Cybersecurity attacks on software logic and error handling within ADS-B implementations: Systematic testing of resilience and countermeasures

Syed Khandker, Hannu Turtiainen, Andrei Costin, and Timo Hämäläinen

***Abstract*—Automatic Dependent Surveillance-Broadcast (ADS-B) is a cornerstone of the next-generation digital sky and is now mandated in several countries. However, there have been many reports of serious security vulnerabilities in the ADS-B architecture. In this paper, we demonstrate and evaluate the impact of multiple cyberattacks on ADS-B via remote radio frequency links that affected various network, processing, and display subsystems used within the ADS-B ecosystem.**

Overall we implemented and tested 12 cyberattacks on ADS-B in a controlled environment, out of which 5 attacks were presented or implemented for the first time. For all these attacks, we developed a unique testbed that consisted of 13 hardware devices and 22 software that ran on Android, iOS, Linux, and Windows operating systems, which result in a total of 36 tested configurations. Each of the attacks was successful on various subsets of the tested configurations. In some attacks, we discovered wide qualitative variations and discrepancies in how particular configurations react to and treat ADS-B inputs that contain errors or contradicting flight information, with the main culprit almost always being the software implementation. In some other attacks, we managed to cause Denial of Service (DoS) by remotely crashing/impacting more than 50% of the test-set that corresponded to those attacks.

Besides demonstrating successful attacks, we also implemented, investigated, and report herein some practical countermeasures to these attacks. We demonstrated that the strong relationship between the received signal strength and the distance-to-emitter might help verify the aircraft's advertised ADS-B position and distance. For example, our best machine learning models achieved 90% accuracy in detecting spoofed ADS-B signals, which may be effectively used to distinguish ADS-B signals of real aircraft from spoofed signals of attackers.

***Index Terms*—ADS-B, 1090ES, UAT, EFB, 1090MHz, 978MHz, aviation, avionics, ATC, ATM, datalink, cybersecurity, vulnerabilities, pentesting, experimental platform, countermeasures.**

I. Introduction

AUTOMATIC Dependent Surveillance-Broadcast (ADS-B) is a surveillance technology where by the position, identity, velocity, and other information of an aircraft are periodically broadcast up to 6.2 times in a second via a radio link to inform other aircraft and the ground station in the vicinity about the aircraft. ADS-B is designed to make air traffic control easier, to eliminate the limitations of the currently used Modes A and C, to improve the positioning accuracy of aircraft via satellite navigation system, and replace the secondary surveillance radar (SSR) in the future. However, ADS-B is not secure because it does not use basic security measures, e.g., authentication, encryption. The U.S. Federal Aviation Administration (FAA) claims that unencrypted data links are necessary due to operational requirements [1]. The lack of basic security mechanisms of the ADS-B signal makes them easy to forge or tamper with, which affects the confidentiality, integrity, and availability of the transmitted aircraft data [2].

At the same time, cyberattacks in the aviation industry are increasing. Wireless attacks such as jamming, Denial-of-Service (DoS), and spoofing are becoming common [3]. For example, a security expert had allegedly broken into an aircraft control system using an in-flight entertainment network [4]. A hacker had demonstrated the possibility of remotely attacking and hijacking an airplane using an Android device [5]. A false hijacking alarm has been triggered in a WestJet flight [6]. Ground testing of ADS-B equipment triggered a fake Traffic Collision Avoidance System (TCAS) alert on a Boeing 737 while landing [7]. Attackers targeting a specific insecure protocol, may formulate many new types of attacks that had not been investigated before in a specific context. For example, to the best of our knowledge, we are the first to notice and subsequently study that two ADS-B signals with the same International Civil Aviation Organization (ICAO) address [1] but different flight information (e.g., location) can induce logical vulnerability of an ADS-B receiver and hence, pose operational and decisional risks. Moreover, many countries already have strict ADS-B mandates. Effective January 1, 2020, aircraft operating in the continental United States are required to be ADS-B-enabled [8]. The European Union has also mandated the gradual use of ADS-B by all aircraft flying over its skies starting in June 2020 [9]. However, all current ADS-B solutions use the only available ADS-B protocol, which is "insecure by default" in many ways. Missing basic security measures and the evolution of transmission-enabled software-defined radio (SDR) technology have made ADS-B vulnerable to unprecedented challenges from cybersecurity attacks. ADS-B receivers are also becoming very handy. The proliferation of mobile devices enables quick deployment of mobile cockpit services using different electronic flight bag (EFB) applications and portable ADS-B transceivers such as SkyEcho2, Sentry, and echoUAT [10]. These mobile solutions, due to their low cost as well as ease of installation and usage, are becoming popular among users of general aviation

S. Khandker, H. Turtainen, A. Costin and T. Hämäläinen are with the Faculty of Information Technology of the University of Jyväskylä, P.O. Box 35, Jyväskylä, 40014 Finland E-mail: syibkhan@jyu.fi, turthzu@jyu.fi, ancostin@jyu.fi, timoh@jyu.fi

[1]Also known as "ICAO24 code".

(GA) aircraft (e.g., business jets and aircraft of hobbyist pilots). However, most affordable mobile cockpit solutions in the current literature are untested against cyberattacks. Many studies had outlined several types of attacks against ADS-B [11], [12], [13], [14], [15], but only a few of them practically and deeply investigated the cybersecurity concerns. The lack of thorough investigation of radio frequency (RF) link-based attacks on ADS-B, the attacks' impact on various ADS-B installations, and the error-handling capabilities of diverse ADS-B setups motivated us to conduct this research. In this article, we revisit the lack of security of ADS-B and investigate several systematic cyberattacks on the ADS-B system. Our main contributions with this work are:

i) Practical implementation of several new (and some existing) attack concepts mainly against ADS-B over an RF link.
ii) Thorough investigation and reporting of responses to attacks against, as well as, system resilience and error-handling capabilities of, a wide range of ADS-B setups.
iii) Effective demonstration of the feasibility and usability of the Received Signal Strength (RSS)-Distance model in distinguishing between genuine and attacker-originated ADS-B signals.

The rest of this article is organized as follows. Details of the ADS-B communication system are described in Section II. Related studies are discussed in Section III. Different attack scenarios are presented in Section IV. Details of our test platform, attack implementation, and experimental setup are presented in Section V. Our results and analysis are explained in Section VI. The detection of ADS-B spoofing by the RSS-Distance model and some other countermeasures are demonstrated in Section VII. Finally, possible workarounds, future studies, and the conclusion are presented in Section VIII.

II. Overview of ADS-B

Using ADS-B, aircraft periodically broadcast their position and other information to the air traffic control (ATC) and to other aircraft in the vicinity. There are two types of ADS-B: 1090ES and UAT978. The 1090ES operates a 1090MHz radio signal to broadcast information worldwide via a Mode S transponder, whereas UAT978 operates at the 978MHz frequency for GA aircraft flying below 18,000 feet in the United States.

ADS-B 1090 is often called a "1090 Extended Squitter (1090ES)". A "squitter" refers to a periodic burst of aircraft-tracking data by a Mode S transponder without interrogation from the controller's radar. There are two types of squitters: short and extended. Since the 1090MHz extended squitter covers all the crucial data, 1090ES is a popular terminology.

The ADS-B functionality is divided into two parts: ADS-B IN and ADS-B OUT. ADS-B IN refers to the receiving, processing, and displaying ADS-B signals from the ATC, aircraft, and other ADS-B OUT-equipped vehicles. ADS-B OUT refers to the transmission of aircraft's Global Navigation Satellite System (GNSS) position, identity, velocity, and other information. Both functions are fully automatic processes that do not require traditional interrogations. Figure 1 shows a simplified view of the ADS-B communication system.

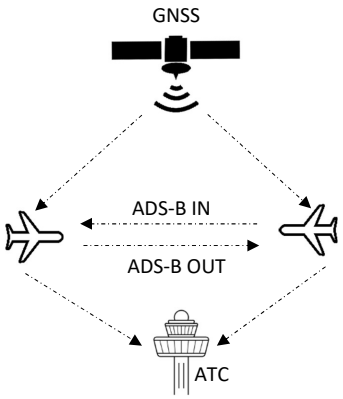


Fig. 1: ADS-B communication concept.

ADS-B is much less expensive to deploy. For example, SSR installation costs around $30 million, whereas ADS-B ground stations have a cost of approximately $4 million. In addition, ADS-B enhances safety by increasing situational awareness, makes the search-and-rescue (SAR) operations more efficient, simplifies the tasks of ATC, optimizes instrument flight rules, and allows for an increase in flight volume.

However, ADS-B also has some downsides. Its main problem is that it does not use message encryption nor authentication. It is a clear-text unauthenticated broadcast protocol, and its details are readily available [16]. Figure 2 shows the message structure of the two types of ADS-B signals. The ADS-B 1090ES signal is modulated using pulse position modulation, which is 112 bits long. A 0.8μs preamble leads the data block. The UAT978 signal is modulated using continuous phase frequency shift keying modulation with a modulation index of 0.6 and a data rate of 1.041667 Mbps. There are two types of UAT978 messages: basic messages and long messages. A basic message contains 144 bits and a long message has 272 bits. For message error correction, UAT789 uses Forward Error Correction (FEC) with Reed-Solomon error correction code. FEC length is 96 bits for short messages and 112 bits for long messages.

III. Related Works

The first ADS-B injection and spoofing attacks were publicly demonstrated in 2012 at the BlackHat USA conference by

	Downlink format	Transponder capability	ICAO address	Data	CRC Parity
Bit	5	3	24	56	24

(a) ADS-B 1090ES

	Synchronization	Payload	FEC Parity
Bit	36	144/272	96/112

(b) ADS-B UAT978

Fig. 2: ADS-B message structures.

Costin and Francillon [12]. They used MATLAB to encode and modulate the ADS-B data and they used Universal Software Radio Peripheral (USRP) SDR as the attacker and Plane Gadget Radar (PGR) as the victim. The spoofed aircraft was visible in Virtual Radar. They warned that low-cost hardware and moderate software effort could pose a threat to a multi-million dollar technology due to its lack of proper security measures. Strohmeier et al. [17] thoroughly analyzed the 1090Mhz communication channel through OpenSky sensor network in central Europe, which in 2014, was seen as capable of capturing about 30% of the European commercial air traffic. They found that ADS-B is highly susceptible to RF attacks, which may impact affected aircraft's collision avoidance and separation abilities. They also reported a high number of message losses caused by growing traffic on the 1090MHz channel. They recommended proper addressing of the ADS-B security issues before its full-scale deployment. A later study [18] suggested fingerprinting, random frequency hopping, public-key cryptography, retroactive key publication, etc., as the secure means of ADS-B broadcast. Schäfer et al. [19] mentioned that attacks on ADS-B can be inexpensive and highly successful. They transmitted fake signals using USRP and tested the reception via the SBS-3 ADS-B receiver. They performed several attacks, e.g., ghost aircraft flooding, ground station flooding, ghost aircraft injection, and virtual trajectory modification. They concluded that critical air traffic management decision processes should not rely on ADS-B-derived data without appropriate countermeasures. Manesh et al. [20] investigated the impacts of ADS-B message injection attacks. In their simulation experiment, they tested the Piccolo autopilot's response to ghost aircraft injection. The sudden appearance of a ghost aircraft close to the autopilot's position triggered a quick descent and a steep turn to gain safety clearance. According to the authors, this type of attack on ADS-B can distract pilots and ground controllers, cause air traffic disturbance, and increase the risk of aircraft collisions. Eskilsson et al. [21] demonstrated an ADS-B attack setup that cost only around $300. They used Python programming language to encode the ADS-B data, HackRF to transmit the signal, and dump1090 with a RTL-SDR transceiver to receive the signal. They warned that the availability of inexpensive attack equipment might encourage many adversaries to carry out attacks.

Portable ADS-B transceivers (e.g., SkyEcho2, Sentry, and echoUAT) connected to smartphones are popular, especially in the GA sector. However, according to Lundberg et al. [22], [23], these mobile setups are not part of the aircraft on-board systems. Thus, they do not meet and are not required to meet the reliability standards applied to traditional avionics. The authors conducted four tests: on the receiver-to-mobile application channel integrity, application-to-receiver channel integrity, EFB data integrity, and receiver integrity. The test results showed that all out of three mobile setups used were vulnerable as they allowed an attacker to manipulate information presented to the pilot. The authors recommended regular software and firmware updates, security-aware software development, and secure data exchange from the device to the application and vice versa. Sjödin and Gruneau [24] demonstrated a new type of attack called "teleporting ghost aircraft". Using HackRF and Sentry, they transmitted reports of an aircraft's position at different altitudes and moving around in an erratic pattern. Thus, the aircraft seems to have been breaking the laws of physics in terms of movement. They further reported that the receiver trust the protocol without any validation. They warned that if the insecure ADS-B gets more deeply integrated into aircraft, it will likely gain more access to internal flight and control systems. If the TCAS relies on ADS-B, an attacker would be able to steer the plane like a puppet. Leonardo et al. [25] developed realistic jamming threat models and analyzed the impact of jamming on crowd-sourced air traffic surveillance. They showed that a high-power jammer could significantly disrupt ADS-B communication and that a ground-based attack is more dangerous than an air-based attack because it can be implemented with very cheap equipment. They proposed two jamming mitigation approaches: network-based mitigation and sensor-based mitigation. For network-based mitigation, they proposed increasing or modifying the distribution of sensors so that the available redundancy can mitigate some of the jamming effects. For sensor-based mitigation, they suggested multi-channel signal processing or multi-channel receiver using sector antennas. Leonardi et al. [26] demonstrated a USRP SDR-based low-cost jammer that could jam ADS-B signal up to approximately 218 km away using an amplifier. They used ICAO standard preambles but random binary data to generate the jammer waveforms. The jamming signal created an interference in the ADS-B channel. As a result, the real signal was distorted fully or partially. At the receiving end, the cyclic redundancy check (CRC) of the signal did not match the payload, so the receiver dropped it assuming possible corruption. Separation of overlapping real and jamming signals was mentioned as a solution, and that it could be done in between the preamble detection and the pulse extraction. However, no further details on the solution were provided. Pearce et al. [27] also tested the impact of an interference attack on the ADS-B signal. They used USRP to produce a fake signal. Constructive interference was formulated by transmitting two signals, a phase interference and a destructive interference, in a 180-degree phase. They found that the destructive interference caused the highest bit error rate of 32.39%. They concluded that due to the insecure nature of ADS-B, even low-technology could exploit it.

IV. Attacks on ADS-B

Several studies had defined various types of attacks on the ADS-B system [12], [18], [19], [21], [28], [29], [30], [31], [32]. The attacks varied according to their goal, setup, and method. In addition to the attacks cited in literature, we propose some new attack ideas. They are briefly summarized in Table I and further discussed below.

A. Aircraft reconnaissance

Anyone can listen to unencrypted ADS-B broadcast using cheap SDR dongles that cost as low as $15. Web-based flight tracking services (e.g., https://flightware.com, https://

TABLE I: Summary of types of ADS-B attacks in literature.

Attack	Method	Implemented in this study	Closest related work
Aircraft reconnaissance	Eavesdropping	✓	[28], [29]
Spoofing or ghost aircraft	Message injection	✓	[12], [32], [21]
Flooding	High-level signal jamming	✓	[19], [30]
ADS-B packets DoS**	High-level signal jamming	✓	[19], [33]
Aircraft disappearance	Message deletion	✓	[19]
Trajectory modification	Message modification	✓	[19]
Logically-invalid data encoding	Fake signal	✓	[24]
False distress signal**	Squawk code modification	✓	[19]
Jamming	Low-level signal jamming	✓	[19], [25]
Specific data-link protocol fuzzing*	Fuzzing*	✓	[34], [35]
Coordinated attack*	Multiple emitter*	✓	[None]
Specific error handling test*	Message modification	✓	[None]

* Our novel idea in the ADS-B context (to the best of our knowledge)
** Our practical demonstration based on existing theoretical idea(s)

flightradar24.com, and many others) gather flight data worldwide through eavesdropping. These services publish these real-time data on their websites. At times, they may hamper the privacy of prominent public figures and ordinary individuals.

B. Spoofing

As ADS-B transmissions are unauthenticated, spoofing can be done with a fake signal that matches the required messaging protocol. A spoofed signal may result in appearance of ghost aircraft on the ATC's screens or on the screens of other airplanes in the vicinity. This could cause significant disruption if real aircraft need to perform evasive maneuvers to avoid the ghost aircraft. In addition, the sudden presence of foreign military aircraft may trigger military action.

C. Flooding

Flooding is an attack where an attacker floods the screen of the ATC or of an aircraft with fake planes. The attack does not require high-end equipment, as ADS-B receivers are designed to detect very weak signals (e.g., around -80 dBm) [36]. Even a very basic flooding attack could potentially disrupt regular air traffic monitoring; and a higher-impact flooding attack can be achieved with transmission-enabled SDR coupled with power amplifiers.

D. False distress signal

ADS-B provides mechanisms for supporting surveillance replies, e.g., Mode A and Mode C. As such, downlink format 5 (DF5) is assigned to the 13-bit identity code that encodes the four octal digits called the "squawk code" assigned by the ATC to the aircraft. Some squawk codes are used only to indicate emergencies or unlawful interference such as aircraft hijacking and radio failure. A distress squawk code can be assigned to a plane without ATC's permission. Squawk codes such as 7500 (aircraft hijacking), 7600 (radio failure or lost communication), and 7700 (emergency) would set off an alarm in the ADS-B network and nearby ATC towers [37]. An attacker may try to alter the squawk code of the targeted aircraft, which would cause a severe disturbance in both the flight and ground operations. Though this operational attack concept was introduced in [19], [12], to the best of our knowledge, we are the first to implement and study it in an extensive experimental setup within the ADS-B context.

E. Coordinated attackers

Among the data fields in an ADS-B message, the ICAO24 code is used as the reference by the receiving software. In subsequent messages, the ADS-B information is displayed and updated against such ICAO24 code. Our pentesting platform allowed us the flexibility to use any ICAO24 code. Thus, multiple attackers could coordinate among themselves, or a single attacker with multiple emitters could coordinate its attacks. During the attack, the "coordinated attackers" used the same ICAO24 code to send multiple signals that contain the same reference (ICAO24 code) but differing values in some of the other ADS-B data fields. We call this type of attack a "coordinated attack" in the sense that the same (or multiple) attacker(s) coordinate to target the same ICAO24 perceived by the same ADS-B receiver. In contrast, non-coordinated attackers may target different ICAO24 codes as perceived by different (or same) ADS-B receivers (e.g., ATC towers or in-flight aircraft) even though there is a minor statistical probability that two distinct attackers may end up targeting the same ICAO24 code perceived by the same ADS-B receiver even though those attackers are *not coordinated*. Since the reference point is the same, the data fields will be updated according to the encoded message of multiple signals in the receiving software. However, some fields in the ADS-B should not be updated throughout the flight, or the updates should follow a standard or common pattern. For example, the flight number should not be updated within a single flight, and position coordinate should be updated smoothly with a clear direction, possibly with a historical fading-out path. However, in a "coordinated attack", the attackers, using multiple emitters, can change the flight number every second or can change the position of the aircraft from one city to another in an instant. This can lead to ATC confusion, and can have many dangerous consequences. In practice, a coordinated attack can also be achieved even with a single attack emitter, since the second emitter can be that of the legitimate aircraft itself, hence the single attacker merely has to coordinate with the aircraft using the same ICAO24 code and will be able to achieve the same effect similar to two "coordinated attackers". To the best of our knowledge, we are the first to propose, implement, and study this type of attack within the ADS-B context.

F. Attacks on ADS-B CRC error handling

Noise in the RF channel can distort an ADS-B signal completely or partially. The CRC allows the receiver to validate the correctness of the transmitted information. Depending on the receiver capability, ADS-B 1090ES supports up to 5-bit error correction using a 24-degree fixed generator polynomial [38]. In order to test the CRC and error-handling capabilities of the software, we deliberately and randomly flipped some message bits in the ADS-B, and then transmitted them to the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAES.2021.3139559, IEEE Transactions on Aerospace and Electronic Systems

target software. Flipping bits is not an effective attack per-se, however it is an interesting test of the integrity of the target software. To the best of our knowledge, we are the first to propose, implement, and study this test within the ADS-B context.

G. DoS attacks on the ADS-B protocol

DoS attacks target the disruption of the availability of services by clogging or shutting down service entities or networks. For example, an attacker may send a massive amount of fake signals to a targeted aircraft or ground station, which may exceed the ADS-B IN capacity. In this situation, an ADS-B system may exhibit abnormal behavior such as not responding, freezing, delivering wrong information. Since the ADS-B service works based on unencrypted radio communication, it cannot block a malicious transmission source. Schäfer et al. [19] implemented an RF-spectrum (low-level) DoS attack by emitting white noise (i.e., DoS not at the ADS-B packet level), which can be more generically categorized as jamming (see Section IV-J). However, to the best of our knowledge, we are the first to propose, implement, and study DoS attacks at the ADS-B packet level that resulted in software crashes in some worst-case scenarios, and can further lead to Remote Code Execution (RCE) exploits.

H. Fuzzing avionics protocols

Fuzzing is a software testing method that finds bugs in implementation, input sanitization, and logic using intentionally malformed inputs and corner-case scenarios. Many ADS-B devices use the Garmin Data Link 90 (GDL-90) protocol to display data on mobile applications. These types of ADS-B transceivers open a WiFi access point which either lacks a password at all or has a weak default one. The mobile device connects to transponder's WiFi network, and receives the ADS-B data through that WiFi channel. Connection to the insecure WiFi network of the ADS-B transceiver may expose the ADS-B transceiver to the risk of fuzzing, which could lead to software crashes in some worst-case scenarios. To the best of our knowledge, we are the first to propose, implement, and study this type of attack within the GDL-90/ADS-B contexts.

I. Logically-invalid data encoding

The allocated bit number defines the maximum and minimum value of each data field in the ADS-B message. For example, the altitude value ranges from -1,000 feet to 50,175 feet, and the velocity value ranges from 0 knots to 1,024 knots. However, since the encoded ADS-B messages are not validated, a technically correct but logically invalid message could be formulated, e.g., regarding the maximum possible velocity of an aircraft at the minimum possible altitude or vice versa. As it is assumed that the onboard system would provide the correct data to the ADS-B system, which will be subsequently transmitted, however the correlation among the data fields is not checked. Hence, an attacker can use this type of discrepancy to launch an attack or to puzzle the ATC.

J. Jamming

Jamming the communication channel to disrupt or suspend service has been a common tactic since World War II. In this type of attack, an adversary introduces a powerful RF signal to overwhelm the system's spectrum, thus denying service to all wireless nodes within the range of the interference. Several types of ADS-B attacks can be made based on this technique such as the following:

1) Signal jamming: This is a very basic type of attack that has been demonstrated by several researchers in different fields of RF communication. An attacker may block the two ADS-B traffic channels (1090MHz 1090ES and 978MHz UAT978) using a high-power RF noise transmission. A jamming attack near a busy airport may limit or stop flight operations. However, important radio spectra on important areas are continuously monitored by the regulatory bodies, therefore detecting and countering the jamming would be easier when the attack is performed in an urban area. In remote areas, where such monitoring is limited or missing, detecting and countering jamming would be more challenging.

2) Aircraft disappearance: This type of attack creates a destructive interference, or alternatively, blocks the targeted aircraft's signal. As a result, the aircraft can disappear from the receiver's screens. This is intelligent jamming as its technical complexity is several magnitudes higher that of basic signal jamming. There are two possible ways to carry out this attack. First, the attacker can generate a timely synchronized inverse of the ADS-B signal and transmit it over the air. The real aircraft's signal and the attacker's inverse signal fully or partially diminish the real ADS-B message. As a result, the ADS-B messages are distorted and thus are dropped by the receivers. However, precise timing synchronization is difficult to achieve and thus, less efficient to perform. Another strategy is to block the signal. However, selective blocking is arduous. Therefore, the attacker could jam the ADS-B channel to prevent a receiver from receiving any legitimate signal. Then the attacker can collect the real signals through another receiver and selectively replay or transmit those signals in a high-power mode, except the targeted aircraft's signal. Thus, the targeted aircraft disappears from the targeted receiver. Investigation of time-synchronized selective aircraft disappearance is left for future work, to demonstrate that the attacker can learn the time pattern of ADS-B broadcasts of its targeted aircraft and thus, can beam highly directional synchronized noise during the exact time slot of the aircraft's ADS-B broadcast to degrade the aircraft's ADS-B messages (i.e., erasing them from the displays of other traffic participants, including the ATC) while leaving intact the ADS-B messages of the other participating emitters.

3) Trajectory modification: This attack can be performed through message modification. For example, an attacker can send a high-power signal to suppress an actual low-power signal. Thus, the attacker replaces a part or all of the target message. However, the need to calculate a new CRC code can make this approach harder. Nonetheless, the attack can also be as an aircraft disappearance attack (see subsection IV-J2), but instead of hiding the targeted aircraft's signal, the attacker

transmits the location of the new trajectory. The actual transmitter or receiver may not be aware of the change in the arbitrary data. Therefore, the attack may remain undetected. As a result, the ATC may give wrong instructions or the TCAS may have an unnecessary reaction.

V. Experimental setup

In this study, we used a total of 36 ADS-B IN configurations (hardware + software). We developed an avionics pentesting platform that uses the Python programming language to control ADS-B messages and protocols. Our platform also uses the GNU Radio Companion (GRC) software to build the signal processing blocks that take the attacking payload's byte order as input and generates the "in-phase" and "quadrature" (I/Q) of the signal. The IQs would subsequently be transmitted over the air using a variety of supported SDRs (e.g., HackRF, BladeRF, Pluto SDR) that can be connected to the platform. Figure 3 shows our experimental setup. More details on the hardware and software can be found in Table II and Table III, respectively. In general, ADS-B 1090ES is much more widely used and adopted than UAT978. Therefore, our strongest focus was on ADS-B 1090ES, whereas our focus on UAT978 was scenario-dependent. All the attacking scenarios listed in Section IV were tested for ADS-B 1090ES. For UAT978, we limited the tests to aircraft reconnaissance, spoofing, flooding, DoS, jamming, and protocol fuzzing attacks. We leave the testing of the rest of the attacking scenarios for UAT978 for future work.

TABLE II: List of hardware used in the experiments.

Device Name	ADS-B IN (RX)	ADS-B OUT (TX)	Attacker Mode
uAvionix SkyEcho2	1090ES UAT978	1090ES	×
uAvionix echoUAT	1090ES UAT978	UAT978	×
ForeFlight Sentry	1090ES UAT978	No	×
Garmin GDL 52	1090ES UAT978	No	×
Plane Gadget Radar (PGR)	1090ES	No	×
Aerobits TR-1W	1090ES	1090ES	×
Aerobits EVAL-TT-SF1	1090ES	No	×
PX4 (Aerobits AERO chip)	1090ES	No	×
Cube Orange	1090ES	No	×
RTL SDR	1090ES UAT978	No	×
HackRF	Programmable	Programmable	✓
BladeRF	Programmable	Programmable	✓
Pluto SDR	Programmable	Programmable	✓

A. Attacking hardware and devices

We used HackRF, Pluto SDR, and BladeRF as the attacking devices. As a part of signal processing, we used the freely available GRC software. In additions, we use the Python programming language to create the attack payload. GRC produces and supplies the IQ of the signal to the transmission-enabled SDR according to the payload. Thus, the RF signal of the ADS-B message was created.

TABLE III: List of software tested.

Software	ADS-B type	Platform
Dump1090-fa v 4.0	1090ES	Linux
Dump1090 v 1.09.0608.14	1090ES	Windows
Dump1090 v 1.15-dev	1090ES	Windows
Dump978-fa v 4.0	UAT978	Linux
PlanePlotter v 6.5.1.1	1090ES	Windows
RTL1090 v 0.9.0.100	1090ES	Windows
RTL1090 v 2.11.3.103	1090ES	Windows
Micro ADS-B v 1.15.1	1090ES	Windows
Mission Planner v 1.3.74	1090ES	Windows
QGround Control 4.1.2	1090ES	Windows
ForeFlight EFB v 13.0.1	1090ES, UAT978	iOS
Stratus Insight EFB v 5.17.3	1090ES, UAT978	iOS
Airmate EFB v 2.3	1090ES, UAT978	iOS
FlyQ EFB v 5.0	1090ES, UAT978	iOS
AvPlan EFB v 7.10.7	1090ES, UAT978	iOS
AvPlan EFB v 1.3.14	1090ES, UAT978	Android
EasyVFR4 EFB v 5.0.866	1090ES, UAT978	iOS
EasyVFR4 EFB v 4.0.870	1090ES, UAT978	Android
OZRunways EFB v 10.10	1090ES, UAT978	iOS
OZRunways EFB v 4.4.1	1090ES, UAT978	Android
Garmin Pilot EFB v 10.5.7	1090ES, UAT978	iOS
Garmin Pilot EFB v 8.0.0	1090ES, UAT978	Android

B. Receiving hardware and devices

In this study, we tested a total of 12 different receiving devices. EFB apps hosted in mobile devices (iOS and Android) accessed data from SkyEcho2, echoUAT, and Sentry through a WiFi connection. Data from a GDL 52 device was accessed via a Bluetooth connection. All the other tested software were run on a laptop, and the data was accessed using a USB connection. As the attacking SDRs also had receiving capability, they could also be used as receivers. When RTL SDR, HackRF, and BladeRF were used as receiving devices (i.e., only for the IQ RF frontend) connected to the dump1090 and dump978 variant, they had identical results because the software did all the heavy-processing in the form of demodulation and decoding. As the transceiver hardware did not affect the results in this case, we omitted the hardware transceiver hardware column from the result tables. However, the different receivers or transceivers had different functionalities. For example, some of them supported only 1090ES; some others, only UAT978; and the rest supported both. Therefore, all the devices and their functionalities are presented in Table II.

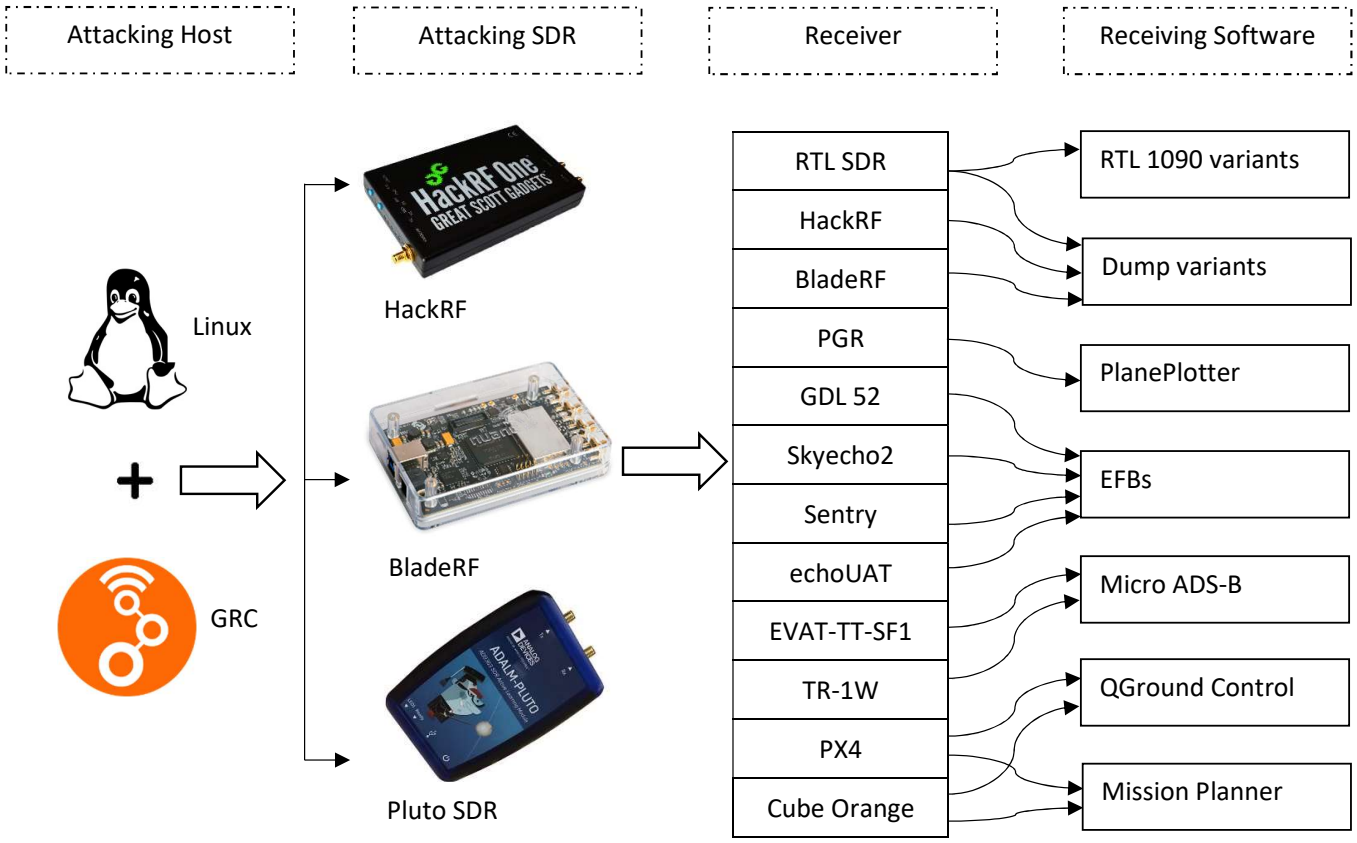


Fig. 3: Experimental attacking setup.

C. Receiving software

There is a wide range of ADS-B receiving software that support various devices. Those that were used in this study are listed in Table III. Different users are likely to use different hardware and software combinations as their preferred ADS-B solution. However, the software may differ in functionality, logic, error-handling capacity, and other behavior. To be able to stage a possible real-life scenario, we tested 22 different ADS-B software. Our software testbed included many desktop-based applications (i.e., those that target ATC or air traffic management (ATM) deployment), mobile-based EFB (i.e., those that target GA and personal users), and various specialized hardware devices (i.e., those that target commercial and military aviation using specialized hardware setups).

VI. Results and Evaluation

During the experiment, we tested 36 different ADS-B IN combinations (of hardware, software, and host). In this section, we describe our findings on the attack scenarios listed in Table I. Some sensitive information in Figure 7, and Figure 9 was blurred, and a real flight number in Figure 8 was replaced with a dummy number.

A. Aircraft reconnaissance

We confirmed what had been stated in many previous studies – that aircraft reconnaissance through eavesdropping is an effortless task. Each of our ADS-B receivers did receive the 1090ES signal from the flying aircraft. We did not receive any UAT978 signal, as this signal is not used in Europe. However, using our platform we were able to produce a UAT978 signal that commercial mobile cockpit devices, e.g., Sentry, SkyEcho2, and echoUAT, properly received and displayed. Eavesdropping sometimes violates privacy, and sharing the eavesdropped data on the internet aggravates the privacy concern. Since the ADS-B signal is not encrypted, there is no way to stop eavesdropping. For example, Figure 4 shows Joe Biden's flight from Wilmington, Delaware to Washington, D.C. for his presidential inauguration.

B. Spoofing

We were able to spoof both ADS-B signals 1090ES and UAT978 signals. All the ADS-B receivers decoded (according to their supported type) our fake signal *without any alert*. Researchers had suggested identifying the fake signal using the Doppler shift, multilateration, and many other machine learning methods [39], [40]. However, none of the tested ADS-B combinations showed any alert. Though spoofing is the simplest and earliest type of attacks of ADS-B, it may still pose a significant threat to the safety and resiliency of ATC. For example, Figure 5 shows a spoofed aircraft as if over the North Korean sky.

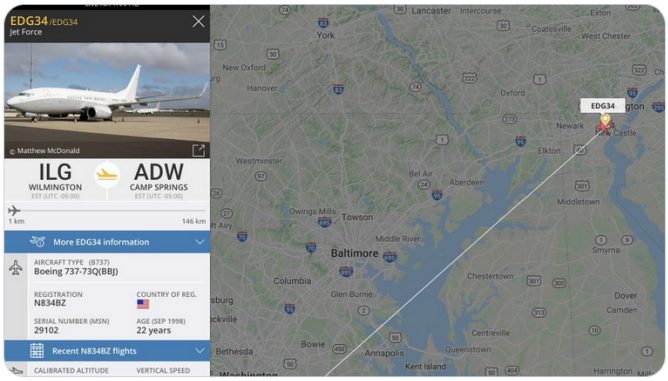


Fig. 4: President Joe Biden's flight for his presidential inauguration (image courtesy: https://twitter.com/flightradar24/status/1351628618187862026).

Fig. 5: Spoofed aircraft over the North Korean sky.

C. Flooding

We flooded the ADS-B receivers with the fake signal. Flooding attacks make it impossible to distinguish fake aircraft from real ones. Similar to spoofing, we observed no alert during our flooding attack in any of the ADS-B combinations. Our general observation is that the flooding attacks had more sensible impact on constrained mobile setups (e.g., the memory, computational power, and screen size) than on their desktop setup counterparts. For example, Figure 6 shows a flooded screen during our test, which indicates that an attacker can literally flood the entire world map.

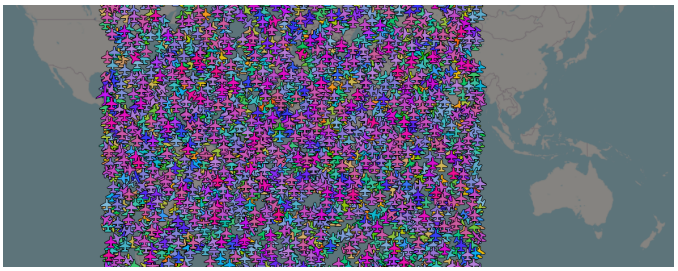


Fig. 6: Flooded screen when using tar1090 software.

D. False distress signal

When a distress signal is transmitted, all the ATCs in the area are immediately alerted that the aircraft has an emergency. We developed a Python script for encoding a false distress squawk code in an ADS-B signal. During our test, the transmission of our fake distress squawk was decoded by all the (supported) receivers. A fake distress squawk code may have severe consequences. For example, it may initiate a false alarm that could lead to air force deployment. In our test, as a part of the visual alert that an aircraft should receive after a distress message, the dump1090-fa software made the aircraft's icon turn red and displayed an alert text with a red background, as shown in Figure 7. However, the alert function was implementation-dependent, and thus, was not available in most of the tested ADS-B configurations (see Table IV). Manuals of commercial aviation ADS-B devices suggest that such devices, when mounted in cabins of for commercial aircraft, be equipped with bring audio-visual alerts for unexpected situations, as described above. However, we were unable to verify due to our lack of access to such devices.

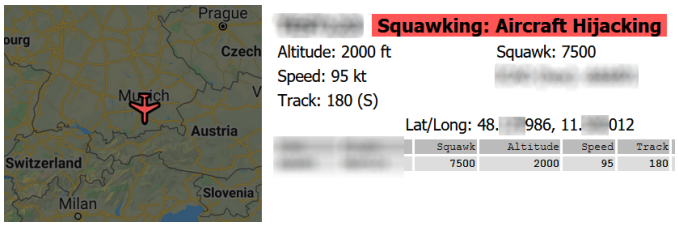


Fig. 7: Fake distress squawk code in the Dump1090 net.

E. Coordinated attackers

All the ADS-B software we are aware of use the ICAO24 code of an ADS-B message as the reference for storing and processing the other data in that message. Hence, to implement the "coordinated attackers" scenario, we used the same ICAO24 code but encoded different ADS-B information into two separate signals and then sent the signals via two separate SDRs towards the tested configuration at the receiving end. We observed that this type of attack created logical vulnerabilities in the receiving software. Instead of a smooth position change the receiver decoded a scattered aircraft position with incoherent coordinates. As a result, the ATC may become confused as to which is the actual location of that aircraft. We summarize the results of the coordinated attacker scenario for all the ADS-B message fields in Table IV, where we use N=2 as the number of coordinated attackers. However, our platform allowed us to perform the attack with N>2, and the only limiting factor was the number of SDRs available and dedicated to the "attacker role." Table IV show disturbing inconsistencies and discrepancies in on how different ADS-B configurations deal with such unexpected scenarios. Even more troubling, in our opinion, is that none of the tested configurations displayed any alert on such inconsistencies during the signal decoding and display stages.

For example, in Table IV we can see some interesting scenarios. First, if the air traffic management team has three ATC locations and each location is combined with one of three different software – for example, dump1090 v 1.09.0608.14, dump1090 v 1.15-dev, and RTL1090 v 0.9.0.10 – then each ATC could see a completely different operational picture due to the coordinated attack because the longitude field in the

TABLE IV: Summary of the effects of multiple coordinated attackers on ADS-B 1090ES.

Configurations		**Effects**							
Hardware	Software	ICAO24	Squawk	Flight	Velocity	Altitude	Latitude	Longitude	Latitude and Longitude
RTL SDR	Dump1090-fa v 4.0	CDA	FLC	FLC	FLC	FLC	**WRG**	**WRG**	**WRG**
	Dump1090 v 1.09.0608.14	CDA	FLC	FLC	FLC	FLC	**WRG**	**WRG**	**WRG**
	Dump1090 v 1.15-dev	CDA	FLC	FLC	FLC	FLC	*FST*	*FST*	*FST*
	Dump978-fa v 4.0	(DNT)	(DNT)	(DNT)	(DNT)	(DNT)	(DNT)	(DNT)	(DNT)
	RTL1090 v 0.9.0.100	CDA	FLC	FLC	FLC	FLC	*FST*	FLC	*FST*
	RTL1090 v 2.11.3.103	CDA	FLC	FLC	FLC	FLC	*FST*	FLC	*FST*
PGR	PlanePlotter v 6.5.1.1	CDA	FLC	FLC	FLC	FLC	FLC	FLC	FLC
EVAL-TT-SF1	Micro ADS-B v 1.15.1	CDA	INA	FLC	FLC	FLC	**WRG**	**WRG**	**WRG**
TR-1W	Micro ADS-B v 1.15.1	CDA	INA	FLC	FLC	FLC	**WRG**	**WRG**	**WRG**
PX4	Mission Planner v 1.3.74	CDA	*FST*	FLC	FLC	**WRG**	FLC	FLC	FLC
	QGround Control v 4.1.2	CDA	INA	INA	INA	FLC	FLC	FLC	FLC
Cube Orange	Mission Planner v 1.3.74	CDA	*FST*	FLC	FLC	**WRG**	FLC	FLC	FLC
	QGround Control v 4.1.2	CDA	INA	INA	INA	FLC	FLC	FLC	FLC
GDL 52	Garmin Pilot v 10.5.7	CDA	INA	$\overline{DSP}$	$\overline{DSP}$	FLC	FLC	FLC	FLC
	Garmin Pilot v 8.0.0 *	CDA	INA	$\overline{DSP}$	$\overline{DSP}$	FLC	FLC	FLC	FLC
Sentry	ForeFlight	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
SkyEcho2	Airmate EFB v 2.3	CDA	INA	FLC	INA	FLC	FLC	FLC	FLC
	AvPlan EFB 7.10.7	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	AvPlan EFB 1.3.14 *	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	EasyVFR4 EFB v 4.0.866	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	EasyVFR4 EFB v 4.0.870 *	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	FlyQ EFB v 5.0	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	ForeFlight EFB v 13.0.1	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	Stratus Insight EFB v 5.17.3	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	OZRunways EFB v 10.10	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	OZRunways EFB v 4.4.1 *	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
echoUAT	Airmate EFB v 2.3	CDA	INA	FLC	INA	FLC	FLC	FLC	FLC
	AvPlan EFB 7.10.7	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	AvPlan EFB 1.3.14 *	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	EasyVFR4 EFB v 4.0.866	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	EasyVFR4 EFB v 4.0.870 *	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	FlyQ EFB v 5.0	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	ForeFlight EFB v 13.0.1	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	Stratus Insight EFB v 5.17.3	CDA	INA	FLC	FLC	FLC	FLC	FLC	FLC
	OZRunways EFB v 10.10	CDA	INA	FLC	INA	FLC	FLC	FLC	FLC
	OZRunways EFB v 4.4.1 *	CDA	INA	FLC	INA	FLC	FLC	FLC	FLC

Note: * = Android version (other EFBs are iOS version); CDA = Count as Different Aircraft; FLC = Fluctuates (i.e., displays alternate values from different attackers); **WRG** = Wrong value(s) altogether; INA = Information Not Available in the application; *FST* = Retain the First received signal's information; DNT = Did Not Test; $\overline{DSP}$ = Disappear.

three software variants behaves differently. The first software (dump1090 v 1.09.0608.14) shows a completely wrong value (encoded as **WRG** in Table IV); the second software (dump1090 v 1.15-dev) retains the first value it received (encoded as *FST* in Table IV); and in the third software (RTL1090 v 0.9.0.10) the value fluctuates (encoded as "FLC" in Table IV) according to the value from each of the attackers. Such effects of a coordinated attack can have high-impact negative effects for ATMs from the operation, coordination, and safety points of view.

Second, let us consider a coordinated attack that targets the same flight number. Under normal circumstances, there should be only one unique flight number for an aircraft in a single time frame in a certain airspace. We can also assign the same flight number to multiple aircraft within our coordinated attack setup, which can confuse the ATC. For example, Figure 8 shows three aircraft with the same flight number "ABCDEFG." In addition, by using multiple SDRs, we can assign multiple flight numbers to the same ICAO24 code, which can further confuse and make uncertain both the human operator and the ATM software.

Third, a good case to discuss is the effect on *Mission Planner v 1.3.74* when it was exposed to a coordinated attack. *Mission Planner v 1.3.74* is a widely used software for flight and mission control of drones and unmanned aerial vehicles (UAVs). Such flight and mission control software can also receive ADS-B IN data via compatible hardware such as Cube Orange with an integrated ADS-B receiver and PX4 with a Universal Asynchronous Receiver Transmitter (UART)-based ADS-B IN sensor. This is a very useful functionality for avoiding any dangerously close paths or potential mid-air collisions. However, the *Mission Planner v 1.3.74* software wrongly computes the altitude (see the value marked **WRG** in Table IV) of the surrounding ADS-B OUT systems when a coordinated attack is performed. This means that the software automatically instructs the drone(s) under it's control to take a flight path or a decision that can lead to the drone's unsafe operation such as to a mid-air or ground collision, due to the incorrect altitude estimation. It is important to note that the effect of the coordinated attack can also be achieved completely unintentionally if two legitimate ADS-B transmitters set by mistake the same ICAO24 code within the ADS-B receiving range of the *Mission Planner v 1.3.74*. Although it is unlikely that such unintentional situations may occur in real life, it is still a possible scenario and cannot be excluded from a risk assessment unless the software is fixed and retested with our suggested methodology.

Finally, in Garmin GDL 52 coupled with the Garmin Pilot application, we observed an interesting information disappearance effect (encoded as $\overline{DSP}$ in Table IV). Two attackers transmitted ADS-B signals that contained the same ICAO24 code but differing values for other fields (e.g., flight number, velocity, and position information). After about 2–3 minutes of the test, the flight number and velocity of the aircraft disappeared from the main application's screen, and the position information fluctuated similar to other EFBs. However the fluctuations started to be more random after the aircraft disappeared from the main application's screen, an effect not observed in other EFBs.

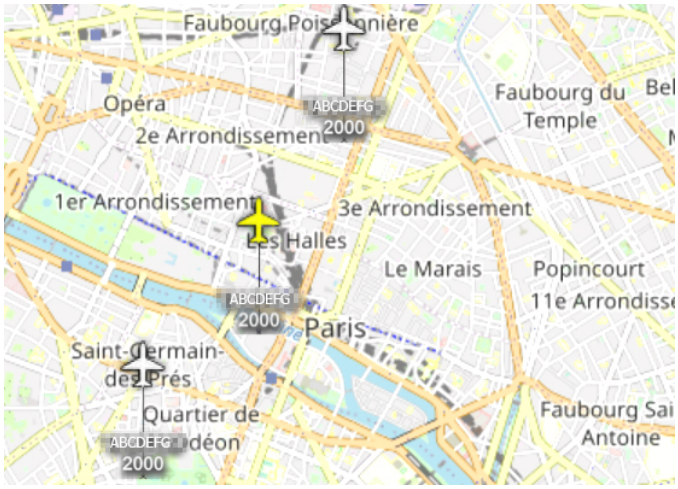


Fig. 8: Virtual Radar displaying the same flight number for multiple aircraft with distinct ICAO24 codes.

F. Attacks on ADS-B CRC error handling

Depending on receiver capability, ADS-B 1090ES supports up to 5-bit error detection and correction [38]. Due to the interference, if the bit error (against the CRC) exceeds a threshold, the receiver assumes that the message is corrupted and drops that message. To test how much error a system can handle in practice, we designed tests where we randomly flipped (i.e., simulated a random error) several bits (up to 3 bits) to test the system's error-handling capacity. Table V summarizes the results of the bit-flip tests. The results show that most setups generally support ADS-B 1090ES CRC error correction only up to 2 error bits.

We observed that all the ADS-B configurations took extra time to decode the message during the error correction. This is due to the processing-intensive nature of error detection and correction, which could lead to resource-allocation variations of DoS attacks, e.g., to possible degradation of the ADS-B decoding/display performance to below the ADS-B minimum operational performance standards (such as Radio Technical Commission for Aeronautics RTCA-260 and RTCA-282) and the ADS-B minimum aviation system performance standards (such as RTCA-242). We also observed that a significant number of messages were dropped, and the percentage of the dropped messages varied during the test. In line with this observation, Leonardo et al. [25] also observed high message drop percentages (50% to 90%) due to interferences.

In addition, we observed behavioral variations or inconsistencies based on the tested configuration. For example, the error-correcting routines themselves introduced other types of software logic errors such as the appearance of ghost aircraft in some of the dump1090 variants while the routine tried to correct the error (a situation that in itself is ironic). We believe this bug occurred most likely due to an *erroneous error-correction* at the implementation level, but we leave the further investigation of the open-source code to future work. While the correction should be deterministic in all cases (based

on the strict mathematical foundations of CRC), in our tests, the efficacy and soundness of the correction depended on the setup and implementation. Some of the setups detected and corrected the error(s) without any side effects, whereas some detected and corrected the error(s) but introduced other bugs. Therefore, inconsistent error correction cause differences in the screen display across large-area ATMs or highly heterogeneous setups in various ATCs. The results shown in Table V also indicate that even though software play a significant role in demodulating and decoding data, sometimes, the hardware also plays a role that must be considered. For example, the EFBs connected to the Sentry did not correct the message, but the same EFB apps (including ForeFlight Mobile) that were connected to echoUAT and SkyEcho2 (both from the same vendor) decoded the message. ADS-B devices in ATMs and ATCs may altogether use different chipsets, different hardware designs, or different firmware, which may cause differences in performance. Finally, the PX4 and Cube Orange entries in Table V, clearly show that even with dedicated hardware for ADS-B devices, the software plays an important role in determining the level, extent, and quality of the CRC error detection and correction. Last but not least, we leave the exploration of the UAT978 FEC error handling for future work.

TABLE V: Summary of the ADS-B CRC error-handling experiments on the 1090ES.

Configurations		Message error		
Hardware	Software	1-bit	2-bit	3-bit
RTL SDR	Dump1090-fa v 4.0	MSD+DE	MSD+DE	NDE
	Dump1090 v 1.09.0608.14	BUG+DE	MSD+BUG+DE	NDE
	Dump1090 v 1.15-dev	DE	MSD+BUG+DE	MSD+BUG+DE
	RTL1090 v 0.9.0.100	MSD+DE	MSD+DE	NDE
	RTL1090 v 2.11.3.103	MSD+DE	MSD+DE	NDE
	Dump978-fa v 4.0	(DNT)	(DNT)	(DNT)
PGR	PlanePlotter v 6.5.1.1	MSD+DE	NDE	NDE
EVAL-TT-SF1	Micro ADS-B v 1.15.1	MSD+DE	MSD+DE	NDE
TR-1W	Micro ADS-B v 1.15.1	MSD+DE	MSD+DE	NDE
PX4	Mission Planner v 1.3.74	MSD+DE	NDE	NDE
	QGround Control v 4.1.2	NDE	NDE	NDE
Cube Orange	Mission Planner v 1.3.74	MSD+DE	MSD+DE	NDE
	QGround Control v 4.1.2	MSD+DE	NDE	NDE
GDL 52	Garmin Pilot v 10.5.7	MSD+DE	MSD+DE	NDE
	Garmin Pilot v 8.0.0 *	MSD+DE	MSD+DE	NDE
Sentry	ForeFlight EFB v 13.0.1	NDE	NDE	NDE
SkyEcho2	Airmate EFB v 2.3	MSD+DE	MSD+DE	NDE
	AvPlan EFB 7.10.7	MSD+DE	MSD+DE	NDE
	AvPlan EFB 1.3.14 *	MSD+DE	MSD+DE	NDE
	EasyVFR4 EFB v 4.0.866	MSD+DE	MSD+DE	NDE
	EasyVFR4 EFB v 4.0.870 *	MSD+DE	MSD+DE	NDE
	FlyQ EFB v 5.0	MSD+DE	MSD+DE	NDE
	ForeFlight EFB v 13.0.1	MSD+DE	MSD+DE	NDE
	Stratus Insight EFB v 5.17.3	MSD+DE	MSD+DE	NDE
	OZRunways EFB v 10.10	MSD+DE	MSD+DE	NDE
	OZRunways EFB v 4.4.1 *	MSD+DE	MSD+DE	NDE
echoUAT	Airmate EFB v 2.3	MSD+DE	MSD+DE	NDE
	AvPlan EFB 7.10.7	MSD+DE	MSD+DE	NDE
	AvPlan EFB 1.3.14 *	MSD+DE	MSD+DE	NDE
	EasyVFR4 EFB v 4.0.866	MSD+DE	MSD+DE	NDE
	EasyVFR4 EFB v 4.0.870 *	MSD+DE	MSD+DE	NDE
	FlyQ EFB v 5.0	MSD+DE	MSD+DE	NDE
	ForeFlight EFB v 13.0.1	MSD+DE	MSD+DE	NDE
	Stratus Insight EFB v 5.17.3	MSD+DE	MSD+DE	NDE
	OZRunways EFB v 10.10	MSD+DE	MSD+DE	NDE
	OZRunways EFB v 4.4.1 *	MSD+DE	MSD+DE	NDE

Note: * = Android version (rest other EFBs are iOS version); MSD = Message(s) Dropped; DE = Decoded; NDE = Not Decoded; BUG = Bug or ghost aircraft(s) introduced; DNT = Did Not Test.

G. DoS attacks on ADS-B protocol level

Due to constraints in computing resources and software design choices, each application or software can decode only a limited number of ADS-B signals in a given time. When the limit is exceeded, a DoS attack can be performed. We burst a very high amount of valid yet fake ADS-B signals (30–100 thousand different ICAO24 codes) in a short amount of time (2–3 min) while trying to perform a successful high-level DoS attack on the target application or hardware device. Depending on the software and the hardware, different ADS-B combinations performed slightly differently when they were exposed to a DoS attack. The performed DoS attack exceeded the ADS-B IN processing capacity of most of the software/applications we tested. Some of them crashed, some of their output clogged, some setups produced garbage outputs that could not be read, and others significantly dropped messages (e.g., did not detect nor process nor show all the transmitted messages). If a setup did not support an ADS-B mode, we mentioned that setup as Not Applicable (see the value marked **NA** in Table VI). The applications that were connected to echoUAT did not crash because echoUAT slowly forward data to the application, hence, indirectly protecting the applications from DoS attacks, but at the expense of dropping legitimate ADS-B packets, which violates the minimal operational specifications of ADS-B. Over the 1090ES ADS-B IN, SkyEcho2 and Sentry can receive and process up to approximately 55 thousand distinct ICAO24 codes per minute, whereas echoUAT surprisingly had a hardware limitation in processing approximately 400 distinct ICAO24 codes per minute. We do not know why such functional discrepancy occurred considering that SkyEcho2 and echoUAT are manufactured by the same vendor.

At the same time, EFBs are graphical user interface-oriented devices or software that are intended to make the service easy and attractive. They define the aircraft location (or "ownship" location) using their built-in GNSS receiver. Based on that location, they show the map of the surrounding area, which is typically 50 to 60 nautical miles in radius. Therefore, we also tested the variations of invisible and silent ADS-B DoS attacks on EFBs using fake aircraft at quite distant locations (e.g., another city, country, or continent) that are generally outside of the EFB's displayed screen, which is mainly centered on the position of the ADS-B receiver (i.e., the attack victim). Thus, there were no visible attacker-injected aircraft on the screen, but the EFB was silently affected by the DoS attack. This new invisible and silent ADS-B DoS attack that we propose and tested would be very challenging (if not impossible) to detect without specific improvements in the ADS-B software (e.g., in the EFB and ATC) aimed at mitigating the list of attacks that we described in this article.

Table VI summarizes the results of our ADS-B-level DoS attack, while we present the complete ADS-B DoS experiment and findings in a separate work.

H. Fuzzing avionics protocols

Fuzzing is a way to discover bugs in software by providing randomized inputs to programs to find test cases of crash causes. Mobile cockpit devices and EFB applications use several different data-link protocols to exchange data, of which the GDL-90 protocol is one of the most popular. We performed protocol fuzzing by forming packets with a real protocol-like format, but some parts malformed by the fuzzing component. As a fuzzing framework, we used the American Fuzzy Lop (AFL) Python implementation (python-afl v.0.7.3). We targeted the IP address of the connected mobile device, and AFL was instructed to send malformed data to it. Of the 10 tested EFBs, fuzzing experiments affected 7 (either crashed or became unresponsive), and the remaining 3 behaved normally during the attack. Table VII summarizes the results of the protocol fuzzing attack, while we present the complete GDL-90 fuzzing experiment and findings in a separate work.

TABLE VI: Summary of the ADS-B DoS attack experiments on both 1090ES and UAT978.

Configurations		Results	
Hardware	Software	1090ES	UAT978
RTL SDR	Dump1090-fa v 4.0	**UNR**	NA
	Dump1090 v 1.09.0608.14	**UNR**	NA
	Dump1090 v 1.15-dev	**UNR**	NA
	Dump978-fa v 4.0	NA	**UNR**
	RTL1090 v 0.9.0.100	CLG	NA
	RTL1090 v 2.11.3.103	CLG	NA
PGR	PlanePlotter v 6.5.1.1	CLG	NA
EVAL-TT-SF1	Micro ADS-B v 1.15.1	CLG	NA
TR-1W	Micro ADS-B v 1.15.1	CLG	NA
PX4	Mission Planner v 1.3.74	CLG	NA
	QGround Control v 4.1.2	CLG	NA
Cube Orange	Mission Planner v 1.3.74	CLG	NA
	QGround Control v 4.1.2	CLG	NA
GDL 52	Garmin Pilot v 10.5.7	CLG	CLG
	Garmin Pilot v 8.0.0 *	CLG	CLG
Sentry	ForeFlight EFB v 13.0.1	**CRA**	**CRA**
SkyEcho2	Airmate EFB v 2.3	**CRA**	**CRA**
	AvPlan EFB 7.10.7	**CRA**	**CRA**
	AvPlan EFB 1.3.14 *	**CRA**	**CRA**
	EasyVFR4 EFB v 4.0.866	MSD	MSD
	EasyVFR4 EFB v 4.0.870 *	MSD	MSD
	FlyQ EFB v 5.0	MSD	MSD
	ForeFlight EFB v 13.0.1	**CRA**	**CRA**
	Stratus Insight EFB v 5.17.3	**CRA**	**CRA**
	OZRunways EFB v 10.10	**CRA**	**CRA**
	OZRunways EFB v 4.4.1*	**UNR**	**UNR**
echoUAT	Airmate EFB v 2.3	MSD	MSD
	AvPlan EFB 7.10.7	MSD	MSD
	AvPlan EFB 1.3.14 *	MSD	MSD
	EasyVFR4 EFB v 4.0.866	MSD	MSD
	EasyVFR4 EFB v 4.0.870 *	MSD	MSD
	FlyQ EFB v 5.0	MSD	MSD
	ForeFlight EFB v 13.0.1	MSD	MSD
	Stratus Insight EFB v 5.17.3	MSD	MSD
	OZRunways EFB v 10.10	MSD	MSD
	OZRunways EFB v 4.4.1 *	MSD	MSD

Note: * = Android version (rest other EFBs are iOS version); UNR= Unreadable output; NA= Not applicable; CLG = Clogged output; MSD = Message dropped; CRA = Crashed.

TABLE VII: Summary of the GDL-90 fuzzing experiments.

Application with GDL-90	Platform	Result
ForeFlight EFB v 13.0.1	iOS	No effect
Stratus Insight EFB v 5.17.3	iOS	**Crashed**
Airmate EFB v 2.3	iOS	**Crashed**
FlyQ EFB v 5.0	iOS	*Unresponsive*
AvPlan EFB v 7.10.7	iOS	**Crashed**
EasyVFR4 EFB v 5.0.866	iOS	No effect
OZRunways EFB v 10.10	iOS	**Crashed**
AvPlan EFB v 1.13.14	Android	**Crashed**
EasyVFR4 EFB v 4.0.870	Android	No effect
OZRunways EFB v 4.4.1	Android	**Crashed**

I. Logically invalid data encoding

While ADS-B can ensure limited data integrity checks via CRC, it does not check by default the validity of the data itself. Therefore, technically correct but logically invalid data can be encoded into ADS-B messages. For example, Figure 9 shows the very high velocity of an aircraft at a very low altitude and vice versa for another aircraft. In our tests, no ADS-B receiving software issued an alert for this kind of irrational data. An attacker can use this to formulate an attack or to puzzle the ATC.

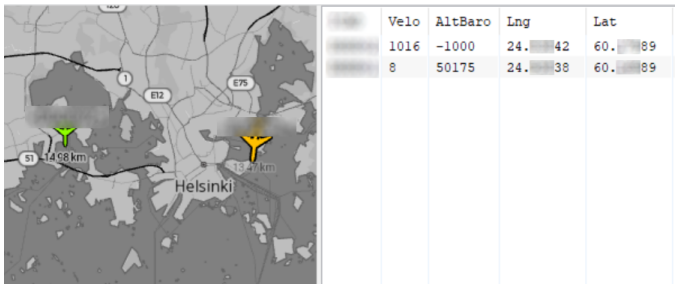


Fig. 9: Logically invalid data displayed in ADS-B Micro.

J. Jamming

1) Signal jamming: This is the oldest type of attack to disrupt any RF-related service. ADS-B 1090 uses a 4.6MHz wide radio spectrum from 1087.7MHz to 1092.3MHz, centering at

1090MHz [41]. On the other hand, UAT978 uses a 1.3 MHz broad spectrum that centers at 978MHz (±0.65MHz) [42]. Almost all of currently available transmission capable SDRs can block these two radio channels using noise transmission. Thus, normal service can be easily suspended. However, an attacker would most likely launch the attack from the ground. Therefore, the jamming attack would not be effective for all the receivers in a wide range of areas. Instead, it could be a local attack. In Figure 10 pink wavy line shows the noise floor, which is below -100 dB. The greenish-yellow wavy line shows the rise of the noise floor around -40 dB for the entire ADS-B 1090ES spectrum due to a jamming attack in our laboratory. None of the receivers in our lab could receive any valid transmission during the signal jamming attack.

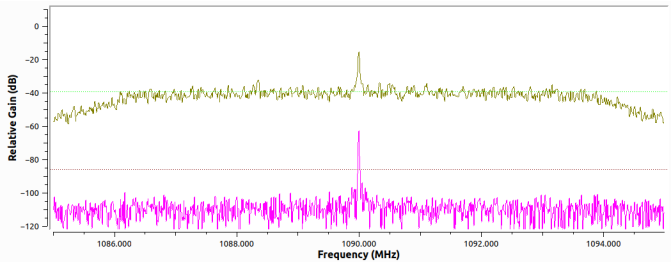


Fig. 10: Rise of the noise floor due to the jamming attack on ADS-B 1090ES.

2) Aircraft disappearance: We used jamming and fake transmission to make a legitimate aircraft disappear. We set a distant receiver using RTL SDR and dump1090. The receiver setup can write the receiving data through the *"./dump1090 –write-json-every* $< t >$*"* command. We jammed the ADS-B channel with a BladeRF using a noise source block of GRC. The jammer produced a noise signal of random values using the Gaussian distribution, which significantly raised the noise floor. We set the jammer near the targeted receiver. The high noise from the jammer degraded the signal-to-noise ratio. As a result, the targeted receiver dropped the legitimate transmission. Then our Python program collected the JSON data from the distant receiver, filtered out the targeted aircraft, created the byte order of the signals, and finally transmitted it into the air using a HackRF at high power mode. We noticed that the targeted aircraft disappeared from the targeted receiver, but the other aircraft were visible. Since the typical range of the ADS-B communication is very large ($\approx$ 300 nautical miles) and there will be many receivers in the targeted area, we doubt that such an attack will be effective in real life, though it may cause some local disturbance. Our main conclusion is that this advanced attack requires huge investments in infrastructure and expertise, which only large organizations or nation-states can afford.

3) Trajectory modification: One way to perform the trajectory modification attack is to further combine *aircraft disappearance* with *aircraft injection* attacks. Therefore, to change the trajectory of a target aircraft, we started with the same strategy that we used to make the plane disappear from the receiver (see above Section VI-J2). In contrast to aircraft disappearance, however, in aircraft injection, after we filter out the data, we periodically broadcast a new flight path (i.e., a modified trajectory) of the targeted aircraft. Hence, the targeted aircraft appeared on receiving displays as having changed its course. Similar to aircraft disappearance, the trajectory modification attack works well in a lab setup but may be hardly practical in the real world in the near future and without considerable technical support.

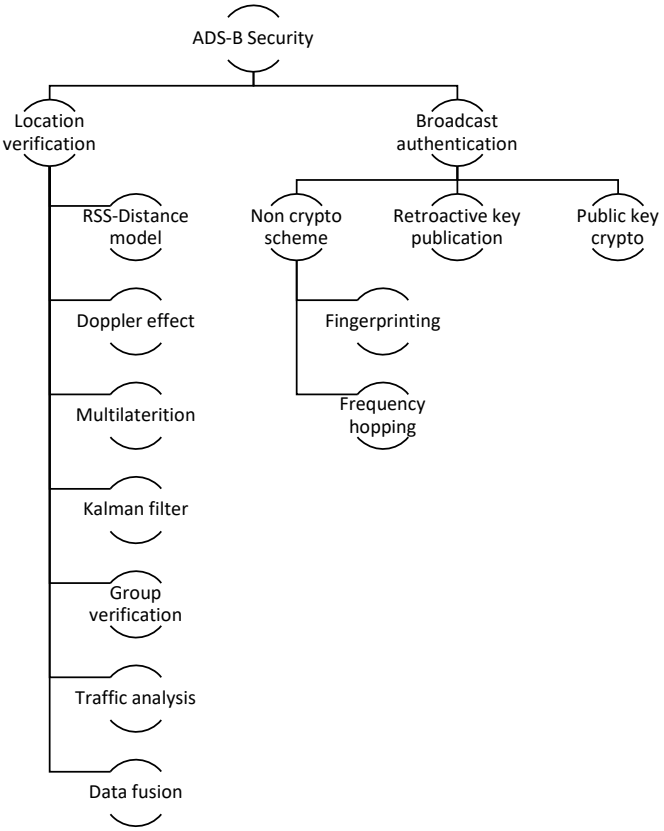


Fig. 11: Ontological tree classification of defensive measures for ADS-B security.

VII. Countermeasures and Defenses for ADS-B Security

Several studies on different approaches to securing the ADS-B communications have been conducted in the past decade [43], [2], [1], [44], [39], [45], [46]. Some promising directions for generic RF communication defenses are explored by the physical layer security (PLS) techniques that were used to secure beamforming [47], [48], [49]. Though we were unable to verify at this point the effectiveness of PLS techniques against our ADS-B attacks, we invite interested readers to further explore the field.

Overall, the proposed defensive solutions in literature can be categorized into two main groups: solutions for location verification and solutions for broadcast authentication. In Figure 11, we present an ontological tree classification of defensive measures for ADS-B security. All the proposed methods have some advantages and disadvantages. Some solutions are very

handy, and some need extensive infrastructure. We suggest that proper guidelines for multiple sources of the same signal, i.e., coordinated attacks, be issued by the regulatory authority. Our tests merely investigated the RSS-Distance model and Doppler effect solutions in a practical manner.

A. Defense using the RSS-Distance model

An RF signal attenuates as it travels through space. The more the signal travels, the weaker it becomes. Thus, the traveled distance and the signal strength are correlated. This phenomenon can be used to verify the source of the signal, i.e., the aircraft. We recorded the three-dimension (3D) distance and the RSS of the aircraft from our laboratory for three days. In Figure 12, the X and Y axes show the distance and the RSS, respectively. The red line shows the raw measurements. The receiving software (dump1090 v1.15dev) provided the RSS in the dBFS unit instead of the standard signal strength in the dBm unit. However, we can observe that the RSS weakened as the aircraft flew farther, regardless of the scale. The raw measurement suffered from noise, so we used the Kalman filter to smooth the noise. The green line shows the Kalman filter values. To make a meaningful model, we applied Python-based *scipy.optimize.curve_fit* function. Finally, we used the blue dotted curve fit data to verify the aircraft's distance (or claimed position) against the RSS.

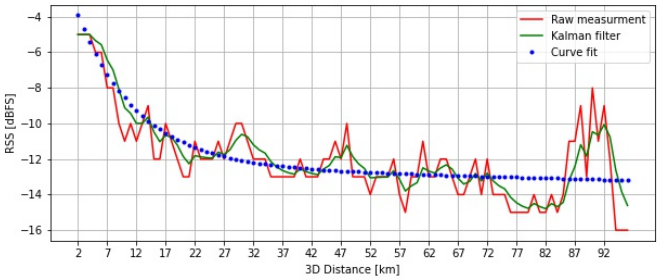


Fig. 12: Example of the the RSS-Distance model created based on the the ADS-B signals from real aircraft.

To distinguish the real aircraft from the spoofed aircraft, we set up a spoofing unit that randomly transmited fake ADS-B 1090ES signals that encoded a random positions. To test our model, we let this spoofing setup be active for three days. The receiver did receive both spoofed and real signals. Based on the given location in the ADS-B message, our setup calculated the 3D distance of the aircraft from the receiver, and then retrieved the possible RSS from the model. If the retrieved RSS and the real-time RSS were close enough, the aircraft was considered legitimate, otherwise, it was considered a fake aircraft. Since the RF signal suffered from noise and fluctuations, we used some tolerance while we compared the retrieved and real RSS values. As an attacker may use different power levels for signals, we tested three different power-level attacks: low power attack (LPA), medium power attack (MPA), and high power attack (HPA). For these three attacks, the RF output gain in the GRC script was set at 10 dB, 20 dB, and 30 dB, respectively. We classified the experiment outcomes into four categories. A True Positive (TP) means the attacker aircraft flagged as attacker aircraft; a True Negative (TN) outcome means the real aircraft was flagged as a real aircraft; a False Positive (FP) outcome means the real aircraft was flagged as an attack; and a False Negatives (FN) outcome means attacker signals were flagged as real aircraft. Using these four parameters and to fully understand model's capabilities, we also calculated the accuracy, precision, recall, and F1 score, as follows:

- "Accuracy" the ratio of the number of correctly predicted observations to the number of total observations. It's formula is Accuracy = (TP+TN)/(TP+FP+FN+TN).
- "Precision" is the ratio of the number of correctly predicted positive observations to the number of total predicted positive observations. Its formula is Precision = TP/(TP+FP).
- "Recall" is the ratio of the number of correctly predicted positive observations to the number of all observations of attacking aircraft. Its formula is Recall = TP/(TP+FN).
- "F1 score" is the weighted average of the precision and the recall. Its formula is F1 score = 2×(Recall × Precision) / (Recall + Precision).

During the experiment, a total of 2,107 test samples were collected. Out of them, 966 were from real airplanes and 1,141 were from attackers' spoofed airplanes. The accuracy metric in Figure 13 shows that high-power attacks are easier to detect, while low-power attacks are harder to detect or else prone to erroneous detection. Similar to the accuracy, the precision also diminishes with low-power attacks. Recall tells us how many predictions were labeled correctly. If the tolerance is high, the recall ratio decreases. The F1 score reveals the accuracy based on precision and recall. The best F1 score was observed during the high-power attack in addition to a high tolerance. Schäfer [50] implemented an RSS profiling-based trajectory verification scheme called VeriFly and evaluated its security by conducting experiments and simulations with real data. Instead of an instantaneous RSS value, the authors used the distribution of RSS as the verification factor. More importantly, the authors did not systematically measure the accuracy of the results in terms of the model versus the outcome; instead, they tried to determine the model parameters that would yield the highest TP and the lowest FN. At the best parameter combination, they achieved approximately 82% success. The work of Schäfer [50] and our RSS-Distance model work are quite different. For example, our model (once pre-trained) provides the result instantly, whereas VeriFly requires cumbersome preparation and conditional calibrations, such as at least 150 ADS-B position messages plus some neighboring messages (e.g., k = 10) within a maximum distance (e.g., 625 meters). Our model does not require such types of conditional calibration. In summary, VeriFly is suitable for post-processing, i.e., after gathering all the messages and checking the valid flights, whereas our model performs real-time instant category profiling (i.e., of legit signals vs. attacker signals) for each position message.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAES.2021.3139559, IEEE Transactions on Aerospace and Electronic Systems

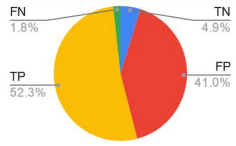
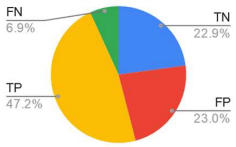
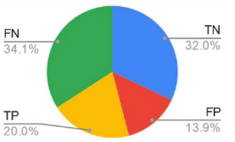
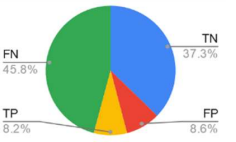
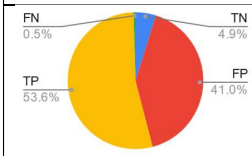
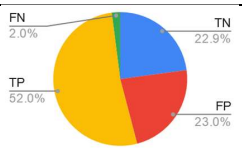
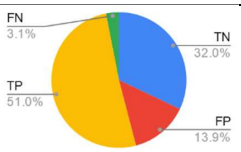
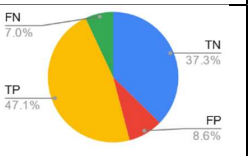
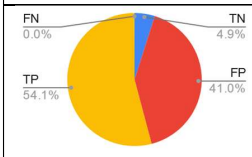
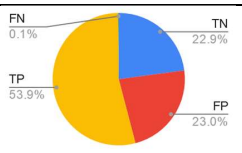
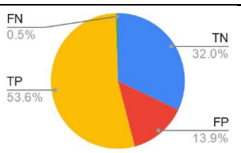
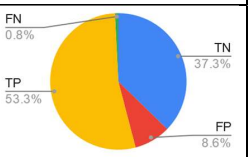
Low Power Attack (LPA)			
Tolerance = ± 1 dBFS	Tolerance = ±2 dBFS	Tolerance =± 3 dBFS	Tolerance = ±4 dBFS
			
Accuracy = 0.57 Precision = 0.56 Recall = 0.97 F1 score = 0.71	Accuracy = 0.70 Precision = 0.67 Recall = 0.87 F1 score = 0.75	Accuracy =0.52 Precision = 0.59 Recall = 0.37 F1 score = 0.45	Accuracy = 0.45 Precision = 0.49 Recall = 0.15 F1 score = 0.22
Medium Power Attack (MPA)			
			
Accuracy = 0.58 Precision = 0.57 Recall = 0.99 F1 score = 0.72	Accuracy = 0.75 Precision = 0.69 Recall = 0.96 F1 score = 0.80	Accuracy = 0.83 Precision = 0.79 Recall = 0.94 F1 score = 0.85	Accuracy = 0.84 Precision = 0.85 Recall = 0.87 F1 score = 0.86
High Power Attack (HPA)			
			
Accuracy = 0.59 Precision = 0.57 Recall = 1.00 F1 score = 0.72	Accuracy = 0.77 Precision = 0.70 Recall = 1.00 F1 score = 0.82	Accuracy = 0.85 Precision = 0.79 Recall = 0.99 F1 score = 0.88	Accuracy = 0.90 Precision = 0.86 Recall = 0.99 F1 score = 0.92

Fig. 13: Results of the detection of the attacker's ADS-B messages when the previously built RSS-Distance model was used.

B. Defense using the Doppler shift

The Doppler shift measures the change in the frequency of a wave in relation to a motion between the transmitter and the receiver. It is a common phenomenon in wireless communications, which is widely used in many applications [51], [40]. However, some studies have suggested using the Doppler shift of an ADS-B signal to verify the velocity, and subsequently, the position of an aircraft [52], [53]. The Doppler shift effect is mainly used to verify whether the signal is coming from a source-in-motion, assuming that the attacker is likely to be in the static mode, while a real aircraft is constantly in motion when it flies.

To test our proposal, we developed a GRC script to record the strongest the strongest positions of the RSS and the frequency in the Fast Fourier Transform (FFT) display. We set the FFT size 32,768 and the sample rate at 250 thousand to produce a fine-granular frequency change (250 kHz / 32,768 = 7.62 Hz) per FFT resolution. We tuned the receiving radio slightly off the center frequency (1090 MHz) to avoid a DC spike (a common problem in SDR). Therefore, the receiving FFT position was around 8,000 instead of $32,768/2 = 16,384$. Figure 14 shows the strongest positions of the RSS and the frequency in the FTT display according to the recorded time. The lower part of the figure show that the RSS increased when the aircraft approached the receiver, and vice versa. Since the aircraft's position was changing, a slight change in the position of the reception frequency was expected in the upper part of the figure. However, despite many attempts, we did not find a good frequency change trend. Had a weaker signal been considered, the noise would have been increased significantly. The ATC is likely to receive a weak ADS-B signals most of the time, since aircraft would not fly in the direct line of sight. Considering our experience with the ADS-B Doppler shift, we conclude that it may be difficult to use the Doppler shift of an ADS-B signal as a reliable indicator of the motion of a valid/authentic ADS-B transponder versus that of a static ADS-B attacker. Even if the motion is verified, it could not block the attacker in motion, e.g., an attacking SDR mounted on a drone or airplane-like UAV, or an attacker SDR planted inside a legitimate flying aircraft.

C. Defense against coordinated attacks

In our view, resiliency to the inconsistencies generated by coordinated attacks in ADS-B messages could (and should)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAES.2021.3139559, IEEE Transactions on Aerospace and Electronic Systems

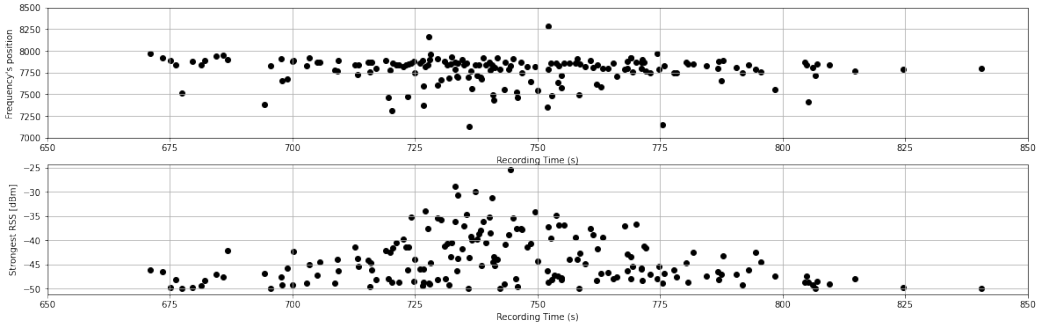


Fig. 14: Position of the strongest RSS in the FFT for evaluation of the Doppler shift.

be achieved by standardizing (across industries, vendors, and geographies) the expected behavior in such anomalous cases. However, to the best of our knowledge, there are no technical or procedural specifications and guidelines for dealing with such cases. In our view, relevant bodies such as Radio Technical Commission for Aeronautics (RTCA), International Civil Aviation Organization (ICAO), U.S. Federal Aviation Administration (FAA), The European Organisation for the Safety of Air Navigation (EUROCONTROL), European Union Aviation Safety Agency (EASA), and Single European Sky ATM Research (SESAR), should issue revised ADS-B specifications and guidelines for ensuring consistent treatment (as well as proper detection and flagging, whether at the hardware and/or software level) of ADS-B messages arising from such coordinated attacks.

D. Defense against other attacks

Below we present some ideas on how to improve existing software so that the user interface or user experience would have sufficient controls for the users in cyberattacks or even when legitimate malfunctions occur.

- Implement simple yet effective detections in software, e.g., detection of anomalous data, illogical data, and fluctuating data.
- Implement better logic to alert the users when the above detections occur, as well as friendly and aerospace-approved ways to notify and handle alerts.
- Offer users the ability to configure some of the display/alert thresholds but provide the software with sensible and well-tested defaults, perhaps based on industry guidelines, specifications, and certifications.

VIII. Conclusion

We practically demonstrated and evaluated the impact of multiple novel and known attacks on ADS-B that are primarily achievable via an RF link and that affect various network, processing, and display subsystems used within the ADS-B ecosystem. Overall, we implemented and tested, in a controlled environment, 12 attacks on ADS-B, of which 5 were presented or implemented for the first time in the field of ADS-B security. For all these attacks, we developed a unique testbed that consisted of 13 hardware devices and 22 software (based on Android, iOS, Linux, and Windows), which resulted in a total of 36 tested configurations. Each of the attacks was successful on various subsets of the tested configurations. In some attacks, we discovered wide qualitative variations and discrepancies in how particular configurations reacted to and treated ADS-B inputs that contained errors or contradicting flight information, and the main culprit was almost always the software implementation. In some other attacks, we managed to cause DoS by remotely crashing/impacting more than 50% of the testset that corresponded to those attacks. Besides demonstrating a few novel attack concepts, we also implemented, investigated, and reported on some practical countermeasures to those attacks. For example, we found and practically demonstrated that the strong relationship between the RSS and the distance-to-emitter may help verify the aircraft's advertised ADS-B position and distance. In some scenarios, we achieved 90% accuracy in detecting spoofed ADS-B signals, and our method might be effectively used to distinguish real aircraft's ADS-B signals from attackers' spoofed signals.

To the best of our knowledge, in terms of the tested configurations and attacks/scenarios, this is the first study and is the largest qualitative and quantitative public study of this kind that targets ADS-B systems. The consistency of our results on a comprehensive range of hardware-software configurations indicates the reliability of our approach and test results. We hope our approach and results can be positively used by research and industry organizations to improve the cybersecurity of today's ever-growing ADS-B deployments.

Acknowledgment

The authors acknowledge the grants of computer capacity from the Finnish Grid and Cloud Infrastructure (persistent identifier *urn:nbn:fi:research-infras-2016072533*).

Major parts of this research supported by cascade funding from the Engage consortium's Knowledge Transfer Network (KTN) project *"Engage - 204 - Proof-of-concept: practical, flexible, affordable pentesting platform for ATM/avionics cybersecurity"* (SESAR Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 783287). All and any results, views, and opinions presented herein are only those of the authors and do not reflect the official position of the European Union (and its organizations and projects, including Horizon 2020 program and Engage KTN).

Part of this research was supported by a grant from the *Decision of the Research Dean on research funding within the Faculty (07.04.2021)* of the Faculty of Information Technology of University of Jyväskylä (The authors thank Dr. Andrei Costin for facilitating and managing the grant).

Hannu Turtiainen also thanks the Finnish Cultural Foundation / Suomen Kulttuurirahasto (https://skr.fi/en) for supporting his Ph.D. dissertation work and research (under grant decision no.00211119) and the Faculty of Information Technology of the University of Jyvaskyla (JYU), in particular, Prof. Timo Hämäläinen, for partly supporting and supervising his Ph.D. work at JYU in 2021–2022.

Last but not least, the authors thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] C. Finke, J. Butts, and R. Mills, "ADS-B Encryption: Confidentiality in the Friendly Skies," in *8th Annual Cyber Security and Information Intelligence Research Workshop*. ACM, 2013.

[2] Z. Wu, T. Shang, and A. Guo, "Security Issues in Automatic Dependent Surveillance - Broadcast (ADS-B): A Survey," *IEEE Access*, vol. 8, 2020.

[3] D. Kožović and D. Djurdjevic, "Spoofing in aviation: Security threats on GPS and ADS-B systems," *Vojnotehnicki glasnik*, vol. 69, 2021.

[4] "Security expert pulled off flight by FBI after exposing airline tech vulnerabilities," http://www.foxnews.com/us/2015/04/16/security-expert-pulled-off-flight-by-fbi-after-exposing-airline-tech.html, 2015, accessed: 2021-06-04.

[5] "Hacker uses an Android to remotely attack and hijack an airplane," https://www.computerworld.com/article/2475081/hacker-uses-an-android-to-remotely-attack-and-hijack-an-airplane.html, 2013, accessed: 2021-06-04.

[6] "WestJet Says It Never Sent Hijack Alarm, Wasn't in Danger," https://www.bloomberg.com/news/articles/2015-01-10/westjet-hijack-signal-called-false-alarm, 2015, accessed: 2021-06-04.

[7] "FAA Warns of ADS-B False Alerts," https://www.flyingmag.com/faa-warns-ads-b-false-alerts, 2017, accessed: 2021-06-04.

[8] FAA, "No Kidding: ADS-B Deadline of Jan. 1, 2020, is Firm," https://www.faa.gov/news/updates/?newsId=90008, 2018, accessed: 2021-06-11.

[9] EASA, "EASA seasonal technical commission," https://www.easa.europa.eu/sites/default/files/dfu/EASA_STC_NEWS_JUNE_2018.pdf, 2018, accessed: 2021-03-02.

[10] P. A. Diffenderfer, D. M. Baumgartner, K. M. Long, S. A. Wilkins, J. G. Menzenski, and C. F. Pertsch, "Evaluation of Using Mobile Devices to Streamline General Aviation Instrument Flight Rules Operations," in *IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 2019.

[11] A. Braeken, "Holistic Air Protection Scheme of ADS-B Communication," *IEEE Access*, vol. 7, 2019.

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

[13] M. Strohmeier, M. Schäfer, V. Lenders, and I. Martinovic, "Realities and challenges of nextgen air traffic management: the case of ADS-B," *IEEE Communications Magazine*, vol. 52, 2014.

[14] Z. Wu, A. Guo, M. Yue, and L. Liu, "An ADS-B Message Authentication Method Based on Certificateless Short Signature," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, 2020.

[15] Z. Wu, T. Shang, and A. Guo, "Security Issues in Automatic Dependent Surveillance-Broadcast (ADS-B): A Survey," *IEEE Access*, vol. 8, 2020.

[16] J. Sun, *The 1090 Megahertz Riddle: A Guide to Decoding Mode S and ADS-B Signals*. TU Delft OPEN Publishing, 2021.

[17] M. Strohmeier, M. Schäfer, V. Lenders, and I. Martinovic, "Realities and challenges of nextgen air traffic management: the case of ADS-B," *IEEE Communications Magazine*, vol. 52, 2014.

[18] M. Strohmeier, V. Lenders, and I. Martinovic, "On the Security of the Automatic Dependent Surveillance-Broadcast Protocol," *IEEE Communications Surveys Tutorials*, vol. 17, 2015.

[19] M. Schäfer, V. Lenders, and I. Martinovic, "Experimental Analysis of Attacks on Next-Generation Air Traffic Communication," in *Applied Cryptography and Network Security*. Springer Berlin Heidelberg, 2013.

[20] M. R. Manesh, M. Mullins, K. Foerster, and N. Kaabouch, "A preliminary effort toward investigating the impacts of ADS-B message injection attack," in *IEEE Aerospace Conference*. IEEE, 2018.

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

[22] D. Lundberg, B. Farinholt, E. Sullivan, R. Mast, S. Checkoway, S. Savage, A. C. Snoeren, and K. Levchenko, "On the security of mobile cockpit information systems," in *ACM SIGSAC Conference on Computer and Communications Security*, 2014.

[23] D. A. Lundberg, "Security of ADS-B Receivers," Ph.D. dissertation, UC San Diego, 2014.

[24] A. Sjödin and M. Gruneau, "The ADS-B protocol and its' weaknesses," Ph.D. dissertation, KTH Royal Institute of Technology, 2020.

[25] M. Leonardi, M. Strohmeier, and V. Lenders, "On Jamming Attacks in Crowdsourced Air Traffic Surveillance," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, 2021.

[26] M. Leonardi, E. Piracci, and G. Galati, "ADS-B vulnerability to low cost jammers: Risk assessment and possible solutions," in *Tyrrhenian International Workshop on Digital Communications - Enhanced Surveillance of Aircraft and Vehicles (TIWDC/ESAV)*, 2014.

[27] N. Pearce, K. J. Duncan, and B. Jonas, "Signal Discrimination and Exploitation of ADS-B Transmission," in *SoutheastCon 2021*, 2021.

[28] D. McCallie, J. Butts, and R. Mills, "Security analysis of the ADS-B implementation in the next generation air transportation system," *International Journal of Critical Infrastructure Protection*, vol. 4, 2011.

[29] M. R. Manesh and N. Kaabouch, "Analysis of vulnerabilities, attacks, countermeasures and overall risk of the Automatic Dependent Surveillance-Broadcast (ADS-B) system," *International Journal of Critical Infrastructure Protection*, vol. 19, 2017.

[30] D. L. Mccallie, *Exploring Potential ADS-B Vulnerabilities in the FAA's NextGen Air Transportation System*. BiblioScholar, 2012.

[31] J. Pollack and P. Ranganathan, "Aviation navigation systems security: ADS-B, GPS, IFF," in *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer, 2018.

[32] B. Haines and N. Foster, "Hackers + Airplanes = No good can come of this (Spoofing ADS-B)," *DEFCON 20*, 2012.

[33] T. Li and B. Wang, "Sequential collaborative detection strategy on ADS-B data attack," *International Journal of Critical Infrastructure Protection*, vol. 24, 2019.

[34] K. Domin, I. Symeonidis, and E. Marin, "Security analysis of the drone communication protocol: Fuzzing the MAVLink protocol," Master's thesis, University of Luxembourg, 2016.

[35] T. Kim, C. H. Kim, J. Rhee, F. Fei, Z. Tu, G. Walkup, X. Zhang, X. Deng, and D. Xu, "RVFuzzer: Finding input validation bugs in robotic vehicles through control-guided testing," in *28th {USENIX} Security Symposium*, 2019.

[36] J. Naganawa and H. Miyazaki, "A Method for Accurate ADS-B Signal Strength Measurement Under Co-Channel Interference," in *Asia-Pacific Microwave Conference (APMC)*, 2018.

[37] "Ongoing police operation at of Amsterdam Schiphol Airport following 'incident' on plane," https://mobile.twitter.com/airlivenet/status/1192168809974632450, 2019, accessed: 2021-06-11.

[38] M. Strohmeier, V. Lenders, and I. Martinovic, "Security of ADS-B: State of the Art and Beyond," *IEEE Communications Surveys and Tutorials*, vol. 17, 2013.

[39] N. Ghose and L. Lazos, "Verifying ADS-B navigation information through Doppler shift measurements," in *IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, 2015.

[40] Z. Yongsheng, H. Dexiu, Z. Yongjun, and L. Zhixin, "Moving target localization for multistatic passive radar using delay, Doppler and Doppler rate measurements," *Journal of Systems Engineering and Electronics*, vol. 31, 2020.

[41] *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*, International Telecommunication Union, 2017.

[42] *Standards and Recommended Practices for the Universal Access Transceiver (UAT)*, International Civil Aviation Organization, 2005.

[43] H. Yang, Q. Zhou, M. Yao, R. Lu, H. Li, and X. Zhang, "A Practical and Compatible Cryptographic Solution to ADS-B Security," *IEEE Internet of Things Journal*, vol. 6, 2019.

[44] Y. Kim, J.-Y. Jo, and S. Lee, "A secure location verification method for ADS-B," in *IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016.

[45] M.-S. Huang, R. Narayanan, Y. Zhang, and A. Feinberg, "Tracking of Noncooperative Airborne Targets Using ADS-B Signal and Radar Sensing," *International Journal of Aerospace Engineering*, 2013.
[46] Y. Kim, J.-Y. Jo, and S. Lee, "ADS-B vulnerabilities and a security solution with a timestamp," *IEEE Aerospace and Electronic Systems Magazine*, vol. 32, 2017.
[47] Z. Lin, M. Lin, T. de Cola, J.-B. Wang, W.-P. Zhu, and J. Cheng, "Supporting IoT with rate-splitting multiple access in satellite and aerial integrated networks," *IEEE Internet of Things Journal*, 2021.
[48] Z. Lin, M. Lin, B. Champagne, W.-P. Zhu, and N. Al-Dhahir, "Secure beamforming for cognitive satellite terrestrial networks with unknown eavesdroppers," *IEEE Systems Journal*, vol. 15, 2020.
[49] ——, "Secure and energy efficient transmission for RSMA-based cognitive satellite-terrestrial networks," *IEEE Wireless Communications Letters*, vol. 10, 2020.
[50] M. Schäfer, "Design and Analysis of VeriFly - A Trajectory Verification Method based on RSS sampling," Master's thesis, University of Kaiserslautern, 2013.
[51] R. Raney, "The delay/Doppler radar altimeter," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, 1998.
[52] N. Ghose and L. Lazos, "Verifying ADS-B navigation information through Doppler shift measurements," in *IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, 2015.
[53] M. Schäfer, P. Leu, V. Lenders, and J. Schmitt, "Secure Motion Verification Using the Doppler Effect," in *9th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2016.