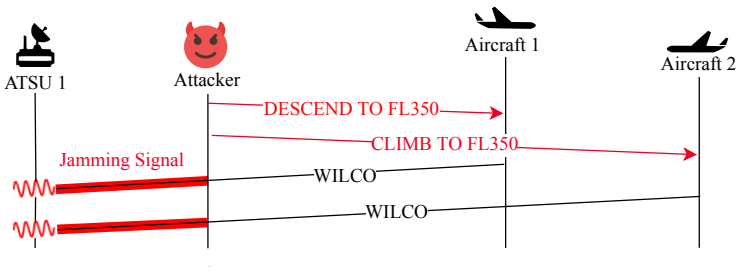


Figure 3.10: A sequence of message exchanges in an altitude change attack forces two crossing aircraft to change altitude and fly at the same flight level. The attacker uses the reactive jamming technique to detect the WILCO response and jam it to prevent the ATSU from receiving it.

point of approach or the time and the place where the flights will cross. The attacker can estimate the closest approach point through trajectory prediction, which is challenging. Neither speed I'll nor direction is constant. Even minor changes to either can lead to significant errors over time. The attacker can initiate an ADS-C contract to circumvent this problem wherein the aircraft periodically sends out route predictions. It is important to note that initiation of ADS-C contracts *does not* require flight crew approval. In [86] the authors examine the accuracy of these predictions. Air traffic controllers often rely on these predictions to maintain aircraft separation as these predictions serve as an early warning system for potential close contacts.

After establishing the closest point of approach, depending on each aircraft's current altitude, the attacker sends *UM20 CLIMB TO AND MAINTAIN [altitude]* and *UM23 DESCEND TO AND MAINTAIN[altitude]* messages. These messages instruct the flight crew to reach the desired altitude. The flight crew will respond to these messages with *WILCO*, *UNABLE*, or *STANDBY*. If the flight crew accepts the requests, then they send a *WILCO* message to the air traffic controller and perform the requested maneuver. However, in some cases, the flight crew may decide to decline the request or negotiate and suggest an alternative. Since the controllers keep track of messages they send and receive, they will consider the flight crew's response as an unexpected message, and in most cases, the controllers will use VHF voice to contact the flight crew and clarify. To avoid such a situation, it is necessary for the attacker to selectively jam messages that contain responses to previous instructions spoofed by the attacker. Refer to Figure 3.10 for a sequence of messages exchanged in this attack.

Similarly, an attacker can send instructions to change the route by manipulating the next waypoint. For example, the attacker can send a *UM63 AT [time] CROSS [position] AT AND MAINTAIN [altitude] AT [speed]* message that provides very specific instructions

```
/NYCODYA.AT1.A6-EGJ001F800750D2A94010CE21171234
FANS-1/A CPDLC Message:
 CPDLC Uplink Message:
  Header:
   Msg ID: 0
  Message data:
   AT [time] CROSS [position] AT AND MAINTAIN [altitude] AT [speed]
    Time: 00:00
    Latitude:   40 42.1' north
    Longitude: 074 00.2' east
    Altitude (QNH): 10000 m
    Ground speed: 300 kts
```

Figure 3.11: A UM63 command to instruct an aircraft to set the following waypoint location and altitude as received and decoded by *acarsdec* and *libacars*. It also provides instructions on the target speed, and time the aircraft should cross the waypoint.

on when to cross a position, at what altitude, and at what speed. Figure 3.11 shows a UM63 CPDLC message with all the necessary parameters. Such a message amends the flight plan and adds a waypoint after the flight crew approves. Route changes are usually correlated with the existing flight plan, temporary restrictions, and NOTAMs (Notice to Airmen). Therefore, such requests may raise suspicion and force the flight crew to contact the ATSU over voice channels. Hence, it is important to ensure that the injected messages do not raise suspicion.

3.4 Proof-of-Concept Implementation

This section describes the implementation of the ACARS message spoofer and the reactive jammer capable of jamming specific messages from predetermined aircraft. This implementation serves as a proof-of-concept for the most important component of the proposed attacks.

3.4.1 ACARS Message Spoofer

To realize the proposed attacks, the primary requirement of the attacker is the ability to spoof ACARS messages. ACARS uses a 2400 bps packet-like system that uses a Telex format for short messages. It uses a VHF carrier in the airband for data transmission. ACARS uses a modulation technique called as minimum shift keying (MSK). In the MSK scheme used for ACARS, a 1200 Hz tone marks a bit switch, and a 2400 Hz tone indicates that the bit remains unchanged. In ACARS MSK, bit transitions occur at zero-crossing, and as a result, the phase is continuous through bit transitions. The MSK-encoded data is modulated onto a VHF carrier using amplitude modulation (AM) to use standard aircraft radio equipment. ACARS supports a baud rate of 2400 Hz. At 48000 samples/sec, each bit is represented by 20 samples, and each symbol is 416.67$\mu secs$

Table 3.1: ACARS Packet Structure.

Field	Size (bytes)	Description
Pre-key	16	A 128-bit sequence of binary 1s
Bit-sync	2	Character '+' and '*'
Character-sync	2	Consecutive <SYN>characters
Header start	1	Character <SOH>
Mode	1	Broadcast/Single RX
Address	7	Aircraft Address
Ack	1	Technical Acknowledgment
Label	2	Message type
Block identifier	1	Sequence number
Start of text	1	<STX>or <ETX>if no text
Text	0-220	Printable characters
Suffix	1	<ETB>or <ETX>if last block
BCS	2	Checksum
BCS Suffix	1	<DEL>

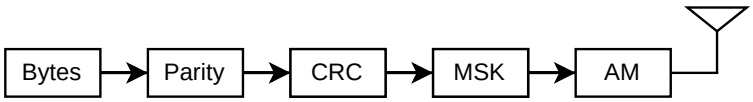


Figure 3.12: Block diagram of ACARS transmitter.

The ACARS transmitter takes in a text message of at most 220 characters. Each 7-bit ASCII character is protected with an odd parity bit and is transmitted with the least significant bit first. A CCITT polynomial is used to calculate a 16-bit CRC. This checksum ensures the integrity of the entire message. Once the message is formatted correctly, and once the CRC is appended, the data undergoes MSK modulation. It is important to note that in ACARS, this modulated signal is not transmitted as it is. The baseband MSK signal is modulated onto a VHF carrier frequency using the amplitude modulation technique. In our transmitter design, we first create the ACARS packet according to the packet structure described in Table 3.1 and then modulate the data as per the specifications using a GNURadio [9] flowgraph.

Refer to Figure 3.12 for a schematic of the ACARS transmitter. GNURadio supports multiple RF-frontends. In our evaluation, we use a USRP B210 from Ettus. The transmitter design is based on various publicly available resources [1, 108]. CPDLC messages follow a specific message structure and are based on the regulation specified in [71]. It also provides an Abstract Syntax Notation One (ASN.1) structure for defining these messages. To encode/decode message strings, these messages use ISO/IEC 8825-2:1996 packet encoding rules (PER) - Basic Unaligned [71]. Like ACARS message, the CPDLC message string contains a separate 16-bit CRC value, specifically to detect errors in the CPDLC message

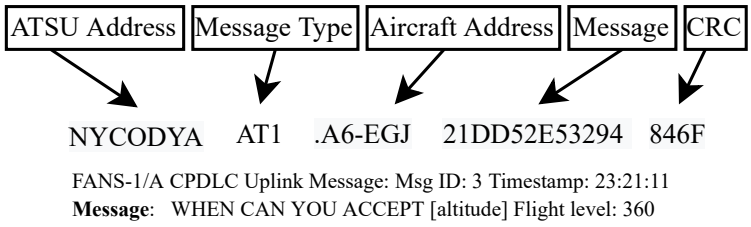


Figure 3.13: The message structure for a CPDLC message and an example CPDLC message received during our analysis and evaluation.

at the application level. Refer to Figure 3.13 for the structure of the CPDLC message and an example message.

The attacker also requires a message receiver to intercept legitimate messages along with the message spoofer. Multiple open-source projects [207, 151, 147] provide the necessary software to receive and decode messages. From these, we rely on the popular ACARS decoder program called *acarsdec* [147]. *Acarsdec* is a program written in C language that interfaces with an RTL-SDR. Along with ACARS messages, with the *libacars* [150] library, it can also decode FANS 1/A CPDLC and ADS-C messages. We evaluate and verify the ACARS message format and CPDLC structure using these tools.

3.4.2 ACARS Message Jammer

An essential component of our attack strategy is the message jammer. A jammer is an RF transmitter that transmits noise that disrupts wireless communications. For this attack, we implement two types of jammers, i) a random noise transmitting jammer and ii) a pulse jammer that transmits a short high powered pulse that distorts the message bits such that the receiver fails CRC checks and rejects the received packet. To evaluate our jammer's performance and effectiveness, we conducted a series of experiments where we checked the ACARS receiver's packet reception rate. The experimental setup is as follows. We use a USRP B210 as the transmitter and an RTL-SDR with *acarsdec* as the receiver to perform these experiments. The transmitted signal contains the ACARS message. This message is combined with the jamming signal using a combiner and is fed directly into the receiver. It is important to note that all these experiments were conducted over hard-wired devices.

Unlike conventional jammers that jam regardless of a transmission, a reactive jammer waits and jams only when it detects specific transmissions. In the past, researchers have explored and evaluated reactive jammers in the context of wireless networks [174, 234, 64]. Wilhelm et al, in [243] show that a reactive jamming prototype based on a USRP2 equipped with a Xilinx Spartan-3 FPGA can achieve a reaction time of a few microseconds. There are fewer time constraints in ACARS, so such a reactive system is feasible. Figure 3.14

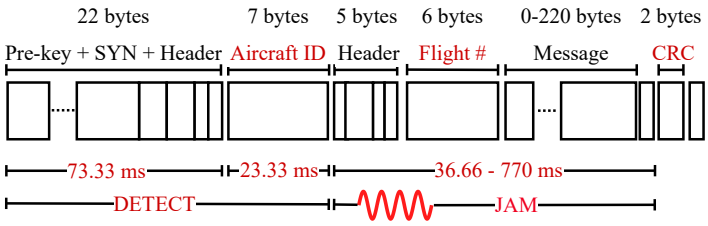


Figure 3.14: Time constraints: the reactive jammer has 36.66 to 770 ms (depending on the message length) to detect an incoming message, check the aircraft ID, and initiate the transmission to jam the ACARS packet.

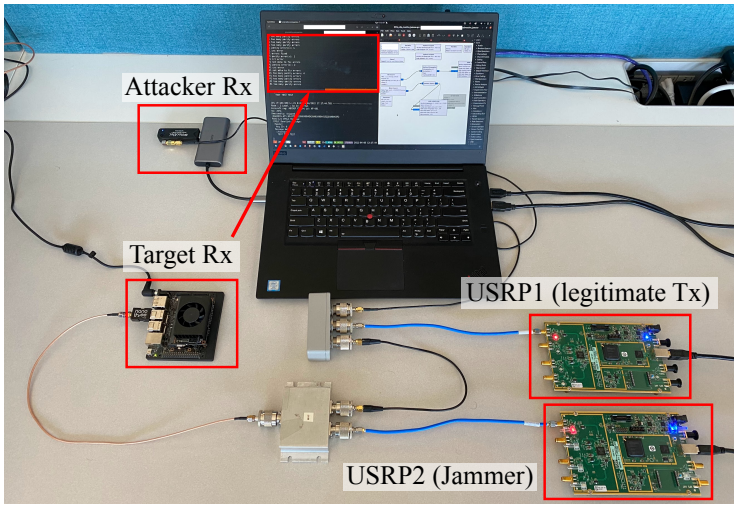


Figure 3.15: Photograph showing the reactive jammer evaluation setup. USRP1 transmits the legitimate message, attacker and target receivers simultaneously receive the messages. However, when the attacker receiver detects a message for a pre-programmed aircraft address, it instructs the jammer to start.

shows the time constraints for jamming an ACARS packet.

We implement the reactive jammer using a combination of custom GNURadio blocks and an open-source software-defined ACARS receiver. Specifically, we modify the signal flow within Acarsdec to send a jamming signal to the jammer once it detects an ACARS message from the target aircraft. With a "start jamming" signal, our modified version of Acarsdec also sends a "stop jamming" signal as soon as it receives the entire message. Figure 3.15 shows a photo of the actual reactive jammer evaluation setup. The attacker can use strategic antenna placement and interference cancellation to avoid receiving its jamming signal. It is essential to stop the jamming signal because a prolonged transmission can draw out attention, and as a result, the attacker can be detected. This way, the attacker keeps the jamming duration short.

3.5 Experimental Evaluation

In this section, we evaluate the effectiveness and performance of the ACARS message jammer. An attacker needs to strategically position itself to facilitate relaxed power requirements for jamming and has a high probability of seeing an intersection as described in Section 3.3.6. Carefully selected and surveyed locations will minimize the costs of the attack and improve the odds of successfully executing the attack. There are 20 air traffic routing centers responsible for en-route services over the contiguous US. The attacker must also determine a portion of airspace with high intersection density. We also perform a geospatial analysis to justify the attacker's position, specifically spoofer/jammer placement. This analysis also helps the attacker to determine a suitable location for executing coordinated multi-aircraft attacks through an analysis of determining the probability of intersecting routes.

3.5.1 ACARS Jammer Performance

We evaluate the jammer performance primarily through the packet reception rate. For this, we count the number of packets transmitted and received over 10 iterations of each experiment. We first evaluate two types of jamming signal sources, i) noise jammer and ii) pulse jammer. We determine the ideal jamming signal source for the reactive jammer based on the evaluation.

Noise Jammer:

For the noise jammer, we use the random noise source from GNURadio. The noise source is configured to generate random Gaussian noise. In the jammer evaluation experiment, we evaluate the packet reception rate for each signal-to-jammer ratio (SJR) value. From Figure 3.16, it can be seen that the packet reception rate is almost 0% for 6 dB SJR.

Pulse Jammer:

Even though noise jammer seems effective in jamming the messages at higher SJR, noise jammer requires continuous transmission at the specified power level. However, with a pulse jammer, an attacker can transmit high powered pulse for a shorter duration than transmitting noise. Recall the transmitter design explained in section 3.4.1; in the modulation scheme, the presence of 1200 Hz tone indicates bit transition, and 2400 Hz tone represents the same bit. An attacker achieves jamming by transmitting a pulse that forces the receiver to compute a wrong CRC by either distorting the bit transitions or by distorting the bits themselves. To evaluate this technique, we analyzed the effect of pulse duration and power

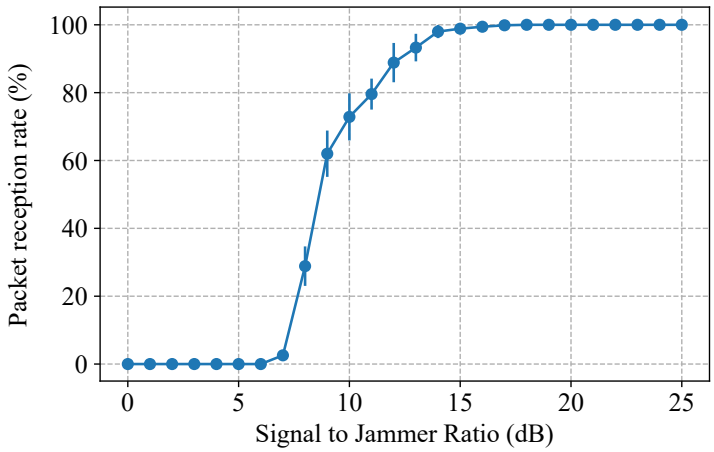


Figure 3.16: Packet reception rate for a jammer that implements a random noise source at various signal jammer ratios.

advantage on the packet reception rate of ACARS transmissions. Figure 3.17 shows the result of this experiment. An 11.66 ms pulse with a 3 dB power advantage or a 6.25 ms pulse with a 10 dB power advantage is sufficient to achieve 97% jamming success.

Reactive Jammer:

Based on the evaluation of the two jamming signal sources, we set the jammer source as random noise with 6 dB SJR. To evaluate the effectiveness of our reactive jammer, we perform an experiment where we measure the receiver's packet reception rate for messages with a length of 5 characters to 75 characters. This experiment's average packet reception rate was *98.85%* with 0.657 standard deviations. In this implementation, all the processing is done on the host PC running Ubuntu 20.04, with an 8^{th} Generation Intel Core i7 processor, and 32 GB of RAM. Since the reactive jammer is implemented on the host PC, background OS tasks, processes, and operations add latency and cause processing overhead. As a result, the reactive jammer's turnaround time is *1.48 ms* which is sufficient for successfully jamming ACARS messages. Performance of the reactive jammer can be further improved by moving reactive jammer logic to an FPGA onboard a USRP or a suitable software-defined radio as described in [243].

3.5.2 Spoofer/Jammer Placement

To model power requirements for successful jamming we use Friis's transmission eq. (3.1) that provides the received power level based on the specified transmitter configuration.

$$P_r = P_t + G_t + G_r + 20log_{10}\left(\frac{\lambda}{4\pi d}\right) \tag{3.1}$$

where P_r is the received power (dBm), P_t is the transmitted power (dBm), G_r receiver antenna gain (dBi), G_t is the transmitter antenna gain (dBi), λ is the wavelength of the carrier, and d is the distance between receiver and the transmitter. For this analysis we consider typical ground-station and airborne equipment with an output power of 25 W [44], the receiver antenna gain of 2.15 dBi [44], aircraft's transmitter antenna gain as -1 dBi [185], attacker's transmitter antenna gain as 3 dBi, and attacker transmit power of 25 W. Using these specifications[5], we calculate the SJR at the target ATSU as a function of its distance to the aircraft and the attacker. Figure 3.18 shows the SJR at various aircraft and attacker distances from the ATSU. For example, an attacker located 20 km from the ATSU can successfully jam messages from an aircraft that is as close as 4.8 km. This is sufficient because at 4.8 km the aircraft is already on the final approach and at this point, the flight crew is no longer relying on CPDLC messages. As the ratio of aircraft - ATSU distance and attacker - ATSU distance increases, the SJR at the ATSU decreases. It is important to note that, from the jammer's perspective, lower SJR means higher jammer success. This analysis also shows that it is more cost-effective for an attacker to jam at the ATSU rather than jamming at the aircraft. Thus, being in the vicinity of the airport is more beneficial for the attacker but far enough to avoid detection by commercial RF interference detection systems like [3].

3.5.3 Probability of Intersecting Routes

In order to execute coordinated attacks on multiple aircraft, it is important to identify regions that see flights whose trajectories intersect such that the aircraft is in close four-dimensional proximity of each other. To identify and analyze such regions, we make use of publicly available ADS-B data obtained through Opensky Network [206]. Opensky Network is a crowd-sourced network of over 3500 sensors that receive ADS-B, Mode-S, traffic collision avoidance system (TCAS), and flight alarm (FLARM) messages. These sensors have been providing air traffic surveillance information since 2013. The network so far has gathered over 25 trillion messages from more than 440,000 aircraft. Opensky network provides access to this vast air traffic surveillance dataset through a REST API for live data

[5] Antenna gains may change depending on the specific antenna model

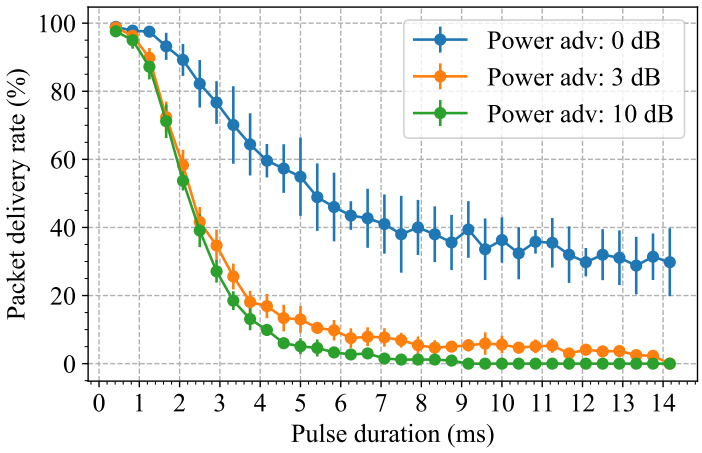


Figure 3.17: Packet reception rate for pulse jamming with varying pulse duration (ms) and power advantage (dB) that the attacker has over the legitimate signal as received by the target receiver.

and an Impala-shell over SSH for historical data. We make use of custom SQL queries to obtain aircraft pairs such that their reported positions at the same timestamp are within a 1000 m^2 bounding box. As a result of the lack of coverage, Opensky data can be noisy, hence, we further filter the data and remove duplicate records.

Next, we divide the continental US into approximately 50 x 50 km cells and map each intersection to these cells. These dimensions were chosen to keep in mind the radio coverage of the attacker. Through this analysis, we identify the most favorable geolocations for an attacker to set up and launch coordinated attacks. We perform this analysis on flight data from the year 2021, there were 592,224 total intersections such that horizontal separation between aircraft $<$ 100 m and altitude $>$ 5000 m. Figure 3.19 shows the probability[6] of seeing at least 1 intersection per day in the respective cell. 0.91% of all the intersections occur in a single cell with an average of 15 intersections a day. 6.44% of intersections occur in the top 10 cells with the most frequent intersections. This map contains an overlay of airspace boundaries of centers that provide en-route services. Based on these boundaries and the locations of ground stations, an attacker can strategically place itself close to these ground stations by combining the jammer performance data obtained from Figure 3.18 and the region-wise probability of seeing intersections data. This analysis helps us to narrow down to a region, to further improve the odds, we analyzed the hourly distribution of intersections for these top 10 cells and found out that maximum intersections occur between 12 Hrs to 22 Hrs (UTC). Through this, we determine that the attacker has a better chance

[6]Areas that do not see any intersection are marked as white.

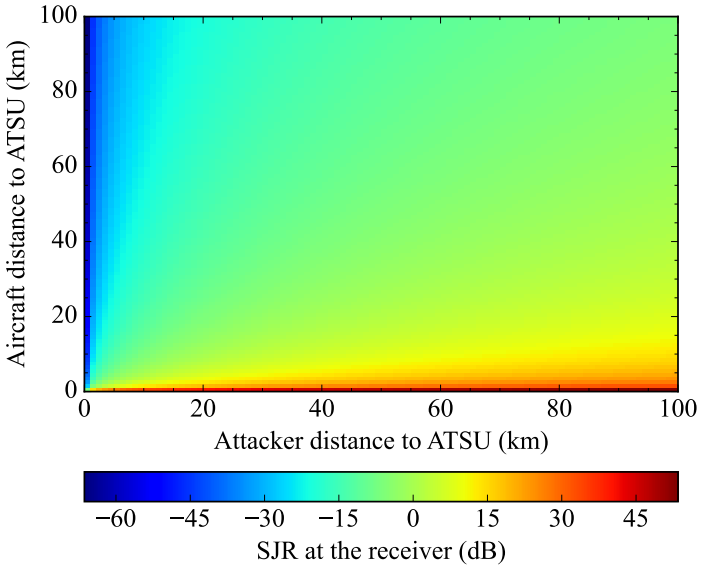


Figure 3.18: A heatmap showing the SJR) at the receiver as a function of the target aircraft's distance to the ATSU and the attacker's distance to the ATSU. (Lower is in favor of the attacker)

of succeeding if they target regions that consistently see at least 1 intersection every day.

3.6 Discussion

3.6.1 Integrated Attacks on Aircraft as a System

Aviation systems are built on multiple redundancies and fallback mechanisms. There is always one other system that the flight crew can use to complete the mission. To execute a complete takeover, it is essential to attack all these individual avionics so that an attack on one system is corroborated by another system that the flight crew may use.

Coordinated Multi-aircraft Attack:

As described in Section 3.3.6, if the attacker is successful, the approaching aircraft will trigger their respective aircraft collision avoidance system and alert the flight crew of traffic in the aircraft's vicinity. This system alerts the flight crew and provides a conflict resolution advisory that instructs the flight crew to climb or descend. Such a system will thwart the attack. However, as shown in [181, 219], such a collision-avoidance system is vulnerable to spoofing attacks. The collision system uses mode-S and ADS-B transmissions to detect

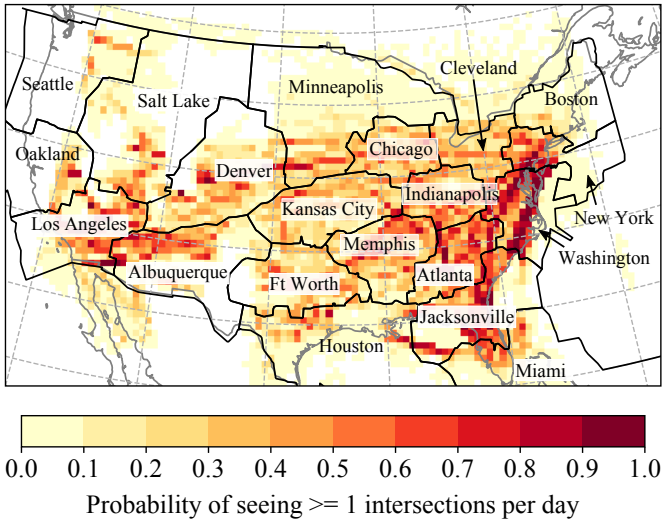


Figure 3.19: Probability of seeing at least 1 intersection per day in the US airspace. This probability map is based on air traffic data from the year 2021. It also shows the boundary of the air traffic center responsible for providing en-route services in the respective airspace.

aircraft in close proximity. In [66, 222] authors demonstrate the possibility of spoofing ADS-B packets.

Altimeter Manipulation

During the landing phase, the flight crew will rely on the ILS for providing vertical and horizontal guidance. A successful altimeter manipulation will cause the flight instrument to show incorrect altitude. However, a trained pilot might be able to detect an error in the instruments after correlating altimeter and ILS measurements. This will cause a significant loss of situational awareness, especially when landing in zero visibility conditions. However, pilots can execute corrective maneuvers and abort the landing if detected early on. The attacker can spoof ILS to enhance the attack, as demonstrated in [204].

3.6.2 Countermeasures

These attacks require precise coordination and synchronization between various attack techniques. In addition, pilots are trained to detect discrepancies in their data. Even if the pilots are instructed to follow the instruments, they rely on instincts. If contradictory to the instruments, if something doesn't feel right, pilots will usually execute fallback mechanisms as demanded by the training. A comprehensive countermeasure uses the public key infrastructure for message signing that assures the sender and the receiver's identity.

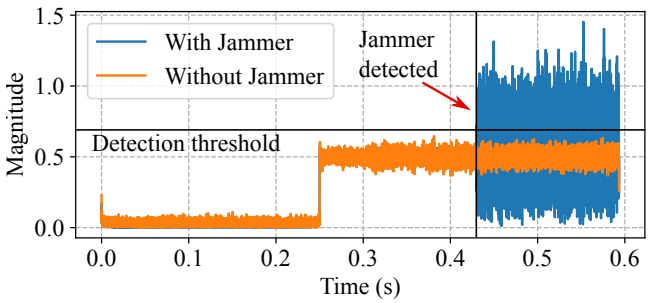


Figure 3.20: Reactive jammer detection (PoC). The receiver sets a threshold for envelop magnitude and bit error. It raises the alarm when both values cross the set threshold.

Given multiple stakeholders with different operating standards and financial support and the need to maintain seamless interoperability, implementing resource-intensive solutions like public-key infrastructure (PKI) is challenging and expensive.

The proposed reactive jammer is hard to detect as it transmits only for the duration of the original message. However, the receiver can deploy physical, and application layer checks to detect a reactive jammer by correlating bit errors and the magnitude of the signal's envelope for each bit duration. Similar to [221]. Under a reactive jamming attack, the jammer will manage to corrupt only a portion of the message. This means the receiver successfully decodes a part of the message. In a non-adversarial setting, the receiver will experience a uniform signal magnitude. However, when the reactive jammer starts, the receiver will experience a sudden increase in the envelop magnitude as well as in the number of bit errors. Figure 3.20 shows a PoC of this technique. Once jamming is detected, the controllers can contact the victim aircraft over a voice channel. The advantage of such a technique is that it can be deployed at ground stations without extensive overhaul.

3.7 Related work

The aviation industry includes a wide range of analog and digital wireless communication and navigation systems. These include simple amplitude-modulated Hf and VHF voice systems to sophisticated satellite navigation systems like GPS. Through the ages, these systems have made flying safer and more efficient than before. However, system developers and manufacturers have failed to consider security as an integral part of the system design. As a result, these systems are vulnerable to cyber-physical attacks that can cause serious damage. Strohmeier et. al. in [224] show the vulnerabilities present in modern aviation systems. In recent years, researchers have repeatedly demonstrated the ability to spoof global positioning system (GPS) [184, 28, 228, 212, 176, 246], ADS-B [66, 222, 161, 205],

ILS [204] and even collision avoidance systems [181, 219].

However, there is very limited work that investigates the security guarantees of aviation datalink applications like CPDLC and ADS-C. The work closest to ours is [216], here, the authors provide an overview of attack strategies and propose a man-in-the-middle attack that specifically targets the handover process. They also suggest cryptographic countermeasures for securing aviation datalink. Even though they suggest the need for selective jamming, they do not study the feasibility and performance of such a jammer. In our work, we not only describe specific attack scenarios that exploit certain message sets but also provide a requirement and performance evaluation of the reactive jammer that enables these attacks.

In [247], the authors present message monitoring, entity camouflage, and MiTM attacks. However, they fail to consider the required message acknowledgment and flight crew input. Such a strategy, even though effective, will lead to attack detection by the ATSU. In [83, 57] describe the possibility of using software-defined radios to enable an attacker to transmit fabricated messages. Specifically, [57] presents an attack targeting the manipulation of take-off speed recommendations, including speeds that the flight crew should use for a safe and efficient take-off. Authors in [209] present an analysis of CPDLC communications, specifically in the context of ATN B1 application that is predominantly used only in Europe. In [217], the authors specifically focus on privacy issues associated with ACARS transmissions. This shows that aviation datalink applications are vulnerable to invasive attacks that require message injection, but they also suffer from a lack of sufficient controls to preserve user privacy. This allows easy access to privileged information without much effort.

Researchers in the past have proposed various application-layer solutions to secure ACARS messages. Most notably [200, 164], in these works, authors have proposed cryptographic solutions that use a symmetrical session key for data encryption at the application layer and public key infrastructure for authenticating and validating entities. Based on these proposals, Aeronautical Radio Inc (ARINC) has developed industry standards 823 P1/2 that provides guidelines for ACARS message security (AMS) [178] Currently, AMS is used explicitly by the US military. Similarly, in [143] the authors propose a secure CPDLC scheme that leverages Elliptic curve cryptography. They evaluate their protocol and provide formal verification using a ProVerif, a security verification tool that uses the Dolev-Yao attacker model. In [216], authors also suggest various non-cryptographic countermeasures that use the aircraft's geo-location to determine if a certain connection is malicious, such non-cryptographic countermeasures provide attack detection capability.

3.8 Conclusion

In this work, we performed a security analysis of vital aviation datalink applications like CPDLC and ADS-C. Specifically, we outlined the requirements for executing a successful attack that has the potential to influence the flight crew's decision-making. We described a battery of attacks that target individual aircraft as well as coordinated attacks that simultaneously target multiple aircraft. We also performed a geospatial analysis of historical air-traffic data and identified 48 vulnerable regions where an attacker has a 90% chance of encountering favorable conditions for coordinated multi-aircraft attacks. We also proposed a reactive jammer to enable the stealthy execution of these attacks. Through experiments and real-world implementation, we demonstrated and evaluated the performance of a reactive jammer that can be programmed to jam specific messages from a particular aircraft with a reaction time of 1.48 ms and with 98.85% jamming success. Even though several works have proposed viable security mechanisms for protecting aviation datalink, aviation authorities around the world have failed to adopt these standards. And till date, these critical systems remain vulnerable and qualify as prime targets.

Chapter 4

An Experimental Study of GPS Spoofing and Takeover Attacks on UAVs

4.1 Introduction

The previous chapters of this thesis have focused on navigation and communication system used mainly by crewed aerial vehicles. This chapter evaluates the effects of GPS spoofing attacks on unmanned aerial vehicle (UAV)s. GPS is a primary navigation aid used by crewed and crewless vehicles. Today, there is a quickly increasing demand for UAVs across various civilian, military, and commercial applications, with market surveys [132] forecasting a doubling of the global retail UAV market in the next five years. Military and domestic law enforcement predominantly use UAVs for surveillance and reconnaissance operations. With their easy-to-use UAVs, manufacturers like DJI [75] and open-source platforms like ArduCopter [41] have enabled mass adoption of UAVs for civilian applications such as geographic surveys, photography, agriculture, recreational racing, package delivery, and many more.

This increased accessibility has also raised serious security and privacy concerns, especially after recent events in which civilian and military establishments were attacked using a slew of low-cost UAVs. For example, Heathrow and Gatwick airports reported several UAVs entering their airspace, significantly disrupting air traffic for several days [173, 210]. There have been reports of terror groups using consumer UAVs laden with explosives to attack critical oil facilities and an airport in the Middle East [235, 160, 72]. Moreover, given these UAVs' low-visibility profile and cross-section, conventional air traffic radar systems are ineffective against these threat vectors. This has spawned a cottage industry of counter

UAV systems, which promise reliable detection and protection against intrusions.

In general, including most of the above threat scenarios, UAVs heavily rely on the Global Positioning System (GPS) for positioning and navigation, particularly where they need to operate autonomously or in a pre-programmed fashion. GPS is an integral part of onboard decision-making that relies on positioning and navigation systems. Hence, GPS is seen as a single point of failure for UAVs. At the same time, GPS has long been known to be vulnerable to jamming and spoofing attacks. The vulnerability can be profound: GPS spoofing provides an attack vector that enables control over the target UAV without compromising the flight control software or the command-and-control radio link. Furthermore, a GPS spoofing attack can be carried out by an attacker that is equipped with an RF transmitter. Since the attacker can generate spoofing signals for any arbitrary location, an attacker's proximity to the target is limited only by the attacker's amplification capabilities. An attacker equipped with a powerful enough transmitter or directional antennas need not be in close proximity of the target.

Widely-reported demonstrations show the feasibility of diverting unmanned ships [28], cars [184], and aerial vehicles [212]. In contrast to these attacks, GPS spoofing has also been explored as an active defense strategy, e.g., safely hijacking UAVs off a protected area [176, 69]. Despite the above demonstrations and the rapidly growing importance of UAVs, there are limited studies on the feasibility of precisely controlling unmanned vehicles, specifically commercial off-the-shelf UAVs, by spoofing GPS signals. Prior work primarily focused on disrupting or altering the motion of the unmanned vehicle in a non-specific direction or performed the analysis on standalone GPS receivers. Kern et al. [141] used simulations to show the possibility of forcing the UAV in the desired direction by manipulating the GPS velocity in the opposite direction. However, this approach led to an uncontrolled acceleration in the simulated environment. Noh et al. [176] provides a taxonomy of strategies to hijack consumer UAVs through GPS spoofing. However, the discussed hijacking approaches are limited to diverting the UAV in one direction, as they don't show the ability to maneuver the UAV, e.g., change the direction after the initial hijacking.

Importantly, no previous work has examined and field-tested such a controlled takeover of UAVs in a controlled real environment outside of a simulator. This state of affairs severely limits the available knowledge on the practicalities of GPS spoofing attacks on modern UAVs. GPS measurements are often fused with measurements from various sensors like inertial sensors, vision sensors, and distance measurement equipment. Given the tightly coupled nature of the system, it is vital to examine the UAV system as a whole.

Consequently, this work aims to understand the *feasibility* and the *requirements* of fully controlling a UAV's movements by spoofing GPS signals alone. We answer the following

research questions:

1. Can an entity (adversarial or active defense) precisely control a UAV's movement by spoofing appropriate GPS signals?
2. What are the requirements and fundamental limitations of such spoofing strategies?

Specifically, we make the following contributions:

- We perform an exhaustive experimental analysis on the behavior of commercial-off-the-shelf (COTS) UAVs under a GPS spoofing attack. We execute our over-the-air spoofing experiments in a 15.24 x 15.24 x 6.7 m anechoic chamber equipped with a state-of-the-art motion capture (MoCap) system from OptiTrack [130] that offers precision tracking. Our setup enables us to characterize the response of the UAVs to different spoofing attacks for the first time in public literature. For experiments that require observing the UAV's behavior over longer distances, we use Arducopter [41].
- Based on our experiments, we enumerate several challenges in accomplishing a complete UAV takeover through GPS spoofing and controlling it without crashing. For example, even with complete knowledge of the current state of the UAV, spoofing a pre-defined static location can cause the UAV to move in an unpredictable direction.
- We design and implement a Real-time GPS Signal Generator (RtGSG) that can be configured to generate any arbitrary trajectory and is capable of making changes to GPS signals in real-time through user input. This enables us to modify the spoofing signal based on observing the UAV's reactions in real-time, giving us better control of the UAV's trajectory and speed. RtGSG can interface with multiple software-defined radio frontends and can be controlled using any peripheral device like a joystick. Our signal generator can also interface with UAV simulators (like Arducopter) and UAV tracking systems (like OptiTrack), providing detailed analysis of the UAV motion. We will release our implementation for further research.
- We evaluate RtGSG on various UAVs from DJI and Autel and analyze the degree to which we can control the UAVs via GPS spoofing. We extract both generic and UAV-specific strategies to achieve complete maneuvering control. We were able to manually control and execute patterns like 180° turns through such a system as demonstrated in the video[1].

[1]Here is the link to a video demonstration of this attack. `https://www.youtube.com/watch?v=EtaQ_BQFn-M`

- Finally, we discuss limitations and highlight that COTS UAVs remain vulnerable (e.g., can be forced to crash or diverted away) to GPS spoofing. The complete takeover and control of the UAV is challenging and requires careful manipulation of the spoofing signals in real-time.

The rest of the chapter is organized as follows. In Section 4.2, Section 4.3, and Section 4.4, we describe the UAV ecosystem and provide a background on GPS and GPS spoofing attacks respectively. In Section 4.5, we study the impact of conventional static-location spoofing and dynamic-path spoofing against consumer UAVs, present insights gained through these experiments, and lay down requirements for a complete takeover. This is followed by Section 4.6, where we implement and evaluate the real-time control strategies we develop based on the challenges and requirements identified. Then, in Section 4.7, we discuss the technical insights learned, the limitations, and the impact of our work. Finally, we provide an overview of the related work and conclude this chapter.

4.2 UAV Ecosystem

UAVs are categorized as consumer, commercial, or military. Technological advancements in electronics and manufacturing diminish the lines between these categories. For example, terror groups [190] have managed to make consumer UAVs combat-ready. Today, even COTS UAVs are capable of beyond visual line-of-sight operations with a payload capacity from 500 g up to 200 kg [39], flight speed up to 70 kmph, and flight height of more than 5 km above sea level. Irrespective of their application, UAVs generally implement the following architecture. The main components of a UAV system are the vehicle itself, the operator, a wireless radio controller, and a ground control station explicitly built for managing autonomous flight. Powerful onboard microprocessors act as flight controllers capable of sensor fusion, navigation, advanced mission planning, and safety-critical decision-making. Refer to Figure 4.1 for a schematic representation of a generic flight controller and its various components. Autonomous flight requires a programmed mission that includes a pre-defined trajectory with waypoints where each waypoint of the flight segment can have its speed and altitude profile. Even consumer UAVs come with a battery of sensors like GPS, vision sensors, inertial sensors (IMUs), and various types of distance measurement equipment (DME) that aid in navigation and position control to provide safe and efficient flight.

Typical UAVs implement a proportional, integral, and derivative (PID) controller for attitude and position control. It is ultimately responsible for driving the motors that generate thrust that moves the UAV as required. The flight controller uses the outputs of the PID controller to determine how fast the motors should spin to achieve and maintain the desired attitude and position. The sensor-fusion algorithm, which can fuse measurements

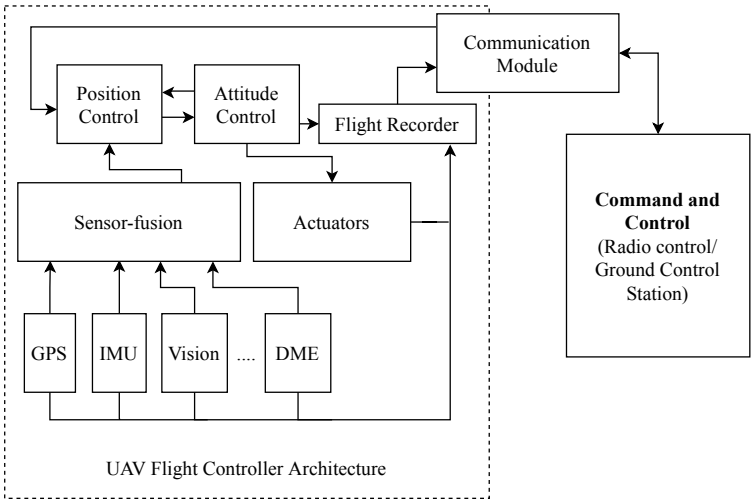


Figure 4.1: Schematic of a generic UAV flight controller architecture depicting various modules like sensor-fusion, position and attitude control, flight recorder, communications unit, and a battery of sensors.

from various onboard sensors like GPS, inertial measurement units, and opti-flow sensors, provides the input to this control loop. Some of the most widely adopted sensor-fusion algorithms are based on an extended Kalman filter (EKF).

Figure 4.2 shows a schematic of a typical PID controller implementation for horizontal position control. A typical PID controller takes the form of:

$$u(t) = k_P e\,(t) + k_I \int_0^t e\,(\tau)\,d\tau + k_D \frac{de\,(t)}{dt} \tag{4.1}$$

where $u(t)$ is required change, $e\,(t)$ is the error in desired and actual values and k_P, k_I, and k_D are the respective gains. A UAV that is set to hover, i.e., the desired location is a fixed value, can experience a drift because of two factors: i) internal measurement errors that arise because of inertial sensors that drift and other faulty measurements, or ii) external factors like wind or someone picking up the UAV and moving it to a different position. When the UAV experiences such drifts, the PID controller issues appropriate commands to actuators that control the motors. This enables the UAV to maintain its position.

These features collectively enable UAVs to carry out fully autonomous flights. Moreover, vision sensors and distance measurement equipment provide automatic obstacle detection and avoidance capability. Typically a UAV supports the following flight modes: i) Manual: The operator is in complete control of the vehicle; the flight controller does not provide any stability control; ii) Stabilize/Loiter: in this mode, the operator is responsible for position control and the flight controller assists in stabilizing the UAV by taking over when the

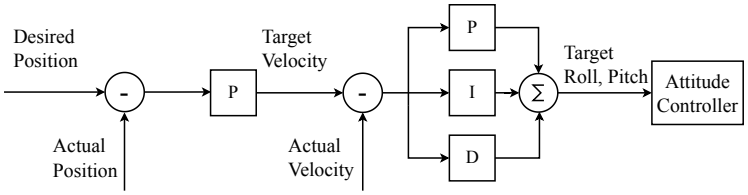


Figure 4.2: Schematic of a typical PID control sequence for position control. First, the desired and actual positions error is used to compute target velocity. Next, the controller uses the difference in target and actual velocity to set the target roll and pitch. These target values govern how fast the motors should spin to achieve the desired position and attitude.

operator does not provide any input and maintain altitude. Additionally, this mode usually has roll and pitch restrictions to maintain thrust. iii) Mission: A complete autonomous operations mode where the flight controller governs the attitude and the three-dimensional position; In this mode, the flight controller executes a predetermined mission, usually a set of waypoints in the form of GPS coordinates, with each flight segment having its speed and altitude profile; and iv) Land: This mode is usually activated at the end of a mission or as a failsafe mechanism. The UAV automatically lands at the current position or a pre-configured location in this mode, usually its takeoff or home location.

The flight controller has certain predetermined operations, called failsafes, that are triggered to ensure the vehicle's safety in case it encounters any errors in flight. These errors can result from faulty sensor measurements, loss of thrust, a malfunctioning battery, or even high-speed winds. For example, Autel UAVs will abort the current mission and land when the battery runs out of charge [46]. Often such failsafes are user-configurable. However, some failsafes cannot be overridden, even through human intervention. Such failsafes are defined as *terminal failsafes*, e.g., EKF variance in ArduCopter and no-fly zone (NFZ) restrictions in DJI drones.

4.3 Global Positioning System (GPS) Overview

GPS is the most widely used navigation system, consisting of 29[2] operational satellites roughly at an altitude of 20,200 km. These satellites continuously transmit individual satellite ephemeris and timing data, allowing a receiver to localize itself with respect to known satellite positions. GPS provides a civilian positioning service with accuracy up to 5 m on the L1 frequency [244]. 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

[2]As of January 1, 2022 - [15]

every 1 ms. Military GPS signals use a longer and secret spreading code. This work focuses on civilian GPS signals, which are widely used even in security-critical applications [215, 98]. The navigation data comprises of five subframes. Each subframe contains 1500 bits transmitted at 50 bps [53]. These subframes contain satellite clock and orbital information. The ephemeris data is updated every 2 hrs and is valid for 4 hrs [80]. Subframes 1, 2, and 3 carry the same data across each frame. The data in subframes 4 and 5 is split into 25 pages and transmitted over 25 navigation data frames. The first subframe mainly contains satellite clock information. The second and third subframes contain the ephemeris data related to the satellite's orbit and are used in computing the satellite's position. Subframes 4 and 5 have the almanac data i.e., the satellite orbital and clock information, with reduced precision. Each satellite transmits all other satellites' almanac data (subframes 4 and 5) while transmitting only its ephemeris data (subframes 2 and 3).

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 switches to tracking and demodulating the navigation message data.
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 (PVT) Estimation module decodes raw navigation bits and calculates the pseudorange between the satellite and the receiver. A receiver requires information from at least four satellites to accurately calculate position, velocity, and time. The main output of the PVT block is the receiver's position in degrees decimal (latitude, longitude) or Earth-Centered Earth Fixed (ECEF) frame and the estimated altitude and velocity of the receiver $\boldsymbol{v}$ decomposed into Easting v_e and Northing v_n velocity in m/s. 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 [97]) for further processing. While UAVs support multiple satellite navigation constellations, GPS is the typical constellation used across all UAV platforms and

manufacturers.

4.4 GPS spoofing attacks

Due to the lack of authentication and public access to satellite spreading codes, modulation techniques, and data structure, GPS is vulnerable to signal spoofing attacks which are physical-layer attacks where the attacker transmits a pre-crafted signal that contains appropriate satellite messages. When the receiver uses these counterfeit signals, the receiver calculates the position, navigation, and timing (PNT) solution initially programmed by the attacker. This deceives the receiver into believing it is at the location spoofed by the attacker rather than its actual position. An attacker can achieve this by either manipulating the navigation messages or modifying the time of arrival of these messages. Additionally, an attacker can reuse current navigation messages to make the attack stealthier.

Broadly, there are two ways of hijacking a target GPS receiver. In the first method, an overshadow attack, the attacker transmits fake GPS signals with enough power to bury the legitimate signals under the noise floor. A receiver can easily detect such an attack because of a sudden loss of lock. The second way is a more stealthy approach. The attacker first synchronizes with legitimate satellite signals. Once it is synchronized, it increases the power of its signal and then slowly starts adding code offsets that move the receiver away from its actual location. In [228], the authors provide requirements for executing such an attack. In both these attacks, the attacker's objective is to hijack the receiver and deceive it into believing it is at a location of the attacker's choosing.

4.4.1 Attacker Goals and Assumptions

In our work, we consider an attacker capable of generating and transmitting GPS signals. In attacking a UAV, an attacker's primary goal is to force the UAV to move to a specific location by spoofing GPS signals. The UAV is assumed to be within the attacker's radio range and can receive spoofing signals. We also assume that the attacker has managed to take over the UAV's GPS receiver by a seamless takeover attack, as explained in [228, 195], or through a non-coherent overshadow attack. Prior work has extensively analyzed the spoofing vulnerability of standalone GPS receivers [228, 175, 191, 138]. The received signal strength of the GPS signals on the ground is typically around -127.5 dBm and, hence, it is trivial for an attacker to overshadow the legitimate signal with the adversarial signal.

Researchers have also demonstrated the ability to steer yachts [28], cars [184], and drones [212, 176] to some extent through various GPS spoofing experiments. This work evaluates the UAV's response to GPS spoofing and strategies to control the *UAV* fully. In the following sections, we describe the limitations of fixed location and dynamic path

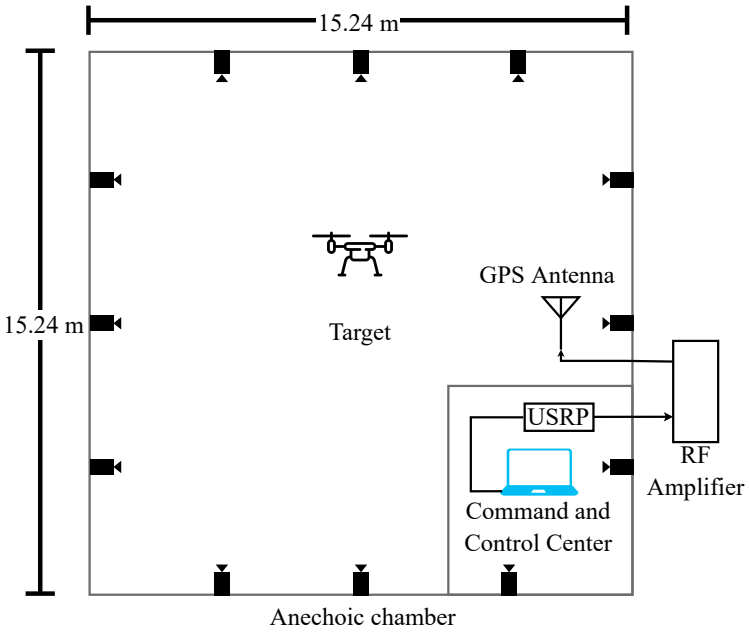


Figure 4.3: A layout of the anechoic chamber equipped with a motion capture system, GPS signal generator, and transmission antenna. The RF amplifier is installed outside the chamber to minimize EM interference inside the chamber.

spoofing attacks that rely on pre-crafted signals on UAVs and explore the possibility of asserting fine-grained control over the UAV based on UAV's retroactions to GPS spoofing and the effect of these retroactions on the process of a complete takeover.

4.5 Evaluation of Conventional GPS Spoofing Attacks

This section aims to categorize and analyze the response of UAVs to pre-defined static-location spoofing and dynamic-path spoofing. Specifically, we analyze the challenges and limitations of GPS spoofing and specify requirements to gain complete control of the target UAV. It is important to note that this work evaluates the UAV's response to GPS spoofing and strategies to take over the *UAV* and not the receiver.

4.5.1 Evaluation Setup

Transmitting GPS signals over the air in an uncontrolled setting is illegal. We perform over-the-air GPS spoofing experiments in a 15.24 x 15.24 m shielded anechoic chamber that provides more than 100 dB of attenuation. Given the tight bounds of the chamber, we are limited to spoofing experiments over shorter distances. The building materials used

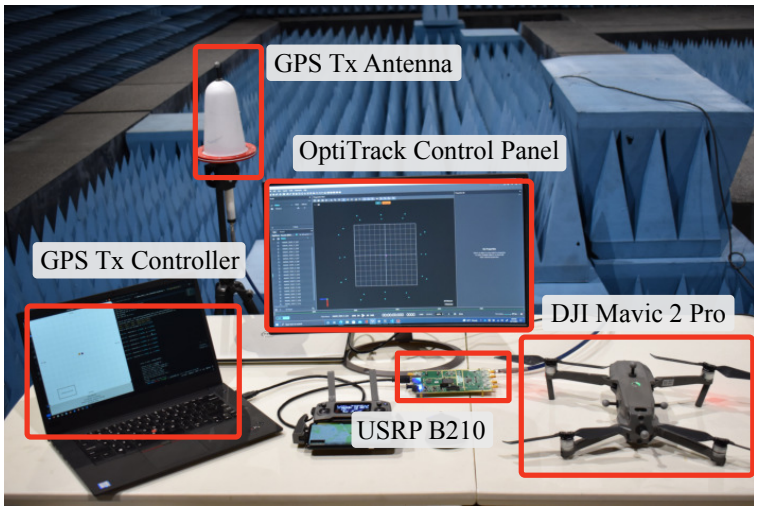


Figure 4.4: A photo of our actual setup featuring RtGSG, OptiTrack control panel, GPS Tx antenna, and DJI Mavic 2 Pro, one of the target UAVs.

in the chamber construction are a source of substantial magnetic interference; hence, the UAV requires constant calibration. Despite the trade-off between safety and realism, the shielding enables us to transmit GPS signals without running into legal issues and without the need for tethered UAV operations. Moreover, environmental factors affecting UAV's performance, like wind and temperature, are virtually non-existent inside the chamber. Reducing the effect of environmental factors ensures that the UAV's motion is affected only by the spoofed GPS locations, creating a best-case setting for evaluating the attacker's requirements to execute a UAV takeover through GPS spoofing.

The anechoic chamber is equipped with a motion capture system that runs 24 OptiTrack cameras that track objects with mm precision and provide live tracking data at 120 Hz [130]. It is important to note that the motion capture system is only used for tracking and recording the UAV's motion. A GPS signal generator [81] with a USRP B210 as the RF frontend was connected to an Ophir 5293 RF amplifier [179] that supports output power up to 50 W. The output of the RF amplifier is fed to an ETS-Lindgren's Model 3181 [84] omnidirectional antenna. Refer to Figures 4.3 and 4.4 for the schematic and the actual photo of our test setup.

We evaluate our attacks on UAVs manufactured by DJI [75] and Autel [45], shown in Figure 4.5. DJI and Autel are two leading consumer and commercial UAV manufacturers, with almost 76% market share owned by DJI alone [131]. For tests where the primary metric is distance, we used popular COTS UAV simulator software that runs ArduCopter [41] firmware along with Gazebo [197], an advanced physics and environment simulator.

Figure 4.5: All UAVs we used in our study. From top left, i) Autel EVO II, ii) DJI Mavic Mini, iii) DJI Mavic Pro, iv) DJI Mavic Air 2, and v) DJI Mavic 2 Pro.

4.5.2 Preliminary Observations

Fallback sensors and non-GPS navigation: Modern UAVs are equipped with vision sensors that can provide positioning information accurate up to 0.3 m horizontally and 0.1 m vertically [77]. UAVs typically fall back to a vision positioning system in a GPS-denied environment. We began our experiment by placing the UAV in the center of the test area and instructed the UAV to take off and maintain an altitude of 2 m. Once we visually verified that the UAV was stable, we started introducing a motion to the generated signals. In this experiment, the spoofed GPS signal introduces a motion such that the receiver believes it is moving along a path 254 m long with a maximum speed of 5.4 m/s. Despite introducing this motion, the target UAV did not budge and hovered steadily in place. The UAV reacted to GPS spoofing once we turned off the vision sensors. As a result, the UAV reacted violently with rapid acceleration to our spoofing. Figure 4.6 shows the result of one such test where one can clearly see the UAV's position (as tracked by the motion capture system) being stable as opposed to a change in GPS measurements. From this experiment, we conclude that the target UAV prioritized vision sensor measurements for positioning and navigation over GPS measurements. This shows that a UAV can survive a GPS spoofing attack by relying on other available sensors.

However, there are some limitations associated with vision sensors; these sensors require optimal lighting conditions and can provide accurate guidance only up to an altitude of a few meters. For example, DJI Mavic 3, the latest UAV from DJI, provides vision positioning only up to a height of 18 m and with flight speed ¡ 6 m/s [78]. To evaluate the effect of GPS spoofing on a UAV switching from vision to GPS positioning, we disabled the downward

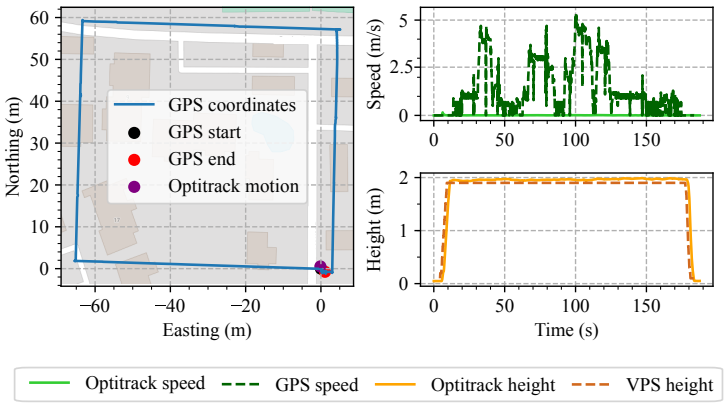


Figure 4.6: Comparison of UAV's GPS measurements and actual motion. The UAV manages to hover despite introducing a motion that takes the GPS receiver for a ride. Ground speed as calculated by the UAV's GPS receiver shows a maximum speed of 5.4 m/s while the ground speed calculated from OptiTrack data is constant at 0 m/s.

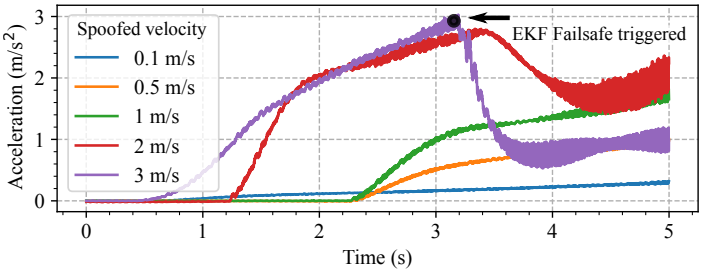


Figure 4.7: Results of first 5 s of the UAV's reaction to different spoofed GPS velocities. For higher GPS velocities, the UAV's more aggressive response is evident from the steep rise in the acceleration values.

vision sensors in-flight to simulate a scenario where the UAV is flying at an altitude greater than 18 m. As soon as we disabled the downward vision sensors, the UAV reacted by accelerating rapidly, eventually crashing into the RF energy-absorbing foam. This incident prompted us to find a suitable target velocity that would allow us to observe the UAV's reaction without creating a safety hazard. We tested multiple target velocities and narrowed them down to a suitable velocity using the UAV simulator. For this, we used a trial-and-error method for different velocity configurations. The initial acceleration of the UAV is directly related to the spoofed GPS velocity and is evident from Figure 4.7.

Terminal failsafes: Some autopilot software like ArduCopter and PX4 implement what we define as a *terminal failsafe*. When such a failsafe is activated, the UAV switches to

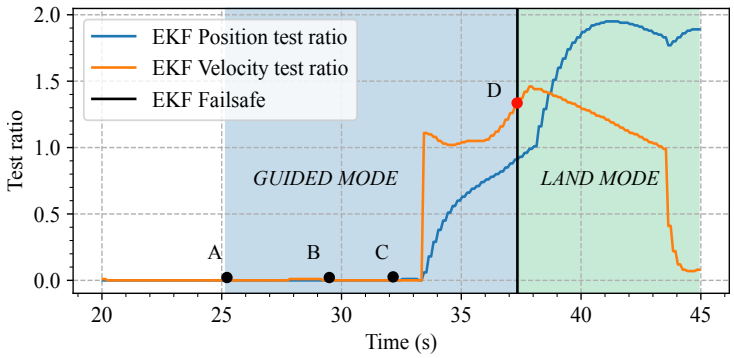


Figure 4.8: EKF variance test ratios as calculated by ArduCopter. At point A, the UAV arms and starts to take off; at point B, it completes takeoff. At point C, the attacker starts spoofing west at 2.5 m/s, and at point D, the UAV activates a terminal failsafe and permanently switches to *LAND* mode.

LAND mode. The flight controller calculates the position and velocity test ratios using EKF innovations after sensor fusion and triggers a failsafe if these test ratios exceed a predetermined threshold [42]. Figure 4.8 shows the effect of GPS spoofing on EKF test ratios where GPS velocity is set to 2.5 m/s. In ArduCopter, when the flight controller switches to *LAND* mode, it still tries to maintain a horizontal position by relying on GPS. Depending on the altitude of the UAV, the attacker has minimal time to control it further.

4.5.3 Impact of Spoofing a Static Location

Despite all the sensors and non-GPS navigation systems that can be incorporated into a UAV, GPS remains the most crucial navigation system. Moreover, unlike non-GPS systems, it also poses a more significant threat. Even a naive adversary that can only transmit static location can cause considerable damage to the UAV. In this experiment, we evaluate the final bearing of the UAV with respect to the takeoff position and the distance that the UAV travels before it loses thrust or till any failsafes are activated. These experiments are conducted in a simulator and real-world settings.

The UAV is programmed to hover at a certain location in this experiment. The attacker spoofs the UAV's actual location, a single static location, i.e., the spoofed place remains unchanged throughout the attack. The attacker's objective in this attack can be to force the UAV to stop and hover. Even though this seems benign, our experiments found that the UAV's response is unpredictable and uncontrollable. Flight controllers use EKF-based sensor fusion algorithms for state estimation, which provides the UAV with increased stability during flights. The UAV's uncontrolled movement can be attributed to the lack of correction required to control the drift and biases that develop in inertial measurements.

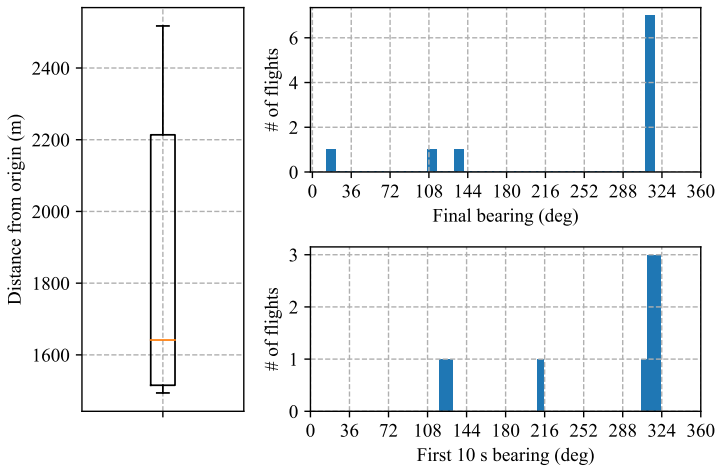


Figure 4.9: A comparison of 10 simulated flights where the attacker executes a naive static single location spoofing attack. The box plot shows the statistics of the distance covered by each flight before a terminal failsafe is activated.

Because of the IMU drifts, the position and velocity estimates obtained from EKF differ from GPS measurements. As a direct result of the discrepancy in these two measurements, the flight controller accelerates to compensate for the difference. Since the PID controller implements a feedback loop, the errors propagate and force the flight controller to make more drastic corrections.

We configured the GPS signal generator to transmit a fixed static location to execute this attack. ArduCopter starts drifting and eventually triggers an EKF variance failsafe due to position and velocity error accumulation. Once the flight controllers trigger the EKF failsafe, the UAV switches to *LAND* mode and aborts any ongoing mission. For this experiment, our evaluation metrics are the distance the UAV travels before an EKF failsafe is triggered, the final bearing of the UAV, and the bearing in the first 10 seconds of the flight. The difference in the final bearing shows how unpredictable such an attack can be. The results of these flights are summarized in Figure 4.9. The average flight distance was 1861.83 m with a standard deviation of 406.39 m.

This shows that not only is the direction random and uncontrollable, but the distance it covers is also unpredictable. This makes such an attack unreliable, especially if the attacker requires the UAV to reach a specific location. A similar experiment was performed on DJI Mavic 2 Pro; given the space constraints of the anechoic chamber, the evaluation metric was just the bearing. The results of this experiment are summarized in Figure 4.10. Unlike the results of the simulator, the real UAV shows much variation in terms of final bearing. Environmental factors deeply affect real sensors and often require re-calibration for normal

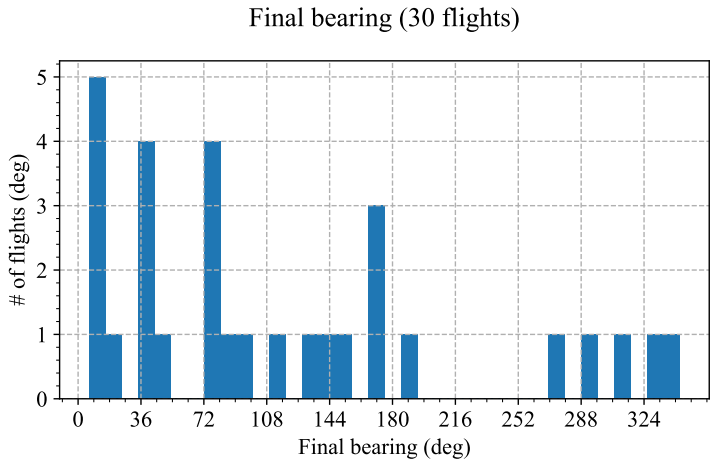


Figure 4.10: A comparison of 30 flights with a naive attacker. The target of these attacks is a DJI Mavic 2 Pro. The final bearing of these flights shows that the UAV's behavior is unpredictable and uncontrollable.

operations. Such factors generally do not apply to the built-in configurations available in simulators, illustrating the limits of their utility.

4.5.4 Impact of Spoofing a Dynamic Path

In this experiment, our goal is to analyze and understand the behavior of a UAV subject to dynamic-path spoofing. We evaluated this attack entirely on real UAVs with live over-the-air GPS signals. After takeoff, the UAV is set to hover at its current location. The attacker's objective is to transmit a signal that forces the UAV to move away from its current position in the direction of the attacker's choosing. In this attack scenario, we pushed the GPS receiver away from its original position by generating a spoofing signal that moves in a specific direction.

In a dynamic-path spoofing scenario, the attacker adds velocity to the spoofed locations after a successful GPS takeover and deceives the UAV into perceiving that it is moving with a heading of $\alpha°$. This activates the attitude and position control mechanism and forces the UAV to move in the opposite direction, i.e., $(\alpha - 180)°$. The UAV's reaction to such an attack is shown in Figure 4.11. Consider a UAV at point A; the attacker introduces a GPS signal that deceives the UAV's GPS receiver into believing it is moving toward point B. As a result, the UAV starts moving toward point C. For evaluating the dynamic path spoofing scenario, we executed five flights in each of the four directions, i.e., north, south, east, and west w.r.t to the origin. Based on the results of our vision to GPS positioning transition experiment described in Section 4.5.2 and the tight space constraints, the magnitude of

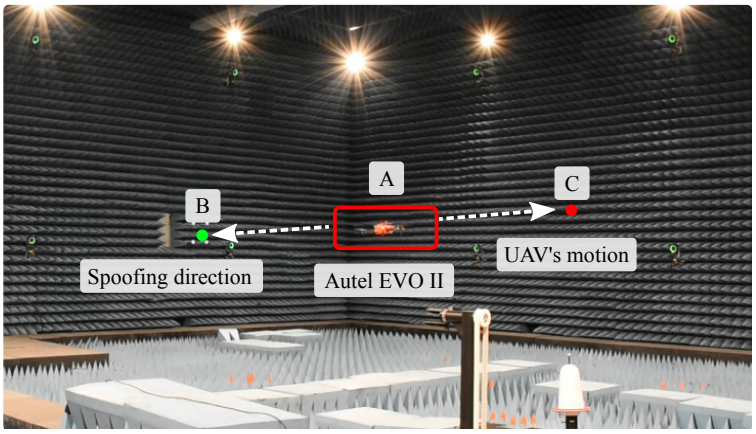


Figure 4.11: Target UAV's response to dynamic path spoofing. When subjected to a GPS signal that simulates a motion in the direction of point B, the UAV responds by moving in the direction of point C.

the spoofed velocity was set to 0.1 m/s. The sequence of each flight was as follows: i) the UAV takes off, ii) once stable, the operator switches to "GPS Only" mode to simulate higher altitude, iii) the spoofer is activated, and iv) as the UAV gets closer to the walls, the operator intervenes and lands the UAV manually. Figure 4.12 shows the response of the target to a spoofed velocity vector $\boldsymbol{v_{en}} = [-0.1, 0]$ that forces the UAV to fly east. Figure 4.13 shows the error in the final bearing of all 20 flights. In all these experiments, we observed that the UAV reacts as expected and goes in the expected direction with an average error of 2.56°. As a next step, we introduced a second change in direction. Specifically, we changed the direction of the spoofed trajectory by 90°. This strategy showed limited success; only 3 out of 17 flights followed the required change in bearing. Moreover, without any velocity control, the target flies a curve, making it impossible to achieve sharp turns because of the momentum that the UAV has already developed due to the spoofing attack.

4.5.5 Key Insights and Lessons Learned

The UAV's response to GPS spoofing can be attributed to the correction maneuver enforced by the position, and attitude control described in Section 4.2. The PID controller responds to the changes in the UAV's actual position and velocity measurements derived from sensor fusion by providing control inputs to compensate for the error between the desired and the actual position. Recall Equation (4.1), over time, as a result of the integral $(k_I \int_0^t e(\tau)\, d\tau)$, the errors are magnified, and the corrections to even minor errors get more aggressive. As a result, the UAV tries harder to overcome the error. Since the spoofed location is consistently going away from the original position, the UAV keeps increasing its velocity due to the aggressive corrections explained earlier. In Figure 4.12, the target's velocity

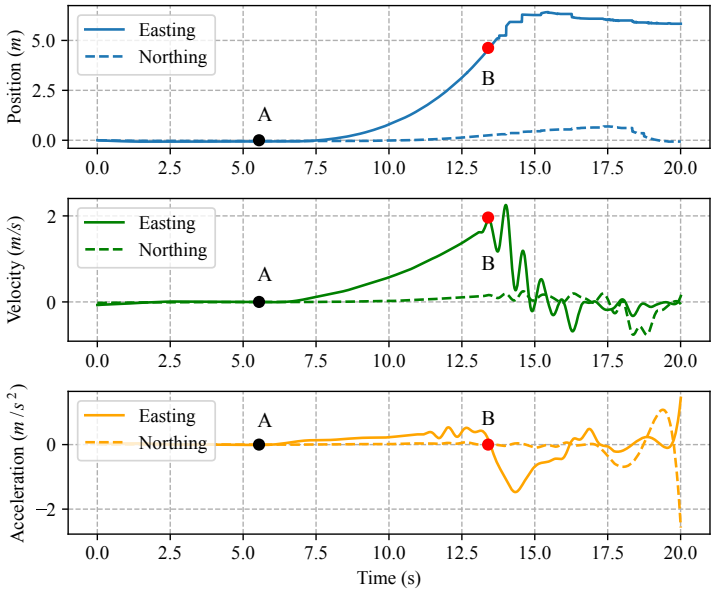


Figure 4.12: position, velocity, and acceleration data for the takeover of a DJI Mavic 2 Pro. The position data was obtained from a motion capture system. At point A, we introduce a motion as $\boldsymbol{v_{en}}$, and at point B, the operator intervenes once the UAV gets closer to the wall.

keeps increasing until the operator takes over.

The UAV achieves the required acceleration by manipulating its pitch and roll. The thrust value $\boldsymbol{T}$ required to maintain its altitude and to stay afloat is given by

$$T = \frac{mg}{cos\theta cos\phi} \tag{4.2}$$

where θ and ϕ are pitch and roll, respectively. According to its dynamics, the quadcopter fulfills the acceleration requirement by manipulating the pitch and the roll. As the magnitude of corrections increases, the magnitude of required acceleration also increases; hence, the UAV keeps increasing the pitch and roll to fulfill the acceleration requirement. In order to achieve the desired acceleration, the UAV tilts[3] so much that it is no longer able to generate enough thrust to keep itself afloat. As a result, it will lose altitude and crash. From this, we conclude that even if the UAV goes in the specified direction with minimal directional errors, the attacker's control over the target is limited only to the UAV's direction. The attacker has no control over UAV's speed. From these experiments, we understand that:

[3]A combination of pitch and roll.

- Spoofed GPS velocity induces acceleration in the UAV
- A UAV will continue accelerating in the initially spoofed direction until it runs out of battery or crashes
- Complete control of the UAV requires direction control as well as speed control

As explained Section 4.5.4, in such an attack, the attacker can *only* control the *direction* and not the speed. Even if the attacker changes the direction of the spoofed trajectory, the attacker is unable to change the direction of the UAV.

Consequently, we establish that the attacker should be able to force the UAV to execute four specific maneuvers to constitute a complete takeover of the UAV. These are i) flight with constant direction, ii) flight with constant velocity, iii) flight with variable direction, e.g., the ability to make tight turns in an environment with strict mobility constraints like an urban setting, and iv) land. These specific maneuvers ensure complete control over the direction of the UAV and the distance it travels. The attack strategies proposed by various researchers in the past do not fulfill these requirements, making the proposed attacks uncontrollable. Moreover, terminal failsafes like EKF variance bounds make it even more difficult to exert control over the target UAV effectively. We identified that for a complete takeover, we need: i) a GPS signal generator capable of user-controlled real-time trajectory manipulation and ii) a strategy for controlling the acceleration of the UAV. In the following section, we outline the strategies we developed to address the challenges mentioned above through experiments.

4.6 Real-time Control of UAV via GPS Spoofing

Based on the insights described above, we conclude the requirement for dynamically manipulating GPS signals in real time based on a UAV's response to the spoofing signal. In this section, we present our real-time GPS signal generator that addresses this need. We also propose and evaluate strategies for post-takeover direction and velocity control.

4.6.1 Real-time GPS Signal Generator

Commercially available off-the-shelf hardware and open-source software GPS signal generators are often limited to predetermined trajectories, i.e., they are not capable of user-controlled real-time trajectory manipulation. This is a requirement that we identified in the previous section. To facilitate this requirement, we built the *Real-time GPS Signal Generator* (RtGSG), a GPS signal generator system that allows real-time trajectory manipulation.

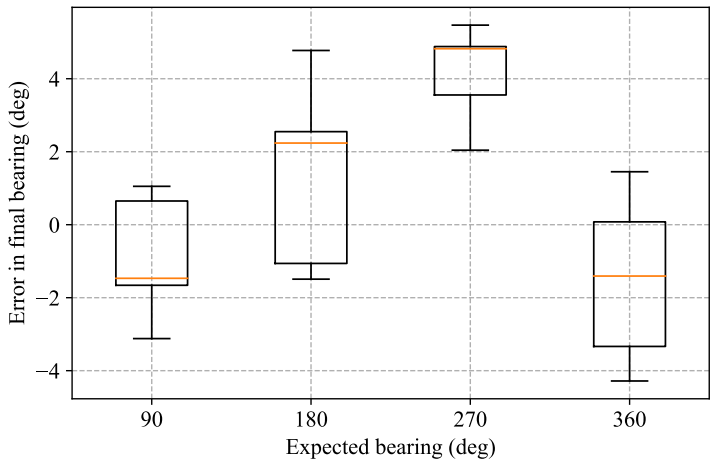


Figure 4.13: A boxplot of error in final bearing against the expected bearing. Five flights were carried out for each expected bearing. A trajectory was generated as described in Section 4.5.

This system is based on an open-source GPS signal simulator, *GPS-SDR-SIM* [81]. RtGSG comprise three main components as shown in Figure 4.14: i) feeder, ii) GPS signal generator, and iii) RF front-end.

The feeder can accept a set of predefined trajectories in the form of a time series that contains positions $P = \{p_1, p_2, p_3, .., p_n\}$ where $p_i = (latitude, longitude, height)$ or an initial location and velocity vector[4] $\boldsymbol{v_{en}}$ as $[ve, vn]$ (*Easting and Northing*) or as speed and bearing. The feeder can also accept velocity vectors through human interface devices like a keyboard or a joystick. This is especially useful in our *Human-in-the-Loop* (HITL) spoofing system where a user can manipulate the GPS signal using a keyboard or joystick, similar to playing a video game. The feeder is responsible for computing the correct location and time and updating the GPS signal generator. The GPS signal generator receives the locations, constructs the navigation message using the supplied satellite ephemeris data in RINEX [101] format, modulates the message, and generates raw IQ samples. Following the generation of IQ samples, the RF front-end module interfaces with software-defined radios like USRPs [6] and LimeSDRs [154] that can transmit the generated samples in real time. It is important to avoid any type of hardware-dependent sample drops as they can cause the target receiver to lose the GPS lock. Sample generation and consumption should be synchronized to avoid buffering IQ samples that can lead to position/time jumps. These modules enable RtGSG to manipulate trajectories and transmit the generated signals on-demand, instantly, and precisely. We will release RtGSG to the scientific community for

[4]We use the following units: all positions are in decimal degrees, velocity is in m/s and bearing is in degrees w.r.t to north, unless stated otherwise.

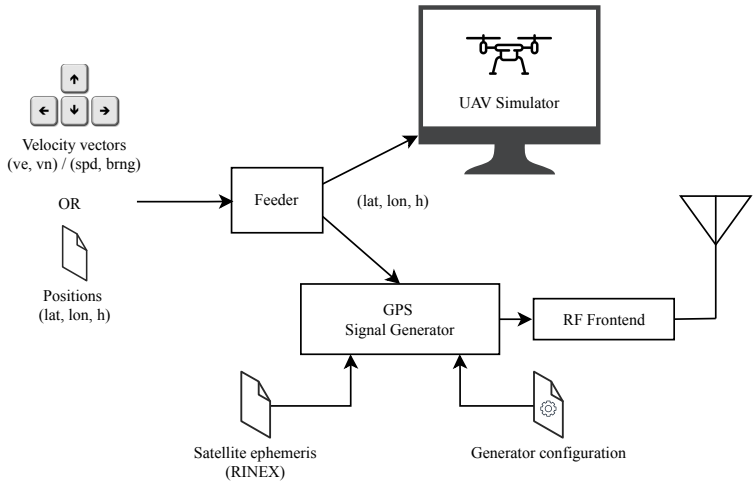


Figure 4.14: A schematic of RtGSG's architecture showing the feeder capable of using positions or velocity vectors with an optional link to send the coordinates directly to a UAV simulator, the GPS signal generator that generates raw IQ samples and interfaces with an RF frontend capable of transmitting the generated GPS signals.

further research.

4.6.2 Strategy for Velocity Control

As demonstrated in Section 4.5.4, a conventional fire-and-forget attack with predetermined trajectories ensures that the UAV goes in a specific direction, but the lack of speed control makes the UAV uncontrollable. At a high level, our system achieves velocity control by deceiving the target into believing that the correction has worked. Then, due to the correction maneuver it executes, it approaches the "target position". However, since the integral term responds to errors from the past, it can often overshoot the target. A PID controller is designed to compensate for this and handle overshoots.

From the lessons learned in Section 4.5.5, GPS velocity induces acceleration in the target UAV, and the direction of the acceleration vector depends on the direction of the GPS velocity vector with respect to its original position. Hence, as the spoofed location approaches the target position, the UAV gradually decelerates. However, the UAV does not immediately stop its motion when the onboard receiver indicates it has arrived at the target position. As a result of initial spoofing, the UAV has already gained momentum and needs to overcome that. On the other hand, even if the attacker "jumps" to the original location, the UAV faces a similar issue, and hence the UAV continues its motion. Thus, the attacker needs to simulate a motion that resembles a UAV's correction maneuver.

In reality, the UAV can be considered as a black box, and hence the attacker does not have access to the correction model implemented by the target UAV. To overcome this

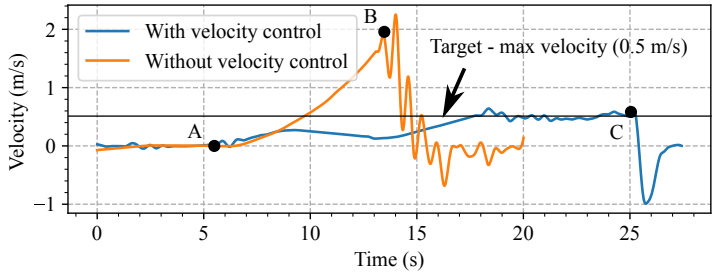


Figure 4.15: A comparison between two test flights with and without velocity control. At points B and C, the operator takes over. Without velocity control, the UAV keeps accelerating.

issue, we came up with a strategy to identify the reaction time of the UAV. The attacker can learn the reaction time by observing the target's response during the course of the attack, specifically the time it takes to achieve the spoofed velocity. Alternatively, the attacker can also estimate this value using quantities such as maximum angular velocity, the maximum tilt angle, and the total thrust that the UAV can generate. Product specifications and data sheets make these values readily available for consumer and commercial UAVs. Moreover, consumer and commercial UAVs are mass-produced, and hence they have set standards that make variations amongst these values within a specific model unlikely. However, certain environmental factors and biases unique to individual sensor units may affect these values.

For a spoofed velocity of $\boldsymbol{v_{at}}$, the target UAV responds by tilting and accelerating to catch up with the GPS velocity. The UAV experiences a lag in achieving the target velocity. We define this lag as the UAV's reaction time. However, the UAV's objective is to correct the position error. Therefore, even if the UAV achieves the target velocity, it is still away from the target position, and hence it continues the correction. This reaction time provides us with the average acceleration of the UAV. We use this value to time the spoofed GPS signal's *return to launch* (RTL) maneuver where the spoofed trajectory starts moving back to the initial position rather than away from it. This is done by changing the direction of $\boldsymbol{v_{at}}$ by 180°, i.e., the new spoofed velocity $\boldsymbol{v\prime_{at}} = -\boldsymbol{v_{at}}$. This maneuver causes the UAV to decelerate. We identified that a well-timed spoofed GPS velocity-induced acceleration/deceleration routine could be used to control the velocity of the UAV — the key is to maintain an average acceleration of 0 m/s^2. In Figure 4.15, we show post-takeover velocity control in action by maintaining the UAV's velocity under a maximum target velocity of 0.5 m/s. To further analyze the effectiveness of this technique, we perform 48 flights that make use of the same reaction time value to control the velocity, and the results are shown in Figure 4.16. For this experiment, the post-takeover UAV velocity was set to 0.5 m/s. In 56.25 % of all 48 flights, we managed to maintain average acceleration below

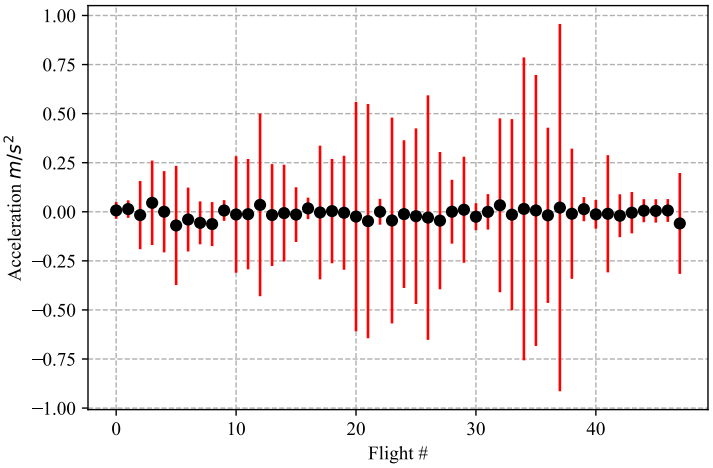


Figure 4.16: The average and standard deviation of instantaneous acceleration values of 48 flights with our velocity control algorithm. Flights with near-zero average acceleration achieve constant velocity.

0.015 m/s^2. For 81.25% of the flights, we were able to maintain an average $|\boldsymbol{v}|$ below the set maximum target velocity of 0.5 m/s with average observation time[5] of 19.4 s.

4.6.3 Strategy for Direction Control

As mentioned in Section 4.5.4, just changing the direction of the spoofed trajectory is not sufficient because of the momentum that the UAV already has gained. To make a controlled turn, it is vital to first null the UAV's velocity components v_e and v_n. In other words, the UAV must be stopped momentarily to enable sharp turns. The velocity control mechanism that we developed earlier was used to decelerate the UAV and reduce its speed to 0 m/s before changing direction. To achieve this, we transmit a GPS signal that forces the UAV to decelerate for a longer duration, enough for the UAV to get its velocity to stay close to 0 m/s. Figure 4.17 shows the deceleration sequence that we developed. Notice how the spoofed V_e shifts between -0.1 m/s and 0.2 m/s. Figure 4.18 shows a representative flight where we employ the deceleration sequence to force the UAV to make a sharp 90° turn towards north.

4.6.4 Human-in-the-Loop (HITL) GPS Spoofing

Through all our experiments, we learned that precise real-time control requires us to control the acceleration of the target UAV. When the reaction time strategy fails, or the attacker

[5]The time duration of the active spoofing attack is the time for which GPS signals entirely control the UAV.

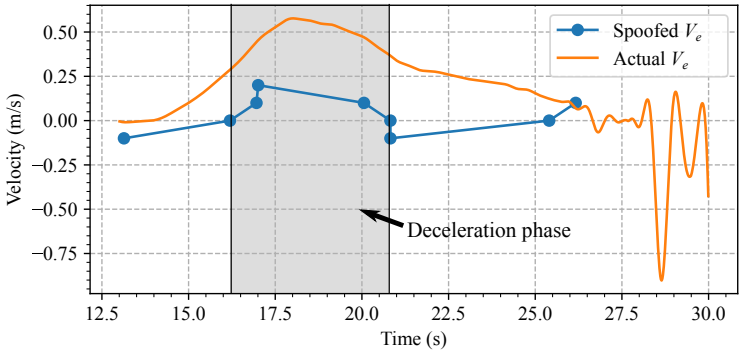


Figure 4.17: Effect of the proposed deceleration maneuver on the velocity of the target UAV. Notice the downward trend of the velocity post deceleration maneuver completion. An attacker can control the rate of deceleration by controlling the spoofed velocity.

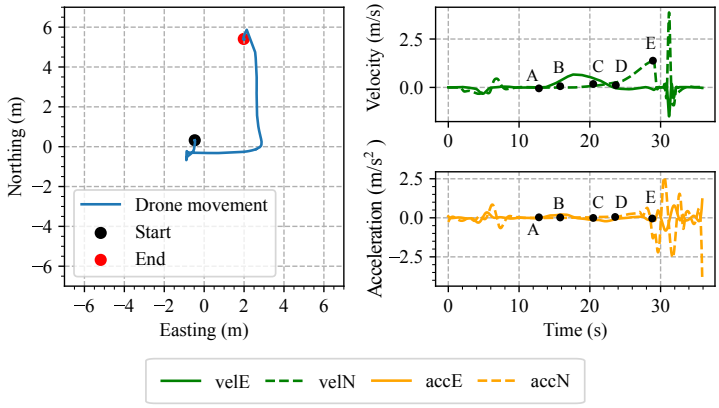


Figure 4.18: The target is executing a controlled sharp 90° turn. The attacker first forces the UAV to fly east. Then, at point B, the attacker starts the deceleration sequence, and at point D, the attacker forces the 90° turn.

cannot enumerate the reaction time, we need a feedback mechanism to observe the target's response to GPS spoofing and manipulate the spoofed trajectory accordingly. With the lessons learned through our spoofing experiments and the real-time capabilities of the GPS signal generator, we finally built and tested a Human-in-the-Loop (HITL) feedback system.

We have explored the possibility of using human intuition and knowledge of the target system to effectively control the target UAV, similar to a video game. To the best of our knowledge, such a system is a first-of-its-kind system designed to control a UAV via GPS spoofing. The HITL control system leverages RtGSG's ability to manipulate spoofed signals in real time through a human interaction device. In our experiments, it was a standard keyboard with arrow keys, but the device is interchangeable. This system relies on the attacker's observation of the UAV's movements and requires human intervention.

In our HITL system, the attacker uses the arrow keys to introduce a velocity vector in the form of $[v_e, v_n]$. This velocity vector governs the signal that the signal generator generates in real time. We modified the peripheral interface to reflect the operator's intentions and not directly apply the inputs to the spoofing signal. This interface shows the spoofed location and optionally the target's location if a system capable of tracking UAVs is available. The attacker then manipulates the velocity vectors based on their obtained understanding of the UAV's model, gaming skills, and intuition as to the UAV's possible reaction. Figure 4.19 shows one such flight where the operator takes manual control of the target UAVs and forces them to make controlled maneuvers, purely through GPS spoofing[6].

Limitations: The main objective of the HITL system is to provide control over the UAV just like a traditional remote-controller, but through GPS spoofing. In the case of a regular controller, the control inputs directly actuate the motors. However, in the case of HITL GPS spoofing, the motors are actuated through the vehicle's attitude and position correction mechanism. This is the primary difference between controlling the UAV via GPS spoofing and controlling using a regular controller. Initially, one of the main challenges of operating the UAV indirectly through GPS spoofing is understanding that, e.g., spoofing signals that move right will result in the UAV drifting to the left. This requires the attacker to understand the motion dynamics of the UAV under GPS spoofing, and it requires training to maintain control of the UAV. Additionally, it also requires good hand-eye coordination

Furthermore, due to parallax misconceptions and response time delays, controlling the UAV via spoofed GPS signals is a much more challenging task than directly controlling the UAV through the original controller. Furthermore, the attacker needs to have a mechanism other than their own eyes to track and observe the target UAV. In other words, we believe that an automated closed-loop system would be an ideal way to execute a perfect takeover because such a system will overcome the discussed limitations.

4.7 Discussion

Forced Landing: The strategies we tested demonstrate an attacker's control over the horizontal position and velocity of the UAV. GPS spoofing can not be directly used to manipulate the height of the UAV and force it to land as the majority of the UAVs rely on non-GPS sensors like rangefinders, downward-facing cameras, and barometers for measuring the altitude This poses a significant challenge because all these sensors are immune to GPS spoofing. For complete control of the UAV, the attacker should also land the UAV. As has been demonstrated earlier [111], an attacker can leverage the terminal failsafe that UAVs

[6] A video demonstration of this attack is available at `https://www.youtube.com/watch?v=EtaQ_BQFn-M`

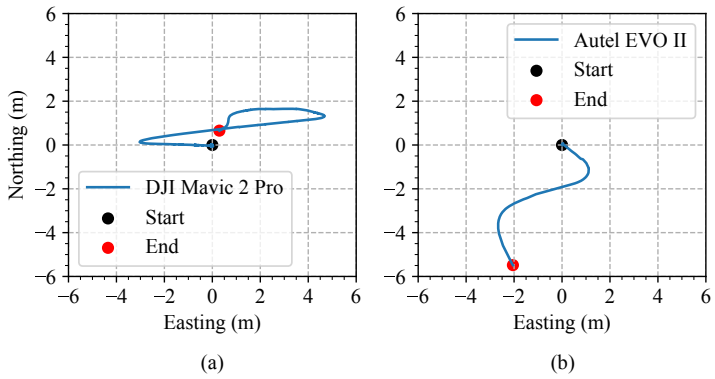


Figure 4.19: Plots (a - DJI Mavic 2 Pro and b - Autel EVO II) show the control using a Human-in-the-Loop control system.

implement to induce a forced landing. Some common events that trigger terminal failsafe are i) restricted zones and ii) EKF errors. Most consumer UAVs implement special geo-fencing around designated no-fly zones (NFZ) [76]. These restrictions prevent the UAV from taking off when the location is inside the no-fly zone. If the UAV accidentally enters a no-fly zone, the flight controller activates the terminal failsafe and lands after a warning. The operator can only control the UAV's horizontal position during this process.

The attacker's strategy thus is to spoof the GPS to a location inside the nearest no-fly zone in order to land the target. Suppose the no-fly location is more than 100 m away from the last spoofed location. In that case, the target temporarily loses the GPS lock but reacquires the attacker's signal within 20 s and initiates the landing sequence. In such a situation, time-to-first-fix (TTFF) for the onboard GPS receiver is typically between under 10 s as the receiver undergoes a warm start [230]. Even if the spoofed location is farther away and a warm start is not possible, the receiver performs a cold start, in which case the TTFF for a typical uBlox receiver 24-28 s [230]. Most of the UAVs that we tested were vulnerable to such an NFZ-forced landing attack. Similarly, a UAV that implements an EKF failsafe can be forced to land by spoofing a motion that causes the position and velocity test ratios explained in Section 4.5.2, to cross the set threshold. It is important to note that not all manufacturers enforce such a failsafe, so such strategies to make the UAV land cannot be applied to every UAV. Table 4.1 summarizes the results of our experiments.

Limitations: Of the UAVs available to us, only the DJI Mavic 2 Pro, DJI Mavic Pro, and Autel EVO II allowed us to toggle downward vision sensors. This is an essential requirement for ensuring safety when testing inside the anechoic chamber, as often the UAV can behave erratically and crash. We tested other UAVs the DJI Mavic Mini and DJI Mavic Air 2,

Table 4.1: A comparison of GPS takeover success and forced landing strategy success.

Model	GPS Takeover	Forced Landing	
		NFZ	EKF Failsafe
DJI Mavic Mini	Unable to test	✓	x
DJI Mavic Air 2	Unable to test	✓	x
DJI Mavic Pro	✓	✓	x
DJI Mavic 2 Pro	✓	✓	x
Autel Evo II	✓	x	x
3DR IRIS (sim)	✓	x	✓

both of which primarily use vision sensors below a certain altitude. However, these models switch to GPS positioning above a certain altitude, which is greater than the height of the anechoic chamber. Since it is illegal to transmit GPS signals in the open air, we could not test these UAVs. But based on our observations and the architecture of these UAVs (which are also from the market leader DJI), it is safe to say that the strategies we proposed apply to UAVs beyond the ones we tested.

In this experimental study, we target GPS. However, most modern UAVs are capable of multi-constellation localization. i.e., they use other satellite navigation systems like GLONASS, Galileo, and BeiDou. Such receivers can continue operations even when GPS is compromised. Multi-constellation receivers pose a challenge for successful spoofing and can be seen as a safeguard against GPS spoofing; however, these systems are known to be as vulnerable to signal spoofing attacks as GPS. Thus a multi-constellation GNSS signal generator can be used by an attacker to overcome this limitation as described for example in [176].

Self-learning Feedback Mechanism: The self-learning feedback mechanism for controlling the spoofed trajectory requires access to precision UAV tracking equipment capable of UAV localization in real-time, which serves as a source of the ground truth. In this system, Kalman-filter-based state estimation can be used to derive the target system's instantaneous acceleration and velocity through position information supplied by the tracking equipment. These values will be input to a predictive engine that can generate a trajectory capable of moving the UAV to the desired location. However, the attacker must consider the tolerable lag between the UAV's actual motion and the spoofed coordinates in this mechanism. This is especially applicable to UAVs that implement an EKF failsafe. Several works propose techniques to track UAVs, including acoustics sensors [59, 63], and works like [102, 68, 87] offer passive RADARs for tracking quadcopters. In [79], the authors suggest using a seeker

UAV equipped with radios for target localization.

Impact of UAV Takeover: The UAV takeover strategies we propose in this work are capable of fine-grained direction and speed control of the target UAV. These capabilities allow an adversary to commandeer a UAV remotely to turn it rogue. Even a single rogue UAV, whether it is executing an autonomous mission by itself or as part of a swarm of UAVs, poses a significant security threat. Thus, we posit that GNSS as an attack vector must be considered systemically in conjunction with inertial, vision, and other sensors in future UAV security design.

On the other hand, precise control over a rogue UAV can help eliminate and investigate the threat, as the proposed techniques can be used defensively to get the rogue UAV to a safe location for further investigation.

Countermeasures: Current state-of-the-art countermeasures can be categorized as cryptographic solutions, physical and application layer solutions, and solutions that leverage IMUs for attack detection. UAVs typically have tight power and weight constraints, and hence the countermeasures should fit within these bounds. Thus, solutions that require minimal modifications to the existing infrastructure are essential.

Cryptographic solutions include techniques that introduce message encryption and authentication [146, 240, 153, 65, 92]. The underlying cryptographic primitives make it difficult for an attacker to synthesize GPS signals for arbitrary locations. However, these solutions are vulnerable to signal replay attacks. Furthermore, these countermeasures require a complete overhaul of the GPS ecosystem and are currently unavailable for evaluation. Additionally, they require high processing power, making it less practical to deploy on UAVs.

Physical and application layer countermeasures detect anomalies in RF properties like signal strength [38], auxiliary peaks [195], angle/direction of arrival [162, 165], and validation of navigation messages like satellite ephemeris and timing information [195, 73]. Some countermeasures also leverage a multi-antenna and multi-receiver setup for attack detection [228]. An attacker can obfuscate physical properties by carefully generating signals such that the signal's physical properties are within set thresholds. An attacker can also completely overshadow adversarial signals, thus burying the legitimate signal under the noise, thereby removing any auxiliary peaks. In addition, an adversary can use multiple antennas to mimic the angle and direction of legitimate signals.

There are proposals to use inertial sensor measurements to detect attacks on GPS through a comparison of independent inertial measurements and obtained GPS measurements [152] [225, 144]. Such solutions often use Kalman-filter-based sensor fusion algorithms already implemented in modern UAVs for state estimation. These techniques are effective

against the proposed UAV takeover attack and can lead to attack detection. However, as shown in [211, 168, 249], an attacker can take over and defeat multi-sensor-fusion algorithms and inertial sensor solutions. Such a detection scheme can be used along with terminal fail-safes [42], which force the UAV to abort any ongoing mission and either hold and hover at the current position or land. However, these techniques are limited to attack detection and do little in terms of attack mitigation and recovery. The receiver will also need to identify and attenuate adversarial signals in order to mitigate and recover from the attacks and ensure uninterrupted operations.

Several countermeasures use successive interference cancellation (SIC) technique [58, 82, 203] and antenna array processing techniques [145, 158, 70] to mitigate GPS spoofing attacks. Most of these mitigation techniques require high processing power or peripheral devices that make it impractical for implementation on UAVs. In [203], the authors have designed a solution specifically for UAVs. Such a countermeasure has the potential to recover from the GPS takeover attack proposed in this work. However, this solution is limited up to 15 dB of a spoofing signal's power advantage over legitimate signals beyond which the UAV cannot recover.

4.8 Related work

Several strategies have been proposed that demonstrate the ability of an attacker to assume control of a UAV via GPS spoofing. These works primarily focus on altering the motion of a UAV by transmitting fake GPS signals. In [176] the authors provide a taxonomy of hijacking consumer drones. However, the anti-drone hijacking strategies they propose are only capable of limited control over the target drone. Through a hard-spoofing attack, their strategies could divert a drone in a specific direction. Beyond that, the strategies proposed in this work do not facilitate the complete takeover of the target UAV as they lack post-takeover direction and velocity control ability. In [141, 107], through simulations, the authors demonstrated the effect of GPS spoofing on a cyber-physical system such as a UAV. Their approach forces the drone to accelerate in a particular direction by manipulating the GPS velocity in the opposite direction. In this approach, the direction of the motion and the UAV's acceleration was uncontrollable and unpredictable. As a result, such an approach cannot precisely control and maneuver the drone through GPS spoofing. In [155], the author demonstrated an attack that targets the follow-me feature specific to a DJI Phantom 3A. In this attack, the controller's mobile phone is targeted rather than the onboard GPS receiver. Since the attack only targets the "follow-me" flight mode, such an attack will not work against other completely autonomous flight modes. Similar works [43, 96, 208] demonstrate identical strategies that are restricted to a specific type of drone or a simulator environment,

or that provide minimal true control over the target UAV.

There has been significant research on developing countermeasures to safeguard GPS receivers against spoofing attacks. Cryptographic solutions [146, 240, 153, 65] prevent attackers from generating counterfeit signals. However, they are still vulnerable to signal replay attacks and are not practical for deploying on a UAV because of additional processing power and key management requirements. The recently launched Galileo's Open Service Network Authentication [92] service that uses TESLA protocol for broadcast authentication is vulnerable to signal replay attacks. Other countermeasures like [238, 48, 162, 228, 166, 165] rely on peripheral hardware devices like multiple receivers and directional antennas, thus making them infeasible for integrating with UAVs. Another line of works [248, 144, 225] demonstrates the application of GPS/IMU sensor fusion to detect GPS spoofing attacks. However, as shown in [249, 168], it is possible to evade detection and defeat such multi-sensor fusion (MSF) algorithms. Shen et al. [211] analyzed the security guarantees of MSF algorithms implemented in terrestrial autonomous vehicles with the specific goal of forcing lane changes. Finally, [195] provides a technique that is based on signal processing and does not require additional hardware or cryptographic measures, but it does not protect against fine-grained spoofing of proximate locations as we have done in this work.

Several works have used non-GPS techniques for a hostile takeover of UAVs. These include the use of lasers to activate obstacle detection and avoidance systems [250], attacking the data-link between the radio-controller and the UAV [198] and [188] where the authors showcase the vulnerabilities present in a popular UAV platform from Parrot [12] as a result of a poorly configured wireless network.

4.9 Conclusion

In this work, we experimentally enumerated and validated various challenges in asserting complete control of a UAV through a GPS spoofing attack. We formulated requirements that constitute a complete takeover. To this extent, we designed, demonstrated, and evaluated strategies that enabled us to control the UAV's speed and direction in real time through well-timed GPS velocity manipulations that resulted in stable, predictable, and controlled flight. To facilitate real-time control, we designed and developed a real-time GPS signal generator capable of on-the-fly trajectory manipulation. We also designed a Human-in-the-Loop GPS spoofing system that can manually control a UAV's motion. Further, we discussed the possibility of incorporating an automatic self-learning feedback mechanism.

In conclusion, we show that even though the GPS receivers of COTS UAVs remain vulnerable to spoofing attacks, the combination of sensors incorporated in UAVs makes it extremely challenging for the attacker to translate a GPS spoofing attack into complete

control over the UAV. We show in this work that — against conventional wisdom — only with a thorough study of a UAV's systemic behavior under GPS spoofing attacks and careful manipulation of the spoofing signals would it be possible to commandeer a UAV through GPS spoofing alone.

Chapter 5

SemperFi: Anti-spoofing GPS Receiver for UAVs

5.1 Introduction

Today, Global Positioning System (GPS) is critical to various safety and security-critical applications. GPS is so ubiquitous that it plays an enabling role in 14 out of 16 industries classified as critical infrastructure [17] 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 [40]. Several researchers have shown that it is possible to modify the course of ships [28], unmanned aerial vehicles [212] and self-driving cars [184] by simply spoofing GPS signals. There is also an increase in GPS spoofing incidents [60] reported from around the world. For example, there are reports of thousands of ships in Shanghai falling victim to GPS spoofing [105]. There are also reports [60] 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 [81], 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 [146, 240, 153, 65] prevent attackers from generating arbitrary false GPS signals. The recently launched Galileo's *Open Service Navigation Message Authentication* [92] 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 [238], noise levels, direction or angle of arrival [165], and other data that are readily available as receiver observables on many COTS GPS receivers. Some countermeasures [195] exploit the difficulty of completely canceling out legitimate GPS signals to detect stealthy, seamless takeover attackers. A few countermeasures propose the use of additional sensors [135] and receivers [228, 166] to detect spoofing attacks. Most 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 [228] of the victim's GPS location without any signal disruption, despite having redundant fail-safe sensors [168]. 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 comprises 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. Specifically, we make the following contributions.

- We design a spoofing mitigation technique leveraging inertial sensors and Extended Kalman Filter (EKF) algorithm that is commonly part of the majority of UAV's built-in GPS fail-safe mechanisms [159] and integrate it with SemperFi's adversarial peak identifier module. In combination with SemperFi's legitimate signal retriever, we show that SemperFi can detect majority of GPS spoofing attacks present in literature and autonomously recover its true location.

- We introduce an active spoofing verification component that forces the UAV to execute a maneuver in the scenario of stealthy adversaries capable of gradually introducing location offsets [249] without triggering the built-in GPS fail-safe mechanisms. We rely on the auxiliary peak tracking technique [195] 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 with varying acceleration with

the goal of minimizing the time-to-trigger 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 a seamless takeover attack. Prior spoofing detection techniques based on inertial sensors are vulnerable to adversaries deviating the UAV's path at a rate that is well within the EKF estimation bounds, and 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, in addition to the data contained within the navigation messages, it is essential to preserve the ToA of the satellite signal itself. 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 [93] and evaluate its performance against both synthetically generated as well as real-world GPS signals using consumer drones like DJI Flamewheel F450 [7] and Holybro S500 [14]. 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 [112], a public dataset of GPS spoofing traces. Our evaluation shows that in the majority of attack scenarios, SemperFi can recover with an accuracy of less than 20 m and within 0.54 (in the 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 disruption.

- Finally, we open source[1] our design, implementation, and evaluation datasets to the community for further research and development.

The rest of this chapter is organized as follows. First, we present an overview of state-of-the-art GPS countermeasures in Section 5.2, followed by Section 5.3 where we describe the attacker's model, challenges associated with GPS spoofing mitigation and present our design and a high-level overview of SemperFi. Next, in Section 5.4, we describe how we implemented adversarial peak identifier, legitimate signal retriever, and pseudorange rectifier,

[1] `https://github.com/harshadms/semperfi/`

the building blocks of SemperFi. In Section 5.5, we present the results of the security and performance evaluation of SemperFi. Next, we go over the flexible-design use case of SemperFi, the limitations of SemperFi, and insights for future work in Section 5.6. Finally, we conclude the chapter with final remarks in Section 5.7

5.2 State-of-the-Art GPS Countermeasures

Several GPS spoofing countermeasures have been proposed in the past. Most of these works focus on building spoofing detection techniques and do not address the challenge of neutralizing the attacker's spoofing signals. In this chapter, we present SemperFi, a single antenna GPS receiver designed explicitly for UAVs 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 [82] and the in-line GPS spoofing mitigation technique [148]. In [82], the receiver uses maximum likelihood estimates after dampening the attacker signal to estimate the correct location. The in-line GPS spoofing mitigation technique [148] implements an extended RAIM method to filter outliers and correlation peak distortion techniques to detect spoofing signals. Both these works cannot distinguish adversarial peaks and fail against strong adversaries such as a seamless takeover attacker. Signal cancellation has been explored to attack GPS receivers in [167] by attenuating a specific satellite. Furthermore, successive interference cancellation has been used to eliminate the near-far problem associated with pseudolites [156]. The authors treat overpowering pseudolites as interference because, despite being a legitimate signal, 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 [163, 162] 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 [228]. Borio *et al.* [52] provide an interference cancellation technique for recovering from GPS jammers. This work statistically models GPS jamming signals, which aids in jamming signal removal.

Several cryptographic solutions have been proposed for securing navigation messages. In [146, 65], the authors propose an asymmetric and hidden marker approach for securing civilian GPS signals from signal-synthesis attacks. In [240], the authors propose an authentication scheme by incorporating digital signatures. Although these cryptographic solutions prevent a signal spoofing attack, they require 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 countermeasures do not prevent record

and replay attack [180].

Several spoofing detection schemes require extra peripherals like multiple antennas [166, 48, 165], which detect discrepancies in the angle of arrival of GPS signals. GPS signals and location estimates are correlated with data from extra IMU sensors [135, 239, 229, 89] for detecting GPS spoofing attacks using vector-based tracking. Extensive work focuses on using EKF to aid in recovering from GPS glitches [225, 103]. ArduPilot has one such implementation. Eichelberger2020spoofproof Our experiments found that a spoofer can avoid detection by controlling the introduced error in the positions. In [249, 170], 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 [168] has been recently shown to be vulnerable to attacks against on-road navigation systems. Several works [228, 137] propose using multiple receivers to detect spoofing signals by comparing the 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 [192], cross-validation of PVT solutions across multiple navigation systems [175] e.g., GPS, GLONASS, Galileo, etc. Just like military signals, researchers have developed spoofing detection techniques that use opportunistic IRIDIUM signals [177] In [136], the authors leverage a crowdsourced network to detect GPS spoofing attacks. In [51], 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 [95] to detect GPS spoofing attacks by identifying legitimate satellites. Works like SPREE [195] and vestigial signal detection [241] provide 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 most GPS spoofing attacks.

5.3 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 first present the attacker's goals and assumptions followed by challenges associated with threat mitigation and location recovery. Finally, we present our design of SemperFi that is capable of autonomous recovery and uninterrupted operations.

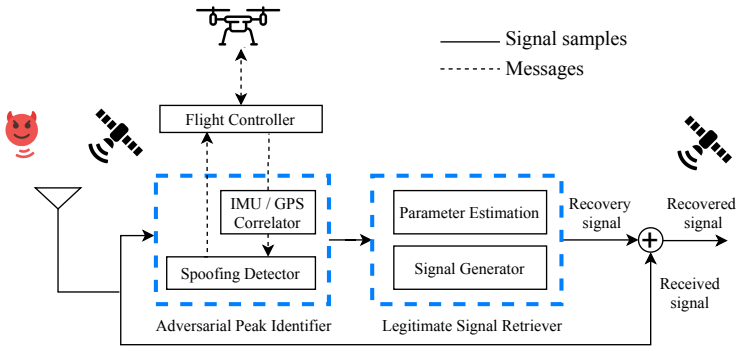


Figure 5.1: High-level overview depicting essential components of SemperFi.

5.3.1 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 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; 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.

This work focuses 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 [228] and drone hijacking strategies proposed in [176]. We assume the following about the attacker. The attacker can be equipped with an omnidirectional or a directional antenna and is allowed to spoof any number of satellites. Our threat model includes attackers with little know-how and sophisticated seamless-takeover attackers [228]. Attackers with little know-how typically use GPS signal generators (both hardware [8] or software [81]) 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 delay or modifying navigation message contents; thereby affecting the pseudorange calculations. The requirements to execute such a seamless takeover attack have been explored in [228]. 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 an attacker's success are explained further in Section 5.5 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.

5.3.2 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 a spoofing signal and a 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 our knowledge, no receiver design in prior work addresses the above three challenges so far. 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 of received sample losses. Continuous tracking of the satellite signals is necessary to estimate code and carrier phase delays directly affecting the PVT estimation. Finally, in the case of a spoofing attack that injects a fake dynamic motion pattern (e.g., diverting the course of a ship or forcing 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 must be modified or extended to handle this attack.

5.3.3 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 5.1.

API relies on a widely adopted extended Kalman filter-based sensor fusion algorithm for UAVs and the spoofing detection methodology based on prior work [195] that demonstrated

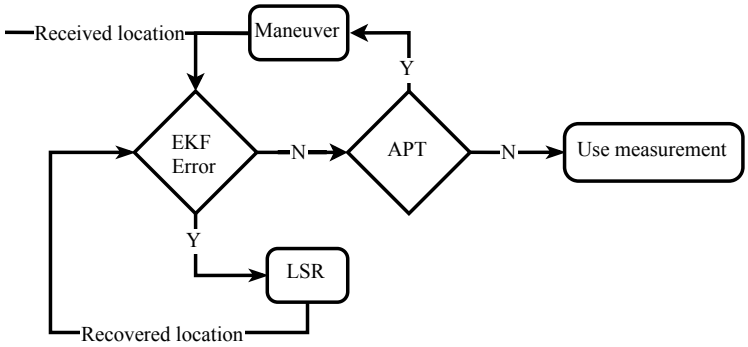


Figure 5.2: Flowchart depicting the operations of SemperFi and crucial decisions it makes to execute the recovery process.

the ability to detect even a strong, seamless takeover attack [228] 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 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 5.5). However, it is shown in prior works [249, 141] 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 [195]. 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 5.2 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. I.e., if LSR receives an "adversarial" message, it 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. Below, we give a brief overview of the signal recovery process.
SIC [182] is a well-known technique used for canceling out interference caused by stronger signals in a CDMA system. The GPS signal from a single satellite can be modeled as

$$S_R = a[k]\tilde{s}_T[k - \tau(k)]e^{j2\pi f_D[k]T_s k + \phi[k]} \tag{5.1}$$

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

$$S_R = S_L + S_{AT} \tag{5.2}$$

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:

$$S_L = S_R - S'_{AT} \tag{5.3}$$

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 the 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.

5.3.4 Adversarial Peak Identifier (API)

The API leverages the measurements obtained from the extended Kalman filter implemented in the UAV. Precise estimation of a vehicle's time-varying position is required for 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 the vehicle controller requires for accurate navigation. The UAV observes its own position and velocity using onboard 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, inaccurate GPS measurements can only trigger this failsafe.

The presence of a valid satellite signal is determined by a peak that forms due to 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) identifies such malicious peaks. Similar to any wireless receiver, the GPS receiver also experiences capture effect [55] and, by default, locks on to the strongest signal and tracks it. Thus, even when 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 accounts for the above-described scenarios. As described in Section 5.3.3, the sensor fusion algorithm will raise an error and identify the signal to eliminate under most spoofing attacks, as any discrepancies between inertial measurements and GPS measurements will trigger the EKF errors. However, it is possible for an attacker to spoof GPS signals without raising this error. In such a scenario, multiple peaks indicate 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 cannot identify the adversarial signal correctly. SemperFi sends a maneuver to the flight controller. This ma-

neuver is represented as a series of velocity vectors with varying acceleration. These velocity vectors are represented in reference to local north east down (NED) frame, and the flight controller can execute the maneuver 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.

5.3.5 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

$$x_{IN}[k] = a[t]\tilde{s}_T[t - \tau(t)]e^{j2\pi f_D[t]T_s t + \phi[t]} \tag{5.4}$$

where $a[t]$ is the signal amplitude, $\tilde{s}_T[t]$ is a filtered and sampled version of the complex baseband GNSS signal. Computation of the correlations which comprise the sampled CAF, in the acquisition block is typically done in the Fourier domain after carrier wipe-off.

$$x[k] = x_{IN}[k] \cdot e^{-j2\pi \check{f}_D t T_s} \tag{5.5}$$

At the peak of the CAF, the parameters $\check{f}_D[t], \check{\tau}[t], \check{\phi}[t]$ correspond to the maximum likelihood estimate of the "true" parameter values, and the discrete Fourier domain representa-

tion of the signal after wipe-off simplifies to

$$X[k] = \mathrm{FFT}_K\{x[k]\} = a[t] * S[t]W_K^{\tau} \tag{5.6}$$

Applying the FFT of the local code replica $D[k]$ is performed by multiplication in the Fourier domain

$$Y[k] = X[k] \cdot D[k] = a[t] * S[t]D[k]W_K^{\tau} \tag{5.7}$$

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

$$R_{xd}(f_D, \tau) = \mathrm{IFFT}_K\{Y[k]\} = a[k] \sum_{n=0}^{K-1} s[n]d[k-n] \tag{5.8}$$

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

$$S_{\mathrm{MAX}} = |R_{xd}(f_D, \tau)|^2\big|_{f_D \approx \hat{f}_D, \tau \approx \hat{\tau}} = |a|^2|K|^2 \tag{5.9}$$

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

$$|a| = \frac{\sqrt{S_{\mathrm{MAX}}}}{K} \tag{5.10}$$

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 to calculate pseudorange to the satellite, an essential component in PVT calculation [196]. In this technique,

a common reception time, which is usually 65-85 ms, is set across all the channels as the propagation time of the closest satellite's signal [196]. 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 this technique, pseudorange is calculated as follows:

$$P^i = c(t_{ref} + t_{rx} + \tau^i) \tag{5.11}$$

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 [196]), 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.

$$P^i_l = c(t_{ref} + t_{rx} + \tau^i_{at} - \Delta\tau^i) \tag{5.12}$$

$$\Delta\tau^i = \tau^i_{at} - \tau^i_l \tag{5.13}$$

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.

5.4 Implementation

The two sub-systems which make up SemperFi are implemented independently 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

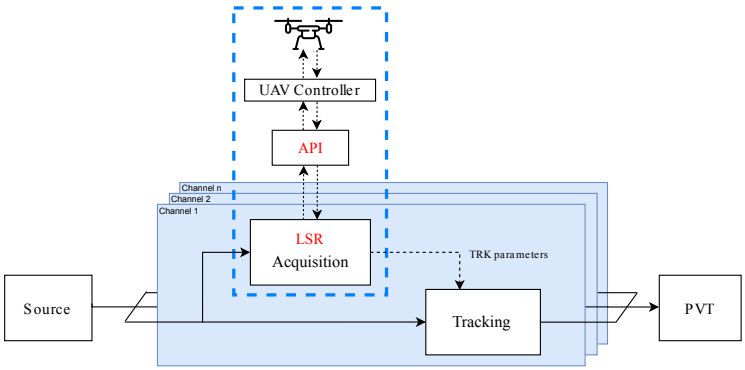


Figure 5.3: LSR is implemented as part of GNSS-SDR and API is implemented as part of the UAV flight controller.

of SemperFi in GNSS-SDR [93] an open-source software-defined GNSS receiver written in C++. We implemented the API module using consumer drones. Refer to Figure 5.3 for a schematic of the implementation. GNSS-SDR follows GNURadio architecture and supports the processing of pre-recorded signals from a file source and software-defined RF-frontends like a USRP [6]. GNSS-SDR follows a hardware receiver's design as described in Section 4.3, 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 [6], specifically, USRP B210 and N210 with SBX-40 daughterboard, for recording and providing raw data.

5.4.1 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 [13], an advanced autopilot system and ArduCopter [41] firmware. Refer to Figure 5.4 for the hardware setup.

When the position and velocity variance crosses a set threshold, the flight controller activates an *EKF Variance* error along with 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

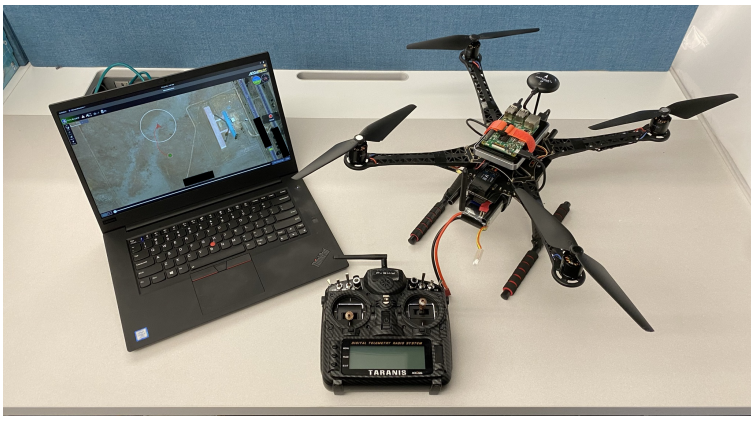


Figure 5.4: Hardware setup showcasing the Holybro S500 drone, the radio controller, and the ground control station.

identification maneuver, the UAV undergoes rapid acceleration/deceleration in an unpredictable direction defined by the NED velocity vector relative to its own body frame. 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* [5] installed on a Raspberry Pi 3B+ 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 5.3.4 and notifies LSR over a TCP socket.

5.4.2 Legitimate Signal Retriever (LSR)

In SemperFi, we implement LSR as a part of GNSS-SDR's acquisition module. As mentioned earlier, we use an 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 [195]. We modified the acquisition block such that the 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, including peak identification, recovery signal generation, and signal recovery. GNSS-SDR allows external communications using TCP sockets as outlined here [94]. 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 5.3.5. We use the Vector-Optimized Library of Kernels [242] 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 scenarios, 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.

5.4.3 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.

5.4.4 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 [199]. 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.

5.5 Security and Performance Evaluation

5.5.1 Theoretical Security Evaluation of Identification Maneuver

In this section, we conduct a theoretical evaluation of peak identification from an attacker's perspective. We establish the fundamental property that ensures the 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

$$\mathbf{x}_{n+1} = \mathbf{F}_n \mathbf{x}_n + \mathbf{G}_n \mathbf{u}_n + \omega_n \tag{5.14}$$

The drone tracks the time-evolution of own state, which in its most basic form is comprised of three-dimensional position, velocity and acceleration $\mathbf{x}_n = [x_n, \dot{x_n}, \ddot{x_n}, x_n, \dot{x_n}, \ddot{x_n}, z_n, \dot{z_n}, \ddot{z_n}]^\top$. 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

$$\mathbf{y}_n = \mathbf{H}_n \mathbf{x}_n + \nu_n \tag{5.15}$$

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

$$\hat{\mathbf{x}}_{n+1} = \hat{\mathbf{F}}_n \hat{\mathbf{x}}_n + \hat{\omega}_n \tag{5.16}$$

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 $\hat{\mathbf{y}}_n$ is comprised of only the three-dimensional position $\hat{\mathbf{y}}_n = [x_n, y_n, z_n]^\top$.

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 a 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 information pertaining to the controller: if the target induces an input to the system model by way of 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 the target can detect. 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 [202]

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \mathcal{N}\left(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t-1}, \mathbf{\Sigma}_{t|t-1}\right) \tag{5.17}$$

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \mathcal{N}\left(\mathbf{x}_t; \hat{\mathbf{x}}_{t|t}, \mathbf{\Sigma}_{t|t}\right) \tag{5.18}$$

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 $\hat{\mathbf{x}}_{t-1|t-1}$ and $\mathbf{\Sigma}_{t-1|t-1}$ through the process model given in (5.16).

$$\hat{\mathbf{x}}_{t|t-1} = \int \hat{\mathbf{f}}(\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \tag{5.19}$$

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

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 $\hat{\mathbf{x}}_{t|t}$ and $\mathbf{\Sigma}_{t|t}$ will increasingly diverge 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 (5.14) according to the Kalman innovation equation

$$\tilde{y} = \mathbf{z}_k - \hat{x}_{k|k-1} \tag{5.20}$$

Since the observations $\mathbf{z}_k$ do not come from the true distribution, but are instead the product of spoofing, the resulting innovation $\tilde{y}$ 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.

$$A = \{x \in \mathbb{R} | -\mathbf{j} < x < \mathbf{j}, |x_n - x_{n-1}| = \mathbf{R}\} \tag{5.21}$$

where **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 given as $P(A)^{\mathbf{n}}$ where **n** is the total number of steps in the maneuver. In this *guessing-game*, the attacker's success depends on the **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[2] 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, the results of which are summarized in Figure 5.5. An attacker can stay undetected longer when it is attacking a UAV with a low-resolution sensor and is incapable of rapid motion. In case of a lower **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 a 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[3] [10] with a 16- bit ADC and acceleration range of $\pm$2g has a resolution of 0.000598 m/s^2. Figure 5.6 shows the mean time to trigger for a UAV that uses this sensor.

5.5.2 Experimental Evaluation of Identification Maneuver

In this section, we analyze the security and performance of identification maneuvers by evaluating them in a simulated environment using Gazebo [197] as well as in a practical real-world setting using real drones. We used ArduCopter [41] for both cases. Figure 5.7 offers a schematic view of our experimental setup and the implementation of the spoofing scenario simulator. All the values presented in this analysis are specific to the UAV that we

[2]Update rate of a typical UAV flight controller

[3]This is the same IMU sensor onboard the UAVs that we used in our evaluation.

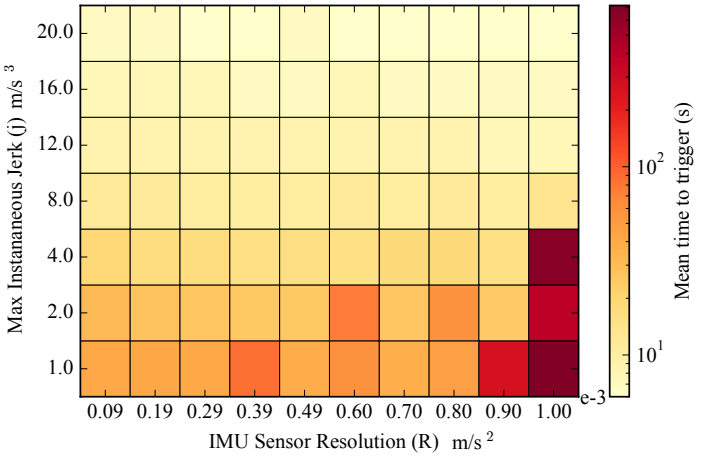


Figure 5.5: This plot shows the effect of sensor resolution and maximum possible instantaneous jerk on time to trigger

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 [195]. 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 three scenarios:

1. Static non-adaptive signal spoofing attack: We assume the UAV is hovering and an attacker starts spoofing a static location. This will force the drone to start drifting because of IMU error accumulation slowly. 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 5.5. 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 attacker's goal 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

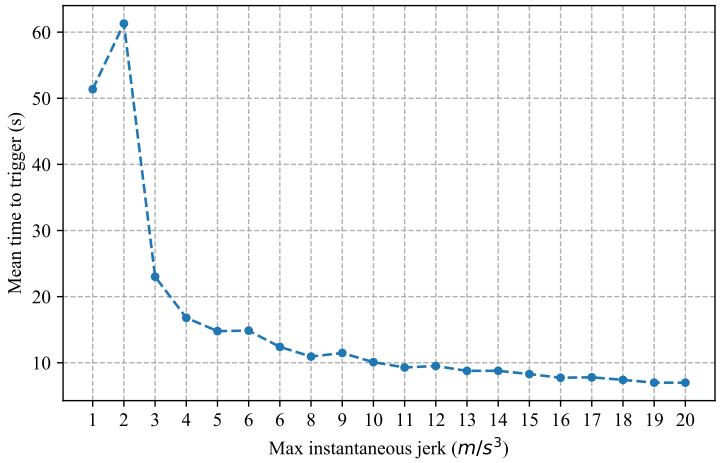


Figure 5.6: A plot showing the mean time to trigger for various **j** values for a UAV that uses an IM-20689 inertial sensor.

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 in scenario 2, the attacker deviates from 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 the auxiliary peak is detected, the UAV performs the identification maneuver and is able to trigger the error in 18.001 s. Refer to Figure 5.8 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

$$t = \frac{\Delta V_{min}}{a_{max}} \tag{5.22}$$

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.

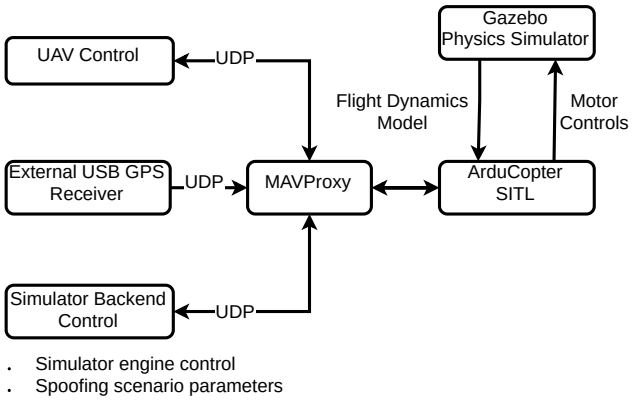


Figure 5.7: Our experimental setup consists of an ArduCopter software-in-the-loop simulator that uses Gazebo as a physics simulator. UAV Control, GPS receiver and Simulator Backend Control are implemented using *dronekit* and interact with the simulator through MAVProxy.

Refer to Figure 5.9 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 5.22, 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.

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 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 specifically, 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.

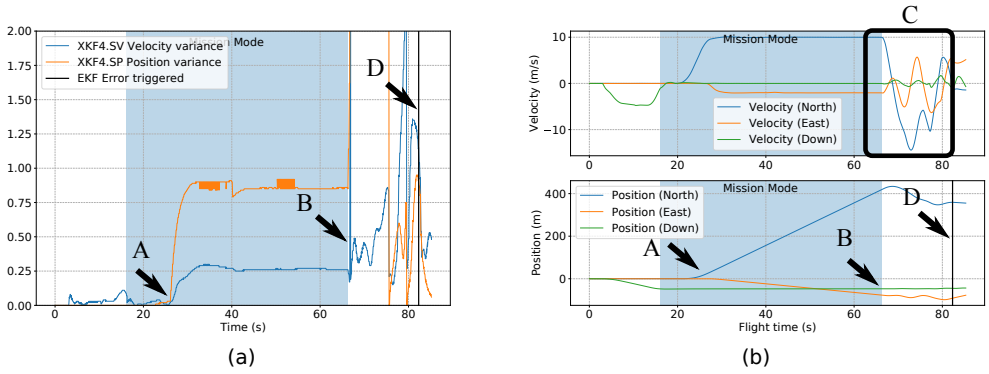


Figure 5.8: A plot showing (a) EKF innovations variance and (b) position data for the moving target adaptive GPS takeover scenario. The spoofer starts introducing offsets at marker A. This is evident from the rise in position variance. At marker B, SemperFi kicks in, pauses the mission and instructs 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

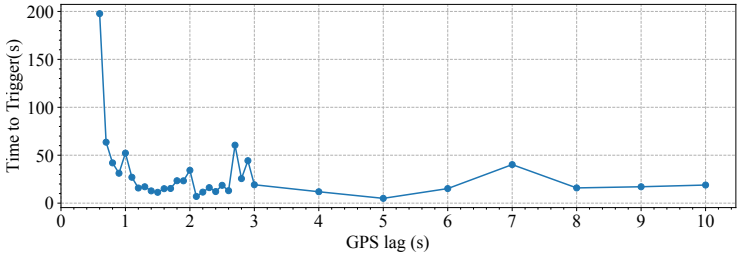


Figure 5.9: The effect of GPS lag in terms of time to trigger the EKF variance

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 ($5m/s^2$) and jerk $20m/s^3$ 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.

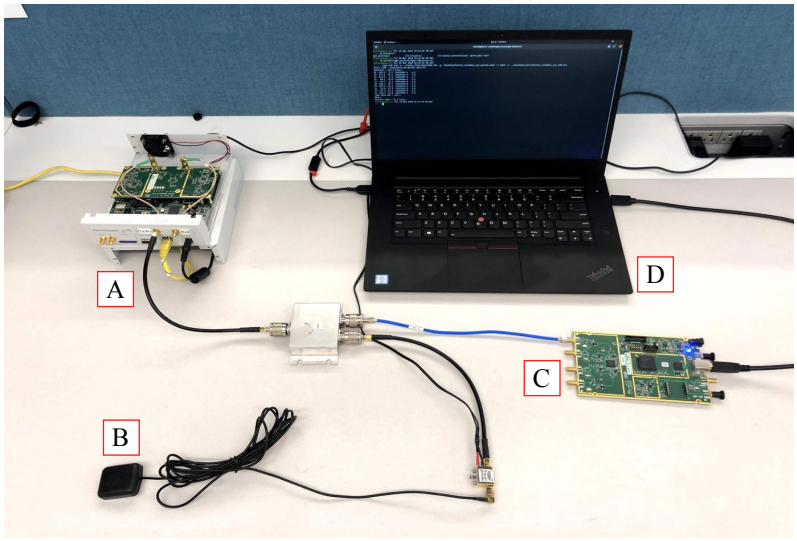


Figure 5.10: Signal recording setup A) GPS signal RX (USRP N210), B) ANT-555 active GPS antenna with a 5V bias-tee, C) GPS signal TX and D) GPS simulator control unit.

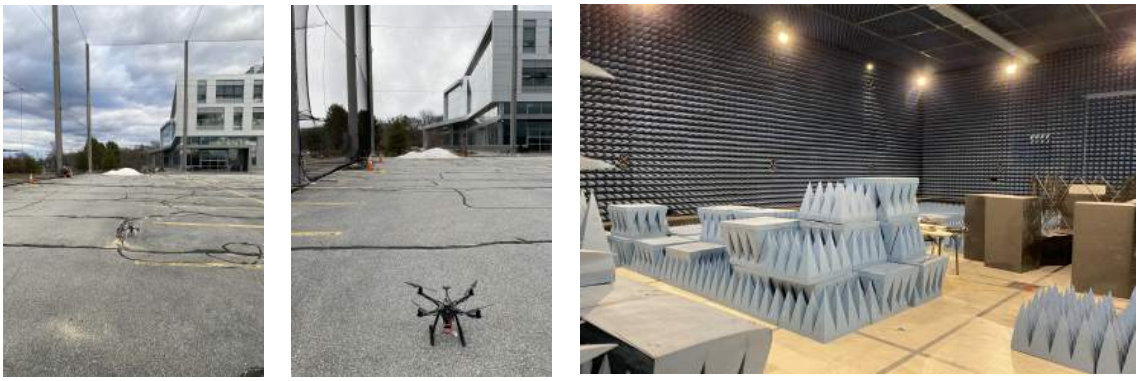
5.5.3 Experimental Performance Evaluation

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 drones.

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) [112], and iii) recorded real-world GPS signals.

GPS Simulator We performed most of our evaluation on synthetic signal traces generated locally using GPS-SDR-SIM [81], 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



(a) (b) (c)

Figure 5.11: The drone testing and evaluation setup. (a) and (b) show the drones that we used (DJI Flamewheel F450 and Holybro S500) in the outdoor UAS testing facility. (c) shows the indoor anechoic chamber used for GPS spoofing and jamming experiments.

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.

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.

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 our locality[4]. We recorded the legitimate GPS signals using the setup shown in Figure 5.10. 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

[4]location anonymized.

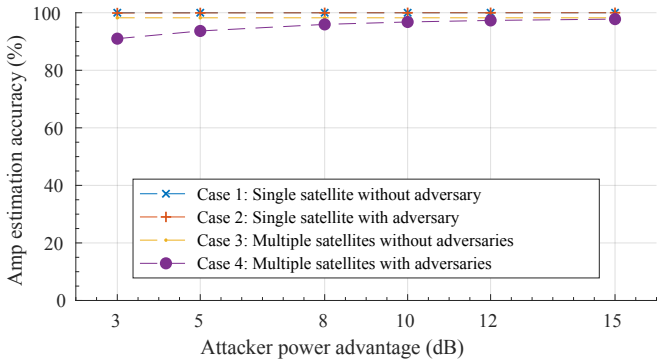


Figure 5.12: Amplitude estimation accuracy in various scenarios. Attacker power advantage does not apply to cases 1 and 3. In case 4, each satellite is spoofed. Power advantage refers to the advantage that the attacker's signal has over the legitimate signal.

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.

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 5.12 for results.

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 (UTM) [18] 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 5.5.3.

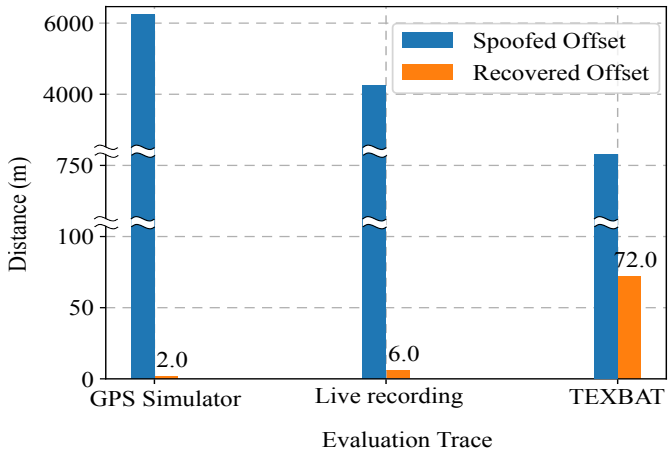


Figure 5.13: The spoofed offset and the recovered offset for three scenarios.

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 5.13 shows the recovery operation results on static scenarios across all three datasets. GPS simulator traces where the spoofed offset is 6.2 km with a 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 5.13 shows varying recovered offset as a result of attacker signals' synchronization with the legitimate signals. More details are present in Section 5.5.3

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 5.14 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

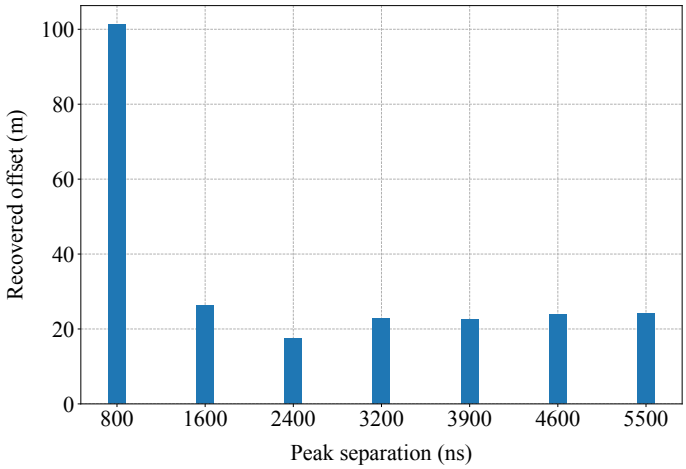


Figure 5.14: The effect of peak separation on the accuracy of the recovered location. The closer the peaks, the harder it gets to accurately track them. The power advantage is set to 3 dB. 3 dB is strong enough to take over the receiver and yet not strong enough to bury the signals under noise. It also allows us to evaluate the effects of signal synchronization on signal recovery.

loop starts tracking the wrong signal. This is evident from the higher recovered location offset for the scenario with peak separation of 800 ns.

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 [228, 112]. 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 plot. Figure 5.17 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 5.16(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 5.15 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° [4] which means, an attacker who is on ground is already has a disadvantage of 15 dB. Moreover, an attacker trying to compensate for this

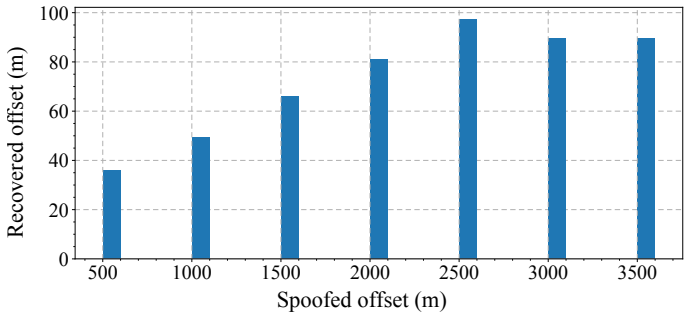


Figure 5.15: Spoofed offset vs offset in recovered location for an attacker with 15 dB power advantage. SemperFi uses Pseudorange Rectifier for recovery. For locations refer to Section 5.5.3.

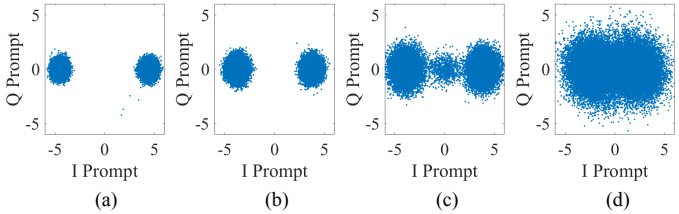


Figure 5.16: Discrete time scatter plot of recovered nav message where the attacker has (a) 3 dB, (b) 5 dB, (c) 10 dB, and (d) 15 dB power advantage. A powerful attacker adds noise and hence distorts legitimate nav messages.

power disadvantage can be easily localized thus making it easier for detection.

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 5.5.3 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 5.1 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

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

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

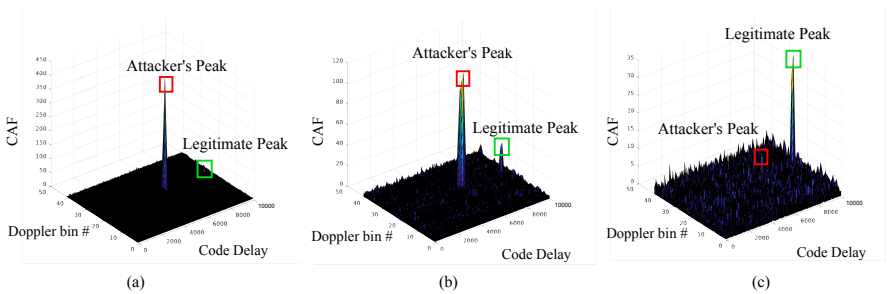


Figure 5.17: Two-step signal attenuation of a strong adversarial signal. (a) shows the original acquisition plot, (b) shows the acquisition plot where the legitimate peak is slightly visible and (c) the final acquisition plot with a fully suppressed adversarial peak.

Table 5.1: 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

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.

5.6 Discussion

5.6.1 Flexible design

SemperFi is designed to be flexible and versatile. In addition to integrating SemperFi into the acquisition module as shown in Section 5.4, we can use SemperFi as a pluggable 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 capabilities can also be extended to 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.

5.6.2 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 [102, 68, 87, 63, 59, 213]. 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 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 [168], 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, specifically high-velocity crosswinds. However, the algorithm can be modified to work with crosswinds by determining the force and velocity of wind as proposed in [172] or by equipping the drone with solid-state anemometers.

5.7 Conclusion

In this chapter, we presented SemperFi, a single-antenna spoofer signal-eliminating GPS receiver capable of providing uninterrupted legitimate locations even in the presence of a strong adversary. We designed and 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 able to identify adversarial peaks by executing flight patterns less than 100 m long and recovering the true location in under 0.54 s for most scenarios, as an identification maneuver is not required for all scenarios. Finally, we release the implementation of our receiver design to the community for usage and further research.

Chapter 6

Conclusion

6.1 Summary

We organized this thesis document into several chapters; each focused on evaluating the security of critical components of automation systems in modern aerial vehicles. In Chapter 2, we described our security analysis of the widely used landing system, ILS. Specifically, we identified the fundamental issues in analog systems and demonstrated wireless attacks to manipulate ILS signals. Next, in Chapter 3 presented our evaluation of the security implications of spoofing and jamming aviation datalink applications, such as ACARS, CPDLC, and ADS-C. In Chapter 4, we demonstrated the possibility of exerting post-takeover control over UAVs via GPS spoofing by forcing the UAV to execute patterns such as 180° turns. Through this experimental analysis, we determined the feasibility and requirements for precise control of UAVs via GPS spoofing. Building on the insights gained through the security evaluation of these critical systems, in Chapter 5, we designed and implemented SemperFi, a single-antenna GPS receiver for UAVs capable of detecting and mitigating several GPS spoofing attacks. Our work emphasizes the importance of interconnected system security and the need to test integrated attack scenarios simultaneously targeting several critical systems and provide a starting point for further research in developing secure autonomous technologies that can safeguard the safety and integrity of modern aerial vehicles.

6.2 Future Direction

It is evident from this thesis that automated systems and wireless technologies, the backbone of the modern transportation ecosystem, are susceptible to wireless interference and that leveraging multiple co-working sensors could be a vital factor in developing robust countermeasures that safeguard these safety-and security-critical systems.

Automated feedback mechanism for UAV takeover: Our experimental evaluation of UAV takeover with GPS spoofing shows the feasibility of such an attack. However, we conclude that spoofing is not trivial, and various factors can interfere with the presented strategies. This limits the success of the proposed techniques. Our trials involving the Human-in-the-loop (HiTL) GPS spoofer showed that an attacker could have greater control over the hijacked UAV if the attacker employs a feedback mechanism that adjusts the spoofed location based on the target UAV's response to the GPS spoofing. One such proposed system will leverage a machine learning model, artificial intelligence, and a UAV tracking system to monitor the UAV's movements and learn from its reactions. Such a closed-loop feedback system will effectively maintain precise control over the UAV. Additionally, there is potential for research into more advanced feedback mechanisms for UAV takeover that are less susceptible to interference and can more effectively maintain control over the target UAV.

Investigating integrated synchronous attacks on the aviation ecosystem: In this thesis, we present a comprehensive security evaluation of two critical aviation systems, the instrument landing system (ILS) and the aircraft communication, addressing, and reporting system (ACARS). Our investigation thoroughly analyzes the redundancies inherent in modern aircraft avionics and how these redundancies can both prevent and mitigate the risks posed by individual systems. Our research findings show that these redundancies increase the complexity of attacks on avionics, as an attacker must simultaneously attack multiple systems so that each system corroborates the attacker-introduced values. For instance, an attacker must also spoof the global navigation satellite system (GNSS) to launch a successful ILS spoofing attack, as flight crews can cross-check their GNSS-derived location with the ILS instruments. Modern urban air mobility programs require the cooperation and collective intelligence of multiple unmanned aerial vehicles (UAVs). Swarm operations present new attack surfaces that require further examination, particularly in light of the recent introduction of ACAS-x. This system extends collision avoidance capabilities to autonomous aerial vehicles and represents a significant step towards the co-existence of these vehicles. Physical layer distance measurement and ranging are essential to such technologies and require special attention. These systems are designed to make a practical attack extremely complex. Hence, it is necessary to determine the feasibility and practicality of attacks through near-to-real-life experiments and evaluations. This work highlights the importance of continued work on the resilience of existing components in aviation security and opens-up new research paths in aviation security.

Modern transportation infrastructure security: Autonomous vehicles, like self-driving cars and aerial delivery drones, are becoming essential components of urban infrastructure.

Terrestrial vehicles majorly rely on CANBus for their in-vehicle network. An attacker can compromise one of the components already connected to the bus or physically insert a rogue device that issues malicious commands. CANBus attacks can be prevented through hardware and software mechanisms that ensure attack detection and mitigation. Terrestrial cellular networks enable vehicles to stay connected with their surroundings by leveraging specific V2X protocols, 5G especially has a variety of location and vehicle-related features. These protocols enable ranging and localization which is of paramount importance. It is necessary to identify the resilience of such networks to a combination of physical layer distance manipulation attacks and logical layer attacks. For example, attackers can leverage specific unprotected MAC layer messages to launch desynchronization attacks that result in denial of service.

6.3 Final Remarks

In this dissertation, we have analyzed the security guarantees of unauthenticated legacy wireless aviation systems, explicitly emphasizing the escalation of threats from localized single-component attacks to the entire vehicle system. Our findings reveal that careful manipulation of RF signals can severely threaten both crewed and autonomous aviation vehicles, leading to potentially catastrophic outcomes. As it stands today, redundant systems, early detection, and human intervention safeguards play a crucial role in recovering from attacks. However, with the increasing reliance on wireless communication and navigation systems and the advent of crewless autonomous vehicles, the effectiveness and validity of human intervention safeguards are significantly limited.

Advancements in SDR technology and the development of sophisticated synchronized multi-attacker models have magnified the risk. Therefore to minimize the risk, it is essential to understand the threats and feasibility of attacks to develop robust countermeasures and to raise awareness of potential security issues. The aviation industry lags behind the curve when it comes to technology which is evident from two major technological failures causing widespread disruptions in flight operations in the US. Moreover, the decision-making process regarding developing and deploying navigation and communication systems in the aviation industry involves technical and non-technical factors. Thus securing these critical systems is a significant challenge as it will require a complete overhaul of the entire ecosystem and re-designing the systems from the ground up with security baked in the design.

In conclusion, it is vital to understand the threats and feasibility of attacks to develop robust countermeasures and safeguards before such cyber-physical systems are deployed in the field, where a *lack of security* translates to a *lack of safety.*

Bibliography

[1] ACARS. http://www.wavecom.ch/content/ext/DecodermiscHelp/default.htm#!worddocuments/acars.htm.

[2] Air Traffic Activity System (ATADS). https://aspm.faa.gov/opsnet/sys/Airport.asp.

[3] Airport-Shield: Protection against radio interferences in airports. https://www.loginshowroom.com/c-esm-comint/airport-shield-interference-detection-and-geolocalisation/.

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

[5] DRONEKIT: Developer tools for drones. https://dronekit.io/.

[6] Ettus Research LLC. http://www.ettus.com/.

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

[8] GNSS and GPS Simulators. https://www.navtechgps.com/departments/simulators/.

[9] GNURadio: A free & open-source software development toolkit for Signal-Processing. https://www.gnuradio.org/.

[10] 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.

[11] North Atlantic Data Link Mandate March 2020 Update. https://www.faa.gov/air_traffic/publications/internationalnotices/intl_2_20002.html.

[12] Parrot: Professional drones made for work. https://www.parrot.com/us.

[13] Pixhawk 4: An advanced autopilot designed and made in collaboration with Holybro and the PX4 team. https://docs.px4.io/master/en/flight_controller/pixhawk4.html.

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

[15] Space Segment. https://www.gps.gov/systems/gps/space/.

[16] Sporty's SP-400 Handheld NAV/COM Aviation Radio.

[17] 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.

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

[19] Aircraft Serious Incident Report Occurrences Number 00/2518 B767-319ER ZK-NCJ, Civil Aviation Authority of New Zealand, 2002.

[20] Introduction to ACARS Messaging Services, International Communications Group, April 2006. https://www.icao.int/safety/acp/inactive%20working%20groups%20library/acp-wg-m-iridium-7/ird-swg07-wp08%20-%20acars%20app%20note.pdf.

[21] Aeronautical Telecommunications - Surveillance and Collision Avoidance Systems, International Civil Aviation Organization, 2007. https://store.icao.int/.

[22] *Commission Regulation (EC) No 29/2009 of 16 January 2009 laying down requirements on data link services for the single European sky (Text with EEA relevance)2009*. 2009. http://data.europa.eu/eli/reg/2009/29/oj/eng.

[23] Eurocontrol Specification on Data Link Services, 2009. https://www.eurocontrol.int/sites/default/files/publication/files/20090128-dls-spec-v2.1.pdf.

[24] Forget any security concern and welcome Air Force One on Flightradar24!, 2011. https://theaviationist.com/2011/11/24/af1-adsb.

[25] Status Report BFU EX010-11, German Federal Bureau of Aircraft Accident Investigation, 2011.

[26] Acceptable Means of Compliance and Guidance Material to Part-SERA, European Aviation Safety Agency, Sep 2012. https://www.easa.europa.eu/sites/default/files/dfu/NPA%202012-14.pdf.

[27] Yaesu FTA-750L, 2012. https://www.yaesu.com/airband/indexVS.cfm?cmd=DisplayProducts&DivisionID=2&ProdCatID=204&ProdID=1777.

[28] UT Austin Researchers Successfully Spoof an $80 million Yacht at Sea, 2013. https://news.utexas.edu/2013/07/29/ut-austin-researchers-successfully-spoof-an-80-million-yacht-at-sea/.

[29] Stick shaker warning on ILS final, June 2014. https://www.onderzoeksraad.nl/en/onderzoek/1949/stick-shaker-warning-on-ils-final.

[30] Hacking A Phone's GPS May Have Just Got Easier, 2015. http://www.forbes.com/sites/parmyolson/2015/08/07/gps-spoofing-hackers-defcon/.

[31] Hawker Siddeley HS121 Trident, 2017. https://www.baesystems.com/en/heritage/hawker-siddeley-hs121-trident.

[32] Aeronautical Telecommunications - Radio Navigational Aids, Volume 1, 2018. https://store.icao.int/.

[33] Modernization of U.S. Airspace, Aug 2018. https://www.faa.gov/nextgen.

[34] Incorrect altimeter setting results in CFIT, 2020. https://generalaviationnews.com/2020/10/19/incorrect-altimeter-setting-results-in-cfit/.

[35] Air Traffic by the Numbers, 2022.

[36] Administration, F. A. UAS Remote Identification, 2021.

[37] AirSatOne. FANS 1/A (ADS-C & CPDLC), CNS/ATM, Datalink, AFIS / ACARS & CPDLC-DCL. https://www.airsatone.com/datalink-fans-1a-cpdlc-ads-c-acars-vhf-satcom-inmarsat-iridium, Accessed: 2023.

[38] Akos, D. M. Who's afraid of the spoofer? GPS/GNSS spoofing detection via automatic gain control (AGC). *NAVIGATION: Journal of the Institute of Navigation* (2012).

[39] Al), A. B. Drone Payloads: Which Drone Can Carry The Most Weight? https://dronesvue.com/drone-payloads-which-drone-can-carry-the-most-weight/.

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

[41] ArduPilot. ArduCopter: Open-source multicoper UAV controller. `https://ardupilot.org/copter/`.

[42] ArduPilot. EKF Failsafe. `https://ardupilot.org/copter/docs/ekf-inav-failsafe.html`.

[43] Arteaga, S. P., Hernández, L. A. M., Pérez, G. S., Orozco, A. L. S., and Villalba, L. J. G. Analysis of the GPS spoofing vulnerability in the drone 3DR solo. *IEEE Access* (2019).

[44] ASRI. Aeronautical Frequency Committee (AFC) VHF Ground Station Installation Guidelines, 2009. `https://asri.aero/wp-content/uploads/2012/07/VhfStationGuidelines.pdf`.

[45] Autel Robotics. Autel Robotics. `https://www.autelrobotics.com/`.

[46] Autel Robotics. User Manual - Autel EVO II Series. `https://cdn.shopify.com/s/files/1/1538/0803/files/EVO_II_Series_User_Manual_En_Issue_Version.pdf`.

[47] Berthier, P., Fernandez, J. M., and Robert, J.-M. SAT: Security in the air using Tesla. In *Proceedings of the IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)* (2017).

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

[49] Boeing. Statistical Summary Of Commercial Jet Airplane Accidents Worldwide Operations — 1959 – 2020. `https://www.boeing.com/resources/boeingdotcom/company/about_bca/pdf/statsum.pdf`.

[50] Boeing. Statistical Summary of Commercial Jet Airplane Accidents Worldwide Operations — 1959 – 2016, Boeing, 2017. `www.boeing.com/news/techissues/pdf/statsum.pdf`.

[51] Borhani-Darian, P., Li, H., Wu, P., and Closas, P. 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).

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

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

[54] Botargues, P. Airbus ap/fd tcas mode: A new step towards safety improvement. *Safety first*, 7 (2009).

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

[56] Brands, S., and Chaum, D. Distance-bounding protocols. In *Workshop on the theory and application of cryptographic techniques on Advances in cryptology* (1993).

[57] Bresteau, C., Guigui, S., Berthier, P., and Fernandez, J. M. On the security of aeronautical datalink communications: Problems and solutions. In *2018 Integrated Communications, Navigation, Surveillance Conference (ICNS)* (2018).

[58] Broumandan, A., Jafarnia-Jahromi, A., and Lachapelle, G. Spoofing detection, classification and cancelation (SDCC) receiver architecture for a moving GNSS receiver. *GPS Solutions* (2015).

[59] Busset, J., Perrodin, F., Wellig, P., Ott, B., Heutschi, K., Rühl, T., and Nussbaumer, T. 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).

[60] C4ADS. Above Us Only Stars: Exposing GPS Spoofing in Russia and Syria, 2019. `https://www.c4reports.org/aboveusonlystars`.

[61] Capt. Dennis M. McCollum. Evaluation of Instrument Landing System DDM Calibration Accuracies. `http://www.dtic.mil/dtic/tr/fulltext/u2/a138301.pdf`.

[62] Cassola, A., Robertson, W., Kirda, E., and Noubir, G. A Practical, Targeted, and Stealthy Attack Against WPA-Enterprise Authentication. In *Proceedings of the 20th Annual Network & Distributed System Security Symposium, NDSS'13* (2013).

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

[64] Chen, G., and Dong, W. Jamcloak: Reactive jamming attack over cross-technology communication links. In *2018 IEEE 26th International Conference on Network Protocols (ICNP)* (2018).

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

[66] Costin, A., and Francillon, A. Ghost in the Air (Traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices. *black hat USA* (2012).

[67] Council, N. R. *Autonomy Research for Civil Aviation: Toward a New Era of Flight*. The National Academies Press, Washington, DC, 2014.

[68] CRFS. RFeye DroneDefense. `https://www.crfs.com/drone-detection/`.

[69] Cyber, R. Ring - Revolutionary Next Generation Counter-UAS System. `https://www.regulus.com`.

[70] Daneshmand, S., Jafarnia-Jahromi, A., Broumandan, A., and Lachapelle, G. A GNSS structural interference mitigation technique using antenna array processing. In *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)* (2014).

[71] Dave Allen, Martine Blaize, Serge Bagieu, Tom Kraft, Peter Skaves, Anne-Sophie Luce, Tony Martin, Mark Joseph, Elizabeth Noon, Marc Barrere. Interoperability Requirements for ATS Applications Using ARINC 622 Data Communication Document, 1998. `https://www.asas-tn.org/library/standardisationsbodies/eurocae/g1-019.pdf/preview_popup/file`.

[72] Debre, I. Saudi Arabian oil facility struck in drone attack, 2021. `https://www.bbc.com/news/world-middle-east-60082786`.

[73] DHS. Positioning, Navigation, and Timing (PNT) Program. `https://www.dhs.gov/science-and-technology/pnt-program`.

[74] Dive, R. Global uav drones market expected to surpass $102,466.7 million and grow at 19.6% cagr in the 2022 to 2030 timeframe. *Research Dive 2022* (August 2022).

[75] DJI. DJI. `https://www.dji.com`.

[76] DJI. Geo Zone Map - Fly Safe - DJI. `https://www.dji.com/flysafe/geo-map`.

[77] DJI. Mavic 2 - Product Information - DJI. https://www.dji.com/mavic-2/info.

[78] DJI. Mavic 3 - Specs - DJI. https://www.dji.com/mavic-3/specs.

[79] Dressel, L., and Kochenderfer, M. J. Hunting drones with other drones: Tracking a moving radio target. In *International Conference on Robotics and Automation (ICRA)* (2019).

[80] Dunn, M. J. Global Positioning Systems Directorate Systems Engineering & Integration Interface Specification. http://www.gps.gov/technical/icwg/IS-GPS-200G.pdf.

[81] Ebinuma, T. Software-Defined GPS Signal Simulator, 2015. https://github.com/osqzss/gps-sdr-sim.

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

[83] Eskilsson, S., Gustafsson, H., Khan, S., and Gurtov, A. Demonstrating ADS-B and CPDLC Attacks with Software-Defined Radio. In *2020 Integrated Communications Navigation and Surveillance Conference (ICNS)* (2020).

[84] ETS-Lindgren. Broadband Mini-Bicon Antenna - ETS-Lindgren. https://www.ets-lindgren.com/products/antennas/broadband-min-bicon-antennas/4005/400502?page=Products-Item-Page.

[85] FAA. Barometric Altimeter Errors and Setting Procedures. https://www.faa.gov/air_traffic/publications/atpubs/aim_html/chap7_section_2.html.

[86] Falk, C., Martin, L., and Brewer-Dougherty, T. Examination of Airborne Position-Time Estimates From Enroute Automatic Dependent Surveillance. In *AIAA Guidance, Navigation, and Control Conference* (2009).

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

[88] Fantacci, R., Menci, S., Micciullo, L., and Pierucci, L. A secure radio communication system based on an efficient speech watermarking approach. *Proceedings of the Security and Communication Networks* (2009).

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

[90] Federal Aviation Administration. Timeline of FAA and Aerospace History. `https://www.faa.gov/about/history/timeline`.

[91] Federal Aviation Administration (FAA). Unmanned Aircraft Systems (UAS). `https://www.faa.gov/uas`.

[92] Fernández-Hernández, I., Rijmen, V., Seco-Granados, G., Simon, J., Rodríguez, I., and Calle, J. D. A Navigation Message Authentication Proposal for the Galileo Open Service. *Navigation* (2016).

[93] Fernandez-Prades, C., Arribas, J., Closas, P., Aviles, C., and Esteve, L. 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).

[94] Fernández-Prades, C., Arribas, J., Esteve, L., Pubill, D., and Closas, P. 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).

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

[96] Gaspar, J., Ferreira, R., Sebastião, P., and Souto, N. Capture of UAVs through GPS spoofing. In *2018 Global Wireless Summit (GWS)* (2018).

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

[98] Goldstein, M., Kirschbaum, J., et al. GPS disruptions: efforts to assess risks to critical infrastructure and coordinate agency actions should be enhanced.

[99] Goodin, D. US spy drone hijacked with GPS spoof hack, report says. *The Register* (2011).

[100] Goward, D. One GPS Mystery Solved, Another Remains. *GPS World* (2023).

[101] Gurtner, W., and Estey, L. Rinex-the receiver independent exchange format-version 3.00. *Astronomical Institute, University of Bern and UNAVCO, Bolulder, Colorado.* (2007).

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

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

[104] Hamilton, B. A. ASRS - Aviation Safety Reporting System. `https://asrs.arc.nasa.gov`.

[105] Harris, M. Ghost ships, crop circles, and soft gold: A GPS mystery in Shanghai, 2019. `https://www.technologyreview.com/s/614689/ghost-ships-crop-circles-and-soft-gold-a-gps-mystery-in-shanghai/`.

[106] Haykin, S. *Communication systems*. John Wiley & Sons, 2008.

[107] He, D., Qiao, Y., Chen, S., Du, X., Chen, W., Zhu, S., and Guizani, M. A friendly and low-cost technique for capturing non-cooperative civilian unmanned aerial vehicles. *IEEE Network* (2018).

[108] Henner, J. ACARS (Aircraft Communications Addressing and Reporting System). Master's thesis, VŠB - Technical University of Ostrava, Fakulta elektrotechniky a informatikyOstrava, 2010. SUPERVISOR: Michal Krumnikl.

[109] Honeywell Aerospace. Review of Aviation Mandates. `https://pages3.honeywell.com/rs/honeywell3/images/hon___aviation_mandates_whitepaper_d3b_revised.PDF`.

[110] Hormann, K., and Agathos, A. The point in polygon problem for arbitrary polygons. *Computational Geometry* (2001).

[111] Huang, L., and Yang, Q. Low-cost GPS simulator GPS spoofing by SDR, 2015.

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

[113] IATA. Industry Statistics Fact Sheet. `https://www.iata.org/en/iata-repository/publications/economic-reports/airline-industry-economic-performance---october-2021---data-tables/`.

[114] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. 2.2.6.1.

[115] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. 4.5.

[116] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. 2.2.4.7.1.

[117] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. 3.1.2.

[118] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. 2.2.4.3.2.

[119] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. 2.2.6.3.2.

[120] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. Appendix A.3/4.

[121] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. 5.3.5.

[122] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. 2.2.5.2.5.

[123] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. E4.2.1.1.

[124] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. E4.2.3.4.

[125] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. 2.2.6.5.

[126] ICAO. Global Operational Data Link Document (GOLD) 2^nd Edition, 2014.

[127] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. 2.2.6.3.6.

[128] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. 2.2.4.8.1.2.

[129] ICAO. *Global Operational Data Link Document (GOLD) 2nd Edition.* 2014, ch. 4.2.3.3.

[130] Inc., N. Motion Capture Systems. `https://www.optitrack.com/`.

[131] Insights, D. I. Top 10 Drone Manufacturer's Market Shares in the US. https://droneii.com/product/drone-manufacturers-ranking.

[132] Intelligence, M. Drones Market - Growth, Trends, COVID-19 Impact, And Forecasts (2022 - 2027), 2022. https://www.mordorintelligence.com/industry-reports/drones-market.

[133] International Telecommunication Union. *Radio Regulations*. 2012. http://search.itu.int/history/HistoryDigitalCollectionDocLibrary/1.43.48.en.101.pdf.

[134] ITU-R - Radiocommunications Sector for ITU. Reception of automatic dependent surveillance-broadcast via satellite and compatibility studies with incumbent systems in the frequency band 1 087.7-1 092.3 MHz.

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

[136] Jansen, K., Schäfer, M., Moser, D., Lenders, V., Pöpper, C., and Schmitt, J. Crowd-GPS-sec: Leveraging crowdsourcing to detect and localize GPS spoofing attacks. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)* (2018).

[137] Jansen, K., Tippenhauer, N. O., and Pöpper, C. Multi-receiver GPS spoofing detection: Error models and realization. In *Proceedings of the 32nd Annual Conference on Computer Security Applications* (2016).

[138] Jiang, X., Zhang, J., Harding, B. J., Makela, J. J., Domi, A. D., et al. Spoofing GPS receiver clock offset of phasor measurement units. *IEEE Transactions on Power Systems* (2013).

[139] Jon S. Warner, R. G. J. A Simple Demonstration that the Global Positioning System (GPS) is Vulnerable to Spoofing, 2003. https://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-03-2384.

[140] Kendoul, F., Fantoni, I., and Nonami, K. Optic flow-based vision system for autonomous 3d localization and control of small aerial vehicles. *Robotics and autonomous systems 57*, 6-7 (2009), 591–602.

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

[142] Kerns, A. J., Wesson, K. D., and Humphreys, T. E. A blueprint for civil GPS navigation message authentication. In *Proceedings of the IEEE/ION Symposium on Position, Location and Navigation Symposium (PLANS)* (2014).

[143] Khan, S., Gurtov, A., Breaken, A., and Kumar, P. A Security Model for Controller-Pilot Data Communication Link. In *2021 Integrated Communications Navigation and Surveillance Conference (ICNS)* (2021).

[144] Khanafseh, S., Roshan, N., Langel, S., Chan, F.-C., Joerger, M., and Pervan, B. GPS spoofing detection using RAIM with INS coupling. In *IEEE/ION Position, Location and Navigation Symposium (PLANS)* (2014).

[145] Konovaltsev, A., Caizzone, S., Cuntz, M., and Meurer, M. Autonomous spoofing detection and mitigation with a miniaturized adaptive antenna array. In *Proceedings of the 27th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2014)* (2014).

[146] Kuhn, M. G. An asymmetric security mechanism for navigation signals. In *International Workshop on Information Hiding* (2004).

[147] Leconte, T. An ACARS SDR decoder for Airspy and rtl-sdr. `https://github.com/TLeconte/acarsdec`.

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

[149] Leipold, F. Session 5: Views of Airlines and Pilots Lufthansa Airlines 2014-05-27, May 2014.

[150] Lemiech, T. Libacars: A library for decoding ACARS message contents. `https://github.com/szpajder/libacars`.

[151] Lemiech, T. VDL Mode 2 message decoder and protocol analyzer, 2017. `https://github.com/szpajder/dumpvdl2`.

[152] Liu, Y., Li, S., Fu, Q., Liu, Z., and Zhou, Q. Analysis of Kalman filter innovation-based GNSS spoofing detection method for INS/GNSS integrated navigation system. *IEEE Sensors Journal* (2019).

[153] Lo, S. C., and Enge, P. K. Authenticating aviation augmentation system broadcasts. In *Proceedings of the IEEE/ION Position, Location and Navigation Symposium* (2010).

[154] Ltd., L. M. LimeSDR: Low cost, open source, apps-enabled software defined radio (SDR) platform. https://limemicro.com/products/boards/limesdr/.

[155] Luo, A. Drones Hijacking - multi-dimensional attack vectors and countermeasures, 2016. https://www.youtube.com/watch?v=u9nFdOvA8eI&ab_channel=SecurityHub.

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

[157] Magazu III, D. Exploiting the automatic dependent surveillance-broadcast system via false target injection.

[158] Magiera, J., and Katulski, R. Detection and mitigation of GPS spoofing based on antenna array processing. *Journal of applied research and technology* (2015).

[159] Mao, G., Drake, S., and Anderson, B. D. Design of an extended Kalman filter for UAV localization. In *2007 Information, Decision and Control* (2007).

[160] Marcus, J. Yemen rebel attack on UAE throws challenge to the region, 2022. https://www.bbc.com/news/world-middle-east-60082786.

[161] McCallie, D. L. Exploring Potential ADS-B Vulnerabilities in the FAA's Nextgen Air Transportation System.

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

[163] McMilin, E., De Lorenzo, D. S., Lee, T., Enge, P., 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).

[164] McParland, T. Application Level Security Considerations. In *Aeronautical Communication Panel Working Group N – Networking Subgroup N4 - Security* (2004).

[165] Meurer, M., Konovaltsev, A., Appel, M., and Cuntz, M. Direction-of-Arrival Assisted Sequential Spoofing Detection and Mitigation. In *Proceedings of the 2016 International Technical Meeting of The Institute of Navigation, Monterey, California* (2016).

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

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

[168] Narain, S., Ranganathan, A., and Noubir, G. Security of GPS/INS based on-road location tracking systems. In *IEEE Symposium on Security and Privacy (S&P)* (2019).

[169] Narain, S., Ranganathan, A., and Noubir, G. Security of GPS/INS Based On-road Location Tracking Systems. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)* (2019).

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

[171] Naval Air Warfare Center. Electronic Warfare and Radar Systems Engineering Handbook, 2013. `http://www.navair.navy.mil/nawcwd/ewssa/downloads/nawcwd%20tp%208347.pdf`.

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

[173] News, B. Heathrow airport: Drone sighting halts departures. `https://www.bbc.com/news/uk-46803713`.

[174] Nguyen, D., Sahin, C., Shishkin, B., Kandasamy, N., and Dandekar, K. R. A real-time and protocol-aware reactive jamming framework built on software-defined radios. In *Proceedings of the 2014 ACM workshop on Software radio implementation forum* (2014).

[175] Nighswander, T., Ledvina, B., Diamond, J., Brumley, R., and Brumley, D. GPS Software Attacks. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security* (2012).

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

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

[178] Olive, M. ACARS Message Security (AMS) as a Vehicle for Validation of ICAO Doc. 9880 Part IV-B Security, Requirements. *ICAO ACP WG-M Meeting 14 M Meeting 14* (2009).

[179] Ophir RF Inc. Ophir RF Model 5293. https://ophirrf.com/wp-content/uploads/2015/09/5293-2.pdf.

[180] Papadimitratos, P., and Jovanovic, A. GNSS-based positioning: Attacks and countermeasures. In *MILCOM 2008-2008 IEEE Military Communications Conference* (2008).

[181] Partners, P. Jeopardising aircraft through TCAS spoofing. https://www.pentestpartners.com/security-blog/jeopardising-aircraft-through-tcas-spoofing/.

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

[183] Patty, A. Fatal consequences of miscommunication between pilots and air traffic controllers. *The Sydney Morning Herald* (2016). https://www.smh.com.au/business/workplace/the-fatal-consequences-of-miscommunication-between-pilots-and-air-traffic-controllers-20160928-grq1d9.html.

[184] Petri, J. How Hackers Can Take Over Your Car's GPS, 2019. https://www.bloomberg.com/news/articles/2019-06-19/threat-of-gps-spoofing-for-autonomous-cars-seen-as-overblown.

[185] Pichavant, C. VHF antenna radio patterns to support ITU WRC-23 Agenda Item 1.7 on Space-based VHF, 2021. https://www.icao.int/safety/FSMP/MeetingDocs/FSMP%20WG11/WP/FSMP-WG11-WP11_Airbus_VHF%20antenna%20pattern%20to%20support%20ITU%20WRC-23%20AI%201.7.doc.

[186] Pierpaoli, P., Egerstedt, M., and Rahmani, A. Altering UAV flight path by threatening collision. In *Proceedings of the IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)* (2015).

[187] Pierpaoli, P., Egerstedt, M., and Rahmani, A. Altering uav flight path by threatening collision. In *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)* (2015), IEEE, pp. 4A4–1.

[188] Pleban, J.-S., Band, R., and Creutzburg, R. Hacking and securing the AR. Drone 2.0 quadcopter: investigations for improving the security of a toy. In *Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications* (2014).

[189] Pöpper, C., Tippenhauer, N. O., Danev, B., and Capkun, S. Investigation of signal and message manipulations on the wireless channel. In *Proceedings of the European Symposium on Research in Computer Security* (2011).

[190] Press, A. UAE Bans Flying of Recreational Drones After Fatal Attack. `https://www.voanews.com/a/uae-bans-flying-of-recreational-drones-after-fatal-attack/6408720.html`.

[191] Psiaki, M. L., and Humphreys, T. E. GNSS spoofing and detection. *Proceedings of the IEEE* (2016).

[192] Psiaki, M. L., O'Hanlon, B. W., Bhatti, J. A., Shepard, D. P., and Humphreys, T. E. Civilian GPS spoofing detection based on dual receiver correlation of military signals. In *Radionavigation Laboratory Conference Proceedings* (2011).

[193] Qiao, H., Liu, Y., Yang, A., and Hancke, G. Preventing Overshadowing Attacks in Self-Jamming Audio Channels. *IEEE Transactions on Dependable and Secure Computing* (2018).

[194] Ranganathan, A., and Capkun, S. Are we really close? Verifying proximity in wireless systems. *IEEE Security & Privacy* (2017).

[195] Ranganathan, A., Ólafsdóttir, H., and Capkun, S. SPREE: A spoofing resistant GPS receiver. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking* (2016), ACM.

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

[197] Robotics, O. Gazebo. `http://gazebosim.org/`.

[198] Rodday, N. M., Schmidt, R. d. O., and Pras, A. Exploring security vulnerabilities of unmanned aerial vehicles. In *IEEE/IFIP Network Operations and Management Symposium (NOMS)* (2016).

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

[200] Roy, A. Secure aircraft communications addressing and reporting system (ACARS), 2004. US Patent 6,677,888.

[201] Sampigethaya, K., Poovendran, R., and Bushnell, L. Assessment and mitigation of cyber exploits in future aircraft surveillance. In *Proceedings of the IEEE Aerospace Conference* (2010).

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

[203] Sathaye, H., LaMountain, G., Closas, P., and Ranganathan, A. SemperFi: Anti-Spoofing GPS Receiver for UAVs. In *Network and Distributed Systems Security (NDSS) Symposium* (2022).

[204] Sathaye, H., Schepers, D., Ranganathan, A., and Noubir, G. Wireless attacks on aircraft instrument landing systems. In *28th USENIX Security Symposium (USENIX Security 19)* (2019).

[205] Schäfer, M., Lenders, V., and Martinovic, I. Experimental analysis of attacks on next generation air traffic communication. In *International Conference on Applied Cryptography and Network Security* (2013).

[206] Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., and Wilhelm, M. Bringing up OpenSky: A large-scale ADS-B sensor network for research. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks* (2014).

[207] SDRPlay. Decoding ACARS messages using SDRuno and MultiPSK, 2017. `https://www.sdrplay.com/docs/SDRuno_ACARS.pdf`.

[208] Seo, S.-H., Lee, B.-H., Im, S.-H., and Jee, G.-I. Effect of spoofing on unmanned aerial vehicle using counterfeited GPS signal. *Journal of Positioning, Navigation, and Timing* (2015).

[209] Sestorp, I., and Lehto, A. CPDLC in Practice: a Dissection of the Controller Pilot Data Link Communication Security, 2019.

[210] Shackle, S. The mystery of the Gatwick drone. `https://www.theguardian.com/uk-news/2020/dec/01/the-mystery-of-the-gatwick-drone`.

[211] Shen, J., Won, J. Y., Chen, Z., and Chen, Q. A. Drift with devil: Security of multi-sensor fusion based localization in high-level autonomous driving under {GPS} spoofing. In *29th USENIX Security Symposium (USENIX Security 20)* (2020).

[212] Shepard, D. P., Bhatti, J. A., and Humphreys, T. E. Drone hack: Spoofing attack demonstration on a civilian unmanned aerial vehicle. *GPS World* (2012).

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

[214] Siegel, D., and Hansman, R. J. Development of an autoland system for general aviation aircraft.

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

[216] Smailes, J., Moser, D., Smith, M., Strohmeier, M., Lenders, V., and Martinovic, I. You talkin'to me? Exploring Practical Attacks on Controller Pilot Data Link Communications. In *Proceedings of the 7th ACM on Cyber-Physical System Security Workshop* (2021).

[217] Smith, M., Moser, D., Strohmeier, M., Lenders, V., and Martinovic, I. Undermining privacy in the aircraft communications addressing and reporting system (ACARS). *Proceedings on Privacy Enhancing Technologies* (2018).

[218] Smith, M., Strohmeier, M., Lenders, V., and Martinovic, I. On the security and privacy of ACARS. In *Proceedings of Integrated Communications Navigation and Surveillance (ICNS)* (2016).

[219] Smith, M., Strohmeier, M., Lenders, V., and Martinovic, I. Understanding realistic attacks on airborne collision avoidance systems. *Journal of Transportation Security* (2022).

[220] Stelkens-Kobsch, T. H., Hasselberg, A., Mühlhausen, T., Carstengerdes, N., Finke, M., and Neeteson, C. Towards a more secure ATC voice communications system. In *Proceedings of the IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)* (2015).

[221] Strasser, M., Danev, B., and Čapkun, S. Detection of reactive jamming in sensor networks. *ACM Transactions on Sensor Networks (TOSN)* (2010).

[222] Strohmeier, M., Lenders, V., and Martinovic, I. On the security of the automatic dependent surveillance-broadcast protocol. *IEEE Communications Surveys & Tutorials* (2014).

[223] Strohmeier, M., Lenders, V., and Martinovic, I. On the Security of the Automatic Dependent Surveillance-Broadcast Protocol. *IEEE Communications Surveys Tutorials* (2015).

[224] Strohmeier, M., Schäfer, M., Pinheiro, R., Lenders, V., and Martinovic, I. On perception and reality in wireless air traffic communication security. *IEEE transactions on intelligent transportation systems* (2016).

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

[226] Tart, A., and Trump, T. Addressing security issues in ADS-B with robust two dimensional generalized sidelobe canceller. In *Proceedings of 22nd International Conference on Digital Signal Processing (DSP)* (2017).

[227] Teso, H. Aircraft hacking: Practical aero series. In *Proceedings of HITB Security Conference* (2013).

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

[229] Titterton, D. H., and Weston, J. L. Strapdown Inertial Navigation Technology. 2nd. *London: Institution of Electrical Engineers* (2004).

[230] u-Blox. NEO-M8 DataSheet. `https://www.u-blox.com/en/ubx-viewer/view/NEO-M8-FW3_DataSheet_UBX-15031086.pdf`.

[231] UPS. UPS Flight Forward Receives FAA's First Full Part 135 Standard Certification for Drone Airline. `https://www.ups.com/us/en/about/news/2020/09/01/ups-flight-forward-receives-faa-first-full-part-135-standard-certification-for-drone-airline.page`, September 2020.

[232] U.S. Department of Transportation. *Nondirectional Beacon (NDB) Installation Standards Handbook*. 1981.

[233] U.S. Department of Transportation. *Instrument Flying Handbook*. 2012.

[234] Vanhoef, M., and Piessens, F. Advanced Wi-Fi attacks using commodity hardware. In *Proceedings of the 30th Annual Computer Security Applications Conference* (2014).

[235] Vincent, J. Recreational drones banned in United Arab Emirates after oil facility attack, 2022. https://www.theverge.com/2022/1/24/22898614/united-arab-emirates-uae-ban-recreational-drone-attack.

[236] Vo-Huu, T. D., Vo-Huu, T. D., and Noubir, G. Interleaving Jamming in Wi-Fi Networks. In *Proceedings of the 9th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (2016).

[237] Warner, J. S., and Johnston, R. G. A simple demonstration that the global positioning system (GPS) is vulnerable to spoofing. *Journal of security administration* (2002).

[238] Warner, J. S., and Johnston, R. G. GPS spoofing countermeasures. *Homeland Security Journal* (2003).

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

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

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

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

[243] Wilhelm, M., Martinovic, I., Schmitt, J. B., and Lenders, V. Short paper: Reactive jamming in wireless networks: How realistic is the threat? In *Proceedings of the fourth ACM conference on Wireless network security* (2011).

[244] Wing, M. G., Eklund, A., and Kellogg, L. D. Consumer-grade global positioning system (GPS) accuracy and reliability. *Journal of forestry* (2005).

[245] Yusupov, L. ADSB-Out: Add-on for SoftRF-Emu, Stratux, etc..., 2017. https://github.com/lyusupov/ADSB-Out.

[246] Zeng, K. C., Liu, S., Shu, Y., Wang, D., Li, H., Dou, Y., Wang, G., and Yang, Y. All your {GPS} are belong to us: Towards stealthy manipulation of road navigation systems. In *Proceedings of the 27th USENIX Security Symposium* (2018).

[247] Zhang, R., Liu, G., Liu, J., and Nees, J. P. Analysis of message attacks in aviation data-link communication. *IEEE Access* (2017).

[248] Zhang, T., and Zhu, Q. Strategic defense against deceptive civilian GPS spoofing of unmanned aerial vehicles. In *Proceedings of the International Conference on Decision and Game Theory for Security* (2017).

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

[250] Zhou, C., Yan, Q., Shi, Y., and Sun, L. DoubleStar: Long-range attack towards depth estimation based obstacle avoidance in autonomous systems. *arXiv preprint arXiv:2110.03154* (2021).

Appendix A

A.1 Overview of Automation in Aerial Vehicles

Automation in aerial vehicles refers to using advanced computer systems and software to control various aspects of flight, from navigation and communication to takeoff and landing. This technology has rapidly advanced over the past several decades, significantly improving aviation safety, efficiency, and reliability. Compared to crewed aircraft, UAVs have a higher dependence on automation due to the absence of a human pilot. However, the underlying principles of automation remain the same. Automation in crewed aircraft can be broken into four broad categories, and the flight management system (FMS) is responsible for managing and coordinating each of these systems.

Maneuver automation: More popularly known as the "autopilot". It involves systems that enable the automatic control of an aircraft's movement, including its altitude, heading, and speed, through the use of autopilot and auto-throttle systems. This type of automation was first introduced in 1914 [67], and it has truly revolutionized the aviation industry as it equipped the aircraft to fly itself with minimal human intervention. This system usually implements some form of a proportional–integral–derivative (PID) controller to adjust control surface movements to achieve the desired pitch, roll, and yaw, rotation components that allow motion in three-dimensional space. Figure A.1 the three fundamental rotational axes that describe the orientation of an object in 3D space. Section 4.2 for details on PID controller implementation in UAVs (fixed-wing aircraft also follow similar principles). Control surfaces are the physical parts that enable to aircraft to change its orientation. These are ailerons, flaps, rudders, and elevators in fixed-wing aircraft (e.g., Boeing 737) and individual propellers on a quadcopter (e.g., DJI Mavic Pro).

Navigation Automation: It involves systems that assist with flight planning, route management, and guidance to ensure the aircraft is on course. These systems use various sensors, such as **GPS**, inertial navigation systems, and radio navigation aids like the **ILS**,

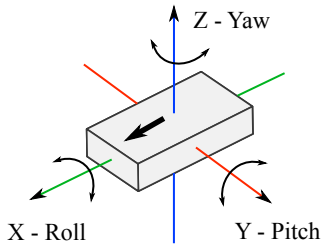


Figure A.1: Three fundamental rotational axes that describe the orientation of an object in 3D space

to determine the aircraft's position and provide guidance to the pilot. It also leverages aviation datalink applications like CPDLC for receiving flight plan updates and strategic Air traffic controllers (ATC) instructions like take-off clearance and post-take-off procedures. Navigation automation is of paramount importance in UAVs. Autonomous vehicles, unlike crewed aircraft, lack offline human oversight[1]; hence, such vehicles have to rely heavily on sensors for situational awareness and localization. For example, the onboard satellite navigation receiver is the primary source of positioning and navigation information. In certain aircraft types, navigation and maneuver automation works closely to provide full self-flying capability. Examples of such a system are:

1. **Autoland:** ILS and autopilot-equipped aircraft can interpret radio signals to determine the aircraft's position with respect to the prescribed approach path and make adjustments to aircraft control surfaces, including throttles, to maneuver the aircraft and safely land. In 1964, United Airlines became the first commercial operator to perform a completely computerized touchdown [90].

2. **En-route Navigation:** This refers to maneuvering the aircraft to follow a flight plan or pre-determined mission. The FMS uses the aircraft's location from various sources like GPS, INS, radio-nav aids, or even vision-based opti-flow systems [140] and calculates correction maneuvers to perform that keep the aircraft on course. Section 4.2 for more details on using satellite navigation in course correction.

Surveillance and Reporting Automation: These systems provide information on the aircraft's position, altitude, and speed to air traffic control and other aircraft. These systems use various technologies such as radar, automatic dependent surveillance-broadcast (ADS-B), aviation datalink applications like automatic dependent surveillance-contract (ADS-C),

[1]UAVs have the remote piloting capability; however, in beyond-line-of-sight scenarios, such control is heavily restricted.

and transponders to detect and report the aircraft's position and status to ground-based stations or other aircraft. ADS-B and ADS-C are key surveillance automation systems being adopted globally as a replacement for radar and void-based surveillance and reporting. ADS-B broadcasts the aircraft's position, velocity, and other information to air traffic control and other aircraft, providing a more accurate and reliable means of tracking aircraft than traditional radar-based systems. ADS-C is similar to ADS-B, except it allows additional reporting on certain flight events, like flight plan updates on a pre-defined contract. Section 3.2 for more information. These technologies together enable collision avoidance systems (TCAS/ACAS/ACAS-x) with automatic conflict resolution advisories [54]. Similarly, in 2019 FAA announced a remote identification technology "Remote ID" that provides ADS-B-like surveillance capabilities for UAVs [36].

Aircraft Maintenance Automation: Aircraft maintenance automation systems monitor and diagnose the health of the aircraft's systems and components, such as the engine, avionics, and other critical systems. These systems use sensors and diagnostic algorithms to detect anomalies or deviations from normal operating parameters and alert the pilot or maintenance personnel to potential problems. The Engine Indication and Crew Alerting System (EICAS) is an example of an aircraft maintenance automation system that provides real-time engine and system status information and alerts the pilot to any malfunctions or abnormal conditions. The EICAS can help ensure safe and reliable operation by detecting and alerting the pilot to potential problems before they become serious. EICAS operation is often coupled with **aircraft communications, addressing, and reporting system (ACARS)**, a digital communication system used to wirelessly transmit aircraft maintenance data to ground engineers in real-time [37].

A.2 Talks and Presentations

1. DEFCON Aviation Village 2019, Las Vegas, NV, USA
    - Wireless Attacks on Aircraft ILS
2. DEFCON Aerospace Village 2020, Online
    - GPS Spoofing 101
    - A Deeper Dive into ILS and ADS B Spoofing
3. TEDx Northeastern 2020, Online
    - Securing Skies: Propelling Cybersecurity in Aviation

4. EUROCONTROL Radio-navigation systems WG 2021, Online
    - Security Analysis of ILS
5. Cyber-alp Retreat 2022, Sachseln, Switzerland
    - An Experimental Study of GPS Spoofing and Takeover Attacks on UAVs

A.3 Video Demonstrations

1. **Instrument Landing System Attacks**
   A video demonstration of our proposed overshadow attack on flight simulator setup and SDRs.
   `https://www.youtube.com/watch?v=Wp4CpyxYJq4`
2. **UAV Takeover via GPS Spoofing**
   A real-time video demonstration of HiTL GPS spoofer controlling a UAV.
   `https://youtu.be/4vfqoYXSHRY`

A.4 Media Coverage

Wireless Attacks on Aircraft Instrument Landing Systems

1. Northeastern University researchers show that hacking an airplane's landing instruments isn't as hard as it should be
   `https://news.northeastern.edu/2019/06/17/northeastern-university-researchers-show-that-hacking-an-airplanes-landing-instruments-isnt-as-hard-as-it-should-be/`
2. The radio navigation planes use to land safely is insecure and can be hacked
   `https://arstechnica.com/information-technology/2019/05/the-radio-navigation-planes-use-to-land-safely-is-insecure-and-can-be-hacked/`
3. When an aircraft landing system is made to enter the spoofing zone
   `https://techxplore.com/news/2019-05-aircraft-spoofing-zone.html`
4. Researchers Hack Aircraft Landing System with $600 Radios
   `https://securityledger.com/2019/05/researchers-hack-aircraft-landing-system-with-600-radios/`
5. Plane Landing Systems Are Easily Hackable, Say Researchers
   `https://www.theregister.co.uk/2019/05/16/airplane_landing_security/`

6. Hackers can control the landing of the airplane by intercepting the radio signals that the pilot relies on and using their own system to guide it to the ground
   https://gigazine.net/gsc_news/en/20190517-hackers-aircraft-landing-fake/

7. The plane it's splained falls mainly without the brain: We chat to boffins who've found a way to disrupt landings using off-the-shelf radio
   https://www.theregister.com/2019/05/16/airplane_landing_security/

8. Radio signals used to land planes easily HACKED using tools amounting to just $600
   https://www.dailymail.co.uk/sciencetech/article-7032603/Radio-signals-used-land-planes-easily-HACKED-using-tools-amounting-just-600.html

9. Hackers can fake radio signals to hijack aircraft landing systems, warn researchers
   https://www.computing.co.uk/news/3075890/hackers-aircraft-landing-fake-radio-signals

10. Spoofing aircraft instrument landing systems with an SDR
    https://www.rtl-sdr.com/tag/instrument-landing-system/

A.5 Project Repositories

1. SemperFi

Link: https://github.com/harshadms/semperfi

This repository contains a PoC implementation of the GPS receiver capable of uninterrupted operations even under adversarial settings. It contains a working prototype of the proposed successive interference cancellation technique for legitimate signal recovery. This prototype is implemented using an open-source software-defined GNSS receiver, GNSS-SDR.

2. Human-in-the-loop GPS Spoofer

Link: https://github.com/harshadms/hitl_spoofer

This repository contains our implementation of the human-in-the-loop GPS spoofer capable of real-time and arbitrary manipulation of spoofed GPS motion through a human interfaces device like a keyboard or a joystick. Such a system allows a user to visually observe the target UAV and control it via GPS spoofing like a video game. It also allows reference input from an external tracking system via UDP sockets. HITL Spoofer is based on the popular open-source GPS signal generator *GPS-SDR-SIM* and can interface with SDRs like LimeSDR and USRP.

3. Anti-spoofing Techniques for GNSS Receivers - Google Summer of Code 2021

Link: `https://github.com/harshadms/gsoc_21_gnss-sdr`

This repository contains a project undertaken as part of Google Summer of Code in 2021. This project aims at implementing well-known physical and logical layer solutions in GNSS-SDR, an open-source software-defined receiver that provides signal processing software for multiple satellite navigation systems. Spoofing detection module implemented in GNSS-SDR checks various properties like power level, carrier-to-noise density ratio (C/No), cross-ambiguity function matrix, position consistency, and clock consistency to determine active spoofing attacks.