On the Security of the FLARM Collision Warning System

Boya Wang
boywang@student.ethz.ch
ETH Zurich
Zurich, Switzerland

Giorgio Tresoldi, Martin Strohmeier,
Vincent Lenders
Cyber-Defence Campus, armasuisse Science + Technology
Thun, Switzerland
first.last@armasuisse.ch

ABSTRACT

In the past decade, the vulnerability of aircraft communications against low-resourced attackers has received significant attention both in the information security community and from aviation industry and regulators. Until now, research on attacks against such communications technologies has focused on larger aircraft, neglecting the technologies used in light aircraft and unmanned aerial vehicles (UAV). As such lighter aircraft make up a large and growing majority of both airspace users and casualties, this is a glaring oversight from a security and safety perspective.

To alleviate this problem, we conduct a first comprehensive security analysis of the FLARM ecosystem, a collision avoidance system focusing on light aircraft and UAV with a large and increasing distribution in many countries. Besides being used in more than 40,000 aircraft, several large providers of air traffic data capture the unauthenticated FLARM data using ground receivers in order to feed crucial downstream applications.

To show the vulnerability of this ecosystem, we design, implement and evaluate a practical spoofing attack, which is capable of interacting with FLARM receivers in the air and on the ground. The demonstrated attack uses low-cost hardware and open-source software and is as such accessible to a wide range of threat actors, resulting in a significant potential impact on the safety of all airspace users. Finally, in order to counter such attacks, we propose a first anomaly detection approach based on physical-layer characteristics and validate it with real-world aircraft data.

CCS CONCEPTS

• **Security and privacy → Distributed systems security**; • **Hardware → Safety critical systems**; *Wireless devices.*

KEYWORDS

FLARM, aircraft, UAV, spoofing, collision avoidance

ACM Reference Format:
Boya Wang and Giorgio Tresoldi, Martin Strohmeier,, Vincent Lenders. 2022. On the Security of the FLARM Collision Warning System. In *Proceedings of the 2022 ACM Asia Conference on Computer and Communications Security (ASIA CCS '22), May 30-June 3, 2022, Nagasaki, Japan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3488932.3517409

ASIA CCS '22, May 30-June 3, 2022, Nagasaki, Japan

ACM ISBN 978-1-4503-9140-5/22/05…$15.00
https://doi.org/10.1145/3488932.3517409

1 INTRODUCTION

Despite many procedural and technical improvements in aviation over the previous decades, mid-air collisions (MACs) remain one of the most prominent threats to any airspace user, i.e., pilots and passengers. Due to their critical nature, MACs are almost always fatal and even a single event can cause a high death toll, with many of the biggest aviation disasters all involving aircraft collisions [49].

For example, an average of 22.6 critical near-MAC has been reported every year between 2000 and 2020 in the US [30]. As most of these near-accidents happened in good visibility and daylight, it is clear that the human visual system is not always sufficient to detect fast objects in due time in order to avoid collisions. Therefore, it is necessary to use technical collision avoidance systems to assist the pilot in detecting and avoiding potential collisions.

So-called airborne collision avoidance systems (ACAS) provide an onboard safety net should normal air traffic control (ATC) procedures fail to keep aircraft separated. These systems are now widely deployed around the globe and have been constantly refined over the past three decades, usually in response to near misses or actual fatal mid-air collisions.

Several wireless protocols exist which aim to reduce the risk of MAC for example the ACAS implementations *ACAS X* and *TCAS* (Traffic Alert and Collision Avoidance System), which are used in larger aircraft such as commercial airliners [40]. Due to the legacy nature of many of the underlying technologies such as Mode S and Automatic Dependent Surveillance – Broadcast (ADS-B), hackers and academic researchers keep reporting security vulnerabilities [5, 34, 35, 46], which have been shown to have a real impact on the security (and thus the safety) of using these protocols [42, 43]. Until now, the focus of such security investigations has been put exclusively on larger, heavy aircraft such as commercial airliners or business jets, which typically use the cost-intensive TCAS.

Among the existing collision avoidance technologies, FLARM is the only one tailored for light aircraft and is widely used in Europe. The PowerFLARM devices produced by FLARM Technology Ltd. are relatively cheap and easy to use on-board. It can calculate a projected flight path based on parameters such as speed, track, turn radius and wind conditions, which is imperative for smaller lighter (even wind-powered) aircraft. Combined with a display, the FLARM system shows relative positions of other aircraft nearby and generates alarm once a potential collision is detected. Nowadays, over 40,000 aircraft are using FLARM, with the number constantly increasing [14]. Even many small drones and UAV are now being equipped with this technology. In recent developments, several governments also show interest in the FLARM ecosystem [21, 50]. While its huge safety contribution is undisputed, there has been no public work examining its security against wireless attacks to date.

Although the FLARM radio protocol features message encryption in order to ensure integrity and confidentiality [9], the protocol has been reverse-engineered and its encryption keys are publicly available [2]. We show that this state of affairs not only enables the reception of confidential messages (as is currently facilitated by several open source projects) but also allows the spoofing of accurate FLARM messages, that cannot be distinguished from those of a real aircraft.

Contributions. We make the following contributions in this work:

- We designed a first effective spoofing attack against the FLARM collision avoidance system. We implement it with low-cost hardware and open-source software, demonstrating that the attack can be conducted by a wide variety of threat actors. This is despite existing cryptographic countermeasures, which are known to be ineffective.
- Using our spoofing system, we demonstrate the vulnerabilities of the FLARM technology in an experimental setting with maximum precautions. Our attack analysis illustrates the impact both on airborne receivers in aircraft and on the ground receivers used by aviation data providers.
- We propose first FLARM spoofing mitigations based on anomaly detection and physical-layer characteristics. We evaluate these countermeasures using large-scale FLARM data collected through the OpenSky Network.

Our security analysis indicates that the current state of the art of FLARM data collecting methodology is not sufficient for monitoring or examining FLARM system security, and thus, concluded that more efforts are needed to address this issue in the future.

The remainder of this work is structured as follows. Section 2 provides the necessary background on the FLARM ecosystem and Section 3 our threat model. Section 4 describes the design of an effective spoofing system. Section 5 evaluates the spoofing attack both on airborne and ground receivers. Section 6 proposes potential mitigations for this attack. Section 7 discusses the results and the lessons learned from our work. Finally, Section 8 presents the related work before Section 9 concludes this paper.

2 BACKGROUND

FLARM (a portmanteau of “flight” and “alarm”) is a system used to prevent potential aviation collision and to raise awareness of the pilot. The system obtains the aircraft’s own position from an internal GPS (or potentially other GNSS) receiver, then calculates a projected flight path considering its speed, acceleration, track, turn radius, wind and other parameters [14].

The flight path prediction is encoded and encrypted as a message which is sent to and received by other aircraft equipped with a FLARM device. Upon receiving such messages, the FLARM system may issue alarms to alert the pilot or show the relative position if other aircraft are within detection range but there is no predicted collision [10].

Figure 1: PowerFLARM Fusion. As one of the two main products of FLARM Technology, it can be installed on aircraft.

Compared to other ACAS, the FLARM system is optimized for light aircraft’s needs. It has low power consumption and is relatively inexpensive to purchase and install. Furthermore, conventional collision avoidance systems usually generate large amount of unnecessary warnings about all aircraft nearby, while light aircraft can be close to each other without danger of collision. The pilot can be overwhelmed by those false-positive alerts, hence not taking it seriously if the aircraft on a real collision course. On the contrary, the FLARM system is more accurate by using a novel flight path prediction algorithm and selectively raising alerts [4].

2.1 The PowerFLARM Fusion Device

PowerFLARM Fusion shown in Figure 1 is the state-of-the-art device manufactured by FLARM Technology [13]. It is designed to protect light aircraft and UAV from aircraft otherwise not visible in time. In particular, it addresses challenges such as flying aircraft in lower airspace and fast increase on high speed aircraft. The current PowerFLARM version has enhanced features compared to the older “Classic” FLARM. More concretely, it has larger detection range, dual antenna diversity, better interference protection, ADS-B compatibility and intuitive obstacle warnings [11].

There are two types of FLARM devices available for customers: PowerFLARM Fusion and PowerFLARM Portable. PowerFLARM Fusion combines the most comprehensive configuration of FLARM system with an easy maintenance interface. As its name shows, PowerFLARM Portable is a device for aircraft where a behind-the-panel installation is not possible. In this project, only PowerFLARM Fusion is used for spoofing system design and evaluation. Therefore, PowerFLARM Fusion will be the only device discussed in the following. All the results should also be able to generalize to PowerFLARM Portable since there is no concrete difference with respect to the implementation of FLARM protocol itself.

Figure 1 shows the current version of PowerFLARM Fusion. It is able to be powered on or connected via several types of hardware interfaces. Furthermore, it can be easily configured by FLARM Hub, which is a web app running on the device [20]. FLARM Hub includes its configuration settings, a radar-like traffic monitor and real-time data port. More technical specifications can be found in Table 1. This demonstrates that PowerFLARM Fusion is light-weighted and versatile for all types of aircraft, making it the go-to choice for smaller aircraft such as gliders and small drones.

Table 1: Technical specifications of the PowerFLARM Fusion, which also includes an ADS-B receiver.

Product Name	PowerFLARM Fusion
Type	Installed
Display	Separate
Recommended for	All aircraft types
Dimensions	41 x 80 x 120 mm
Mass	285 g
Power supply	12-32 V DC
Power consumption	165 mA 12 V DC
Data ports, storage	2(D-sub DE-9 and RJ45), USB
Transponder/ADS-B receiver	Included

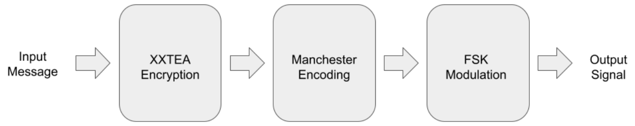


Figure 2: A FLARM input message is first encrypted using XXTEA, then Manchester encoded, finally FSK modulated.

2.2 The FLARM Protocol

Figure 2 illustrates FLARM's high-level sending procedure. An input message is encrypted, Manchester-encoded, and finally modulated onto the channel with frequency shift keying (FSK). Upon receiving FLARM messages, the receiver reverses this procedure.

2.2.1 Modulation. FLARM devices are based the nRF905 chip. Depending on the geographical area they operate in, they transmit in the SRD860 band or in the ISM-band that can be used freely [7]. In Europe, Africa and Asia the two frequencies 868.2 MHz and 868.4 MHz are used, sending one to two messages per second per frequency. On 868.2MHZ, it transmits from 0.4s to 0.8s; On 868.4MHZ, it transmits from 0.4s to 1.2s [17]. In the Americas, Oceania and Israel another undisclosed frequency hopping scheme is in place, in order to comply with local regulations. The FLARM devices modulate the signal onto the carrier using Frequency Shift Keying: spaces (0) are sent at $-50\,kHz$ and marks (1) at $50\,kHz$. The symbols are Manchester encoded.

Table 2: FLARM packet structure with an example.

Fields	Preamble	Start	Radio ID	Type	Payload
Bit Length	20	24	24	8	184
Example	a:aa:66	31:fa:b6	8b:08:de	20	7e:47[...]

2.2.2 Encoding. The packet structure is shown in Table 2: every transmission is preceded by the standard preamble for the nRF905 chip [10101010101001100110] (Manchester encoded). It is then followed by the header that contains 7 bytes. The first three bytes of the header statically contain 0x31fab6, a sync word indicating the start of the data. The next three bytes (in little-endian) are the device's 24-bit radio ID: either the aircraft's unique ICAO address assigned by the national Civil Aviation Authority or the default one assigned by the manufacturer. The type of radio ID also influences the header's last byte which is 10 in case of an ICAO address and 20 in case of the default ID. Table 3 shows the decrypted content.

Table 3: FLARM packet structure with its functions. "x" represents bits with unknown function.

Byte No.	Bits		Function
0	DDDD	DDDD	Device address
1	DDDD	DDDD	
2	DDDD	DDDD	
3	00BB	0000	BB = 10 or 01
4	VVVV	VVVV	Vertical speed
5	xxxx	xxVV	Unknown
6	GGGG	GGGG	GPS status
7	TTTT	GGGG	Plane type
8	LLLL	LLLL	Latitude
9	LLLL	LLLL	
10	AAAA	ALLL	
11	AAAA	AAAA	Altitude
12	NNNN	NNNN	Longitude
13	NNNN	NNNN	
14	xxxx	NNNN	
15	MMxx	xxxx	Multiplying factor
16	HHHH	HHHH	Horizontal
17	SSSS	SSSS	speed (N/S) for
18	KKKK	KKKK	collision
19	TTTT	TTTT	forecast
20	EEEE	EEEE	Horizontal
21	WWWW	WWWW	speed (E/W) for
22	PPPP	PPPP	collision
23	QQQQ	QQQQ	forecast

2.2.3 Trajectory Prediction. As FLARM is a proprietary product, there is little public information about the exact inner workings of the trajectory prediction algorithm that powers the collision alert function. One version has been developed by ONERA in France and been licensed to FLARM Technology Ltd [18].

At a high level, the documentation [8, 12] describes it as follows: The device calculates its own predicted flight path for about the next 20 seconds. This prognosis is based on immediate past and current vectors, including but not limited to aircraft type, speed, vertical speed, turning radius etc. In addition, it uses a movement model that has been optimized for the respective user. According to the prediction of flight path, traffic collision warnings are issued based on the time remaining to the predicted collision. The warning consists of the distance, relative bearing, and altitude difference to the intruder. According to the manual of PowerFLARM Fusion [12], there are three levels of warnings with different types of annunciations: The first warning is issued around 18 seconds before impact, the second warning is issued around 12 seconds before impact and the third warning is issued around 8 seconds before impact. The warning is active as long as the collision risk remains and will change accordingly.

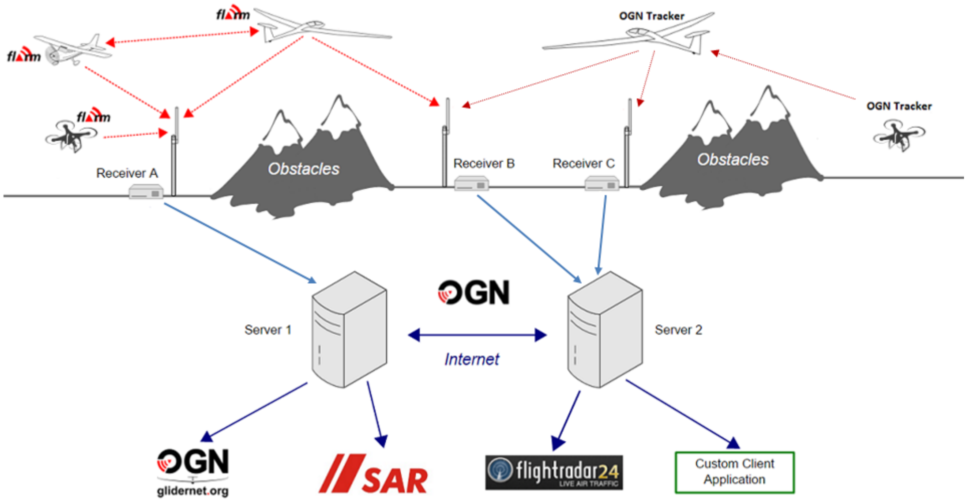


Figure 3: Architecture of OGN from [29]. It consists of servers, crowdsourced ground receivers, aircraft with FLARM devices on board and corresponding downstream applications.

However, the inventors of FLARM have described an approach to trajectory prediction for low-cost collision avoidance systems like FLARM in [4]. Their approach, dubbed interacting multiple model (IMM) algorithm, uses the GPS position and velocity vector as input (using a 1 second sampling rate), while the output is a set of position predictions for the next 20 seconds, which is assumed to be a good time period for slower and smaller aircraft.

The IMM algorithm is then structured into three stages. First, it estimates the aircraft's state of motion. Then the speed of the aircraft relative to the air and its turn rate are estimated and wind estimations are incorporated. In a second step, aircraft maneuvers are detected. Third, a pilot behavioral model is used to predict the turn rate relative to the ground. Eventually, based on the obtained estimates, the flight path is predicted [4]. In particular, the integration of wind and typical gliding behavior into the system differentiates it from collision avoidance systems like TCAS.

In our security analysis in this paper, we will abstract away from the exact algorithm and treat the system as a black box.

2.3 FLARM Security and Encryption

From 2008 onwards, FLARM Technology began to encrypt all transmissions, claiming that "[t]he channel is encrypted to ensure safety, integrity, and privacy. Users can freely configure the level of privacy they require" [10]. This is in stark contrast to practically all other wireless aviation communication technologies, whose standards do not implement or foresee any use of cryptographic measures [47].

Since 2015, the message encryption used in FLARM implements the Corrected Block Tiny Encryption Algorithm (XXTEA) [26], a block cipher based on the Tiny Encryption Algorithm [54] (which FLARM used previously). It is notable for its simplicity of implementation and thus low computational requirements and low power consumption, making it suitable for the glider and small UAV environment. The encrypted payload has a fixed length of 24 bytes. In the encoding process, this payload is prepended with preamble, sync word and address. In addition, the checksum is appended at the end of payload. The whole package is then modulated and sent.

Despite the fact that messages are encrypted in this way, the FLARM protocol is still considered to be insecure in several ways. First, the original XXTEA algorithm with 6 rounds is considered broken since 2010, when Yarrkov published a chosen-plaintext attack needing only 2^{59} queries for a block size of 212 bytes or more [58]. Apparently, FLARM still uses the XXTEA algorithm with 8 rounds and has not modified it despite the publication of the attack.

Second, and more urgently, the encryption keys used are centrally-held, and not regularly rotated. It seems that the encryption keys are only available to partners and other manufacturers in order to implement compatible FLARM products. However, a reverse-engineering document of the protocol itself was published in 2008 under the pseudonym Hiram Yaeger [56]. It describes the packet format in detail as well as the encryption keys. In response to these revelations, FLARM has changed the encryption key several times in the following years. For instance, it changed the keys (and updated the encryption algorithm) in 2015 [52]. However, those changes were quickly reverse-engineered again and published [2]. Again, in April 2017, FLARM introduced a new set of keys, which were subsequently leaked [52]. There is a total of six static keys that are combined with both the UNIX UTC timestamp shifted six bits to the right and the device's address to obtain four dynamic keys that are used in the XXTEA algorithm. Using these keys, the receiving of FLARM messages is possible for everyone. Consequentially, the sending of FLARM messages is also feasible.

2.4 Platforms Using FLARM Ground Receivers

While FLARM is used primarily for situational awareness and airborne collision avoidance, the wireless nature of FLARM allows for the reception of the signals in a crowdsourced fashion, similar to other technologies used to track large aircraft (e.g., Flightradar24[1],

[1] https://flightradar24.com

which uses ADS-B). Due to the practical situation explained in the previous section, these platforms can receive the encrypted FLARM signals and decrypt them using the published keys.

2.4.1 Open Glider Network. Open Glider Network (OGN) is a project that aims to create and maintain a unified tracking platform for aircraft equipped with FLARM and OGN trackers [29]. It provides freely available tracking data to anyone. OGN consists of:

- Servers that receive and forward data.
- A device database that allows users to register new aircraft.
- Ground receivers that are owned, operated and maintained by community members. These receivers listen to and decode the location data of aircraft in their vicinity, then feed the information to servers.
- Software that can be installed on PC or mini-board computers (e.g. Raspberry Pi). It could be used to build receivers.
- Websites and applications that utilize and display the data.

Figure 3 shows the architecture of OGN. FLARM messages are sent from FLARM device on board. OGN ground receivers receive the messages and decode them to feed information into OGN servers. Servers are connected with each other and share information with further downstream applications.

2.4.2 The OpenSky Network. The OpenSky Network [38] is a non-profit association providing open access to real-world ATC data. Since the launch of the project in 2012, it has collected raw data and archived it in a large historical database. The database is used by researchers from different fields to analyze and improve ATC processes. The main ATC technologies used by OpenSky are ADS-B and Mode S. Since late 2018, it also supports FLARM.[2]

An advantage of the OpenSky Network is that the interface of historical database is well-defined and freely available to institutional researchers. Users can query the database using customized commands to get historical and live FLARM records. On the opposite, OGN does not allow the re-distribution of data older than 24 hours [27]. Hence, the OpenSky Network provides a good technical option for FLARM data analysis. For this work, we obtained permission from the OpenSky Network to conduct a security analysis of the downstream effects of attacking the FLARM ecosystem. We collaborated closely with the technical staff of the OpenSky association in order to minimize the impact on the system and its users and mitigate any side effects.

2.4.3 Downstream Applications. Beyond platforms such as OGN and OpenSky distributing FLARM ground receivers directly, there are many applications that source air traffic data (including FLARM data) via these sites. Examples of such downstream applications are shown in Figure 3 for OGN. Besides showing the FLARM-based glider traffic on their own website, the data is further delivered to Flightradar24 and to crucial Search and Rescue (SAR) operations.[3] From Flightradar24, a large commercial tracker with many business relationships, the data is further distributed not only to aviation enthusiasts around the world but also to airlines, air navigation service providers and other aviation industry members requiring accurate data.

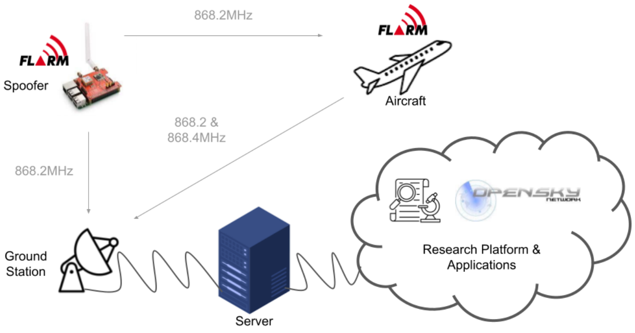


Figure 4: Overview of the attack scenario. The spoofing device can feed signals to both receivers on aircraft and receivers of aviation data provider on the ground.

The OpenSky Network provides a further class of downstream applications, its historical data is used by hundreds of academic and institutional researchers around the world and has resulted in more than 250 papers to date. The use cases of the air traffic data range from air traffic management, to wireless security to climate change research and earth sciences [44]. Naturally, the data should be as accurate as possible and OpenSky introduces various integrity checks at the level of the receiver as outlined in [37]. However, at the level of individual spoofing attacks, there are no currently implemented detection mechanism as we show in Sec. 5.

In conclusion, while FLARM data received on the ground is not (or at least should not) be used operationally for air traffic control, it is influencing air traffic management planning at many institutions.

2.5 SoftRF

SoftRF is an open-source project published by Linar Yusupov [59]. It implements an alternative aviation collision avoidance transceiver in C, supporting several protocols including FLARM. The goal of this software is to support enthusiasts who want to build their own ACAS with low-cost hardware such as a Raspberry Pi.

3 THREAT MODEL

Traditionally, aviation has implicitly and explicitly used an electronic warfare threat model to deal with deliberate communication interference. This means that until the early 2010s, when hackers and academics demonstrated the ease of spoofing aircraft and air traffic control systems with cheap software-defined radio hardware the consideration and defense against such attacks was left entirely to the military. Adversaries outside the state and military realm were considered to have inferior technological and financial capabilities as well as the required knowledge about aviation communication technologies [47].

As this model has been overtaken by the technical developments of the recent past, we consider a much less powerful attacker in this work. The commercial-off-the-shelf equipment required to execute the proposed attacks in the real-world is straightforward to acquire, and is composed of any SDR able to transmit on VHF frequencies, costing around $300-500, and a suitable antenna. This is sufficient

[2] See https://opensky-network.org
[3] A complete list of applications is available at http://wiki.glidernet.org/about

to attack a ground receiver in the line of sight of a simple stationary ground attacker.

If the goal for the same stationary ground attacker is to attack an airborne receiver, an amplifier to increase transmission power so signals reach the aircraft is likely needed, which is available for approximately $600 (possibly more depending on desired transmission range). Airborne attacks are aided by the fact that FLARM-equipped light aircraft and drones are typically flying at much lower altitudes (below 1000 m) compared to, for example, ADS-B-equipped commercial airliners.

As the acquisition of this equipment does not require a large budget, attacks can be performed by any motivated adversary, but exclude some very low-resource attackers. With regards to suitable software, FLARM decoders are openly available but encoders/senders are not. While we develop one in this work to demonstrate the practical attack, we do not make it publicly available in order to not unnecessarily decrease the ease of creating an attack for a malicious adversary. Fig. 4 provides an illustration of the attack scenario.

4 SPOOFING SYSTEM DESIGN & IMPLEMENTATION

To demonstrate the vulnerability introduced by using static and known encryption keys, we design a spoofing system which is able to interact with all current FLARM devices, on aircraft and on the ground. It consists of a sender based on an embedded device with radio frequency (RF) capabilities. This spoofing system is able to generate signals that are indistinguishable from those of an authentic FLARM device. These could potentially generate alerts that can hardly be distinguished by pilots, and therefore, cause considerable distractions.

4.1 Requirements

The spoofing system should send well-formatted FLARM messages containing all required fields, e.g. device identifier, timestamp, speed, location etc. The spoofed signals should be received by the OGN ground receivers which will show its impact on any application using that data. In addition, the spoofing system is required to be able to interact with other authentic FLARM device, i.e. an authentic FLARM device in an aircraft could receive the spoofing coordinates and generate corresponding alarms if the spoofing target is in alarm range. Notice that the spoofing coordinates are not necessarily consistent with the spoofing device's physical location. This design gives an adversary the ability to spoof targets located anywhere.

Following the requirements mentioned above, the spoofing system consists of two conceptual parts: a FLARM message sender and an OGN ground receiver.

4.2 Raspberry Pi-Based SoftRF FLARM Sender

The spoofing FLARM sender is built on Raspberry Pi, Dragino Lora/GPS HAT and SoftRF software.

We choose the Raspberry Pi platform because it is a cheap, easy-to-get device supporting SoftRF. There are also comprehensive publicly accessible instructions on Raspberry Pi configuration. From an adversary's perspective, these characteristics allow it to launch the attack with a low budget and demonstrate that such a spoofing attack can easily be reproduced by people with relatively little professional background.

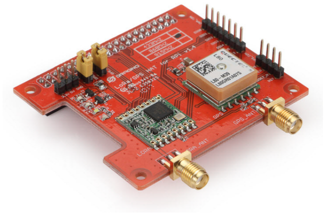


Figure 5: The Dragino Lora/GPS HAT contains a GPS chip, which can feed a GPS signal to the Raspberry Pi [55].

Specifically, we used a Raspberry Pi 4 Model B in our spoofing FLARM sender. But it is noted that the previous version, e.g. Raspberry Pi 3 Model B, should also be able to support the sender with little difference in performance.

Dragino Lora/GPS HAT (Figure 5) is an expansion module for use with Raspberry Pi [55]. It is based on an SX1276/SX1278 transceiver with an add-on L80 GPS that can provide GPS information for applications on the Raspberry Pi. The HAT can be pre-configured to one of the three frequency bands 433 MHz, 868 MHz and 915 MHz. For our FLARM spoofing system, 868 MHz is used. Its programmable bit rate is up to 300 kbps by design, but the actual average rate could be lower [55]. Although a GPS antenna is already integrated into the L80 GPS, an external antenna can be connected to enhance its performance.

SoftRF receives GPS information from the Dragino Lora/GPS HAT, encrypts, encodes and modulated it into FLARM format. Originally, it can only send out its real physical location. To achieve arbitrary GPS location spoofing, we revise the code such that it could read from a configuration file and send out any location information written in it.

After making these changes, the GPS coordinates delivered by the Dragino Lora/GPS HAT are not used anymore. However, getting the time from the GPS signal is still very useful as it ensures that the time interval of sending spoofing FLARM messages are correct. Indeed, it is possible to set the sending time and frequency purely in the software, hence easing the dependence on GPS signals. However, in this case more effort will be needed in order to correctly take care of the time synchronization. To avoid any potential problem from synchronization and keep the code as simple as possible, we choose to continue using the GPS PPS in order to trigger the sending behavior at the right time.

4.3 Raspberry Pi Based OGN Ground Receiver

The second part of our spoofing system consists of an OGN ground receiver which is able to decode and feed spoofing FLARM messages into the OGN server software. The receiver is based on another Raspberry Pi using a common DVB-T dongle for reception and the corresponding software downloaded from the OGN project [29].

With similar reasons mentioned in the previous section, we again chose Raspberry Pi as part of the hardware. As for DVB-T dongle, we decided to use the RTLSDR RTL2832U DVB-T V3 [32]. It can be used with an active antenna and provide stable performance with

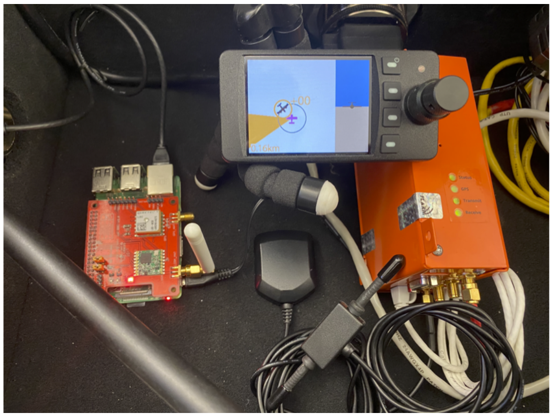


Figure 6: Experimental setup in a Faraday cage. It contains a spoofing sender, a ground receiver and an authentic PowerFLARM Fusion device.

a relatively low cost of about $30. The RTLSDR RTL2832U DVB-T V3 is also recommended on OGN website as the most suitable choice among several DVB-T dongles [28]. We use antenna kit bundled with the dongle for convenience. Using this hardware, the OGN project network provides the software needed for building a typical OGN ground receiver with detailed instructions [29]. After installing, the user can check the local console output in order to see any FLARM messages received by it (ordinarily by gliders/small aircraft using FLARM in the line of sight).

5 EVALUATION

To evaluate our spoofing system, we carried out two laboratory tests, one with the original PowerFLARM Fusion device and one with a receiver feeding to the OpenSky Network, which supports the OGN feeder software. As the FLARM End User License Agreement [19] expressly forbids the spoofing of their devices,[4] we obtained an R&D exception from FLARM Technology Ltd. for our experiments. These two experiments were designed to demonstrate that our spoofing system has the capability of interacting with both authentic FLARM devices deployed in aircraft and FLARM-collecting data platforms. They were further designed to be conducted in an environment that should be as realistic as possible while fully preventing any negative effects on outside users.

In the second part of this chapter, more detailed analysis on system limits is shown to provide insights into the ability of a potential adversary. The analysis demonstrates that an adversary with one single spoofing device can spoof multiple targets, which has the potential to cause considerable distraction to pilots.

5.1 Legal and Ethical Issues

Considering the criticality of collision avoidance systems in aviation and the potentially serious influence on operational FLARM systems, maximum precautions were taken to not cause interference while conducting a safe and yet realistic evaluation of the proposed attack system. We conducted all experiments inside a Faraday cage. This way, we could ensure that no spoofing signal would be leaked and hence potentially received by any other authentic FLARM devices.

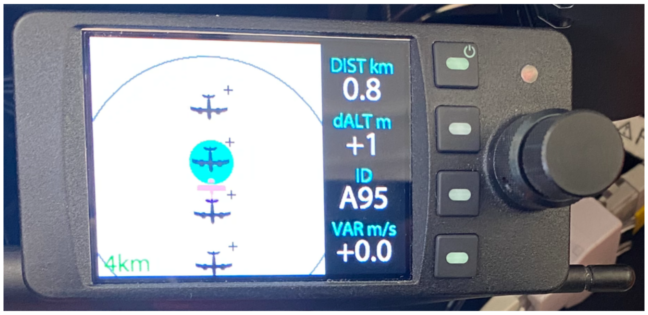


Figure 7: Four spoofed aircraft targets flying past the PowerFLARM Fusion receiver in a straight line.

Besides the R&D exception in order to test the authentic PowerFLARM device, we have further collaborated closely with the OpenSky Network in order to illustrate the spoofing problem to them. The spoofing of the ground receiver feeding to OpenSky was conducted as shortly as possible in order to illustrate the issue to them. Furthermore, we used coordinates that were outside the usual operating range of FLARM aircraft, in secluded areas and with no authentic callsigns of the spoofed aircraft in order to minimize the effects. After the demonstration, the OpenSky Network could delete the facetious flights from their database so that they would not affect their research datasets.

5.2 Setup

In the laboratory test, we set up one Raspberry Pi FLARM sender with Dragino HAT and a GPS antenna, one Raspberry Pi OGN ground receiver with RTLSDR RTL2832U DVB-T V3 dongle and its antenna. An ad-hoc version of SoftRF is run on the sender to read spoofing target settings from a text file. The OGN ground receiver decodes and feeds FLARM messages into the OpenSky Network.

As Figure 6 shows, a display is connected to PowerFLARM Fusion to show any target that is received by it. In order to further exclude the possibility of any interference, we set PowerFLARM Fusion's location in the middle of Black Sea by spoofing its GPS signal and conduct all experiments there. All spoofing activities happened inside the Faraday cage without measurable signal leakage.

5.3 Results

5.3.1 Spoofing of PowerFLARM Device. As Figure 7 shows, we set four spoofing targets as large jets flying in a straight line facing to PowerFLARM Fusion.

In Figure 8, we observed that these targets triggered the same alarms on the display as the ones in real collision scenarios. Therefore, we conclude that the spoofing system is able to cause alert in authentic FLARM system and create concrete distraction to pilots.

5.3.2 Spoofing of a FLARM Ground Receiver. Figure 9 demonstrates the effectiveness of our spoofing setup in targeting ground receivers

[4] "It is forbidden to intentionally feed artificially generated signals to the FLARM device, its GPS antenna or the external/internal GPS antenna connections, unless agreed with FLARM Technology in writing for limited R&D activities."

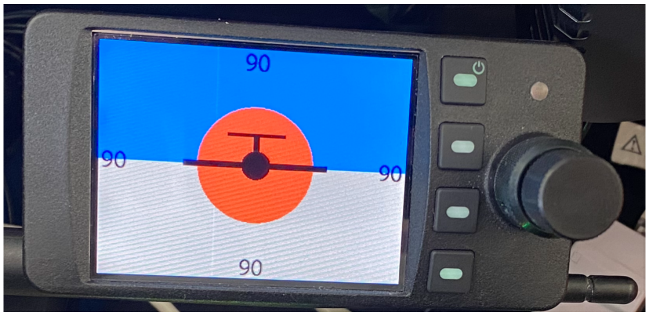
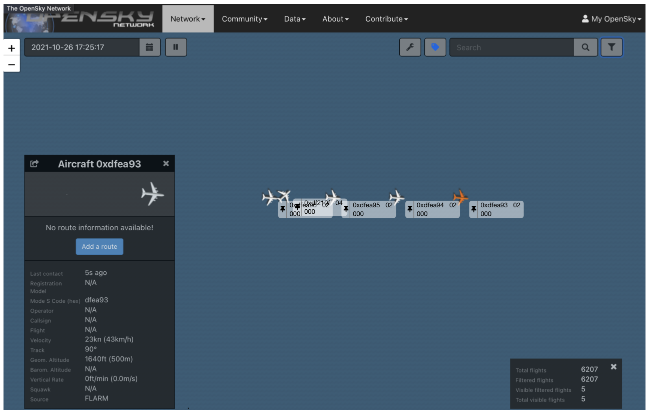


Figure 8: The PowerFLARM Fusion receiver strongly warns the pilot of a potential head-on collision.



2021-10-26 17:25:17

Aircraft 0xdfea93

No route information available!

Last contact	5s ago
Registration Model	N/A
Mode S Code (hex)	dfea93
Operator	N/A
Callsign	N/A
Flight	N/A
Velocity	23kn (43km/h)
Track	90°
Geom. Altitude	1640ft (500m)
Barom. Altitude	N/A
Vertical Rate	0ft/min (0.0m/s)
Squawk	N/A
Source	FLARM

Total flights	6207
Filtered flights	6207
Visible filtered flights	5
Total visible flights	5

Figure 9: Successful wireless spoofing of five aircraft into a FLARM ground receiver feeding into the OpenSky Network.

feeding to aviation data providers and their downstream applications. We feed four spoofing aircraft on four close trajectories in decimal coordinates: (43, 32.94), (43, 32.92), (43, 32.90), (43, 32.88). The ground station (which displays it duly on the OpenSky web interface) and PowerFLARM are set on trajectory (43, 33).

This demonstrates clearly that the spoofed data received is considered accurate and trustworthy. OpenSky confirmed to us that the data would be entered into the database and distributed further using the REST API. This is despite the checks for data integrity that OpenSky has implemented (for more information on these checks, see [37]).

5.4 Limits of the Spoofing System

To better understand the abilities of a spoofing adversary, we evaluated theoretical and practical limits of our system. To be more specific, we would like to show the maximum number of spoofing targets that can be produced by one single spoofing system. It is insightful because more spoofing targets it generates, more distractions the spoofing behavior can cause (such as a Denial of Service), which, in conclusion, could be more damaging or dangerous to the current FLARM system.

5.4.1 Theoretical Spoofing Limit. The theoretical sending limit is defined by sending rate of RF front end. In our spoofing system, Semtech SX1276-based HopeRF RFM95W module is used to emulate Nordic NRF905-based PTR8000 RF module which is used in PowerFLARM Fusion [23]. A GitHub project accessed from [23] also shows that the SX1276 module has a bit rate of 100 kbps on average, which is equal to 50 kbps effective bit rate for the payload after the Manchester operation.

Table 4: Spoofing Packet Field With Byte Length

Field	Byte Length
Preamble	1
Synchronization Word	1
Address	3
Payload	24
Checksum	2
Total	31

The second part related to theoretical limit is the packet length. Table 4 shows all the fields with corresponding byte length. In total, the packet has 31 bytes.

$$\begin{aligned} Theoretical\_RF\_Limit &= \frac{50\,kbps}{31\,bytes \times 8 \times 1\,packet\;per\;second} \\ &= 201.6\;targets\;per\;second \end{aligned} \quad (1)$$

Equation 1 shows us that the theoretical RF front end limit is 201.6 targets per second. Since one spoofing target should send at least one packet per second, an adversary can generate at most 201 spoofing targets using one single spoofing device. It is worth noticing that our spoofing sender can only send out packets from one frequency channel on 868.2 MHz.

5.4.2 Practical Spoofing Limit. In practice, the spoofing limit is defined by the number of actual FLARM messages that are received by the receiver in one second. This limit cannot go beyond the theoretical RF front end limit mentioned in previous section, but it can be lower. To measure this practical limit, we ran the sender in full capacity for 10 minutes, recorded all received messages in a file and took the average. It turned out that the practical spoofing limit is 66.6 messages per second, which is about one third of the theoretical limit. We speculate that the reason why the practical spoofing limit is far below the theoretical limit above is that FLARM receiver only accepts incoming messages during expected time slot, i.e. from 0.4s to 0.8s as mentioned in [17]. By multiplying this factor to the theoretical limit above, the estimated practical limit is 80.6 messages per second, which is much closer to our actual measurement 66.6 messages per second.

6 COUNTERMEASURES

We propose anomaly detection as a first-line defense and explore further based on currently available data. To achieve this, we first build a historical database using FLARM records from OpenSky Network. Then, we conduct two anomaly checks using this database to evaluate our proposal.

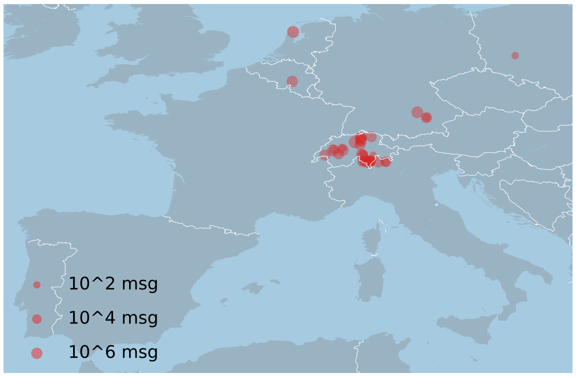


Figure 10: Map of FLARM receiver locations in the OpenSky Network. The circle size indicating the number of decoded FLARM messages it received in 2020. Overall, 70,284,107 messages were received and analyzed.

6.1 Historical Database Building

We establish a pipeline to retrieve, decode and decrypt historical FLARM data stored in OpenSky Network [38] and build a database that contains 70,284,107 FLARM messages throughout 2020.

As a first attempt at showcasing into the data, we find that, compared to Mode S or ADS-B messages, FLARM data still has significantly less coverage with the OpenSky Network. On average, there is only two to three FLARM messages per second through the whole year of 2020. One of the reasons for this is that the number of feeding receivers is limited.

Figure 10 is a map of receiver locations in OpenSky. The bubble size shows the number of decoded FLARM messages on this receiver in 2020. Overall, there are 35 OGN FLARM receivers (including one mobile receiver) feeding into the OpenSky Network.

Furthermore, all the receivers are located in Europe and most of them are distributed inside Switzerland. The limited number of receivers explains why there are only few FLARM messages per second through the year of 2020 on average. Hence, increasing the number of available FLARM receivers could help collect more information and gain more comprehensive understanding on the current FLARM system. In addition, the centralized distribution of receivers can introduce bias into the database analysis. All conclusions drawn from this database can only reflect the situation in Europe, mostly the situation in Switzerland.

6.2 Anomaly Detection

Based on the historical data, we evaluate two potential transparent mitigation measures against spoofing attacks on FLARM.

6.2.1 *Sanity Check.*

For the sanity check approach, we used the methodology proposed recently by Jansen et al. [16] for the ADS-B protocol. The authors built a system that analyzed ADS-B records for detecting different kinds of attacks. Since there are similarities on usage scenario of ADS-B and FLARM in aviation collision avoidance, it makes sense to design anomaly detection check on FLARM system using the same principle.

However, it is worth noticing that the boundary of those checks should be adapted to FLARM system and the physical characteristics of the aircraft using it. The sanity check verifies message content with respect to defined ranges shown in Table 5. Looping through all the records in our database, all the parameters are in range but vertical speed. There are 313,352 FLARM records with out-of-range vertical speed in total.

Table 5: FLARM sanity check parameters.

Category	Parameter	Range
Position	Latitude	-90 to 90 degree
	Longitude	-180 to 180 degree
	Altitude	-3 to 20,000 m
Movement	Speed	0 to 334 m/s
	Heading	-360 to 360 degree
	Vertical Speed	-50 to 50 m/s

After taking a closer look at those records, we concluded that it was more likely that there existed an implementation error in the FLARM decoder. This bug in the code could potentially cause misinterpretation on the value of vertical speed. It is noted that those out-of-range records were received from 32 different receivers with a coverage of 16 different types of aviation target. This fact further consolidated our reasoning because it is unlikely that all these differently located receivers are spoofed by several types of targets. More efforts will be needed to validate and debug this issue.

However, we argue that sanity checks could still be helpful by filtering out abnormal FLARM message before feeding it into FLARM platforms or databases.

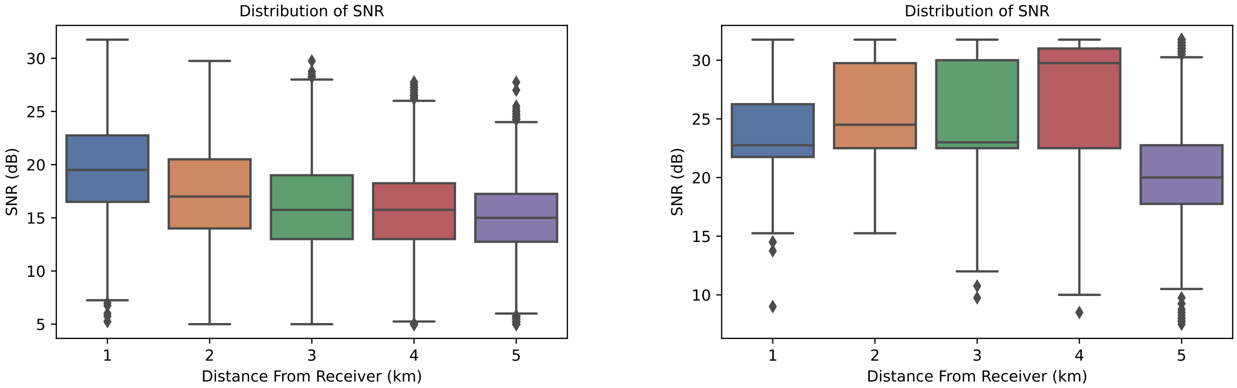
6.2.2 *Signal-to-Noise Ratio Check.*

As the OpenSky Network offers the signal-to-noise ratio (SNR) along with the delivered FLARM messages, we use it to improve the spoofing detection. Similar to other physical-layer charactersitics, SNR can serve as an anomaly check because it reveals whether the signal strength is within normal range from a certain distance. For example, a spoofing sender which is far away from a targeted FLARM device and receiver may have much weaker SNR than it should have while triggering the alarm on FLARM device. In addition, as the distance increases between the receiver and the sender, the median SNR should decrease.

We checked the distribution of SNR data which was collected during our spoofing experiment (Figure 11b) and compare it with the one which is derived from analyzing all signals received by a random receiver in OpenSky Network database during the whole year of 2020 (Figure 11a).

First, it is noticed that the signal strength in the spoofing test varies a lot from the one in a benign setting. For instance, the signal strength in the spoofing test is too strong while the receiver-sender distance should be 4 km. Second, the pattern of SNR median is also different, and thus, is skeptical. As the distance increases, the median of SNR does not decrease accordingly. Therefore, SNR check can be useful for detecting abnormal behaviors and identifying spoofing attacks.

While we acknowledge that because of safety reasons the experimental setup of the spoofing attack does not very closely resemble a potential real-world attack (e.g., with regards to proximity to the receiver), our experiments show that the approach is both feasible and practical.



(a) SNR distribution of FLARM signals received by a receiver in OpenSky Network. SNR decreases as the distance increases.

(b) SNR distribution of FLARM signals during a spoofing test. The median of SNR does not decrease as the distance increases

Figure 11: SNR distributions of normal and spoofed receiver, showing significant differences between the two scenarios.

7 DISCUSSION

As our evaluation shows, spoofing attacks are straightforwardly possible, with an attacker being able to completely control the target's inputs. This means that both aircraft and ground receivers can be manipulated at will, causing safety issues in the air and issues with downstream applications on the ground.

7.1 Practical Attack Impact

The impact of a FLARM spoofing attack is two-fold. First, it is obvious that the safety of any FLARM user in the air can be impacted, as collision avoidance is a crucial last-resort safety net. Whereas we illustrated a spoofing attack of non-existing targets, also called "ghost aircraft" attack in the literature [5], the complete control over the channel means that other attacks can be derived, e.g., the modification of the trajectories of other real aircraft [35].

This type of attack impact has been discussed at length for similar aviation technologies such as ADS-B and TCAS and examined for example by Smith et al. in a practical simulator setting [42]. While the pilots subjected to deliberately malfunctioning CAS did not experience immediate safety issues, both short- and long-term trust in the safety system is being reduced after a cyber attack. Furthermore, under less optimal circumstances (e.g., bad weather, other instruments malfunctioning), there can be a loss of situational awareness and added stress on the pilot may lead to potentially fatal mistakes. For typically less experienced private pilots of light aircraft and gliders, this effect may be compounded when dealing with unfamiliar collision avoidance display behavior [33].

Second, there is a direct impact on FLARM-processing aviation data providers, including OpenSky, OGN, Flightradar24 and their downstream applications. As discussed in Section 2.4, these are not mere harmless entertainment for enthusiast "planespotters" but serious business and non-profit endeavors powering global ecosystems. Incorrect historical data could falsify research results, incorrect live data could cause unnecessary investigations by security services (e.g., if an aircraft is falsely shown to veer off track). As a somewhat mitigating factor it can be argued that RF attacks are not the only, or even easiest, way to attack these data providers. Feeding FLARM or ADS-B data is typically weakly authenticated (if at all) and suffers from potential integrity issues on all levels from the aircraft to downstream applications [37].

7.2 Upgrade Time and Legacy Infrastructure

While not as bad as for large commercial aircraft systems, compared to consumer hardware, deployment times are still long also for FLARM/light aircraft. Hard- and software upgrades typically take years, even decades in many settings. Furthermore, even if FLARM itself were fully secure, the compatibility with the notoriously insecure ADS-B would open up similar issues to those discussed in this work for airborne receivers. Eventually, it is a trade-off between maximum interopability, which strongly benefits safety procedures, and strong, potentially complex or non-robust, security measures.

7.3 Transparent Countermeasures

As we have proposed in this work, one solution that does not change the already deployed hard- and software – and thus avoids the time lag issue – is to use transparent countermeasures, which use for example physical-layer characteristics to detect anomalies. For FLARM's big brother, ADS-B, there have been many such proposals in the literature, exploiting signal strength, time difference of arrival (TDOA), Doppler shift, angle of arrival and others [46]. While many problems here need to be solved, for example calibration and what to do with false positives, it is a promising low-cost approach. The proprietary aspect of FLARM could actually be of help here, as the company can iterate much more quickly than the global standards that are used in commercial aviation could.

7.4 Proprietary Technology and Cryptography

Despite its significant advantages on light aircraft, there is controversy concerning the design choices of the system with regards to encryption. FLARM was invented and is owned by a commercial entity, FLARM Technology, which has opted to not open-source the protocol. This represents a difference in approach to other aviation protocols such as Mode S, ADS-B or ACAS, which are laid down in public standards. Some stakeholders feel that a crucial safety technology should be as fully understood as possible [6].

The reverse engineering of the protocol and its keys has resulted in a reality where other software products offer compatibility with the FLARM protocol (providing additional utility to the wider FLARM ecosystem). From a purely information security perspective, there is now no obvious rationale for the encryption of the FLARM protocol. Beyond computer security, there may still be a legal and a business rationale of such an approach, as even weak encryption may fulfill e.g. a necessary duty of care or offer added legal protection in some countries. In the end, however, this state of affairs does not currently make FLARM any more secure against RF attacks than other existing aviation protocols, which have been designed without any cryptographic security measures in mind from the very beginning.

8 RELATED WORK

Through the years there has been an increased awareness of cyber threats in aviation, fueled by a growing body of research on attacks and countermeasures. A lot of previous work has been focused on the ADS-B system in particular, which has strong conceptional similarities with FLARM, and thus is of high relevance. We will discuss both attacks and potential countermeasures in this section.

8.1 Attacks

In 2010, Purton et al. analyzed critical elements in ADS-B system and assessed potential vulnerabilities in the transmission and computation information path [31]. While [31] provided technical solutions to a specific attack scenario, different attack vectors were well studied in [22] with recommended solutions that could be incorporated into ADS-B implementation plan. A practical, low-cost and moderately sophisticated attack against ADS-B was demonstrated in [5]. It managed to take a concrete step towards real attack against ADS-B compared to previous work with a focus on theoretical aspects of insecurity. Assuming a strong attacker using a multi-device setup, Moser et al. demonstrated the feasibility of attacking ADS-B communication even under strong physical-layer countermeasures such as multilateration [25]. Recent work has shown that such a strong attacker setting is increasingly realistic [24]. Finally, while there is almost a complete lack of cryptographic measures in currently deployed ATC protocols, other instances of weak avionics cryptography have surfaced in the literature [41].

8.2 Countermeasures

In addition to the large amount of research on attacks, defensive proposals were also discussed in the literature. For a full overview of the research into countermeasures, the reader is referred to [46, 47].

There are several research directions for security countermeasures, such as physical-layer security, anomaly detection and cryptography. Physical-layer security is particularly attractive for aviation legacy systems: attacks on ADS-B have been identified using several different primitives, including TDOA [3, 25, 45], Doppler shifts [15, 36], direction of arrival [53]. All options could be used with FLARM, although in some cases with a lower resolution due to the lower bit rate compared to ADS-B.

The authors in [45, 48] suggest machine learning on physical-layer features such as received signal strength and TDOA collected from ADS-B/SSR data to learn the space of states normally occupied by aircraft and detect abnormal states. However, there are often multiple explanations and causes for abnormal behavior, making anomaly detection a difficult engineering problem in practice.

Cryptography remains the most effective means to secure communication and is a popular research area in aviation protocols. Proposals for ADS-B include identity-based encryption [51], format-preserving encryption [1] and retro-active key publication [39]. Despite these proposals, many authors have also pointed out incompatibility with current systems, a major downside of cryptographic countermeasures in a slow-moving industry [39, 57].

Compared to previous work, our project differs in that we focused specifically on the FLARM protocol, which was neglected before in security research. It is also worth noticing that, despite the similarities in the attack scenarios, the usage of FLARM by light aircraft and drones may affect potential countermeasures (due to such targets' very different capabilities and behavior).

9 CONCLUSION

In this project, which is the first work to demonstrate practical attacks on FLARM, we developed a spoofing system for the FLARM protocol and analyzed historical FLARM records. We show that it is possible to generate authentic FLARM messages with entry-level knowledge using affordable hardware. A potential adversary is able to generate a considerable amount of spoofing targets through one spoofing system, possibly impacting a pilot's situational awareness.

As shown in our laboratory test, existing countermeasures including the ongoing use of proprietary encryption are not enough. Therefore, we proposed two anomaly detection approaches. While sanity checks can provide a first line of defense, we show that physical-layer countermeasures such as those based on SNR can be more useful for effectively detecting spoofing attacks.

Using historical FLARM data collected from the OpenSky Network, we found that the current data is not sufficient for monitoring or deeply analyzing the FLARM system situation. The number and coverage of feeding sensors are both limited, which may introduce bias. We urge that more research efforts are needed with respect to the shown security issues in the FLARM system.

ACKNOWLEDGMENTS

We thank both FLARM Technology and the OpenSky Network Association for their helpful support in this investigation.

REFERENCES

[1] Richard Agbeyibor, Jonathan Butts, Michael Grimaila, and Robert Mills. 2014. Evaluation of format-preserving encryption algorithms for critical infrastructure protection. In *Int. Conf. on Critical Infrastructure Protection*. 245–261.

[2] Anonymous Author. 2015. Glider Anti-Collision Radio Protocol Version 6. Retrieved February 8, 2022 from https://pastebin.com/YK2f8bfm

[3] Richard Baker and Ivan Martinovic. 2016. Secure Location Verification with a Mobile Receiver. In *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy* (Vienna, Austria) *(CPS-SPC '16)*. Association for Computing Machinery, New York, NY, USA, 35–46.

[4] Thomas Ivar Baumgartner and Urban Maeder. 2009. Trajectory prediction for low-cost collision avoidance systems. In *28th Digital Avionics Systems Conference*.

[5] Andrei Costin and Aurelien Francillon. 2012. Ghost in the Air(Traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices. In *BLACKHAT 2012, July 21-26, 2012, Las Vegas, NV, USA*, EURECOM (Ed.). Las Vegas.

[6] Sergio Elia. 2015. Petition against FLARM decision to encrypt the communication protocol. Retrieved February 8, 2022 from https://www.change.org/p/mr-urs-rothacher-flarm-chairman-petition-against-flarm-decision-to-encrypt-the-communication-protocol

[7] FLARM Technology. [n. d.]. On what frequency does FLARM operate? Retrieved February 8, 2022 from https://support.flarm.com/hc/en-us/articles/360015919933-On-what-frequency-does-FLARM-operate-

[8] FLARM Technology 2020. *PowerFLARM Portable Manual*. FLARM Technology, Switzerland. https://flarm.com/wp-content/uploads/man/PowerFLARM_Portable_Manual_EN.pdf.

[9] FLARM Technology. 2020. System Design and Interoperability. Retrieved February 8, 2022 from https://flarm.com/wp-content/uploads/man/FTD-062-System-Design-and-Interoperability-1.1.pdf/

[10] FLARM Technology 2021. *The Affordable Collision Avoidance Technology for General Aviation and UAV*. FLARM Technology, Switzerland. https://flarm.com/wp-content/uploads/man/FLARM-General-EN.pdf.

[11] FLARM Technology 2021. *PowerFLARM - See & Be Seen*. FLARM Technology, Switzerland. https://flarm.com/wp-content/uploads/man/PowerFLARM-EN.pdf.

[12] FLARM Technology 2021. *PowerFLARM Fusion User and Maintenance Manual*. FLARM Technology, Switzerland. https://flarm.com/wp-content/uploads/man/FTD-078-PowerFLARM-Fusion-User-and-Maintenance-Manual.pdf.

[13] FLARM Technology 2021. *Products*. FLARM Technology, Switzerland. https://flarm.com/products/.

[14] FLARM Technology. 2021. Traffic & Collision Warning. Retrieved February 8, 2022 from https://flarm.com/technology/traffic-collision-warning/

[15] Nirnimesh Ghose and Loukas Lazos. 2015. Verifying ADS-B navigation information through Doppler shift measurements. In *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*.

[16] Kai Jansen, Liang Niu, Nian Xue, Ivan Martinovic, and Christina Pöpper. 2021. Trust the Crowd: Wireless Witnessing to Detect Attacks on ADS-B-Based Air-Traffic Surveillance. In *28th Network and Distributed System Security Symposium*.

[17] Paweł Jałocha. 2017. Frequency Choice. Retrieved February 8, 2022 from https://github.com/UAVTracking/UAVTrackingProtocol/issues/4#issuecomment-336865347

[18] Claude Le Tallec and Boris Gravier. 2002. Device for improving the security of aircraft in visual flight regime. US Patent 6,438,492.

[19] FLARM Technology Ltd. 2017. End User License Agreement. Retrieved February 8, 2022 from https://flarm.com/wp-content/uploads/man/FTD-019-EULA-en.pdf

[20] FLARM Technology Ltd. 2021. PowerFLARM Fusion. Retrieved February 8, 2022 from https://flarm.com/products/powerflarm/powerflarm-fusion/

[21] Lockheed Martin. 2020. Swiss Army Chooses Lockheed Martin's Indago 3 UAS For Tactical Reconnaissance And Surveillance. Retrieved February 8, 2022 from https://news.lockheedmartin.com/swiss-army-chooses-lockheed-martin-indago3-uas-tactical-reconnaissance-surveillance

[22] Donald McCallie, Jonathan Butts, and Robert Mills. 2011. Security analysis of the ADS-B implementation in the next generation air transportation system. *International Journal of Critical Infrastructure Protection* 4 (08 2011), 78–87.

[23] mirakonta. 2021. SoftRF. Retrieved February 8, 2022 from https://github.com/mirakonta/NRF905-emulation-with-SX1276

[24] Daniel Moser. 2021. *Modern Attacker Models and Countermeasures in Wireless Communication Systems–The Case of Air Traffic Communication*. Ph. D. Dissertation. ETH Zurich.

[25] Daniel Moser, Patrick Leu, Aanjhan Ranganathan, Fabio Ricciato, and Srdjan Capkun. 2016. Investigation of Multi-device Location Spoofing Attacks on Air Traffic Control and Possible Countermeasures. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. 375–386.

[26] Roger M Needham and David J Wheeler. 1997. *Tea extensions*. Technical Report. Cambridge University.

[27] Open Glider Network. [n. d.]. OGN data usage rules. Retrieved February 8, 2022 from https://www.glidernet.org/ogn-data-usage/

[28] Open Glider Network. 2019. DVB-T Dongles. Retrieved February 8, 2022 from http://wiki.glidernet.org/dvbt-dongles

[29] Open Glider Network. 2019. Open Glider Network Wiki. Retrieved February 8, 2022 from http://wiki.glidernet.org/start

[30] Bureau of Transportation. 2020. *Number of Pilot-Reported Near Midair Collisions (NMAC) by Degree of Hazard*. Technical Report 800-853-1351. U.S. Department of Transportation. Retrieved February 8, 2022 from https://www.bts.gov/content/number-pilot-reported-near-midair-collisions-nmac-degree-hazard

[31] Leon Purton, Hussein Abbass, and Sameer Alam. 2010. Identification of ADS-B System Vulnerabilities and Threats. *ATRF 2010: 33rd Australasian Transport Research Forum* (2010).

[32] RTL-SDR. 2016. New RTL-SDR blog units now available. Retrieved February 8, 2022 from https://www.rtl-sdr.com/new-rtl-sdr-blog-units-now-available-in-store-hf-via-direct-sampling-software-switchable-bias-tee-less-noisespurs/

[33] Christoph G Santel, Paul Gerber, Simon Mehringskoetter, Verena Schochlow, Joachim Vogt, and Uwe Klingauf. 2014. How Glider Pilots Misread the FLARM Collision Alerting Display. *Aviation Psych. and Applied Human Factors* (2014).

[34] Harshad Sathaye, Domien Schepers, Aanjhan Ranganathan, and Guevara Noubir. 2019. Wireless attacks on aircraft instrument landing systems. In *28th USENIX Security Symposium (USENIX Security 19)*. 357–372.

[35] Matthias Schäfer, Vincent Lenders, and Ivan Martinovic. 2013. Experimental Analysis of Attacks on Next Generation Air Traffic Communication. In *Applied Cryptography and Network Security*. 253–271.

[36] Matthias Schäfer, Patrick Leu, Vincent Lenders, and Jens Schmitt. 2016. Secure Motion Verification Using the Doppler Effect. In *9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec '16)*. 135–145.

[37] Matthias Schäfer, Martin Strohmeicr, Matthew Smith, Markus Fuchs, Vincent Lenders, and Ivan Martinovic. 2018. OpenSky report 2018: assessing the integrity of crowdsourced mode S and ADS-B data. In *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. IEEE, 1–9.

[38] Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm. 2014. Bringing up OpenSky: A large-scale ADS-B sensor network for research. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. 83–94.

[39] Savio Sciancalepore and Roberto Di Pietro. 2018. SOS - Securing Open Skies. In *Security, Privacy, and Anonymity in Computation, Communication, and Storage*, Guojun Wang, Jinjun Chen, and Laurence T. Yang (Eds.). Springer International Publishing, Cham, 15–32.

[40] Skybrary. 2021. Mid-Air Collision. Retrieved February 8, 2022 from https://www.skybrary.aero/index.php/Mid-Air_Collision#Defences

[41] Matthew Smith, Daniel Moser, Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. 2017. Economy class crypto: Exploring weak cipher usage in avionic communications via ACARS. In *International Conference on Financial Cryptography and Data Security*. Springer, 285–301.

[42] Matthew Smith, Martin Strohmeier, Jonathan Harman, Vincent Lenders, and Ivan Martinovic. 2020. A View from the Cockpit: Exploring Pilot Reactions to Attacks on Avionic Systems. In *27th Network and Distributed System Security Symposium*.

[43] Matthew Smith, Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. 2020. Understanding Realistic Attacks on Airborne Collision Avoidance Systems. *arXiv preprint arXiv:2010.01034* (2020).

[44] Martin Strohmeier. 2020. Research Usage and Social Impact of Crowdsourced Air Traffic Data. In *Proceeding of The 8th OpenSky Symposium*, Vol. 59.

[45] Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. 2015. Intrusion Detection for Airborne Communication Using PHY-Layer Information. In *Detection of Intrusions and Malware, and Vulnerability Assessment*. 67–77.

[46] Martin Strohmeier, Ivan Martinovic, and Vincent Lenders. 2020. Securing the Air–Ground Link in Aviation. *International Series in Operations Research & Management Science* (2020), 131–154.

[47] Martin Strohmeier, Matthias Schäfer, Rui Pinheiro, Vincent Lenders, and Ivan Martinovic. 2017. On Perception and Reality in Wireless Air Traffic Communication Security. *IEEE Trans. on Intelligent Transportation Systems* 18, 6 (2017).

[48] Martin Strohmeier, Matt Smith, Matthias Schäfer, Vincent Lenders, and Ivan Martinovic. 2017. Crowdsourcing security for wireless air traffic communications. In *2017 9th International Conference on Cyber Conflict (CyCon)*. IEEE, 1–18.

[49] swissinfo. 2021. Five die in plane and glider crashes in eastern Switzerland. Retrieved February 8, 2022 from https://www.swissinfo.ch/eng/five-die-in-plane-and-glider-crashes-in-eastern-switzerland/46703756

[50] FLARM Technology. 2020. UK Department for Transport, CAA funding FLARM installations. Retrieved February 8, 2022 from https://flarm.com/uk-department-for-transport-caa-funding-flarm-installations/

[51] Gowri Thumbur, N.B. Gayathri, P. Vasudeva Reddy, Md. Zia Ur Rahman, and Aime' Lay-Ekuakille. 2019. Efficient Pairing-Free Identity-Based ADS-B Authentication Scheme With Batch Verification. *IEEE Trans. Aerospace Electron. Systems* 55, 5 (2019), 2473–2486.

[52] Giorgio Tresoldi. 2017. *A Flarm Receiver for the OpenSky Network*. Technical Report. ETH Zurich, Zurich.

[53] Wenyi Wang, Geng Chen, Renbiao Wu, Dan Lu, and Lu Wang. 2015. A low-complexity spoofing detection and suppression approach for ADS-B. In *2015 Integrated Communication, Navigation and Surveillance Conference (ICNS)*. 1–24.

[54] David J Wheeler and Roger M Needham. 1994. TEA, a tiny encryption algorithm. In *International workshop on fast software encryption*. Springer, 363–366.

[55] Dragino Wiki. 2017. Lora/GPS HAT. Retrieved February 8, 2022 from https://wiki.dragino.com/index.php?title=Lora/GPS_HAT

[56] Hiram Yaeger. 2008. FLARM PROTOCOL VERSION 4 (2008). Retrieved February 8, 2022 from http://www.dotmana.com/weblog/wp-content/uploads/FLARM-RADIO-PROTOCOL-VERSION-4-2008.txt

[57] Haomiao Yang, Qixian Zhou, Mingxuan Yao, Rongxing Lu, Hongwei Li, and Xiaosong Zhang. 2019. A Practical and Compatible Cryptographic Solution to ADS-B Security. *IEEE Internet of Things Journal* 6, 2 (April 2019), 3322–3334.

[58] Elias Yarrkov. 2010. Cryptanalysis of XXTEA. *IACR Cryptol. ePrint Arch.* 2010 (2010), 254.

[59] Linar Yusupov. 2021. SoftRF. Retrieved February 8, 2022 from https://github.com/lyusupov/SoftRF