OPENSKY: A SWISS ARMY KNIFE FOR AIR TRAFFIC SECURITY RESEARCH

Martin Strohmeier, Ivan Martinovic, University of Oxford, UK
Markus Fuchs, SeRo Systems, Germany
Matthias Schäfer, University of Kaiserslautern, Germany
Vincent Lenders, armasuisse, Switzerland

Abstract

The Automatic Dependent Surveillance - Broadcast (ADS-B) protocol is one of the key components of the next generation air transportation system. Since ADS-B will become mandatory by 2017 in the European airspace, it is crucial that aspects such as its security and privacy are promptly investigated by the research community. However, as expensive specialized equipment was previously necessary to collect real-world data on a large scale, such data has not been freely accessible until now. To enable researchers around the world to conduct experimental studies based on real air traffic data, we have created OpenSky, a participatory sensor network for air traffic research. In this paper, we describe the setup and capabilities of OpenSky, and detail some of the research into air traffic security that we have conducted using OpenSky.

1. Introduction

The Automatic Dependent Surveillance-Broadcast (ADS-B) protocol is one of the key components of the next generation air transportation system. As it becomes mandatory in the European airspace in 2017, it is crucial that the research community promptly investigates some open aspects including its security and privacy. However, large-scale real-world data was previously only accessible to few industrial and governmental groups because it required specialized and expensive equipment. To enable researchers around the world to conduct experimental studies based on real data, the Universities of Oxford and Kaiserslautern, in conjunction with armasuisse, have created OpenSky, a participatory sensor network for air traffic research.

OpenSky uses different types of off-the-shelf sensors distributed over Europe and is run by researchers and volunteers. The hardware is low-cost and connected over the Internet. Currently comprising 27 sensors, the sensor network covers an area of more than 1 million km^2. It captures more than 40% of Europe's commercial air traffic and offers access to more than 23 billion historical ADS-B messages. After two years of operating OpenSky, we present an overview of some of the research findings that could be of great interest to the aviation community.

Our paper makes the following contributions:

1. We describe the general setup of OpenSky, its features, our development efforts over the years, and some of the obstacles that we have encountered. OpenSky collects and processes a plethora of data. Besides the primary message types identification, position and velocity of ADS-B-equipped aircraft, information on emergencies, priority, capability, navigation accuracy category, and operational modes are collected. Furthermore, we store metadata for each message, including reception timestamps and the raw message itself, as well as physical layer information such as the received signal strength.

2. We summarize some of the academic research on security that has been done using OpenSky.

a) We have used the collected ADS-B data to localize aircraft independently from their own positional claims with low-cost hardware, using multilateration and our own localization techniques. Compared to the latter, our approach improves surveillance range, detection speed, and location accuracy for both legitimate aircraft and attackers in real-world environments. We also successfully conduct practical attacks on our multilateration systems and analyze the requirements of such attacks.

b) We have used OpenSky data to show that it is possible to securely verify sequences of

978-1-4799-8940-9/15/$31.00 ©2015 IEEE

location claims from aircraft. Our approach can securely verify any 2-D track with a minimum of three verifiers or any 3-D track with four verifiers. This secure track verification is lightweight, passive, does not require any time synchronization among the sensors, does not need to keep the location of the verifier's secret, and does not require specialized hardware.

c) We have used OpenSky to develop a transparent intrusion detection system (IDS) for air traffic control protocols based on physical layer information. By using an aircraft's received signal strength and message arrival times, we apply hypothesis testing and anomaly detection schemes, to accurately detect simulated attackers within a short time. Our IDS does not require cooperation by the aircraft, and can be implemented without changes to the existing protocols. Compared to secure localization or tracking approaches, it requires fewer sensors for successful intrusion detection with a low false alarm rate.

d) We have used the data to fingerprint transponder types and analyze the landscape of used transponders in Europe. We exploit some distinguishing features on the data link layer such as the behavior of the random generator that provides the backoff between ADS-B messages. The different transponders in commercial aircraft show very distinct behavior when we take a close look at the precise time periods between two subsequent messages (position, velocity, or call sign). We found various distinct classes and matched them with their aircraft types by cross-referencing with open source data.

e) We have used OpenSky to detect unusual events happening in the coverage area of the sensor network, providing potential insights into business movements and political events. By combining OpenSky's sensor data with publicly available databases about 24-bit ICAO identifiers, aircraft types and airlines, we are able to track various types of activity. By employing time series analysis, we can detect outliers and unusual events such as the World Economic Forum in Davos.

2. Overview of the ADS-B Protocol

The American Federal Aviation Administration (FAA) and its European pendant EUROCONTROL appointed ADS-B as the successor of traditional primary and secondary surveillance radar (SSR) technologies. ADS-B provides a new paradigm for air-traffic control (ATC) where aircraft and other surveillance objects use onboard satellite navigation systems such as GPS to retrieve their own position and velocity. This information is broadcasted twice per second by the transmitting subsystem ADS-B Out. The messages are received by ATC stations on the ground and by nearby aircraft, if equipped with ADS-B In. ADS-B offers many further fields such as ID, intent, urgency code, and uncertainty level.

Two ADS-B data link standards are currently in use, Universal Access Transceiver (UAT) and 1090 MHz Extended Squitter (1090ES). UAT has been created specifically for the use with aviation services such as ADS-B. It uses the 978MHz frequency and offers a bandwidth of 1Mbps. Since UAT requires fitting new hardware, as opposed to 1090ES, it is currently only used for general aviation in FAA-mandated airspaces. Commercial aircraft, on the other hand, employ SSR Mode S with Extended Squitter, a combination of ADS-B and traditional Mode S known as 1090ES (see Figure 1). This means that the ADS-B function can be integrated into traditional Mode S transponders. From here on, we focus on the commercially used 1090ES data link. The complete ADS-B specification can be found in the standards documents [2-4], succinct, higher-level descriptions of the protocol are given, e.g., in [5].

Vulnerabilities

Since there is no encryption of ADS-B message content, any passive adversary with a receiver listening on the 1090 MHz channel can eavesdrop on messages sent out by aircraft. While this may pose potential risks of privacy breaches (e.g., the possibility of tracking private planes), this is a by-product of ADS-B's open design and such honest-but-curious attackers are not considered further in this work. Similarly out of scope are non-selective jamming attacks, which are inherent to the wireless

medium and must be dealt with through conventional anti-jamming techniques.

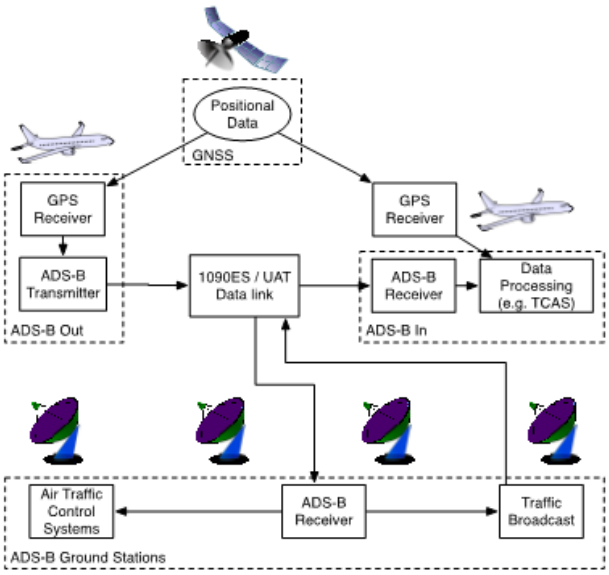


Figure 1. Simplified Schematic of ADS-B [1]

Outside these inherent vulnerabilities, an attacker that can actively interfere with ATC communication poses a much more severe threat to security than a passive one. With the introduction of software-defined radios (SDR) and receiver implementations freely available on the Internet, a somewhat knowledgeable attacker can exercise full control over the ADS-B communication channel. This means that the attacker is able to modify and inject ADS-B messages into ATC systems and manipulate radar screens, affecting the situational awareness of pilots and controllers. There are a multitude of such active attacks (for a more thorough overview see, e.g., [5-9]), which only require standard off-the-shelf hardware to execute, including:

- **Ghost Aircraft Injection / Flooding:** As demonstrated in [6, 7], ADS-B messages injected onto the 1090 MHz channel, claiming to be non-existing aircraft (so-called ghosts), are hard to detect. Especially under difficult weather conditions, injecting one or many different ghost aircraft may lead to serious distress.
- **Aircraft Disappearance:** Selectively jamming (as described in [8]) all ADS-B messages by a single aircraft would make the aircraft vanish from the ADS-B channel completely.
- **Aircraft Spoofing:** Every ADS-B message requires an identifier. This identifier can simply be replaced with an arbitrary one. Copying a known and trusted aircraft identifier may, for example, reduce the likelihood for alarms when an unexpected object is detected on the radar [7].
- **Virtual Trajectory Modification:** By selectively jamming an aircraft's messages and replacing them with modified location and heading data a discrepancy between the real aircraft position and the one received by ATC is created [7].

3. OpenSky

OpenSky (https://opensky-network.org) is a participatory sensor network with the goal to collect ADS-B messages in its sensing range for further analysis. The current sensor deployment in Central Europe and the coverage is shown in Figure 2. The sensors are provided by or distributed to volunteers and send their data to a central location where all data is decoded, evaluated and stored in a database. In contrast to other services that are available over the Internet, e.g. Flightradar24, the focus lies on the collection – and later retrieval – of as many raw messages (including metadata) as possible for future research purposes. The metadata comprises precise time stamps, physical layer data, and sensor location which can be used for deeper analysis compared to an aggregated and abstract view of the airspace as provided by other services. In the following section we firstly present the current setup of the system with a detailed description of its components.

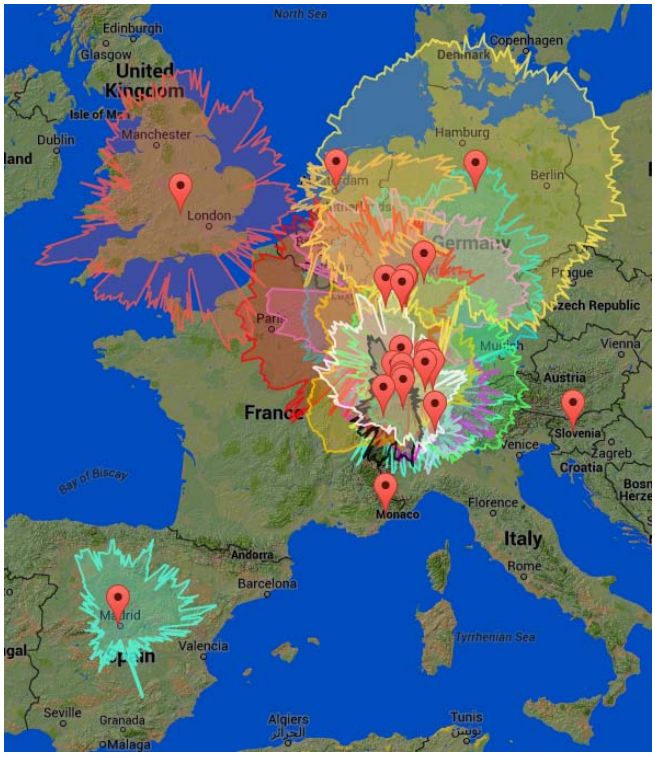


Figure 2. OpenSky's Sensor Deployment and the Resulting Coverage in Central Europe

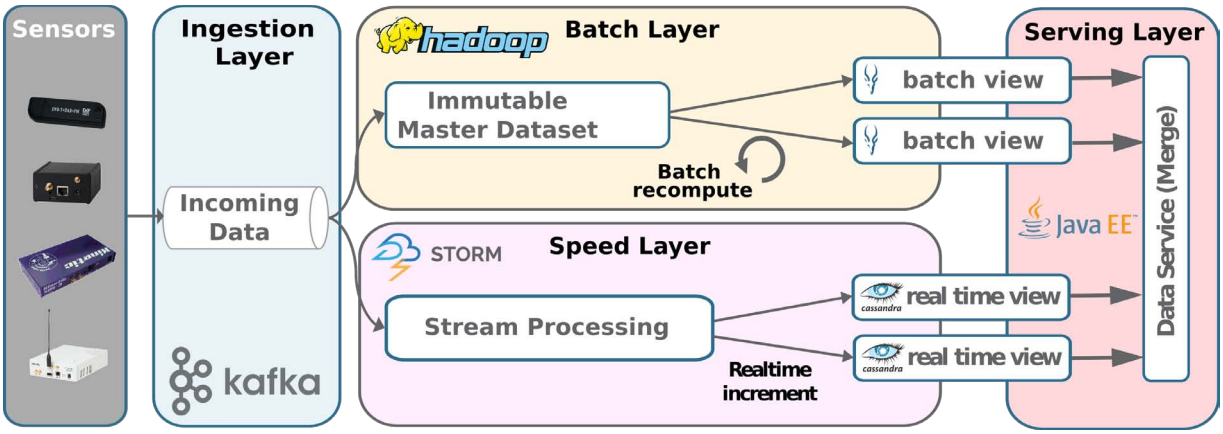


Figure 3. OpenSky Architecture Overview

Section 3.2 shows the problems we encountered with the original architecture based on a relational database management system (RDBMS) and its performance limitations. Finally, we give some numbers that reflect the network's capabilities.

3.1 Current Setup

In this section, we describe the current setup of OpenSky, from the supported sensors to its backend architecture.

Sensors

OpenSky currently supports four types of sensors: Kinetic Avionics' *SBS-3 station*, *Radarcape*, *dump1090*-based sensors and *ASTERIX CAT21*-based sensors. Most of these sensors are targeted for avionic enthusiasts and non-professional use. Their lack of certification contributes to affordable prices for our volunteers and research institutions. SBS-3 is a commercial all-in-one solution that supports the reception of aircraft data and voice communication. It

features an open protocol to stream data and remote configuration. The Radarcape is a standalone receiver based on the *Mode-S Beast* decoder board in conjunction with a Beaglebone single-board computer running a Linux operating system. The latter allows modifications and implementation of custom software, which is not possible with the SBS-3. Moreover, an integrated GPS receiver vastly improves the accuracy of time stamps for each message. Dump1090 is an ADS-B software decoder, which turns software-defined radios such as cheap USB sticks based on the RTL2832U chipset and originally used for DVB-T into ADS-B receivers. The ASTERIX CAT21 format is EUROCONTROL's standard format for exchanging aggregated ADS-B surveillance information.

Besides the raw ADS-B messages, sensors send additional information to the server. Radarcape and SBS-3 include a local rolling time stamp with an accuracy of 1 ns and 50 ns, respectively. In addition to that, the Radarcape can also send the received signal strength (RSS) for each message using a custom firmware we developed for OpenSky. Every sensor, independent of its type, is identified by a unique serial number so we can determine the receiver of each message including its position, which we infer from a central management database with information about all of the sensors deployed.

Backend Architecture

All data transmitted by the sensors need to be stored and processed. During three years of operation and a constantly growing load on the system, we recently implemented an architecture for big data systems, which is inspired by Nathan Marz' Lambda Architecture [11]: BigSense. It consists of four horizontally scalable layers (see Figure 3): an ingestion layer, a batch layer, a speed layer, and a serving layer.

The ingestion layer serves as a buffer for incoming messages. It is a scalable queuing system where data can be consumed any number of times until a pre-defined retention period has ended. The underlying software system is Apache Kafka [12], a distributed messaging and queuing system. It partitions data among multiple machines to allow heavy data streams, which might be larger than the capabilities of a single machine. Moreover, replication contributes to high availability in case of a hardware failure. Due to its distributed and flexible nature, the whole system can be extended on the fly without any downtime.

Incoming data streams are further consumed by the batch and speed layer individually, i.e. the same data pass both layers in parallel. Whereas the batch layer serves as an archive for the raw data and is optimized for large jobs that analyze the whole dataset, the speed layer offers real-time capability for low-latency tasks. In effect, real-time algorithms like multilateration can be run in the speed layer without interrupting complex analyses.

In the batch layer, messages from Kafka are periodically consumed and stored onto a Hadoop Cluster. Hadoop [13] is a framework that consists of a distributed file system, HDFS, and an implementation of the MapReduce algorithm for parallel data processing. Processed data are also called batch views and made available via an SQL-like query language. These views are updated periodically and might be re-generated from scratch, e.g., if there has been a mistake in the decoding algorithms. However, there are no guarantees on the delays of those batch jobs.

To compensate for the batch processing delay, which can be in the order of few hours, the speed layer implements a real-time stream processing system using Apache Storm [14]. It produces real-time views containing the same type of information as in the batch layer to provide live flight tracking data and support real-time capabilities in general.

As batch and real-time views span different time horizons the serving layer provides a unified interface for the end user. It transparently retrieves data from both, batch and real-time views, and returns a merged and consistent result. In this way, the internals of the architecture remain hidden for the end users.

3.2 Experience

When we started the project in 2012 data arriving at the server were processed by a processing unit which decoded each message according to the ADS-B standard and generated the respective database queries to store information in a central relational database management system (RDMS). We had chosen MySQL because of its maturity and community support with extensive documentation. Our database schema also followed the immutability paradigm and contained all raw messages and the

decoded information for aircraft position, velocity, identity and status. Starting with three sensors for a comparably small measurement, our objective was not building a large-scale sensor network. However, there was a demand for data in the research community. As the project gained attention, additional sensors were installed pushing more and more data into the database. The RDMS approach had several disadvantages which ultimately resulted in data loss and down times.

In the first place, the database induced a single-point-of-failure. Data loss has been caused once by a broken hard disk. Although the server was running a RAID system, which should handle single disk failures, an error in the RAID controller caused another disk to fail at the same time and the whole database was lost. Since the recovery of the database took some time and we were only doing periodic backups, this failure led to a considerable data loss. Unfortunately, this failure happened during an important long-term experiment, which had to be repeated afterwards.

Another disadvantage of the RDMS-architecture is its lack of scalability. As we did not expect OpenSky's growth in the beginning, the MySQL server was equipped with a 1TB hard drive, which soon reached its capacity limits. Migrating all data turned out to be a considerable bottleneck and we had to shut down the network for more than a day to maintain a consistent state. Another important aspect which is subject to scalability is the databases' throughput. When we started with only 170 inserts per second the database was far below its limits. With the growing number of sensors, insert rates increased to over several thousand per second, which could not be handled by the RDMS anymore.

3.3 Capabilities

By the end of May 2015, OpenSky's database contained more than 20 billion ADS-B messages that emerged from a total of 5.7 million flights. This corresponds to more than 40% of all commercial air traffic over Europe. At peak traffic hours we receive over 1000 ADS-B messages per second, which can be decoded and transformed into live flight information in less than 20ms (average). This not only includes the evaluation of position, identity and status reports but also includes running multilateration algorithms in the speed layer on all the ADS-B data to verify the reported positions. Although the algorithms are still under development, the low delay demonstrates OpenSky's real-time capabilities.

Besides these real-time capabilities, the system architecture enables large-scale analysis over all data that we collected so far. Basic operations like getting distinct aircraft or computing message rate statistics for the sensors for a time span of one year do not take longer than 10 minutes. Parallel computation on a cluster and well-documented library support (e.g. MapReduce) enable researchers answering virtually any research question within a reasonable amount of time. In the following section, we present some exemplary research use cases.

4. Security Research using OpenSky

Considering the extremely small likelihood of the introduction of cryptographic means to authenticate the content of ADS-B messages on the foreseeable future, other approaches to security and privacy of ATC protocols have to be considered. The sheer quantity of ADS-B data collected by OpenSky offers a perfect opportunity to develop and test such approaches, some of which are presented in this chapter.

4.1 Aircraft Location Verification

Besides securing the communication - and thus the location data - of ADS-B using cryptography, there are other potential approaches to ensure the integrity of air traffic management. The general idea of location verification is to double check the authenticity of location claims made by aircraft and other ADS-B participants. This is inherently different from the verification of the broadcast sources and messages. The baseline is to establish means to find the precise location of a sender, effectively offering some redundancy and thus the ability to double check any claims made independently from the aircraft.

Multilateration

Multilateration (MLAT) is a popular form of *co-operative independent surveillance* and has been successfully employed for decades in military and civil applications. If the precise distance between four or more known locations and an unidentified location can be established, it is a purely geometric task to find the unknown point. We can, for example,

use the received ADS-B signals, which travel at the speed of light to estimate the distance. Since we do not know the absolute time a message needed to travel from an airplane to a receiver, we have to employ the time difference of arrival (TDOA).

As MLAT is already an established concept within air traffic control systems, we use it to analyse the quality of OpenSky's data and furthermore as a baseline comparison to develop improved localization algorithms. This is required as MLAT has a number of drawbacks:

1. MLAT is highly susceptible to noisy environments and even small measurement errors outside a small area. An important quality metric for a deployment and its MLAT accuracy with respect to the target object's (the sending aircraft, in case of ADS-B) relative position is the *geometric dilution of precision*, or GDOP. It describes the effect of deployments on the relationship between the errors of the obtained time measurements and their resulting impact on the errors in the object's calculated position, or formally:

   $$\Delta\text{LocationEstimate} = \Delta\text{Measurements} \cdot \text{GDOP}$$

   GDOP is widely used in positioning systems such as GPS, where good ratings for this multiplier are commonly considered to be below 6, with 10 to be fair and everything over 20 to be of poor quality [15].

2. Theoretically, four or more sensors are sufficient to compute a position of an object in 3D space. However, it is very difficult to get the precise altitude of an aircraft when all the receivers are on the ground (i.e. in one plane) and do not provide sufficient elevation angle diversity. In that case, the *vertical* dilution of precision (VDOP) may be too large, so that only horizontal coordinates are calculated for aircraft surveillance and the altitude is obtained by other means.

3. While not a security challenge per se, MLAT systems are very expensive. ADS-B needs only one receiver for accurate wide area surveillance; MLAT requires every signal to be received by at least four stations with little noise. Geographical obstacles (e.g., mountain ranges, oceans) make it even more difficult to install a comprehensive wide area system at the desired service level.

4. A determined and resourceful attacker could spoof wireless signals such that using their TDOAs for localization would result in a position of the attacker's choice. This is shown in [16] for the case of GPS. While based on TDOAs, too, GPS is different as only a single receiver is attacked. The authors further discuss the case of spoofing a group of distributed GPS receivers similar to MLAT. They find that a system of multiple receivers severely restricts the attacker placement, each receiver making an attack exceedingly more difficult. As modern aircraft use GPS for navigation, the results of this GPS vulnerability study are also applicable to the aviation scenario (in the easiest scenario, an attacking sender is placed on board the aircraft).

Considering some of these drawbacks, and the fact that MLAT is currently considered the main security solution for unauthenticated ATC networks (e.g., by the ICAO [17], and in academic circles [8]), we argue that there is an urgent need for other TDOA-based approaches that improve on these problems and provide an immediate practical increase in security.

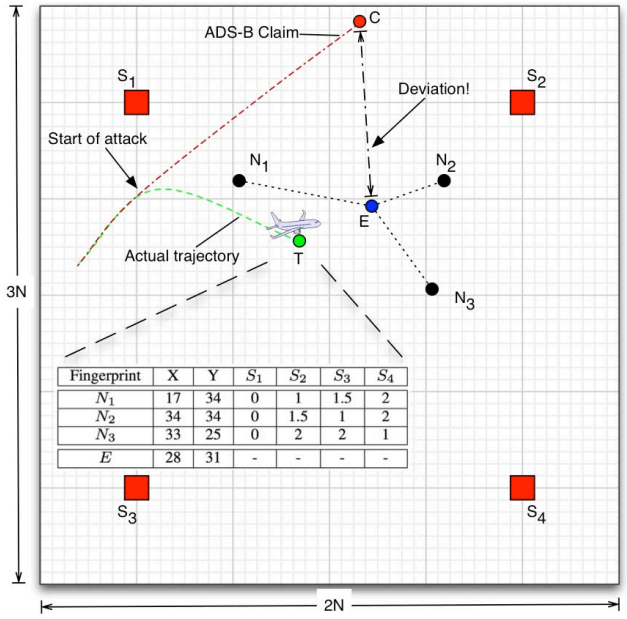
K-Nearest Neighbors

To counter the discussed disadvantages of MLAT, we used the classic k-Nearest Neighbors (k-NN) algorithm to develop an alternative solution to verify the location of aircraft based on the physical security properties of TDOA measurements [18]. In a first offline phase, we constructed a 2D grid over a typical flight altitude of 38,000 ft (ca. 11,582 m) with a size of 2 degrees longitude and 2 degrees latitudes. For each grid cell, a fingerprint vector F_{TDOA} is pre-generated, containing the TDOAs that would be expected from a sender at this position as received by the deployed OpenSky sensors in range.

In the subsequent online phase, new message data is analyzed and the location verified. The k-NN algorithm finds the closest points from our training grid that match the fingerprints of our test data as illustrated in Figure 4. More concretely, after setting the number of nearest neighbors to k, we match the received fingerprint $R = TDOA_1, \ldots, TDOA_n$ to the

saved grid fingerprint F based on their Euclidean distance

$$D_{R,F} = \sqrt{\left(\sum_{i=1}^{n} \left(R_{TDOA_i} - F_{TDOA_i}\right)^2\right)}.$$



Fingerprint	X	Y	S_1	S_2	S_3	S_4
N_1	17	34	0	1	1.5	2
N_2	34	34	0	1.5	1	2
N_3	33	25	0	2	2	1
E	28	31	-	-	-	-

Figure 4. Illustration of the k-NN Algorithm [18]

Evaluation

We first compare our location estimation method with the GPS-based ADS-B position claims of legitimate flight data, to ensure its accuracy. We use a data set of over 100,000 positional ADS-B messages from a two-week sample where every message has been seen by 5 sensors, providing us with the necessary TDOA measurements. All location claims are on the grid in terms of latitude and longitude, while the mean altitude is 11,148.8m (σ=687.59 m). Table 1 shows the location estimation quality using k-NN with different numbers of receivers (3, 4 and 5 sensors) over an area of 33,000 km^2 with k=5.

Table 1. Evaluation of the k-NN Algorithm

Error [m]	3 S	4 S	5 S	MLAT
Mean	311.8	147.3	122.3	199.5
Median	145.4	95.8	84.9	91.9
99%ile	2,469.6	983.7	870.6	1306.7

We also compared k-NN with a linearized MLAT algorithm using the same TDOA measurements from 5 sensors. The results show that with a 600 m^2 grid size, k-NN does 14.2% better than MLAT on mean errors, increasing to 41% for a 50m^2 grid size. Overall, we find that k-NN does better than MLAT on noisy TDOA measurements such as those we experienced in our real-world data. Especially the more outlier-sensitive metrics RMSE and mean improve with k-NN while MLAT generally shows good median results. Since k-NN does not suffer from dilution of precision, this is to be expected as the mean GDOP in our dataset is 24.35 (σ =8.06). Taking only "good" values below 10 into account, MLAT are bound to metrics improve vastly. However, doing this also decreases the number of usable messages by over 90%, reinforcing the fact that k-NN is useful in a much larger area. Of course, there is no reason why several different TDOA approaches could not easily complement each other.

4.2 Secure Track Verification

So called location verification schemes provide means to verify the validity of claimed positions and thus, may be used to detect attacks that are based on the injection of fake position reports. However, these schemes have high system requirements such as tight time synchronization, specialized hardware, or the use of additional protocols dedicated to the verification of claimed positions [19]. In addition to that, location verification protocols are not designed with respect to the inherent movement of aircraft. Considering the sluggish adoption process of new technologies in air traffic surveillance, we can conclude that these schemes do not constitute realistic countermeasures to the afore-mentioned security threats. Instead, passive solutions that do not require changes to the existing infrastructure are needed.

A promising candidate, which satisfies these requirements, is secure track verification [19]. The scheme has been designed based on a system model, which is fully compatible with the ADS-B standard. The scheme exploits the mobility of aircraft by considering *sequences* of location claims (i.e., track claims) instead of single locations. The constant movement of aircraft results in a continuous change in the distance between aircraft and receivers and thus, in different propagation delays (compare Figure 5). It has been demonstrated in [19] that these differences can be measured even with single low-cost receivers. The authors of [19] provided proof that a stationary attacker cannot spoof these

differences in propagation delay for more than three receivers at the same time. Thus, by comparing the measured with the expected difference in propagation delay at a minimum of four different locations, at least one of the receivers will detect a deviation. Simulations have shown that in this way, a set of trusted receivers are able to perfectly distinguish (i.e. without false positives or negatives) between fake and true location claims. Since each receiver can measure the required values independently from the others, there is no synchronization amongst receivers or transmitters needed.

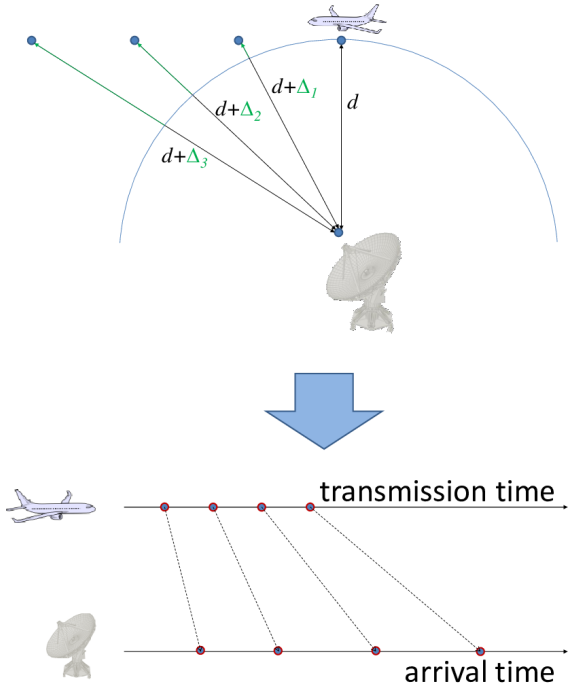


Figure 5. Principle of Secure Track Verification

The simplicity of the scheme proposed in [19] makes it particularly suitable for use in aviation. Cheap devices could be deployed and operated without interfering with existing infrastructure. In addition, the scheme leaves some room for improvement and variation. For instance, trusted aircraft equipped with ADS-B In could also perform the verification process and in that way, collaborative schemes could help to extend the range and improve the security. Preliminary results suggest that mobile verifiers (such as trusted aircraft) could help to further relax system requirements and accelerate the detection of erroneous or spoofed tracks.

4.3 Physical Layer Intrusion Detection

As argued in [1] and [20], we believe that given the current state of the ADS-B roll out, there is a strong need for transparent countermeasures as cryptographic means are not a feasible option in the medium term due to the requirements discussed above. Air traffic management as a critical infrastructure system has many characteristics of supervisory control and data acquisition (SCADA) systems. Cardenas et al. 21] note that threats on these systems need to be dealt with by defense-in-depth mechanisms and anomaly detection schemes. They argue an adversary may hide the specific exploits but cannot conceal their ulterior goals and intentions.

Indeed, there must be a noticeable adverse effect to the physical system (i.e., the management of air traffic), otherwise the attack may even be ignored, e.g., when somebody is simply relaying live ADS-B data. As such physical effects are achieved through injection of malicious data, which does not match the expected behavior, an anomaly detection system can help with the discovery of the attacker and provide the base for defense-in-depth mechanisms. A high rate of attack detection is at the heart of any such system where non-detection might cause disastrous consequences. However, in the real world low false positive rates are just as crucial. While they can normally be sorted out by using voice communication with the aircraft, constant nagging and false alarms can potentially have an adverse effect on overall system safety.

Attacker Model

We assume that the attacker injects a ghost aircraft, either collected at an earlier time and replayed, or created from scratch. In both cases, we assume a non-naive attacker who has sufficient knowledge to inject valid-looking messages that are well formed with reasonable content, withstanding a superficial check. This means the attacker creates correctly formatted ADS-B messages, covering the expected types (position, velocity, identification) in valid sequential orders and spacings according to the standard specification DO-260B. We also assume the attacker uses a legitimate ICAO address and reasonable flight parameters (e.g., believable altitude and speed) to create a valid-looking aircraft that cannot be distinguished from a real one using standard ATC procedures.

Based on these assumptions, we simulate three attackers that use different RSS patterns. *Attacker 1* uses a straight-forward constant sending strength, resulting in a Gaussian distribution due to the noisy nature of the channel. For *Attacker 2,* the RSS is a random variable X, within the limits of the assumed hardware (an off-the-shelf SDR). The strongest *Attacker 3* adjusts the sending strength in an attempt to be in line with the position the injected messages are representing to the attacked sensors, matching the attacked aircraft's behavior more closely.

Anomaly Detection Approach

We combine selected flight features (e.g., autocorrelation, Pearson correlation between distance and RSS, and signal features based on the number of antennas) in a one-class classification problem. One-class classifiers try to separate one class of data, the target data, from the rest of the feature space. Our target class is a well-sampled class of aircraft behavior based on collected RSS data. The outlier class is unknown and online target samples are used at the time of learning. The process creates an n-dimensional classifier, where n is the number of features. For new samples, this classifier decides if they fit into the expected space or if they are rejected (i.e., classified as an anomaly worth investigating).

The data for our anomaly detection example consisted of an OpenSky sample of 7,159 flights, each flight with 200 or more received messages. We test several different classifiers with 5-fold cross validation and the fraction of outliers in training set to zero (i.e., all training samples are accepted as legitimate). While the training sets are drawn from our collected sample of legitimate flights only, the separate test sets for each attacker have an added 2% of falsely-injected data (amounting to 143 flights) to be detected by the classifier. To verify our models and test our IDS, the RSS patterns of the attackers are simulated as described above.

Evaluation

Figure 6 illustrates the anomaly detection using a 2D Parzen classifier with 200 collected samples. Attacker 1 and 2 are entirely classified as anomaly here, while attacker 3 creates few false positives.

As shown in the full detection results in Table 2, our classifier can accurately detect all attackers 1 and 2 without false negatives and one single false positive (less than 0.01%), using a small RSS sample of 200 messages. At the standard rate of 5.4 ADS-B messages per second, this allows detection in less than 40 seconds, assuming no message loss. Even with a typical loss of 30% [20], this can be achieved in less than one minute.

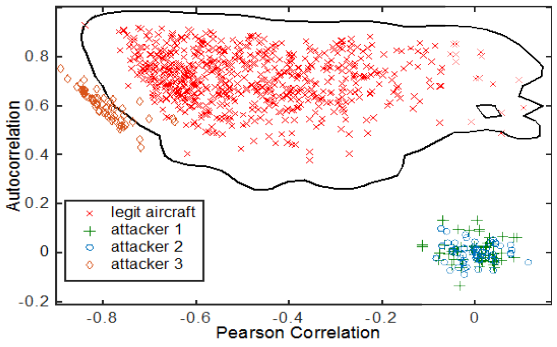


Figure 6. Anomaly Detection Example [22]

Table 2. Evaluation of the Anomaly Detection

	Detection rate
Legit flights (FPs)	<0.01%
Attacker 1	100%
Attacker 2	100%
Attacker 3	98.8%

4.4 Transponder Fingerprinting

As we analyze patterns of aircraft messages to identify anomalous and malicious activity, we exploit the fact that there are a multitude of different transponder types in ADS-B-equipped aircraft today. These transponders exhibit a number of different behaviors on the data link level as well as the physical layer, which can be utilized to validate incoming messages. Copying these characteristics is complex and difficult and restricts the space for creating malicious ADS-B message content.

Feature Selection

As an example, we look at the behavior of the random generator that generates the backoff between the periodically broadcasted ADS-B messages. The different transponders in commercial aircraft show very distinct behavior when we have a close look at the precise time periods between two subsequent messages (position, velocity, or call sign).

A standard implementation of the ADS-B protocol broadcasts three types of messages in a regular manner:

- **Position messages**: The aircraft broadcasts a message with its own position on average every 0.5 seconds. A random backoff mechanism is used to send the next message after a time interval drawn from [0.4, 0.6] seconds as defined in the specification.
- **Velocity messages**: The aircraft broadcasts a message with its current velocity on average every 0.5 seconds. Similar to the position messages, the random backoff interval is specified to be between 0.4 and 0.6 seconds.
- **Identification messages**: The aircraft broadcasts a message with its own ICAO 24-bit identifier on average every 5 seconds. Their backoff interval is randomly drawn between [4.8, 5.2] seconds.

One example of this backoff property between positional messages is shown in Figure 7. Hence, the only information needed about a message is its arrival time t_i in the form of an absolute or relative time stamp. Indeed, while ADS-B is not encrypted, exploiting such timing and inter-arrival information between various message types is naturally possible even with fully encrypted messages when the same backoff patterns are followed.

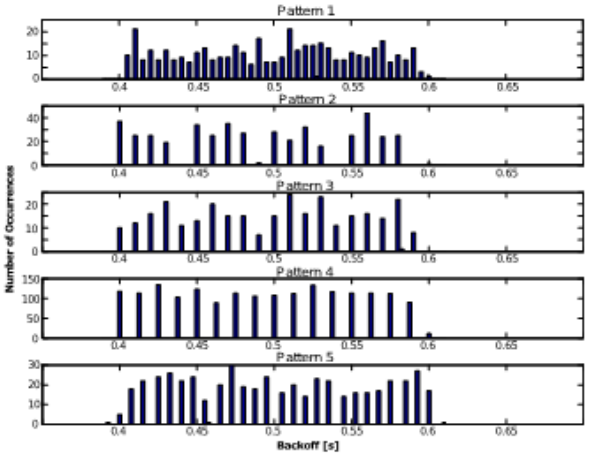


Figure 7. Illustration of Different Message Types

Evaluation

Based on all feature combinations, we discovered six main types of transponder behavior in our dataset. These classes are each exhibited by between 5% and 30% of the monitored aircraft. We further encountered numerous special cases with unique feature combinations at least in our data set, making these aircraft more identifiable and traceable, even when their correct ID is not broadcasted.

To verify the stability of our results over time, we trained all flights collected in our original dataset and looked for flights with the same ICAO identifier over the following week. With 1,287 returning flights, we found that the estimation of the transponder type stayed the same with 99.8% likelihood.

Applications

There are some potential applications for the fingerprinting of ADS-B transponders and/or aircraft in general which we discuss in the following.

a) *Intrusion detection:* One obvious use is for security. As described in Section 4.3, security approaches that do not require any modifications to the deployed ADS-B systems and protocols are severely needed. Such countermeasures can function alongside the current system without disrupting it and still provide a significant improvement in terms of security. Intrusion detection systems can use a multitude of learned features to tell apart normal from suspicious behavior. Fingerprints of any kind can provide such features, which an attacker has to adequately mimic when inserting false data onto the wireless channel. The more features are available to the IDS for monitoring, the more complex it becomes to mount an undetected attack.

b) *Privacy implications*: Flight privacy is an aircraft's ability to prevent unauthorized parties from tracking its current or past location. It helps preserve aircraft operators/owners interests, in terms of safety or sensitive business information which could be compromised if it were possible to e.g. track the movements of large companies' CEOs [23]. The current standard for flight privacy in the US is the so-called Block Aircraft Registration Request (BARR) mechanism. When private aircraft owners or operators request it, the FAA does not make the information about this aircraft's flights

public any more. They are subsequently also excluded them from web trackers such as Flightradar24. Naturally, the possibility of transponder fingerprinting has a number of privacy implications in relation to the aforementioned. While the ability to fingerprint specific aircraft or types of aircraft is not usually considered a problem for scheduled airliners, this could be different for private or business aircraft. ADS-B was developed to be open by design without concern for privacy mechanisms, offering anyone with a receiver the opportunity to follow the identity movements of broadcasting aircraft, facilitated by large internet services such as FlightRadar24 (or, indeed, OpenSky). However, there are concerns within the general aviation community. UAT, which we did not analyze here but have no reason to believe that it exhibits no such patterns, offers such a privacy mechanism. More concretely, an aircraft can generate a non-conflicting, random, temporary ID to avoid consistent tracking over time by third-party services. However, this generated ID can only be used under visual flight rules while not receiving ATC services, severely limiting its usefulness. Besides this, it has been shown that the DO-282B privacy solution has serious weaknesses, as the real ID of the aircraft and its random ID are correlated [23]. Yet, even when this could be fixed very easily from a security perspective, it would not close the fingerprinting-based privacy issue discussed in this work, as it is wholly unrelated.

c) *Business intelligence*: The dataset can provide an interesting picture of the current ADS-B transponder market. As mentioned above, this data is not necessarily easily available and prove interesting for competitors or market researchers (although there are paid options offering some of this data, e.g. the Aviation Week Intelligence Network). The data can for example easily be broken down into segments, showing the proliferation of certain transponder types or manufacturers in different countries or regions; alternatively, it would be possible to analyze trends over time. Public flight trackers do not offer this type of information and without the raw messages available in OpenSky the presented fingerprinting approach is infeasible.

4.5 Event Detection

We use OpenSky in an attempt to detect unusual events happening in the coverage area of the sensor network. In this section, we discuss one approach that can successfully detect large-scale events similar to the World Economic Forum in Davos.

By combining OpenSky's sensor data with publicly available databases about 24-bit ICAO identifiers, aircraft types and airlines, we try to track various types of activity. Since such data is not available directly from aircraft vendors and airlines, we conducted our mapping with open source data freely accessible on the Internet. Using the database available in the Planeplotter software (http://www.coaa.co.uk/planeplotter.htm), we could map the ICAO number received in the ADS-B messages of any given flight to its aircraft type as saved in the database. The ICAO number also provides the current airline of the corresponding aircraft, giving another interesting classification feature. We used a version of the SQLite database file database.sqb downloaded in November 2014, containing 120,149 rows of aircraft data.

For our following analysis, we used three distinct features, the number of distinct *business aircraft*, the number of distinct *military-related aircraft*, and the number of distinct *helicopters* seen per single day.

Figure 8 shows the three discussed features over a time span of 65 days from December 1, 2013 to February 5, 2014 collected by two OpenSky sensors located in Switzerland. We mapped the ICAO numbers of all flights seen by these sensors to the features named above based on the Planeplotter database. We corrected any existing mistakes, missing information and outdated entries by hand.

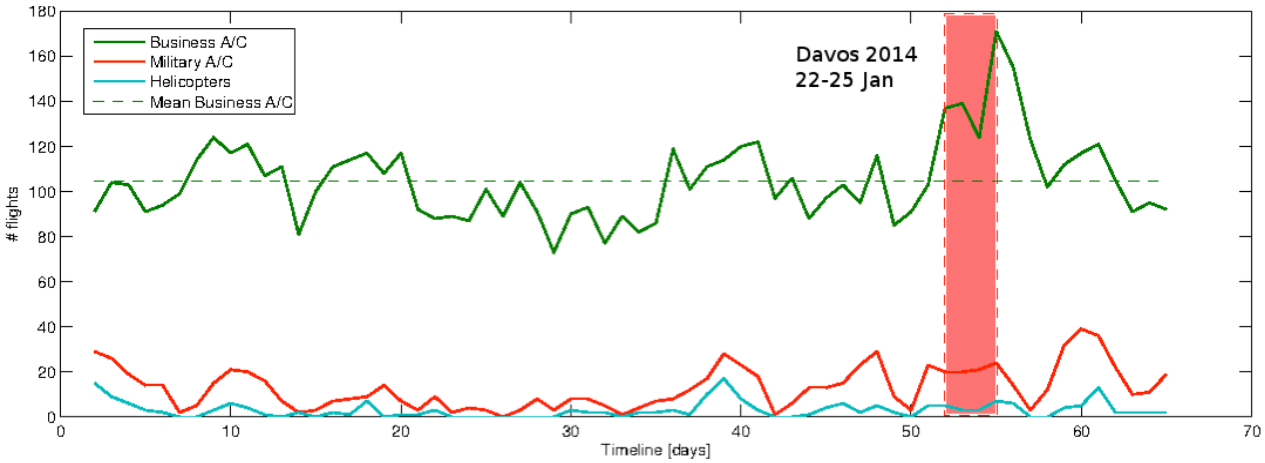


Figure 8. Example of a Time Series Analysis for the WEF 2014

By employing time series analysis, we can detect outliers as shown in the area highlighted in red. While there is little change in helicopters and military aircraft during the WEF compared to its run-up, we notice a distinct 75% increase from the mean business aircraft activity and a 45% increase over the highest previous peaks.

The described approach can provide potential insights into business movements and political events, even when the actually transmitted identifiers are pseudonymised. As long as the mapping to the type of aircraft is available, events attracting many smaller and/or business aircraft can reliably be detected.

There are some natural pitfalls in our analysis as presented here. First, the data quality and consistency needs to be ensured. Variations in the reception quality of sensors (caused, e.g., by construction or other disturbances) may distort the underlying data. Naturally, this type of open-source intelligence and anomaly detection also cannot tell us *what* event exactly is happening but it is a first step in a typical open source intelligence process.

5. Conclusion

In this paper, we describe the setup and capabilities of OpenSky, a participatory sensor network for air traffic research. OpenSky has collected and stored over 20 billion ADS-B messages to date, enabling researchers all over the world to conduct experimental studies based on real, large-scale data.

As ADS-B will become operational in many airspaces within the next few years, it is crucial that the research community thoroughly and promptly investigates its security and privacy. We have described such research in this paper, ranging from the secure localization and track verification of aircraft using low-cost ADS-B sensors, over intrusion detection based on physical layer characteristics and fingerprinting, to the detection of unusual events within the sensor range of OpenSky.

However, the research activities described in this paper constitute only a fraction of the possibilities. OpenSky is an attempt to provide researchers with realistic air traffic communication data of high quality, thus all OpenSky data is freely available on request. Several research groups are already working with OpenSky and all further participation on https://opensky-network.org is highly welcome.

6. References

[1] M. Strohmeier, V. Lenders, I. Martinovic, “On the Security of the Automatic Dependent Surveillance-Broadcast Protocol,” Communications Surveys & Tutorials, IEEE 17(2), 2015

[2] RTCA Inc., “Minimum Aviation System Performance Standards for Automatic Dependent Surveillance Broadcast (ADS-B),” DO-242A (including Change 1), Dec. 2006.

[3] ——, “Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance–Broadcast (ADS-B) and Traffic Information Services–Broadcast (TIS-B),” DO- 260B with Corrigendum 1, Dec. 2011.

[4] ——, "Minimum Operational Performance Standards for Universal Access Transceiver (UAT) Automatic Dependent Surveillance–Broadcast," DO-282B with Corrigendum 1, Dec. 2011.

[5] 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, no. 2, pp. 78–87, Aug. 2011.

[6] A. Costin and A. Francillon, "Ghost in the Air (Traffic): On Insecurity of ADS-B Protocol and Practical Attacks on ADS-B Devices," in Black Hat USA, 2012.

[7] M. Schäfer, V. Lenders, and I. Martinovic, "Experimental Analysis of Attacks on Next Generation Air Traffic Communication," in Applied Cryptography and Network Security. Springer, 2013, pp. 253–271.

[8] B. Kovell, B. Mellish, T. Newman, and O. Kajopaiye, "Comparative Analysis of ADS-B Verification Techniques." The University of Colorado, Boulder 4 (2012).

[9] K. Sampigethaya, R. Poovendran, S. Shetty, T. Davis, and C. Royalty, "Future E-Enabled Aircraft Communications and Security: The next 20 years and beyond," Proceedings of the IEEE, 99(11), 2040-2055, 2011

[10] M. Wilhelm and I. Martinovic, J. Schmitt, and V. Lenders, "Short Paper: Reactive Jamming in Wireless Networks: How Realistic Is the Threat?" Proceedings of the Fourth ACM Conference on Wireless Network Security, pp. 47–52, 2011.

[11] N. Marz and J. Warren, "Big Data: Principles and Best Practices of Scalable Realtime Data Systems," Manning Publications Co., 2012

[12] Apache Kafka, A Publish-Subscribe Messaging Rethought as a Distributed Commit Log, Accessed in June 2015, [Online], Available: http://kafka.apache.org/

[13] Apache. Hadoop Website, Accessed in June 2015, [Online], Available: http://hadoop.apache.org

[14] Apache Storm, Distributed and Fault-Tolerant Realtime Computation, Accessed in June 2015, [Online], Available: http://storm.apache.org

[15] M. Mosavi and H. Azami, "Applying Neural Network Ensembles for Clustering of GPS Satellites," International Journal of Geoinformatics, vol. 7, no. 3, 2011.

[16] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the Requirements for Successful GPS Spoofing Attacks," Proceedings of the 18th ACM Conference on Computer and Communications Security. ACM, 2011.

[17] ICAO, "Guidance Material: Security Issues Associated with ADS-B," Tech. Rep., 2014

[18] M. Strohmeier, V. Lenders and I. Martinovic, "Lightweight Location Verification in Air Traffic Surveillance Networks", Proceedings of the 1st ACM Workshop on Cyber-Physical System Security (CPSS '15). Pages 49–60. ACM. April, 2015.

[19] M. Schäfer, V. Lenders, and J. Schmitt, "Secure Track Verification," Proceedings of the IEEE Symposium on Security and Privacy (S&P). 2015.

[20] M. Strohmeier, M. Schäfer, V. Lenders, and I. Martinovic, "Realities and Challenges of NextGen Air Traffic Management: The Case of ADS-B," Communications Magazine, IEEE, vol. 52, no. 5, May 2014.

[21] A. Cardenas, S. Amin, Z. Lin, Y. Huang, C. Huang, and S. Sastry "Attacks Against Process Control Systems: Risk Assessment, Detection, and Response," Proceedings of the 6th ACM symposium on Information, Computer and Communications Security (ASIACCS), pages 355-366. ACM, 2011.

[22] M. Strohmeier, V. Lenders and I. Martinovic, "Intrusion Detection for Airborne Communication using PHY-Layer Information", SIG SIDAR Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA), Milan, Italy, July 2015.

[23] K. Sampigethaya, S. Taylor, and R. Poovendran. "Flight Privacy in the NextGen: Challenges and Opportunities," Integrated Communications, Navigation and Surveillance Conference (ICNS), pages 1–15. IEEE, 2013.