JYU DISSERTATIONS 546

Syed Ibrahim Khandker

Positioning Services in Different Wireless Networks

A Development and Security Perspective

UNIVERSITY OF JYVÄSKYLÄ
FACULTY OF INFORMATION TECHNOLOGY

JYU DISSERTATIONS 546

Syed Ibrahim Khandker

Positioning Services in Different Wireless Networks

A Development and Security Perspective

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi Agoran auditoriossa 2
elokuun 17. päivänä 2022 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, auditorium 2, on August 17, 2022, at 12 o'clock.

JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2022

Editors
Marja-Leena Rantalainen
Faculty of Information Technology, University of Jyväskylä
Päivi Vuorio
Open Science Centre, University of Jyväskylä

ISBN 978-951-39-9360-3 (PDF)
URN:ISBN:978-951-39-9360-3
ISSN 2489-9003

Permanent link to this publication: http://urn.fi/URN:ISBN:978-951-39-9360-3

ABSTRACT

Khandker, Syed Ibrahim
Positioning Services in Different Wireless Networks: A Development and Security Perspective
Jyväskylä: University of Jyväskylä, 2022, 80 p. (+included articles)
(JYU Dissertations
ISSN 2489-9003; 546)
ISBN 978-951-39-9360-3 (PDF)

In this Internet of things era, accurate positioning is crucial for all devices. Fingerprinting is a prevalent positioning technique in a Wi-Fi network that uses received signal strength (RSS) from the available access point to provide satisfactory user positioning in the indoor environment, where global navigation satellite systems perform poorly. However, the fingerprinting positioning system (FPS) faces many challenges. This thesis proposes development of the FPS by employing new technology, such as device-to-device communication, and new ideas such as error prediction and RSS quantization. The aviation and maritime sector uses automatic dependent surveillance-broadcast (ADS-B) and the automatic identification system (AIS) to share aircraft and ship locations with other entities, respectively. This thesis identifies some severe security vulnerabilities of both technologies. An aviation and maritime pentesting platform has been set up to analyze the security of these two technologies thoroughly. The platform contains heterogeneous ADS-B and AIS devices such as mobile cockpit information systems, commercial transponders, various ADS-B and AIS receivers, and autopilot for drones. Various radio frequency-link-based attacks were implemented using software-defined radio. Along with existing attack ideas such as spoofing, denial of service, and jamming, this thesis demonstrates some novel attack concepts in ADS-B and AIS contexts, such as the coordinated attack, protocol fuzzing, and false distress signal. The results show that both ADS-B and the AIS are exposed to cyberattacks, leading to a system crash in the worst case. It has been found that most of the ADS-B setups correct up to 2-bit errors, whereas AIS configurations do not correct any errors. The investigation in this thesis demonstrates that an RSS and distance-to-emitter relation could distinguish real and fake ADS-B signals. The consistency of the results for a comprehensive range of hardware-software configurations indicates the reliability of the approach and test results.

Keywords: Fingerprinting, ADS-B, AIS, 1090ES, UAT978, Positioning, Wireless Networks

TIIVISTELMÄ (ABSTRACT IN FINNISH)

Khandker, Syed Ibrahim
Paikannuspalvelut erilaisissa langattomissa verkoissa: Näkökulmia kehitykseen ja turvallisuuteen
Jyväskylä: University of Jyväskylä, 2022, 80 s. (+artikkelit)
(JYU Dissertations
ISSN 2489-9003; 546)
ISBN 978-951-39-9360-3 (PDF)

Laitteiden tarkka sijaintitieto on esineiden internetissä korvaamatonta. Sisätiloissa laitteiden sijaintia havainnoidaan langattomassa verkossa sormenjälkipaikannuksen (Fingerprinting Positioning System, FPS) avulla, sillä satelliittipohjaiset ratkaisut kuten GPS toimii sisätiloissa huonosti. Sormenjälkipaikannus käyttää signaalin voimakkuutta (RSS) laitteen ja vastaanottimen välillä riittävän tarkan sijaintitiedon saamiseksi. Sormenjälkipaikannuksessa on kuitenkin ongelmansa. Tässä väitöskirjassa ehdotetaan uudenlaista tekniikkaa FPS:n parantamiseksi kuten laitteiden välistä kommunikaatiota, virheenkorjausta ja RSS:n kvantisointia. Ilmailu- ja merenkulkutoimialoilla käytetään ADS-B (Automatic Dependent Surveillance-Broadcast) ja AIS (Automatic Identification System) protokollia paikannusdatan jakamiseen liikennöinnin osapuolten kesken.

Tässä väitöskirjassa esitetään useita vakavia haavoittuvuuksia molemmista protokollista. Väitöskirjassa esitellään myös kyseisten protokollien penetraatiotestaukseen tarkoitettu järjestelmä. Järjestelmä sisältää useita ABS-B ja AIS laitteita kuten ohjaamoinformaatiojärjestelmiä, kaupallisia transpondereita, useita vastaanottimia, sekä autopilottijärjestelmä drooneille. Ohjelmistoradioilla toteutettiin useita radiosignaalin välityksellä tapahtuvia hyökkäyksiä kyseisiä laitteita kohtaan. Tavanomaisten palvelunestohyökkäysten, häirinnän ja väärentämisen lisäksi toteutettiin myös uudenlaisia hyökkäyksiä kuten koordinoitu hyökkäys ja hätäsignaalin väärentäminen.

Tuloksista havaitaan, että kyseiset protokollat ovat haavoittuvaisia kyberhyökkäyksille ja pahimmassa tapauksessa niitä käyttävät järjestelmät myös kaatuvat. Tutkimuksessa havaittiin, että useimmat ADS-B:tä käyttävät järjestelmät kykenevät korjaamaan signaalivirheitä kahteen bittiin asti, mutta AIS:ää käyttävät järjestelmät eivät korjanneet virheitä ollenkaan. Tässä väitöskirjassa myös esitellään kuinka RSS:ää ja etäisyyttä voidaan hyödyntää valheellisten signaalien havainnointiin. Tuloksien johdonmukaisuus laajalla laiteskaalalla osoittaa käytetyn menettelytavan sekä tuloksien luotettavuuden.

Avainsanat: Sormenjälkipaikannus, ADS-B, AIS, 1090ES, UAT978, Paikannus, Langattomat verkot

Author

Syed Ibrahim Khandker
Faculty of Information Technology
University of Jyväskylä
Finland

Supervisors

Professor Timo Hämäläinen
Faculty of Information Technology
University of Jyväskylä
Finland

Professor Ismo Hakala
Information Technology Unit
Kokkola University Consortium Chydenius
Finland

Professor Tapani Ristaniemi
Faculty of Information Technology
University of Jyväskylä
Finland

Doctor Andrei Costin
Faculty of Information Technology
University of Jyväskylä
Finland

Reviewers

Professor Ivan Martinovic
Department of Computer Science
University of Oxford
UK

Assistant Professor Miguel Pardal
Instituto Superior Técnico
Universidade de Lisboa
Portugal

Opponent

Professor Heidi Kuusniemi
School of Technology and Innovations
University of Vaasa
Finland

ACKNOWLEDGEMENTS

First of all, I would like to thank the Almighty for His blessings upon my family and giving me the strength, courage, and wisdom to complete this dissertation.

I would like to express my deepest gratitude to Professor Timo Hämäläinen, Professor Ismo Hakala, Professor Tapani Ristaniemi, and Dr. Anderi Costin for their guidance, encouragement, and valuable suggestions, which have been absolutely necessary throughout my doctoral research. I want to extend sincere thanks to my co-authors, Riaz Mondal, Joaquín Torres Sospedra, and Hannu Turtiainen, for their cooperative support. I am grateful to the reviewers of my dissertation, Professor Ivan Martinovic and Assistant Professor Miguel Pardal, for their constructive comments. Special thanks to Professor Heidi Kuusniemi for being my opponent in the public defense.

I would like to thank the Faculty of Information Technology, University of Jyväskylä, for providing financial support. I also appreciate the Ellen and Arturi Nyyssösen foundation and Riitta and Jorma J. Takanen foundation for their financial support of my research.

I humbly remember Khaled Nuri, Dr. Sher E Khoda, and Osima Edson, who acted as elder brothers in my Finnish life. Their helping hand made it easy for me to settle down in a foreign land and focus on my study. I am grateful to Belayet Robin (S21RB) for introducing me to the radio frequency world in my school life, which I am still chasing.

My warmest gratitude goes to my uncles and aunts. After my father's early death, it would not have been possible for me to come to this stage without their guidance and support. I am also very grateful to all my brothers, especially Babu Bhaiya, for his suggestion and financial support for my higher studies.

I want to thank my wife, Sadia Khandker, for taking care of the family, encouraging me when I am frustrated, and tolerating my undisciplined activities for the last four years. Lots of love to my very special little daughter Nawal, my rabbit!

Finally, I would like to dedicate this dissertation to my parents, who have provided me with a beautiful childhood, unconditional love, and the best possible support in all ups and downs of my life.

Jyväskylä 17.08.2022
Syed Ibrahim Khandker

LIST OF ACRONYMS

3D	Three-Dimensional
3GPP	3rd-Generation Partnership Project
ADS-B	Automatic Dependent Surveillance–Broadcast
AFL	American Fuzzy Lop
AIS	Automatic Identification System
AP	Access Point
ATC	Air Traffic Control
CA	Certification Authority
CDF	Cumulative Distribution Function
CPA	Closest Point of Approach
CPFSK	Continuous-Phase Frequency-Shift Keying
CRC	Cyclic Redundancy Check
CSTDMA	Carrier Sense Time Division Multiple Access
D2D	Device-to-Device
DoS	Denial of Service
EASA	European Aviation Safety Agency
EFB	Electronic Flight Bag
eNB	Evolved Node B
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FPS	Fingerprinting Positioning System
GA	General Aviation
GMSK	Gaussian Minimum Shift Keying
GNSS	Global Navigation Satellite System
GRC	GNU Radio Companion
HPA	High-Power Attack
ICAO	International Civil Aviation Organization
IMO	International Maritime Organization
IoT	Internet of Things
IQ	In-Phase and Quadrature
KNN	K-Nearest Neighbor
LGP	Log-Gaussian Probability

LPA	Low-Power Attack
MCIS	Mobile Cockpit Information System
MMSI	Maritime Mobile Service Identity
MOB	Man Overboard
MPA	Medium-Power Attack
NA	Not Applicable
NMA	National Maritime Authorities
NRZI	Non-Return-to-Zero Inverted
NRZS	Non-Return-to-Zero Space
PE	Positioning Error
PGR	Plane Gadget Radar
PPM	Pulse Position Modulation
PSR	Primary Surveillance Radar
RF	Radio Frequency
RFID	Radio Frequency Identification
RSS	Received Signal Strength
SDR	Software-Defined Radio
SOTDMA	Self-Organized Time Division Multiple Access
SSR	Secondary Surveillance Radar
TCAS	Traffic Collision Avoidance System
TCPA	Time to the Closest Point of Approach
TDMA	Time Division Multiple Access
TFP	Test Fingerprints
TOA	Time of Arrival
TUT	Tampere University of Technology
UAT	Universal Access Transceivers
UE	User Equipment
UJI	University of Jaume I
USRP	Universal Software Radio Peripheral
VTS	Vessel Traffic Service
WC	Weighted Centroid

LIST OF FIGURES

LIST OF TABLES

CONTENTS

LIST OF INCLUDED ARTICLES

PI S Khandker, R Mondal, T Ristaniemi. Device diversity effects on RF fingerprinting based 3D positioning system. *8th International Conference on Localization and GNSS (ICL-GNSS), Guimaraes, Portugal*, 2018.

PII S Khandker, J Torres-Sospedra, T Ristaniemi. Improving RF fingerprinting methods by means of D2D communication protocol. *Electronics*, 8 (1), 97, 2019.

PIII S Khandker, R Mondal, T Ristaniemi. Positioning error prediction and training data evaluation in RF fingerprinting method. *10th International Conference on Indoor Positioning and Indoor Navigation (IPIN), Pisa, Italy*, 2019.

PIV S Khandker, J Torres-Sospedra, T Ristaniemi. Analysis of received signal strength quantization in fingerprinting localization. *Sensors*, 20 (11), 3203, 2020.

PV S Khandker, H Turtiainen, A Costin, T Hämäläinen. Cybersecurity attacks on software logic and error handling within ADS-B implementations: Systematic testing of resilience and countermeasures. *IEEE Transactions on Aerospace and Electronic Systems*, Early access article.

PVI H Turtiainen, A Costin, S Khandker, T Hämäläinen. GDL90fuzz: Fuzzing "GDL-90 data interface specification" within aviation software and avionics devices–a cybersecurity pentesting perspective. *IEEE Access*, 10, 21554-21562, 2022.

PVII S Khandker, H Turtiainen, A Costin, T Hämäläinen. On the (in)security of 1090ES and UAT978 mobile cockpit information systems – an attacker perspective on the availability of ADS-B safety- and mission-critical systems. *IEEE Access*, 10, 37718-37730, 2022.

PVIII S Khandker, H Turtiainen, A Costin, T Hämäläinen. Cybersecurity attacks on software logic and error handling within AIS implementations: A systematic testing of resilience. *IEEE Access*, 10, 29493-29505, 2022.

1 INTRODUCTION

1.1 Background and Motivation

Localization is a fundamental human need. With the help of the star Polaris, the human race navigated from one place to another for several thousand years. In the modern era, a variety of localization techniques have emerged. Of these, radio frequency (RF)-based positioning systems are widely used. The main advantage of using RF as the means of positioning is that it can be operated remotely with extremely fast speed. Therefore, global navigation satellite system (GNSS)-based services, such as GPS, Galileo, BeiDou, and GLONASS, are very popular. Most safety- and mission-critical systems, such as aviation systems, maritime systems, terrestrial vehicles, and ordinary smartphones, use GNSS-based positioning services. The aim, mission, and strategy of different positioning services depend on the nature and demands of the targeted operation. For example, if a ship, aircraft, or vehicle in an outdoor environment wants to share its location, it must first obtain its position coordinates through a GNSS service. It then shares the information with others using media, such as radio links. In this case, the integrity of the positioning service depends mainly on the security of the radio link. However, if people in an indoor environment want to determine their position using a GNSS system the results might be poor because the satellite signal's blockage degrades the positioning performance. Therefore, the challenge of developing a global-scale indoor localization system remains unsolved. According to Kurschl et al. (2008), a single positioning system cannot meet all the requirements set by the industry. Consequently, continuous research on a variety of positioning technologies is necessary. This thesis investigates positioning services in the indoor environment, aviation, and maritime sectors. These three domains have been chosen because of their importance in the near future. For example, now people are staying in indoor premises more than anytime before, and this trend is gradually increasing. So developing a faultless indoor positioning system is very important. Due to economic growth, international trade, tourism, and affordable ticket price, air and maritime transportation are expanding. An efficient localiza-

tion and surveillance system for these two transportation modes is very crucial because of the safety of passengers or goods.

The obstruction of satellite signals substantially hampers satellite-based localization in indoor environments. Therefore, alternative indoor localization technologies have been developed using optic (Bergen et al., 2015), ultrasound (Perez et al., 2009), dead reckoning (Jimenez et al., 2009), radio frequency identification (RFID) (Liu et al., 2019), ultra-wideband (Ruiz and Granja, 2017), visible light (De-La-Llana-Calvo et al., 2020), and Bluetooth technology (Ji et al., 2015). Most of these are based on communication technologies and require additional hardware to function properly. However, the demand for an Internet connection at any time anywhere has boosted the use of Wi-Fi networks. Researchers have shown that the ability to use Wi-Fi networks can solve the indoor positioning problem. As a result, Wi-Fi fingerprinting-based localization has emerged as a viable and cost-effective solution for indoor positioning systems (He and Chan, 2016). The strategy relies on the idea that every indoor location can be identified by a unique signal feature known as a fingerprint. A typical Wi-Fi fingerprint consists of received signal strength (RSS) measurements from multiple access points (APs) to provide a fingerprint of the radio conditions of a location. Later, the location of a fingerprint with unknown position information can be calculated using a similar kind of fingerprint previously recorded with a reference location. However, this fingerprint-based positioning system often suffers from significant positioning errors due to noise, obstruction, and the unstable behavior of radio signals. As a result, in the indoor environment where humans spend approximately 80% of their time an effective and error-free positioning system is lacking. This has motivated us to research the fingerprinting positioning system (FPS).

Situational awareness in various transportation industries depends on localization information to a large extent. In aviation, automatic dependent surveillance-broadcast (ADS-B) is a surveillance technology used to broadcast an aircraft's identity, position, speed, etc., to other aircraft or control centers. ADS-B has been mandated in several countries as a cornerstone of the next-generation air transportation surveillance system to make air transportation safer and to meet future challenges. Despite providing many useful services, ADS-B falls short in terms of security. The main problem is that sufficient security measures (e.g., authentication, encryption) were not considered when the protocol was designed. The Federal Aviation Administration (FAA) claims that unencrypted data links are necessary due to operational requirements (Finke et al., 2013). The missing security features and the advancement of transmission-enabled software-defined radio (SDR) technology have forced ADS-B to face unprecedented security challenges. A variety of ADS-B attacks have been outlined in the literature (Costin and Francillon, 2012; Schäfer et al., 2013; Strohmeier et al., 2014; Braeken, 2019; Wu et al., 2020); however, only a few studies have investigated these concerns practically using ADS-B packet data. The lack of thorough investigation of RF link-based attacks on ADS-B, the unknown impact of attacks on various ADS-B installations, and the need to identify the error-handling capabilities of diverse ADS-B setups have motivated us to research ADS-B.

The automatic identification system (AIS) is a ship-tracking system used to improve navigation safety and avoid collisions in maritime transportation. It periodically transmits a ship's name, position, speed, etc., to nearby vessels and other naval entities. In 2004, the International Maritime Organization (IMO) mandated the use of the AIS (IMO, 2004). The AIS protocol was designed three decades ago when RF attacking equipment was not widely available. Therefore, like ADS-B, the AIS also did not consider some security measures, such as authentication and encryption. However, current attack tools and knowledge have made the AIS vulnerable to cyberattacks. Several examples of AIS exploitation have been reported (Bateman, 2021; Zwirko, 2019; MarineTraffic, 2019). These reports imply that the AIS has already been exploited at the national or military level. However, only a few researchers in academia have looked into such dangerous security weaknesses (Balduzzi et al., 2014; Cruz et al., 2018; Marques et al., 2019; Androjna et al., 2021). Moreover, many new AIS hardware, software, transponder, and mobile applications have been developed, which have remained untested against cyberattacks. Also of note is that attackers have developed new intelligent attacking strategies and tools. Therefore, evaluating attacks on modern AIS setups is critical. This has motivated us to research the AIS.

1.2 Research Objectives and Scope

This thesis focuses on the following three topics:

1. Analyzing Wi-Fi fingerprinting-based indoor localization.
2. Investigating the security issues in ADS-B, focusing on aircraft localization.
3. Exploring the security concerns regarding the AIS, focusing on ship localization.

Several studies have investigated different aspects of the FPS. For example, Torres-Sospedra et al. (2015) studied the effect of distance metrics, Xia et al. (2017) studied the impact of the applied method, Liu et al. (2017) investigated sources of error, Zhuang et al. (2016) demonstrated the use of filters to reduce errors, and Song et al. (2019) used powerful image processing tools in the FPS. Our objectives regarding the FPS are as follows:

- To employ new technology to share fingerprint and localization experience to improve the positioning accuracy.
- To Find an effective error prediction technique so that the end user can be informed about the service quality.
- To Analyze the core part of the FPS, called RSS, for a simplified and secured FPS.

In aviation, ADS-B is used to share positioning and other crucial traffic-related information. Given the advantages, many countries are considering ADS-B deployment. The US and Europe have already mandated the use of ADS-B to fly through their skies, beginning in early 2020 (FAA, 2018; EASA, 2018). However, researchers have raised concerns about the security of this system because no authentication and encryption were considered during the design of the protocol. Our objectives in ADS-B research are as follows:

- To discover new vulnerabilities of ADS-B by attacking the system over RF links.
- To test different cyberattacks on a wide range of ADS-B setups to gain real-life experience.
- To investigate the possible countermeasures against attacks.

In maritime transportation, the AIS has been in use for approximately two decades (IMO, 2004). The goal of the AIS is to ensure the situational awareness of marine vessels and other entities by exchanging position coordinates and additional traffic-related information. However, the AIS falls short in security, as it failed to consider basic security measures during the protocol design. Our objectives for the AIS research are as follows:

- To discover new vulnerabilities of the AIS by implementing different cyber-attacks over RF links.
- To evaluate the impact of attacks on a broad range of AIS solutions.

Our AIS vulnerabilities detection approach could help to effectively address AIS attacks and develop proper countermeasures in the near future.

Each of the research focuses and the corresponding publications have addressed different positioning-related problems. Figure 1 shows the publication topics within the scope of the research area.

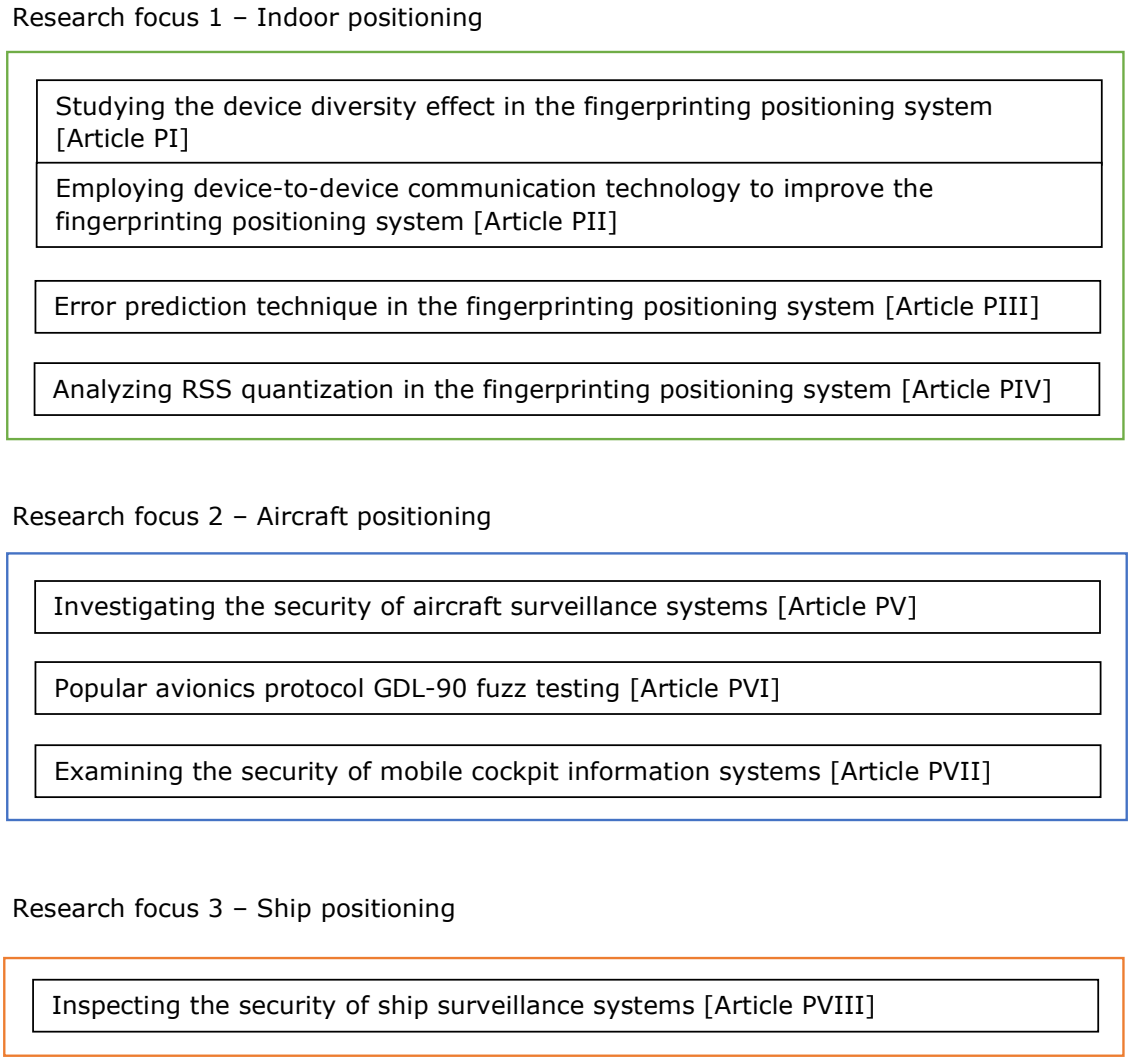


FIGURE 1 Publication topics within the scope of the thematic focuses

1.3 Main Contributions of the Thesis

This thesis summarizes eight publications concerning FPS, ADS-B, and AIS that all use RF as the means of localization. Regarding the FPS, our goal was to increase the accuracy, whereas, regarding ADS-B and AIS, we mainly investigated existing and novel attacking concepts and possible countermeasures. The main contributions of the thesis can be concisely described as follows:

We investigated how the FPS can be developed in Articles PI to PIV. The device diversity effect in the radio map was studied in Article PI. In Article PII, we demonstrated how device-to-device (D2D) communication could help to develop positioning accuracy. The study shows that positioning errors could be reduced by approximately 44% by sharing fingerprints and positioning experiences with other devices. In Article PIII, we studied how positioning errors can be predicted in the FPS so that end users can be notified of the errors to avoid frustration. Furthermore, an error prediction method could help service providers to adopt a

different strategy to improve service quality. In Article PIV, we analyzed the RSS quantization. The test result shows that if a quantized RSS fingerprint can carry the major characteristics of a radio condition, it is satisfactory for fingerprint-based localization. Our proposed method could simplify the hardware configuration, enhance security, and reduce approximately 40–60% of storage space and data traffic.

Different security aspects of ADS-B were thoroughly investigated in this thesis. In Article PV, we developed a test bed consisting of 13 hardware and 22 software programs from 4 operating systems, resulting in 36 configurations. Including 5 novel attack concepts, we practically demonstrated 12 attacks on ADS-B. Our proposed coordinated attack created logical vulnerabilities in most of the setups. More than 50% of the configurations were impacted/crashed due to the denial of service (DoS) attack. Distress alarms such as aircraft hijacking were triggered easily. We noticed that although data integrity checking is defined in the ADS-B protocol, the data validity checking remains neglected. Therefore, we were able to pass technically correct but logically incorrect data through the ADS-B communication. Different kinds of jamming attacks and their variants, such as aircraft disappearance and trajectory modification, were tested. Avionics protocol fuzz testing was extensively studied in Article PVI. We observed that around 56% of electronic flight bag (EFB) applications crashed or became unresponsive due to protocol fuzzing. In Article PVII, we evaluated the security of six mobile cockpit information systems (MCISs). We found that all of them are vulnerable to cyberattacks. System crashes due to a DoS attack is a common problem for this type of mobile setup. To check the error correction capability of the ADS-B setups, an error-handling test was conducted in Article PV. Most of the ADS-B configurations support up to 2-bit error correction. As a part of the countermeasures, we investigated an RSS-distance model and the Doppler shift effect of the ADS-B signal. Although the model achieved 90% accuracy in spoofing signal detection in the best case, the Doppler shift effect experiment did not show any favorable result.

AIS security concerns were extensively studied in Article PVIII. We conducted 11 attacks/tests on 19 AIS setups. Along with some existing attack concepts (e.g., DoS, spoofing, flooding), we demonstrated several novel attack ideas, such as a coordinated attack, overwhelming alerts, and logically invalid data encoding. The result shows that the DoS attack impacted 89% of the tested setups. The coordinated attack, overwhelming alerts, jamming, flooding, etc., are all very effective against the AIS. The test result shows that although the AIS has error detection functionality, it does not support error correction. However, an error correction feature could help to reduce RF pollution. We also identified an AIS preamble-related implementation flaw, which could affect the interoperability of different AIS devices.

1.4 Organization of the Thesis

This thesis is organized as follows:

Chapter 2 provides the relevant knowledge concerning the FPS. Possible sources of error, popular positioning algorithms, and database descriptions are explained. Subsequently, some developments regarding the FPS are presented based on the published articles.

Chapter 3 contains details of our ADS-B experiment. Various existing and novel attacks on ADS-B are demonstrated. Defense strategies using the RSS-distance model and Doppler shift are also discussed.

Chapter 4 details different attacks on and tests of the AIS. Technical aspects of the AIS, our test bed description, and attacks are explained. Some defensive measures in the current literature are also discussed.

Finally, in Chapter 5 we draw conclusions by unifying the important outcomes presented in the dissertation. Furthermore, we discuss possible future studies relevant to the topics.

Author's Role in Included Articles

The author of this thesis is the main person responsible for conceiving, designing, and conducting most of the experiments and is regarded as the first author of all the publications except Article PVI. The author also did most of the writing, with the coauthors making valuable contributions. Riaz Mondal assisted in Articles PI and PIII with constructive comments. Joaquín Torres-Sospedra helped with writing and drawing figures in Articles PII and PIV. Dr. Andrei Costin arranged funding and test equipment for ADS-B and AIS experiments. He also assisted with experiment designing, brainstorming, suggestions, comments, and writing in Articles PV to PVIII. Hannu Turtiainen acted as the main contributor for Article PVI, and the author of the thesis contributed with some experiments and writing. Hannu Turtiainen also contributed in Articles PV to PVIII with programming and writing.

2 FINGERPRINTING POSITIONING SYSTEM

Localization is an essential aspect of everyday living. Many emergency services (e.g., rescue and navigation), commercial activities (e.g., advertisement and delivery), and even fancy gaming require position information to work properly. Satellite-based positioning systems such as GPS, Galileo, and BeiDou have solved most outdoor positioning-related problems. However, in indoor environments, these GNSS-based positioning systems' performance is severely degraded due to the obstruction of satellite signals. To solve this problem, many indoor positioning systems have been developed. Of these, the FPS has become a viable and cost-effective technical solution. This chapter contains the details of our FPS research.

2.1 Basic Principle

In the FPS, RSS measurements are used as the key means to perform the localization tasks. RSS is a measurement of a radio signal's strength. In a Wi-Fi network, the RSS from different APs creates a unique pattern for a particular geographical point called a fingerprint. Due to the attenuation characteristics of the radio signal, depending on geographical location, this pattern changes. Therefore, a fingerprint remains unique for a particular location. Therefore, a location can be identified based on a radio signal fingerprint. The FPS works according to the database correlation approach in which the position of user equipment (UE) is estimated by correlating that equipment's current fingerprint with a previously recorded database. The FPS has two phases, the training phase and the testing phase. Fingerprints are collected and stored with associated location information during the training phase through a site survey or crowdsourcing. In the testing phase, a fingerprint with an unknown location is compared to the database to determine with which training fingerprints the testing fingerprint closely matches. Generally, signal distance is considered for the purpose of matching. To calculate a signal distance between a training fingerprint and a testing fingerprint, many

types of distance metrics can be used, such as Euclidean, Sorensen, or Manhattan distance metrics (Torres-Sospedra et al., 2015). If P and Q are two RSS vectors of test and training fingerprints, respectively, and their length is n, the signal distance between them can be calculated as follows:

$$\text{distance}_{\text{euclidean}}(P,Q) = \sqrt[2]{\sum_{i=1}^{n}(P_i - Q_i)^2}, \tag{1}$$

$$\text{distance}_{\text{sorensen}}(P,Q) = \frac{\sum_{i=1}^{n}|P_i - Q_i|}{\sum_{i=1}^{n}(P_i + Q_i)}, \tag{2}$$

$$\text{distance}_{\text{manhattan}}(P,Q) = \sum_{i=1}^{n}|P_i - Q_i|. \tag{3}$$

Figure 2 depicts the general concept of the FPS. A radio map or database is created during the training phase by collecting fingerprints from different locations. In the testing phase, to retrieve a test fingerprint's location, its RSS vector is compared with the database. The, most similar training fingerprints are selected depending on the signal distance. Finally, the test fingerprint's location is computed based on the locations of the short-listed training fingerprints.

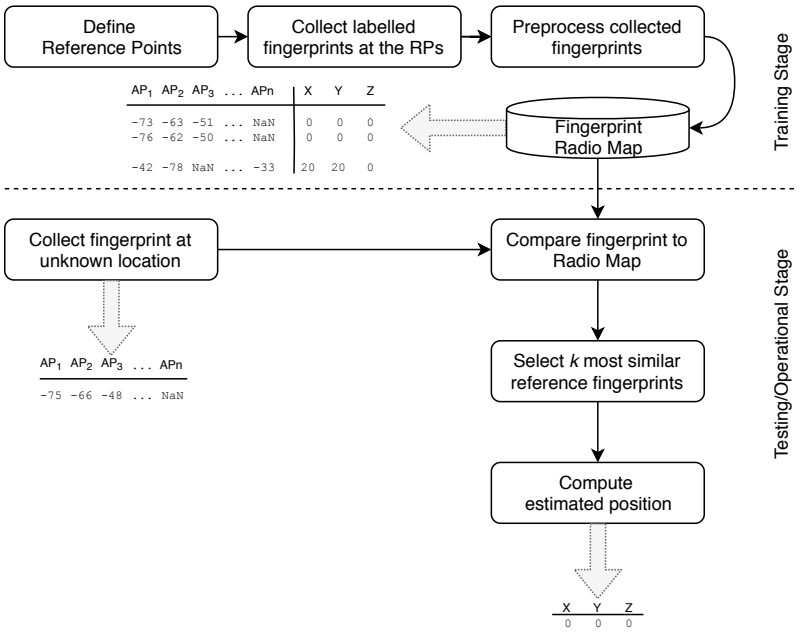


FIGURE 2 Basic architecture of the fingerprinting positioning system

2.2 Sources of Error

The FPS works according to the database correlation method. Any discrepancy between the training and testing data could affect the positioning performance. The core part of the FPS is the RSS, which can be influenced by many natural

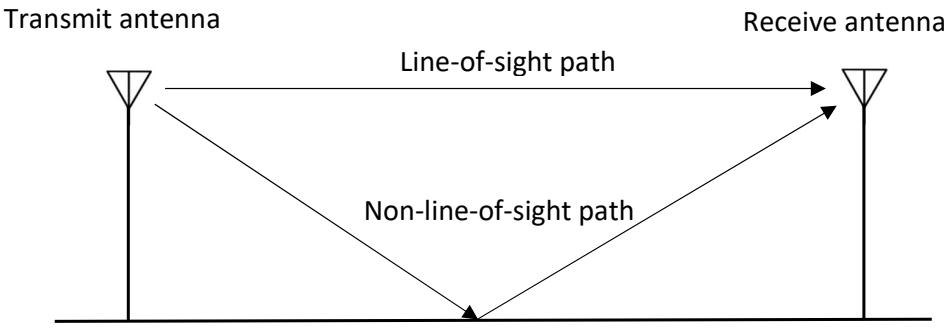


FIGURE 3 Multipath propagation of radio signal

and artificial causes. Moreover, different positioning algorithms yield different results. Here, we discuss some important sources of error in the FPS.

A Wi-Fi signal is an electromagnetic wave. Any electromagnetic wave is affected by natural phenomena such as reflection, refraction, diffraction, absorption, and attenuation (Stein, 1998). These phenomena impact the strength of the signal. As a result, the core part of the FPS is affected. For example, a radio signal is generally reflected by the surrounding objects. Any object near the transmitter or receiver can cause multipath propagation of a signal. Figure 3 shows that a radio signal could travel from an emitter to a receiver via line-of-sight and non-line-of-sight paths. In practice, a signal could be reflected by many objects. Each reflected path experiences a different attenuation and phase shift. At the receiver end, all the paths are added up. The summation of multipath signals can affect the signal strength constructively, destructively, or a mix of both. The reflection depends on obstacle patterns and materials that continuously change in an indoor environment, such as humans, furniture, and construction materials. Because the obstruction pattern always changes in an indoor environment, it causes an RSS difference for a fingerprint from the same location over time, which induces errors in the fingerprinting positioning. Sen et al. (2013) proposed using physical layer information to extract the signal strength and angle of only the direct path to avoid this problem.

Any substantial difference between the training and testing data could impact the positioning performance. For example, data or fingerprints are collected by site surveying or crowdsourcing. A coverage gap could be created if contributors do not cover all the areas adequately during the training phase. Talvitie et al. (2014) studied the coverage gap effect by randomly losing training fingerprints. They reported that 20% of the coverage gap contributed to approximately 10% of the positioning performance degradation. Additionally, incorrectly reported information (e.g., RSS or position) during any phase would result in erroneous positioning performance.

People use various types of smartphones, the network card or chipset for which comes from different manufacturers. The sensitivity of all chipsets is not equal. Furthermore, the antenna design, height, position, and gain differ from one device to another. Therefore, different devices might sense an identical radio

signal at different RSS levels. This effect is called the device diversity effect, which is considered a source of error. To address this problem, instead of using absolute RSS values, Mahtab Hossain et al. (2013) used the difference in RSS observed by two receivers to create robust fingerprints. Their signal strength difference-based localization algorithms outperformed the traditional RSS-based FPS with approximately 20% better accuracy.

Liu et al. (2017) reported that the AP density had a considerable effect on the positioning performance. A higher number of APs increases the reference of a fingerprint, which enhances the distinctiveness of that fingerprint. However, increasing the AP number also increases the RSS numbers, which tend to be erroneous. Moreover, due to the coverage limits of a Wi-Fi network, all APs are not available for all fingerprints. Therefore, using an optimal number of APs is desirable. Setting a universal optimal number of APs is challenging because every indoor environment is different in shape and size. In practice, the use of 5 to 20 APs per fingerprint is the most common in the current literature (Torres-Sospedra and Moreira, 2017).

The density of the training fingerprint also contributes to the positioning performance. Due to the nature of the radio signal, the RSS value fluctuates, which increases the possibility of a fingerprint being corrupted in its signal domain. The temporal average of fingerprints increases the probability of recording the actual radio condition of a geographical area. Furthermore, to define an operational fingerprint's location, the positioning algorithm tries to find a closely matched training fingerprint, expecting the training fingerprint to be geographically located near the test fingerprint. If all the training fingerprints are sparsely located, the algorithm is forced to select the best possible option, which already introduces a mix of errors. However, if the training fingerprints' density is too high, it would be time-consuming and computationally burdensome to calculate the position that does not improve the performance after a particular saturation level (Moghtadaiee and Dempster, 2014; Xia et al., 2017). The optimal density depends on the test bed setup.

Crowds, temperature, and humidity also affect the radio signal. Because these variables change over time, it could affect the positioning performance. The coverage area of an AP can be increased or decreased over time, resulting in a substantial change in the RSS. Finally, there are many positioning algorithms available. Each algorithm uses different techniques and distance metrics to estimate the position, resulting in different positioning performances (Nessa et al., 2020).

2.3 Positioning Algorithms

Since Bahl and Padmanabhan (2000) first demonstrated the FPS, researchers have proposed many algorithms that mainly focus on improving positioning performance. In this section, we discuss some popular algorithms.

K-Nearest Neighbor The K-nearest neighbor (KNN) algorithm classifies instances based on their similarity. Due to the low computation effort and better positioning accuracy, it has become the most popular and widely used algorithm in the FPS (Xie et al., 2016; Bi et al., 2018). This algorithm calculates signal distances between the test fingerprint and each training fingerprint using different distance metrics (e.g., Eqs. (1), (2), (3)). The distances are then sorted in descending order. Finally, the K amount of minimum signal distance containing training fingerprints' mean location is given as the calculated location of the test fingerprint. Theoretically, K can be any positive number. However, in practice, K is limited from 1–5. Too high of a K value could induce error by averaging the training fingerprint location from a distant area.

Weighted Centroid The weighted centroid (WC) algorithm calculates the position of a test fingerprint based on the weighted average of the positions of APs available in the fingerprint (Blumenthal et al., 2007; Razavi et al., 2015). The set of all hearable APs by the device is denoted by AP_h. The following formula is used to calculate the WC-based test fingerprint's location:

$$P_i(x,y,z) = \frac{\sum_{j=1}^{n} \left(w_{ij} \times AP_{hj}(x,y,z)\right)}{\sum_{j=1}^{n} w_{ij}}. \tag{4}$$

Here, n is the size of the set AP_h and w is the weight function. The value of w is implementation-dependent; we used $w = 10^{rss/10}$. If the positions of the APs are not given in the database, they need to be calculated first according to the same concept of Eq. (4). In this case, the location of the fingerprints from where an AP is heard can be used to calculate the AP's location. After that, the test fingerprint's position is calculated.

Log-Gaussian Probability The log-Gaussian probability (LGP) algorithm calculated a Gaussian likelihood function ι_i for each training fingerprint against the test fingerprint (Honkavirta et al., 2009; Laitinen et al., 2015). If the commonly heard AP's (N_{ap}) RSS set for the test fingerprint is P_k and is $Q_{\hat{i},k}$ for the training fingerprint, then ι_i can be calculated as follows:

$$\iota_i = \sum_{k=1}^{N_{ap}} \log\left(\frac{1}{\sqrt{2\pi\sigma_{ap}^2}} \exp\left(-\frac{\left(P_k - Q_{\hat{i},k}\right)^2}{2\sigma_{ap}^2}\right)\right). \tag{5}$$

Here, σ_{ap} is a noise variance; to maintain consistency with previous studies, we used $\sigma_{ap} = 7$ (Lohan et al., 2017). The highest ι_i containing training fingerprint's location is selected as the test fingerprint's location.

K-Means Clustering The K-means clustering algorithm has been used in the FPS by many researchers (Arya et al., 2013; Razavi et al., 2015). This algorithm begins with a set of training fingerprints Q_i, where $i = 1, 2, ..., n$, and a predefined

maximum cluster number K. The task is to choose K centers c_k to minimize the following distance function:

$$d\left(Q, c\right) = \sum_{i=1}^{n} \left|Q_i - c_k\right|. \tag{6}$$

The K-means clustering algorithm works as follows:

1. The first center c_1 is chosen uniformly at random from Q.
2. A new center c_k is chosen from Q with probability

$$\frac{D\left(Q_i\right)^2}{\sum_{i=1}^{n-1} D\left(Q_i\right)^2}.$$

   Here, $D(Q_i)$ denotes the shortest RSS distance from a fingerprint to the already chosen cluster center.
3. Step 2 is repeated until all K centers are chosen.
4. For each c_k, training fingerprints are assigned to it that are closer to it than any other c_k.
5. A new c_k is computed from the mean of all training fingerprints that belong to the previous c_k.
6. Steps 4 and 5 are repeated until c no longer changes.

Depending on the RSS set, this algorithm determines in which cluster a test fingerprint could belong. The delegated cluster's center is considered as the test fingerprint's position. In the published articles, we have used Euclidean distance and the Davies–Bouldin criterion in MATLAB to determine the optimal value of K.

Convolutional Neural Network Researchers have suggested many machine learning and neural network-based algorithms to address the fingerprinting positioning-related problem. Among them, the convolutional neural network (CNN) has drawn significant attention for its strong feature-extracting capability. Convolution can replace the general matrix multiplication that reduces the computation complexity in the neural network. Some studies have demonstrated better floor detection and positioning accuracy using this method (Song et al., 2019; Qin et al., 2021). The RSS values are first normalized into the 0 to 1 range. The RSS vector is then converted into an $M \times N$ rectangle or $N \times N$ square. This grid can be considered the image of that fingerprint. Through convolution and max pooling, the feature of the image is extracted and trains the network by supplying corresponding position information as the category. During the operational phase, the CNN predicts the position information through a maximum vote at the fully connected layer. Figure 4 shows how a fingerprint can be transformed into an image, which can later be processed by image processing tools.

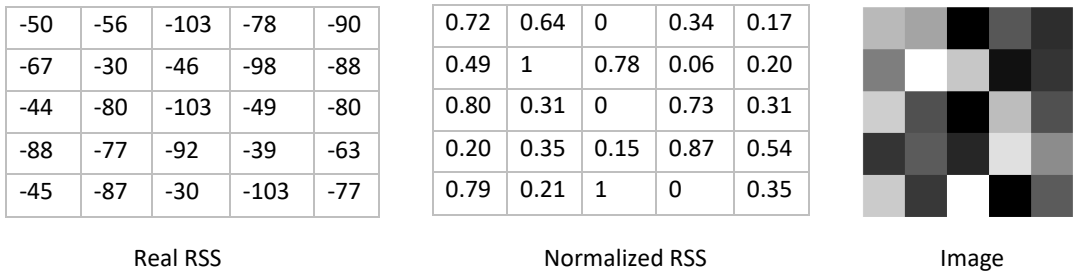


FIGURE 4 Transformation of a fingerprint into an image

2.4 Fingerprint Databases

In the different articles of this thesis, we have used five fingerprint databases. Research groups from various universities (e.g., the Tampere University of Technology (TUT), the University of Jaume I (UJI), the University of Minho, and the University of Mannheim) created these databases and later made them publicly available for research purposes. The important features of all the databases are listed in Table 1. Descriptions of these databases are available in Article PIV .

TABLE 1 Facts about the databases

Database name	Coverage (m^2)	Number of training samples	Number of testing samples	Number of APs	Positioning accuracy (m)	Floor detection (%)	Reference
TUT	22,570	697	3951	991	9.39	91.75	Lohan et al. (2017)
UJIIndoorLoc	108,703	19,936	1111	520	7.74	90.28	Torres-Sospedra et al. (2014)
Minho	1000	4973	810	11	4.7	NA	Moreira et al. (2017)
Mannheim	312	14,300	5060	28	3.01	NA	King et al. (2006)
Library	308	576	3120	620	2.34	100	Mendoza-Silva et al. (2018)

2.5 Development of the FPS

The overall accuracy of the FPS is from a few meters to around a hundred meters (Mautz, 2012; Peral-Rosado et al., 2018). However, the test bed settings and applied methods vary in different studies. Therefore, comparing the positioning accuracy of different studies might not be fair. Although some studies reported that the accuracy of the FPS is around 2–3 m (Bahl and Padmanabhan, 2000; Youssef et al., 2003), it can reach a higher averaged error when the conditions are not favorable, for example, if low density of training fingerprints, fewer APs in the environment, or external resources (e.g., inertial sensors, magnetometer, floor plan) are not considered (Xiao et al., 2016; Potortì et al., 2017; Lohan et al., 2017; Meneses et al., 2019). Thus, the accuracy depends on test bed settings, the quality of the radio map, and the impact of the error. In this section, we discuss our efforts to develop some aspects of the FPS.

2.5.1 Error Prediction

Many artificial and natural causes make the propagation of a radio signal unpredictable in indoor environments. Movable furniture, people, and other obstacles create reflections, refractions, and multipath interference, which significantly affect the RSS of a signal that degrades the FPS accuracy. However, the GNSS-based positioning system performs poorly in indoor areas due to the blockage of satellite signals. Therefore, there is no alternative system to verify the FPS performance. However, evaluating the FPS performance for many modern-day services and applications (that work based on precise location information) is very important. In Article PIII, we demonstrated how an error could be predicted in the FPS. Our method works on the idea that if a few spatially adjacent test fingerprints (TFPs) show calculated/estimated position similarity, it could be assumed that the estimated position is correct because the result is verified several times. However, if they show scattered positioning results, it can be said that the results are erroneous. Figure 5 demonstrates the proposed concept.

Figure 5 shows that, during the position estimation, there may be three scenarios for a few test fingerprints from a closely located area (having a real positions cluster radius (r_{real})). The estimated positions cluster radius (r_{est}) can be smaller, approximately the same, or greater than r_{real}. If r_{real} is a small value, then for the first two scenarios the error would not be very large. However, when r_{est} is greater than r_{real}, the error depends on how much greater the r_{est} is; this

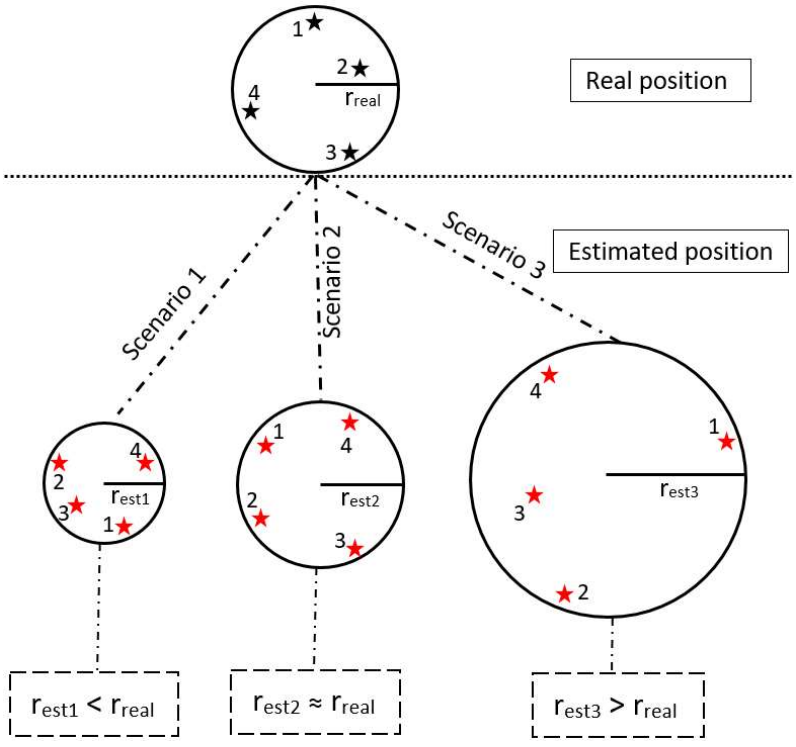


FIGURE 5 Conceptual positions of four fingerprints

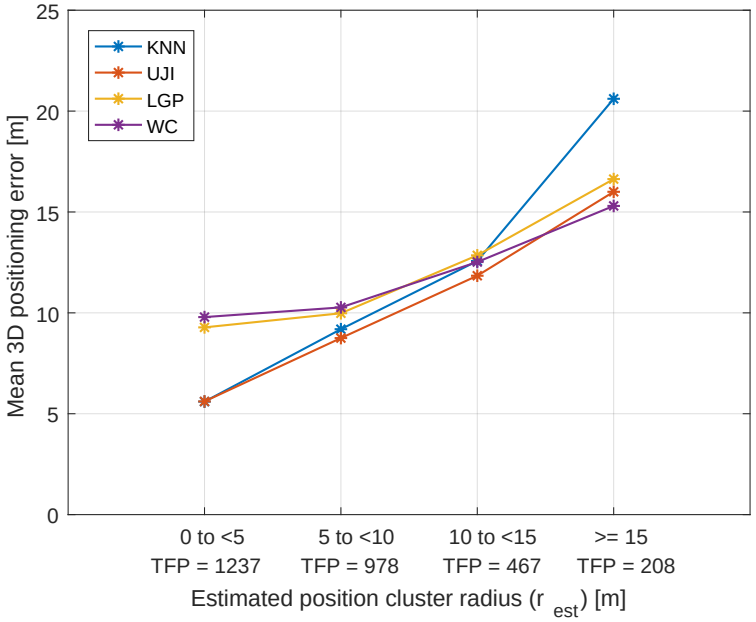


FIGURE 6 Relationship between pe and r_{est}

is the most common scenario in the FPS. The r_{real}, r_{est}, and positioning error (pe) roughly show the following relationship:

$$pe \propto \frac{r_{est}}{r_{real}}. \tag{7}$$

Because r_{real} is controllable from the user or service provider side, if r_{real} maintains a constant value, the relationship can be expressed as follows:

$$pe \propto r_{est}. \tag{8}$$

According to Eq. (8), it is possible to get an idea about the pe with the help of r_{est}, where the test fingerprints' real position information is not needed. Using this idea in Article PIII, we conducted an experiment on the TUT database using four different algorithms. Figure 6 shows that if r_{est} increases, the positioning error also increases, through all four algorithms. So, the positioning error can be predicted by observing the estimated position cluster radius.

2.5.2 D2D Communication-Assisted FPS

The number of mobile phone subscribers and data use have been rapidly increasing. To meet this demand, direct communication between mobile devices called D2D communication has been specified in (3GPP, 2013). This system allows data to be exchanged among devices without it routing through a base station. Therefore, a significant amount of data, such as that in content sharing and multi-party gaming, can be offloaded, which can enhance the cellular network's throughput, spectrum utilization, and energy efficiency. We present technical details of D2D communication in Article PII, where we also propose a collaborative FPS based on this concept. The proposed method is based on the idea that if multiple devices

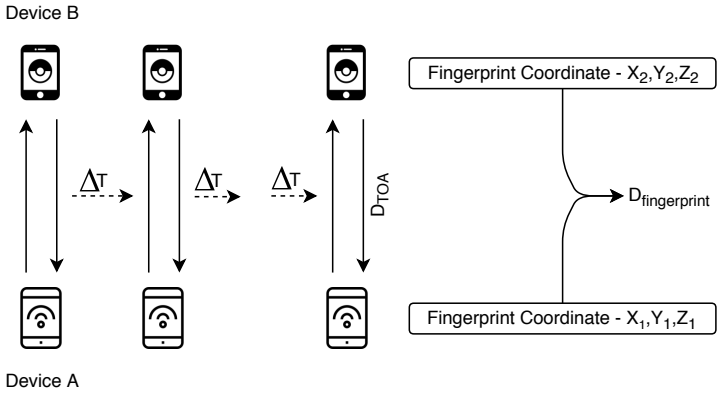


FIGURE 7 D2D communication-assisted FPS

share their fingerprinting positioning experience and some D2D communication aspects, such as signal time of arrival, can be exploited, it may help to improve the positioning accuracy. Figure 7 shows the proposed D2D communication-assisted FPS.

If device A needs its location information, the evolved node B (eNB) can provide it using A's fingerprint. Additionally, A can request another device B to share B's position. Device B can retrieve its position from the eNB based on its fingerprint and share with A. Device A can now calculate the distance between A and B using two methods—the signal's TOA method (D_{TOA}) or based on the eNB given coordinates ($D_{fingerprint}$). If these two distance measurements are relatively close, the position information derived from the FPS can be assumed to be valid because another technique verifies it. This process can be repeated several times after a short time interval, ΔT. At each time, the error indicator (δ) can be calculated as follows:

$$\delta_i = |D_{TOA_i} - D_{fingerprint_i}|. \tag{9}$$

Here, $i = 1, 2, ..., n$. From the several measurements, device A can select the minimum δ containing FPS response to improve confidence. However, if device A cannot make a conclusive decision based on this collaborative process, it can switch back to the traditional process and be satisfied with the non-collaborative FPS response given by the eNB.

We explain the D2D communication-assisted FPS process and result in detail in Article PII. Because, technologically, D2D communication has not been implemented yet, we emulated a scenario where fingerprints from the other device can be shared. We tested the idea on the TUT database using the KNN algorithm. The performance of the proposed method and the traditional method are compared in Figure 8. During the collaboration process, the secondary device could be either stationary or in motion; therefore, we studied both scenarios. In both cases, the cumulative distribution function (CDF) of the positioning error in Figure 8 shows that the proposed D2D communication-assisted FPS outperforms the traditional FPS.

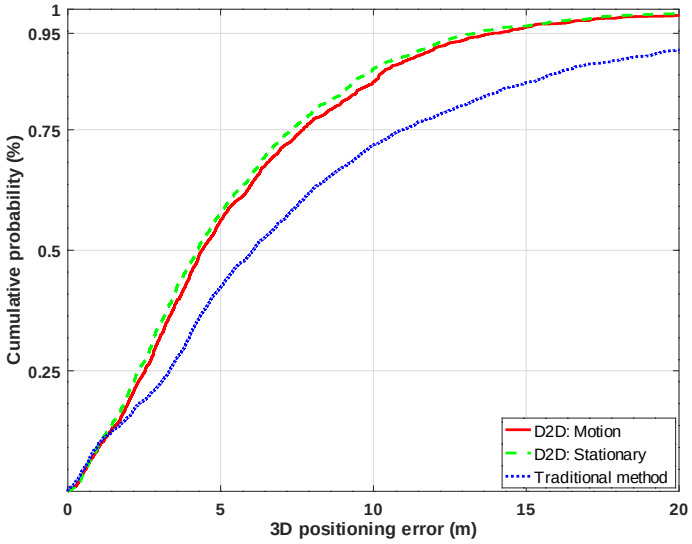


FIGURE 8 Traditional vs. D2D communication-assisted FPS performance

2.5.3 Analysis of RSS Quantization

Generally, the RSS values are expressed in dBm units with 1-dBm granularity. The primary purpose of RSS is to evaluate signal and service quality, which needs fine-granular RSS. However, in the FPS, in addition to RSS data, reference AP labels are also recorded. Because a fingerprint has so many reference APs, a coarse-granular RSS may result in the same positioning accuracy as that of a finger-granular RSS. However, according to some studies, storing, exchanging, and processing fine-granular raw RSS values can occupy a considerable amount of memory (Mizmizi and Reggiani, 2016; Richter et al., 2018) and can be a potential threat to users' location privacy (Li et al., 2014; Konstantinidis et al., 2015). In this case, an efficient quantization method could be useful. In Article PIV, we thoroughly investigate the RSS quantization in five publicly available databases.

Figure 9 shows that the spreading of RSS values is mostly between -30 and -100 dBm. To keep the same length for all the fingerprints, it is common practice to use a very weak signal value (e.g., -103 dBm) for the non-heard AP. Therefore, the RSS from -30 to -103 dBm is effectively used in the fingerprints. Therefore, quantizing the RSS from -30 to -103 dBm could be sufficient for the FPS. The following equations are used to convert the RSS value from dBm to mW and vice versa:

$$P_{\mathrm{mW}} = 1\,\mathrm{mW} \times 10^{(P_{\mathrm{dBm}}/10)}, \tag{10}$$

$$P_{\mathrm{dBm}} = 10 \times \log_{10}\left(\frac{P_{\mathrm{mW}}}{1\,\mathrm{mW}}\right). \tag{11}$$

Generally, quantization happens at the absolute signal energy level. Therefore, we used Eq. (10) to convert all RSS values from dBm to mW before applying quantization. However, we found that, instead of mW, if quantization is applied to dBm, it provides the same results because they are the same value represented by two distinct units.

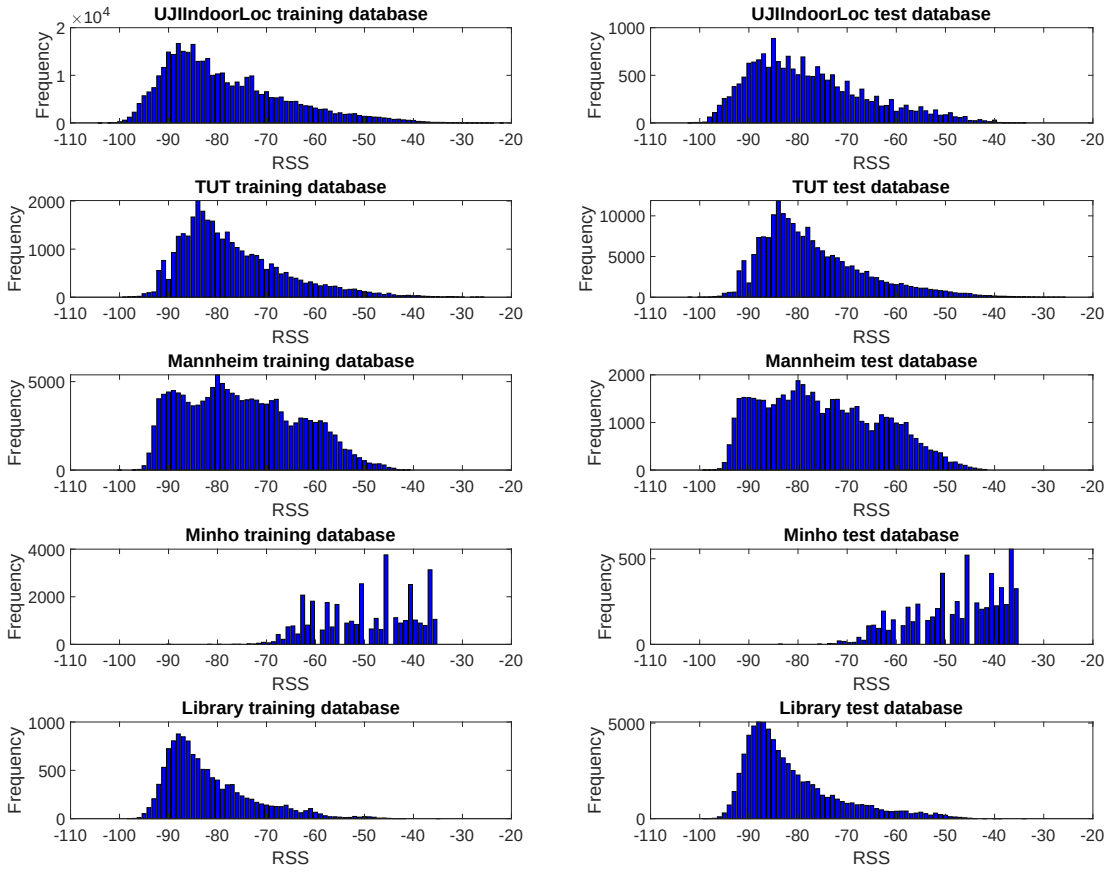


FIGURE 9 RSS distribution in the databases

Quantizer Figure 9 shows that the usable RSS range for FPS is $-30\,\text{dBm}$ to $-103\,\text{dBm}$, which is equivalent to $10^{-3}\,\text{mW}$ to $10^{-10.3}\,\text{mW}$. Therefore, we can set the maximum energy index, $E_{\max} = -3$, and minimum energy index, $E_{\min} = -10.3$. We propose four different formulas (f_1, f_2, f_3, f_4) for seven different bit numbers (n), where $n = 2, 3, 4, 5, 6, 7, 8$ and $i = 1, \ldots, 2^n$, to divide the signal energy for each quantization level. The first formula yields a linear quantization; depending on the applied bit number, each level shares an equal energy index

$$f_1 = (E_{\max} - E_{\min})/(2^n - 1). \tag{12}$$

In the next three formulas, we applied nonlinear quantization to reflect the nature of signal propagation. From the maximum energy index, at each quantized level, the energy index is reduced as follows:

$$f_{2(i)} = (E_{\max} - E_{\min}) \times \frac{\sum_{i=1}^{i_{th}} \sqrt{2^{i-1}}}{\sum_{i=1}^{2^n} \sqrt{2^{i-1}}}, \tag{13}$$

$$f_{3(i)} = (E_{\max} - E_{\min}) \times \frac{\sum_{i=1}^{i_{th}} \sqrt[3]{2^{i-1}}}{\sum_{i=1}^{2^n} \sqrt[3]{2^{i-1}}}, \tag{14}$$

$$f_{4(i)} = (E_{\max} - E_{\min}) \times \frac{\sum_{i=1}^{i_{th}} \log\left(2^{i-1}\right)}{\sum_{i=1}^{2^n} \log\left(2^{i-1}\right)}. \tag{15}$$

Quantized Fingerprint We randomly chose a fingerprint to check its characteristics before and after quantization. This fingerprint had 61 RSS values. Figure 10 shows the fingerprints in traditional and quantized RSS representation based on linear quantization according to Eq. (12). The X-axis shows the reference AP, and the Y-axis shows the RSS values. Figure 10 shows that 2-bit quantization is not enough to accommodate all the characteristics of the traditional fingerprint. At 3-bit quantization, the situation improves. Most of the original fingerprint properties appear to be at a lesser magnitude in the 4-bit quantization. However, this lower bit quantization may result in many identical fingerprints, a problem that we broadly discuss in Article PIV. Our experiment in Article PIV shows that at 4-bit quantization, the granularity is 4.86 dBm, which is not very high. Furthermore, the number of reference APs and a wide variety in signal levels also help to distinguish a quantized fingerprint from other quantized fingerprints. This is

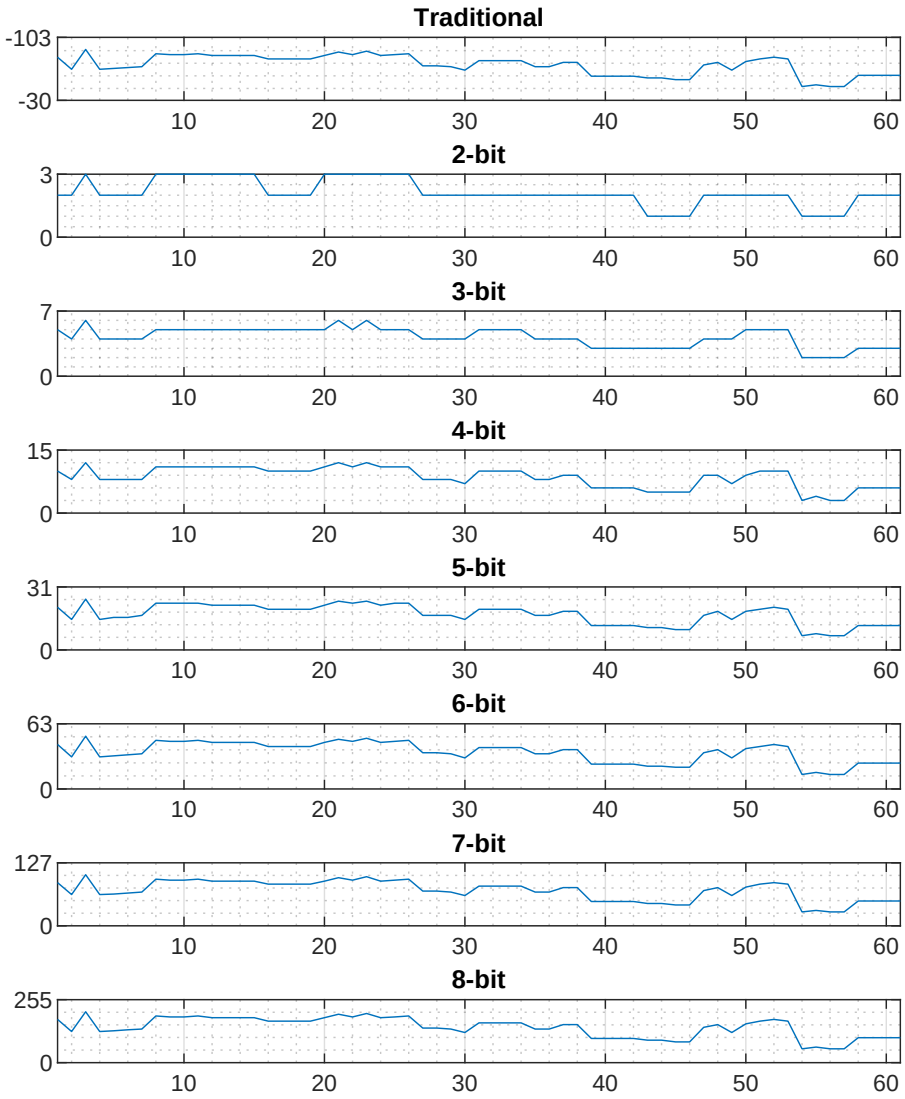


FIGURE 10 Traditional vs. quantized RSS fingerprint

why a 4-bit quantized fingerprint could achieve the same positioning accuracy as the traditional one. However, the situation might vary from sample to sample and environment to environment. Therefore, we conducted experiments on five databases to test the positioning performance of quantized fingerprints.

Positioning Performance Figure 11 demonstrates the positioning performance of quantized fingerprints for five databases. The red dashed line compares results with the traditional RSS representation. Due to inferior performance and capacity, we did not consider 1-bit quantization. Across all the databases, the 2-bit quantization results in a large *pe*. Positioning accuracy improves considerably with 3-bit quantization. However, the 4-bit quantization results in almost the same positioning accuracy as that of traditional RSS. Positioning accuracy barely improves after 4-bit quantization. Different formulas have a slightly different impact on the positioning performance. Formula f_1 shows the best performance in all the databases.

Floor Detection Performance The floor detection percentage in the Library, TUT, and UJIIndoorLoc databases is shown in Figure 12. The Manheim and Minho databases did not have floor-related information. Figure 12 shows that, like positioning performance, the floor detection performance is poor at 2-bit quantization. The situation significantly improves when 3-bit quantization is applied. Beyond 3-bit quantization, the floor detection performance remains almost steady.

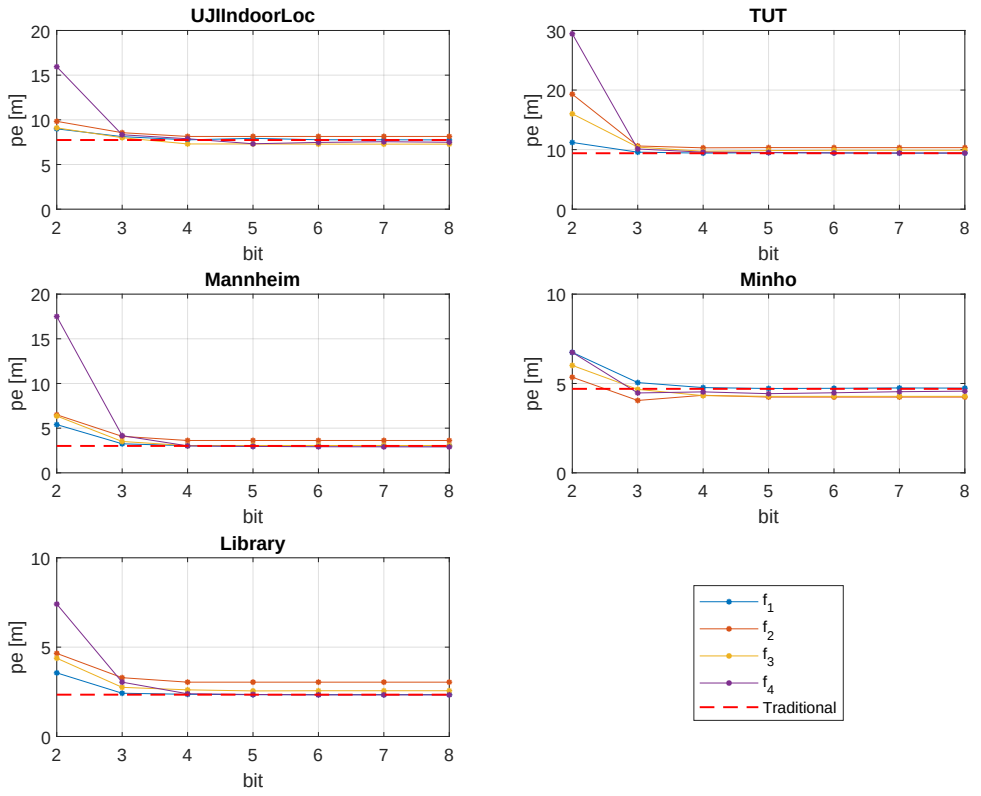


FIGURE 11 Positioning performance of quantized fingerprints

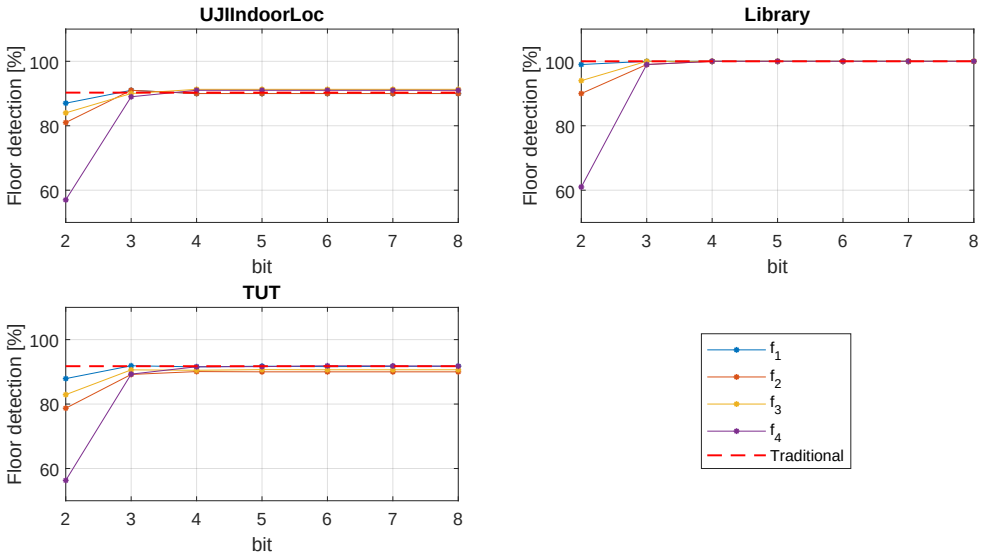


FIGURE 12 Floor detection performance of quantized fingerprints

The experimental results show that in the FPS, a 4-bit quantization could yield the same positioning and floor detection performance as that of traditional RSS representation. Because the proposed method uses only 4-bit quantization, it could offer the following advantages:

Simplification The experiment results show that a maximum 4-bit or 16-level quantization is sufficient for the FPS. As a result, a 4-bit quantizer can be utilized instead of a traditional 8-bit quantizer, simplifying the hardware configuration and the sensing process.

Less Storage The proposed method uses fewer bits, and therefore less memory will be occupied. Figure 13 shows the quantized training fingerprints database size at different bit levels. All the given formulas have different efficiency and quantization techniques. As a result, memory sizes vary depending on the formula. However, on average, 40–60% of memory space in the training databases can be saved by 4-bit quantization.

Less Traffic In the FPS, the test fingerprints need to be exchanged between client and server. Using lower bit quantization could reduce the test fingerprint data size. As a result, there is a potential opportunity to cut the network traffic while exchanging the test fingerprints. Figure 14 illustrates that, compared to the traditional approach, the suggested 4-bit quantization can reduce network traffic by 40–60% on average.

Enhancing Security Traditional RSS includes real signal strength data. Storing and exchanging this raw data could jeopardize a user's location privacy. An adversary between the client device and the positioning server may intercept and get the data. The user's position could then be determined using a radio signal

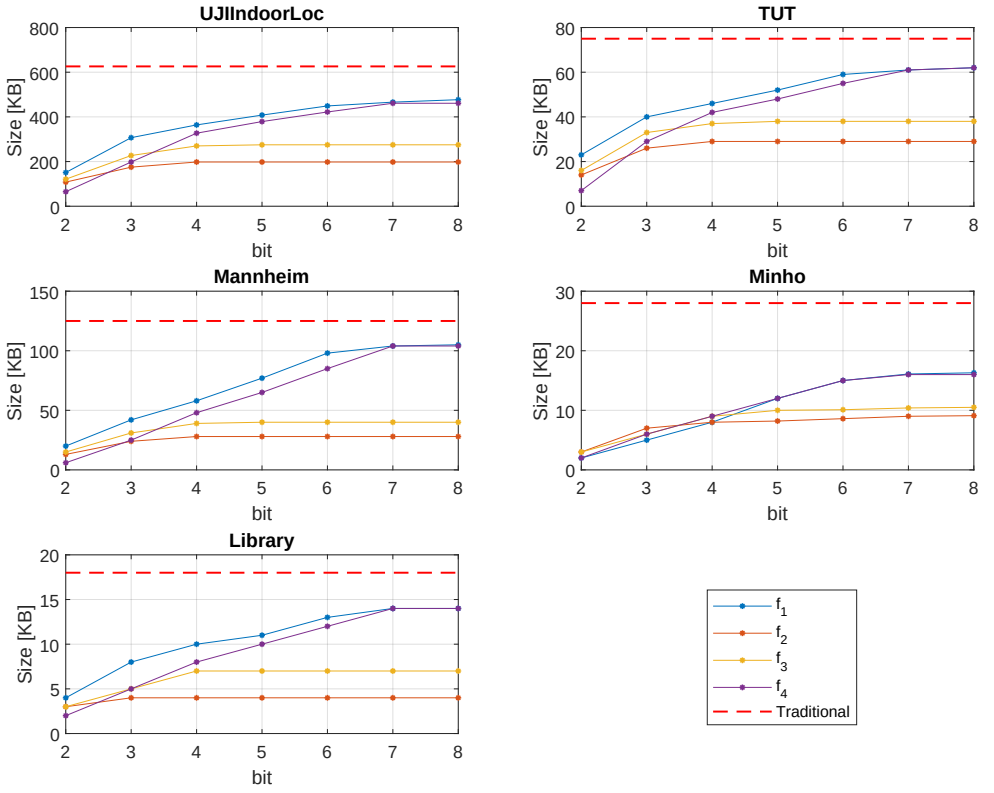


FIGURE 13 Training database size according to formula and bit number

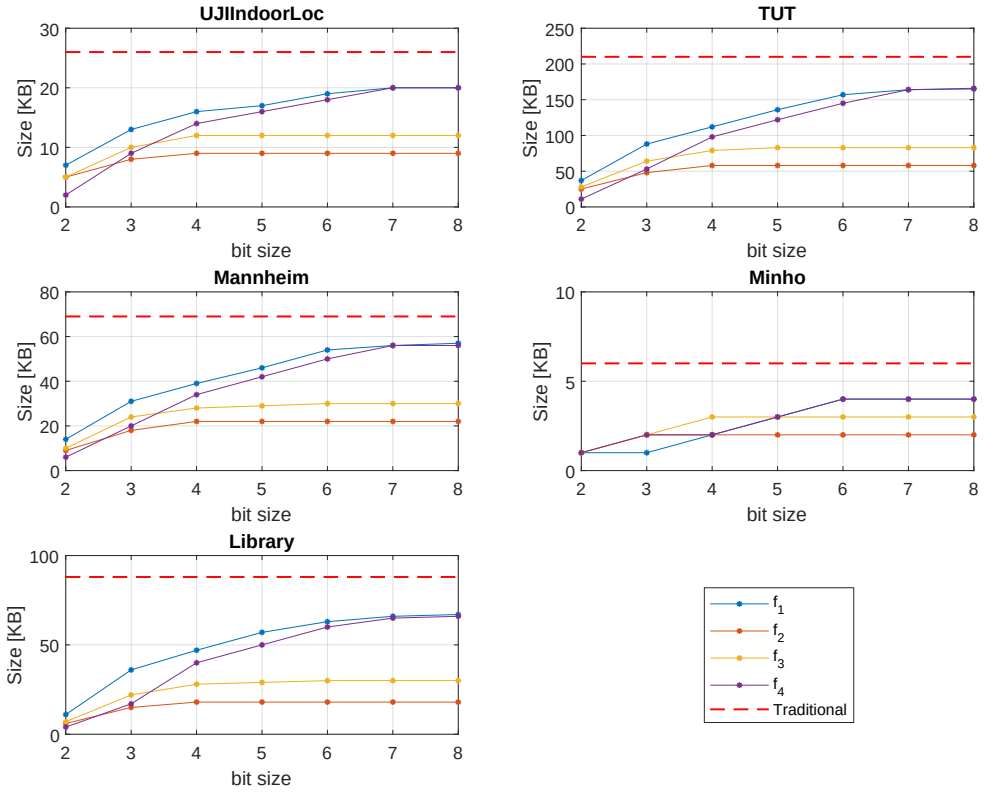


FIGURE 14 Test database size according to formula and bit number

propagation model. However, our suggested method uses mapped RSS values based on a hidden quantization formula and bit number rather than actual RSS values. As a result, obtaining the real RSS and position information would be significantly difficult for any adversary.

2.6 Summary of FPS Study

This chapter is dedicated to the development of FPS. In this regard, three contributions are made. These are how to predict the positioning error, how the FPS accuracy can be improved with the assistance of D2D communication, and a deep analysis of RSS quantization, which shows that 4-bit quantization is sufficient for the FPS. As the structure and setups of the indoor environments vary, it is challenging to apply one testbed's solution to another. Therefore, customized methods need to be applied for an efficient FPS depending on the environment and available resources. Nonetheless, the contributions of this thesis on FPS deliver a general development perspective, which can be effectively applied for FPS development in industry and academia. The next chapter investigates the security aspects of aircraft localization.

3 AUTOMATIC DEPENDENT SURVEILLANCE–BROADCAST

ADS-B has been mandated in several countries as a cornerstone of the next-generation air transportation surveillance system. However, researchers have raised concerns about the security of this system. This chapter contains the details of our ADS-B research.

3.1 Background

In the very early stages of aviation, aircraft position used to be determined by a primary surveillance radar (PSR). A PSR emits a radio pulse. If that wave is reflected by an aircraft (or any other object), some energy is returned to the antenna. The range of an aircraft can be calculated based on the time difference between the emitted pulse and the received pulse, whereas antenna azimuth determines the aircraft's bearing. However, knowing an aircraft's identity became essential with the growing number of aircraft. The secondary surveillance radar (SSR) was designed to address this issue. The SSR can continually interrogate to determine an aircraft's identity and altitude using mode A and C radio pulse. However, in the modern busy air transportation system, SSR is considered slow and inefficient. It is also blamed for RF pollution, lost targets, identity errors, and more. Mode S was designed to address these problems. Mode S employs selective interrogation of aircraft based on the aircraft ID, also known as the International Civil Aviation Organization (ICAO) address or ICAO24 code, which is a unique 24-bit address assigned to each aircraft. The ADS-B idea was later developed based on mode S. ADS-B is a surveillance technology that periodically broadcasts an aircraft's position, identification, velocity, and flight-related data to other aircraft and ground stations over a radio link.

Two different data link technologies meet the ADS-B requirements, ADS-B 1090 and Universal Access Transceivers (UAT) 978. The 1090 MHz channel is used by ADS-B 1090 to transmit flight-related information via mode S transpon-

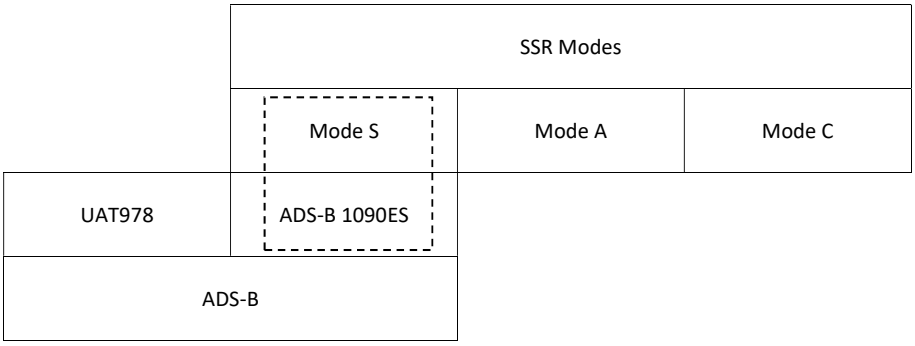


FIGURE 15 ADS-B hierarchy

ders all over the world. However, UAT978 operates in the 978 MHz frequency currently applicable in the USA for aircraft flying below 18,000 feet, focusing on general aviation (GA). ADS-B 1090 is divided into two groups depending on squitter pulses, short squitter and extended squitter. Short squitter contains some protocol information (e.g., downlink format [DF], capability [CA]), aircraft ID, and a cyclic redundancy check (CRC). A short squitter extended with other information such as altitude, position, and velocity is called an extended squitter. In our research, we focused on the extended squitter to facilitate all the information. Figure 15 shows the mapping of ADS-B within SSR.

3.2 ADS-B

ADS-B is a surveillance technology that broadcasts an aircraft's location, velocity, identity, and other flight-related data to other aircraft and air traffic control (ATC) in the vicinity, typically once a second. Figure 16 illustrates the ADS-B communication system. The most critical aspect of ADS-B is the precise location coordinates of an aircraft determined by the GNSS, which is crucial for safety-critical systems such as the traffic collision avoidance system (TCAS). There are two functionalities of ADS-B, ADS-B OUT and ADS-B IN. ADS-B OUT sends information to ATC and other aircraft regarding an aircraft's GPS location, altitude, ground speed, and other data. In contrast, ADS-B IN receives, processes, and displays ADS-B signals.

ADS-B 1090 Extended Squitter ADS-B 1090 extended squitter (ES) is operated on 1090 MHz, just like mode A/C/S transponders, but no interrogation is needed. The maximum transmission rate of ADS-B 1090ES is 6.2 messages/second; however, in practice, the signal is transmitted once a second. Pulse position modulation (PPM) is used to modulate the signal. The length of ADS-B 1090ES is 112 bits. A 0.8 μs preamble is added before the data block. Unlike UAT978, there is no weather information exchanging capacity on it. ADS-B 1090ES is approved, being tested, and functional in many parts of the world, such as the USA, EU, Australia, and Japan (FAA, 2018; EASA, 2018). Figure 17 shows the ADS-B 1090ES message structure.

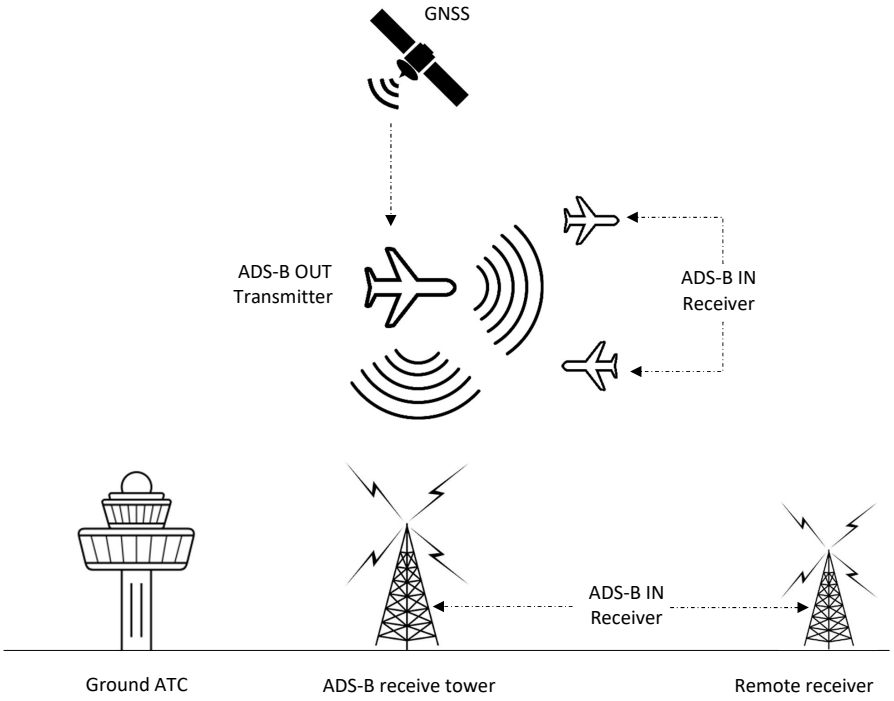


FIGURE 16 ADS-B communication system

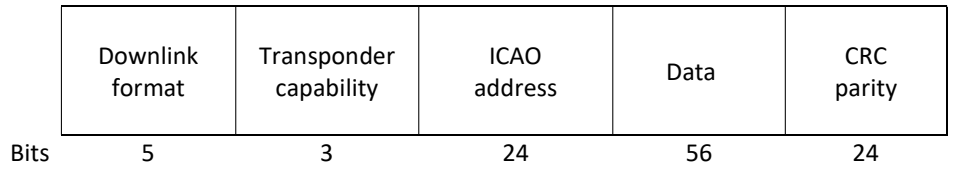


FIGURE 17 ADS-B 1090ES message structure

UAT978 UAT978 operates on the 978 MHz frequency. Compared with 1090ES, UAT978 has some additional facilities. For example, UAT978 supports weather information services, allowing pilots to improve safety. UAT978 uses continuous phase frequency shift keying (CPFSK) modulation with a data rate of 1.041667 Mbps and a modulation index of 0.6. UAT978 communications are divided into two categories, basic messages and long messages. The length of the basic message is 144 bits, whereas the length of the long message is 272 bits. UAT978 uses the Reed–Solomon error correction code as forward error correction (FEC). The FEC length is 96 bits and 112 bits for the basic and long messages, respectively. Figure 18 shows the UAT978 message structure.

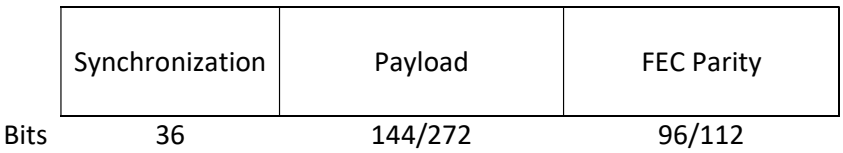


FIGURE 18 UAT978 message structure

3.3 Security Concerns

ADS-B is designed to make the ATC job easier by eliminating the constraints of modes A, C, and S, improving aircraft positioning accuracy via GNSS, and eventually replacing the SSR. However, ADS-B is insecure because it lacks basic security features such as authentication and encryption. Missing basic security mechanisms makes ADS-B easy to forge or tamper with, which affects the confidentiality, integrity, and availability of the transmitted aircraft data (Wu et al., 2020). Despite these security shortcomings, all ADS-B users have to use the currently available protocol, which is insecure. Furthermore, the evolution of transmission-enabled SDR technology has increased security challenges for ADS-B. Because with the help of transmission-enabled SDR, it is possible to produce almost any kind of radio signal with little cost and effort. Researchers have outlined many different types of ADS-B attacks. Some earlier studies are discussed below.

Costin and Francillon (2012) conducted the first ADS-B attack in 2012. They used Universal Software Radio Peripheral (USRP) to transmit a MATLAB-generated attack payload. A Plane Gadget Radar (PGR) received the counterfeit signal and displayed a spoofed aircraft on a virtual radar. They warned that ADS-B technology could be attacked using low-cost setup and recommended proper security measures before full deployment. Schäfer et al. (2013) demonstrated attacks on ADS-B using USRP. They conducted several attacks, such as spoofing, a false alarm, aircraft disappearance, ghost aircraft flooding, ground station flooding, and virtual trajectory modification. They concluded that the ATC process should not rely on ADS-B data until proper countermeasures are taken against the attack. Strohmeier et al. (2014) thoroughly analyzed the 1090MHz ADS-B channel using the OpenSky network. They reported that SDR could pose a significant threat to insecure ADS-B communication. They also observed that increased traffic on the ADS-B channel causes a large number of message losses. In a separate study, they suggested fingerprinting, random frequency hopping, public-key cryptography, and retroactive key publication as secure means of ADS-B (Strohmeier et al., 2015). Lundberg et al. (2014) evaluated the security of different MCIS setups. According to the authors, a mobile setup is not part of an aircraft's onboard system; therefore, the reliability does not meet the required standards of traditional avionics. They tested three mobile setups and found all of them allow an attacker to manipulate information presented to the pilot. To improve the security of MCIS setups, they recommended secure software, regular updates, and a secure data exchange system. To observe the spoofing attack's impact, Manesh et al. (2018) placed a ghost aircraft near to their *own aircraft* position in the Piccolo autopilot. The sudden appearance of a ghost aircraft prompted a rapid descent and steep turn to achieve safety clearance. Eskilsson et al. (2020) demonstrated a low-cost attack setup. Using a Python program, they generated the attack payload, which was subsequently transmitted into the air using a HackRF. Dump1090 software fed by RTL-SDR confirmed the success of the spoofing attack. They expressed

TABLE 2 ADS-B attacks implemented in this study

Attack	Method	Closest related work
Aircraft reconnaissance	Eavesdropping	McCallie et al. (2011); Manesh and Kaabouch (2017)
Spoofing or ghost aircraft	Message injection	Costin and Francillon (2012); Eskilsson et al. (2020)
Flooding	High-level signal jamming	Schäfer et al. (2013); Mccallie (2012)
ADS-B packets DoS**	High-level signal jamming	Schäfer et al. (2013); Li and Wang (2019)
Aircraft disappearance	Message deletion	Schäfer et al. (2013)
Trajectory modification	Message modification	Schäfer et al. (2013)
Logically invalid data encoding	Fake signal	Sjödin and Gruneau (2020)
False distress signal**	Squawk code modification	Schäfer et al. (2013)
Jamming	Low-level signal jamming	Schäfer et al. (2013); Leonardi et al. (2021)
Specific data-link protocol fuzzing*	Fuzzing*	Domin et al. (2016); Kim et al. (2019)
Coordinated attack*	Multiple emitter*	[None]
Specific error-handling test*	Message modification	[None]

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

concern that the availability of cheap attacking equipment may encourage many adversaries to launch attacks. Sjödin and Gruneau (2020) demonstrated data injection and flooding attacks using HackRF and Sentry. They reported that the ADS-B system does not check the data validity; the receiver blindly trusts the protocol. In addition to the attacks mentioned in the literature, we propose several novel attack ideas and practically implement some existing attack concepts. Table 2 summarizes the various types of ADS-B attacks implemented in this study and in the literature.

3.4 ADS-B Experiment Setup

Our heterogeneous test bed consisted of various avionics hardware and software supported by different operating systems. Setups varied depending on the nature of the experiment. For example, in Article PV, 13 hardware devices and 22 pieces of software from 4 operating systems (resulting in 36 configurations) were considered. In Article PVI, fuzz test was done for 16 EFB applications. Forty-four 1090ES and 24 UAT978 MCIS configurations were tested in Article PVII. Details on ADS-B hardware and software and their functionalities are available in Articles PV to PVII. Python programming language was used to create 1090ES attack payloads according to the protocols. We used Yusupov's script (Yusupov, 2021a) to encode altitude and position information. We extended the software's functionality by writing the codes for encoding flight information, velocity, and squawk into the 1090ES signal.

A UAT978 long message generator was also provided by Yusupov (Yusupov, 2021b). To create UAT978 payloads, we used Yusupov's program. We slightly modified Larroque's Reed–Solomon codec to generate the FEC (Filiba, 2020). After that, we created the final payload by adding synchronization bits and serializing all the parts in the correct order. GNU Radio Companion (GRC) software was used to generate the in-phase and quadrature (IQ) of the signal from the payload. The IQs were transmitted on the air using HackRF, BladeRF, and Pluto SDR.

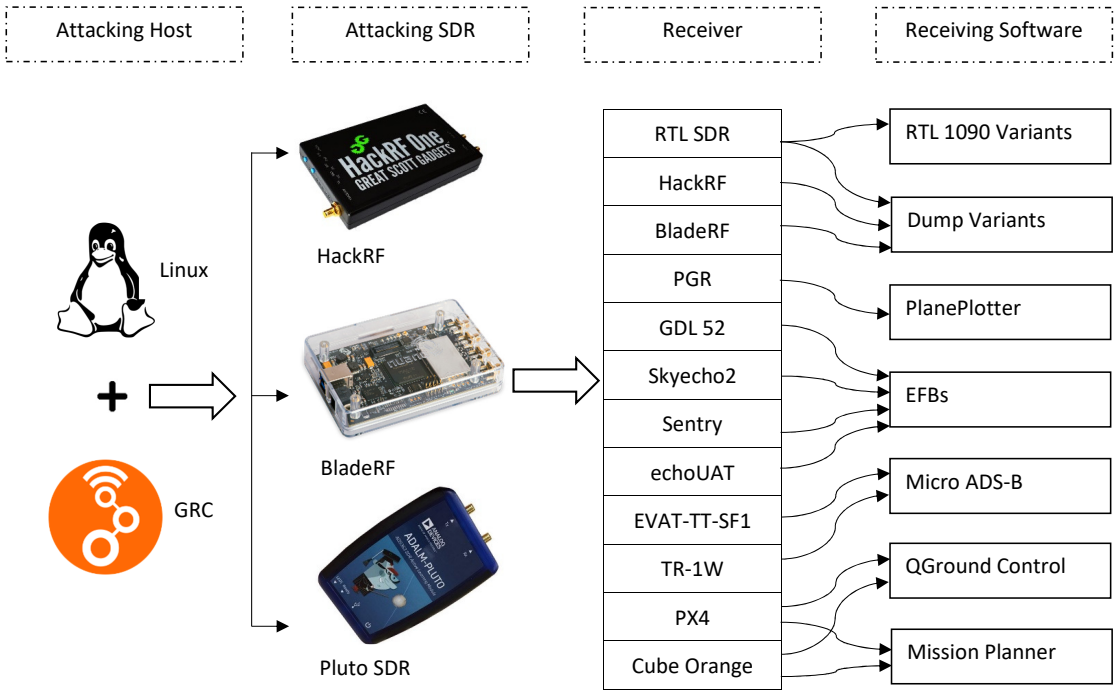


FIGURE 19 Experimental attack setup

Although one transmitter is enough for most of the experiments, we used three of them to check the availability and diversity of attacking devices. Figure 19 shows the experimental setup used in Article PV. Compared with UAT978, ADS-B 1090ES is much more widely used and adopted. Therefore, ADS-B 1090ES was the primary focus of our research. All the attacking scenarios listed in Table 2 were tested for ADS-B 1090ES. However, aircraft reconnaissance, flooding, jamming, protocol fuzzing, and spoofing attacks were considered for the UAT978 tests.

3.5 Practical Attacks on ADS-B

In Articles PV to PVII, we demonstrated and explained some existing and novel attacks on ADS-B in detail. In this section, we summarize them. Some sensitive information in different figures has been blurred.

Aircraft Reconnaissance All the tested receivers received 1090ES signals from the flying aircraft, which is kind of an effortless task. We could not receive any UAT978 signal from aircraft because this is not used in Finland. However, we produced UAT978 signals in the laboratory and successfully tested the reception via supported receivers. Eavesdropping is impossible to avoid because the ADS-B signal is neither encrypted nor requires any authentication. Privacy could be violated via eavesdropping, and spreading the eavesdropped data on the Inter-

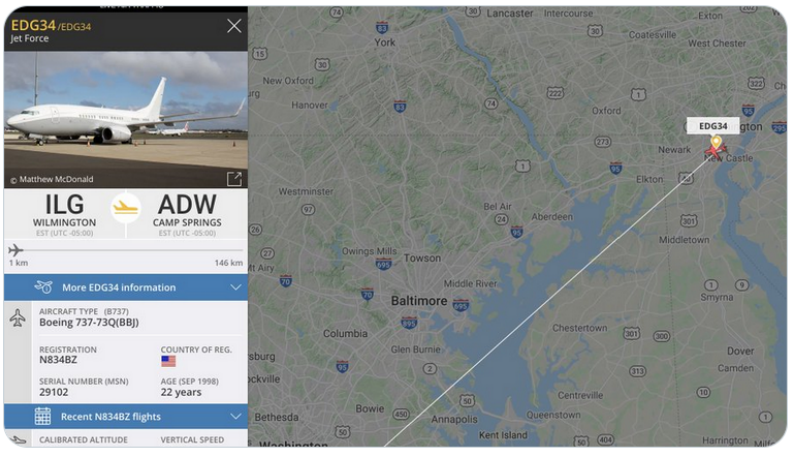


FIGURE 20 President Joe Biden's flight for his presidential inauguration

FIGURE 21 Spoofed aircraft over North Korea

net exacerbates the problem. For example, Joe Biden's trip from Wilmington, Delaware to Washington DC for his presidential inauguration is shown in Figure 20.[1]

Spoofing We successfully produced fake ADS-B 1090ES and UAT978 signals in our laboratory. Without any warning, all ADS-B receivers decoded our spoofed signal. Although spoofing is the most basic and earliest type of ADS-B attack, it can nevertheless represent a severe risk to ATC operations. Figure 21 shows a spoofed aircraft over North Korea.

Flooding Flooding attacks make it impossible to distinguish fake aircraft from real ones. We did not observe any alarm in any of the receivers during the flooding attack. We observed that mobile configurations constrained by memory, processing power, and screen size were more vulnerable to flooding attacks than their desktop counterparts. Figure 22 exhibits a flooded screen.

False Distress Signal We developed a Python script to encode squawk code in the ADS-B signal. When a distress squawk code is transmitted, it immediately

[1] Image courtesy: https://twitter.com/flightradar24/status/1351628618187862026

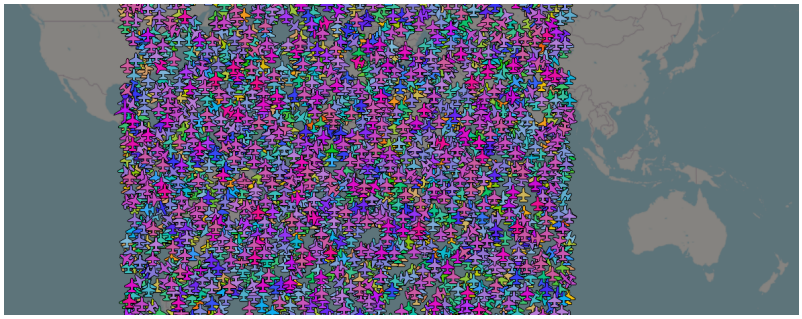


FIGURE 22 The flooded screen of tar1090 software

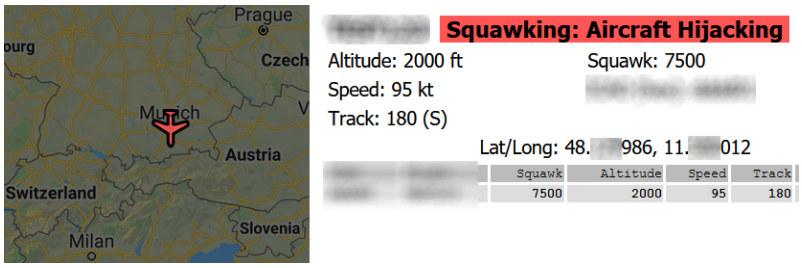


FIGURE 23 Fake squawk code in the Dump1090 net

alerts all the ATCs and planes in the vicinity that the aircraft is facing an emergency. A fake distress squawk code could have serious consequences, such as triggering a false alarm, leading to Air Force deployment. Figure 23 shows a false hijacking alert in dump1090.

Coordinated Attack When multiple attackers/attacking devices coordinately attack a single aircraft, we call it a coordinated attack. Here, we briefly explain the coordinated attack concept. The ICAO24 code is a unique property of an aircraft; therefore, it is used as the reference in ADS-B communication. In subsequent messages, the ADS-B information is displayed and updated against that ICAO24 code. Our attack setup offered us the flexibility to use any ICAO24 code. Using two transmitters, we transmitted two ADS-B 1090ES signals having the same ICAO24 code but differing values in some of the other ADS-B data fields. Because the reference point (ICAO24 code) is the same, on the receiving side, ADS-B information was updated according to the received data. However, some data fields are static and should not be updated throughout a flight, or changes in dynamic data should follow a linear or standard pattern. For example, the flight number should not be updated during a single flight, but when we used two flight numbers against the same ICAO24 code in two transmitters at the receiver end, we observed that the flight number fluctuated every second. In the same way, when we used two (very distant) location coordinates for a single ICAO24 code, the targeted aircraft changed its position from one city to another in an instant. This situation can lead to ATC confusion and can have many dangerous consequences. We present the coordinated attack results for all the ADS-B setups in Article PV, where we considered the attack for each ADS-B data field.

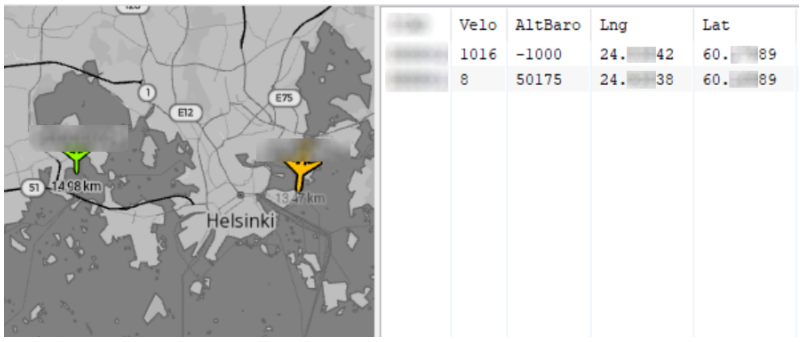


FIGURE 24 ADS-B Micro displays logically incorrect data

We observed that the coordinated attack triggered logical discrepancies at some receivers. For example, some software showed wrong values, some retained the first signal's value, and in some, the flight and velocity information disappeared.

Logically Invalid Data Encoding Although data integrity checking is defined in the ADS-B protocol using CRC, the data validity checking is neglected. Therefore, we were able to encode and pass technically correct but logically invalid data through the ADS-B communication. Figure 24 depicts a high-speed aircraft at a low altitude and vice versa for another aircraft.

Jamming In a jamming attack, an attacker uses a powerful RF signal to overwhelm the communication channel, preventing service to all wireless nodes within range of the interference. This tactic can be used to formulate several ADS-B attacks. Some of them are as follows:

- ***Signal jamming*** ADS-B 1090 uses a 4.6 MHz-wide radio spectrum from 1087.7 MHz to 1092.3 MHz, centering at 1090 MHz (ITU, 2017). On the other hand, UAT978 uses a 1.3 MHz broad spectrum that centers at 978 MHz (ICAO, 2005). Using BladeRF, we were able to block these two bands with white noise transmission, and no receiver received valid ADS-B transmission. In Figure 25, the pink wavy line shows the noise floor during normal time. The greenish-yellow wavy line shows that the noise level significantly increased during the attack. Nevertheless, if the signal jammer is not

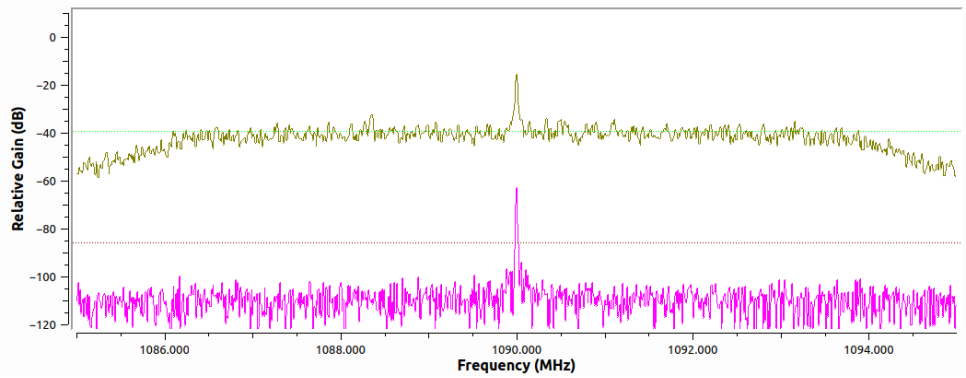


FIGURE 25 Increase in the noise floor due to the jamming attack

very powerful, this type of attack would not be effective in a vast area.

- ***Aircraft disappearance*** Using the jamming and spoofing attack, we formulated the aircraft disappear attack. To begin, using a BladeRF, we jammed the 1090 MHz channel so that the targeted receiver could not receive any valid signal from the flying aircraft. Using a distant RTL-SDR receiver with dump1090, we then collected the legitimate ADS-B data. Using the *"./dump1090 –write-json-every $< t >$"* command in dump1090, we were able to write the data in a file. After that, using a Python program, we created the attack payload based on the data of that file but filtered out the targeted aircraft's data that we wanted to disappear. Finally, we transmitted that attack payload into the air using HackRF in high-power mode. We observed that the targeted aircraft had vanished from the targeted receiver, but the other planes were still visible.

- ***Trajectory modification*** A trajectory modification attack works the same way as an aircraft disappearance attack. However, after disappearing the targeted legitimate aircraft from the targeted receiver, we transmitted a series of false position coordinates of that aircraft. So on the receiver, the aircraft reappeared on a new flight path.

Fuzzing Avionics Protocol Fuzzing is a technique for finding bugs or vulnerabilities in software by providing randomized inputs to programs, which may lead to a crash in the worst case. MCIS setups use a variety of data-link protocols to exchange data between devices and applications. GDL-90 is the most prevalent of these protocols. We performed GDL-90 protocol fuzzing by forming packets with a real protocol-like format, but some parts were malformed by the fuzzing component. The technical details of our fuzz test are available in Article PVI. In our experiment, the American Fuzzy Lop (AFL) Python implementation (python-afl v 0.7.3) was used as a fuzzing framework. We instructed the AFL to send faulty data to the IP address of the mobile device. Of the 16 tested EFBs, 9 either crashed or became unresponsive during the experiment. We present the complete test results in Article PVI.

ADS-B Error-Handling Test ADS-B signals can be entirely or partially distorted by noise. Therefore, CRC is used to check the received signal's integrity. ADS-B 1090ES supports up to 5-bit error correction using a 24-degree fixed generator polynomial (Strohmeier et al., 2013). To test the error-handling capability of ADS-B setups, we randomly flipped some bits and transmitted that message. We observed that most of the setups supported up to 2-bit error correction. We present the complete result of the error-handling test in Article PV. We noticed that all the ADS-B setups took extra time to decode an erroneous message. The test results also indicate that although software plays a vital role in demodulating and decoding data, hardware can sometimes play a role. For example, the ForeFlight application connected to the Sentry did not correct the message, but the same application connected to echoUAT and SkyEcho2 did.

DoS Attack Each application or program can decode only a limited amount of ADS-B signals in a particular period due to processing resources and software design limitations. A massive amount of ADS-B signals from an adversary could exceed that limit, leading to a DoS attack on ADS-B IN functionality. To test the resilience of the ADS-B setups, we burst 30,000 to 100,000 ADS-B signals in a short amount of time. Our DoS attack exceeded the ADS-B IN or receiving capacity of most tested applications/software. As a result, we noticed abnormal behavior by the receivers. For example, some of them crashed, some of their outputs clogged, some setups produced garbage outputs that could not be read, and others dropped messages (e.g., did not detect, process, or show all the transmitted messages). We present the complete DoS attack results in Article PV. We extended the investigation of the DoS attack impact on popular MCIS devices, such as Garmin GDL 52, ADL 180, and SensorBox. Most of the configurations were affected by the attack. Detailed DoS attack results on MCIS are presented in Article PVII. We also looked into how DoS attacks affected ADS-B OUT's performance. Among the tested MCIS devices, SkyEcho2 and echoUAT had ADS-B OUT functionality. SkyEcho2 could transmit 1090ES signals, while echoUAT supported the sending of UAT978 signals. We performed ADS-B IN DoS in these two devices by bursting 10,000 ADS-B signals and checked their ADS-B OUT performance. The DoS attack on ADS-B IN reduced approximately 15% ADS-B OUT capacity of SkyEcho2, whereas no significant impact on the echoUAT was observed.

3.6 Countermeasures

Over the last decade, researchers have proposed various techniques to secure ADS-B communication (Huang et al., 2013; Finke et al., 2013; Ghose and Lazos, 2015; Kim et al., 2016, 2017; Yang et al., 2019; Wu et al., 2020). All the offered solutions have some benefits and drawbacks. Some solutions are easy to implement, whereas some need extensive infrastructure. We studied the RSS-distance model and Doppler effect solutions in a practical manner.

3.6.1 RSS-Distance Model

Due to attenuation, an RF signal weakens the farther it travels. Therefore, the signal strength and the distance traveled are correlated. This relationship can be used to verify the distance of a signal's source (e.g., an aircraft) from a reception point. To create an RSS-distance model, we recorded the 3D distance and ADS-B signal RSS of aircraft from our laboratory for three days. Figure 26 shows the created model. The red line plots the raw measurements. Because the raw measurement was noisy, we used a Kalman filter to smooth it out. We then applied the Python-based *scipy.optimize.curve_fit* function to create the final model. Again, we recorded measurements from real aircraft for three days to distinguish

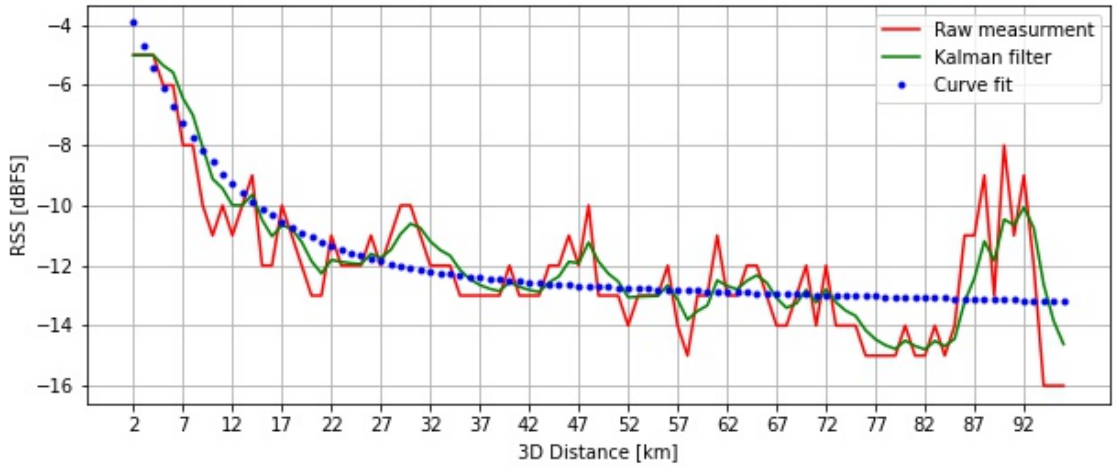


FIGURE 26 Created RSS-distance model

the real aircraft from the spoofed aircraft. During this period, our attack setup randomly transmitted fake signals. The receiver captured both spoofed and real signals. A total of 2,107 test samples were collected during the experiment. Of them, 966 were from real aircraft, and 1,141 were from attackers' spoofed aircraft. From the given location coordinate in the ADS-B message, the distance of an aircraft from our laboratory was calculated. We then retrieved the possible RSS value for that distance from the model. If the retrieved RSS and the real-time RSS were close enough, the aircraft was considered legitimate; otherwise, it was considered a fake aircraft. Generally, the RSS of a signal is affected by noise, temperature, humidity, and other factors. Therefore, we employed some tolerance when comparing the retrieved and actual RSS values.

The power level of an attack signal is uncertain, so we considered three scenarios—low-power attack (LPA), medium-power attack (MPA), and high-power attack (HPA). The RF output gain in the GRC script was set at 10 dB, 20 dB, and 30 dB for these three attacks, respectively. We classified the experiment outcomes into four categories—true positive, true negative, false positive, and false negative. We calculated accuracy, precision, recall, and F1 score based on these four categories to better understand the model's spoofing detection capabilities. The accuracy metric in Figure 27 reveals that compared with LPAs or MPAs, HPAs are relatively easy to detect. Attack detection precision also remains low in LPAs. Recall indicates how many predictions were accurately categorized. The recall ratio decreases when the tolerance is large. The F1 score reveals a combined result of precision and recall, which is found to be best during HPAs, in addition to a high tolerance. From Figure 27, the overall understanding is that attacks are relatively easy to detect when the RSS is strong, and weak signal attacks are prone to erroneous detection.

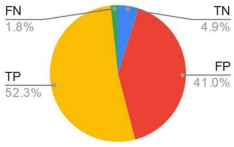
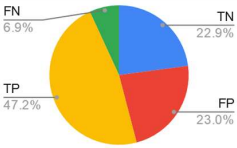
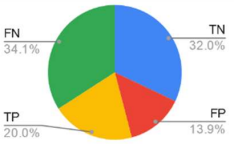
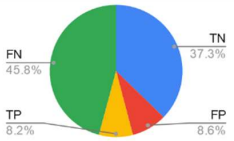
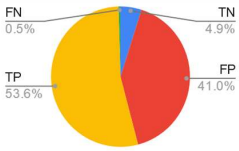
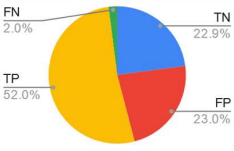
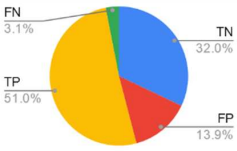
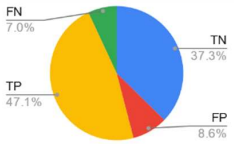
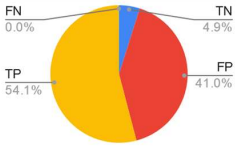
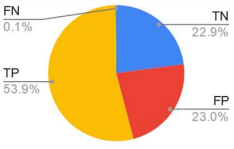
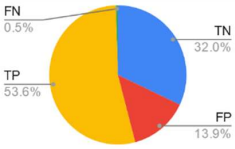
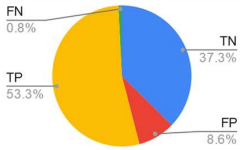
LPA			
Tolerance = ± 1 dBFS	Tolerance = ±2 dBFS	Tolerance =± 3 dBFS	Tolerance = ±4 dBFS
			
Accuracy = 0.57 Precision = 0.56 Recall = 0.97 F1 score = 0.71	Accuracy = 0.70 Precision = 0.67 Recall = 0.87 F1 score = 0.75	Accuracy = 0.52 Precision = 0.59 Recall = 0.37 F1 score = 0.45	Accuracy = 0.45 Precision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

FIGURE 27 Model's spoofing signal detection performance

3.6.2 Doppler Shift

The Doppler shift is the change in frequency of a signal due to motion. Some studies have suggested using the Doppler shift to verify the signal source motion (Ghose and Lazos, 2015; Schäfer et al., 2016). Thus, an attack signal from a static source can be identified. We developed a GRC script to record the strongest RSS and its position in a fast Fourier transform (FFT) display. The FFT size was set to 32,768, and the sample rate was set at 250,000; this resulted in 7.62 Hz per FFT resolution. Figure 28 shows the recorded data from a flying aircraft. The lower part of the figure shows that RSS increased as the aircraft approached the receiver and vice versa. However, during the pass of an aircraft, we did not observe any significant change in frequency (in the upper part of the figure). Despite many attempts, we were unable to identify a good frequency change pattern. Therefore, we conclude that it might be challenging to use the ADS-B signal's Doppler shift effect as a reliable indicator of the motion of an authentic ADS-B transponder versus a static attacker. Moreover, an attacker can also be in motion, for example, using a drone.

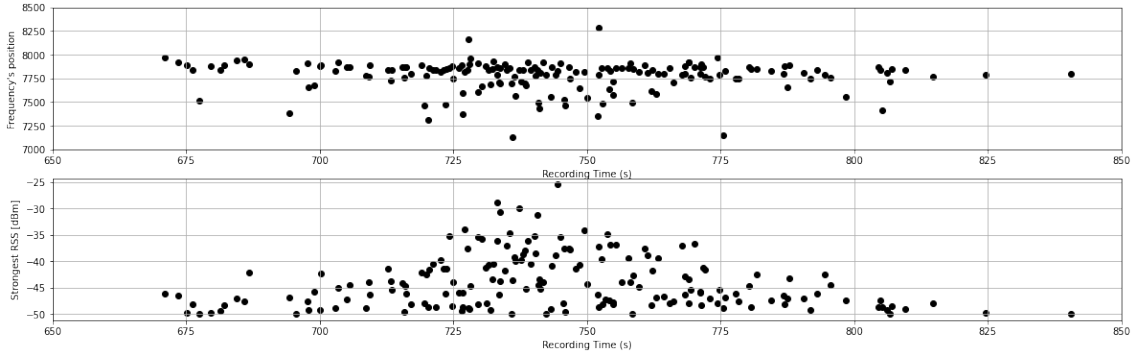


FIGURE 28 Doppler shift evaluation experiment

3.7 Summary of ADS-B Study

This chapter investigates different aspects of ADS-B. The experiment result shows that ADS-B is exposed to severe security problems. The insecure protocol design can be identified as the main culprit. Poor software design also can be blamed in many cases. As the system is already deployed in many parts of the world, making any significant change in the protocol would be very costly and cumbersome. Instead, some back-compatible solution would be realistic. We practically investigate the RSS-distance model and Doppler shift to distinguish the real and fake signals. The first approach shows a 90% success rate in the best case, while the latter one did not show good performance. However, seeking solutions based on radio signal characteristics is the first line of defense. We suggest relevant stakeholders to review the protocol and work together to design a back-compatible solution to ensure the safety of ADS-B against cyberattacks such as those demonstrated in this thesis. The next chapter investigates the security aspects of ship localization.

4 AUTOMATIC IDENTIFICATION SYSTEM

Ships sailing across isolated waterways are exposed to unique challenges, such as rough weather, pirate attacks, and collision with other vessels. To minimize the risk, the IMO announced the mandatory installation of the AIS for ships exceeding 300 tons of gross weight and for all passenger ships, regardless of size (IMO, 2004). This chapter contains the details of our AIS research.

4.1 AIS overview

The AIS is a ship identification system that works by attaching a transponder to the vessel that periodically transmits the ship's name, call sign, location, speed, and other navigation-related data using a radio signal. It plays an important role in integrating navigation, collision avoidance, maritime supervision, accident investigation, search and rescue operations, and weather forecasts. Although technically and operationally distinct, the AIS can be compared with the ADS-B system of aviation. An AIS-equipped vessel, base station, or satellite can receive a signal, and the received data is typically displayed by chart-plotting software. Due to its easy installation and effectiveness, the AIS is gaining immense popularity. From big ships to small leisure crafts, all are being equipped with the AIS. According to the popular vessel-tracking service provider, approximately 570,000 vessels are fitted with the AIS (Vessel-Finder, 2021).

Due to the earth's curvature and antenna height, the typical range of AIS signals is limited to approximately 40 nautical miles. However, various organizations have been experimenting with detecting AIS communications using satellite-based receivers known as SAT-AIS since 2005 to expand coverage. In 2009, ORBCOMM launched AIS-enabled satellites to demonstrate the ability to collect AIS messages from space. On July 12, 2010, the Norwegian AISSat-1 satellite was launched into polar orbit to improve surveillance of maritime activities in Northern Europe. On April 20, 2011, the Indian Space Research Organisation launched Resourcesat-2 containing an SAT-AIS payload for monitoring maritime

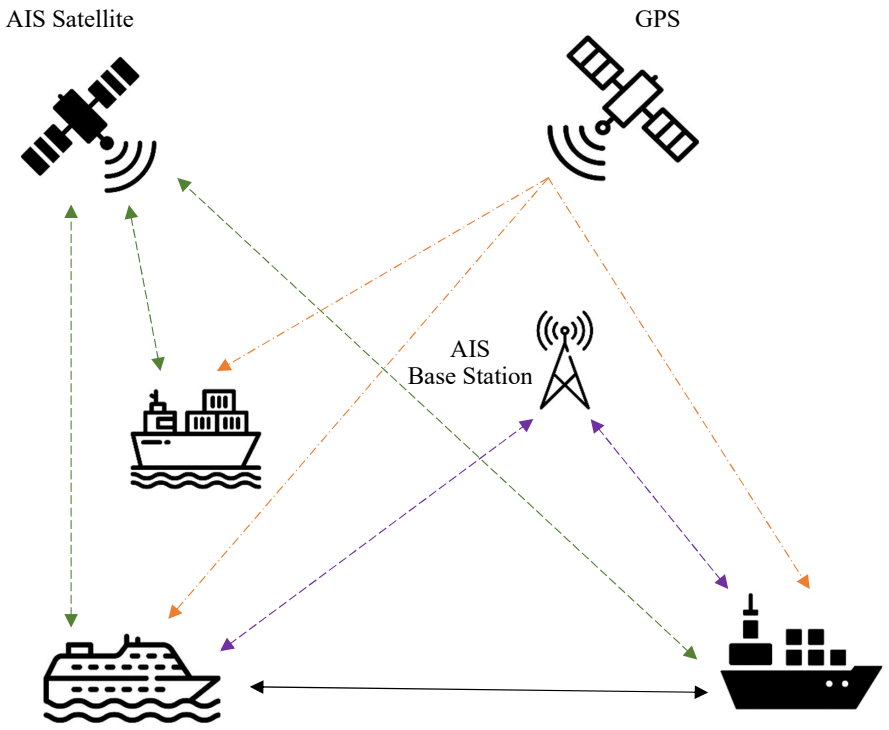


FIGURE 29 AIS communication concept

traffic in the Indian coastal area. Some satellites from the USA, Canada, Denmark, Norway, and India are already in orbit to expand the AIS service. However, the significant quantity of simultaneous AIS signals within a satellite's vast footprint is the fundamental challenge with SAT-AIS. The AIS uses a time division multiple access (TDMA) radio access scheme where 4,500 time slots are available each minute on two channels. These limited time slots can be overwhelmed by the number of AIS transmitters within the satellite's reception footprints, resulting in message collisions. Nonetheless, many types of research are being undertaken to develop efficient SAT-AIS with constellations of nano-satellites. Figure 29 shows the AIS communication concept. Different vessels can exchange information directly or via a base station or satellite, as shown in Figure 29. A total of 64 types of AIS messages are defined in the protocol, such as position report, base station report, voyage-related report, etc. Of these, 27 are currently in use, and the other 37 are reserved for the future. A complete list of AIS messages is available in (Coast-guard, 2021).

4.2 RF Characteristics

In Section 4.1, we have seen that the AIS can be operated through terrestrial and satellite-based communication. Because the terrestrial-based AIS is the most popular and currently operational, we shall discuss the terrestrial-based AIS in this

section. The AIS uses two marine band channels. The channels are as follows:

1. Channel A at 161.975 MHz
2. Channel B at 162.025 MHz

Dual channels are used to mitigate RF interference, enhance link capacity, and allow switching of the channels without communications loss from other ships. The AIS uses the TDMA channel access method, where each ship is assigned a particular time slot or frame for the transmission (ITU, 2014). Generally, a single message is transmitted in each time slot; however, multiple slots can sometimes be used for single message transmission. According to the protocol, a frame length is 256 bits, and the data transmission rate is 9,600 bits/second. Thus, the timing limit of a frame is $256/9600 = 26.66$ milliseconds, which results in 2,250 slots per minute per channel or 4,500 slots per minute on both channels. Gaussian minimum shift keying (GMSK) modulation is used to modulate the AIS RF signal with a bandwidth-time product of 0.4. Non-return-to-zero inverted (NRZI) encoding is used to encode the data. An AIS frame structure is shown in Figure 30.

Ramp-up 8 bit	Preamble 24 bit	Start flag 8 bit	Payload 168 bit	CRC 16 bit	Stop flag 8 bit	Buffer 24 bit

FIGURE 30 AIS frame structure

4.3 Security Issues

Like ADS-B, AIS also lacks basic security measures, such as encryption or authentication, which makes it vulnerable to cyberattacks. There have already been some incidents of exploitation of the AIS. According to an Internet ship tracking site, a British warship was spotted near Sevastopol, Crimea, escalating tensions between Russia and the UK (Bateman, 2021). However, an onboard camera showed that the ship was approximately 300 km away. North Korean vessels often change their ship identity to avoid sanctions (Zwirko, 2019). Near the Strait of Hormuz, unknown entities falsely claimed to be US warships (MarineTraffic, 2019).

These incidents indicate that the AIS has already been exploited at the military level. In academia, however, only a few researchers have looked into such dangerous security weaknesses (Mathapo, 2007; Larsen et al., 2011; Balduzzi et al., 2014; Marques et al., 2019; Cruz et al., 2018; Androjna et al., 2021). Among them, only Balduzzi et al. (2014) demonstrated AIS packet-level attacks, while others looked into the decoding and the possibility of building a cheap transponder using SDR. Balduzzi et al. demonstrated some attacks using fake AIS signals, such as service availability disruption, false alerts, spoofing, etc. Since their

study, technology has advanced dramatically; new hardware, software, transponder, and mobile applications have been developed. However, attackers have also become smarter with new hacking ideas and tools. Therefore, evaluating attacks on modern AIS setups is very important.

4.4 AIS Experiment Setup

Our AIS test bed includes a commercial transponder, a professional AIS receiver, some RTL-SDR-based mobile receivers, and two attacking SDRs, resulting in 19 AIS setups. A list of hardware and software and their functionalities is presented in Article PVIII. We acquired a maritime mobile service identity (MMSI) number from the Finnish transport and communications agency for this experiment. Most of the previous studies used *AISTX* to generate the AIS payload (TrendMicro, 2014). However, the original version of *AISTX* produces only a single AIS frame at a time, which is very slow to test attacks such as DoS or flooding. To eliminate this limitation, we modified *AISTX* to produce a user-defined number of AIS frames using a single command in a file. GRC was used to produce the IQ samples based on the data of the file. HackRF and BladeRF transmitted the IQ samples in the air.

The attack impacts were evaluated using a Matsutec HP-33A AIS transponder, a Quark-elec QK-A027 AIS receiver, Windows-based ShipPlotter (COAA, 2021), and OpenCPN (OpenCPN, 2021) software, as well as several Android applications. Apart from the HP-33A transponder, all other setups used QK-A027 or RTL-SDR as the RF front end. A program called SDR Sharp was used on the Windows platform to tune the AIS signal. The resulting audio was fed to AIS-Mon (MarineTraffic, 2021), which decoded the AIS signal. The decoded data was provided to the OpenCPN using a UDP port. Another Windows-based program, ShipPlotter, had a built-in decoder, so it did not need AISMon but only the audio from the SDR Sharp. The RTL AIS driver application performed the decoding task on the Android platform. The decoded messages were shared with different navigation applications by another application called AIS Share (EbcTech, 2021). The QK-A027 has a built-in decoder and could share the decoded data through a TCP port. Therefore, the Android applications did not require additional decoding software while using QK-A027. We could not configure *Ships v 4.07* with QK-A207 because that application does not support a TCP connection. Figure 31 shows the experimental attack setup. Modified *AISTX* supplied the attack payload to GRC, where the IQs of the signal are generated. HackRF and BladeRF were used to transmit the IQs into the air. We could transmit the signal on both AIS channels by switching the frequency. Furthermore, using two attack SDRs, we were able to transmit the AIS signal on both channels at the same time. All the receivers received, demodulated, decoded, and displayed the AIS data.

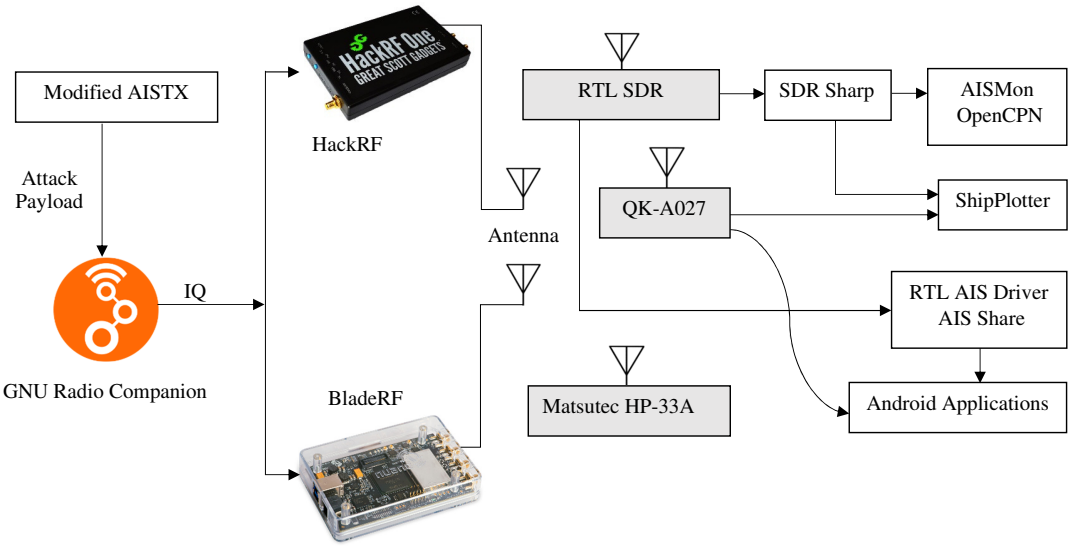


FIGURE 31 AIS attack setup

4.5 Practical Attacks on the AIS

In Article PVIII, we demonstrated 11 attacks on/tests of the AIS. This section summarizes the observed effects on receiving hardware and software. Some sensitive information in different figures has been blurred.

Spoofing Our counterfeit signal resulted in fake ships in all the receivers, as already accomplished by Balduzzi et al. (2014). AIS spoofing could have severe consequences (Bateman, 2021). For example, the sudden presence of enemy ships in a country's territory could lead to military deployment. Figure 32 depicts a fake ship on the Helsinki Airport runway.

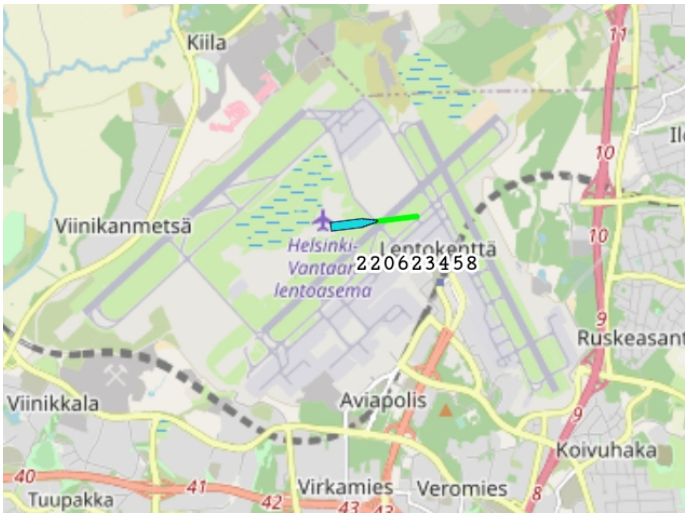


FIGURE 32 A spoofed ship

FIGURE 33 Matsutec HP-33A AIS transponder showing an MOB alert

FIGURE 34 Collision alert in ShipPlotter

Man Overboard In the AIS, the Man Overboard (MOB) alert is used when a person falls off a ship and needs to be rescued immediately. The rescue workers also use it during their operations. A small beacon is used to transmit the MOB signal containing the beacon's location. A type 1 message with navigation status 14 and an MMSI starting with 972 is used in a MOB signal. Figure 33 shows our fake MOB alert. This kind of false alert could trigger costly rescue operations and waste time.

Collision Alert From the AIS data, the system always calculates the closest point of approach (CPA) and the time to the closest point of approach (TCPA) values for other ships to determine any possible collision. Using our program, we could encode any location for a fake ship. We, therefore, placed a fake ship close to the *own ship's* location in different software to observe the effect. We noticed that when CPA and TCPA values fell below the threshold, it triggered a collision alert. Using this kind of false collision alert, attackers could try to change the course of a ship to bring it into their area of interest. Figure 34 shows a false collision alert.

Jamming Generally, a jamming attack is executed by transmitting overpowered white noise to a radio channel so that valid data get distorted. In our laboratory, we successfully performed this attack. Despite the success in a controlled environment, such an attack would have limited impact in large water areas. However, using valid AIS data, we could flood the system so that there would be no available slot for the legitimate data. The TDMA scheme in AIS sets 2,250 time slots per minute per channel. This means a receiver would be able to receive AIS

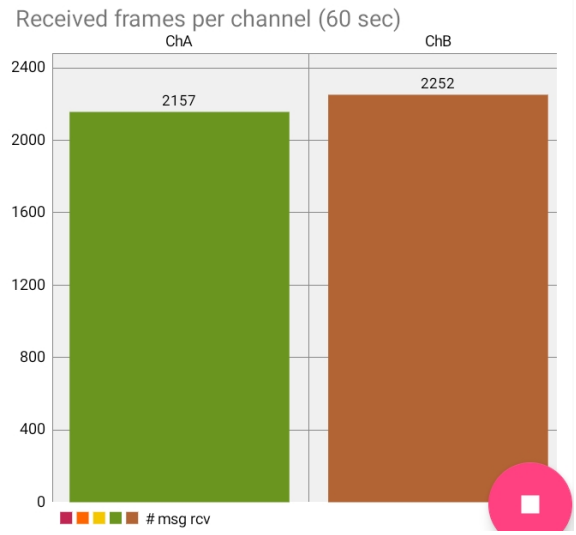
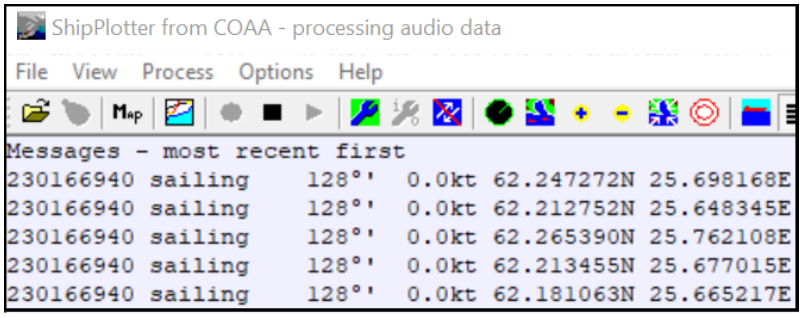


FIGURE 35 Signal-receiving statistics for one minute in AIS Share

signals from a maximum of 2,250 ships in a minute per channel. We generated two files containing a huge number of AIS messages using the modified *AISTX*. Through HackRF and BladeRF, we transmitted those two files to both AIS channels. Figure 35 shows signal reception statistics in AIS Share. The two attacking SDRs consumed roughly 96% of channel A's capacity and 100% of channel B's capacity. Thus, the channels can be jammed with valid AIS packets.

Overwhelming Alerts Previously, we saw there are two types of alerts in the AIS, collision alerts and MOB alerts. We triggered 1,000 collision alerts and the same amount of MOB alerts using counterfeit signals. We noticed that in some software, thousands of audio-visual alerts created a chaotic situation called "alert fatigue" (Hassan et al., 2019). A true positive alert may remain unnoticed in such a situation. Furthermore, the ShipPlotter software crashed as a result of this attack. We also noticed that some applications did not put out any alert at all, which could lead to dangerous consequences in case of an actual alert situation. We present the details of the overwhelming alerts attack results in Article PVIII.

Coordinated Attack We conducted a coordinated attack in the AIS in the same way as we did in ADS-B. Like the ICAO24 code in ADS-B, the MMSI number is used as the reference by the AIS software. In successive messages, information on a particular ship is updated against the MMSI number. Using HackRF and BladeRF, we transmitted two AIS signals on the same channel. In both signals, only the MMSI number was the same, and the rest of the data fields (e.g., ship name, call sign, position, speed, navigation status, vessel type, and dimension) were different. On the receivers, we observed that all the AIS data started to fluctuate during the attack, showing the alternate values every second as those were encoded into two transmitted signals. This fluctuation may lead to confusion in Vessel Traffic Services (VTSs) or among other ships. For example, a cargo ship could be shown as a passenger ship or the position of a vessel could change from



(a) Zero-speed ship moving position in ShipPlotter

AIS target list

MMSI	Name	Call	SoG	CoG	Type	Nav Status
[redacted]	ABC	XYZ123	39.9	350	Tanker	Under way sailing
[redacted]	ABC	XYZ123	96.9	262	Cargo Ship	High Speed Craft
[redacted]	ABC	XYZ123	40.1	301	Passenger Ship	Power-driven vessel

(b) Different ships having the same name and call sign in OpenCPN

FIGURE 36 Illogical AIS data in different software

one place to another in an instant on the AIS screen, thus producing dangerous consequences. Coordinated attack results for the important AIS data fields are presented in Article PVIII.

Logically Invalid Data Encoding During the signal transmission and reception, some data bits may become affected by the noise. Therefore, AIS uses data integrity checks using CRC. However, it does not check the validity of the data. It is possible to encode technically accurate but logically incorrect data. For example, Figure 36(a) shows a ship changes its position, but its speed remains zero. Figure 36(b) shows that three vessels have the same name and call sign, although their speed, course, and types are different.

Error-Handling Test According to the protocol, the AIS uses CRC for error detection (ITU, 2014). To test the error detection capability of our test bed AIS setups, we flipped a data bit and transmitted that frame via an SDR. We noticed that all the test setups detected that error and dropped that message. Figure 37 shows error detection in the AISMon software. The occurrence of an error in a radio transmission is a regular incident. Therefore, along with error detection, ADS-B uses an error-correction mechanism. However, we found that the AIS uses only error detection but no error correction. An error-correction system could save the AIS signals from being corrupted by noise on the channel, thus decreasing RF pollution. We recommend at least a 1-bit error-correcting scheme.

Visual Navigation Disruption Counterfeit AIS transmission can seriously disrupt AIS-aided visual navigation. For example, AIS base stations are stationary and are generally located on the coast. Using the same MMSI number in some type 4 messages and changing the position coordinates values, it is possible to

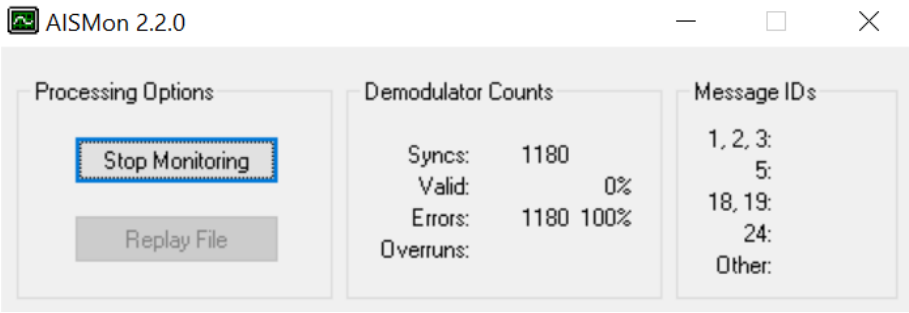


FIGURE 37 Error detection in AISMon

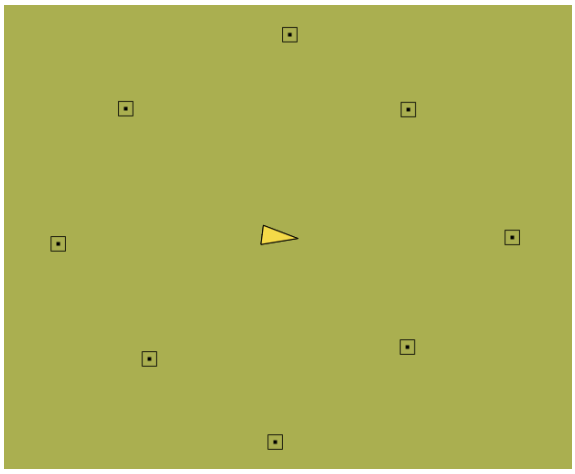
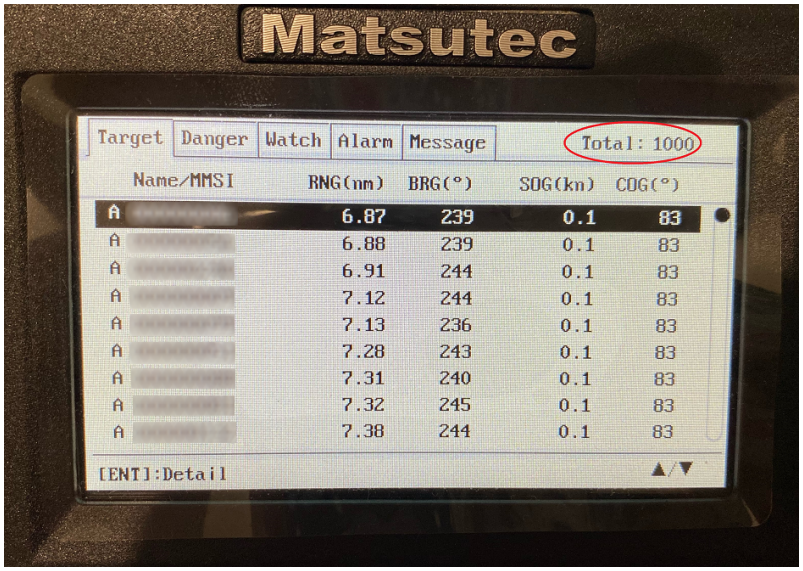


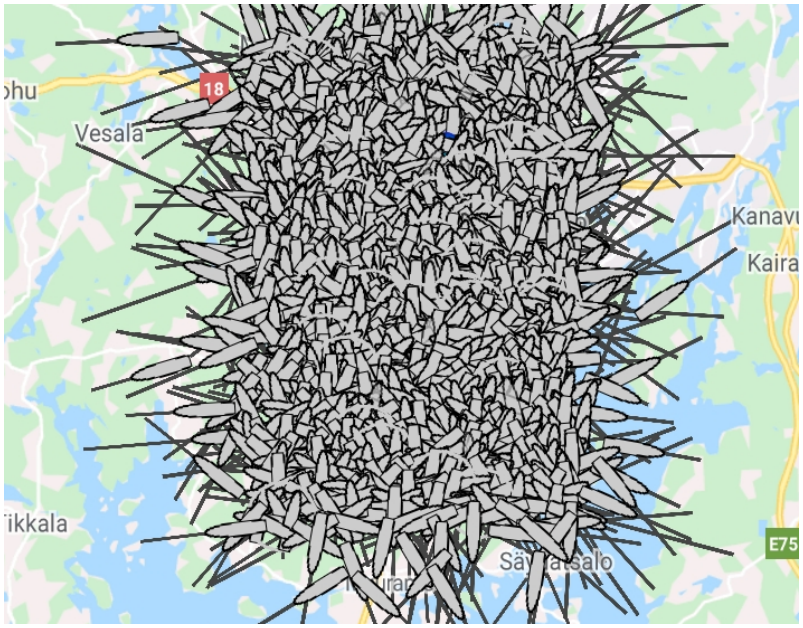
FIGURE 38 Fake base stations encircling a ship in the OpenCPN software

create such a situation where it seems the base station is moving towards a ship. In another scenario, we encircled a ship with stationary base stations, shown in Figure 38.

DoS We tested the resilience of all our test bed AIS setups using a DoS attack. The attack was performed by transmitting a massive amount of AIS signals. Because our program could produce a user-defined number of AIS messages by a single command in a file, we created 200,000 type 1 messages with different MMSI numbers. We then transmitted those signals into the air using a HackRF. Due to the fixed computing resources and software design, almost all the receivers can decode and display up to a limited number of AIS signals. For example, the Boat Beacon application can display approximately 100 MMSIs, and the Matsutec transponder can display 1000 MMSIs. When the amount of received signals exceeds an AIS setup's capacity, some abnormal behaviors are observed. Some AIS setups crashed during our DoS attack, and some clogged or became unresponsive. The attack impacted approximately 89% of the setups. The detailed result of the DoS attack is presented in Article PVIII. Figure 39(a) shows the Matsutec transponder's maximum display capacity. Figure 39(b) shows a flooded screen of the Boat Beacon application.



(a) Clogged screen of Matsutec HP-33A



(b) Flooded screen of Boat Beacon

FIGURE 39 DoS and flooding attack in the AIS

AIS Preamble Test During our experiments, we noticed that the QK-A027 receiver did not receive an AIS signal from HackRF or BladeRF, but it received a signal from the Matsutec transponder. Therefore, we thoroughly investigated the matter and found a preamble-related issue that we call an "AIS preamble-related implementation flaw."

A 24-bit preamble alternating zeros and ones is used in the AIS (ITU, 2014). According to ITU (2014) Annex 2, "this sequence may begin with a 1 or a 0 since NRZI encoding is used." However, in the same document, Annex 7 says, "this sequence always starts with a 0." Annex 2 focuses on the self-organized time division multiple access (SOTDMA), which is mainly used by class A AIS de-

vices on commercial vessels. Annex 7 focuses on the carrier sense time division multiple access (CSTDMA) used primarily by class B shipborne AIS stations on leisure boats or lighter ships. However, regardless of class, if ships are within the AIS range, they must be visible to one another via AIS as per requirements (ITU, 2014). Therefore, the International Electrotechnical Commission (IEC) standards IEC 62287-1 (IEC, 2017a), IEC 62287-2 (IEC, 2017b), and IEC 61993-2 (IEC, 2012) require all *equipment under test* to receive the AIS signal regardless of whether the preamble begins with 0 or 1.

Allowing the transmission preamble to start with 1 and expecting the reception preamble to start with 0 could cause some confusion. For example, the QK-A027 receiver recognized the AIS signal when the preamble started with 0 but dropped it when the preamble began with 1. Indeed, the way AIS preamble instructions are given in the specification can easily mislead the designer. The instruction says the AIS uses NRZI encoding so that the preamble may start with 1 or 0. Therefore, we checked the situation after NRZI encoding. The NRZI encoding has some variants, and the AIS uses the non-return-to-zero space (NRZS) variant of NRZI, where the waveform changes due to the occurrence of 0. The NRZI encoding also depends on the previous bit level, so we also considered the ramp-up (*000000000*) bits during the investigation. Figure 40 shows that the two waveforms are not exactly the same after the NRZI encoding. A MATLAB code by Dembovskis (2015) also showed that a preamble starting with 0 and a preamble starting with 1 do not yield the same results after NRZI encoding. The detailed results of the AIS preamble test are presented in Article PVIII.

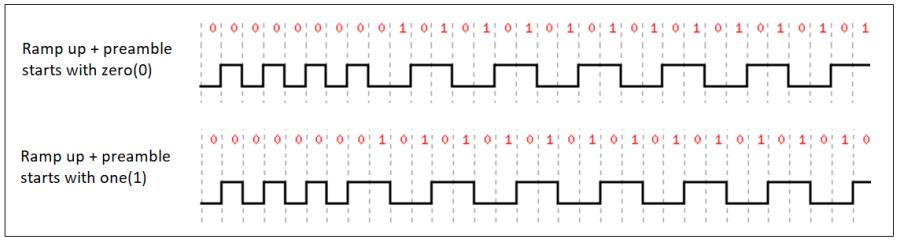


FIGURE 40 NRZI conversion of ramp-up bits and preamble bits

4.6 Defence Against Attacks

Several attacks on the AIS are practically demonstrated in Section 4.5. Such attacks can affect the integrity, authenticity, confidentiality, and availability of AIS information (Parker, 2012). Therefore, defense against such attacks is critical. More systematic research is needed to implement an effective defense strategy, which is beyond the scope of this thesis. However, we set that as our future work. Various researchers have proposed some solutions in the literature. We briefly discuss these below.

To secure the AIS data, Su et al. (2017) presented a digital certificate-based

identity authentication mechanism. In this process, a vessel should generate both private and public keys. Later, the public keys can be distributed through a trustworthy organization called the Certification Authority (CA). If a vessel wants to hide its identity from the CA, it should send several digital signatures to the CA. Upon receiving them back, the vessel selects one randomly. Goudossis and Katsikas (2018) also proposed a similar type key-based solution. They suggested that IMO and National Maritime Authorities (NMA) be involved in private and public key generation, respectively. For remote and insecure areas, such as the coast of Somalia, a micro-satellite or military patrol boat can be involved in the key distribution. A study by Bothur et al. (2017) found that all the electronics systems of a ship are susceptible to cyberattacks. Although they did not specifically mention any solution to the AIS vulnerabilities, they think a proper policy and protecting the security of data, applications, hosts, and networks could help to prevent such attacks. Leite et al. (2021) used an image processing technique to detect AIS attacks. Upon receiving the signal, AIS software plots a ship icon on the map, which changes the mean intensity of pixels in that area. The authors compared the display pixel intensity against a threshold. If the pixel intensity went above that limit, it activated the alerting system. They claimed a 93% success rate. Sciancalepore et al. (2021) proposed an authentication framework called "AuthAIS". They used a type 8 message to exchange cryptography-related data. In their framework, the precise timing of AIS signal transmission of a ship remains hidden, which is shared with other ships by a trusted third party. An attacker is unlikely to know a ship's AIS transmission schedule. Thus, the counterfeit signal transmitted at the wrong time would not be authenticated by other ships.

4.7 Summary of AIS Study

Several existing and novel attacks on the AIS have been demonstrated in this chapter. Our research result shows that cyberattacks on the AIS could be potentially harmful and threaten the safety of ship surveillance systems. The insecure protocol design can be blamed for this problem. We suggest basic security measures, such as authentication and encryption in the AIS protocol to make it secure and resilient against cyberattacks. However, because the AIS has been deployed and operational since 2004, making any significant change in the protocol would be very costly and cumbersome. Instead, some back-compatible solution would be realistic. The AIS message types 28 to 64 are reserved for future use. Some of them can be used to exchange security-related data. Due to time and scope limitations, we could not investigate the implementation of security features of the AIS in this thesis, which we plan for our future research. Relevant stakeholders, researchers in academia, and the industry can effectively use our approach and outcomes to investigate and improve the security of the AIS.

5 CONCLUSION

In this IoT era, all devices need to determine and share their position to be part of a network or to access location-based services. This thesis investigated localization challenges and security in three types of networks—Wi-Fi, aviation, and maritime networks.

Numerous things affect the Wi-Fi fingerprinting positioning accuracy. To improve the accuracy, we investigated different aspects of the FPS in this thesis. First, we studied the device diversity effect in Article PI. When the fingerprints are collected in a crowdsourced manner, keeping the reference of the fingerprint-recording device type could help to have less RSS deviation and thus better positioning accuracy. We proposed a D2D communication-assisted FPS in Article PII. Sharing the fingerprints and positioning experience with other devices resulted in 44% fewer positioning errors in our experiment. Because the FPS often suffers from a large number of errors, we developed a real-time error prediction technique in Article PIII. Instead of one, we calculated the locations of multiple test fingerprints. If all the calculated locations are close together, it indicates the reliability of the calculation. Thus, based on the estimated position cluster radius, a possible error can be predicted. We analyzed the RSS quantization in Article PIV. Based on the experiment on five publicly available databases, we demonstrated that 4-bit quantization is sufficient for the FPS. If a quantized RSS fingerprint can carry the key properties of a radio environment, it is satisfactory for fingerprint-based localization. In the FPS, along with RSS, AP references are also used, which helps to distinguish one fingerprint from another even if low-bit quantization is used. Our proposed method could simplify the hardware configuration, improve security, and reduce approximately 40–60% of storage space and data traffic. Nonetheless, as the GNSS is not designed for indoor localization and therefore provides poor performance in the indoor environment, the challenge of developing a global-scale indoor positioning system remains unsolved. Like other methods, the FPS also faces many challenges (as mentioned in Section 2.2). It is hard to find an effective universal solution for all problems. We need to apply customized methods depending on the nature and available resources of the network and environment.

In the aviation sector, ADS-B plays a major role in broadcasting an aircraft's location and other flight-related information to another aircraft and the ATC tower. Through various online services (e.g., flightradar24[1], flightaware[2]), it is possible to track down almost all aircraft. Our study in Articles PV to PVII found that ADS-B is insecure and exposed to cyberattacks. In Article PV, including 5 novel attack concepts, we practically demonstrated a total of 12 attacks. We examined in depth the security of avionics protocol GDL-90 in Article PVI, and found that around 56% of tested EFB applications crashed or became unresponsive due to the fuzzing attack. MCIS devices were extensively tested in Article PVII. The total number of 1090ES and UAT978 setups were 44 and 24, respectively. The DoS attack affected approximately 63% and 37% of them, correspondingly. We noticed that the DoS attack reduced the ADS-B OUT efficiency of the Skyecho2 device by 15%. The result shows that computing resource-constrained MCIS installations are more vulnerable to cyberattacks than powerful transponder or desktop setups. Our attacks on/tests of ADS-B reveal an alarming picture regarding its security. Many kinds of attacks were implemented at a low cost and with little effort. Moreover, knowledge and attack resources are easily obtainable. If proper security measures are not considered in the protocol, it would be nearly impossible to stop such attacks. Defense measures using signal strength or Doppler shift are the first line of defense, which would not be sufficient against intelligent attacks. However, because ADS-B has already been deployed in many parts of the world, it would be cumbersome and costly to make any significant changes in the protocol. In our opinion, relevant stakeholders, such as ICAO, FAA, the European Aviation Safety Agency (EASA), and the European Organisation for the Safety of Air Navigation (EUROCONTROL), should pay attention and work together to design a back-compatible solution to ensure the safety of ADS-B against cyberattacks such as those demonstrated in this thesis.

In maritime transportation, vessels use the AIS to share their location and other navigation-related data with different maritime entities. In Article PVIII, we demonstrated that, like ADS-B, the AIS is also exposed to cyberattacks. We practically implemented 11 attacks/tests against 19 AIS configurations, where most of the attacks were found to be effective. In addition to existing attack concepts, such as DoS and jamming, some novel ideas such as a coordinated attack and overwhelming alerts indicate that a multi-million dollar ship's navigation security can be affected by a low-cost attack setup. Insufficient security measures during the protocol design can be blamed for these vulnerabilities. Along with the AIS, big commercial vessels use radar for navigation; this is why no major physical accident has happened yet due to an AIS attack. However, a secure AIS protocol is needed to reduce the navigation cost and make the system fully automatic. Among the solutions mentioned in the literature, we found the proposal by Sciancalepore et al. (2021) very interesting because of its back-compatibility features. In our opinion, relevant bodies, such as IMO, the US coast guard, the ITU, and researchers in academia and the industry can effectively use our ap-

[1] https://www.flightradar24.com
[2] https://flightaware.com

proach and outcomes to investigate and improve the security of the AIS.

The research is ongoing. Future work will be devoted to further investigating the security of ADS-B and the AIS. Although some European countries are testing UAT978, it is not currently in use in Europe, and its research is very limited there. However, UAT978 offers many attractive features, such as free access to weather information services. We plan to extend a security investigation of UAT978 offered services. We also plan to investigate, design, implement, and test some defensive measures against AIS attacks in our future studies.

YHTEENVETO (SUMMARY IN FINNISH)

Tässä väitöskirjassa tutkitaan paikannuspalveluiden kehitystä ja turvallisuutta erilaisissa langattomissa verkoissa. Artikkelit PI-PIV on omistettu sormenjälkipaikannussysteemien (FPS) kehittämiselle. Artikkeleissa PV–PVII tutkitaan automatiikkaan perustuvan valvonnan lähetysten (ADS-B) turvallisuutta, ja artikkelissa PVIII käsitellään automaattisten tunnistusjärjestelmien (AIS) turvallisuutta.

Ihmiset viettävät paljon aikaa sisätiloissa, joissa satelliiteista riippuvaiset paikanninohjelmat eivät toimi kunnolla. FPS:n avulla sisätilojen aiheuttamia paikannusongelmia voidaan korjata, sillä se hyödyntää olemassa olevaa Wi-Fi verkostoa. Sormenjälkipaikannuksessa on kuitenkin haasteensa. Esimerkiksi monet keinotekoiset ja luonnolliset syyt vaikuttavat vastaanotetun radiosignaalin voimakkuuteen (RSS), joka heikentää FPS:n toimivuutta. Lisäksi käytetyt algoritmit, etäisyysmatriisit, AP-tiheys, sormenjälkien tiheys, jne. vaikuttavat myös FPS:n tehokkuuteen. Jotta FPS:n suorituskykyä voidaan kehittää, teimme empiiristä tutkimusta artikkeleissa PI-PIV. Viittä julkisesti saatavilla olevaa sormenjälkien datapankkia käytettiin tutkimuksemme aikana. Artikkelin PI tulokset osoittavat, että kun sormenjälkiä kerätään joukkoistetulla tavalla, sormenjälkiä tallentavan laitteen tyypin säilyttäminen voisi auttaa vähentämään RSS-poikkeamaa ja siten parantamaan paikannustarkkuutta. Artikkelissa PII ehdotimme laitteiden välistä (D2D) viestintää tukevaa FPS:ää. Ehdotettu metodi jakaa sormenjäljet ja paikannuskokemuksen muiden laitteiden kanssa D2D-kommunikaatiota hyödyntäen. Tulokset osoittivat, että ehdotetun metodimme avulla paikannusvirheet vähenisivät noin 44%. FPS:n virheiden ennustustekniikka esitellään artikkelissa PIII. Tässä tekniikassa lasketaan yhden sijaan useiden testisormenjälkien sijainnit. Jos kaikki laskelmoidut sijainnit ovat lähellä toisiaan, on FPS-vaste luotettava. Virheindikaattorina käytetään sijaintiklusterin arvioitua sädettä. Artikkelissa PIV tehdyt kokeet osoittavat, että 4-bittinen kvantisointi riittää FPS:lle. Jos kvantisoitu RSS-sormenjälki voi kantaa radioympäristön tärkeimmät ominaisuudet, se on riittävä myös sormenjälkeen perustuvaan paikannukseen. Perinteiseen RSS:n verrattuna ehdottamamme 4-bittinen kvantisointi voi yksinkertaistaa laitteiston kokoonpanoa, parantaa turvallisuutta ja vähentää tarvittavaa varastointitilaa ja tietoliikennettä noin 40–60%.

ADS-B-teknologiaa käytetään ilmailualalla lähettämään lentokoneen sijaintitietoja ja muita tarpeellisia lentotietoja muille lentokoneille ja ATC-torneille. Sitä pidetään seuraavan sukupolven valvontajärjestelmien kulmakivenä, joka tekee ilmailusta turvallisempaa ja jonka avulla tulevaisuuden haasteisiin voidaan vastata. Vaikka ADS-B tarjoaakin monia hyödyllisiä palveluja, sen turvallisuudessa on parantamisen varaa. Artikkelissa PV toteutimme erilaisia hyökkäyksiä useita ADS-B-asetelmia vastaan käyttäen ohjelmistomääriteltyä radiota. Kaikki hyökkäykset osoittautuivat haitallisiksi ja saattavat uhata ADS-B-järjestelmää. Artikkelissa PVI järjestimme ohjelmistotestin (fuzz test) elektronisille lentolaukkusovelluksille (EFB). Tulokset osoittivat, että noin 56% testatuista sovelluksista kaa-

tui tai lakkasi reagoimasta testihyökkäyksen vuoksi. Artikkelissa PVII arvioimme suosittujen ohjaamoissa käytettyjen tietojärjestelmien turvallisuutta ja sietokykyä. 1090ES-asemia testattiin 44 ja UAT978-asemia 24 kappaletta. DoS-hyökkäys vaikutti 1090ES-asemista noin 63%:n ja UAT978-asemista noin 37%:n. Artikkelien PV–PVII tutkimustuloksista paljastui, että tämänhetkinen ADS-B-protokolla on haavoittuvainen erilaisille kyberhyökkäyksille.

Vastatoimena loimme RSS etäisyysmallin, jolla todellinen ja väärennetty ADS-B-signaali voidaan erottaa toisistaan. Tämä malli toimii paremmin, kun hyökkäyssignaali oli voimakas. Doppler siirtymäkokeessa ei saatu myönteisiä tuloksia. Väärennettyjen signaalien havaitseminen signaalin voimakkuuden tai Dopplersiirtymän avulla on vain ensimmäinen puolustuslinja. Meidän mielestämme tärkeiden sidosryhmien, kuten Kansainvälinen siviili-ilmailujärjestö (ICAO), Yhdysvaltain ilmailuhallinto (FAA), Euroopan unionin lentoturvallisuusvirasto (EASA) ja European Organization for the Safety of Air Navigation (EUROCONTROL), pitäisi huomioida nämä ongelmat ja tehdä yhteistyötä suunnitellakseen ratkaisu, jolla ADS-B-järjestelmän turvallisuus voidaan varmistaa tässä väitöskirjassa esiteltyjä kyberhyökkäyksiä vastaan.

Meriliikenteessä alukset käyttävät AIS-järjestelmää jakaakseen sijaintinsa ja muita navigointiin liittyviä tietoja eri merenkulkualan toimijoiden kanssa. Kansainvälinen merenkulkujärjestö (IMO) määräsi AIS-järjestelmän käytön vuonna 2004. ADS-B:n tavoin myös AIS:n turvallisuudessa on puutteita. Artikkelissa PVIII esitellään 11 hyökkäystä 19:ää AIS-konfiguraatiota vastaan, ja useimmat hyökkäykset osoittautuivat tehokkaiksi. Olemassa olevien hyökkäyskonseptien, kuten palveluneston ja häirinnän lisäksi jotkin uudet ideat, kuten koordinoitu hyökkäys ja hälytysten suuri yhtäaikainen määrä osoittavat, että halvalla hyökkäystentorjunta-asetelmalla voidaan vaikuttaa useiden miljoonien dollareiden arvoisen aluksen navigointiturvallisuuteen. Mielestämme asiaankuuluvat elimet, kuten Kansainvälinen merenkulkujärjestö (IMO), Yhdysvaltain rannikkovartiosto, Kansainvälinen televiestintäliitto (ITU) sekä korkeakoulujen ja alan muut tutkijat voivat käyttää lähestymistapaamme ja tuloksiamme tehokkaasti AIS-järjestelmän turvallisuuden tutkimiseen ja parantamiseen.

REFERENCES

3GPP 2013. Feasibility study for Proximity Services (ProSe). TR 22.803 V12.2.0.

Androjna, A., Perkovič, M., Pavic, I. & Mišković, J. 2021. Ais data vulnerability indicated by a spoofing case-study. Applied Sciences 11 (11).

Arya, A., Godlewski, P., Campedel, M. & du Chéné, G. 2013. Radio Database Compression for Accurate Energy-Efficient Localization in Fingerprinting Systems. IEEE Transactions on Knowledge and Data Engineering 25 (6), 1368-1379.

Bahl, P. & Padmanabhan, V. 2000. RADAR: an in-building RF-based user location and tracking system. In Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings., Vol. 2, 775-784.

Balduzzi, M., Pasta, A. & Wilhoit, K. 2014. A Security Evaluation of AIS Automated Identification System. In Proceedings of the 30th Annual Computer Security Applications Conference. New York, USA: ACM. ACSAC '14, 436–445.

Bateman, T. 2021. HMS Defender: AIS spoofing is opening up a new front in the war on reality. https://www.euronews.com/next/2021/06/28/hms-defender-ais-spoofing-is-opening-up-a-new-front-in-the-war-on-reality. (Accessed 04.08.2021).

Bergen, M. H., Arafa, A., Jin, X., Klukas, R. & Holzman, J. F. 2015. Characteristics of Angular Precision and Dilution of Precision for Optical Wireless Positioning. Journal of Lightwave Technology 33 (20), 4253-4260.

Bi, J., Wang, Y., Li, X., Qi, H., Cao, H. & Xu, S. 2018. An Adaptive Weighted KNN Positioning Method Based on Omnidirectional Fingerprint Database and Twice Affinity Propagation Clustering. Sensors 18 (8), 1-9.

Blumenthal, J., Grossmann, R., Golatowski, F. & Timmermann, D. 2007. Weighted centroid localization in zigbee-based sensor networks. In 2007 IEEE International Symposium on Intelligent Signal Processing, 1-6.

Bothur, D., Zheng, G. & Valli, C. 2017. A critical analysis of security vulnerabilities and countermeasures in a smart ship system. In The Proceedings of 15th Australian Information Security Management Conference, 81-87.

Braeken, A. 2019. Holistic Air Protection Scheme of ADS-B Communication. IEEE Access 7, 65251-65262.

COAA 2021. ShipPlotter. http://www.coaa.co.uk/shipplotter.htm. (Accessed 15.01.2022).

Coast-guard 2021. AIS Messages. https://www.navcen.uscg.gov/?pageName=AISMessages. (Accessed 17.01.2022).

Costin, A. & Francillon, A. 2012. Ghost in the Air (Traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices. BlackHat USA 1, 1-12.

Cruz, F. R. G., Gania, R. C. M., Garcia, B. W. C. & Nob, J. C. R. 2018. Implementing Automatic Identification System Transmitter on Software Defined Radio. In IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), 1-4.

De-La-Llana-Calvo, A., Lázaro-Galilea, J.-L., Gardel-Vicente, A., Rodríguez-Navarro, D., Rubiano-Muriel, B. & Bravo-Muñoz, I. 2020. Analysis of Multiple-Access Discrimination Techniques for the Development of a PSD-Based VLP System. Sensors 20 (6).

Dembovskis, A. 2015. AIS message extraction from overlapped AIS signals for SAT-AIS applications. University of Bremen. Ph. D. Thesis.

Domin, K., Symeonidis, I. & Marin, E. 2016. Security analysis of the drone communication protocol: Fuzzing the MAVLink protocol. University of Luxembourg. Master's Thesis.

EASA 2018. EASA seasonal technical commission. https://www.easa.europa.eu/sites/default/files/dfu/EASA_STC_NEWS_JUNE_2018.pdf. (Accessed 02.03.2021).

EbcTech 2021. AIS Share. https://play.google.com/store/apps/details?id=eu.ebctech.ais_share. (Accessed 11.12.2021).

Eskilsson, S., Gustafsson, H., Khan, S. & Gurtov, A. 2020. Demonstrating ADS-B and CPDLC Attacks with Software-Defined Radio. In Integrated Communications Navigation and Surveillance Conference (ICNS). IEEE, 1B2-1-1B2-9.

FAA 2018. No Kidding: ADS-B Deadline of Jan. 1, 2020, is Firm. https://www.faa.gov/news/updates/?newsId=90008. (Accessed 06.11.2021).

Filiba, T. 2020. Tomerfiliba/reedsolomon. https://github.com/tomerfiliba/reedsolomon/blob/master/reedsolo.py. (Accessed 02.08.2021).

Finke, C., Butts, J. & Mills, R. 2013. ADS-B Encryption: Confidentiality in the Friendly Skies. In 8th Annual Cyber Security and Information Intelligence Research Workshop. ACM, 1-4.

Ghose, N. & Lazos, L. 2015. Verifying ADS-B navigation information through Doppler shift measurements. In IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), 4A2-1-4A2-11-11.

Goudossis, A. & Katsikas, S. K. 2018. Towards a secure automatic identification system (AIS). Journal of Marine Science and Technology 24 (2), 410-423.

Hassan, W. U., Guo, S., Li, D., Chen, Z., Jee, K., Li, Z. & Bates, A. 2019. NoDoze: Combatting Threat Alert Fatigue with Automated Provenance Triage. In 26th Annual Network and Distributed System Security Symposium (NDSS), 1-40.

He, S. & Chan, S. . G. 2016. Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons. IEEE Communications Surveys Tutorials 18 (1), 466-490.

Honkavirta, V., Perala, T., Ali-Loytty, S. & Piche, R. 2009. A comparative survey of WLAN location fingerprinting methods. In 6th Workshop on Positioning, Navigation and Communication, 243-251.

Huang, M.-S., Narayanan, R., Zhang, Y. & Feinberg, A. 2013. Tracking of Non-cooperative Airborne Targets Using ADS-B Signal and Radar Sensing. International Journal of Aerospace Engineering 2013 (521630).

ICAO 2005. Standards and Recommended Practices for the Universal Access Transceiver (UAT). International Civil Aviation Organization.

IEC 2012. Maritime navigation and radiocommunication equipment and systems – Automatic identification systems (AIS). Part 2: Class A shipborne equipment of the automatic identification system (AIS) – Operational and performance requirements, methods of test and required test results.

IEC 2017a. Maritime navigation and radio communication equipment and systems – Class B. shipborne equipment of the automatic identification system (AIS) – Part 1 : Carrier-sense time division multiple access (CSTDMA) techniques.

IEC 2017b. Maritime navigation and radiocommunication equipment and systems – Class B. Shipborne equipment of the automatic identification system (AIS) – Part 2: Self-organising time division multiple access (SOTDMA) techniques.

IMO 2004. AIS transponders. https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx. (Accessed 04.08.2021).

ITU 2014. Technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band. https://www.itu.int/rec/R-REC-M.1371-5-201402-I.

ITU 2017. Reception of automatic dependent surveillance broadcast via satellite and compatibility studies with incumbent systems in the frequency band 1 087.7-1 092.3 MHz. https://www.itu.int/pub/R-REP-M.2413.

Ji, M., Kim, J., Jeon, J. & Cho, Y. 2015. Analysis of positioning accuracy corresponding to the number of BLE beacons in indoor positioning system. In 17th International Conference on Advanced Communication Technology (ICACT), 92-95.

Jimenez, A. R., Seco, F., Prieto, C. & Guevara, J. 2009. A comparison of Pedestrian Dead-Reckoning algorithms using a low-cost MEMS IMU. In IEEE International Symposium on Intelligent Signal Processing, 37-42.

Kim, T., Kim, C. H., Rhee, J., Fei, F., Tu, Z., Walkup, G., Zhang, X., Deng, X. & Xu, D. 2019. RVFuzzer: Finding input validation bugs in robotic vehicles through control-guided testing. In 28th USENIX Security Symposium USENIX Security 19, 425-442.

Kim, Y., Jo, J.-Y. & Lee, S. 2016. A secure location verification method for ADS-B. In IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), 1-10.

Kim, Y., Jo, J.-Y. & Lee, S. 2017. ADS-B vulnerabilities and a security solution with a timestamp. IEEE Aerospace and Electronic Systems Magazine 32 (11), 52–61.

King, T., Kopf, S., Haenselmann, T., Lubberger, C. & Effelsberg, W. 2006. COMPASS: A Probabilistic Indoor Positioning System Based on 802.11 and Digital Compasses. In Proceedings of the 1st International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, 34-40.

Konstantinidis, A., Chatzimilioudis, G., Zeinalipour-Yazti, D., Mpeis, P., Pelekis, N. & Theodoridis, Y. 2015. Privacy-Preserving Indoor Localization on Smartphones. IEEE Transactions on Knowledge and Data Engineering 27 (11), 3042-3055.

Kurschl, W., Gottesheim, W., Mitsch, S., Prokop, R., Schönböck, J. & Beer, W. 2008. Large-Scale Industrial Positioning and Location Tracking Are We There Yet? In 7th International Conference on Mobile Business, 251-259.

Laitinen, E., Talvitie, J. & Lohan, E. 2015. On the RSS biases in WLAN-based indoor positioning. In IEEE International Conference on Communication Workshop (ICCW), 797-802.

Larsen, J. A., Mortensen, H. P. & Nielsen, J. D. 2011. An SDR based AIS receiver for satellites. In 5th International Conference on Recent Advances in Space Technologies - RAST2011, 526-531.

Leite, J., Walmor, C., de Moraes, C. C., de Albuquerque, C. E. P., Machado, R. C. S. & de Sá, A. O. 2021. A Triggering Mechanism for Cyber-Attacks in Naval Sensors and Systems. Sensors 21 (9).

Leonardi, M., Strohmeier, M. & Lenders, V. 2021. On Jamming Attacks in Crowdsourced Air Traffic Surveillance. IEEE Aerospace and Electronic Systems Magazine 36 (6), 44-54.

Li, H., Sun, L., Zhu, H., Lu, X. & Cheng, X. 2014. Achieving privacy preservation in WiFi fingerprint-based localization. In IEEE Conference on Computer Communications, 2337-2345.

Li, T. & Wang, B. 2019. Sequential collaborative detection strategy on ADS-B data attack. International Journal of Critical Infrastructure Protection 24, 78-99.

Liu, K., Wang, Y., Lin, L. & Chen, G. 2017. An Analysis of Impact Factors for Positioning Performance in WLAN Fingerprinting Systems Using Ishikawa Diagrams and a Simulation Platform. Mobile Information Systems 2017, 1-20.

Liu, M., Wang, H., Yang, Y., Zhang, Y., Ma, L. & Wang, N. 2019. RFID 3-D Indoor Localization for Tag and Tag-Free Target Based on Interference. IEEE Transactions on Instrumentation and Measurement 68 (10), 3718-3732.

Lohan, E., Torres-Sospedra, J., Leppäkoski, H., Richter, P., Peng, Z. & Huerta, J. 2017. Wi-Fi Crowdsourced Fingerprinting Dataset for Indoor Positioning. Data 2 (4), 32.

Lundberg, D., Farinholt, B., Sullivan, E., Mast, R., Checkoway, S., Savage, S., Snoeren, A. C. & Levchenko, K. 2014. On the security of mobile cockpit information systems. In ACM SIGSAC Conference on Computer and Communications Security, 633–645.

Mahtab Hossain, A., Jin, Y., Soh, W.-S. & Van, H. N. 2013. SSD: A Robust RF Location Fingerprint Addressing Mobile Devices' Heterogeneity. IEEE Transactions on Mobile Computing 12 (1), 65-77.

Manesh, M. R. & Kaabouch, N. 2017. Analysis of vulnerabilities, attacks, countermeasures and overall risk of the Automatic Dependent Surveillance-Broadcast (ADS-B) system. International Journal of Critical Infrastructure Protection 19, 16-31.

Manesh, M. R., Mullins, M., Foerster, K. & Kaabouch, N. 2018. A preliminary effort toward investigating the impacts of ADS-B message injection attack. In IEEE Aerospace Conference, 1-6.

MarineTraffic 2019. US warns merchant shipping of Iranian GPS spoofing threat. https://www.marinetraffic.com/hr/maritime-news/article/26535. (Accessed 05.09.2021).

MarineTraffic 2021. AISMon. https://help.marinetraffic.com/hc/en-us/articles/205339707-AISMon. (Accessed 12.08.2021).

Marques, M. M., Teles, D., Lobo, V. & Capela, G. 2019. Low-cost AIS Transponder using an SDR device. In OCEANS MTS/IEEE Seattle, 1-4.

Mathapo, K. F. 2007. A Software-Defined Radio Implementation of Maritime AIS. University of Stellenbosch. Master's Thesis.

Mautz, R. 2012. Indoor Positioning Technologies. ETH Zurich, Department of Civil, Environmental and Geomatic Engineering, Institute of Geodesy and Photogrammetry.

McCallie, D., Butts, J. & Mills, R. 2011. Security analysis of the ADS-B implementation in the next generation air transportation system. International Journal of Critical Infrastructure Protection 4 (2), 78-87.

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

Mendoza-Silva, G., Richter, P., Torres-Sospedra, J., Lohan, E. & Huerta, J. 2018. Long-Term WiFi Fingerprinting Dataset for Research on Robust Indoor Positioning. Data 3 (1).

Meneses, F., Moreira, A., Costa, A. & Nicolau, M. J. 2019. Radio Maps for Fingerprinting in Indoor Positioning. In Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation. Academic Press. Intelligent Data-Centric Systems, 69 - 95.

Mizmizi, M. & Reggiani, L. 2016. Design of RSSI based fingerprinting with reduced quantization measures. In International Conference on Indoor Positioning and Indoor Navigation (IPIN), 1-6.

Moghtadaiee, V. & Dempster, A. G. 2014. Design protocol and performance analysis of indoor fingerprinting positioning systems. Physical Communication 13, 17-30.

Moreira, A., Silva, I., Meneses, F., Nicolau, M. J., Pendao, C. & Torres-Sospedra, J. 2017. Multiple simultaneous Wi-Fi measurements in fingerprinting indoor positioning. In International Conference on Indoor Positioning and Indoor Navigation (IPIN), 1-8.

Nessa, A., Adhikari, B., Hussain, F. & Fernando, X. N. 2020. A Survey of Machine Learning for Indoor Positioning. IEEE Access 8, 214945-214965.

OpenCPN 2021. Chart Plotter and Navigational software. https://opencpn.org. (Accessed 12.08.2021).

Parker, D. B. 2012. Toward a New Framework for Information Security? John Wiley and Sons, Ltd.

Peral-Rosado, J. A. D., Raulefs, R., López-Salcedo, J. A. & Seco-Granados, G. 2018. Survey of Cellular Mobile Radio Localization Methods: From 1G to 5G. IEEE Communications Surveys Tutorials 20 (2), 1124-1148.

Perez, M. C., Ureña, J., Hernandez, A., Jimenez, A., Ruiz, D., Alvarez, F. J. & Marziani, C. D. 2009. Performance comparison of different codes in an ultrasonic positioning system using DS-CDMA. In IEEE International Symposium on Intelligent Signal Processing, 125-130.

Potortì, F., Park, S., Jiménez Ruiz, A. R., Barsocchi, P., Girolami, M., Crivello, A., Lee, S. Y., Lim, J. H., Torres-Sospedra, J., Seco, F., Montoliu, R., Mendoza-Silva,

G. M., Pérez Rubio, M. D. C., Losada-Gutiérrez, C., Espinosa, F. & Macias-Guarasa, J. 2017. Comparing the Performance of Indoor Localization Systems through the EvAAL Framework. Sensors 17 (10).

Qin, F., Zuo, T. & Wang, X. 2021. CCpos: WiFi Fingerprint Indoor Positioning System Based on CDAE-CNN. Sensors 21 (4).

Razavi, A., Valkama, M. & Lohan, E. 2015. K-Means Fingerprint Clustering for Low-Complexity Floor Estimation in Indoor Mobile Localization. In IEEE Globecom Workshops, 1-7.

Richter, P., Leppakoski, H., Lohan, E. S., Yang, Z., Jarvinen, K., Tkachenko, O. & Schneider, T. 2018. Received Signal Strength Quantization for Secure Indoor Positioning via Fingerprinting. In 8th International Conference on Localization and GNSS (ICL-GNSS), 1-6.

Ruiz, A. R. J. & Granja, F. S. 2017. Comparing Ubisense, BeSpoon, and DecaWave UWB Location Systems: Indoor Performance Analysis. IEEE Transactions on Instrumentation and Measurement 66 (8), 2106-2117.

Schäfer, M., Leu, P., Lenders, V. & Schmitt, J. 2016. Secure Motion Verification Using the Doppler Effect. In 9th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 135–145.

Schäfer, M., Lenders, V. & Martinovic, I. 2013. Experimental Analysis of Attacks on Next-Generation Air Traffic Communication. In 11th International Conference on Applied Cryptography and Network Security. Springer Berlin Heidelberg, 253-271.

Sciancalepore, S., Tedeschi, P., Aziz, A. & Di Pietro, R. 2021. Auth-AIS: Secure, Flexible, and Backward-Compatible Authentication of Vessels AIS Broadcasts. IEEE Transactions on Dependable and Secure Computing, 1-1.

Sen, S., Lee, J., Kim, K.-H. & Congdon, P. 2013. Avoiding Multipath to Revive Inbuilding WiFi Localization. In Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services. Association for Computing Machinery. MobiSys '13, 249–262.

Sjödin, A. & Gruneau, M. 2020. The ADS-B protocol and its' weaknesses. KTH Royal Institute of Technology. Ph. D. Thesis.

Song, X., Fan, X., Xiang, C., Ye, Q., Liu, L., Wang, Z., He, X., Yang, N. & Fang, G. 2019. A Novel Convolutional Neural Network Based Indoor Localization Framework With WiFi Fingerprinting. IEEE Access 7, 110698-110709.

Stein, J. C. 1998. Indoor Radio WLAN Performance Part II : Range Performance in a Dense Office Environment. https://www.erasme.org/IMG/experience_attenuation.pdf.

Strohmeier, M., Lenders, V. & Martinovic, I. 2013. Security of ADS-B: State of the Art and Beyond. IEEE Communications Surveys and Tutorials 17.

Strohmeier, M., Lenders, V. & Martinovic, I. 2015. On the Security of the Automatic Dependent Surveillance-Broadcast Protocol. IEEE Communications Surveys Tutorials 17 (2), 1066-1087.

Strohmeier, M., Schäfer, M., Lenders, V. & Martinovic, I. 2014. Realities and challenges of nextgen air traffic management: the case of ADS-B. IEEE Communications Magazine 52 (5), 111–118.

Su, P., Sun, N., Zhu, L., Li, Y., Bi, R., Li, M. & Zhang, Z. 2017. A Privacy-Preserving and Vessel Authentication Scheme Using Automatic Identification System. In 5th ACM International Workshop on Security in Cloud Computing. ACM. SCC '17, 83–90.

Talvitie, J., Lohan, E. S. & Renfors, M. 2014. The effect of coverage gaps and measurement inaccuracies in fingerprinting based indoor localization. In International Conference on Localization and GNSS (ICL-GNSS 2014), 1-6.

Torres-Sospedra, J., Montoliu, R., Martínez-Usó, A., Avariento, J. P., Arnau, T. J., Benedito-Bordonau, M. & Huerta, J. 2014. UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. In International Conference on Indoor Positioning and Indoor Navigation (IPIN), 261-270.

Torres-Sospedra, J., Montoliu, R., Trilles, S., Óscar Belmonte & Huerta, J. 2015. Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems. Expert Systems with Applications 42 (23), 9263 - 9278.

Torres-Sospedra, J. & Moreira, A. 2017. Analysis of Sources of Large Positioning Errors in Deterministic Fingerprinting. Sensors 17 (12), 2736.

TrendMicro 2014. AISTX. https://github.com/trendmicro/ais. (Accessed 27.09.2021).

Vessel-Finder 2021. Vessel database. https://www.vesselfinder.com/vessels. (Accessed 04.08.2021).

Wu, Z., Shang, T. & Guo, A. 2020. Security Issues in Automatic Dependent Surveillance-Broadcast (ADS-B): A Survey. IEEE Access 8, 122147–122167.

Xia, S., Liu, Y., Yuan, G., Zhu, M. & Wang, Z. 2017. Indoor Fingerprint Positioning Based on Wi-Fi: An Overview. ISPRS International Journal of Geo-Information 6 (5).

Xiao, J., Zhou, Z., Yi, Y. & Ni, L. M. 2016. A Survey on Wireless Indoor Localization from the Device Perspective. ACM Computing Surveys 49 (2), 25:1-25:31.

Xie, Y., Wang, Y., Nallanathan, A. & Wang, L. 2016. An Improved K-Nearest-Neighbor Indoor Localization Method Based on Spearman Distance. IEEE Signal Processing Letters 23 (3), 351-355.

Yang, H., Zhou, Q., Yao, M., Lu, R., Li, H. & Zhang, X. 2019. A Practical and Compatible Cryptographic Solution to ADS-B Security. IEEE Internet of Things Journal 6 (2), 3322-3334.

Youssef, M., Agrawala, A. & Shankar, A. U. 2003. WLAN location determination via clustering and probability distributions. In Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 143-150.

Yusupov, L. 2021a. lyusupov/ADSB-Out. https://github.com/lyusupov/ADSB-Out. (Accessed 10.09.2021).

Yusupov, L. 2021b. lyusupov/UAT-test-signal. https://github.com/lyusupov/UAT-test-signal. (Accessed 10.09.2021).

Zhuang, Y., Li, Y., Qi, L., Lan, H., Yang, J. & El-Sheimy, N. 2016. A Two-Filter Integration of MEMS Sensors and WiFi Fingerprinting for Indoor Positioning. IEEE Sensors Journal 16 (13), 5125-5126.

Zwirko, C. 2019. North Korean vessels exploiting tracking system flaws to evade sanctions: report. https://www.nknews.org/2019/06/north-korean-vessels-exploiting-tracking-system-flaws-to-evade-sanctions-report. (Accessed 04.08.2021).

ORIGINAL PAPERS

PI

DEVICE DIVERSITY EFFECTS ON RF FINGERPRINTING BASED 3D POSITIONING SYSTEM

by

S Khandker, R Mondal, T Ristaniemi 2018

8th International Conference on Localization and GNSS (ICL-GNSS), Guimaraes, Portugal

https://doi.org/10.1109/icl-gnss.2018.8440899

2018 8th International Conference on Localization and GNSS (ICL- GNSS), 26-28 June 2018, Guimares, Portugal

Device Diversity Effects on RF Fingerprinting Based 3D Positioning System

[1] Syed Khandker [2] Riaz Mondal [3] Tapani Ristaniemi
Faculty of Information Technology
University of Jyvaskyla
Jyvaskyla, Finland
[1]syed.i.khandker@student.jyu.fi [2]riaz.u.mondal@student.jyu.fi [3]tapani.ristaniemi@jyu.fi

***Abstract*—This paper presents the device diversity effects on Radio Frequency (RF) fingerprinting framework in the indoor environment for the three-dimensional positioning system. RF Fingerprinting positioning was studied inside of a multistory university building by devices from multiple manufacturers at different times. The goal was to study the effects of equipment heterogeneity on positioning accuracy in Wireless Local Area Network (WLAN). The performance evaluation demonstrates that RF fingerprints from homogeneous devices maintain better similarity in a crowdsourced database. By keeping the reference of user device information, better positioning accuracy can be achieved with less amount of training data.**

***Index Terms*—RF Fingerprinting, Indoor positioning, k-nearest neighbor, k-means clustering**

I. Introduction

Indoor localization is a critical tool for fast developing location-based service [1]. It plays an essential role in navigation, security, and healthcare industries. The proliferation of wireless network devices has encouraged researchers to develop a wide range of different positioning methods to estimate User Equipment (UE) location such as RF fingerprinting method. RF fingerprinting method refers to a database correlation method where UE position is determined by comparing UE's RF fingerprint with correlation RF fingerprint database that is associated with known locations [2]. Typical RF fingerprint or signature consists of radio measurements from multiple base stations / Access Points (AP), i.e., Received Signal Strength (RSS) or path-loss measurements, to provide a unique fingerprint of the radio conditions at a specific geographical location [3]. RF fingerprinting method consists of two phases, training phase, and testing phase. During the training phase, fingerprints associated with known locations are stored in a database, and this process allows to build a correlation database with a reference radio and location measurements. In this way a radio environment map can be created to estimate positions of UEs, having no accurate position information.

Location of reference fingerprints is usually determined by accurate reference position measurement, e.g., using Global Navigation Satellite System (GNSS). In the indoor area where the direct signal of GNSS satellite is not available, it becomes more challenging to create an RF fingerprint database due to the need of the particular device or software to obtain accurate positioning information. The use of specialized support to build a training database may aggravate the chance of hardware and software level dissimilarity between the data collection phase and the testing phase. Devices used by users are varied by brand, model, and manufacture. Receiving sensitivity of network card or chipset is not also equal [4], therefore the sensing of a radio signal by different devices would not be the same. Different devices would sense an identical signal at different signal strength level. However, RSS is the crucial element to distinguish the RF fingerprints. The minor variation in RSS reveals the closer physical stand among the RF fingerprints. Heterogeneous devices experience a radio condition independently, that would put effects on radio signal based positioning system.

In this paper, we have investigated the effects of device diversity on RF fingerprinting based three-dimensional coordinate positioning system. To justify the diversity effect, at data collection phase the mobile devices from four different manufacturers were considered. Data was collected from a five-story university indoor premises. Nearest Neighbor (NN), K-Nearest Neighbor (kNN), and k-means clustering algorithm were used to evaluate the positioning performance.

This paper is organized as follows: Section II describes the causes of hardware-based RSS variations. Section III contains the description of the database we used in the experiments. In Section IV, we discussed RF fingerprinting based positioning algorithms. Finally, in Section V, the performance evaluation of device diversity on RF fingerprinting based indoor positioning is presented.

II. Device Level RSS Variation

In an indoor environment where spatial characteristics (e.g., shadowing and multipath effects) are different across space are vulnerable to RSS variation. Moreover, environmental factors, cell breathing, transmission power adjustment, and device types also put a significant impact on radio condition [5]. Depending on the chip, WLAN RF receive sensitivity varies, for example, at 802.11G standard WCN3620 Wi-Fi chip used by Sony shows typical receive sensitivity -75.1 dBm whereas BCM4334 of Broadcom widely used in Samsung devices has a typical rating of -76 dBm at WLAN RF port [6], [7]. Table

978-1-5386-6984-6/18/$31.00 ©2018 IEEE

I shows that the popular 802.11 network protocols also have variety in range, speed, and response frequency.

TABLE I: 802.11 Protocols

Name	Speed	Indoor Range	Frequency
802.11/AC	1 Gbps	115 Feet	5 GHz
802.11/N	300 Mbps	230 Feet	2.4 GHz, 5 GHz
802.11/G	54 Mbps	125 Feet	2.4 GHz
802.11/B	11 Mbps	115 Feet	2.4 GHz

The coverage range of all the standards are not equal, therefore it is likely to be unequal in physical distance for the same amount of path-loss component. The devices which were considered in the experiment among them Sony supports 802.11 a/b/g/n/ac + MIMO while Samsung or Asus capable of 802.11 b/g/n. Being updated in standard Sony would be able to cooperate with 5GHz frequency's wireless router where others would be limited to the 2.4GHz band. Device specifications of Wi-Fi chips reveal that depending on the communication speed typical minimum receive sensitivity changes. Broadcom and Qualcomm both Wi-Fi chips have around -97 dBm, -90 dBm, and -76 dBm sensitivity for 1Mbps, 11Mbps, and 54Mbps receptively [6], [7]. Furthermore, the mapping between the actual RF energy to the received signal strength indicator range can vary from one manufacturer to another [4].

III. Database Description

Real life Wi-Fi crowdsourced RF fingerprints collected by the research group of the Tampere University of Technology has been used in this paper [8]. The five-story building has a footprint of about 22,570m^2. During the measurement period, a total of 991 Media Access Control (MAC) addresses were heard. Data were collected by using different Android devices in the user's hands, with the screen up. Devices supported both the 2.4GHz and 5GHz frequency range, the RSSs coming from both frequency bands were stored. However, the Android tool for data collection does not make the differentiation between RSS collected in different frequency bands. Fingerprints collected by Samsung (SM-A310F & SM-A510F), Sony (E5823 & SGP771), Asus (Nexus 7 & Transformer Prime TF201), and Huawei (Y360-U61 & T1 7.0) were considered for this research as those were the considerably higher amount in number. "Fig. 1" shows the overall position of training and testing signatures.

IV. Matching Algorithm

A. Signature Concept

RF fingerprint or signature consists of APs, corresponding RSS values, and location information. The *i*th signature S_i in training phase is given by

$$S_i = \{\gamma_i, \mu_i, \rho_{ixyz}\} \tag{1}$$

where $\gamma_i = [\beta_1, \beta_2, ..., \beta_k]$ is a column vector presenting a set of APs detected by UE from the location ρ_{ixyz} . k is the total number of detected APs of *i*th signature. Vector μ_i represents the RSS values $[\mu_1, \mu_2, ..., \mu_k]$ for corresponding

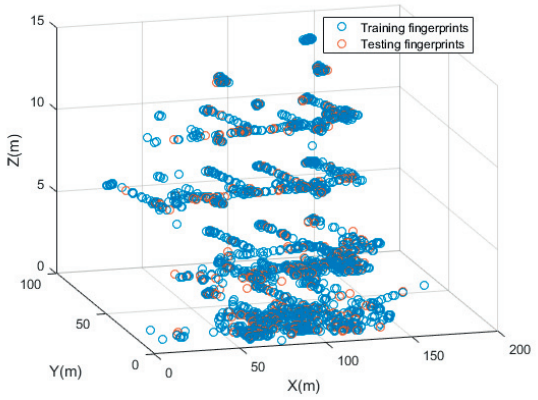


Fig. 1: RF fingerprint database

$[\beta_1, \beta_2, ..., \beta_k]$. Testing signatures at online phase do not contain location information.

B. k Nearest Neighbor

K-Nearest Neighbor (kNN) algorithm is also known as a distance based classifier that classifies instances based on their similarity. In order to satisfy the acceptable localization accuracy with low computation effort kNN has been used for WLAN UE positioning by several researchers [9], [10], [11]. To find out the position of testing signature $S^E = \{\gamma^E, \mu^E\}$ the algorithm works follow:

1: **While** $\{S_i\} \neq \emptyset$ **do**
2: **select** $S_i^R = \{\gamma_i^R, \mu_i^R\}\ from\ \{S_i\}$
3: $\{S_i\} = \{S_i\} - S_i^R$
4: **extract** μ_d^E *and* $\mu_{i,d}^R$ *from* S^E *and* S_i^R *according* γ^E
5: **calculate** similarity distance $\mathrm{d}(S^E, S_i^R)$
6: **end while**
7: **find** k minimum $\mathrm{d}(S^E, S_i^R)$
8: **return** the mean coordinate of the k training signatures

C. Nearest Neighbor

While only one neighbor is considered to determine the location information is called Nearest Neighbor (NN) algorithm. It always finds out the shortest RSS distance between current signature and visited signature. While all the signatures in the database are compared, then set the shortest distance holding signature as current signature and terminate the execution. Compare to the kNN algorithm stated above at step 7 instead of k minimum $\mathrm{d}(S^E, S_i^R)$ it finds 1 minimum $\mathrm{d}(S^E, S_i^R)$.

D. k-means Clustering

The k-means algorithm is a widely used clustering technique in scientific and industrial applications [12]. It has been successfully used in indoor mobile localization and also in outdoor positioning as an energy efficient RF fingerprinting method [13], [14]. This algorithm begins with a set of training signatures S_i where i = 1,2,...,n and a pre-defined maximum

cluster number k. The task is to choose k centers c_k so as to minimize the following distance function,

$$d\,(S,c) = \sum_{i=1}^{n} |S_i - c_k| \tag{2}$$

$D(S_i)$ denotes the shortest RSS distance from a RF fingerprint to the already chosen cluster center. k-means algorithm performs the following steps:
1: The first center c_1 is chosen uniformly at random from S.
2: A new center c_k is chosen from S with probability $\frac{D(S_i)^2}{\sum_{i=1}^{n-1} D(S_i)^2}$
3: Step (2) is repeated until all k centers are chosen.
4: For each c_k, training signatures are assigned to it which are closer to it than any other c_k.
5. New c_k is computed from the mean of all training signatures that belongs to the previous c_k.
6. Steps (4) and (5) are repeated until c no longer changes.

Euclidean distance and Davies-Bouldin criterion from Matlab implementation have been used to find out optimal value of k.

V. Result and Discussion

A. Experiment Set-up

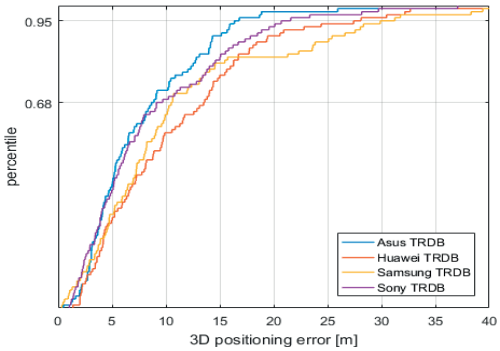
All the testing signatures were selected in a way that, every single of them has at least two training signatures within a five-meter radius, from each device group. Otherwise, it would be hard to distinguish the cause of effects, whether the repercussions occur due to device heterogeneity or due to the disparity in physical distance.

TABLE II: Number of training and testing signatures

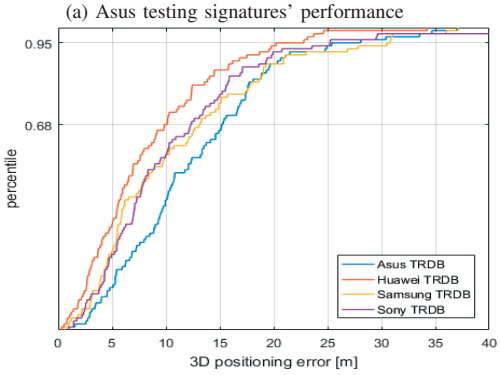
Name	Number of training signatures	Number of testing signatures
Asus	304	100
Huawei	500	100
Samsung	660	100
Sony	719	100

B. Evaluation

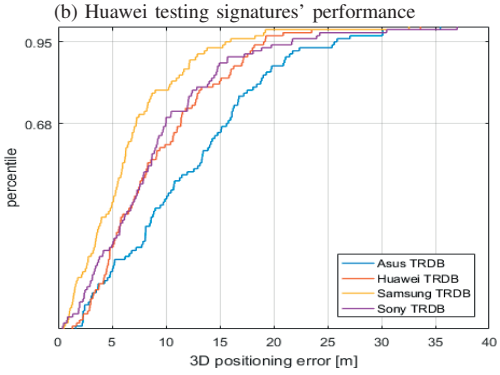
kNN algorithm has been used to analyze the positioning accuracy of 100 testing signatures from each device group for own, and other devices created Training Database (TRDB). We set k = 3 to maintain the best trade-off between performance and analyzing capability. Cumulative Distribution Function (CDF) has been used to express the percentage of test signatures against any positioning error value (in meter). For example, in "Fig. 2(a)", 68% Asus test signatures resulted 8.5 meters positioning error or less in Asus TRDB. The benefit of using CDF expression is, it illustrates the performance of entire test population. In a normal distribution, approximately 68% and 95% of the observations fall within first and second standard deviation of the mean respectively; therefore we focused on 68-95 rule. In the "Fig. 2" we can see each device's testing signatures are showing better performance on own device created training database.



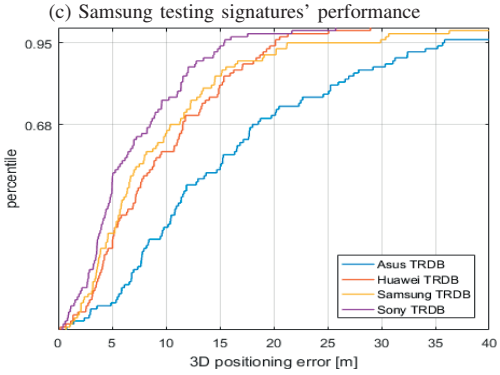
(a) Asus testing signatures' performance



(b) Huawei testing signatures' performance



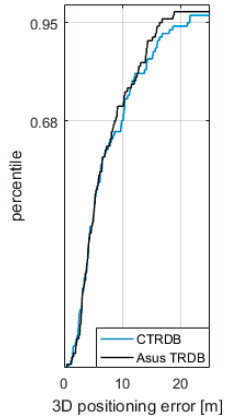
(c) Samsung testing signatures' performance



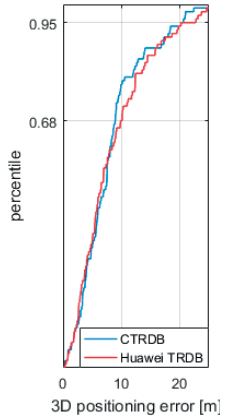
(d) Sony testing signatures' performance

Fig. 2: Cross devices performance evaluation by kNN

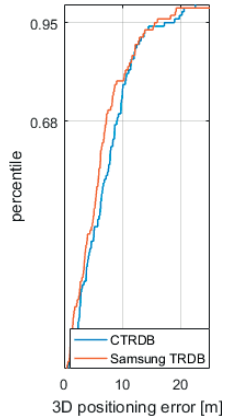
Asus has 304 training signatures whereas Sony or Samsung has more than double of that. But Asus-collected testing signatures perform better on training database built by Asus compare to the other training databases. Huwaei, Samsung, and Sony testing signatures show the same trend. An issue can be raised that, testing fingerprints were physically close to their own device created training fingerprints. But as we mentioned, the testing databases were created in a way that from each training devices there will be at least two signatures within a five-meters radius, "Fig. 1" also shows the compact position of the fingerprints. Moreover, each testing signature database was analyzed by Combined Training Database (CTRDB) of all training signatures, where there were 2183 training signatures altogether. "Fig. 3" shows that the position accuracy in own DB and CTRDB is very close.



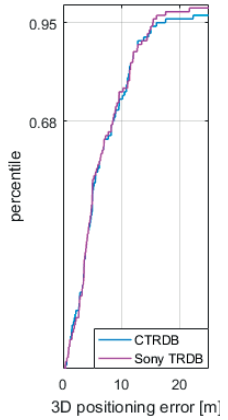
(a) Asus testing signatures



(b) Huawei testing signatures



(c) Samsung testing signatures



(d) Sony testing signatures

Fig. 3: Performance comparison on TRDB and CTRDB

The CDF of positioning error for the four test databases in "Fig. 3" shows that by using significantly less amount of training data nearly same result can be achieved. Difference between the amount of training signature in CTRDB and own device created TRDB were 1879, 1683, 1523, and 1464 for Asus, Huawei, Samsung, and Sony respectively. That indicates the scope of reducing computational complexity, reducing the time to respond, and in some cases higher accuracy. Of course, the number of training signatures plays a vital role; a higher amount of training data can achieve better positioning accuracy. However, only increase in the number of training signature does not improve the position accuracy. Another research carried out on same crowdsourced fingerprints database [15], they also concluded that when several heterogeneous devices and software are used, there will be an inherent deterioration in the positioning accuracy compared to the single device single software approach. The comparison of the mean three-dimensional positioning error by different algorithms for the different testing database is shown in "Fig. 4" to "Fig. 7"

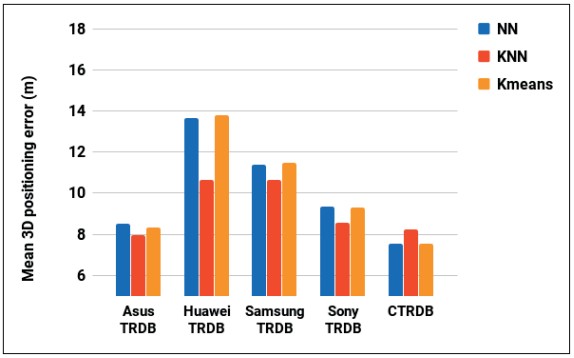


Fig. 4: Asus testing signatures' performance

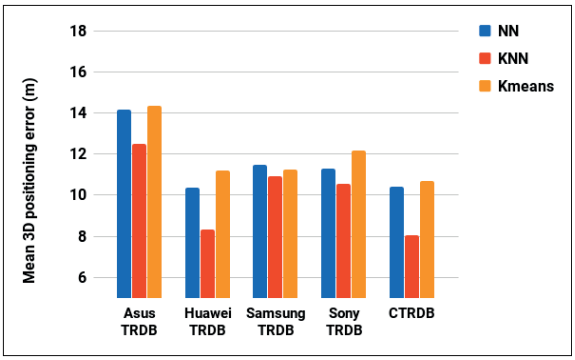


Fig. 5: Huawei testing signatures' performance

Positioning accuracy by k-means and NN algorithms also reveal the same characteristic that, on own device created TRDB testing signatures perform better. The best performance comes from kNN algorithm. TUT research group found 24.73m to 8.73m mean three-dimensional positioning error,

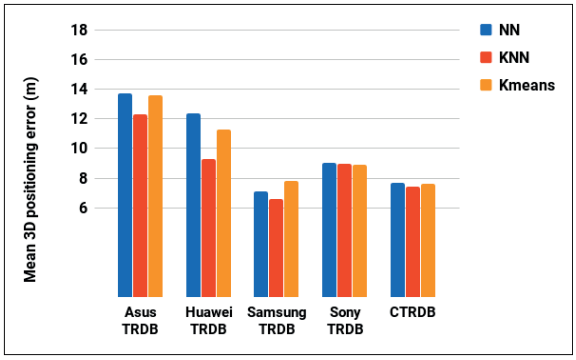


Fig. 6: Samsung testing signatures' performance

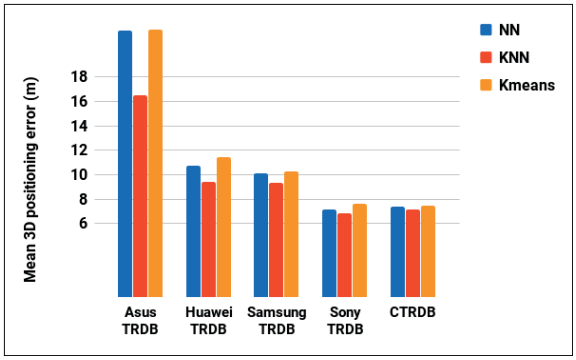


Fig. 7: Sony testing signatures' performance

depending on different algorithms [8]. In our single device approach, we calculated it 6.87m, 6.59m, 7.96m, and 8.35m for Sony, Samsung, Asus, and Huawei respectively by kNN method. Even though the setup and the approach are different, but it indicates that, along with a higher amount of training data, the less variation in hardware (since the software was same) can provide improved performance.

C. Suggestion

Experiments show that in the real world situation due to the difference in hardware, devices may not sense the same radio condition what may lead to having inaccurate location information from the service provider (SP). Beside daily life user equipment localization, Internet of Things (IoT) would also deal with various devices; therefore RF fingerprinting with device reference would be very useful. An extra layer, e.g., the device information layer can be added to the traditional correlation database as shown in "Fig. 8". While training database is created by warwalking or wardriving than devices used in data collection process are known to SP. If online Minimization of Drive Test (MDT) data is used for training purpose than by MAC ID, or International Mobile Equipment Identity (IMEI), or Electronic Serial number (ESN), device-level information can be obtained [16]. According to authors' current knowledge, inside of MDT data WLAN signal strength information has not been included yet but experts are investigating the issue. While training signatures contain device information (e.g., device name, model), at testing phase SP can respond to location information request according to the device type. Split segments of traditional correlation database to serve own device group's positioning request. In case of having no or less amount of training data by own device group, the system can look for training information from other segments or combined database.

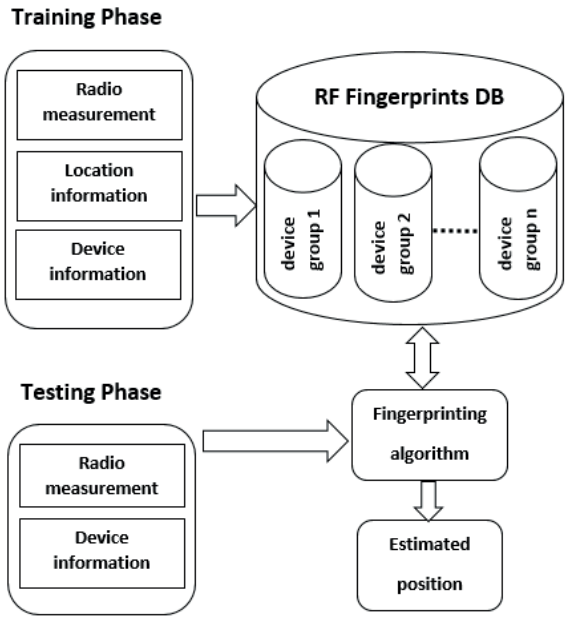


Fig. 8: RF fingerprint positioning process

VI. Conclusion

This paper has evaluated the impact of device heterogeneity on RF fingerprinting based indoor positioning system. The result indicates that if the training device and the testing device are homogeneous than positioning information can be obtained with less amount of training data and positioning accuracy improves. We suggest splitting the training database into small segments to store the fingerprints according to the device profile. At online phase, positioning request can be answered according to device type. In case of a new device or inadequate amount of training data for any device group, split segments can be merged to play traditional correlation database role.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug 2002.

[2] R. Mondal, J. Turkka, T. Ristaniemi, and T. Henttonen, "Positioning in heterogeneous small cell networks using mdt rf fingerprints," in *2013 First International Black Sea Conference on Communications and Networking (BlackSeaCom)*, July 2013, pp. 127–131.

[3] S. A. R. Zekavat and R. M. Buehrer, Eds., *Handbook of Position Location*. John Wiley & Sons, Inc., sep 2011. [Online]. Available: https://doi.org/10.1002/9781118104750
[4] K. Kaemarungsi, "Distribution of wlan received signal strength indication for indoor location determination," in *2006 1st International Symposium on Wireless Pervasive Computing*, Jan 2006, pp. 6 pp.–.
[5] S. Lee, S. W. Choi, C. Laoudias, and S. Kim, "Heterogeneous device tracking with rss variation mitigation over a radio map," *IEEE Wireless Communications Letters*, vol. 5, no. 5, pp. 552–555, Oct 2016.
[6] (2016, Sep.) Wcn3620 wireless connectivity ic. [Online]. Available: https://developer.qualcomm.com/qfile/29369
[7] (2016, Sep.) Preliminary data sheet bcm4329. [Online]. Available: http://www.cypress.com/file/298626/download
[8] E. S. Lohan, J. Torres-Sospedra, H. Leppkoski, P. Richter, Z. Peng, and J. Huerta, "Wi-fi crowdsourced fingerprinting dataset for indoor positioning," *Data*, vol. 2, no. 4, 2017.
[9] J.-H. Kim, K. S. Min, and W.-Y. Yeo, "A design of irregular grid map for large-scale wi-fi LAN fingerprint positioning systems," *The Scientific World Journal*, vol. 2014, pp. 1–13, 2014. [Online]. Available: https://doi.org/10.1155/2014/203419
[10] I. J. Quader, B. Li, W. (patrick Peng, and A. G. Dempster, "Use of fingerprinting in wi-fi based outdoor positioning," 2007.
[11] F. Yu, M. Jiang, J. Liang, X. Qin, M. Hu, T. Peng, and X. Hu, "5g wifi signal-based indoor localization system using cluster k-nearest neighbor algorithm," *International Journal of Distributed Sensor Networks*, vol. 10, no. 12, p. 247525, 2014. [Online]. Available: https://doi.org/10.1155/2014/247525
[12] P. Berkhin, "Survey of clustering data mining techniques," Tech. Rep., 2002.
[13] A. Razavi, M. Valkama, and E. S. Lohan, "K-means fingerprint clustering for low-complexity floor estimation in indoor mobile localization," in *2015 IEEE Globecom Workshops (GC Wkshps)*, Dec 2015, pp. 1–7.
[14] A. Arya, P. Godlewski, M. Campedel, and G. du Chn, "Radio database compression for accurate energy-efficient localization in fingerprinting systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1368–1379, June 2013.
[15] Z. Peng, P. Richter, H. Leppkoski, and E. S. Lohan, "Analysis of crowdsensed wifi fingerprints for indoor localization," in *2017 21st Conference of Open Innovations Association (FRUCT)*, Nov 2017, pp. 268–277.
[16] W. A. Hapsari, A. Umesh, M. Iwamura, M. Tomala, B. Gyula, and B. Sebire, "Minimization of drive tests solution in 3gpp," *IEEE Communications Magazine*, vol. 50, no. 6, pp. 28–36, June 2012.

PII

IMPROVING RF FINGERPRINTING METHODS BY MEANS OF D2D COMMUNICATION PROTOCOL

by

S Khandker, J Torres-Sospedra, T Ristaniemi 2019

Electronics, 8 (1), 97

https://doi.org/10.3390/electronics8010097

electronics

MDPI

Article

Improving RF Fingerprinting Methods by Means of D2D Communication Protocol

Syed Khandker [1,*], Joaquín Torres-Sospedra [2] and Tapani Ristaniemi [1]

[1] Faculty of Information Technology, University of Jyväskylä, Mattilanniemi 2, 40100 Jyväskylä, Finland; tapani.ristaniemi@jyu.fi

[2] Institute of New Imaging Technologies, Universitat Jaume I, Av. Vicente Sos Baynat s/n, 12071 Castellón de la Plana, Spain; jtorres@uji.es

* Correspondence: syed.i.khandker@student.jyu.fi

Received: 4 December 2018; Accepted: 11 January 2019; Published: 16 January 2019

Abstract: Radio Frequency (RF) fingerprinting is widely applied for indoor positioning due to the existing Wi-Fi infrastructure present in most indoor spaces (home, work, leisure, among others) and the widespread usage of smartphones everywhere. It corresponds to a simple idea, the signal signature in a location tends to be stable over the time. Therefore, with the signals received from multiple APs, a unique fingerprint can be created. However, the Wi-Fi signal is affected by many factors which degrade the positioning error range to around a few meters. This paper introduces a collaborative method based on device-to-device (D2D) communication to improve the positioning accuracy using only fingerprinting and the direct communication to nearby devices. The results presented in this paper show that the positioning error can be reduced around 44% by considering D2D communication in the operational stage without adding new infrastructure for fingerprinting or complex resource-consuming filters. Moreover, the presence of very large errors is significantly reduced when the collaborative positioning based on D2D is available.

Keywords: RF Fingerprinting; D2D; 5G; indoor positioning

1. Introduction and Motivation

With the rapid development of communication technology, more and more electronic equipment is being connected to the internet, which has led to a rise in innovative applications in the smart everything everywhere, such as smart homes, smart cities, and so on. The widespread application of Global Navigation Satellite System (GNSS) technology (such as GPS, Galileo, GLONASS) has solved most of the outdoor positioning problems. However, the localization accuracy sharply declines once the receiver enters a non-line of sight (NLOS) environment. Therefore, indoor positioning systems have been extensively investigated, and intense efforts have been devoted to enhancing the localization performance [1,2]. Among the current indoor positioning schemes, for instance, Radio Frequency Identification (RFID), Ultra-WideBand (UWB), Dead reckoning, image-based technology, and ultrasonic which strongly depend on extra devices, the Radio Frequency (RF) fingerprint positioning system has become a very promising and competitive technical solution for its high precision positioning by low-cost [3].

The strategy relies on the idea that every indoor location can be identified by a unique signal feature known as a fingerprint. Typical RF fingerprint (from now on, fingerprint) consists of radio measurements from multiple Access Points (AP), i.e., Received Signal Strengths (RSS) or path-loss measurements to provide a fingerprint of radio conditions at a specific location. The location of a fingerprint can be estimated using the known location of similar fingerprints previously recorded. One of the critical challenges to support such fingerprinting localization is to create and maintain

accurate fingerprint databases [4]. A site survey is often used to collect fingerprints from a targeted area. However, site surveying is a time-consuming and labor-intensive process. It requires several measurements of each fingerprint to obtain the statistical value of signal strength. Often professional work is needed for efficient site surveying. Moreover, a well-built fingerprint database become outdated as soon as the environment changes. For instance, the redistribution of APs after maintenance might degrade the accuracy of the fingerprint-based positioning system. To solve the problems mentioned above, some new kind of radio map construction techniques have been proposed, including data collection with the help of volunteers, simultaneous localization and mapping, propagation model prediction, RSS prediction based on exist fingerprints, fingerprint construction using passive crowdsourcing data [5], or by means of unsupervised techniques.

The overall accuracy of Wi-Fi based fingerprint positioning is from few meters to around hundred meters [6,7]. However, comparing the positioning accuracy of the different methods by using the published results of different studies might not be fair, since researchers are using different evaluation environments, methodologies and test-beds. Although some traditional works report that the accuracy of Wi-Fi based fingerprint positioning is around 2–3 m [8,9], it can reach a higher averaged error when the fingerprint density (including the distance between reference points and/or number of fingerprints per reference point) in the radio map is not dense enough [10] or when external resources (e.g., floor plan, magnetometer, barometer, inertial sensors) are not considered [11–13]. Thus, the accuracy highly depends on the scenario (building materials, obstacles, presence of people, among others) and on the quality of the generated radio map (deployed anchors/APs, number of independent fingerprints per reference point, distance between reference points, device diversity, among many other features).

Some previous works have identified that the noise present in the radio map and the operational fingerprint is one of the primary sources of the positioning errors [14]. The white noise present in the fingerprints in a given location has been usually modeled by a Gaussian distribution centered in the averaged RSS value with a standard deviation around 2–4 dBm depending on the scenario and environment. Although the best distribution tends to be slightly skewed [15], the Gaussian distribution has been well established and widely used in fingerprinting methods. There have been many attempts to minimize this noise, including a more detailed characterization of the RSS values (e.g., by using the Logarithmic Gaussian distance for fingerprint comparison [16]), averaging consecutive fingerprints [17], employing multiple Wi-Fi interfaces to obtain robust averaged fingerprints from independent readings [18], using ensembles as positioning system [19], or applying Gaussian Process Regression Modeling [20], among others. Most of them, requires to increase the computational complexity of the positioning algorithms and, therefore, the energy consumption at the device and/or the remote server side.

In this paper, we propose a collaborative method based on device-to-device (D2D) communication to improve the positioning accuracy. The proposed collaborative method is based on the core idea behind the ensemble, and multiple interfaces approach. The probability of having a large, or very large, positioning errors is generally lower when the prediction is based on multiple devices and/or multiple independent predictions. In a place with multiple mobile units nearby (e.g., smartphones), they can be considered independent interconnected sensors for positioning purposes. Each mobile unit could use fingerprinting to roughly estimate its position and D2D communications with other devices to refine the final position estimation by sharing the roughly estimated positions and calculating the estimated distances between the mobile units.

Green communication, as energy-efficient communication, is of paramount importance nowadays since the environment protection and energy-saving are inevitable trends [21]. A comprehensive study of energy-efficient tradeoffs involving green communication and wireless was introduced in [22]. Our paper explores the usage of D2D communication to enhance the performance of the indoor positioning systems based on fingerprinting. This approach can be considered simpler and more efficient than those based on multiple-sensor fusion and advanced filtering (Kalman filters and/or Particle filter). Moreover, the positioning accuracy is improved without adding additional

infrastructure (namely new Wi-Fi APs or Bluetooth low energy beacons) and, therefore, without increasing the energy consumption of the infrastructure needed for positioning.

The remaining of this paper is organized as follows. Section 2 introduces a review of the works related to fingerprinting, collaborative/cooperative positioning and D2D communications. Section 3 describes the basics of the RF fingerprinting technique, D2D communication aspects, Time of Arrival (TOA) of RF signal and the proposed D2D communication assisted RF fingerprint positioning method. Section 4 is devoted to the experimental setup, performance analysis of the proposed method under different conditions and discussion. Section 5 presents the conclusions and draws the research lines for further work.

2. Related Work

D2D cooperative localization has received extensive interest from the robotics, optimization, and wireless positioning research communities [23–25]. The core concept of cooperative localization is the information exchange among users in order to increase their localization accuracy. Cooperative localization was initially considered for addressing the localization problem with limited information about the current position of the mobile unit [26], but this positioning approach can also been extended to those cases where the accuracy is around a few meters, as in Wi-Fi based fingerprinting, in order to obtain more accurate position estimates and reduce positioning ambiguities [27].

Papapostolou et al. [28] showed real-time fingerprint exchange among the users to refine UE's initial location. They achieved up to 43% error reduction for 200 user collaboration, but a collaboration of such a high amount of users might be impractical in some cases or scenarios. For 20 users collaboration, they found 16% mean location error reduction through their cooperative location refinement algorithm. For the experiments, they simulated a radiomap with the IEEE 802.11 channel and the indoor log-distance path loss model in a rectangular area of 80×80 m^2 with 5 AP.

Iwase at al. [29] proposed a solution to reduce the accumulative positioning error through cooperative positioning among multiple pedestrians using smartphone and pedestrian dead reckoning. Their proposed method introduces linkage structures to simplify the trajectories of pedestrians. These structures work as a constraint to reduce the number of variables to be estimated. Moreover, they introduced the concept of communications points, where the Wi-Fi ranging error is expected to be low. Therefore, the user's tracks can be correct using this information. They found the positioning accuracy improved as the number of participants in the cooperative positioning operation increased. Two experiments were conduced: the first one was indoors and using real data from smartphones and up to 8 people, the second one corresponded to a large-scale simulation.

Vaghefi et al. [30] used round-trip time measurement for cooperative localization technique in LTE network, according to the performed simulations the higher number of collaborator, the higher positioning accuracy, but increasing number of candidates increases the complexity exponentially. They concluded ten candidates provide a balance between complexity and the performance under a LTE communication and positioning scenario.

Bargshady et al. [31] used particle filter to integrate RSS and TOA of RF signal for improving cooperative localization precise in indoor environment. But they used only three moving objects in their simulation along with eight fixed anchors. They concluded improvised positioning trend stemming from hybrid cooperative mode.

Karlsson et al. [32] showed smartphone based Wi-Fi signal measurements exchange via Bluetooth connection between the users. They used Bluetooth signal strength difference between users to calculate relative distance, which is used to evaluate the probability distribution functions of their states. Two simulation scenarios were carried out with 10 and 100 users. Their proposed method increases the average positioning performance by 28% and 22% respectively for the two cases.

Chen et al. [33] proposed a cooperative localization method to combine the fingerprint-based algorithm and the physical constraint of pairwise physical distances to refine the localization estimates for multiple users simultaneously. They found the number of peers and peers selection have a

substantial impact on accuracy performance. The experimental area was a corridor surrounding a rectangular area. They selected 80 reference points in the corridor, where the distances between two adjacent points were around 1.5–3 m. A total of 150 combination of several predefined positions were used to simulate the multiple users condition. The technique to measure the relative distance between two units was based on acoustic ranging, if they were close to each other (less than 3 m).

Cui et al. [34] studied a real-time positioning based on the D2D real-time communication, which will be reliable in future 5G cellular networks. The ranging and positioning performance were evaluated with four impulse radio waveforms. Their proposed method provided sufficient accuracy for the future real-time positioning applications.

Raveneau et al. [35] used D2D collaboration to eliminate redundant fingerprints from a crowdsourced database. According to them, intercommunication among the devices would help to know whether a fingerprint need to be recorded or the database already handled a fingerprint from that location by other devices.

Yin et al. [36] focused on a GNSS/5G integrated positioning methodology with the D2D range and angle measurements. They analyzed the characteristics of a GNSS/5G integrated system and proposed a high-efficiency D2D positioning measure protocol, named crossover multiple-way ranging, which consumes fewer communication resources. Their simulation results show that their proposed integrated positioning methodology outperforms the non-integrated one especially with more mobile terminals and accurate D2D measurements. The evaluation test-bed was in the outdoor area. The real-time kinematic positioning technique was applied to collect GNSS data with a centimeter level accuracy. They used simulated 5G D2D data.

Most of the previous studies were conducted on simulation data or in a laboratory environment. Compare to those our study reflects almost all the practical aspects of the real-world scenario. According to the authors best knowledge, this is a novel work where the D2D communication protocol is used to improve the RF fingerprint positioning method in such a sizable three-dimensional environment.

3. Material and Methods

This section includes the proposed method and introduces theoretical background concerning the proposal. Section 3.1 shows a brief description of the RF fingerprint positioning method. The D2D communication protocol in Section 3.2 explains the feasibility of using the protocol in the proposed method. Clock synchronization is crucial in TOA-based distance calculation, but it is impractical to have all the devices' synchronized clock in a wireless network. Section 3.3 shows the technique to calculate the TOA-based distance despite asynchronous clocks. The proposed method is explained in Section 3.4.

3.1. RF Fingerprint Positioning

RF fingerprint positioning refers to database matching or correlation method where UE's position can be estimated by comparing UE's RF fingerprint with a database previously recorded. Each reference (or training) fingerprint in the database is associated to a known location, which commonly corresponds to a local coordinate (X, Y), or a global coordinate (latitude, longitude), or a label (e.g., *Main Corridor*). Generally, the fingerprint database is built by site surveying, crowdsourcing or by other means. Different distance measurement metrics, e.g., Jaccard, Manhattan, Minkowski, etc., can be used to determine the signal distance between training and testing fingerprint [37,38]. Figure 1 depicts the general concept of RF fingerprint positioning method.

We have used Euclidean distance throughout the research since it is the most commonly used distance measure used in the current literature. If P and Q are two RSS vectors of test and training fingerprint respectively, the signal distance between them can be calculated as follows:

$$distance_{euclidean}(P,Q) = \sqrt[2]{\sum_{i=1}^{n}(P_i - Q_i)^2} \quad (1)$$

Here n is the length of the RSS vectors. Generally, the signal distance between the test fingerprint and all the training fingerprints are calculated. The test fingerprint's coordinate is estimated by computing the coordinates of the k reference/training fingerprints reporting the shortest signal distance. In the most simple case, the coordinate corresponding to the reference fingerprint reporting the lowest signal distance.

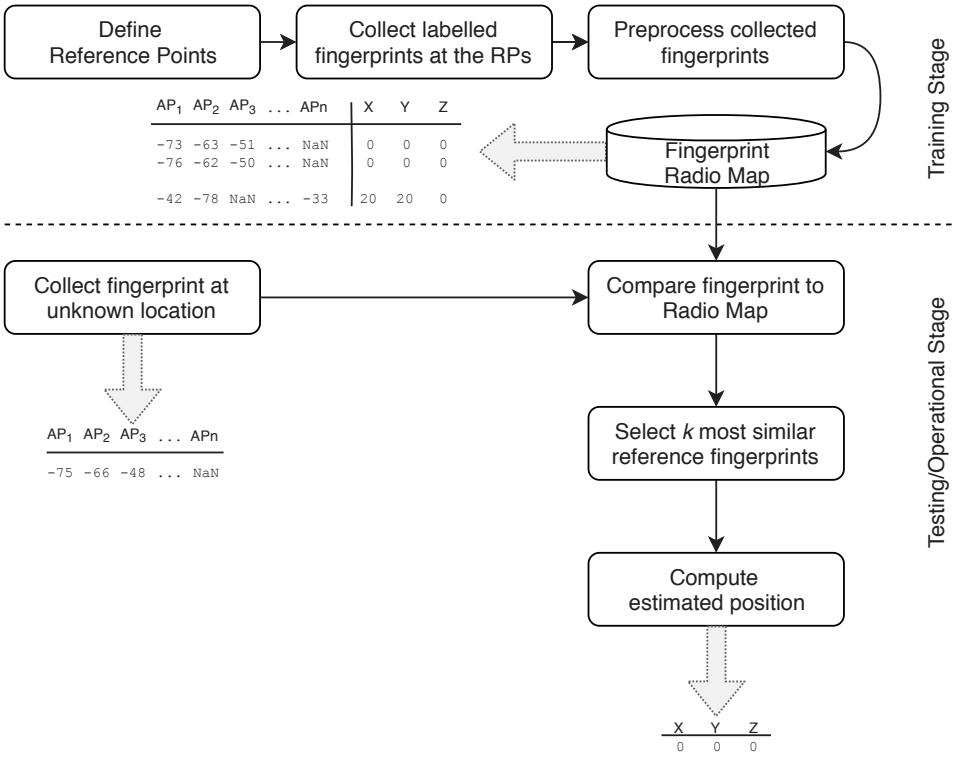


Figure 1. Diagram of matching-based RF fingerprint positioning system with examples of the radio map, operational fingerprint and estimated position.

3.2. D2D Communication Aspects

D2D communication has been specified in 3GPP LTE Release 12, that enables direct communication between nearby devices to handle particular applications such as proximity services (ProSe), content sharing, multi-party gaming, etc [39]. It is an exciting and innovative feature of next-generation cellular networks. By exploiting D2D communication, it is possible to enhance throughput, spectrum utilization and energy efficiency of the cellular network. To address future network challenges of dynamic environment adaptability, and productive use of available resources in 5G, D2D communication is considered as a potential tool. Concerning spectrum usage, D2D communication is primarily classified into two types, e.g., inband and outband. In inband D2D communication, cellular communication and D2D communication use the same spectrum licensed to the network operator, whereas outband D2D communication uses unlicensed spectrum (e.g., the free 2.4 GHz ISM band) where cellular communication does not occur [39]. Use of outband eliminates the

interference between D2D communication and network users. In terms of control, outband technology is divided into two categories, controlled and autonomous. In control mode, the radio interface is controlled by the cellular network while in autonomous mode, control of D2D communication belongs to the end users. The basic architecture of the 3GPP D2D communication architecture is shown in Figure 2.

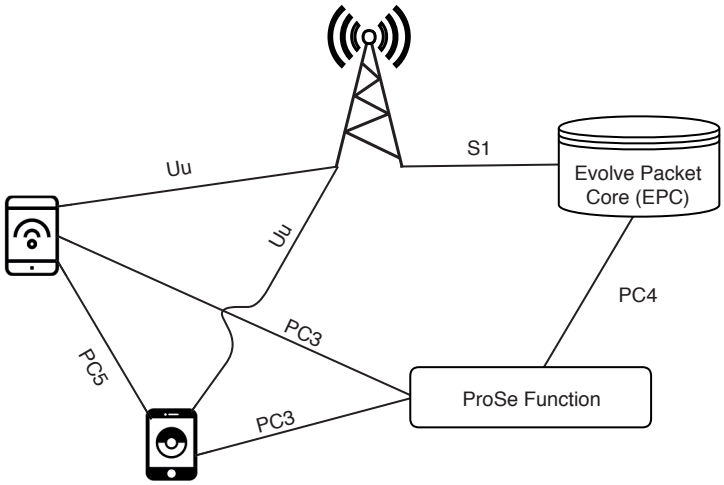


Figure 2. Basic architecture of D2D communication.

A device that wants to establish D2D communication needs to contact the ProSe function through PC3 logical interface to obtain permission. After the authorization completed, the device can start the discovery process. From a user's perspective, there are two types of peer discovery techniques, Evolved Packet Core (EPC)-level discovery and direct discovery. At direct discovery, a device would search other devices with ProSe capability in their proximity using the PC5 interface. When two (or more) ProSe-enabled devices have discovered each other, they can start direct communication over the direct link between them. From a network perspective, peer discovery can be controlled hardly or softly by the base station [40]. The interface between the EPC and the ProSe function is called PC4. Generally, a D2D link connects a device to another following in a single-hop communication. A multi-hop network composed of D2D connections is also possible. In a multihop D2D network, the central devices act as relays between two devices. In 5G, cell size will be smaller, and the density of the devices will be higher, 150 devices per cell are assumed in 3GPP for evaluating D2D discovery [41]. As a result, less inter-device distance is expected. Small inter-device distance increases the probability of Line of Sight (LOS) communication among devices that results in a better TOA-based ranging accuracy [42]. Without any relay, the effective D2D communication range is expected to be around 30 m [43].

Due to the distributed structure of control and mobile devices being computationally poor to apply complex encryption mechanisms and other security technologies, D2D communication can be vulnerable to malicious attacks. Ramirez [44] mentioned three major security and privacy problems, these are ubiquitous data collection, unexpected uses of data, and heightened security risks. Many researchers [45,46] have studied RF fingerprint based wireless security system. Zhang et al. [47] proposed a D2D communication security authentication process based on RF fingerprint identification. The advantage of using fingerprint for security purpose is that fingerprint computes the inherent hardware differences of the transmitter for identification, so this method cannot be completely copied. Their simulation results show that when Signal to Noise Ratio (SNR) is higher than 8 dB, the device recognition rate is 100%. RF fingerprint can be used for positioning and security purpose simultaneously.

3.3. TOA Based Distance Determination

Generally, the time required for a signal to travel from one node to another is used to estimate the distance between the nodes. According to our proposed method, a device would request another device to exchange their position information, TOA of the same communication can be utilized to measure the distance between them. A brief process of two-way time of arrival is illustrated in Figure 3.

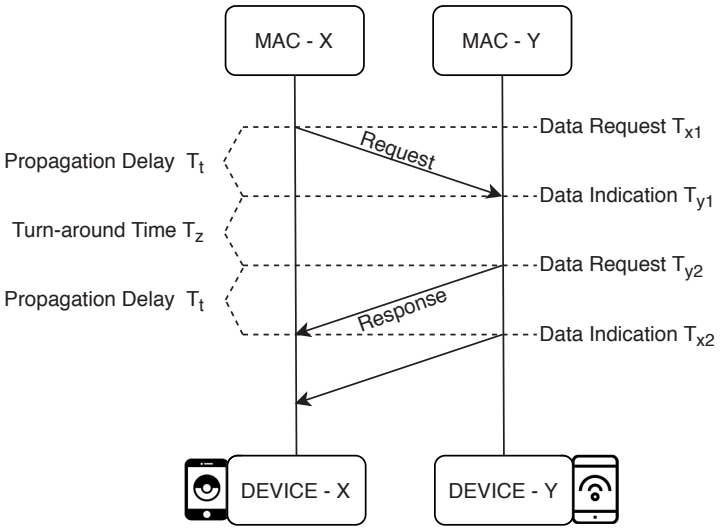


Figure 3. Two-way TOA diagram.

Two-way time of arrival is a ranging protocol [48], based on the reception of timestamp packets distance between two devices can be estimated in the absence of time synchronization. Assuming the propagation delays of the request and response frames are the same, the TOA and physical distance between the two devices can be calculated as follows:

$$T_t = \frac{(T_{x2} - T_{x1}) - (T_{y2} - T_{y1})}{2} \tag{2}$$

$$d_{xy} = T_t \times c \tag{3}$$

Here c is the speed of light. Akiyama et al. [49] proposed a method using a smartphone to measure the TOA, employ modulated light with a signal for the time-of-flight short measurement, they achieved positioning error of less than 10 mm. However, in an indoor environment, most connections between devices are NLOS as the LOS paths are blocked by wall, ceiling, or furniture. The LOS path is often mixed with multiple time-delayed NLOS paths. NLOS can degrade the D2D ranging since they have significant positive biases which make the measured pair-wise distances much larger than their actual values [50]. Some studies have focused on distinguishing NLOS connections from LOS connections. An identification technique based on maximum likelihood estimation is described in [51]. By exploring physical layer information 95% LOS was identified in [52]. Cui et al. [34] showed how to detected actual TOA from an NLOS / multipath channel from received signal energy. According to them in a multipath channel with a low signal to noise ratio, some low energy level value may appear before the maximum energy value. By setting a threshold value, actual TOA can be determined as shown in Figure 4.

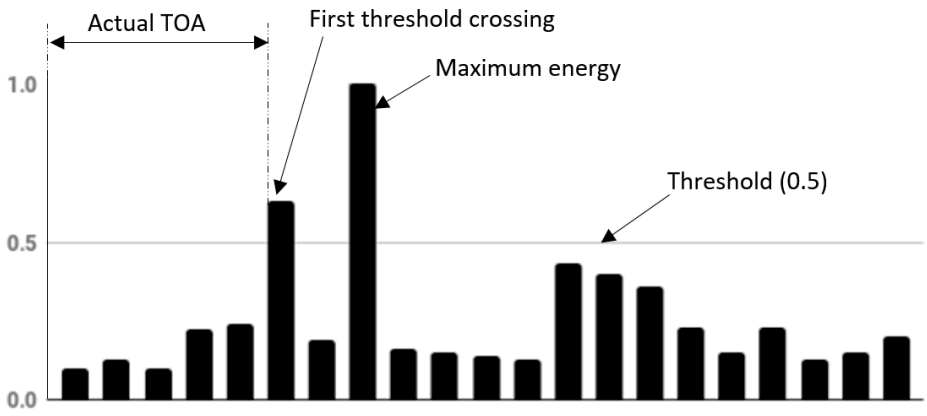


Figure 4. TOA estimation based on the received signal energy.

3.4. Description of the Proposed Method: D2D Communication Assisted RF Fingerprint Positioning

A D2D communication underlying cellular network is considered as a key technique to alleviate the exponential growth of user demands for high data rates, low power consumption, and spectrum efficiency. In D2D communication, two devices under the proximity to each other can communicate directly without passing through the Evolved Node B (eNB) (described in Section 3.2). Through direct communication, a device would be able to exchange information with other devices, e.g., location, quality of service, observe radio condition, etc. Information obtained from D2D communication can be used to verify the location of the User Equipment (UE) to mitigate the erroneous position information received from the fingerprinting method. Figure 5 shows the D2D communication assisted fingerprint positioning method that we propose in this work. Let's assume that, device *A* needs to know its position. The eNB can provide device *A*'s position based on device *A*'s fingerprint. Due to numerous factors that affect the strength of a radio signal, e.g., multi-path propagation, obstruction, and temperature, the provided position information can be inaccurate. Device *A* can ask another device *B* about *B*'s position, *B* can respond to *A* with *B*'s position information obtained from eNB.

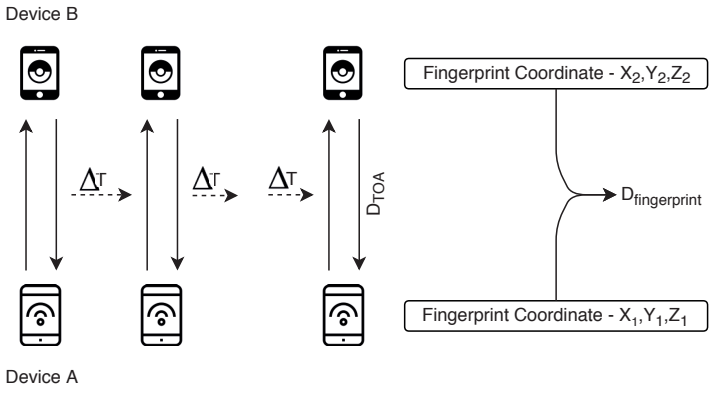


Figure 5. D2D communication assisted RF fingerprint positioning method.

From their communication, device *A* would be able to calculate the physical distance between them by two procedures. Firstly, by signal's time of arrival method (D_{TOA}) secondly, from their coordinate, that they received from eNB ($D_{fingerprint}$). If these two values are the same or very close, then it can be supposed that the obtained position information from the fingerprinting method is correct as the outcome of the fingerprint positioning is verified by another method. This process can be repeated several times after a short time interval ΔT.

$$\delta_i = |D_{TOA_i} - D_{fingerprint_i}| \tag{4}$$

Out of the several measurements $i = 1, 2, ..., n$ device A can select fingerprint positioning response from the eNB containing minimum error indicator (δ) to increase the positioning confidence. If device A is unable to make a conclusive decision from the process, it should be satisfied with the position information received from eNB. Figure 6 shows the flowchart of the proposed method.

In Figure 6, a UE with unknown position requests eNB for its location information, at the same time it requests another device to send that device's position. From these two devices multiple inter-communication, UE can calculate error indicator (δ) several times according to Equation (4). By accepting minimum δ containing response from eNB, UE can fine-tune its location information. The main idea of our proposed method is to verify the fingerprint positioning response by TOA-based distance in order to rectify the erroneous response of eNB. To increase the confidence level of positioning the verification process (or communication with other devices) is repeated several times within a short period, this process creates the advantage to select the best positioning response. Generally, the motion of the human in an indoor environment is languid, so the change of position in a few milliseconds does not put an observable effect. The number cooperation with other devices depends on the user, each measurement increases the positioning calculation complexity (see Section 4.6); therefore the amount of cooperation (δ calculation) should be kept low. We initialized the δ cordiality, $\tau = 3$ in the proposed Algorithm 1. Our algorithm works as follow:

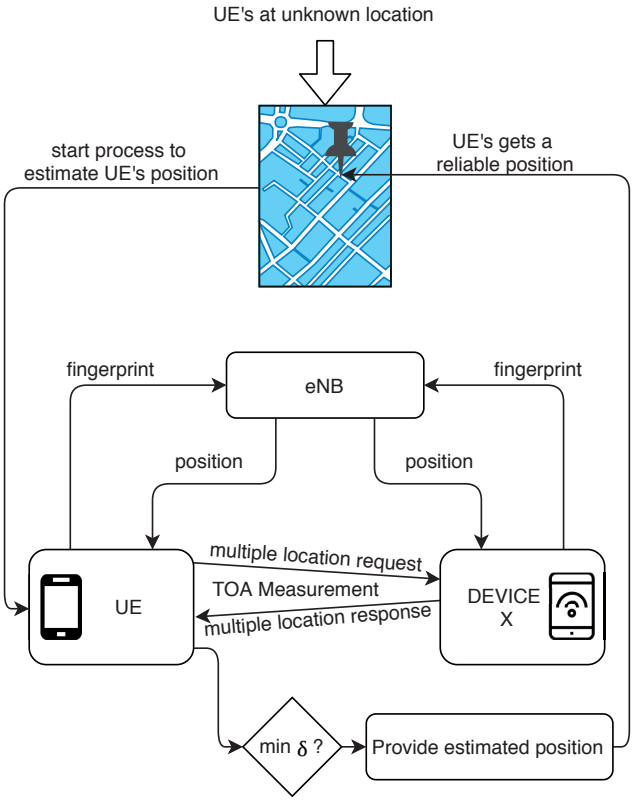


Figure 6. Flowchart of the proposed method.

Algorithm 1 Output = D2D(τ = 3)

1: **while** cardinality of $\delta \leq \tau$ **do**
2: `Step 1:` UE with unknown location information sends a request to the eNB to know it's location.
3: `Step 2:` The eNB responses UE's request and send UE's location according to the fingerprinting method (see Section 3.1).
4: `Step 3:` UE sends a request to another nearby device to know other device's location.
5: `Step 4:` Other device responses UE's request and send it's location information to the UE (Other devices obtain their location from the eNB according to step 1–2).
6: `Step 5:` UE calculates the physical distance between UE and another device by two different procedures, and calculates δ.
`Step 5.1:` Calculate distance by coordinate information obtained from the system through RF fingerprinting.
`Step 5.2:` Calculate distance from TOA measurement (here TOA has been emulated by true physical distance).
`Step 5.3:` Calculate δ.
7: **end while**
8: Fingerprint positioning response from the eNB containing the minimum δ value is selected as UE's position.

Collected fingerprints from the targeted area generally stored in the server located at under the control of eNB. For D2D direct communication, responsibility data management and security belong to the participated device, and this topic is under intensive research [53,54]. According to our proposed method, only position information of other device is needed to be passed through the D2D communication link. Such a small amount of information can easily be fit within 1 kilo bits packet including address, header, encryption etc. To prevent the exploitation of the data devices should communicate with other secured devices, with whom confidentiality, integrity, and privacy are maintained under a common agreement.

4. Experiment and Results

This section introduces the evaluation setup, including the database description and shows the main results derived from the evaluation of the proposed method and the effects of the D2D distance, motion direction, amount of D2D measurements, among others.

4.1. Database Description

In this research article, we have used the Wi-Fi crowdsourced fingerprint database created by a research group from Tampere University of Technology (TUT) [13]. The database and the benchmarking software are distributed under the open-source MIT license and can be found on the EU Zenodo repository [55]. The measurements were taken in a five-floor building in Tampere (23.85580° N, 61.44585° E), Finland. The five-floor building has a footprint of about 22,570 m^2 (208 m $\times$ 108 m). Total 4648 fingerprints were collected which were then split uniformly randomly 15% for the training and 85% for the test purpose. Volunteer users installed the android application on their devices and reported the correct location (based on a manual input on the map) to the server. The server stored the location reported by the user, the time stamp, the device model, the MAC addresses of the heard access points and the RSS from each AP. A local coordinate system e.g., (x, y, z) = (123.45, 14.71, 0) was used instead of the global WGS84 commonly used in other applications. During the measurements, a total of 991 AP were heard, the exact position of the APs was not known. Figure 7 shows the fingerprint database.

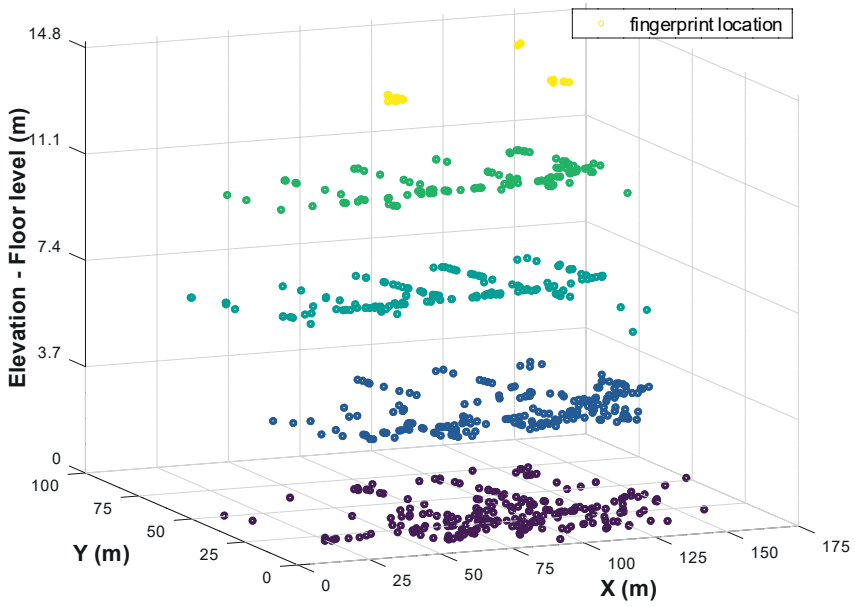


Figure 7. Visual representation of the reference fingerprints location. Fingerprints are colored according to the elevation (floor level).

According to the benchmark result provided by the TUT research group, the mean 2D and 3D positioning errors of this database were in the order of 8–10 m in the best cases. They made it deliberately challenging by keeping the training data very low, only 15% (i.e., 0.03 fingerprints/m^2). The fingerprints locations were recorded manually and accurately, so the physical distance between fingerprints (in other senses, devices) according to the coordinate value are true. To the best of the authors' knowledge, this database or any other standard fingerprint database available on the internet does not contain TOA information among the fingerprints. We have emulated true physical distance among the fingerprints as TOA-based distance.

4.2. D2D Communication Assisted RF Fingerprint Positioning

From the test database, 1860 test fingerprints were chosen to evaluate the proposed method, since all of them are not suitable for the experiment carried out in this work. Training database remained unaltered, containing 697 fingerprints as in Zenodo repository. From several communications with a secondary device (see Section 3.4) minimum δ containing coordinate were selected as the test fingerprint's position according to the proposed method. On the other hand, the traditional k-NN method has been applied to compute the same test fingerprint's coordinate to compare the effectiveness of the proposed method. If the UE (the primary device that needs to know the position) is stationary, then it can not get the benefit from the proposed method to select the best result out of several options since it would experience the same positioning information all the time. UE needs to change its positioning, but the other secondary devices can either be in motion or stationary. Figure 8 shows the comparison between our proposed method and the traditional fingerprint positioning method. This comparison is based on the empirical Cumulative Distribution Function (CDF) of the 3D positioning error highlighting the median error (50%) and the spherical error (95%) as suggested by the ISO18305:2018 standard for real-time locating systems [56], and the third quartile (75%) as suggested by the EvAAL framework [57]. For this experiment, we considered two scenarios for the behavior of the secondary devices: Motion and Stationary. Physical distance among the devices was chosen randomly. CDF of 3D positioning error in Figure 8 shows that compared to the traditional positioning method, our proposed method provides a significant amount of better accuracy. The mean 3D positioning accuracy through traditional method is 8.68 m, that is similar to the results provided in [13]. Our proposed method managed to rectify the error and significantly reduce the mean 3D error.

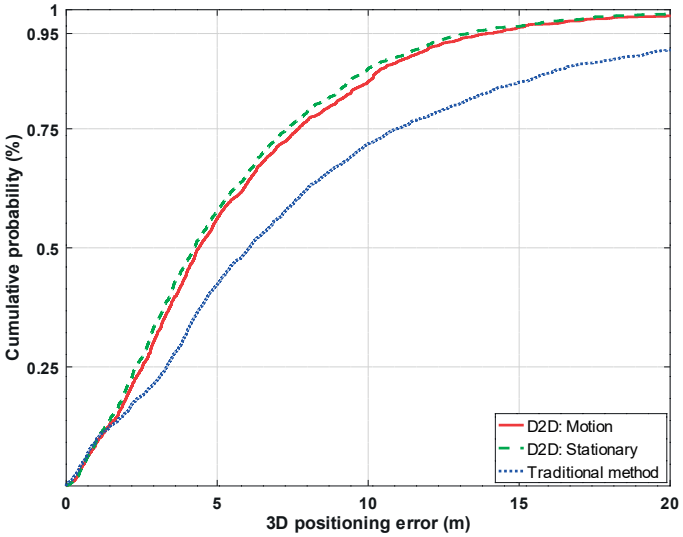
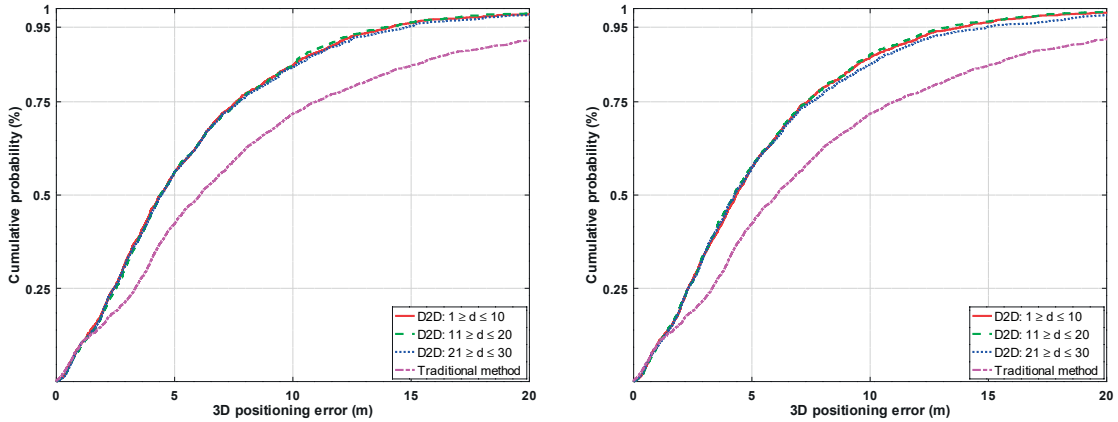


Figure 8. CDF of the positioning errors provided by the proposed method based on D2D communications (motion and stationary scenario for the secondary devices) and traditional method based on k-NN.

The results indicate that the proposed method provided better results if the other secondary devices are stationary, with a mean 3D error of 5.34 m. Nonetheless, our proposed method also provides good results, with a 5.59 m mean positioning accuracy, when the primary device exchanges information with devices which are in motion. It is worth to highlight that the proposed method prevents very large positioning errors. With the traditional method, almost 10% of operational samples have a positioning error higher than 20 m, whereas the presence of such kind of errors is marginal when D2D communications are considered.

4.3. D2D Distance Effect

To observe the effect of physical distance among the devices, in the proposed method three distance ranges were set-up: (1) 1 to 10 m; (2) 11 to 20 m, and (3) 21 to 30 m. In both scenarios, motion and stationary, the physical distance between UE and other devices did not put a significant impact. Figure 9 shows that, regardless of the distance range (denoted by d) among the devices, the positioning accuracy remains almost the same.



(**a**) Other devices are in motion (**b**) other devices are stationary

Figure 9. Effect of physical distance among the devices in the proposed method shown as the CDF of the positioning errors.

Based on the outcomes of Figures 8 and 9, it can be stated that the accuracy level improves significantly if the RF fingerprint positioning information is verified with a true value. Moreover, some features of the reference device (distance, motion/stationary, etc.) do not affect the D2D collaborative positioning.

4.4. Effect of Motion Direction

In an open space, it is highly likely that during the movement a user changes her/his direction. Since the motion did not affect the positioning accuracy in the previous experiment, it indicates that the moving direction might not have a significant effect on the proposed method. Effect of the change of direction has been investigated in this experiment. Since different test fingerprint has different position and orientation, it is impractical to maintain common axis reference for angle calculation. However, since the database contains local coordinate, we used the origin point (XY = [0, 0]), UE coordinate, and other devices coordinate to make a triangle and calculate the angle of motion. In Figure 10 it can be seen that 351 test fingerprint (FP) took assistance from such devices which changed their course between 0° to 30° angular direction resulted in a mean 6.12 m error.

307 test fingerprints communicated with other devices which were moving from 30° to 60° angle caused 5.59 m mean error. 403 test fingerprints communicated with other devices which were moving from 60° to 90° angle caused 5.42 m mean error. 276 test fingerprints communicated with other devices which were moving from 90° to 120° angle caused 5.28 m mean error. 223 test fingerprints communicated with other devices which were moving from 120° to 150° angle caused 5.84 m mean error. 290 test fingerprints communicated with other devices which were moving from 150° to 180° angle caused 5.37 m mean error. From all the segments in Figure 10 it can be seen that the mean positioning error is very close to the overall mean positioning error of the system, no particular characteristic is observable due to the change of movement direction. However, the highest difference in mean error is between the devices moving from 0° to 30° (6.12 m) and the devices moving from 90° to 120° (5.28 m).

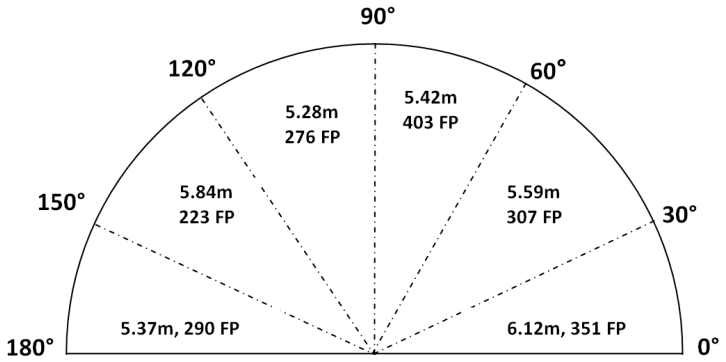


Figure 10. Positioning accuracy based on other devices' moving direction.

4.5. Amount of Measurements

D2D communication assisted positioning performances mentioned above are based on three measurements, that means a device communicated with another device for three times. With the purpose to check if the accuracy results are affected by the number of measurements considered, a test considering up to seven measurements was carried out. Results are shown in Figure 11.

From Figure 11 it can be seen that four and five times measurement provides better accuracy than that of baseline three times measurement. Six and seven times measurement between UE and other devices provide even better accuracy. However, the computational load increases as the number of considered references are included. There should be a trade-off between positioning accuracy and the number of measurement to keep the calculation complexity and energy consumption as low as possible.

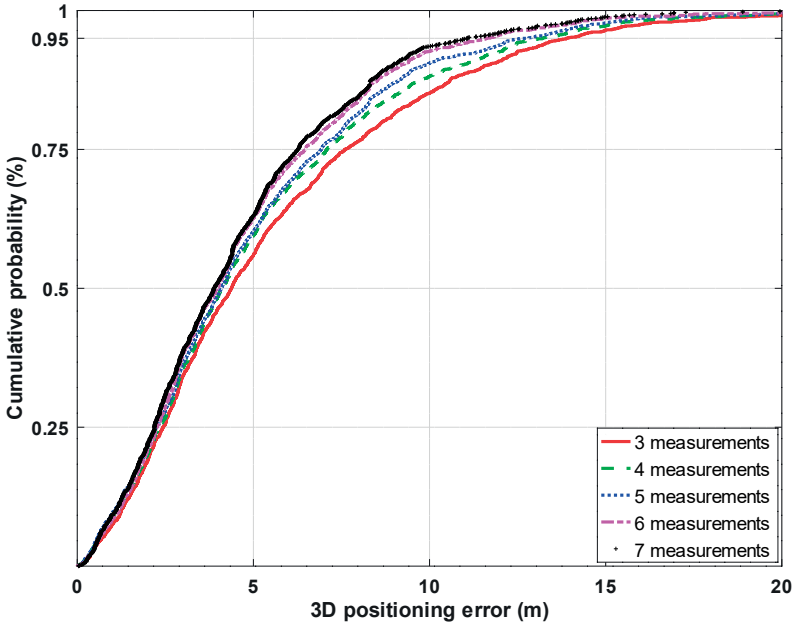


Figure 11. Effect of the increase on the amount of measurements in the proposed method shown as the CDF of the positioning errors.

4.6. Processing Time vs. Performance

Compare to the traditional fingerprint positioning, D2D collaborative positioning takes much longer processing time. The additional time is spent to communicate with other devices, acquiring positioning information for own and other devices, and processing delay. The more cooperation, the more processing time. Our laboratory computer was running on windows 8.1 operating system with Intel core i-5 processor having 2.30 GHz clock speed and 8 GB RAM. Positioning program was coded in Matlab 8.4 software. Figure 12 shows the relation between performance and processing time.

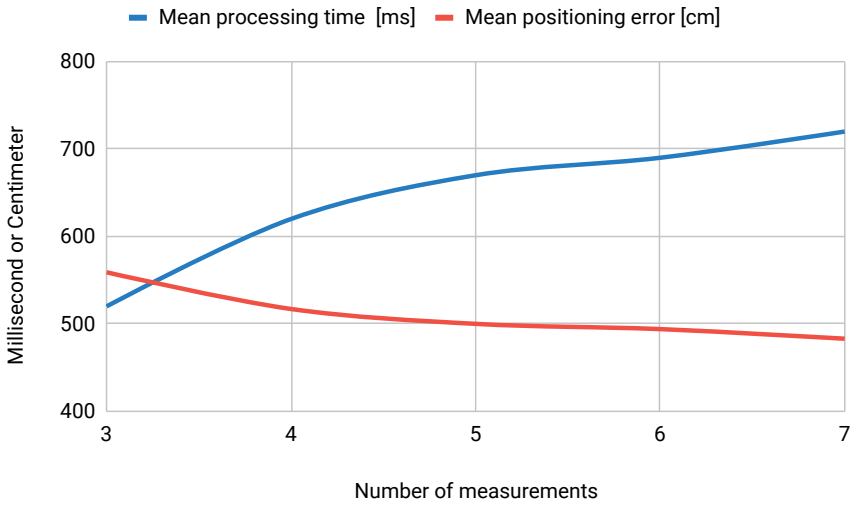


Figure 12. Processing time vs. performance (Instead of meter, positioning error has been expressed in centimeter unit to make the figure more sensible where two scale -time and performance- come closer to each other.).

According to the traditional fingerprint method the recorded mean processing time was 103 millisecond (ms) per test fingerprint, that resulted in 8.68 m mean positioning error. On the contrary, when the proposed method was applied, positioning error decreased but processing time raised over 500 ms per test fingerprint. From Figure 12 it can be seen that processing time raised around 520 ms but positioning error decreases to 559 cm (or 5.59 m) when UE starts to communicate with other devices at least 3 times. Ascending trend of processing time and descending trend of positioning error can be seen in Figure 12. A decision regarding optimal resource management can be made based on this type of plotting.

4.7. Accuracy Limit

All the experiments in this paper were carried out over a public dataset which was collected by means of crowdsourcing. In contrast to professional deployments with a dense radio map (small grid and many training fingerprints), the density of training fingerprints in the selected database was not so high only 0.03 fingerprints/m^2. This low density might have contributed to degrading the positioning accuracy. An additional experiment was carried out to compare the accuracy of the proposed D2D method (best case) with respect to the maximum possible accuracy for the database. This maximum accuracy is computed as the physical distance of the test fingerprint location with respect the nearest training fingerprint's location in the real-world (not RSS-space). Figure 13 shows the comparison between the maximum accuracy limit of the database and the best-achieved accuracy through the proposed method.

The best mean 3D positioning accuracy through the proposed method is 4.83 m while the maximum mean accuracy limit for the database is 1.34 m. Apart from the unavoidable error, our proposed method offers mean 3.50 m level accuracy for such a vast and complex indoor environment which has strong merit.

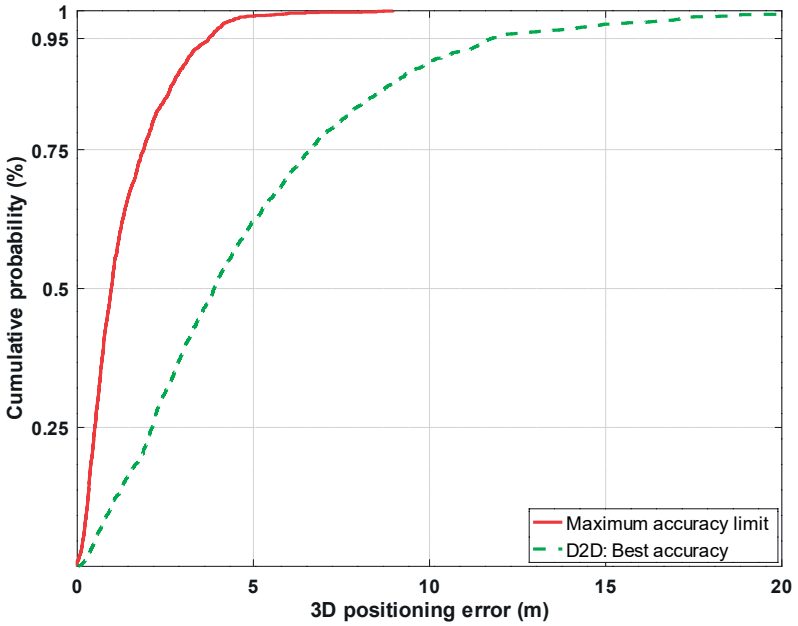


Figure 13. The accuracy limit in the proposed method shown as the CDF of the positioning errors.

4.8. Error Indicator

We further investigated the relationship between positioning error and error indicator δ (see Equation (4)). The results of this analysis are shown in Table 1. No measurement was found containing a value of δ equal to 0. From Table 1 it can be seen that when the value of δ increases, positioning error

also increases. Therefore, δ can be used as an indicator, when UE experience high δ value it should keep communicating with other devices in order to have better positioning result.

Table 1. Relationship between error indicator and positioning error.

Value of δ	Amount of Test Fingerprints	Mean 3D Positioning Error
<0.5 m	575	3.92 m
0.5 m to <1 m	360	4.61 m
1 m to <2 m	450	4.73 m
2 m to <3 m	197	5.29 m
3 m to <4 m	120	5.86 m
4 m to <5 m	65	6.53 m
>5 m	93	8.20 m

4.9. Successful Floor Detection

Finally, we checked the successful floor detection rate. In Figure 14 it can be seen compare to the traditional method, our proposed method shows slightly better correct floor detection rate with 94.46%. Although the absolute increase of performance (as successful floor detection rate) has been just 1.45% with respect to the traditional method, the relative floor detection error has been reduced in a 20% (from 6.99% to 5.45%). While all the benchmark results for correct floor detection rate were in between 85% to 92% in the previous study [13].

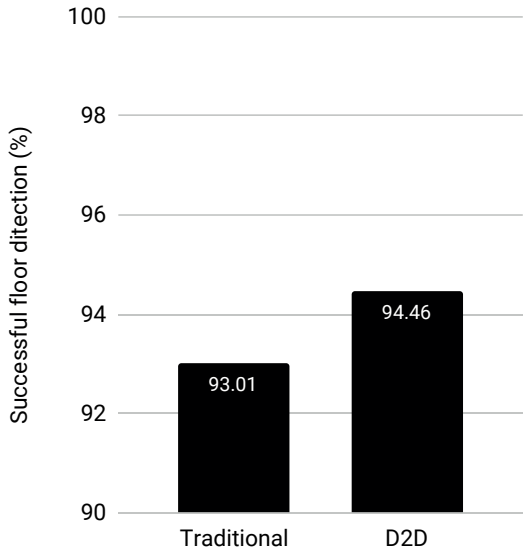


Figure 14. Floor detection rate (%) of the traditional fingerprinting and D2D-based proposed methods.

5. Conclusions

This paper mainly focused on the advantage of using D2D communication features in the RF fingerprint positioning method. Experiments were done on a dedicated RF fingerprint database where the measurement was very accurate and reflected all most all the real world aspects. Since the database does not contain TOA information we had to emulate that by true physical distance, and that has been supported by the proper literature. The main contribution of this paper is to invent a procedure to verify the RF fingerprint positioning information with true value in order to eliminate noisy answer from the system. Through the proposed method about 44% better mean 3D positioning accuracy

was achieved. Positioning accuracy for the different D2D communication scenario was evaluated. It has been observed that motion and distance among the devices do not hamper the proposed idea. Moreover, according to the presented results, it seems that the motion direction has not a huge impact on the system accuracy. However, by increasing the amount of measurement better accuracy can be achieved, i.e., better results can be obtained by using more measurements between the two devices. Although increasing the number of measurements from three to seven reduces the positioning error in around a 13% (from 5.59 m to 4.83 m), this increase is at the expense of increasing the computational burden around a 40% (from 520 ms to 720 ms). If the accuracy and the computational burden are balanced, the optimal number of measurements is between 3 and 4.

Apart from TOA, the true physical distance among the devices can be measured by sound, infrared light, camera flash, the signal strength of D2D communication, etc. In the future, we shall try to transfer the calculation burden to the eNB side. Instead of D2D distance, we will explore the UE-eNB true distance, which can also be used to verify the RF fingerprint positioning output. Finally, as further work, we will consider to include power consumption restrictions in the indoor positioning systems due to the rising interest in developing more energy-efficient protocols and algorithms to reduce their economic and environmental impact and promote Green Communications.

Author Contributions: S.K. conceived, designed and performed the experiments; S.K. and J.T.-S. wrote the paper. J.T.-S. and T.R. guided the paper writing and reviewed the paper. All authors read, revised and approved the final manuscript after the peer-reviewing process.

Funding: Syed Khandker express his warm thanks to Ellen and Artturi Nyyssösen foundation for its financial support. Joaquín Torres-Sospedra gratefully acknowledges funding from European Union's Horizon 2020 Research and Innovation programme under the Marie Skłodowska Curie grant agreement No. 813278 (A-WEAR: A network for dynamic wearable applications with privacy constraints, http://www.a-wear.eu/).

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Liu, H.; Darabi, H.; Banerjee, P.; Liu, J. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Trans. Syst. Man Cybern. Part C* **2007**, *37*, 1067–1080. [CrossRef]
2. Deak, G.; Curran, K.; Condell, J. A survey of active and passive indoor localisation systems. *Comput. Commun.* **2012**, *35*, 1939–1954. [CrossRef]
3. He, S.; Chan, S.G. Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 466–490. [CrossRef]
4. Zhou, X.; Chen, T.; Guo, D.; Teng, X.; Yuan, B. From one to crowd: A survey on crowdsourcing-based wireless indoor localization. *Front. Comput. Sci.* **2018**, *12*, 423–450. [CrossRef]
5. Luo, C.; Hong, H.; Chan, M.C. PiLoc: A self-calibrating participatory indoor localization system. In Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, Berlin, Germany, 15–17 April 2014; pp. 143–153.
6. Mautz, R. *Indoor Positioning Technologies*; ETH Zurich: Zurich, Switzerland, 2012.
7. Del Peral-Rosado, J.A.; Raulefs, R.; López-Salcedo, J.A.; Seco-Granados, G. Survey of Cellular Mobile Radio Localization Methods: From 1G to 5G. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 1124–1148. [CrossRef]
8. Bahl, P.; Padmanabhan, V. RADAR: An in-building RF-based user location and tracking system. In Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Tel Aviv, Israel, 26–30 March 2000; pp. 775–784.
9. Youssef, M.; Agrawala, A.; Shankar, A.U. WLAN location determination via clustering and probability distributions. In Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, Fort Worth, TX, USA, 26–26 March 2003.
10. Meneses, F.; Moreira, A.; Costa, A.; Nicolau, M.J. Chapter 4: Radio Maps for Fingerprinting in Indoor Positioning. In *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*; Conesa, J., Ed.; Academic Press: Cambridge, MA, USA, 2019; pp. 69–95.

11. Potortì, F.; Park, S.; Jiménez Ruiz, A.R.; Barsocchi, P.; Girolami, M.; Crivello, A.; Lee, S.Y.; Lim, J.H.; Torres-Sospedra, J.; Seco, F.; et al. Comparing the Performance of Indoor Localization Systems through the EvAAL Framework. *Sensors* **2017**, *17*, 2327. [CrossRef]
12. Xiao, J.; Zhou, Z.; Yi, Y.; Ni, L.M. A Survey on Wireless Indoor Localization from the Device Perspective. *ACM Comput. Surv.* **2016**, *49*. [CrossRef]
13. Lohan, E.S.; Torres-Sospedra, J.; Leppäkoski, H.; Richter, P.; Peng, Z.; Huerta, J. Wi-Fi Crowdsourced Fingerprinting Dataset for Indoor Positioning. *Data* **2017**, *2*, 32. [CrossRef]
14. Torres-Sospedra, J.; Moreira, A. Analysis of Sources of Large Positioning Errors in Deterministic Fingerprinting. *Sensors* **2017**, *17*, 2736. [CrossRef]
15. Kaemarungsi, K.; Krishnamurthy, P. Analysis of WLAN's received signal strength indication for indoor location fingerprinting. *Pervasive Mob. Comput.* **2012**, *8*, 292–316. [CrossRef]
16. Cramariuc, A.; Huttunen, H.; Lohan, E.S. Clustering benefits in mobile-centric WiFi positioning in multi-floor buildings. In Proceedings of the 2016 International Conference on Localization and GNSS (ICL-GNSS), Barcelona, Spain, 28–30 June 2016; pp. 1–6.
17. Prasithsangaree, P.; Krishnamurthy, P.; Chrysanthis, P. On indoor position location with wireless LANs. In Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Lisboa, Portugal, 15–18 September 2002; pp. 720–724.
18. Moreira, A.; Silva, I.; Meneses, F.; Nicolau, M.J.; Pendao, C.; Torres-Sospedra, J. Multiple simultaneous Wi-Fi measurements in fingerprinting indoor positioning. In Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, Japan, 18–21 September 2017; pp. 1–8.
19. Trawiński, K.; Alonso, J.M.; Hernández, N. A multiclassifier approach for topology-based WiFi indoor localization. *Soft Comput.* **2013**, *17*, 1817–1831. [CrossRef]
20. Richter, P.; Toledano-Ayala, M. Revisiting Gaussian Process Regression Modeling for Localization in Wireless Sensor Networks. *Sensors* **2015**, *15*, 22587–22615. [CrossRef] [PubMed]
21. Chen, J.; Chen, X.; Liu, T.; Lei, L. Toward Green and Secure Communications over Massive MIMO Relay Networks: Joint Source and Relay Power Allocation. *IEEE Access* **2017**, *5*, 869–880. [CrossRef]
22. Mahapatra, R.; Nijsure, Y.; Kaddoum, G.; Hassan, N.U.; Yuen, C. Energy Efficiency Tradeoff Mechanism Towards Wireless Green Communication: A Survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 686–705. [CrossRef]
23. Savarese, C.; Rabaey, J.M.; Beutel, J. Location in distributed ad-hoc wireless sensor networks. In Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal, Salt Lake City, UT, USA, 7–11 May 2001; pp. 2037–2040.
24. Howard, A.; Matarić, M.J.; Sukhatme, G.S. Localization for Mobile Robot Teams: A Distributed MLE Approach. In *Experimental Robotics VIII*; Siciliano, B., Dario, P., Eds.; Springer: Berlin/Heidelberg, German, 2003; pp. 146–155.
25. Cheng, B.H.; Hudson, R.E.; Lorenzelli, F.; Vandenberghe, L.; Yao, K. Distributed Gauss-Newton method for node loclaization in wireless sensor networks. In Proceedings of the IEEE 6th Workshop on Signal Processing Advances in Wireless Communications, New York, NY, USA, 5–8 June 2005; pp. 915–919.
26. Fox, D.; Burgard, W.; Kruppa, H.; Thrun, S. Collaborative Multi-Robot Localization. In *Mustererkennung 1999*; Förstner, W., Buhmann, J.M., Faber, A., Faber, P., Eds.; Springer: Berlin/Heidelberg, German, 1999; pp. 15–26.
27. Buehrer, R.M.; Wymeersch, H.; Vaghefi, R.M. Collaborative Sensor Network Localization: Algorithms and Practical Issues. *Proc. IEEE* **2018**, *106*, 1089–1114. [CrossRef]
28. Papapostolou, A.; Xiao, W.; Chaouchi, H. Cooperative fingerprint-based indoor localization using Self-Organizing Maps. In Proceedings of the 2011 7th International Wireless Communications and Mobile Computing Conference, Istanbul, Turkey, 4–8 July 2011; pp. 1814–1819.
29. Iwase, T.; Shibasaki, R. Infra-free indoor positioning using only smartphone sensors. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation, Montbeliard-Belfort, France, 28–31 Octorber 2013; pp. 1–8.
30. Vaghefi, R.M.; Buehrer, R.M. Cooperative RF pattern matching positioning for LTE cellular systems. In Proceedings of the 2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC), Washington, DC, USA, 2–5 September 2014; pp. 264–269.

31. Bargshady, N.; Pahlavan, K.; Alsindi, N.A. Hybrid WiFi/UWB, cooperative localization using Particle Filter. In Proceedings of the 2015 International Conference on Computing, Networking and Communications, Garden Grove, CA, USA, 16–19 February 2015; pp. 1055–1060.
32. Karlsson, M.; Karlsson, F. Cooperative indoor positioning by exchange of bluetooth signals and state estimates between users. In Proceedings of the 2016 European Control Conference (ECC), Aalborg, Denmark, 29 June–1 July 2016; pp. 1440–1444.
33. Chen, L.; Yang, K.; Wang, X. Robust Cooperative Wi-Fi Fingerprint-Based Indoor Localization. *IEEE Internet Things J.* **2016**, *3*, 1406–1417. [CrossRef]
34. Cui, X.; Gulliver, T.A.; Song, H.; Li, J. Real-Time Positioning Based on Millimeter Wave Device to Device Communications. *IEEE Access* **2016**, *4*, 5520–5530. [CrossRef]
35. Raveneau, P.; D'Alu, S.; Rivano, H. Localisation based on Wi-Fi fingerprints: A crowdsensing approach with a device-to-device aim. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 13–17 March 2017; pp. 321–325.
36. Yin, L.; Ni, Q.; Deng, Z. A GNSS/5G Integrated Positioning Methodology in D2D Communication Networks. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 351–362. [CrossRef]
37. Torres-Sospedra, J.; Montoliu, R.; Trilles, S.; Óscar, B.; Huerta, J. Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems. *Expert Syst. Appl.* **2015**, *42*, 9263–9278. [CrossRef]
38. Minaev, G.; Visa, A.; Piché, R. Comprehensive survey of similarity measures for ranked based location fingerprinting algorithm. In Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, Japan, 18–21 September 2017; pp. 1–4.
39. 3rd Generation Partnership Project (3GPP). TR 22.803 V12.2.0, Feasibility Study for Proximity Services (ProSe). Available online: http://www.3gpp.org/ftp/Specs/archive/22_series/22.803/22803-c20.zip (accessed on 10 November 2018).
40. Höyhtyä, M.; Apilo, O.; Lasanen, M. Review of Latest Advances in 3GPP Standardization: D2D Communication in 5G Systems and Its Energy Consumption Models. *Future Internet* **2018**, *10*, 3. [CrossRef]
41. 3rd Generation Partnership Project (3GPP). TR 36.843 V12.0.1, Study on LTE Device to Device Proximity Services. Available online: http://www.3gpp.org/ftp/Specs/archive/36_series/36.843/36843-c01.zip (accessed on 15 November 2018).
42. Hara, S.; Anzai, D.; Yabu, T.; Lee, K.; Derham, T.; Zemek, R. A Perturbation Analysis on the Performance of TOA and TDOA Localization in Mixed LOS/NLOS Environments. *IEEE Trans. Commun.* **2013**, *61*, 679–689. [CrossRef]
43. Singh, V.K.; Chawla, H.; Bohara, V.A. A Proof-of-Concept Device-to-Device Communication Testbed. *arXiv* **2016**, arXiv:1601.01398.
44. Ramirez, E. Privacy and the IoT: Navigating Policy Issues—International Consumer Electronics Show. Available online: https://www.ftc.gov/system/files/documents/public_statements/617191/150106cesspeech.pdf (accessed on 14 November 2018).
45. Shi, Y.; Jensen, M.A. Improved Radiometric Identification of Wireless Devices Using MIMO Transmission. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 1346–1354. [CrossRef]
46. Li, Q.; Fan, H.; Sun, W.; Li, J.; Chen, L.; Liu, Z. Fingerprints in the Air: Unique Identification of Wireless Devices Using RF RSS Fingerprints. *IEEE Sens. J.* **2017**, *17*, 3568–3579. [CrossRef]
47. Zhang, Z.; Guo, X.; Lin, Y. Trust Management Method of D2D Communication Based on RF Fingerprint Identification. *IEEE Access* **2018**, *6*, 66082–66087. [CrossRef]
48. IEEE Computer Society. *IEEE Standard 802.15.4a*; IEEE: New York, NY, USA, 2007.
49. Akiyama, T.; Sugimoto, M.; Hashizume, H. SyncSync: Time-of-arrival based localization method using light-synchronized acoustic waves for smartphones. In Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Banff, AB, Canada, 13–16 Octorber 2015; pp. 1–9.
50. Qi, Y.; Kobayashi, H.; Suda, H. Analysis of wireless geolocation in a non-line-of-sight environment. *IEEE Trans. Wirel. Commun.* **2006**, *5*, 672–681.
51. Riba, J.; Urruela, A. A non-line-of-sight mitigation technique based on ML-detection. In Proceedings of the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, QC, Canada, 17–21 May 2004.

52. Zhou, Z.; Yang, Z.; Wu, C.; Shangguan, L.; Cai, H.; Liu, Y.; Ni, L.M. WiFi-Based Indoor Line-of-Sight Identification. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 6125–6136. [CrossRef]
53. Haus, M.; Waqas, M.; Ding, A.Y.; Li, Y.; Tarkoma, S.; Ott, J. Security and Privacy in Device-to-Device (D2D) Communication: A Review. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1054–1079. [CrossRef]
54. Jin, B.; Jiang, D.; Xiong, J.; Chen, L.; Li, Q. D2D Data Privacy Protection Mechanism Based on Reliability and Homomorphic Encryption. *IEEE Access* **2018**, *6*, 51140–51150. [CrossRef]
55. Lohan, E.S.; Torres-Sospedra, J.; Richter, P.; Leppäkoski, H.; Huerta, J.; Cramariuc, A. Crowdsourced WiFi database and benchmark software for indoor positioning. *Analysis* **2017**. [CrossRef]
56. ISO. *Information Technology–Real Time Locating Systems–Test and Evaluation of Localization and Tracking Systems (ISO/IEC 18305:2016)*; International Organization for Standardization: Geneva, Switzerland, 2016; pp. 1–76.
57. Potortì, F.; Crivello, A.; Palumbo, F. Chapter 11: The EvAAL Evaluation Framework and the IPIN Competitions. In *Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation*; Conesa, J., Ed.; Academic Press: Cambridge, MA, USA, 2019; pp. 209–224.

PIII

POSITIONING ERROR PREDICTION AND TRAINING DATA EVALUATION IN RF FINGERPRINTING METHOD

by

S Khandker, R Mondal, T Ristaniemi 2019

10th International Conference on Indoor Positioning and Indoor Navigation (IPIN), Pisa, Italy

https://doi.org/10.1109/IPIN.2019.8911821

2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 30 Sept. - 3 Oct. 2019, Pisa, Italy

Positioning Error Prediction and Training Data Evaluation in RF Fingerprinting Method

Syed Khandker [1] Riaz Mondal [2] Tapani Ristaniemi [3]
Faculty of Information Technology, University of Jyvaskyla, Finland [1,3]
Magister Solutions Ltd. Jyvaskyla [2]
syed.i.khandker@student.jyu.fi [1] riaz.uddin.mondal@magister.fi [2] tapani.ristaniemi@jyu.fi [3]

***Abstract*—Radio Frequency (RF) fingerprinting-based localization has become a research interest due to its minimum hardware requirement and satisfiable positioning accuracy. However, despite the significant attention this topic has gained, most of the research focused on the calculation of position estimates. In this paper, we propose a simple and novel method that can be used as an indicator of fingerprinting positioning error. The method is based on cluster radius evaluation of multiple fingerprinting data during the test phase, which can be used by a Location Based Service (LBS) provider to predict the user position estimation accuracy. This method can be used effectively in real-time to predict the estimation error and thereby assists the LBS to offer a better quality of service. The cluster radius also reveals the quality of the recorded radio map.**

***Index Terms*—RF Fingerprinting, Indoor positioning, Crowdsourcing, KNN, Log-Gaussian probability, Weighted-Centroid.**

I. Introduction

The propagation of radio signals in an indoor environment is unpredictable due to the presence of people and movable obstacles which create reflections, refractions, and multi-path interference that impair the radio signal based precise positioning. Besides, there is no alternative system to verify the RF fingerprint (from now on, fingerprint) based positioning performance since GNSS does not perform well enough in the indoor area. However, there are many situations where precise location information is highly needed. So knowing the positioning accuracy is crucial for many modern days services and applications. End users could be informed about the estimated position error to avoid frustration in case the system gives wrong position information. Moreover, service providers could adopt a new strategy to achieve a higher service quality if the error can be predicted in advance.

For an efficient error prediction, knowing the possible sources of error is essential. Some studies reported that the coverage gap between the training and testing phase is the source of a significant amount of error [1], [2]. In the crowdsourcing data collection process, ordinary users collect location fingerprints by installing a particular application on their mobile devices. Generally, the contributors to a crowdsourced database, work independently in a freestyle manner. Therefore, it is highly likely that data collectors would cover some area multiple times or they may not visit some other areas at all, this may lead to creating an unbalance radio map of a geographical area. As a result, in the radio map, some areas would be covered by a large number of training data, whereas some other areas may contain less amount of fingerprints. A large amount of training fingerprints often increases the positioning accuracy, but it would occupy the more memory, increases the calculation complexity, rises user positioning time, and in some cases degrades the positioning accuracy [3]. So, an optimal amount of training data is desirable which will reduce the coverage gap as well as increase the positioning accuracy.

978-1-7281-1788-1/19/$31.00 © 2019 IEEE

In this paper, we investigated on two issues: i) how to predict positioning error ii) how to detect a unbalance radio map. Our study would help to predict the positioning error, finding out the coverage gap, decision of adding new training data on the radio map and choosing the right positioning method based on the quality of the training database. Fingerprint data from a five-storey university building of Tampere University of Technology (TUT) has been used in our study, that reflects almost every aspects real-world scenario. Four different positioning approaches have been applied to check the effectiveness of the study.

This paper is organized as follows: Section II describes the related works. Information regarding fingerprint database has been stated in Section III. Section IV contains the descriptions of different fingerprinting positioning approach. Section V includes the proposed method and experiment results followed by a conclusion in Section VI.

II. Related Work

Compare to position calculation and improvement studies, less attention has been paid on the error prediction strategy. In [4] authors showed that strong Received Signal Strength (RSS) in a fingerprint increase the probability of containing more accurate information compare to the weaker ones, they used strong signal strength feature as a positioning accuracy indicator. In [5] error was predicted by a set of the best estimates. The error indicator was calculated from the average distance between the position of the best estimate and all the other better estimates. The paper [6] proposes a mechanism to predict the positioning error by investigating the relationship between spatial distance and radio signal distance. In [7] the uncertainty of a fingerprinting localization was predicted

by relating the location error with the conditional entropy in the location posterior probability distribution. Gaussian distribution based indicators were studied in [8], [9]. Recently a positioning error prediction technique based on the device-to-device communication protocol has been proposed in [10]. Most of the previous works rely on signal strength information or other devices to predict the positioning error. However, our approach is self-dependent, it predicts the positioning error based on cluster radius evaluation of a few fingerprints. Unlike some previous works, it does not require any deep mathematical calculation or searching for strong RSS values to predict the error. Our proposed method also helps to evaluate the quality of the training data.

III. Database Description

Crowdsourced fingerprints collected by the research group of the TUT have been used in this paper. The database and the benchmarking software are distributed under the open-source MIT license and can be found on the Zenodo repository [11]. It is a general-purpose open-access repository developed under the European OpenAIRE program. It allows researchers to deposit data sets, research software, reports, and any other research related digital artifacts. The measurements were taken in a five-storey university building in Tampere, Finland. The building has a footprint of about 22,570m^2 (208 m x 108 m). Total 4648 fingerprints were collected which were then split uniformly randomly 15% for the training and 85% for the test purpose. A total of twenty-one android devices with an identical application were used by a group of student to collect the fingerprints. The contributors reported the location based on a manual input on the map to the server. The server stored the time stamp, the device model, the MAC addresses of the heard access points (AP), the RSS from each AP, and the location reported by the user. A local coordinate system, e.g., (x, y, z) = (123.45, 14.71, 0) was used instead of the global WGS84 commonly used in other applications. During the measurements, a total of 991 AP were heard, the exact position of the APs was not known. Fig. 1 shows the fingerprint database.

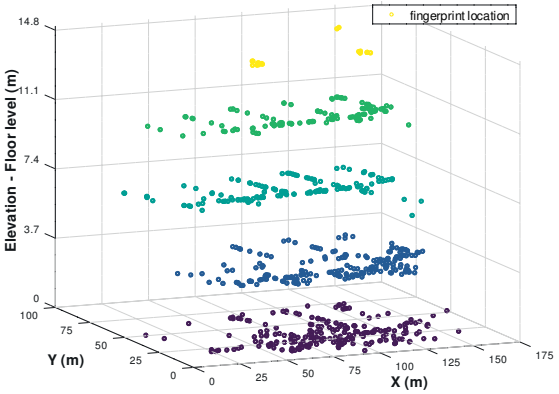


Fig. 1: Visual representation of the fingerprints location.

IV. Positioning Approaches

A. Log-Gaussian Probability (LGP)

Log-Gaussian probability works by computing a Gaussian likelihood function ι_i for all the training fingerprints with respect to the test fingerprint [12], [13]. If the commonly heard AP'(N_{ap}) RSS set for the test fingerprint is Q_k, and for the training fingerprint $P_{i,k}$. Then ι_i can be calculated

$$\iota_i = \sum_{k=1}^{N_{ap}} log\left(\frac{1}{\sqrt{2\pi\sigma_{ap}^2}} exp\left(-\frac{\left(Q_k - P_{i,k}\right)^2}{2\sigma_{ap}^2}\right)\right) \quad (1)$$

Here σ_{ap} is a noise variance that represents both shadowing and measurement error effects. The location of the training fingerprint with the highest ι_i is selected as the position of the test fingerprint. We have used σ_{ap} = 7 in our study to maintain consistency with other studies [14].

B. Weighted Centroid (WC)

The position of the test fingerprint in the weighted centroid approach is computed as the weighted average of the positions of APs recorded in the fingerprint [15], [16]. Denoting the set of all hearable access point by the device by AP_h. The weighted centroid-based estimate of test coordinates is computed as

$$P_i(x,y,z) = \frac{\sum_{j=1}^{n}\left(w_{ij} \times AP_{hj}(x,y,z)\right)}{\sum_{j=1}^{n} w_{ij}} \quad (2)$$

Where n is the size of the set AP_h, and w is the weight function. In our experiment we used $w = 10^{rss/10}$ to calculate the weight metric. The position of the APs are not given in the database, therefore at first, we had to estimate the position of the APs. According to the same concept of Eq. 2 instead of fingerprint's position, APs positions were calculated. The positions of the fingerprints from where an AP was heard, were used to estimate the position of that AP.

C. Universitat Jaume I (UJI)

The research group of Universitat Jaume I (UJI) uses the KNN classifier, but their data representation is different. Generally, RSS is expressed in negative dB unit, but some distance and similarity measures do not allow the use of negative value. They proposed to use three alternatives to represent the RSS in positive value [17]. At first, minimum possible RSS for an AP is obtained, if *i* represents the AP in the *x* fingerprint than

$$Positive_i(x) = RSS_i - min \quad (3)$$

$$Exponential_i(x) = \frac{exp(\frac{positive(x)}{\alpha})}{exp(\frac{-min}{\alpha})} \quad (4)$$

$$powed_i(x) = \frac{(positive_i(x))^{\beta}}{(-min)^{\beta}} \quad (5)$$

The denominator constant (α) was set to 24 and the exponent (β) was set to the mathematical constant e. In this paper we choose to use Eq. 3, with Sorensen distance measure. Non-heard AP's RSS was set to -103 dB. Sorensen distance between two RSS vectors (P,Q) can be calculated as:

$$distance_{sorensen}(P,Q) = \frac{\sum_{i=1}^{n} |P_i - Q_i|}{\sum_{i=1}^{n} (P_i + Q_i)} \quad (6)$$

n represent the length of the RSS vectors.

D. K-Nearest Neighbor (KNN)

KNN approach is also known as a distance-based classifier that classifies instances based on their similarity. In order to satisfy the acceptable localization accuracy with low computation effort, KNN has been used for fingerprinting positioning by many researchers [18], [19]. In this approach, signal distances between the test fingerprint and each training fingerprint are calculated. Than for K = 1, the lowest signal distance containing training fingerprint's location is given as the estimated location of test fingerprint. We have used Euclidean distance in this approach since it is the most commonly used distance measure used in the current literature. If P and Q are two RSS vectors having n length, then their signal distance is:

$$distance_{euclidean}(P,Q) = \sqrt[2]{\sum_{i=1}^{n} (P_i - Q_i)^2} \quad (7)$$

V. Method and Experiments

A. Method Explanation

In the fingerprinting method, there are two types of fingerprint data, training fingerprint, and test fingerprint. Training fingerprint contains RSS values from multiple APs, and location information where test fingerprints consist of only RSS values. The geographical location from where a fingerprint is recorded is called as the true or real position of that fingerprint. On the other hand, a test fingerprint contains only RSS values, the position which is calculated from the RSS values of a test fingerprint is known as the estimated position of that test fingerprint. The main idea behind our method is, if multiple spatially adjacent test fingerprints reveal nearly the same estimated positions such as all the estimation positions are close to each other, then it can be said that the estimated position is correct since the calculated position is cross-checked multiple times. If test fingerprints from a closely located area do not show estimated position similarity, then it can be assumed that there is some positioning error. A positioning error is calculated between the real and estimated position of that test fingerprint. Fig. 2 demonstrates the proposed concept. The black star marks represent the real positions of four test fingerprints forming a cluster. This cluster's radius is called real positions cluster radius (r_{real}), whereas the red star marks show their corresponding estimated positions in three different scenarios, forming three different clusters with radius r_{est1}, r_{est2}, and r_{est3} respectively. Here the radius of a cluster is calculated as the distance between its centroid and furthest position of the fingerprints.

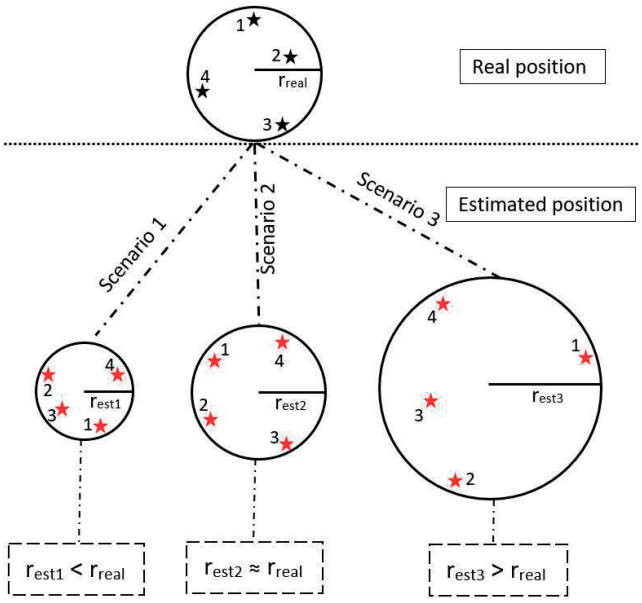


Fig. 2: Conceptual positions of four fingerprints.

As we can see from Fig. 2, in the first and second scenarios $r_{est1} < r_{real}$ and $r_{est2} \approx r_{real}$, if the value of r_{real} is small, the differences between the real and estimated positions will not be significantly high. The estimated positions may not be exactly equal to the real locations but close enough. In these two scenarios, to have a large scale positioning error, all of these fingerprints' estimated positions need to be nearly equally wrong at a time, that is very unlikely because different test fingerprints contain different RSS values. However, in scenario 3, $r_{est3} > r_{real}$ implies that estimated positions are quite far from the real locations, that is the most common scenario in fingerprinting positioning. If the estimated position cluster radius (r_{est}) is greater than the real position cluster radius then it can be inferred that the additional space has been occurred due to the presence of error and there is a relation between r_{real} and r_{est} in terms of positioning error (pe). It was found that compared to r_{real}, the larger the r_{est} the more pe is. They approximately show the following relation:

$$pe \propto \frac{r_{est}}{r_{real}} \quad (8)$$

To be confirmed that the test fingerprints are spatially close to each other and there is a less probability of having physical obstacle among them r_{real} should be low, e.g., 1~2 meters.

Obstacle inside the test cluster may put a significant effect on the radio condition of that area, which may lead to having massive RSS anomaly among the fingerprints; subsequently, test fingerprints of the cluster may show inconsistency in estimated position performance. On the other hand, if there are no changes in RSS values among the test fingerprints of a cluster, then all of them may reveal the same estimated location. Therefore, there should be little spatial distance among the test fingerprints within a test cluster to have low-level RSS variation. Several studies have shown that for separation of one-meter between two fingerprints creates on average 3~5 dB of RSS deviation [20], [21]. The average walking speed of a human is approximately $1.4m/s$. During the position estimation phase, for a moving device, if the system collects few test fingerprints from the user's device for a shorter period (e.g., 1~3 seconds) or a stationary user goes through a simple calibration process by walking few steps around to send a few test samples to the system, then most likely these test samples are spatially close to each other which leads to a small r_{real} value. We have observed that in most of the cases r_{est} larger than r_{real}, mainly because of the change of radio signal obstruction pattern between training and testing phase, and due to insignificant training data [2]. Since r_{real} will constantly be maintained a small value (1~2 meters), from different experimental results with varying training dataset, it was found that the mean positioning error of multiple test fingerprints bears a relation with the cluster radius formed by their estimated positions. This relation can be approximately stated as follows:

$$pe \propto r_{est} \tag{9}$$

From the Eq. 9, it can be seen that the estimated position cluster radius indicates the positioning error of test fingerprints even without the real positions.

B. Experiment Result

From the test database, 2980 test fingerprints out of 3951 were chosen to evaluate the experiment, since all of them are not suitable for the experiments carried out in this work, e.g., some of them had not enough neighbors within short range. Training database remained unaltered, containing 697 fingerprints (15% of the whole database) as in Zenodo repository [11]. We selected four range groups (in meter), i.e., 0 to <5, 5 to <10, 10 to <15, and ≥ 15 for r_{est} to express a statistically meaningful relation of r_{est} with pe. The value of r_{real} was set to $2m$. Each test cluster was comprised of 3 to 5 test fingerprints (TFP). Fig. 3 shows that when r_{est} increases the mean 3D positioning error also increases through all the four positioning approaches. Clusters of 1237 test fingerprints resulted within 0 to <5 meters range, through KNN and UJI approach their mean pe was slightly above 5m, but LGP and WC showed worse performance with around 9m accuracy. pe was escalated by 978 test fingerprints whose r_{est} were 5 to <10 meters range. The inclined trend of pe continues with an increase of r_{est} till the end, regardless of the applied

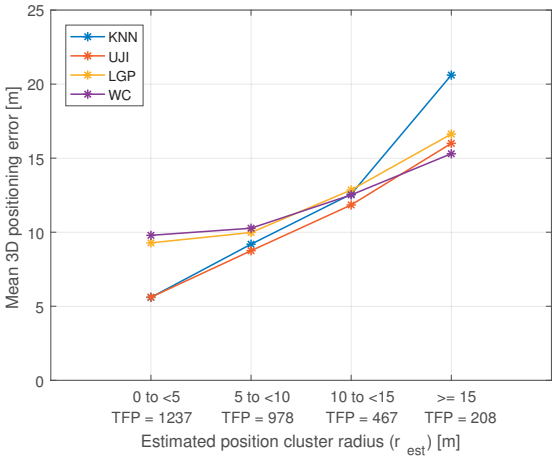
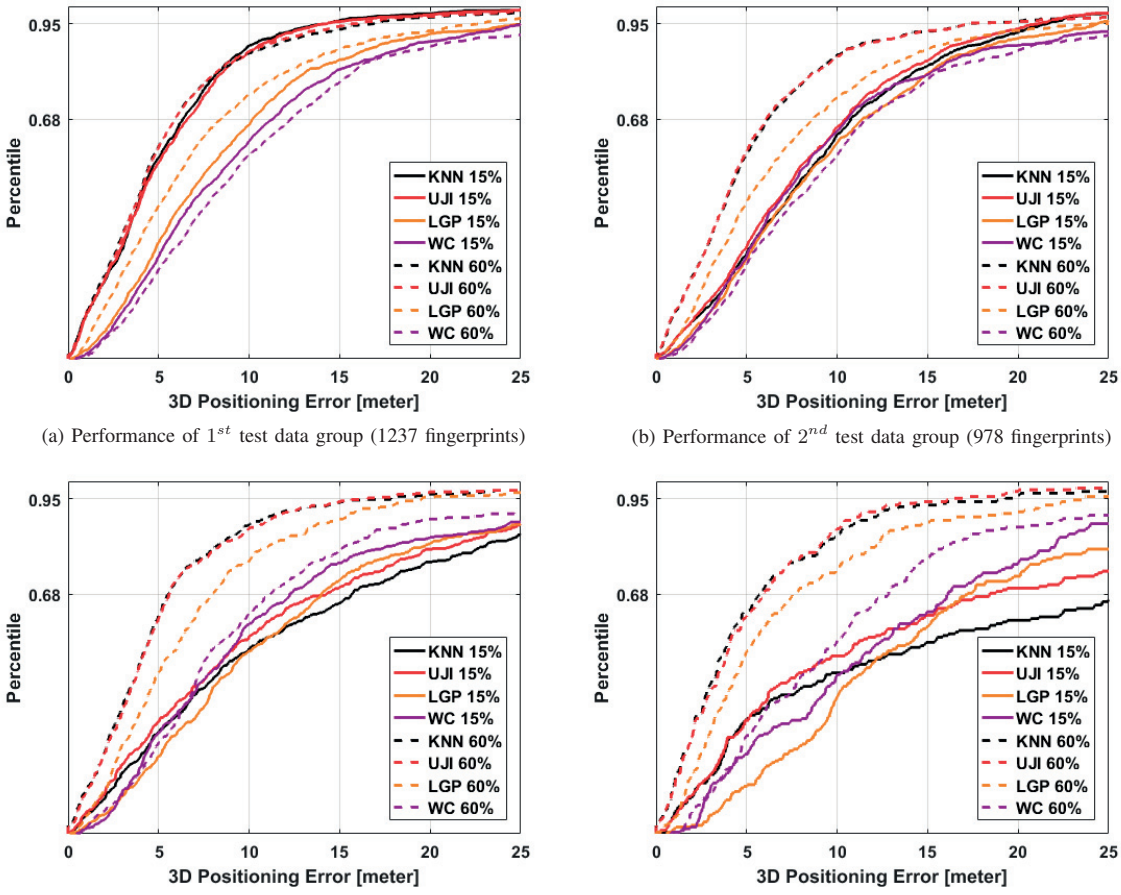


Fig. 3: Relation between positioning error (pe) and estimated position cluster radius (r_{est}).

positioning approach. Worst positioning performance was shown by the test fingerprints having the r_{est} more than or equal $15m$. So we can see when r_{est} increases pe also increases. By observing the estimated position cluster radius, the positioning error can be predicted.

The experiment showed how to predict the positioning error. Many factors may have contributed on pe, e.g., obstruction, coverage gap, erroneous information in the fingerprint, and device diversity effect, etc. The r_{real} value was only 2m. Therefore there is less probability of having obstacles inside the cluster. Device diversity effect in crowdsourced fingerprint database does not affect in a large scale [22]. A dedicated application was used to collect the fingerprints, and the coordinates were selected manually. According to the TUT research group, there could be an average 0.5m error in location coordinate in the collected data [11]. So apparently a significant portion of the error occurred due to the coverage gap. That means that during the data collection process, contributors did not cover all the area equally. That might lead to an unbalanced radio map, subsequently an unstable positioning performance. Moreover, one can also argue that the amount of training data was inadequate (only 15% of the whole database), a high densely training database may show a different scenario. In the following experiment, the amount of training data has been increased. Our focus is to find out the optimal amount of training data for better positioning accuracy. More specifically, the relation between r_{est} and training data. Based on this relation, the system would be able to decide whether to accept or ignore any reported fingerprint in an automatic crowdsourcing process.

The unused 971 test fingerprints (3951-2980) from the first experiment have been shifted to the training database



(a) Performance of 1^{st} test data group (1237 fingerprints)

(b) Performance of 2^{nd} test data group (978 fingerprints)

(c) Performance of 3^{rd} test data group (467 fingerprints)

(d) Performance of 4^{th} test data group (208 fingerprints)

Fig. 4: Comparison of the positioning performance for high and low amount of training data.

to increase the amount of training data. In the previous experiment, there were four groups of test data according to r_{est} range, containing 1237, 978, 467, and 208 test fingerprints. For simplicity, we call them 1^{st}, 2^{nd}, 3^{rd}, and 4^{th} test data group, respectively. One test data group does not affect another. Therefore, at the time of one test data group's positioning estimation, other test data groups can also be a part of the training data. Thus the amount of training data were raised to 60% of the whole database.

Solid lines in Fig. 4(a-d) express Cumulative Distribution Function (CDF) of pe for the first experiment (less amount of training data, 15% of the whole database), while dashed lines show the CDF of pe for the larger amount of training data (60% of the whole database). In Fig. 4(a-d) solid lines show that with an increase of r_{est} range (1^{st} group to 4^{th} group) positioning performance degrades sharply regardless of the applied positioning approach. A larger amount of training data helps to improve performance (dashed line). Each positioning approach in Fig. 4(d) experienced dramatic improvement with more training data, Fig. 4(c) and Fig. 4(b) also reveals the same trend. However, in Fig. 4(a), WC's performance degraded, no significant changes for KNN and UJI. LGP improved a bit, but previously it had very week performance. The explanation for the worthless improvement with high densely training data in Fig. 4(a) could be, the radio map was already saturated (lower r_{est} indicates that phenomena). More training data did not bring any significant improvement. Based on this experiment result, it can be said that r_{est} also indicates the quality of training data.

Since the increase of training data affects the positioning performance, we further checked how it does affect r_{est}. From Fig. 5 it can be seen that, mean r_{est} declined when the amount of training data is increased in all the range group except first

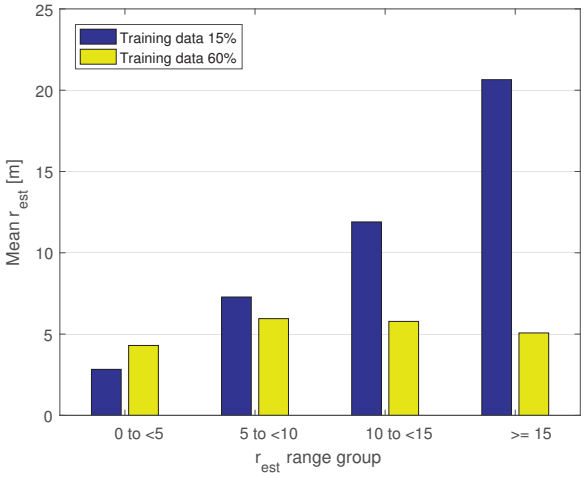


Fig. 5: Comparison of average r_{est} value.

one. The first group had already a decent amount of effective training data. Therefore, the addition of more training point did not improve the r_{est}; subsequently, the positioning performance also did not progress (see Fig. 4(a)). However, in the rest other range groups previously with higher r_{est} value hints that quality of the training data is not satisfactory. When more training data was added for those group, the r_{est} decreased. As a result, positioning accuracy improved significantly (see Fig. 4(b-d)). Exact numerical relation between r_{est} and the optimal amount of training data varies depending on the test bed, but the experimental result proves that they are correlated. An adequate amount of training data would set a low level of r_{est} value (e.g., 3~5 meters). During the design phase based on r_{est} value, a system could be able to set an optimal amount of training data to achieve the best performance.

VI. Conclusion

In this paper, the positioning error prediction, and training database evaluation technique in RF fingerprinting method have been discussed. It has been found that few spatially adjacent fingerprints' estimated position cluster radius indicates the magnitude of the error. The radius also demonstrates the quality of the recorded radio map. Use of real fingerprint data from a large three-dimensional area and four different positioning approaches in the experiments have validated the effectiveness of the proposed techniques. Findings of this study are very simple to implement and would help to make many kinds of effective decisions to improve the quality of fingerprinting positioning. The main limitation of our proposed method is, the user's position needs to be slightly changed while the positioning request is being made. If the user remains stationary, then a positioning request for an identical fingerprint will be made several times, that limit the effectiveness of the proposed method. In our future studies, we shall work to eliminate this limitation.

Acknowledgment

The authors express their warm thanks to the Riitta and Jorma J. Takanen foundation for its financial support.

References

[1] J. Talvitie, E. S. Lohan, and M. Renfors, "The effect of coverage gaps and measurement inaccuracies in fingerprinting based indoor localization," in *International Conference on Localization and GNSS 2014 (ICL-GNSS 2014)*, June 2014, pp. 1–6.

[2] J. Torres-Sospedra and A. Moreira, "Analysis of sources of large positioning errors in deterministic fingerprinting," *Sensors*, vol. 17, no. 12, 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/12/2736

[3] M. Muñoz-Organero and C. Brito-Pacheco, "Improving accuracy and simplifying training in fingerprinting-based indoor location algorithms at room level," *Mobile Information Systems*, vol. 2016, pp. 1–16, 2016. [Online]. Available: https://doi.org/10.1155/2016/2682869

[4] Y. Li, Z. He, Z. Gao, Y. Zhuang, C. Shi, and N. El-Sheimy, "Towards robust crowdsourcing-based localization: A fingerprinting accuracy indicator enhanced wireless/magnetic/inertial integration approach," *IEEE Internet of Things Journal*, pp. 1–1, 2019.

[5] H. Lemelson, M. B. Kjærgaard, R. Hansen, and T. King, "Error estimation for indoor 802.11 location fingerprinting," in *Location and Context Awareness*, T. Choudhury, A. Quigley, T. Strang, and K. Suginuma, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 138–155.

[6] V. Moghtadaiee, A. G. Dempster, and B. Li, "Accuracy indicator for fingerprinting localization systems," in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, April 2012, pp. 1204–1208.

[7] R. Berkvens, H. Peremans, and M. Weyn, "Conditional entropy and location error in indoor localization using probabilistic wi-fi fingerprinting," *Sensors*, vol. 16, no. 10, 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/10/1636

[8] C. Beder, A. McGibney, and M. Klepal, "Predicting the expected accuracy for fingerprinting based wifi localisation systems," in *2011 International Conference on Indoor Positioning and Indoor Navigation*, Sep. 2011, pp. 1–6.

[9] P. Marcus, M. Kessel, and M. Werner, "Dynamic Nearest Neighbors and Online Error Estimation for SMARTPOS," *International Journal On Advances in Internet Technology*, vol. 6, no. 1 and 2, 2013.

[10] S. Khandker, J. Torres-Sospedra, and T. Ristaniemi, "Improving rf fingerprinting methods by means of d2d communication protocol," *Electronics*, vol. 8, no. 1, 2019. [Online]. Available: http://www.mdpi.com/2079-9292/8/1/97

[11] E. S. Lohan, J. Torres-Sospedra, P. Richter, H. Leppkoski, J. Huerta, and A. Cramariuc, "Crowdsourced WiFi database and benchmark software for indoor positioning," sep 2017. [Online]. Available: https://doi.org/10.5281/zenodo.889798

[12] V. Honkavirta, T. Perala, S. Ali-Loytty, and R. Piche, "A comparative survey of wlan location fingerprinting methods," in *2009 6th Workshop on Positioning, Navigation and Communication*, March 2009, pp. 243–251.

[13] E. Laitinen, J. Talvitie, and E. Lohan, "On the rss biases in wlan-based indoor positioning," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, June 2015, pp. 797–802.

[14] E. S. Lohan, J. Torres-Sospedra, H. Leppakoski, P. Richter, Z. Peng, and J. Huerta, "Wi-fi crowdsourced fingerprinting dataset for indoor positioning," *Data*, vol. 2, no. 4, 2017. [Online]. Available: http://www.mdpi.com/2306-5729/2/4/32

[15] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, "Weighted centroid localization in zigbee-based sensor networks," in *2007 IEEE International Symposium on Intelligent Signal Processing*, Oct 2007, pp. 1–6.

[16] A. Razavi, M. Valkama, and E. Lohan, "K-means fingerprint clustering for low-complexity floor estimation in indoor mobile localization," in *2015 IEEE Globecom Workshops (GC Wkshps)*, Dec 2015, pp. 1–7.

[17] J. Torres-Sospedra, R. Montoliu, S. Trilles, Ó. Belmonte, and J. Huerta, "Comprehensive analysis of distance and similarity measures for wi-fi fingerprinting indoor positioning systems," *Expert Syst. Appl.*, vol. 42, pp. 9263–9278, 2015.

[18] Y. Xie, Y. Wang, A. Nallanathan, and L. Wang, "An improved k-nearest-neighbor indoor localization method based on spearman distance," *IEEE Signal Processing Letters*, vol. 23, no. 3, pp. 351–355, March 2016.

[19] J. Bi, Y. Wang, X. Li, H. Qi, H. Cao, and S. Xu, "An adaptive weighted knn positioning method based on omnidirectional fingerprint database and twice affinity propagation clustering," *Sensors*, vol. 18, no. 8, 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/8/2502

[20] M. Ivani and I. Mezei, "Distance estimation based on rssi improvements of orientation aware nodes," in *2018 Zooming Innovation in Consumer Technologies Conference (ZINC)*, May 2018, pp. 140–143.

[21] R. K. Mahapatra and N. S. V. Shet, "Experimental analysis of rssi-based distance estimation for wireless sensor networks," in *2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, Aug 2016, pp. 211–215.

[22] Y. Ye, B. Wang, X. Deng, and L. T. Yang, "On solving device diversity problem via fingerprint calibration and transformation for rss-based indoor localization system," in *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug 2017, pp. 1–8.

PIV

ANALYSIS OF RECEIVED SIGNAL STRENGTH QUANTIZATION IN FINGERPRINTING LOCALIZATION

by

S Khandker, J Torres-Sospedra, T Ristaniemi 2020

Sensors, 20 (11), 3203

https://doi.org/10.3390/s20113203

sensors

MDPI

Article

Analysis of Received Signal Strength Quantization in Fingerprinting Localization

Syed Khandker [1,*], Joaquín Torres-Sospedra [2,3] and Tapani Ristaniemi [1]

1 Faculty of Information Technology, University of Jyväskylä, Mattilanniemi 2, 40014 Jyväskylä, Finland; tapani.ristaniemi@jyu.fi
2 Institute of New Imaging Technologies, Universitat Jaume I, Av. Vicente Sos Baynat s/n, 12071 Castellón de la Plana, Spain; jtorres@uji.es or torres@ubikgs.com
3 UBIK Geospatial Solutions, Av. Vicente Sos Baynat s/n, 12071 Castellón de la Plana, Spain
* Correspondence: syed.i.khandker@student.jyu.fi

Received: 17 May 2020; Accepted: 2 June 2020; Published: 4 June 2020

Abstract: In recent times, Received Signal Strength (RSS)-based Wi-Fi fingerprinting localization has become one of the most promising techniques for indoor localization. The primary aim of RSS is to check the quality of the signal to determine the coverage and the quality of service. Therefore, fine-resolution RSS is needed, which is generally expressed by 1-dBm granularity. However, we found that, for fingerprinting localization, fine-granular RSS is unnecessary. A coarse-granular RSS can yield the same positioning accuracy. In this paper, we propose quantization for only the effective portion of the signal strength for fingerprinting localization. We found that, if a quantized RSS fingerprint can carry the major characteristics of a radio environment, it is sufficient for localization. Five publicly open fingerprinting databases with four different quantization strategies were used to evaluate the study. The proposed method can help to simplify the hardware configuration, enhance security, and save approximately 40–60% storage space and data traffic.

Keywords: fingerprinting; quantization; indoor positioning

1. Introduction and Motivation

Positioning is an important part of our daily life. Essential services, e.g., navigation, health care, personnel management, and emergency rescue, require localization information to work effectively. The predominant localization technology (e.g., GPS and Galileo) using satellite signals has solved most of the outdoor positioning-related problems. However, the satellite-based positioning in indoor is severely degraded due to the blockage of the satellite signals by the obstacles. Therefore, alternative indoor localization technologies have been recently developed using optic [1], ultra-sound [2], dead reckoning [3], ultra-wideband [4], RFID [5], visible Light [6], and Bluetooth technology [7]. Most of them are based on communications technologies and require the deployment of additional hardware components in the environment to work effectively. Dead reckoning is only able to provide relative positioning (displacement from the origin) and needs a secondary technology to support absolute positioning and reduce the error drift. However, Wi-Fi fingerprinting localization has become a very promising and competitive technical solution for indoor positioning systems for its cost-effective high precision performance [8] for either smartphone [9,10] or autonomous vehicle [11,12] applications.

Wi-Fi fingerprinting localization is a technique that uses RSS measurement to perform localization tasks. RSS is easy to obtain with the current Wi-Fi interfaces without requiring any additional hardware. Moreover, unlike Channel State Information (CSI)-based techniques [13], it is not restricted to a particular LAN card and operating system. Therefore, RSS-based positioning has attracted considerable attention from researchers [14–16]. RSS is a measurement of the strength of a radio signal

that performed at the receiver end. In a wireless network, RSS from different Access Points (AP) creates a unique pattern for a geographical area, which is called Wi-Fi fingerprint. It is assumed that a fingerprint from a recording point remains unique for that location. Therefore, fingerprints can be used to retrieve the location. Generally, the RSS values are expressed in dBm unit with 1-dBm granularity. The main aim of RSS is to check the signal quality for evaluating the coverage and the quality of service; therefore, fine-granular RSS is needed. However, in fingerprinting positioning along with RSS values, reference AP labels are also recorded. Since a fingerprint has many reference AP, instead of fine-granular RSS, coarse-granular RSS could be able to provide the same positioning accuracy. Moreover, storing, exchanging, and processing raw RSS value can occupy a considerable amount of memory [17,18], as well as be a potential threat to user's location privacy [19,20]. In this case, a quantization method can be useful.

Quantization is the process of mapping a set of values to a particular value. It is inherent, for instance, in analog-to-digital converters and it has been widely studied in different positioning technologies [18,21–25]. In Wi-Fi fingerprinting, at the receiver end, a sensor captures the energy of a signal transmitted by the AP. This signal strength is a continuous number and fluctuates over time. For making sense out of it and for different kinds of signal processing, e.g., quality checking, ranging, and positioning, the signal power needs to be divided and expressed by a set of finite numbers, which is called quantization. The output (l) of a quantization depends on the used bit number. If n bits are used, the total output would be

$$l = 2^n \tag{1}$$

Figure 1 shows an example of linear quantization.

P_{max} and P_{min} are the maximum and minimum power of a signal. In Figure 1, we can see each quantized level shares a range of input power. Typically, the strength of a Wi-Fi signal is in between 1 and 10^{-10} mW. Through the quantization process, a large range of P_{max} and P_{min} can be scale down to a smaller set of numbers, where the granularity of output level depends on the used bit number. Figure 1 shows that more bits lead to finer output levels, while fewer bits lead to coarser output levels. The IEEE 802.11-2012 standard recommends to quantized the received signal power in the range of 0 to $\leq$ 8 bits, where the output levels are called Received Signal Strength Indicator (RSSI) [26]. However, quantization is implementation-dependent and done in a proprietary manner by different manufacturers. P_{max}, P_{min}, and applied bit number vary from manufacturer to manufacturer, which may contribute to the device heterogeneity problem, along with some other heterogeneity contributors such as the design of the antenna, and antenna orientation. Moreover, some manufacturers express the quantized value of receives power (e.g., 0,1,2,. . . ,N) where some other provides RSS directly in dBm unit (e.g., -30 or -40) [27]. This makes a great confusion between RSS and RSSI. However, solving this confusion is out of the aim of this paper. More information regarding the relationship between RSS and RSSI can be found in [26–28].

Generally, in dBm unit, RSS values are in the range of 0 to -100 dBm, and expressed by 1-dBm granularity [29]. To express these 101 levels, at least 7 bits are needed. Some studies showed that, instead of this fine-granular RSS, coarse-granular or fewer-bit quantized RSS could provide the same positioning accuracy [17,18,30]. However, most of the simulation-based previous studies investigated the quantization effect on Wireless Sensor Networks (WSN) or Distributed Sensor Networks (DSN). RSS quantization in the Wi-Fi network focusing the fingerprinting positioning purpose has not been studied enough. Why coarse-granular RSS or reduced-bit quantization can provide the same amount of positioning accuracy is not investigated yet. In the real scenario, vast numbers of AP, different mobile devices, and obstacles can have a great impact on quantization. Lack of thorough investigation of RSS quantization on real data motivated us to conduct this research. The main contributions of this study are followings:

- We thoroughly investigated the usefulness of reduction bit quantization considering why it can perform the same positioning accuracy as that of traditional RSS.

- Reduction bit quantization also reduces the data size. We investigated how much disk space and network traffic can be saved if reduced-bit quantization were used.
- We analyzed the RSS quantization effect on five different publicly open fingerprinting databases. Our findings are coherent and reflect all the real-life scenarios.

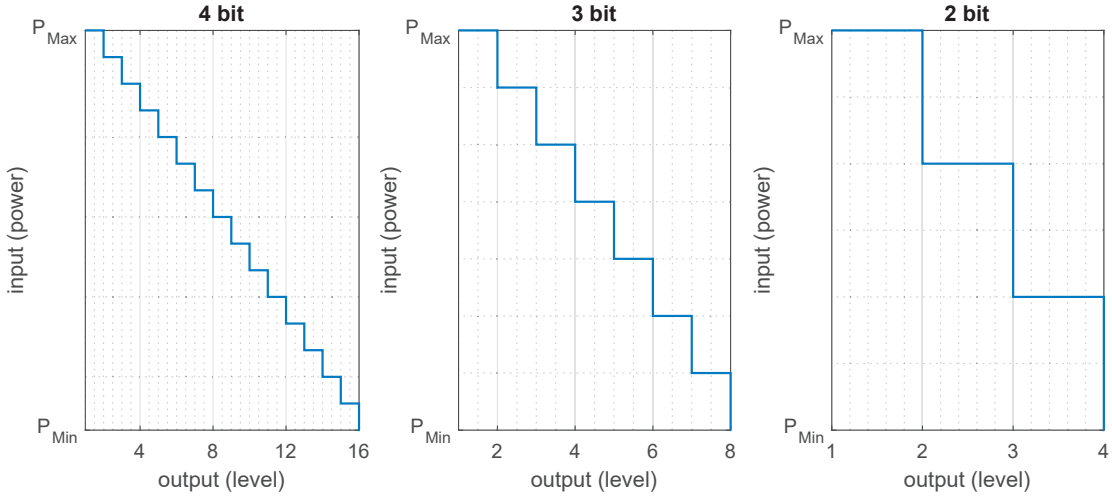


Figure 1. RSS quantization concept.

The remaining of this paper is organized as follows. Section 2 introduces a review of the works related the quantization. Section 3 describes the fingerprinting databases and explains the method. Section 4 is devoted to the different kinds of experiments and results. In Section 5, we discuss the findings and observations. Section 6 presents the conclusions and draws the research lines for further work.

2. Related Work

Since the primary purpose of RSS is not localization, only a few works have been done on the technical aspect of RSS (i.e., quantization) focusing on the localization, especially in the Wi-Fi network. This section reviews the existing works that address different aspects of signal strength quantization.

Bahl and Padmanabhan introduced RADAR, the first indoor positioning system based on *k*-NN for Wi-Fi fingerprinting in 2000 [31]. Although some *k*-NN variants [32–35] and other advanced methods, e.g., probabilistic models [36,37] or SVM-based [38], have also been applied with fingerprinting, we focus on the simple *k*-NN model as it is still widely used nowadays [39–43].

Jarvinen et al. [44] proposed RSS quantization to secure two-party computation (STPC) to preserve the privacy in fingerprint-based positioning. Their proposal decreases the overhead of STPC evaluation. They implemented an approximately 500 times faster privacy-preserving indoor localization protocol using quantized RSS. They observed that a 50 % reduction on quantized bits has around 5% impact on the accuracy of localization.

Nguyen et al. [45] observed that, in a WSN at the outdoor environment, the quantized scheme performed more stably and had smaller errors compared to the indoor space; this is because the wireless signals in the indoor environment suffered from multi-path and reflection effects from walls and obstacles. Both RSS and angle of arrival measurement were used in their study. They mentioned that, after a certain number of quantization levels, positioning accuracy does not improve.

Ababneh [46] mentioned that in a distributed sensor network (DNS) the localization error becomes highly dependent on the total number of bits and the pattern of the distribution of the sensors, especially when the number of bits allowed is relatively small. He proposed two novel low-complexity

bit allocation algorithms to solve optimal computational complexity. The simulation results show a significant reduction in computational complexity.

Richter et al. [17] studied the effects of the RSS quantization, focusing privacy-preserving. According to them, quantized RSS is beneficial to create ciphertext for encrypting purpose. Therefore, quantization can be an excellent tool for privacy preservation. Their non-uniform 4-bit quantization yields the same positioning accuracy as that of the unquantized system.

Nui et al. [47] performed a comparative study on location estimator in WSNs, based on quantized data, and compared it with localization approaches using analog data. They suggested that, to reduce communication costs, and sensor power consumption, local sensors should send quantized measurements to the fusion center. The simulation experiment showed that the positioning precision loss due to quantization reduces as the number of bits increases, which is almost negligible when 6 bits are used.

Li et al. [48] proposed quantized received signal strength-based target localization method to save energy and communication bandwidth in WSN. Their proposed algorithm, firstly, combines the particle swarm optimization (PSO) with their proposed formula to generate a fixed set of thresholds and then uses those to quantize the measured RSS. Based on the obtained quantized RSS data, PSO and optimized algorithms are used for accurate and efficient positioning.

Gao et al. [30] showed that, in fingerprinting positioning, the full resolution RSSI measurements are unnecessary. According to them, raw RSSI are noisy, exhibit a high degree of variability, and occupy a massive amount of memory. In their study, through a non-uniform quantization process, it was possible to reduce the RSSI data volume by approximately 72% without compromising the localization accuracy, while the positioning accuracy was around 2 m.

Torres-Sospedra and Moreira [49] analyzed the sources of very large positioning errors. They identified that the representation of the received signal strength as an integer value, which can be seen as a first quantization added to all RSSI measurements, added some uncertainty to the reference fingerprints, especially if the distance between the AP and the receiver is moderate or high. Depending on the AP distribution, which in most scenarios were deployed for communication purposes, and the noise level in the RSSI measurements, there can be some large areas in the operational environment where the collected fingerprints are similar. Thus, the fingerprint methods cannot properly operate or provide accurate position estimations on them.

Mizmizi and Reggiani [18] showed that the computational complexity could be limited by adapting the RSSI quantization. Their simulation of WSN contained many beacons over a limited squared area on a single floor with a two-dimensional coordinate system. They observed that there is an optimal quantization level number, over which positioning accuracy cannot be improved without increasing the number of beacons. However, if the quality of the RSSI measurements increases, then it is possible to reduce the number of quantization levels at the same performance. The exact meaning of the "quality of the RSSI" was not mentioned.

Shi et al. [50] studied the relationship among quantization level, network configuration parameters, and the lower bound of the positioning error based on the quantized RSSI in WSN. Their simulation suggests that localization error variance cannot be improved by only increasing the quantization level after a particular value; this is due to the noise in the RSSI readings. When the quantized RSSI value interval is comparable to the noise, increasing the quantization level becomes less useful.

Krishnamachari [51] showed a positioning performance analysis by using Cramér–Rao bound (CRB). He mentioned that 3 bits of RSS quantization in WSN suffices to give a lower bound that is very close to best possible.

Patwari and Hero [52] used the Cramer–Rao Bound (CRB) to compare the minimal attainable variances of unbiased sensor location estimators in WSN. They considered that RSS measurements are always going to be quantized while an analog-to-digital converter converts it. If there are many levels,

then the effect of the quantization is minimal. According to them, 3 bits of quantization are sufficient for an RSS-based positioning system.

Most of the previous works were done in WSN or DSN. According to our best knowledge, only two studies were done on RSS quantization in the Wi-Fi network focusing on user's privacy preservation [17,44]. In general, the RSS quantization effect in the Wi-Fi network has not been studied enough. Moreover, simulation results of the previous studies may not reflect all the aspects of a real-world scenario. This study focused on a step-by-step investigation of the quantization process on five different publicly open databases by four different quantization strategies. Furthermore, we considered applying quantization on the absolute signal energy level, which is a novel contribution to the Wi-Fi fingerprinting localization. Finally, the quantization of RSS measurements in fingerprinting has usually been applied to reduce the radio map size and optimize the computation of fingerprint-based methods. However, quantization only partially reduces the complexity of fingerprinting as the number of APs and reference fingerprints are not altered. Other relevant works that have focused on reducing the radio map include a focus on removing useless APs from the radio map [53], applying unsupervised learning to cluster/split the radio map [33,38,54,55], using radio-signal propagation knowledge to filter fingerprints out the radio-map [34,56–58], combining clustering with RSSI-based rules [59], and, even, reducing on-the-fly the reference samples and APs of the radio map [60]. In this study, we focused on the nominal case where no other optimizations have been performed. This ensures that the study reflects the benefits of quantization without the interference of other approaches, such as advanced fingerprint models or optimization rules.

3. Materials and Methods

In this section, we describe fingerprinting databases and the RSS quantization process.

3.1. Database Description

Five publicly open fingerprinting databases were used in this research. The first one is from the University of Jaume I, Spain. The author presented their work at the 2014 International Conference on Indoor Positioning and Indoor Navigation conference; the name of this database is UJIIndoorLoc [61]. The second one is from the Tampere University of Technology (TUT) Finland; we named it as TUT database [62]. The third database was collected by the research group of the University of Minho, Portugal, and is called Minho database [63]. The name of the next database is Mannheim. It was collected by the research group from the University of Mannheim, Germany [64]. The final database was collected from a library building of the University of Jaume I, which is different from the first database; we call it the Library database [65]. Short descriptions and comparison of these databases are discussed below.

3.1.1. UJIIndoorLoc Database

This multi-building, multi-floor localization database is the first publicly available Wi-Fi fingerprinting database created by the researchers from the University of Jaume I, Spain. Twenty users and 25 devices were deployed to collect the fingerprinting data from 3 buildings with 4 or 5 floors depending on the building. Two Android applications were used to create the database. In total, this database contains 21,049 samples. Apart from the RSS and location information, features such as BuildingID, FloorID, UserID, and timestamp are given for each sample. The location points were uniformly distributed to the users with the restriction that any reference point should be covered by, at least, two users. The longitude and latitude coordinates are given in meters with UTM from WGS84.

3.1.2. TUT Database

Fingerprints from the TUT university building were collected by a research group from that university from January 2017 to August 2017. The total area of that building is approximately 22,570 m^2

containing five floors. In total, 991 AP were recorded during the collection period. Out of 4648 collected fingerprints, 15% were used for the training and 85% were used for the testing. All fingerprints contained local coordinate values, e.g., (x, y, z) = (41.45, 14.71, 3.7), not the GPS one. Twenty-one Android devices having an identical application were used during the crowd-sourced data collection process. According to the TUT research group, approximately 10 m of mean 3D positioning accuracy was achieved through different positioning methods.

3.1.3. Minho Database

The research group from the University of Minho, Portugal collected the third database of this study from a university building that resembles an industrial floor plant, with a total area of around 1000 m^2. The data were collected in July 2017. The data collection setup was based on a Raspberry Pi 3 Model B with its internal Wi-Fi interface, and four additional USB Wi-Fi interfaces (Edimax EW-7811un). The database contains five sets of files, one for each one of the Wi-Fi interfaces. Since there is no significant impact of change in the Wi-Fi interface in our study, we used the data from the fifth interface in our work. In total, 5783 fingerprints were collected, among which 4973 were labeled as training fingerprints, and the rest were used as test fingerprints. The RSS from 11 different APs were recorded in each fingerprint.

3.1.4. Mannheim Database

A research group from the University of Mannheim, Germany, collected this database. The primary purpose of this database is to create a digital compass. Therefore, apart from the RSS and location, angular information was recorded by a USB powered digital compass. Samples were collected from the hallway of an office building. For each training and testing point, samples were collected 20 and 100 times, respectively. However, to speed up the experiment, we randomly selected ten samples per testing points. We had 14,300 training samples and 5060 testing samples. Local X,Y coordinate was used to define the location of the fingerprints.

3.1.5. Library Database

The UJI research group has been collecting the Wi-Fi measurement from their university library premises for the last 25 months, which has 308 m^2 of footprint. The fingerprints were recorded from the bookshelves area of the third and fifth floors of the library. These two floors are adjoining floors despite their numbering. Wi-Fi signal could travel from one story to another through stairs, but no line-of-sight path is available. That data have been recording for the last 25 months for the same place. We used the latest data (25th month) in this study. The database contains 576 training fingerprints and 3120 testing fingerprints. As with the TUT database, they also used a local coordinate system. In total, 620 AP were heard during the collection time. A Samsung Galaxy S3 smartphone was used to collect the fingerprint data with dedicated software. The authors reported the mean 2D positioning accuracy of 2.34 m.

3.2. Facts in the Databases

Facts regarding fingerprinting positioning in all theses database are summarized in Table 1.

Positioning performance based on traditional RSS was calculated using the *k*-NN algorithm and Euclidean distance. To keep similarity with previous works, $k = 1$ was used in UJIIndoorLoc, TUT, and Mannheim; and $k = 3$ was used in Minho and Library. Positioning accuracy for the TUT database was based on 3D positioning; for the other databases, it was 2D positioning. Minho and Mannheim have no floor information; therefore, floor detection is Not Applicable (NA) in these two.

Table 1. Facts in the databases.

Database	Coverage (m^2)	Number of Training Sample	Number of Testing Sample	Number of AP	Positioning (m) Accuracy	Floor Detection (%)
UJIIndoorLoc	108,703	19,936	1111	520	7.74	90.28
TUT	22,570	697	3951	991	9.39	91.75
Minho	1000	4973	810	11	4.7	NA
Mannheim	312	14,300	5060	28	3.01	NA
Library	308	576	3120	620	2.34	100

3.3. Method

We briefly state the quantization process in the Introduction. In this subsection, the proposed quantization method is explained in detail.

3.3.1. Quantization

The Wi-Fi signal is an electromagnetic wave whose intensity is attenuated as it propagates through space. When that wave reaches the antenna of a sensor, the sensor captures the energy from the signal. This energy expresses the strength of that signal. Despite the signal strength being continuous, it needs to be expressed to a countable finite number of elements to make that sensible and valuable for a different kind of signal processing, e.g., checking the quality of the signal, ranging, and positioning. To express the continuous value of the received signal to finite numbers, it generally goes through a process called quantization. If P_i is the absolute received power (in mW), using an S-level quantization, we obtain

$$Q(P_i) = \begin{cases} 0, & L_0 \leq P_i < L_1 \\ 1, & L_1 \leq P_i < L_2 \\ . & . \\ . & . \\ . & . \\ S-1, & L_{S-1} \leq P_i < L_S \end{cases} \tag{2}$$

Here, $L_0, \ldots, L_S$ defines the range of the quantization levels, where $L_0 = \infty$ mW and $L_s = -\infty$ mW. The quantized power is $Q(P_i) = 0, 1, \ldots\ldots, S-1$. If we look at the distribution of RSS in all the databases in Figure 2, we can see that the spreading of RSS is between -30 and -100 dBm. However, there could be some negligible amount of RSS reading beyond this range. It is common practice to use very weak signal value, e.g., -103 dBm for the non-heard AP in the fingerprints, to maintain the same length for all the fingerprints. Thus, we can see RSS values in the scale of -30 to -103 dBm are mostly used in fingerprinting positioning. RSS beyond this range will have no or negligible impact in fingerprinting localization. Therefore, we considered quantizing the RSS only from -30 to -103 dBm. The databases provide RSS in dBm unit. The following equations can convert RSS value from dBm to mW and vice versa.

$$P_{\text{mW}} = 1 \text{ mW} \times 10^{\frac{P_{\text{dBm}}}{10}} \tag{3}$$

$$P_{\text{dBm}} = 10 \times \log_{10}(P_{\text{mW}}/1 \text{ mW}) \tag{4}$$

Since quantization happens at the absolute signal energy level, at first, we converted all RSS from dBm unit to mW unit using Equation (3) and then applied quantization. However, we found that, instead of mW unit, if quantization is applied on dBm unit, it provides the same results because they are just the same value represented by two different units.

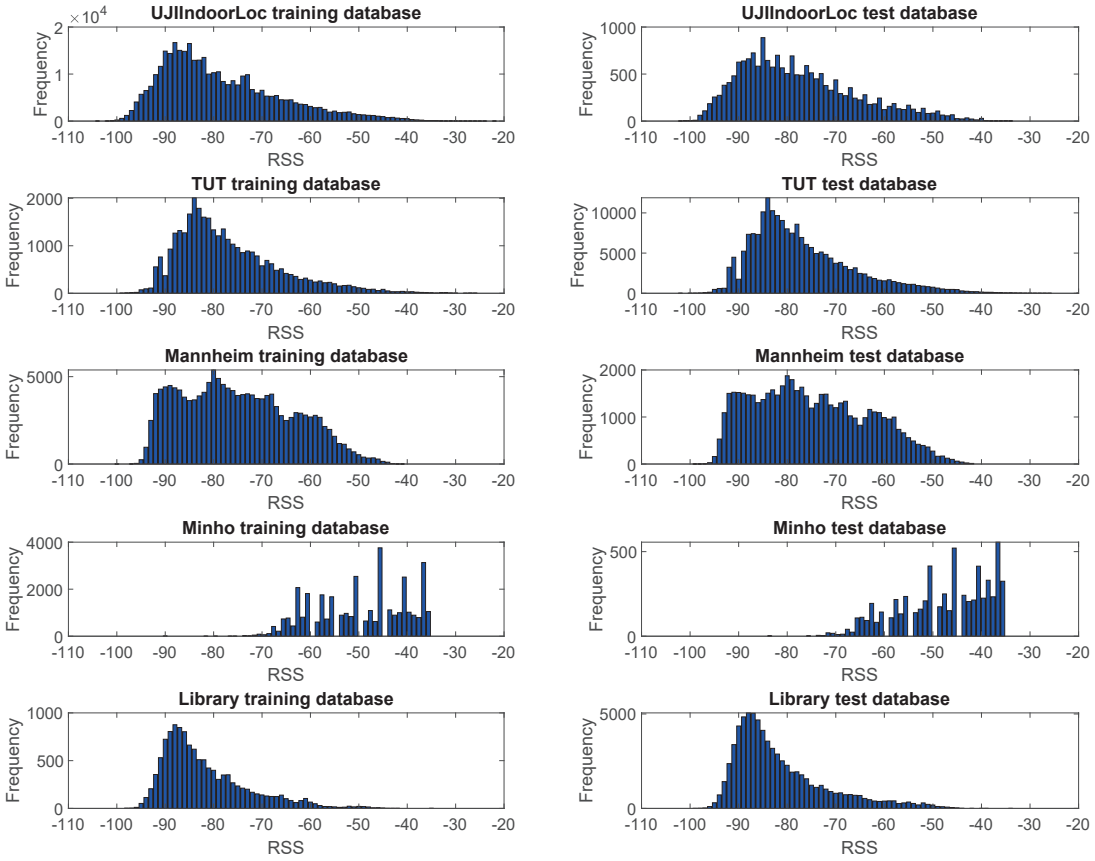


Figure 2. RSS distribution in the databases.

3.3.2. Quantizer

Figure 2 shows that the usable RSS range for fingerprinting localization is −30 to −103 dBm, which is equivalent to 10^{-3} to $10^{-10.3}$ mW. Thus, we can set maximum energy index, $E_{max} = -3$, and minimum energy index, $E_{min} = -10.3$. We proposed four different formulas (f_1, f_2, f_3, f_4) for seven different bit numbers (n) where $n = 2, 3, 4, 5, 6, 7, 8$ and $i = 1, \ldots, 2^n$ to divide the signal energy for each quantization level. The first formula results a linear quantization; depending on applied bit number, each level shares an equal amount of energy index.

$$f_1 = (E_{max} - E_{min})/(2^n - 1) \tag{5}$$

In the next three formulas, we applied non-linear quantization to reflect the nature of signal propagation. From the maximum energy index, in each quantized level, the energy index is reduced as follows

$$f_{2(i)} = (E_{max} - E_{min}) \times \frac{\sum_{i=1}^{i_{th}} \sqrt{2^{i-1}}}{\sum_{i=1}^{2^n} \sqrt{2^{i-1}}} \tag{6}$$

$$f_{3(i)} = (E_{max} - E_{min}) \times \frac{\sum_{i=1}^{i_{th}} \sqrt[3]{2^{i-1}}}{\sum_{i=1}^{2^n} \sqrt[3]{2^{i-1}}} \tag{7}$$

$$f_{4(i)} = (E_{max} - E_{min}) \times \frac{\sum_{i=1}^{i_{th}} log\left(2^{i-1}\right)}{\sum_{i=1}^{2^n} log\left(2^{i-1}\right)} \tag{8}$$

Figures 3 and 4 show the 3- and 4-bit quantizers, respectively.

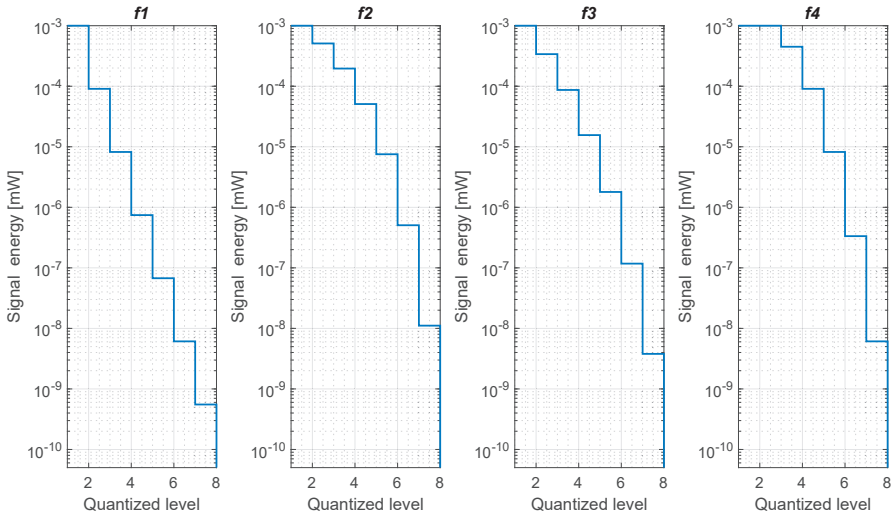


Figure 3. Three-bit quantization using proposed formulas.

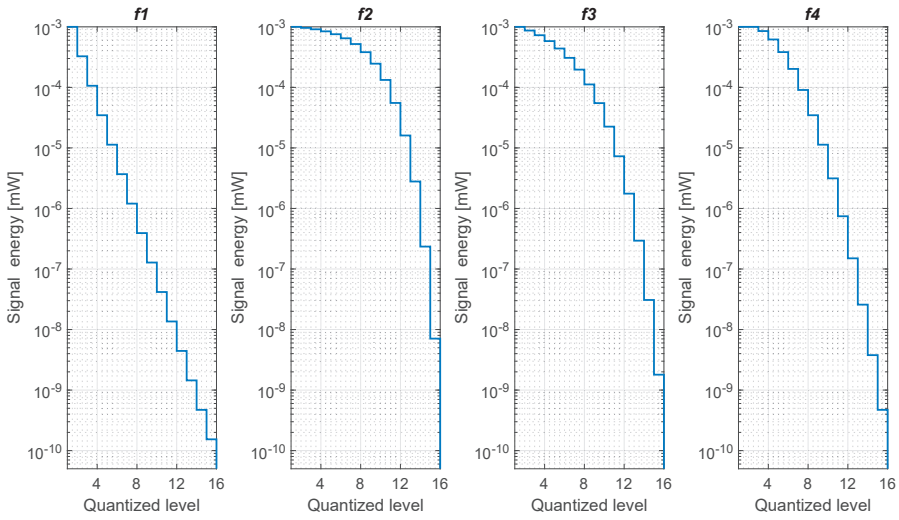


Figure 4. Four-bit quantization using proposed formulas.

3.3.3. Reason for the Same Positioning Performance by the Traditional and Quantized RSS Fingerprint

The proposed quantization scales down the traditional RSS values. To check the characteristic of an entire fingerprint before and after quantization, we randomly choose a sample from the TUT database. This example is based on linear quantization. This particular sample received signals from 61 APs. Figure 5 shows the fingerprints in traditional and quantized RSS representation, where the X-axis shows the reference AP and the Y-axis shows the RSS values. The 2-bit quantization has maximum capacity of four output levels, where each level represent $\mathrm{s}(-30-(-103))/(2^2-1) = 24.33$ dBm equivalent signal power. Thus, 24.33 dBm traditional RSS is mapped to each quantized level, where scaling ratio is 24.33:1; similarly for 3 to 5 bits, the scaling ratio is 10.42:1, 4.86:1, and 2.35:1, respectively. If the scaling ratio gets closer, more characteristics of the original fingerprint appear. Figure 5 shows that 2-bit quantization misses many characteristics of the traditional one, while the situation improves at 3-bit quantization. The 4-bit quantization seems to have most of the characteristics of the original fingerprint at a lower magnitude.

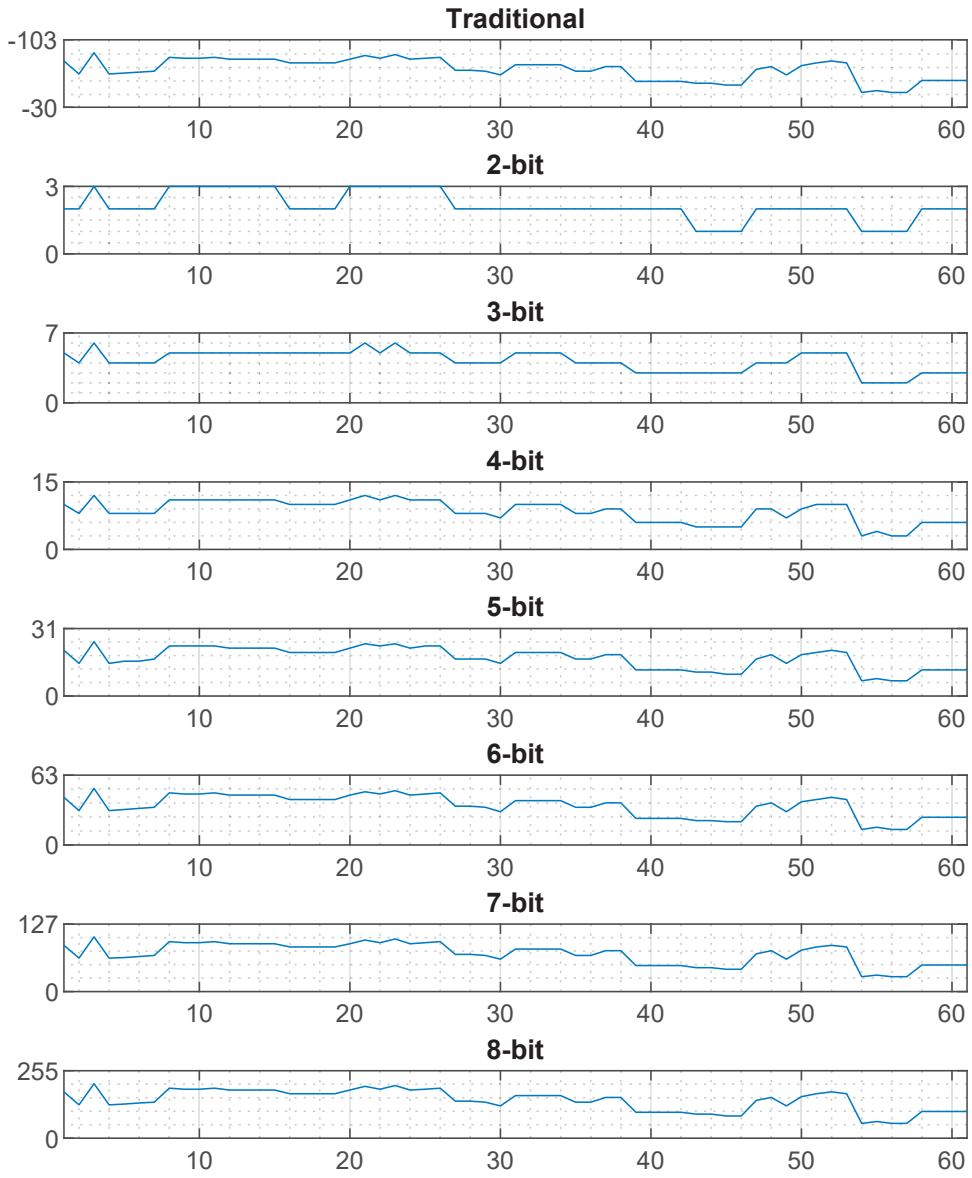


Figure 5. Traditional vs. quantized RSS fingerprint.

For the same amount of attenuation, a strong Wi-Fi signal travels less distance than that of a weak signal. The 2-bit quantization's granularity is 24.33 dBm. According to the ITU-R indoor propagation model, for this amount of attenuation, a 2.4 GHz Wi-Fi signal can travel approximately 7 m when it is strong or more than 30 m when it is weak [66]. Here, we considered approximately -30 dBm as strong and -70 dBm as weak signal levels. Due to this high-granular RSS, depending on the composition of a fingerprint (e.g., number of AP and signal quality), there is a possibility of having identical quantized fingerprints for many different locations, which is a problem. That is why we can see a significant amount of positioning error for low-bit quantization in Figure 6. If the granularity is reduced, the problem mitigates. At 4-bit quantization, the granularity is only 4.86 dBm. This amount of attenuation can cover approximately 1 m and 6 m at the strong and weak states, respectively. Since the attenuation covers less distance, the possibility of having identical quantized fingerprint for

the different locations also diminishes. Besides, the number of reference APs and a wide variety in signal level also helps a quantized fingerprint to be distinct from other quantized fingerprints. If each quantized RSS fingerprint is unique at their signal space, they will result in the same positioning accuracy as that of traditional RSS fingerprint. That is why 4-bit quantized fingerprint can achieve the same positioning accuracy as the traditional one. However, the situation may vary from sample to sample and environment to environment. Therefore, we carried a series of experiments on five different databases.

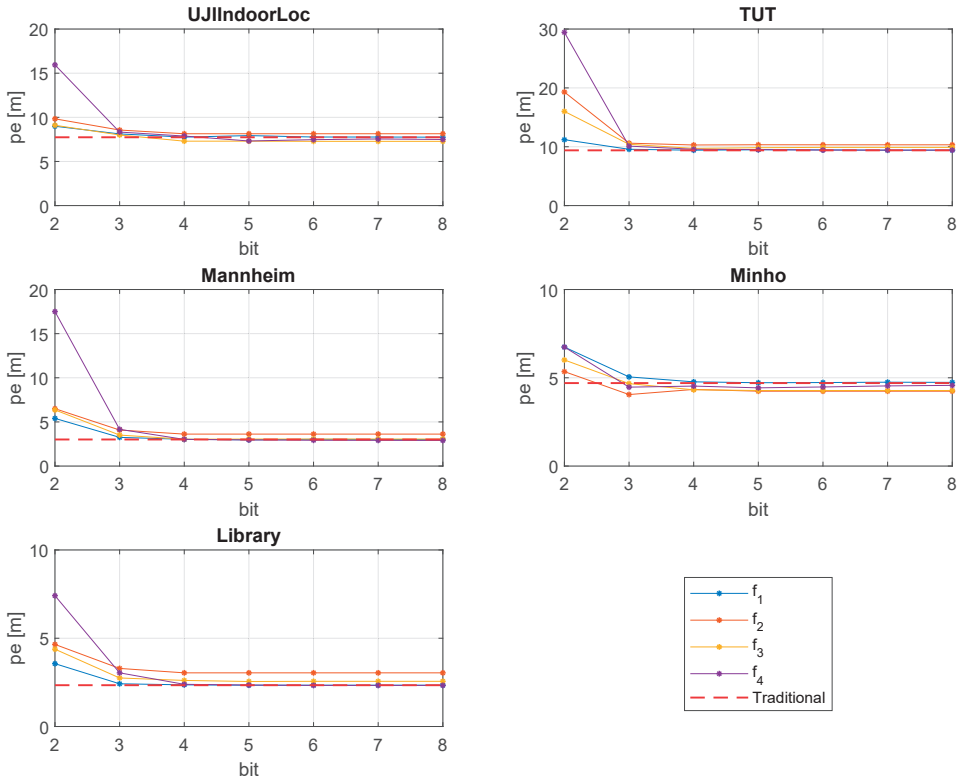


Figure 6. Positioning performance.

4. Experiment and Results

This section provides information regarding experiment and localization performance for different quantization setup.

4.1. Experiment Setup

Previous studies on these databases were mostly done based on popular Euclidean distance and *k*-NN algorithm [61–65,67]. To keep consistency with earlier studies, $k = 1$ was used in UJIIndoorLoc, TUT, and Mannheim. $k = 3$ was used in Minho and Library. If we change any parameters (in this case, *k* value), the experimental results would show not only the effect of proposed quantization but also a mixed impact of quantization and change of *k* value. However, our main target was to identify only the quantization effect. Therefore, we kept the $k = 1$ in some databases, and $k = 3$ in some other databases to keep similarity with the previous study. The distance between the ground truth and the calculated position (based on RSS) of a fingerprint is called positioning error (pe), which has been expressed in meters. Fingerprints in the TUT database have exact height information (not just the floor level number, but height in meters); therefore, we calculated 3D pe for the TUT database, For the other databases, it was 2D positioning. Our laboratory computer had an Intel Core i5-6600 processor with a clock speed of 3.3 GHz and 8 GB RAM, using Matlab 2018a programming software.

4.2. Positioning Performance

Figure 6 shows the quantization effect on fingerprinting positioning for all the databases. To compare the result with the traditional RSS representation, a red dashed line is drawn. We did not consider 1 bit because that results in a huge amount of *pe*. Two-bit quantization also shows a large *pe*. However, from 3-bit quantization, positioning accuracy dramatically improves. Dimension, environment, and the number of training and testing samples are quite different in all the databases. As a result, the localization performance slightly varies by different quantization formula. However, the 4-bit quantization results in almost the same positioning accuracy as that of traditional RSS. Beyond 4-bit quantization, positioning accuracy does not improve. Formulas f_1 and f_4 provide slightly better performance then that of formulas f_2 and f_3.

4.3. Floor Detection Performance

We also checked the efficiency of our proposed method for successful floor detection. Figure 7 shows the successful floor detection rate in Library, TUT, and UJIIndoorLoc. The other databases had no floor related information. In Figure 7, we can see that, in most cases, the performance of 2-bit quantization is not satisfactory. However, from 3-bit quantization, the successful floor detection rate starts to get closer to that of traditional RSS results. Depending on the database, performance slightly varies by the different formulas. Formula f_1 is the best performer in all the databases.

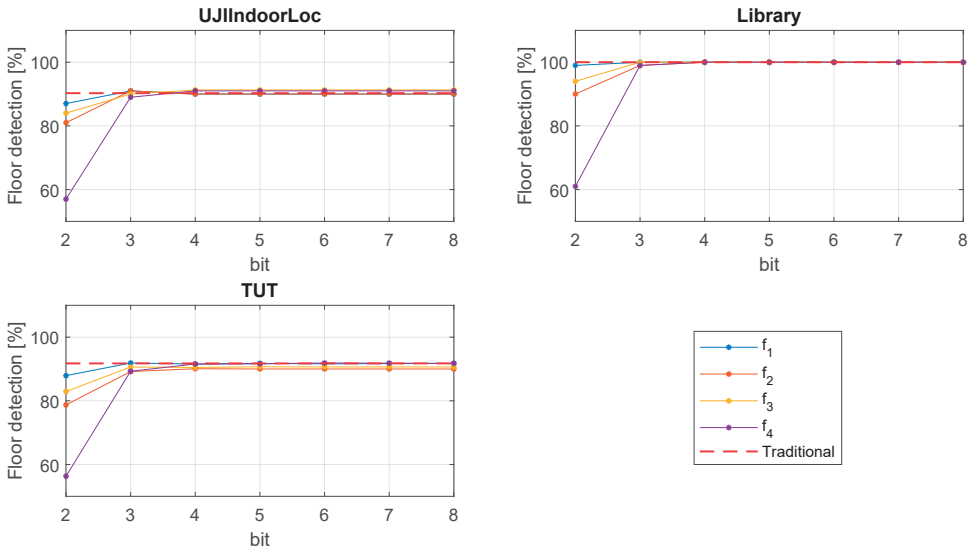


Figure 7. Floor detection performance.

5. Discussion

Our proposed method uses only the effective portion of RSS for fingerprinting positioning purposes. The result from the experiments shows that our proposed quantization method provides the same positioning performance as that of the traditional RSS representation by using only 4 bits. Since our proposed method uses only 4-bit quantization, it may bring the following advantages.

5.1. Simplification

Figure 2 shows that RSS from -30 to -103 dBm (or equivalent from 10^{-3} to $10^{-10.3}$ mW) is used in fingerprinting. Besides, a maximum 4-bit or 16-level quantization of that range is sufficient for the localization. Therefore, instead of a traditional 8-bit quantizer, a 4-bit quantizer can be used, which would simplify the hardware configuration and the sensing process.

5.2. Enhancing Security

Traditional RSS contains real signal strength information. Storing and exchanging this raw information could be a potential threat to the user's location privacy. An adversary in between client device and positioning server may eavesdrop and get the information; later, through a radio signal propagation model, the position of the user could be determined. However, our proposed method does not use the real RSS values, but the mapped one according to a hidden quantization formula and bit number. Therefore it would be significantly harder for any adversary to get the real RSS, and subsequently the position.

5.3. Less Storage

The representation of traditional RSS and quantized RSS are different. Since we use fewer bits, less memory will be occupied. Figure 8 shows the comparison of the storage space of the training databases, which generally located at the positioning server side. The efficiency and quantization strategy of all the proposed formulas are not the same. As a result, different formulas result in different memory sizes. In Figures 3 and 4 we can see f_2 and f_3 are biased to the strong part of the signal. Therefore, with the increase of bit number, the strong part of the signal gets more quantized, while a big chunk of the weak part of the signal is quantized to a single level.

As a result, disk size through f_2 and f_3 does not increase gradually. However, f_1 and f_4 quantize the signal linearly and show a gradual increase of memory size with the rise of bit number. In Figure 8, we can see that, compared to the traditional RSS, on average, 40–60% memory space in the training databases can be saved by 4-bit quantization.

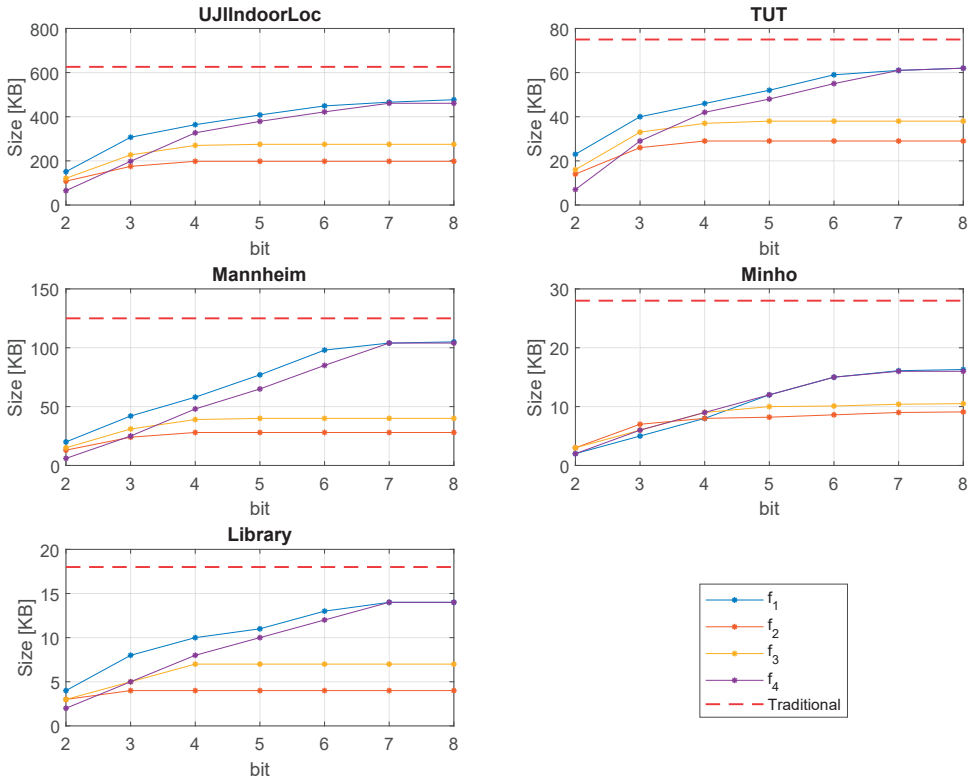


Figure 8. Training database storage size comparison.

5.4. Less Traffic

Users having no positioning information would request the server by exchanging test/online fingerprint. Thus, the data need to be exchanged between the client device and the positioning server. Since our proposed method scales down the data size, compared to the traditional RSS, there is less traffic while exchanging the online fingerprint. Figure 9 shows the test data size. Since the default values of non-heard APs are handled at the server-side, we excluded the non-heard AP reading from all the test fingerprints. Figure 9 shows that, through the proposed method, there is an excellent opportunity to reduce the network traffic while exchanging the online fingerprint. Depending on the database and applied formula, compared to the traditional approach, on average, 40–60% network traffic reduction is possible by the proposed 4-bit quantization.

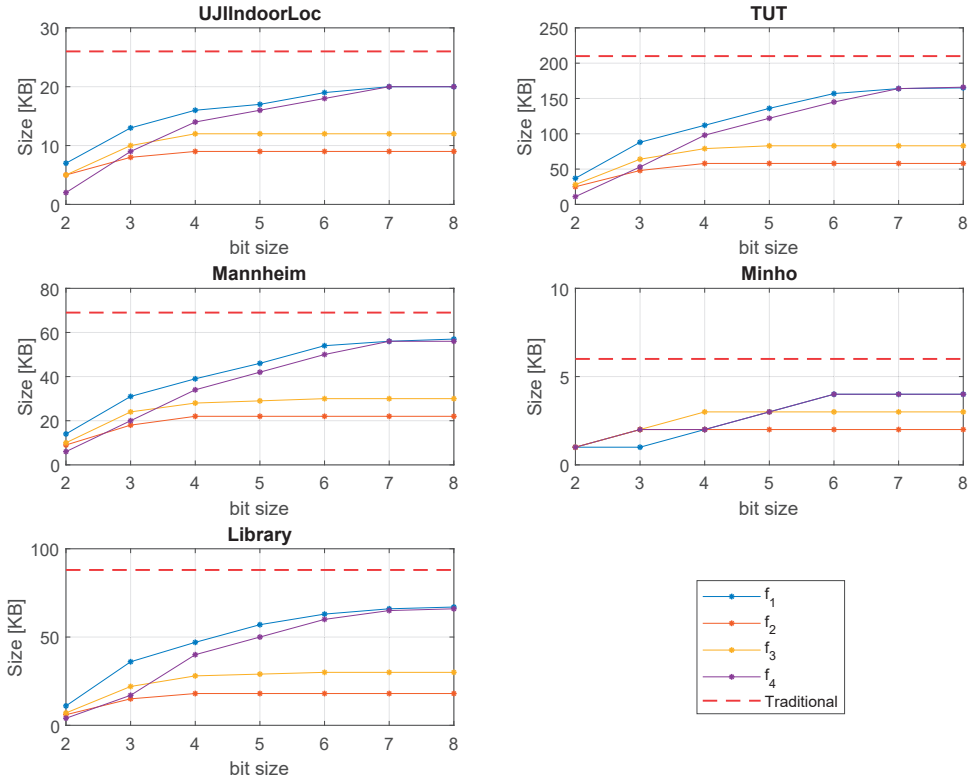


Figure 9. Test database storage size comparison.

5.4.1. Observation

Based on the outcome of the experiments, we can draw the following observations:

- The effective range of RSS is from −30 to −103 dBm for fingerprinting localization. Any value beyond this range can be considered as noise.
- In Wi-Fi fingerprinting localization, 4-bit quantization is enough.
- To have good quality fingerprints, a well planned, organized Wi-Fi network is desirable.
- Sometimes, the number of APs per fingerprint in the crowd-sourced database is very high. RSS from the reliable or known AP should be used.
- Hot-spot or mobile AP can contribute a significant amount of noise. Data from those should be ignored.

6. Conclusions

In this study, we investigated the received signal strength quantization effect for localization in five different databases by four different formulas. The environment, data collection strategy, coverage area, number of training and testing fingerprints, number of floors, and number of APs were different from one database to another. Without having any correlation in all the databases, we found that 4-bit quantization can achieve the same localization accuracy as that of traditional RSS. Generally, the traditional RSS is quantized by 8 bits. However, most Wi-Fi chip manufacturers do not reveal their quantization strategy. In general, we can see the typical range of the RSS is between 0 dBm and −100 dBm that needs 101 different levels, which is in the range of at least 7 bits. In our study, we found that, for fingerprinting positioning purposes, 4-bit or 16-level quantization is sufficient. The proposed quantization method can help to simplify the hardware configuration, conceal the real RSS, save approximately 40–60% storage space, and data traffic. Since the quantization happens just after sensing the signal, we used the signal strength in the mW unit. However, we found that, instead of mW format, if quantization is done based on the dBm unit, that also provides the same localization accuracy. The performance of all the quantization formulas slightly varies. Nonetheless, linear quantization is a consistent performer. We achieved the performance limit of traditional RSS through 4-bit quantization.

One limitation of this study is that quantization was analyzed using *k*-NN as the main localization algorithm. Probably, quantization will have a similar impact on the results reported by the variants based on weighted *k*-NN. Therefore, as further work, we shall focus on how to reduce the computational load and increase localization accuracy through an efficient combination of RSS quantization, radio map clustering, AP filtering, and advanced fingerprint algorithms.

Author Contributions: S.K. conceived, designed, and performed the experiments; S.K. and J.T.-S. wrote the paper; and J.T.-S. and T.R. guided the paper writing and reviewed the paper. All authors have read and agreed to the published version of the manuscript.

Funding: Syed Khandker expresses his warm thanks to the Riitta and Jorma J. Takanen foundation for its financial support. Joaquín Torres-Sospedra is funded by the Torres-Quevedo Programme of the Spanish government, Grant No. PTQ2018-009981 (Project INSIGNIA).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bergen, M.H.; Arafa, A.; Jin, X.; Klukas, R.; Holzman, J.F. Characteristics of Angular Precision and Dilution of Precision for Optical Wireless Positioning. *J. Lightwave Technol.* **2015**, *33*, 4253–4260. [CrossRef]
2. Perez, M.C.; Ureña, J.; Hernandez, A.; Jimenez, A.; Ruiz, D.; Alvarez, F.J.; De Marziani, C. Performance comparison of different codes in an ultrasonic positioning system using DS-CDMA. In Proceedings of the 2009 IEEE International Symposium on Intelligent Signal Processing, Budapest, Hungary, 26–28 August 2009; pp. 125–130.
3. Jimenez, A.R.; Seco, F.; Prieto, C.; Guevara, J. A comparison of Pedestrian Dead-Reckoning algorithms using a low-cost MEMS IMU. In Proceedings of the 2009 IEEE International Symposium on Intelligent Signal Processing, Budapest, Hungary, 26–28 August 2009; pp. 37–42.
4. Jiménez Ruiz, A.R.; Seco Granja, F. Comparing Ubisense, BeSpoon, and DecaWave UWB Location Systems: Indoor Performance Analysis. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 2106–2117. [CrossRef]
5. Liu, M.; Wang, H.; Yang, Y.; Zhang, Y.; Ma, L.; Wang, N. RFID 3-D Indoor Localization for Tag and Tag-Free Target Based on Interference. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 3718–3732. [CrossRef]
6. De-La-Llana-Calvo, A.; Lázaro-Galilea, J.L.; Gardel-Vicente, A.; Rodríguez-Navarro, D.; Rubiano-Muriel, B.; Bravo-Muñoz, I. Analysis of Multiple-Access Discrimination Techniques for the Development of a PSD-Based VLP System. *Sensors* **2020**, *20*, 1717. [CrossRef]
7. Ji, M.; Kim, J.; Jeon, J.; Cho, Y. Analysis of positioning accuracy corresponding to the number of BLE beacons in indoor positioning system. In Proceedings of the 2015 17th International Conference on Advanced Communication Technology (ICACT), Seoul, Korea, 1–3 July 2015; pp. 92–95.

8. He, S.; Chan, S..G. Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 466–490. [CrossRef]
9. Georgiou, K.; Constambeys, T.; Laoudias, C.; Petrou, L.; Chatzimilioudis, G.; Zeinalipour-Yazti, D. Anyplace: A Crowdsourced Indoor Information Service. In Proceedings of the 2015 16th IEEE International Conference on Mobile Data Management, Pittsburgh, PA, USA, 15–18 June 2015; Volume 1, pp. 291–294.
10. Moreira, A.; Meneses, F. Where@UM—Dependable organic radio maps. In Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Calgary, AB, Canada, 13–16 October 2015.
11. Nguyen, D.; Recalde, M.E.V.; Nashashibi, F. Low speed vehicle localization using WiFi fingerprinting. In Proceedings of the 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), Phuket, Thailand, 13–15 November 2016. [CrossRef]
12. Hernández, N.; Hussein, A.; Cruzado, D.; Parra, I.; Armingol, J.M. Applying low cost WiFi-based localization to in-campus autonomous vehicles. In Proceedings of the 20th IEEE International Conference on Intelligent Transportation Systems, ITSC 2017, Yokohama, Japan, 16–19 October 2017.
13. Zhang, Y.; Li, D.; Wang, Y. An Indoor Passive Positioning Method Using CSI Fingerprint Based on Adaboost. *IEEE Sens. J.* **2019**, *19*, 5792–5800. [CrossRef]
14. Talvitie, J.; Renfors, M.; Lohan, E.S. Distance-Based Interpolation and Extrapolation Methods for RSS-Based Localization With Indoor Wireless Signals. *IEEE Trans. Veh. Technol.* **2015**, *64*, 1340–1353. [CrossRef]
15. Huang, C.; Manh, H. RSS-Based Indoor Positioning Based on Multi-Dimensional Kernel Modeling and Weighted Average Tracking. *IEEE Sens. J.* **2016**, *16*, 3231–3245. [CrossRef]
16. Achroufene, A.; Amirat, Y.; Chibani, A. RSS-Based Indoor Localization Using Belief Function Theory. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1163–1180. [CrossRef]
17. Richter, P.; Leppakoski, H.; Lohan, E.S.; Yang, Z.; Jarvinen, K.; Tkachenko, O.; Schneider, T. Received Signal Strength Quantization for Secure Indoor Positioning via Fingerprinting. In Proceedings of the 2018 8th International Conference on Localization and GNSS (ICL-GNSS), Guimarães, Portugal, 26–28 June 2018. [CrossRef]
18. Mizmizi, M.; Reggiani, L. Design of RSSI based fingerprinting with reduced quantization measures. In Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Madrid, Spain, 4–7 October 2016. [CrossRef]
19. Li, H.; Sun, L.; Zhu, H.; Lu, X.; Cheng, X. Achieving privacy preservation in WiFi fingerprint-based localization. In Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 2337–2345. [CrossRef]
20. Konstantinidis, A.; Chatzimilioudis, G.; Zeinalipour-Yazti, D.; Mpeis, P.; Pelekis, N.; Theodoridis, Y. Privacy-Preserving Indoor Localization on Smartphones. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 3042–3055. [CrossRef]
21. Hernandez, A.; Ureña, J.; Mazo, M.; Jimenez, A.; Garcia, J.J.; Marziani, C.; Ochoa, A.; Villadangos, J.M.; Jimenez, J.A.; Alvarez, F.J. A comparison of computing architectures for ultrasonic signal processing. In Proceedings of the IEEE International Workshop on Intelligent Signal Processing, Faro, Portugal, 1–3 September 2005; pp. 38–40.
22. Rodríguez-Navarro, D.; Lázaro-Galilea, J.L.; Bravo-Muñoz, I.; Gardel-Vicente, A.; Tsirigotis, G. Analysis and Calibration of Sources of Electronic Error in PSD Sensor Response. *Sensors* **2016**, *16*, 619. [CrossRef] [PubMed]
23. Jiao, J.; Deng, Z. Deep combining of local phase quantization and histogram of oriented gradients for indoor positioning based on smartphone camera. *Int. J. Distrib. Sens. Netw.* **2017**, *13*. [CrossRef]
24. Hilsenbeck, S.; Bobkov, D.; Schroth, G.; Huitl, R.; Steinbach, E. Graph-Based Data Fusion of Pedometer and WiFi Measurements for Mobile Indoor Positioning. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '14, Seattle, WA, USA, 13–17 Sepember 2014; Association for Computing Machinery: New York, NY, USA, 2014; pp. 147–158. [CrossRef]
25. Kim, B.Y.; Cho, J.-S.; Park, Y..; Kim, K.-D. Implementation of indoor positioning using LED and dual PC cameras. In Proceedings of the 2012 Fourth International Conference on Ubiquitous and Future Networks (ICUFN), Phuket, Thailand, 4–6 July 2012; pp. 476–477.
26. Coleman, D.D.; Westcott, D.A. *CWNA Certified Wireless Network Administrator Official Study Guide, Exam CWNA-106*; Sybex: Hoboken, NJ, USA, 2014; p. 92.

27. Attila, B.; Lung, C. On the relationship between received signal strength and received signal strength index of IEEE 802.11 compatible radio transceivers. *Carpathian J. Electron. Comput. Eng.* **2013**, *6*, 15–20.
28. Gao, H. Investigation of Context Determination for Advanced Navigation Using Smartphone Sensors. Ph.D. Thesis, University College London, London, UK, 2019.
29. Zanella, A. Best Practice in RSS Measurements and Ranging. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2662–2686. [CrossRef]
30. Gao, W.; Nikolaidis, I.; Harms, J.J. RSSI quantization for indoor localization services. In Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 8–13 October 2017. [CrossRef]
31. Bahl, P.; Padmanabhan, V.N. RADAR: an in-building RF-based user location and tracking system. In Proceedings of the IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064), Tel Aviv, Israel, 26–30 March 2000; Volume 2, pp. 775–784.
32. Beomju Shin.; Jung Ho Lee.; Taikjin Lee.; Hyung Seok Kim. Enhanced weighted K-nearest neighbor algorithm for indoor Wi-Fi positioning systems. In Proceedings of the 2012 8th International Conference on Computing Technology and Information Management (NCM and ICNIT), Seoul, Korea, 24–26 April 2012; Volume 2, pp. 574–577.
33. Hu, X.; Shang, J.; Gu, F.; Han, Q. Improving Wi-Fi Indoor Positioning via AP Sets Similarity and Semi-Supervised Affinity Propagation Clustering. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 109642. [CrossRef]
34. Chen, W.; Chang, Q.; Hong-tao, H.; Wang, W. A novel clustering and KWNN-based strategy for Wi-Fi fingerprint indoor localization. In Proceedings of the 2015 4th International Conference on Computer Science and Network Technology (ICCSNT), Harbin, China, 19–20 December 2015; Volume 1, pp. 49–52.
35. Razavi, A.; Valkama, M.; Lohan, E. K-Means Fingerprint Clustering for Low-Complexity Floor Estimation in Indoor Mobile Localization. In Proceedings of the 2015 IEEE Globecom Workshops (GC Wkshps), San Diego, CA, USA, 6–10 December 2015. [CrossRef]
36. Youssef, M.A.; Agrawala, A.; Udaya Shankar, A. WLAN location determination via clustering and probability distributions. In Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, (PerCom 2003), Fort Worth, TX, USA, 23–26 March 2003; pp. 143–150.
37. Berkvens, R.; Peremans, H.; Weyn, M. Conditional Entropy and Location Error in Indoor Localization Using Probabilistic Wi-Fi Fingerprinting. *Sensors* **2016**, *16*, 1636. [CrossRef]
38. Lee, C.; Lin, T.; Fang, S.; Chou, Y. A novel clustering-based approach of indoor location fingerprinting. In Proceedings of the 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), London, UK, 8–11 September 2013; pp. 3191–3196.
39. Caso, G.; Nardis, L.D. Virtual and Oriented WiFi Fingerprinting Indoor Positioning based on Multi-Wall Multi-Floor Propagation Models. *Mob. Netw. Appl.* **2016**, *22*, 825–833. [CrossRef]
40. Renaudin, V.; Ortiz, M.; Perul, J.; Torres-Sospedra, J.; Jiménez, A.R.; Pérez-Navarro, A.; Mendoza-Silva, G.M.; Seco, F.; Landau, Y.; Marbel, R.; et al. Evaluating Indoor Positioning Systems in a Shopping Mall: The Lessons Learned From the IPIN 2018 Competition. *IEEE Access* **2019**, *7*, 148594–148628. [CrossRef]
41. Anagnostopoulos, G.G.; Kalousis, A. A Reproducible Analysis of RSSI Fingerprinting for Outdoor Localization Using Sigfox: Preprocessing and Hyperparameter Tuning. In Proceedings of the 2019 International Conference on Indoor Positioning and Indoor Navigation, Pisa, Italy, 30 September–3 October 2019.
42. Khandker, S.; Mondal, R.; Ristaniemi, T. Positioning Error Prediction and Training Data Evaluation in RF Fingerprinting Method. In Proceedings of the 2019 International Conference on Indoor Positioning and Indoor Navigation, Pisa, Italy, 30 September–3 October 2019.
43. Rojo, J.; Mendoza-Silva, G.M.; Ristow Cidral, G.; Laiapea, J.; Parrello, G.; Simó, A.; Stupin, L.; Minican, D.; Farrés, M.; Corvalán, C.; et al. Machine Learning applied to Wi-Fi fingerprinting: The experiences of the Ubiqum Challenge. In Proceedings of the 2019 International Conference on Indoor Positioning and Indoor Navigation, Pisa, Italy, 30 September–3 October 2019.
44. Järvinen, K.; Leppäkoski, H.; Lohan, E.; Richter, P.; Schneider, T.; Tkachenko, O.; Yang, Z. PILOT: Practical Privacy-Preserving Indoor Localization Using OuTsourcing. In Proceedings of the 2019 IEEE European Symposium on Security and Privacy (EuroS P), Stockholm, Sweden, 17–19 June 2019; pp. 448–463. [CrossRef]
45. Ln Nguyen, T.; D Vy, T.; Shin, Y. An Efficient Hybrid RSS-AoA Localization for 3D Wireless Sensor Networks. *Sensors* **2019**, *19*, 2121. [CrossRef]

46. Ababneh, A. Low-Complexity Bit Allocation for RSS Target Localization. *IEEE Sens. J.* **2019**, *19*, 7733–7743. [CrossRef]
47. Niu, R.; Vempaty, A.; Varshney, P.K. Received-Signal-Strength-Based Localization in Wireless Sensor Networks. *Proc. IEEE* **2018**, *106*, 1166–1182. [CrossRef]
48. Li, Z.; Chung, P.J.; Mulgrew, B. Distributed target localization using quantized received signal strength. *Signal Process.* **2017**, *134*, 214–223. [CrossRef]
49. Torres-Sospedra, J.; Moreira, A. Analysis of Sources of Large Positioning Errors in Deterministic Fingerprinting. *Sensors* **2017**, *17*, 2736. [CrossRef] [PubMed]
50. Shi, H.; Li, X.; Shang, Y.; Ma, D. Cramer-Rao Bound Analysis of Quantized RSSI Based Localization in Wireless Sensor Networks. In Proceedings of the 11th International Conference on Parallel and Distributed Systems (ICPADS'05), Fukuoka, Japan, 20–22 July 2005; Volume 2, pp. 32–36. [CrossRef]
51. Krishnamachari, B. *Networking Wireless Sensors*; Cambridge University Press: New York, NY, USA, 2005.
52. Patwari, N.; Hero, A.O., III. Using Proximity and Quantized RSS for Sensor Localization in Wireless Networks. In Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications, WSNA '03, San Diego, CA, USA, 19 September 2003; ACM: New York, NY, USA, 2003; pp. 20–29. [CrossRef]
53. Eisa, S.; Peixoto, J.; Meneses, F.; Moreira, A. Removing useless APs and fingerprints from WiFi indoor positioning radio maps. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation, Montbeliard-Belfort, France, 28–31 October 2013.
54. Chen, Y.; Lymberopoulos, D.; Liu, J.; Priyantha, B. Indoor Localization Using FM Signals. *IEEE Trans. Mob. Comput.* **2013**, *12*, 1502–1517. [CrossRef]
55. Altintas, B.; Serif, T. Improving RSS-Based Indoor Positioning Algorithm via K-Means Clustering. In Proceedings of the 17th European Wireless 2011—Sustainable Wireless Technologies, Vienna, Austria, 27–29 April 2011.
56. Gallagher, T.J.; Li, B.; Dempster, A.G.; Rizos, C. A sector-based campus-wide indoor positioning system. In Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation, Zurich, Switzerland, 15–17 September 2010.
57. Marques, N.; Meneses, F.; Moreira, A. Combining similarity functions and majority rules for multi-building, multi-floor, WiFi positioning. In Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney, Australia, 13–15 November 2012.
58. Moreira, A.; Nicolau, M.J.; Meneses, F.; Costa, A. Wi-Fi fingerprinting in the real world - RTLS@UM at the EvAAL competition. In Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Calgary, AB, Canada, 13–16 October 2015.
59. Torres-Sospedra, J.; Quezada-Gaibor, D.; Mendoza-Silva, G.M.; Nurmi, J.; Koucheryavy, Y.; Huerta, J. New Cluster Selection and Fine-grained Search for k-Means Clustering and Wi-Fi Fingerprinting. In Proceedings of the 2020 International Conference on Localization and GNSS (ICL-GNSS), Tampere, FI, USA, 2–4 June 2020.
60. Yu, F.; Jiang, M.; Liang, J.; Qin, X.; Hu, M.; Peng, T.; Hu, X. 5 G WiFi Signal-Based Indoor Localization System Using Cluster -Nearest Neighbor Algorithm. *Int. J. Distrib. Sens. Netw.* **2014**, *2014*. [CrossRef]
61. Torres-Sospedra, J.; Montoliu, R.; Martínez-Usó, A.; Avariento, J.P.; Arnau, T.J.; Benedito-Bordonau, M.; Huerta, J. UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. In Proceedings of the 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Busan, Korea, 27–30 October 2014; pp. 261–270.
62. Lohan, E.; Torres-Sospedra, J.; Leppäkoski, H.; Richter, P.; Peng, Z.; Huerta, J. Wi-Fi Crowdsourced Fingerprinting Dataset for Indoor Positioning. *Data* **2017**, *2*, 32. [CrossRef]
63. Moreira, A.; Silva, I.; Meneses, F.; Nicolau, M.J.; Pendao, C.; Torres-Sospedra, J. Multiple simultaneous Wi-Fi measurements in fingerprinting indoor positioning. In Proceedings of the 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, Japan, 18–21 September 2017. [CrossRef]
64. King, T.; Kopf, S.; Haenselmann, T.; Lubberger, C.; Effelsberg, W. COMPASS: A Probabilistic Indoor Positioning System Based on 802.11 and Digital Compasses. In Proceedings of the 1st International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, Los Angeles, CA, USA, 23–29 September 2006; pp. 34–40. [CrossRef]
65. Mendoza-Silva, G.; Richter, P.; Torres-Sospedra, J.; Lohan, E.; Huerta, J. Long-Term WiFi Fingerprinting Dataset for Research on Robust Indoor Positioning. *Data* **2018**, *3*, 3. [CrossRef]

66. Rath, H.K.; Timmadasari, S.; Panigrahi, B.; Simha, A. Realistic indoor path loss modeling for regular WiFi operations in India. In Proceedings of the 2017 Twenty-Third National Conference on Communications (NCC), Chennai, India, 2–4 March 2017. [CrossRef]
67. Torres-Sospedra, J.; Montoliu, R.; Trilles, S.; Belmonte, Ó.; Huerta, J. Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems. *Expert Syst. Appl.* **2015**, *42*, 9263–9278. [CrossRef]

PV

CYBERSECURITY ATTACKS ON SOFTWARE LOGIC AND ERROR HANDLING WITHIN ADS-B IMPLEMENTATIONS: SYSTEMATIC TESTING OF RESILIENCE AND COUNTERMEASURES

by

S Khandker, H Turtiainen, A Costin, T Hamalainen 2021

IEEE Transactions on Aerospace and Electronic Systems, Early access article

https://doi.org/10.1109/taes.2021.3139559

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

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

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

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

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

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

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

I. Introduction

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

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

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

[1] Also known as "ICAO24 code".

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

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

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

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

II. Overview of ADS-B

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

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

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

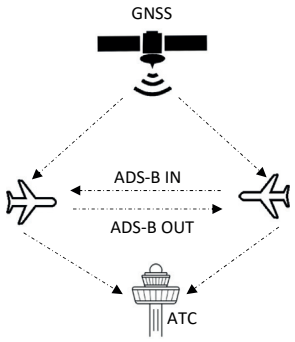


Fig. 1: ADS-B communication concept.

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

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

III. Related Works

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

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

(a) ADS-B 1090ES

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

(b) ADS-B UAT978

Fig. 2: ADS-B message structures.

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

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

IV. Attacks on ADS-B

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

A. Aircraft reconnaissance

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

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

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

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

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

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

B. Spoofing

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

C. Flooding

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

D. False distress signal

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

E. Coordinated attackers

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

F. Attacks on ADS-B CRC error handling

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

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

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

G. DoS attacks on the ADS-B protocol

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

H. Fuzzing avionics protocols

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

I. Logically-invalid data encoding

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

J. Jamming

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

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

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

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

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

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

V. Experimental setup

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

TABLE II: List of hardware used in the experiments.

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

A. Attacking hardware and devices

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

TABLE III: List of software tested.

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

B. Receiving hardware and devices

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

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

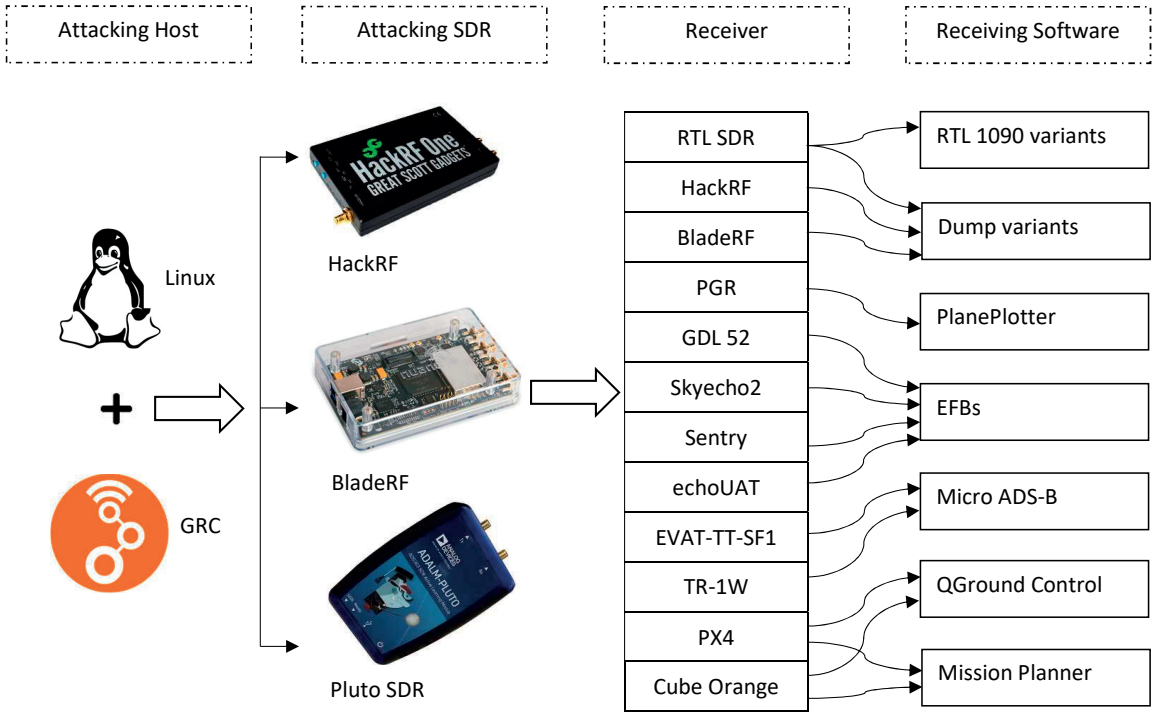


Fig. 3: Experimental attacking setup.

C. Receiving software

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

VI. Results and Evaluation

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

A. Aircraft reconnaissance

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

B. Spoofing

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

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

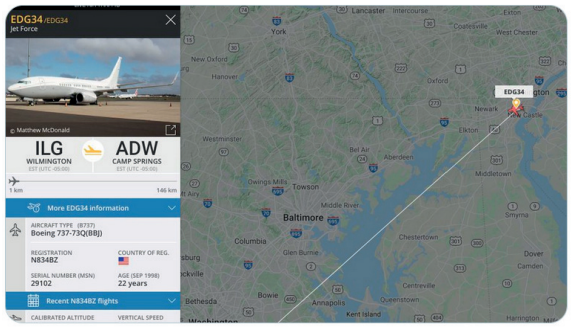


Fig. 4: President Joe Biden's flight for his presidential inauguration (image courtesy: https://twitter.com/flightradar24/status/1351628618187862026).

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

C. Flooding

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

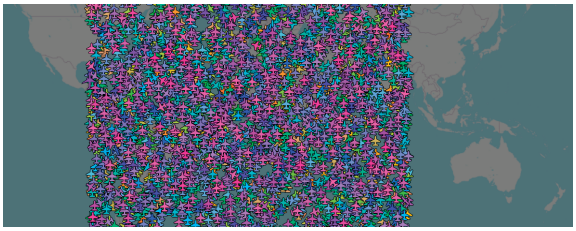


Fig. 6: Flooded screen when using tar1090 software.

D. False distress signal

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

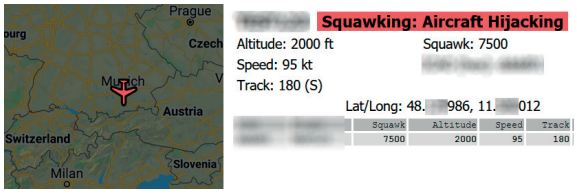


Fig. 7: Fake distress squawk code in the Dump1090 net.

E. Coordinated attackers

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

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

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

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

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

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

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

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

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

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

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

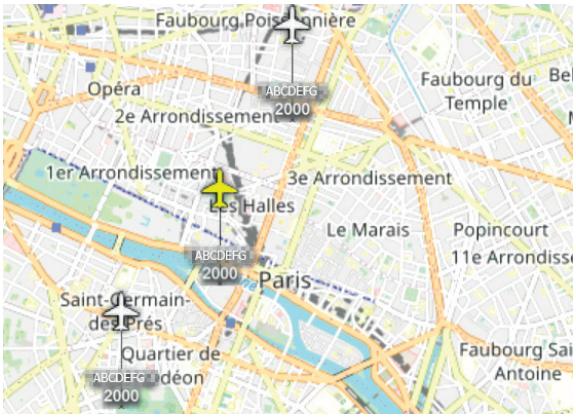


Fig. 8: Virtual Radar displaying the same flight number for multiple aircraft with distinct ICAO24 codes.

F. Attacks on ADS-B CRC error handling

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

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

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

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

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

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

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

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

G. DoS attacks on ADS-B protocol level

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

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

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

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

H. Fuzzing avionics protocols

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

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

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

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

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

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

I. Logically invalid data encoding

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

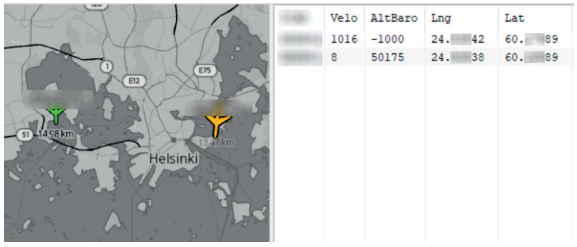


Fig. 9: Logically invalid data displayed in ADS-B Micro.

J. Jamming

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

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

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

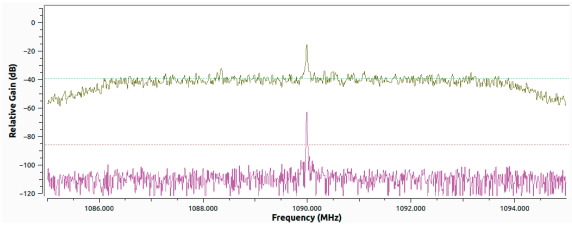


Fig. 10: Rise of the noise floor due to the jamming attack on ADS-B 1090ES.

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

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

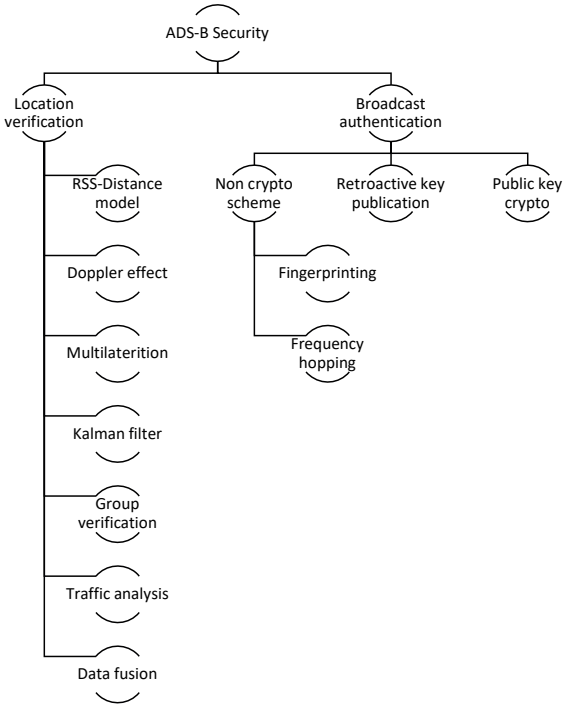


Fig. 11: Ontological tree classification of defensive measures for ADS-B security.

VII. Countermeasures and Defenses for ADS-B Security

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

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

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

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

A. Defense using the RSS-Distance model

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

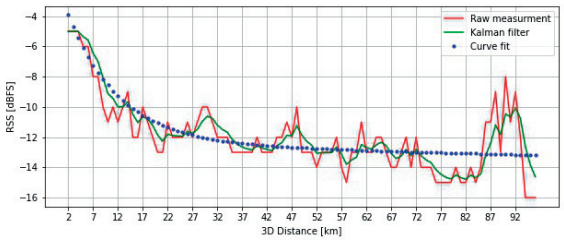


Fig. 12: Example of the the RSS-Distance model created based on the the ADS-B signals from real aircraft.

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

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

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

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

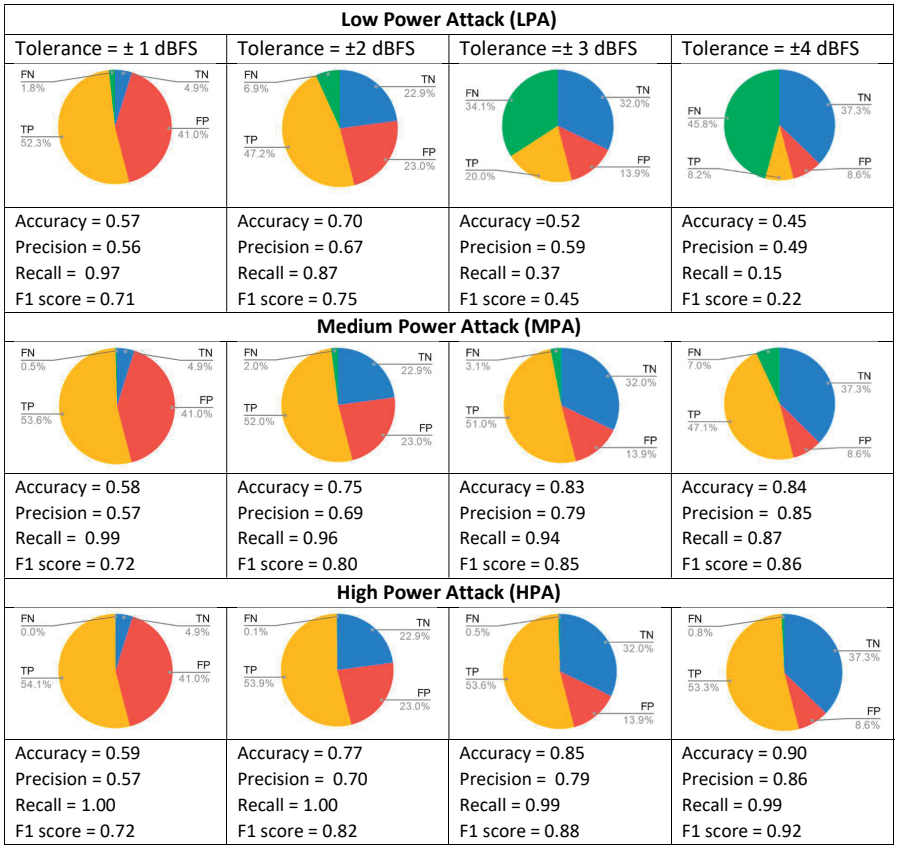


Fig. 13: Results of the detection of the attacker's ADS-B messages when the previously built RSS-Distance model was used.

B. Defense using the Doppler shift

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

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

C. Defense against coordinated attacks

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

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

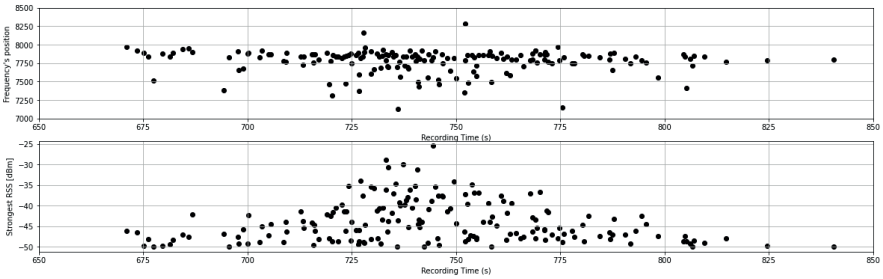


Fig. 14: Position of the strongest RSS in the FFT for evaluation of the Doppler shift.

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

D. Defense against other attacks

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

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

VIII. Conclusion

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

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

Acknowledgment

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

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

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

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

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

REFERENCES

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[41] *Reception of automatic dependent surveillance broadcast via satellite and compatibility studies with incumbent systems in the frequency band 1 087.7-1 092.3 MHz*, International Telecommunication Union, 2017.

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

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

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

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

PVI

GDL90FUZZ: FUZZING "GDL-90 DATA INTERFACE SPECIFICATION" WITHIN AVIATION SOFTWARE AND AVIONICS DEVICES–A CYBERSECURITY PENTESTING PERSPECTIVE

by

H Turtiainen, A Costin, S Khandker, T Hamalainen 2022

IEEE Access, 10, 21554-21562

https://doi.org/10.1109/ACCESS.2022.3150840

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 26, 2021, accepted January 26, 2022, date of publication February 10, 2022, date of current version March 2, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3150840

GDL90fuzz: Fuzzing - GDL-90 Data Interface Specification Within Aviation Software and Avionics Devices–A Cybersecurity Pentesting Perspective

HANNU TURTIAINEN, ANDREI COSTIN, SYED KHANDKER, AND TIMO HÄMÄLÄINEN
Faculty of Information Technology, University of Jyväskylä, FI-40014 Jyväskylä, Finland

Corresponding author: Hannu Turtiainen (hannu.ht.turtiainen@jyu.fi)

This work was supported in part by the Engage Consortium's Knowledge Transfer Network (KTN) funding for project *"Engage—204—Proof-of-Concept: Practical, Flexible, Affordable Pentesting Platform for ATM/Avionics Cybersecurity"* Single European Sky ATM Research (SESAR) Joint Undertaking under the European Union's Horizon 2020 Research and Innovation Program under Grant 783287, in part by the Finnish Grid and Cloud Infrastructure (FGCI) Persistent Identifier under Grant urn:nbn:fi:research-infras-2016072533, in part by the Decision of the Research Dean on Research funding within the Faculty of Information Technology of the University of Jyväskylä, and in part by the Finnish Cultural Foundation under Grant 00211119.

ABSTRACT As the core technology of next-generation air transportation systems, the Automatic Dependent Surveillance-Broadcast (ADS-B) is becoming very popular. However, many (if not most) ADS-B devices and implementations support and rely on Garmin's Datalink 90 (GDL-90) protocol for data exchange and encapsulation. This makes it essential to investigate the integrity of the GDL-90 protocol especially against attacks on the core subsystem availability, such as denial-of-service (DoS), which pose high risks to safety-critical and mission-critical systems such as in avionics and aerospace. In this paper, we consider GDL-90 protocol fuzzing options and demonstrate practical DoS attacks on popular electronic flight bag (EFB) software operating on mobile devices. Then we present our own specially configured avionics pentesting platform and the GDL-90 protocol. We captured legitimate traffic from ADS-B avionics devices. We ran our samples through the state-of-the-art fuzzing platform American Fuzzy Lop (AFL) and fed the AFL's output to EFB apps and the GDL-90 decoding software via the network in the same manner as legitimate GDL-90 traffic would be sent from ADS-B and other avionics devices. The results showed worrying and critical lack of security in many EFB applications where the security is directly related to the aircraft's safe navigation. Out of the 16 tested configurations, our avionics pentesting platform managed to crash or otherwise impact 9 (56%). The observed problems manifested as crashes, hangs, and abnormal behaviors of the EFB apps and GDL-90 decoders during the fuzzing test. Our developed and proposed systematic pentesting methodology for avionics devices, protocols, and software can be used to discover and report vulnerabilities as early as possible.

INDEX TERMS GDL-90, ADS-B, attacks, cybersecurity, pentesting, resiliency, DoS, aviation, avionics.

I. INTRODUCTION

In the United States aviation sector, the Federal Aviation Administration (FAA) is pushing a shift from secondary surveillance radar (SSR) interrogations to the more modern Automatic Dependent Surveillance-Broadcast (ADS-B) technology in air traffic control. As of January 2020, all aircraft operating in the continental United States are required to use ADS-B [1]. European aviation is following suit – the gradual shift to mandatory ADS-B broadcasting already started in June 2020 [2]. ADS-B offers many benefits over SSR, such as enhanced and fully automatic situational awareness of all aircraft and air traffic control (ATC) in the vicinity, increased system efficiency by eliminating interrogation processes, and cost-effective implementation. Moreover, FAA and its stakeholders are actively experimenting with ADS-B for

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Liu.

commercial space transportation applications [3]. Due to ADS-B's efficiency, light weight, and cost-efficient features, it is gaining popularity among all types of users. Using a portable ADS-B transceiver (e.g., SkyEcho2, Sentry, and echoUAT) as a mobile cockpit solution is very trendy nowadays, especially in the general aviation sector. Such portable ADS-B devices provide services through and electronic flight bag (EFB) application hosted on a mobile tablet or smartphone. ADS-B devices (e.g., SkyEcho2, Sentry, and echoUAT) exchange data mainly using the Garmin DataLink 90 (GDL-90) protocol, one of the de facto standard technologies that are leading in the avionics industry. GDL-90 is also used in many integrated flight deck (IFD) systems and electronic flight instrumentation systems (EFISs) such as Garmin's G1000, Avidyne's IFD440/540, and EX5000, as well as in many mobile cockpit devices and EFB applications (such as AvPlan, Naviator, and Airmate). Due to the wide use of GDL-90, any potential vulnerability in it poses elevated cybersecurity risks to avionics systems as well as safety risks to the passengers and crew lives. Researchers have reported several types of security threats involving ADS-B, such as ghost aircraft, aircraft disappearance, denial-of-service (DoS) [4]–[6]. However, protocol fuzzing in mobile cockpit systems has not been thoroughly investigated yet, which has motivated us to conduct this study. This study is important as it systematically addresses the discovery of potential bugs and cybersecurity vulnerabilities within GDL-90 implementations. Our main contributions with this work are as follows.

1) To the best of our knowledge, we are the first to propose, develop, and execute a systematic fuzzing platform and experiments aimed specifically at the GDL-90 protocol (although our method is easily extensible to more avionics and aerospace data-link protocols).
2) We are the first to discover and report safety-critical DoS vulnerabilities in a handful of the most popular aviation apps and mobile EFBs as a result from fuzzing the GDL-90 inputs.

The rest of this article is organized as follows. Different fuzzing aspects are discussed in Section II. In Section III, we introduce our attack strategy. We present the results in Section IV. We discuss related works in Section V. Finally, in Section VI, we discuss possible workarounds and future work as we conclude this paper.

II. BACKGROUND

In this section we briefly present background technologies and techniques used in our experiments.

A. FUZZING

Fuzzing (or fuzz testing) is an automated software testing method for finding implementation and input sanitization bugs by using intentionally malformed or randomized inputs. It was originally developed by Professor Barton Miller and his team of students at the University of Wisconsin Madison in 1989 [7]. With fuzzing, a generator is used to create random and semi-random (known to be dangerous) data usually sampled from real inputs. Such data are inputed in to the software being tested, and the software's behaviour is observed. Fuzzing is based on the premise that bugs exist in every program and therefore, a consistent and systematic approach will eventually cover them [8]. Fuzzing is a blind testing technique with caveats, such as the possibility of missed program paths due to the random nature of the input mutations [9]. In our experiments, we targeted the GDL-90 protocol, which means that we used protocol fuzzing by forging packets with a real protocol-like format but with some parts malformed. (This topic will be discussed further in Section III-D).

Flag Byte	Message ID	Message Data	CRC	Flag Byte

FIGURE 1. GDL-90 message format.

In this study, we used the American Fuzzy Lop (AFL) as our core fuzzing toolset. AFL is a security-oriented greybox fuzzer originally developed by Michal Zalewski [10]. It is a proven, easy-to-use, stable, and effective fuzzer that utilizes performance optimizations to decrease unnecessary runtime. It uses an instrumentation-guided genetic algorithm to fuzz the software being tested with brute force. In essence, AFL takes the user-supplied sample test cases one by one,trims them, and mutates the trimmed versions with traditional fuzzing strategies. The behavior of the software being tested is recorded, and interesting test cases are recorded for further use and for runtime modifications of the fuzzer [9]. AFL is currently maintained by Google Open Source and is licensed with Apache License 2.0 [9], [11].

B. GDL-90 PROTOCOL

The Garmin DataLink 90 (GDL-90) format is supported by many aviation hardware and software (see Table 3). It is described in the RTCA DO-267A standard as a messaging structure based on asynchronous high-level data link control (HDLC), with some modifications to better suit avionics data interfaces [12], [13]. The basic GDL-90 message format is presented in Figure 1.

The message starts with a Flag Byte ($0 \times 7E$), followed by a one-byte Message ID, which specifies the type of message being transmitted. The message type sets the message data content and length. All the message definitions are listed in Table 1.

A two-byte frame check sequence (16-bit CRC, LSB first) is calculated for the data and appended to the message, and the message ends with another flag byte. If a flag byte ($0 \times 7E$) or a control-escape character (CEC, $0 \times 7D$) is present in the original message, the message byte is XOR'd with 0×20, and a CEC is prefixed to it. Thus, the integrity of the message is preserved. The receiving end checks the incoming traffic for the Flag Bytes and captures the data between them. The captured data are inspected for CECs. If a CEC is found, it is

TABLE 1. GDL-90 message IDs.

Message ID	Message Name
0	Heartbeat
2	Initialization
7	Uplink Data
9	Height Above Terrain
10	Ownship Report
10	Ownship Geometric Altitude
20	Traffic Report
30	Basic Report
31	Long Report

discarded, and the byte after it is XOR'd again to return its old form properly. The CRC for the message data part of the full GDL-90 message is calculated and verified. If it is deemed valid, the message is ready for use. GDL-90 devices in operation transmit a heartbeat message once every second, followed by an ownship report. In between these "pulses" other messages such as traffic reports can be transmitted. In our experiments, we focused on three types of messages:

- Heartbeat messages,
- Traffic reports; and
- Ownship reports

A heartbeat message is used for the devices to indicate that they are operational and to submit information about their status. Two status bytes in the message tell information about the transmitter in Boolean fashion. This information includes "battery low," "Global Positioning System (GPS) fix," "maintenance requirement," etc. flags. A timestamp is also present in the message after the status bytes.

Traffic reports are at the output in each second for each proximate target. GDL-90 expects at least 32 simultaneous targets to be handled, but more can be processed depending on the uplink configurations and the interface baud rate. Traffic report data use 27 bytes to represent each needed attribute. Table 2 shows the fields of the traffic report data in order.

An ownship report message follows the traffic report format. It is always in the output even without a proper GPS fix. It broadcasts the transmitter information to the network.

C. GDL-90 PROTOCOL EXTENSIONS

Some vendors have their own interpretation of the protocol outside of the Garmin standard. For example, Uavionix's SkyEcho2 mainly uses the standard messaging types, but it outputs its ownship message with the message type code 101. On the other hand, ForeFlight's Sentry extends the protocol and does not communicate with the standard message types. Sentry transmits messages with IDs 37 and 38, which are longer than the standard heartbeat, ownship, and traffic messages and most likely contain multiple message types

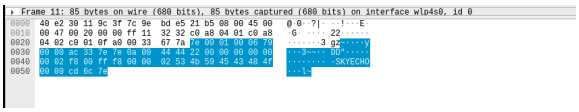


FIGURE 2. Heartbeat messages of SkyEcho2 proprietary GDL-90 extension as captured and decoded by wireshark software.

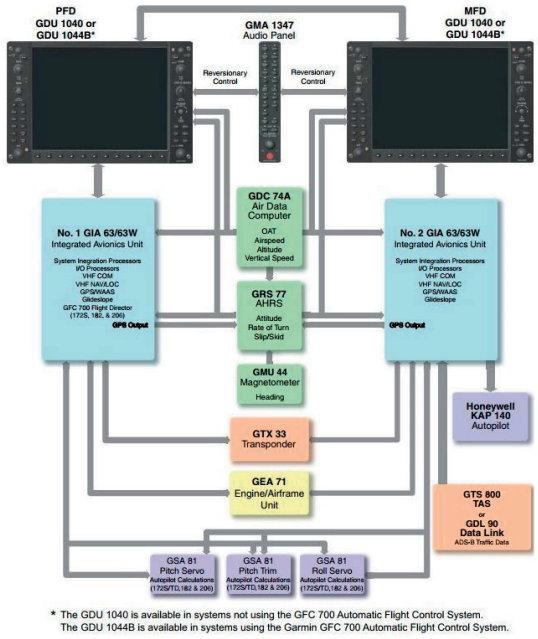


FIGURE 3. System diagram of Garmin G1000 EFIS/IFD [15]. Note the GDL-90 inputs going into No. 2 GIA 63/63W that, in turn, controls the auto-pilot Honeywell KAP 140 [14].

in a single packet. The ForeFlight EFB supports both devices. It broadcasts messages to the network. When the app is accepting traffic, it sends *"i-want-to-play-ffm-udp"*; and when it goes to sleep it sends *"i-cannot-play-ffm-udp."* It also identifies itself to the network by broadcasting *"App: ForeFlight, GDL90: port:4000"* messages. For our experiments, we did not delve deeper into the ForeFlight protocol as it was not necessary. We were able to capture, modify, resend, and receive Sentry packets just like with the other devices. Thus, the integration with AFL was quite straightforward. Figure 2 shows a Skyecho-ecoded heartbeat packet in Wireshark.

Figure 3 depicts the system diagram of Garmin G1000 – a real-world EFIS/IFD/avionics system. It is important to note that GDL-90 inputs go to the GIA 63/63W avionics unit that is also *directly controlling the auto-pilot systems* such as Bendix/King KAP-140 [14]. Therefore, any GDL-90 vulnerabilities present within the avionics units *have a potential direct effect on the auto-pilot systems*. Therefore, it is important to discover such GDL-90 (and other data-link protocol) vulnerabilities as fast and as efficiently as possible, for example, using our approach and results.

TABLE 2. GDL-90 traffic/ownship report fields.

Field	Description	Length (bits)
Traffic Alert Status	0: No alert, 1: Traffic alert for this target, 2–15: Reserved	4
Target Identity (type)	0: ADS-B with ICAO address, 1: ADS-B with self-assigned address, 2: TIS-B with ICAO address, 3: TIS-B with track file ID, 4: Surface Vehicle, 5: Ground station beacon, 6–15: Reserved	4
Participant Address	Unique address	24
Latitude and Longitude	Encoded range -180 to 180 degrees, resolution approximately 2.14577×10^{-5}	24×2
Altitude	Pressure altitude (referenced to 29.92 inches Hg), encoded using a 25-foot resolution, offset by 1,000 feet. 0xFFF is invalid/unavailable.	12
Miscellaneous Indicators	Bits 0 and 1: Additional information for the Track/Heading field, Bit 2: Report derived from ADS-B or extrapolated from the previous report, Bit 4: Air/ground state	4
Integrity and Accuracy	Integrity and accuracy of the traffic reported (in nautical mile thresholds)	4×2
Horizontal Velocity	Velocity in knots. Above 4094 knots, the value will hold at 0xFFE.	12
Vertical Velocity	Velocity in 64 feet per minute, +/- 32,578 feet per minute	12
Track/Heading	Weighted heading value, resolution 360/256 (approx. 1.4 degrees)	8
Emitter Category	Type of vehicle in use represented by an integer. Values 0–21 are in use and the rest are reserved.	8
Call Sign	8 ASCII characters (0–9, A–Z). The space is only used for padding in the end.	64
Emergency/Priority Code	0: No emergency, 1: General emergency, 2: Medical emergency, 3: Minimum fuel, 4: No communication, 5: Unlawful interference, 6: Downed aircraft, 7–15: Reserved	4
Reserved	Reserved for future use	4

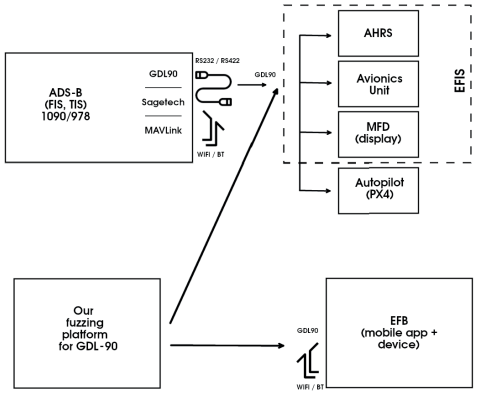


FIGURE 4. Overview of the GDL-90 test-bench and positioning of our fuzzing platform (for GDL-90 and similar avionics data-link protocols).

III. FUZZING ATTACKS ON GDL-90

A. DIAGRAMS OF OUR APPROACH

In Figure 4 we present a high-level diagram[1] of where GDL-90 outputs and inputs are connected in real-world systems and where our platform can be connected during the execution of GDL-90 fuzzing. It is important to note that discovering or triggering such protocol implementation vulnerabilities does not necessarily require physical or adjacent proximity. In another study of ours, we demonstrated that carefully crafted wireless ADS-B communications can be used to achieve the same goals, crash EFB/ADS-B apps or ADS-B avionics devices, which can be due to the GDL-90 or ADS-B vulnerabilities, or a handful of other reasons [5], [6]. This is possible because many ADS-B devices with an ADS-B IN function provide processed data using GDL-90 protocol encoding.

[1]This setup is part of a larger pentesting platform for aviation/avionics and maritime technologies [5], [6].

B. ADVANTAGES OF OUR APPROACH

Using the GDL-90 fuzzing approach that we developed and propose in this paper has the following main advantages:

1) Does not require aviation-spectrum wireless transmission (e.g., ADS-B) and thus, avoids any radio interference and lowers the costs, as SDR devices are not required (i.e., it works directly at the GDL-90 receiving point);
2) Is not limited to the capacity of radio channels and thus, can perform fuzzing/testing at considerably higher speeds (e.g., WiFi/ethernet has higher a throughput than the ADS-B RF link);
3) Works closer to the source of the possible GDL-90 implementation problems and thus, avoids the extra layer(s) introduced by higher protocols' (such as ADS-B's) processing chains, which could be sources of bottlenecks, false negatives/positives, and air-transmission regulatory challenges.

C. OVERALL HARDWARE-SOFTWARE SETUP

Our attacks were made simple by the fact that the common GDL-90 enables WiFi ADS-B devices (such as SkyEcho2, echoUAT, and Sentry) using connectionless UDP packets to send data. Therefore, we were able to easily capture, manipulate, and resend the packets to the applications without issues. First, we observed the packets transmitted in the WiFi networks created by the Sentry and SkyEcho2 with the Wireshark [16] network packet inspection tool. We applied the GDL-90 dissector [17] lua-script to Wireshark to identify and analyze the packets. We also transmitted ADS-B traffic messages via HackRFOne to the receivers. We copied the

required messages from the packet captures and saved them as samples for the fuzzer. Depending on the device and its configuration, we either left the different message types as separate samples or left them as one in the case of Sentry. In addition to the samples that we gathered from real device networks, we also utilized Eric Dey's GDL-90 code [18] to simulate Stratus [19] and SkyRadar [20] ADS-B receivers and created samples for those. In total, we tried four different samples with the applications. Some applications worked with only one sample-specific sample set. The simulated SkyRadar sample set was deemed the best generalization of the four, due to which it was the most widely used in our tests.

We were inspired by Eric Dey's GDL-90 code [18] and made our own GDL-90 sender script for fuzzing purposes. We chose AFL as our fuzzer of choice since we were adamant that the input coverage with AFL would be sufficient. We set up our environment as a Docker container with AFL and our sender/fuzzing script. With our sender script, the target IP address and the target port must be set at the beginning. When the parameters are set, we can start fuzzing. As we used UDP packets over WiFi, the applications at the mobile phone end were not aware that the device at the other end was not legitimate; therefore, the testing was realistic. However, as we had no feedback from the mobile device through the network to the fuzzer, we could not have AFL recording the exact input that made an app crash. We could only observe the applications. Running the fuzzer over the network with a packet sending delay made the fuzzing quite slow for AFL standards. However, the applications that were affected the most crashed within the first 60 minutes of the test. For the initial test, the target and the attacking PC were both connected to the same home network via a WiFi access point that ran OpenWRT 17.01.0 [21] or via an ethernet to a router.

Overall, our test setup works on the one-click-test principle. After the Docker container is built, a test can be started by running a script with four arguments: the IP address of the attacked device, the UDP port (4000 or 43211 in our tests), the sample folder (one of our four offerings), and the output folder (arbitrary and useful for resuming long fuzzing sessions). Logs are saved to the specified output folder. With the inclusion of Docker, the setup is easy, as each component is installed automatically. Figure 5 shows a status display during the test.

D. AFL SETUP

We used AFL's Python implementation (python-afl v.0.7.3) and the latest AFL as of date (afl-fuzz v.2.57b) in our tests. As our test setup was quite slow, we specified "quick and dirty" mode (-d option), which skips deterministic steps and usually yields faster results. This limited the depth that we could achieve with the tests; however, we discovered that this mode was perfectly adequate for many applications to falter. With the non-deterministic mode on and with the sample variety low, our longest (one-hour) fuzzing sessions reached

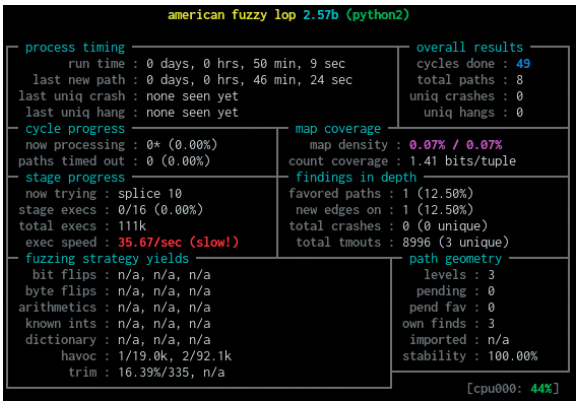


FIGURE 5. Example of an AFL run status.

TABLE 3. List of software exposed to fuzzing attacks ("software under test").

Name	Price	Version (Android / iOS)	Installations (Android / iOS)
OzRunways	Free	v.4.5.6 / v.10.1.9	50k+ / –
iFlightPlanner	Free	– / v.4.5.6	– / –
Airmate	Free	v.1.6.1 / v.2.3	50k+ / –
AvPlan	$76.16/year	v.1.3.28 / v.8.1.2	5k+ / –
Levil Aviation	Free	– / v.1.3	– / –
FLYQ EFB	Free	– / v.5.0	– / –
EasyVFR4	Free	v.4.0.870 / v.4.0.899	100+ / –
Horizon	Free	v.3.0 / v.3.0	10k+ / –
ForeFlight	Free	– / v.13.4	– / –
Pilots Atlas	$89.99/year	– / v.5.13.0	– / –
Xavion	Free	– / v.2.81	– / –
SkyDemon	$162/year	v.3.15.0 / v.3.15.3	100k+ / –
Stratus Insight	$99.99/year	– / v.5.17.3	– / –
Naviator	$34.99/year	v.4.2.2 / –	100k+ / –
Traffic (by Control-J) traffic	Free	v.0.0.2.4 / –	10+ / –
Eric Dey's GDL-90 gdl90etdey	Free	github repo commit d7e5936	–

at least 50 cycles. A cycle in AFL means that the fuzzer went through all the interesting test cases [22]. Therefore, we argue that the tests were quite thorough within the limitations of the samples we acquired. We observed that the crashes occurred at several stages of the fuzzing cycles. Even if the test applications did not crash, the usability of the data they presented was greatly hindered due to the malformed input data (see the details in the results in Section IV).

E. GDL-90 FUZZING TARGETS

In Table 3, we present a comprehensive list of the targeted software. We targeted mostly mobile EFB apps, but we also tested some open-source tools. For Eric Dey's GDL-90 code [18], we targeted only the decoding script.

F. HIGH-LEVEL GDL-90 ATTACK DESCRIPTION

A possible cybersecurity attack involving vulnerable GDL-90 implementations could look as follows:

1) *At the research time*: An exploitable GDL-90 vulnerability is first discovered (e.g., using our implementation-independent GDL-90 protocol fuzzing approach).
2) *At the design/manufacturing time*: An adversary designs and puts on the market an ADS-B-capable and GDL-90-compatible ''backdoored'' device that contains the GDL-90 exploitation payloads and attack vectors. The ''backdoor'' could be implemented at the hardware or at the firmware level in such a way to avoid the detection at the (re-)certification time (similar to the Volkswagen emission engine control unit manipulation scandal [23]).
3) *At the usage time*: The ''backdoored'' ADS-B-capable device sends or activates the GDL-90 exploitation payload. Such exploitation payloads could be activated conditionally, such as at certain altitudes, within certain geo-fence areas, and upon receiving a ''secret knock'' ADS-B message.
4) *At the usage time*: Alternatively, the discovered GDL-90 vulnerability can be reconstructed back to a specially crafted triggering ADS-B payload/message. Therefore, it may even be possible to trigger the GDL-90 vulnerability without ''backdoored'' hardware, by simply sending a specially crafted ADS-B payload/message.
5) Ultimately, backdoors have been shown to be implanted even in military-grade chips [24]. Therefore, it is more than reasonable to believe that backdoor implanting is also feasible for ADS-B devices destined for avionics/EFIS/IFD/EFB setups within commercial/general aviation and amateur aircraft.

IV. RESULTS

The fuzzing results are presented in Table 4. Of the 15 tested mobile EFB applications, 6 crashed (4 iOS-only and 2 iOS+Android) and 2 became unresponsive (1 iOS-only and 1 Android-only). In addition to mobile the EFB apps, Eric Dey's open-source GDL-90 [18] decoder experienced several dozen of unique crashes during a day-long fuzzing session on a normal PC (Linux). We focused only on fuzzing Eric Dey's GDL-90 decoder, leaving its network component out of the equation. The unique errors and crashes that we recorded were related to the different inputs that generated Python assertion statement failures which, in turn, were due to the faulty lengths of the messages. (Finding such issues is exactly the aim of fuzzing tests in general.) These results allow us to assume that Eric Dey's open-source GDL-90 [18] could pose stability, availability, and DoS-resiliency issues if deployed or operated ''as-is'' in real-world systems and devices.

In one of our recent works [5], [6], we tested almost the same set of mobile apps and devices for DoS attacks via the ADS-B layer and found that 6 of the mobile apps in Table 4 were impacted by the ADS-B IN DoS attack, which possibly affected over 200,000 mobile application installs worldwide. In [5], [6], we tested a total of 68 different ADS-B configurations (mobile and non-mobile) for the ADS-B IN DoS attack. We managed to crash 25% of them mostly within 2 minutes, while overall, the DoS attack impacted 51.47% of the tested configurations. In comparison, the fuzzing results presented in this paper have similarly worrying results in terms of aviation safety and lack of resiliency to cybersecurity attacks such as DoS. Attacks on core subsystem availability (such as DoS) pose high risks to safety-critical and mission-critical systems such as avionics and aerospace.

TABLE 4. Details of the mobile applications (apps) considered ''attacked software''.

App Name	GDL-90: Android	GDL-90: iOS	Comparison with our ADS-B-level DoS attacks [5], [6]
OzRunways	CRA (once)	CRA	CRA
Stratus Insight	NA-P	CRA	CRA
iFlightPlanner	NA-P	CRA	DNW
AirMate	DNW	CRA	CRA
AvPlan	CRA / UNR	CRA	CRA
Levil Aviation	NA-P	CRA	DNT
FlyQ EFB	NA-P	UNR	DNC
EasyVFR4	DNC	DNC	DNC
Horizon	DNC	DNC	DNT
ForeFlight	NA-P	DNC	CRA
Pilots Atlas	NA-P	DNC	DNC
Xavion	NA-P	DNC	DNT
Traffic (by Control-J)	DNC	NA-P	DNT
Naviator	UNR	NA-D	DNW
SkyDemon	DNC	DNC	DNW

Android = Samsung Galaxy A21s, Android v 11
iOS = Apple iPhone SE, iOS v 13.3
Acronyms: CRA = Crash; UNR = Unresponsive/Hang; DNC = Did Not Crash; DNT = Did Not Test; DNW = Did Not Work (e.g., did not connect to hardware, did not receive data); NA-G = Not Available for this Geography/Country/Region; NA-P = Not Available for this Platform; NA-D = Not Available for this Device.

A. VISUAL OBSERVATIONS

All the mobile application crashes were observed by visually inspecting the device and software under test. The crashes happened either by themselves or while trying to operate the software (e.g., after any touch input, movement of the map, or zooming in/out) while the test was running. For each tested configuration that was impacted, the crashes were observed and confirmed at least three times (unless noted otherwise) before the result was registered.

Although FlyQ did not crash, it became unresponsive and had to be closed by the user. OzRunways on Android crashed, but the result was not consistently repeated. The Naviator app on Android did not crash during the test. However, it consistently closed the GDL-90 ADS-B input on an error state in each of our attempts and recovered only after a restart. Otherwise, the application remained functional. Most of the test applications showed some abnormal behavior,

such as an irrationally flinching map screen, fluctuating GPS data (due to the GPS positioning taken from the GDL-90 messages), alerts (due to plane proximity or altitude readings), and other non-standard or device operator-alerting behavior. Therefore, the apps marked with DNCs (Did Not Crash) in Table 4 should not be considered conclusively stable [25]. The applications that did not crash in our tests in this study may crash with some other sample data or testing methods.

V. RELATED WORKS

A. SOFTWARE FUZZING

Reliable and efficient aerial communication is at the heart of aerospace safety. Any defects in this safety-critical technology may cost human lives and property. However, modern protocols and the accompanying software are not always up to the task. Several studies have shown numerous viable attacks on these protocols and software [4], [26]. Developers, researchers, and hackers are using many tools to find out the security vulnerabilities of this kind of mission-critical system. Here, we discuss a few of them.

The success stories and the open-source nature of AFL have encouraged researchers to customize this fuzzer for different tasks [10]. Numerous studies have added many functionalities to the AFL (e.g., pathfinding, sample creation, and coverage) to improve its performance and effectiveness [27]–[35]. As a result, AFL has been added to commercial off-the-shelf (COTS) binaries [36]–[38]. It has also received modifications for its parallel-run capabilities [39].

B. ATTACKS AND FUZZING ON AVIONICS DATA-LINK

The Micro Air Vehicle Link (MAVLink) communication protocol is a bidirectional communication protocol that is used in drones and ground control stations. It offers different types of messages that can be transmitted reliably in an efficient package [40]. However, Domin *et al.* reported a crash of MAVLink-capable software in their protocol fuzzing tests in 2016 [41]. They were able to crash a virtual drone with a random payload by incrementally increasing the payload bytes from 1 to 255, thus increasing the length of the whole message. An open-source MAVLink fuzzing software is available [42].

PX4 is a widely avaible and extremely popular flight controller that also supports the MAVLink protocol as well as data from ADS-B IN-capable devices (such as Aerobits AERO and uAvionix pingRX). Alias Robotics [43] presented a general cybersecurity overview of PX4 from threat modeling and static analysis perspectives and, in this context, introduced the Robot Vulnerability Database (RVD). Subsequently, Jang *et al.* [44] performed a thorough static analysis of various PX4 firmware codebases.

Other communication protocols are also used for drones in particular. Rudo and Zeng [45] showed fuzzing results on the file transfer protocol/session initiation protocol (FTP/SIP) and session description protocol (SDP) embedded in consumer-grade drones. They raised concerns about the state of security of such drones with commercial drone software. They demonstrated GPS navigation and other subsystem failures (e.g., video feed and motor issues). Multiple studies have shown that the Internet-of-Things (IoT) and embedded devices are quite vulnerable [46], [47].

With regad to drone security issues, Kim *et al.* published their robotic vehicle (RV) fuzzing tool called "RVFuzzer" [48]. This tool was designed to highlight missing or faulty validation checks for control inputs. These bugs and missing features may cause physical disruptions, such as mission failures or crashes, on RVs, such as drones, if exploited. The authors constructed the RVFuzzer to employ three distinct strategies for searching input validation bugs, such as control parameter mutation, one-dimensional mutation, and multidimensional mutation. Throughout their evaluation, they discovered 89 input validation bugs from two control programs. Since the attacks do not require any code injection or other invasive procedures, they cannot be detected by security solutions [48]. Hence more specific code improvements and internal security audits for source codes under development are required.

C. ATTACKS ON AVIONICS SYSTEMS AND PROTOCOLS

The research community has adamantly scrutinized the security of ADS-B communication over the years. In 2004, Korzel *et al.* [49] demonstrated issues with the data integrity of the protocol due to erroneous inputs and data dropouts. Further concerns over the authenticity, security, confidentiality, and integrity of such protocol have been periodically raised since [50]–[52].

Several researcher have frequently demonstrated attacks against the ADS-B protocol. Costin and Francillon [4] conducted the first practical ADS-B message injection and spoofing attacks. Schäfer *et al.* [26] exposed several attacks such as ghost aircraft attacks and virtual trajectory modification on budget devices. Sjödin and Gruneau [53] used HackRF SDR to demonstrate data injection and flooding attacks on the Sentry ADS-B transceiver. They concluded that the device does not validate the messages from the ADS-B protocol. McCallie *et al.* [54] classified such attacks and explored their consequences, which resulted in worrying results.

Portable ADS-B transceivers (e.g., SkyEcho2, Sentry, and echoUAT), which are operated with iPads and other tablets, are favored by many general aviation pilots due to their ease of setup, ease of use, and affordable pricing. As these devices are not part of the onboard avionics per se, Lundberg *et al.* [55], [56] pointed out that they do not, nor do they need to, meet the standards of traditional avionics (e.g., RTCA, ARINC, and EUROCAE). The authors also found vulnerabilities on all of their test samples and recommended further product improvements to the device and software designers.

VI. CONCLUSION

In this paper, we presented our study of the impact of GDL-90 protocol fuzzing on a range of popular mobile EFBs and some standard PC software. Our results showed a worrying lack of security in many EFB applications where the security is directly related to aircraft's safety navigation. Of the 16 configurations that we tested herein, our avionics pentesting platform managed to crash or otherwise impact 9 configurations (56%). During the fuzzing test, we observed crashes, hangs, and other abnormal behaviors of the EFB apps and GDL-90 decoders. The consistency of our test results on a heterogeneous and representative set of EFBs (on the Android, iOS, and Linux platforms) indicates the reliability of our approach and results. DoS attacks can be devastating for mission-critical systems such as in avionics and aerospace, where the availability and reliability of the system are crucial. However, we hope that our results and presented methodology can motivate the standardization and regulatory bodies, as well as the industry and air traffic organizations, to improve the requirements for and the implementation checks of avionics devices and apps with regard to resiliency to cybersecurity attacks, and in particular, resiliency to DoS attacks. To ensure the adequate safety of such mission-critical systems, multidimensional security measures need to be taken. For avionics devices and related software/firmware, upgrading their defence against cyberattacks should be considered a continuous process, and thus, related research and development need to be sustained along with the operation of such devices and technologies.

ACKNOWLEDGMENT

All results, views, and opinions presented herein are only those of the authors and do not reflect the official position of the European Union and its organizations and projects, including the Horizon 2020 Program and Engage KTN. The authors thank Dr. Andrei Costin for facilitating and managing the grant.

The work of Hannu Turtiainen also acknowledges and thanks the Finnish Cultural Foundation / Suomen Kulttuurirahasto (https://skr.fi/en) for supporting his Ph.D. dissertation work and research and the Faculty of Information Technology of JYU, in particular, Prof. Timo Hämäläinen, for partly supporting and supervising his Ph.D. work at JYU in 2021–2022.

REFERENCES

[1] *No Kidding: ADS-B Deadline of Jan. 1, 2020, is Firm*. Accessed: Jun. 11, 2021. [Online]. Available: https://www.faa.gov/news/updates/?newsId=90008

[2] EASA. (2018). *Easa Seasonal Technical Commission*. Accessed: Mar. 2, 2021. [Online]. Available: https://www.easa.europa.eu/sites/default/files/dfu/EASA_STC_NEWS_JUNE_2018.pdf

[3] N. Demidovich, "Federal aviation administration incremental flight testing of automatic dependent surveillance-broadcast (ADS-B) prototype for commercial space transportation applications," in *Proc. ITEA 32nd Annu. Int. Test Eval. Symp.* Washington, DC, USA: Federal Aviation Administration, Aug. 2015.

[4] A. Costin and A. Francillon, "Ghost in the air (Traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices," in *Proc. Black Hat USA*, 2012, pp. 1–12.

[5] S. Khandker, H. Turtiainen, A. Costin, and T. Hamalainen, "Cybersecurity attacks on software logic and error handling within ADS-B implementations: Systematic testing of resilience and countermeasures," *IEEE Trans. Aerosp. Electron. Syst.*, early access, Dec. 31, 2021, doi: 10.1109/TAES.2021.3139559.

[6] S. Khandker, H. Turtiainen, and A. Costin, "Practical denial-of-service and combined high-level attacks on real-world ADS-B, ATC, ATM hardware and software," 2021.

[7] B. Miller. *Fuzz Testing of Application Reliability*. Accessed: May 25, 2021. [Online]. Available: http://pages.cs.wisc.edu/~bart/fuzz/

[8] OWASP. *Fuzzing*. Accessed: Jun. 30, 2021. [Online]. Available: https://owasp.org/www-community/Fuzzing

[9] G. O. Source. *Github: Afl*. Accessed: Jun. 30, 2021. [Online]. Available: https://github.com/google/AFL

[10] M. Zalewski. *American Fuzzy Lop*. Accessed: Jun. 30, 2021. [Online]. Available: https://lcamtuf.coredump.cx/afl/

[11] A. S. Foundation. *Apache License, Version 2.0*. Accessed: Jul. 1, 2021. [Online]. Available: https://www.apache.org/licenses/LICENSE-2.0

[12] *RTCA DO-267: Minimum Aviation System Performance Standards (MASPS) for Flight Information Services-Broadcast (FIS-B) Data Link*, RTCA, Washington, DC, USA, 2014.

[13] *GDL 90 Data Interface Specification*, Garmin, Olathe, KS, USA, 2007.

[14] Bendix/King. (2021). *KAP 140 Autopilot System*. [Online]. Available: https://www.bendixking.com/content/dam/bendixking/en/documents/document-lists/downloads-and-manuals/006-18034-0000-KAP-140-Pilots-Guide.pdf

[15] Garmin. (2021). *G1000 System*. [Online]. Available: https://buy.garmin.com/en-U.S./U.S./p/6420

[16] *Wireshark Homepage*. Accessed: May 25, 2021. [Online]. Available: https://www.wireshark.org/

[17] B. Kyser. *Github: Gdl90Dissector*. Accessed: May 25, 2021. [Online]. Available: https://github.com/BrantKyser/gdl90Dissector

[18] E. Dey. *Github: Gdl90*. Accessed: May 5, 2021. [Online]. Available: https://github.com/etdey/gdl90

[19] *Stratus ADS-B Receiver*. Accessed: May 5, 2021. [Online]. Available: https://stratusbyappareo.com/products/stratus-ads-b-receivers/

[20] SkyRadar Radenna LLC. *SkyRadar ADS-B Receiver*. Accessed: May 5, 2021. [Online]. Available: https://www.skyradar.net/skyscope-receiver/receiveroverview.html

[21] OpenWrt. *OpenWrt Project*. Accessed: Jul. 1, 2021. [Online]. Available: https://openwrt.org/

[22] Google. *AFL User Guide*. Accessed: Jul. 1, 2021. [Online]. Available: https://afl-1.readthedocs.io/en/latest/user_guide.html

[23] M. Contag, G. Li, A. Pawlowski, F. Domke, K. Levchenko, T. Holz, and S. Savage, "How they did it: An analysis of emission defeat devices in modern automobiles," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 231–250.

[24] S. Skorobogatov and C. Woods, "Breakthrough silicon scanning discovers backdoor in military chip," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* London, U.K.: Quo Vadis Labs, 2012, pp. 23–40.

[25] M. Muench, J. Stijohann, F. Kargl, A. Francillon, and D. Balzarotti, "What you corrupt is not what you crash: Challenges in fuzzing embedded devices," in *Proc. NDSS*, 2018, pp. 1–15.

[26] M. Schäfer, V. Lenders, and I. Martinovic, "Experimental analysis of attacks on next generation air traffic communication," in *Applied Cryptography and Network Security*, M. Jacobson, M. Locasto, P. Mohassel, and R. Safavi-Naini, Eds. Berlin, Germany: Springer, 2013, pp. 253–271.

[27] C. Lemieux and K. Sen, "FairFuzz: A targeted mutation strategy for increasing greybox fuzz testing coverage," in *Proc. 33rd ACM/IEEE Int. Conf. Automated Softw. Eng.*, Sep. 2018, pp. 475–485.

[28] N. Nichols, M. Raugas, R. Jasper, and N. Hilliard, "Faster fuzzing: Reinitialization with deep neural models," 2017, *arXiv:1711.02807*.

[29] K. Patil and A. Kanade, "Greybox fuzzing as a contextual bandits problem," 2018, *arXiv:1806.03806*.

[30] R. K. Prakash, I. Vasudevan, I. Indhuja, T. Thangarasan, and C. Krishnan, "Hardiness sensing for susceptibility using American fuzzy lop," in *Proc. ITM Web Conf.*, vol. 37, 2021, pp. 1–4.

[31] M. Rajpal, W. Blum, and R. Singh, "Not all bytes are equal: Neural byte sieve for fuzzing," 2017, *arXiv:1711.04596*.

[32] L. Sun, X. Li, H. Qu, and X. Zhang, "AFLTurbo: Speed up path discovery for greybox fuzzing," in *Proc. IEEE 31st Int. Symp. Softw. Rel. Eng. (ISSRE)*, Oct. 2020, pp. 81–91.

[33] J. Wang, B. Chen, L. Wei, and Y. Liu, "Superion: Grammar-aware greybox fuzzing," in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng. (ICSE)*, 2019, pp. 724–735.
[34] X. Yuan, L. Pan, and S. Luo, "Binary fuzz testing method based on LSTM," *IOP Conf., Mater. Sci. Eng.*, vol. 612, Oct. 2019, Art. no. 032192.
[35] G. Zhang and X. Zhou, "AFL extended with test case prioritization techniques," *Int. J. Model. Optim.*, vol. 8, no. 1, pp. 41–45, Feb. 2018.
[36] Y. Chen, D. Mu, J. Xu, Z. Sun, W. Shen, X. Xing, L. Lu, and B. Mao, "Ptrix: Efficient hardware-assisted fuzzing for COTS binary," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2019, pp. 633–645.
[37] S. Dinesh, N. Burow, D. Xu, and M. Payer, "RetroWrite: Statically instrumenting COTS binaries for fuzzing and sanitization," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2020, pp. 1497–1511.
[38] Y. Zheng, A. Davanian, H. Yin, C. Song, H. Zhu, and L. Sun, "FIRM-AFL: High-throughput greybox fuzzing of iot firmware via augmented process emulation," in *Proc. 28th Secur. Symp.*, 2019, pp. 1099–1114.
[39] J. Ye, B. Zhang, R. Li, C. Feng, and C. Tang, "Program state sensitive parallel fuzzing for real world software," *IEEE Access*, vol. 7, pp. 42557–42564, 2019.
[40] (2021). *Mavlink Developer Guide*. [Online]. Available: https://mavlink.io/en/
[41] K. Domin, I. Symeonidis, and E. Marin, "Security analysis of the drone communication protocol: Fuzzing the MAVLink protocol," in *Proc. ORBIlu*, 2016, pp. 1–7.
[42] Auterion. (2019). *Github: MAVLink Fuzz Testing*. [Online]. Available: https://github.com/Auterion/mavlink-fuzz-testing
[43] Alias Robotics. *The Cybersecurity Status of PX4*. Accessed: Jul. 1, 2021. [Online]. Available: https://aliasrobotics.com/files/cybersecurity_status_PX4.pdf
[44] J.-H. Jang, Y.-S. Kang, and J.-H. Lee, "Static analysis and improvement opportunities for open source of UAV flight control software," *J. Korean Soc. Aeronaut. Space Sci.*, vol. 49, no. 6, pp. 473–480, Jun. 2021.
[45] D. Rudo and D. Kai Zeng, "Consumer UAV cybersecurity vulnerability assessment using fuzzing tests," 2020, *arXiv:2008.03621*.
[46] A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti, "A large-scale analysis of the security of embedded firmwares," in *Proc. 23rd USENIX Secur. Symp.*, 2014, pp. 1–22.
[47] A. Costin, A. Zarras, and A. Francillon, "Automated dynamic firmware analysis at scale: A case study on embedded web interfaces," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, May 2016, pp. 437–448.
[48] T. Kim, C. H. Kim, J. Rhee, F. Fei, Z. Tu, G. Walkup, X. Zhang, X. Deng, and D. Xu, "RVFuzzer: Finding input validation bugs in robotic vehicles through control-guided testing," in *Proc. 28th Secur. Symp.*, 2019, pp. 425–442.
[49] J. Krozel, D. Andrisani, M. Ayoubi, T. Hoshizaki, and C. Schwalm, "Aircraft ADS-B data integrity check," in *Proc. 4th Aviation Technol., Integr. Oper. (ATIO) Forum*, 2004, p. 6263.
[50] K. Samuelson, E. Valovage, and D. Hall, "Enhanced ADS-B research," in *Proc. IEEE Aerosp. Conf.*, Oct. 2006, pp. 1–7.
[51] R. G. Wood, "A security risk analysis of the data communications network proposed in the nextgen air traffic control system," Ph.D. dissertation, Dept. Inf. Comput. Sci., Oklahoma State Univ., Stillwater, OK, USA, 2009. [Online]. Available: https://search.proquest.com/dissertations-theses/security-risk-analysis-data-communications/docview/305083310/se-2?accountid=11774
[52] L. Purton, H. Abbass, and S. Alam, "Identification of ADS-B system vulnerabilities and threats," *Proc. 33rd Australas. Transp. Res. Forum (ATRF)*, 2010, pp. 1–16.
[53] A. Sjödin and M. Gruneau, "The ADS-B protocol and its' weaknesses: Exploring potential attack vectors," KTH Skolan för Elektroteknik och Datavetenskap, Stockholm, Sweden, Tech. Rep., Jun. 2020.
[54] D. McCallie, J. Butts, and R. Mills, "Security analysis of the ADS-B implementation in the next generation air transportation system," *Int. J. Crit. Infrastruct. Protection*, vol. 4, no. 2, pp. 78–87, Aug. 2011.
[55] D. Lundberg, B. Farinholt, E. Sullivan, R. Mast, S. Checkoway, S. Savage, A. C. Snoeren, and K. Levchenko, "On the security of mobile cockpit information systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 633–645.
[56] D. A. Lundberg, "Security of ADS-B receivers," Ph.D. dissertation, Dept. Comput. Sci. Eng., UC San Diego, San Diego, CA, USA, 2014.

HANNU TURTIAINEN received the B.Sc. degree in electronics engineering from the University of Applied Sciences, Jyväskylä, Finland, and the M.Sc. degree in cybersecurity, in 2020. He is currently pursuing the Ph.D. degree in software and communication technology with the University of Jyväskylä, Finland. He is also working in the IoT field as a Cybersecurity and Software Engineer with Binare.io, a deep-tech cybersecurity spin-off from the University of Jyväskylä. His research interests include machine learning and artificial intelligence in the cybersecurity and digital privacy.

ANDREI COSTIN received the Ph.D. degree from the EURECOM/Telecom ParisTech, under co-supervision of Prof. Francilon and Prof. Balzarotti, in 2015. He is currently a Senior Lecturer/Assistant Professor of cybersecurity at the University of Jyväskylä (Central Finland), with a particular focus on the IoT/firmware cybersecurity and digital privacy. He is also the CEO/Co-Founder of Binare.io, a deep-tech cybersecurity spin-off from the University of Jyväskylä, focused on innovation and tech-transfer related to the IoT cybersecurity. He has been publishing and presenting at more than 45 top international cybersecurity venues, both academic (Usenix Security and ACM ASIACCS) and industrial (BalckHat, CCC, and HackInTheBox). He is the author of the first practical ADS-B attacks (BlackHat 2012) and has literally established the large-scale automated firmware analysis research areas (Usenix Security 2014)-these two works are considered seminal in their respective areas, being also most cited at the same time.

SYED KHANDKER received the M.Sc. degrees in web intelligence and service engineering from the University of Jyväskylä, Finland, in 2016. He is currently pursuing the Ph.D. degree with the Faculty of Information Technology, University of Jyväskylä. Since his childhood, he has been a Radio Enthusiast and holds an Amateur Radio Operator License. He has authored four journal and conference publications. His research interests include RF fingerprint positioning, automatic dependent surveillance-broadcast, automatic identification system, wireless communications, and artificial intelligence.

TIMO HÄMÄLÄINEN has over 25 years of research and teaching experience related to computer networks. He has lead tens of external funded network management related projects. He has launched and leads master's programs with the University of Jyväskylä (currently SW and communication engineering) and teaches network management related courses. He has more than 200 internationally peer-reviewed publications and he has supervised 36 Ph.D. theses. His current research interests include wireless/wired network resource management (the IoT, SDN, and NFV) and network security.

PVII

ON THE (IN)SECURITY OF 1090ES AND UAT978 MOBILE COCKPIT INFORMATION SYSTEMS – AN ATTACKER PERSPECTIVE ON THE AVAILABILITY OF ADS-B SAFETY AND MISSION-CRITICAL SYSTEMS

by

S Khandker, H Turtiainen, A Costin, T Hamalainen 2022

IEEE Access, 10, 37718-37730

https://doi.org/10.1109/ACCESS.2022.3164704

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 20, 2022, accepted March 21, 2022, date of publication April 4, 2022, date of current version April 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3164704

On the (In)Security of 1090ES and UAT978 Mobile Cockpit Information Systems–An Attacker Perspective on the Availability of ADS-B Safety- and Mission-Critical Systems

SYED KHANDKER, HANNU TURTIAINEN, ANDREI COSTIN, AND TIMO HÄMÄLÄINEN
Faculty of Information Technology, University of Jyväskylä, 40014 Jyväskylä, Finland

Corresponding author: Andrei Costin (ancostin@jyu.fi)

This work was supported in part by the Engage Consortium's Knowledge Transfer Network (KTN) Project 204, in part by the Finnish Grid and Cloud Infrastructure—FGCI (persistent identifier urn:nbn:fi:research-infras-2016072533), in part by the Research Dean for Research through the Faculty of Information Technology of the University of Jyväskylä, and in part by the Finnish Cultural Foundation under Grant 00221059.

ABSTRACT Automatic dependent surveillance-broadcast (ADS-B) is a key air surveillance technology and a critical component of next-generation air transportation systems. It significantly simplifies aircraft surveillance technology and improves airborne traffic situational awareness. Many types of mobile cockpit information systems (MCISs) are based on ADS-B technology. MCIS gives pilots the flight and traffic-related information they need. MCIS has two parts: an ADS-B transceiver and an electronic flight bag (EFB) application. The ADS-B transceivers transmit and receive the ADS-B radio signals while the EFB applications hosted on mobile phones display the data. Because they are cheap, lightweight, and easy to install, MCISs became very popular. However, due to the lack of basic security measures, ADS-B technology is vulnerable to cyberattacks, which makes the MCIS inherently exposed to attacks. Attacks are even more likely for the MCIS, because they are power, memory, and computationally constrained. This study explores the cybersecurity posture of various MCIS setups for both types of ADS-B technology: 1090ES and UAT978. Total six portable MCIS devices and 21 EFB applications were tested against radio-link-based attacks by transmission-capable software-defined radio (SDR). Packet-level denial of service (DoS) attacks affected approximately 63% and 37% of 1090ES and UAT978 setups, respectively, while many of them experienced a *system crash*. Our experiments show that DoS attacks on the reception could meaningfully reduce transmission capacity. Our coordinated attack and fuzz tests also reported worrying issues on the MCIS. The consistency of our results on a very broad range of hardware and software configurations indicate the reliability of our proposed methodology as well as the effectiveness and efficiency of our platform.

INDEX TERMS Cybersecurity, attacks, ADS-B, ATC, ATM, UAT978, 1090ES, availability, DoS.

I. INTRODUCTION

THE demand for air transportation has been steadily increasing over the last few decades. The Federal Aviation Administration (FAA) predicts that the number of passengers in commercial aviation will increase to 1.15 billion by 2033 [1]. On the other side of the Atlantic, Eurocontrol predicts 1.6 billion air passengers in its sky per year by the early 2030s [2]. In addition, air cargo transportation, military aircraft, and unmanned aerial vehicles are expected to boost air traffic in the coming years. As a result, the number of aircraft in the airspace will continue to increase, and the airspace will become even more crowded. For reasons such as the safety of navigation, increased airspace capacity, improved flight safety, and future navigation needs, in 2004 the FAA initiated the Next Generation Air Transportation System (NextGen) project. NextGen focuses on the modernization of America's air transportation system to make flying even safer, more efficient, and predictable. One of its aspirations is to gradually transform the current obsolete

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenbao Liu.

and imprecise radar-based air traffic control (ATC) and air traffic management (ATM) systems into a fully digital and satellite-based navigation system. To implement this, the FAA chose the Automatic dependent surveillance-broadcast (ADS-B) system to be a core part of future air navigation technology in the US. In 2011, the EU also mandated a gradual ADS-B requirement starting in June 2020 [3]. The core idea of ADS-B is to periodically broadcast the position and other flight-related information of an aircraft to the ATC and other aircraft in the vicinity via radio frequency (RF) data link. The ADS-B communication system's construction and maintenance costs are expected to be only one-tenth of radar-based navigation [4]. This simplified air navigation technology is gaining popularity all over the world.

Light weight yet effective ADS-B technology is easy to adapt and use. For example, the ADS-B transceiver and smartphone-based mobile cockpit information system (MCIS) is very trendy in the general aviation (GA) sector. In this system, a small ADS-B transceiver is connected to a smartphone or other smart device that displays the navigation data to the pilot through an electronic flight bag (EFB) application. It also transmits global navigation satellite system (GNSS) location, flight information, and other useful information via the ADS-B antenna. MCIS setups cost around 500–1000 dollars. The affordable price and the ease of installation make such setups attractive to pilots of private planes.

Studies show that firmware vulnerabilities are quite common in Internet-of-Things (IoT) and embedded devices [5], [6] and this is also the case for ADS-B technology. The main reason for this insecurity is that ADS-B does not utilize basic security measures such as authentication and encryption. There have been many reports of ADS-B exploitation in the industry [7], [8] and in academia [9]–[12]; therefore, MCISs can be labelled inherently insecure. Even though many studies investigated the security of ADS-B, the security assessment of MCIS remains particularly under-researched. Compared to the powerful transponder or desktop setups, these power-, memory-, and computationally constraint mobile setups could be more vulnerable against cyberattacks. Nonetheless, the use of mobile setups is increasing rapidly. The 21 EFB applications used in this study were downloaded more than 650,000 times from the Google play store, leaving alone other non-tested applications and iOS platform's download numbers aside. Assessing the security of such safety- and mission-critical systems against modern cyberattacks has motivated us to conduct this research. Our main contributions with this work are:

1) We present a systematic and comprehensive study of the (in)security of different commercial-grade MCISs.
2) We test the impacts of the attacks on a large number of EFB applications.
3) To the best of our knowledge, we implement and demonstrate the first-ever ADS-B attacks over UAT978.
4) We demonstrate that the UAT978 and 1090ES implementations are comparably vulnerable to generic and available cyberattacks.

The rest of this article is organized as follows: Section II introduces the relevant background on ADS-B and MCIS. Related studies are discussed in Section III. Details of our test platform and experiment setup are presented in Section IV. Attacks on MCIS are explained in Section V. Attack results are evaluated in Section VI. We discuss some solutions in Section VII. Finally, with Section VIII we conclude this article.

II. BACKGROUND

Modern aviation has relied only on primary surveillance radar (PSR) for a long time. With PSR, the position of the aircraft is measured by the distance and the angle to the radar, but the identity of the aircraft remains unknown. For this purpose, secondary surveillance radar (SSR) was developed. SSR transmits interrogation pulses using RF signals, which are known as Mode A and Mode C. These pulses allow the SSR to continuously interrogate the identity and the barometric altitude of an aircraft. However, the SSR systems have reached the limit of their operational capability. Mode A communication is limited to 4096 unique codes, which poses an issue for very busy modern air transportation. Therefore, a more advanced aircraft communication protocol is needed. Mode S was designed to solve these problems. Mode S is an SSR process that allows selective interrogation of aircraft according to an aircraft's unique 24-bit code called the International Civil Aviation Organization (ICAO or ICAO24) address. Based on Mode S, ADS-B's concept was evolved and it is now considered the future replacement of SSR.

ADS-B is a surveillance technique that relies on aircraft broadcasting their identity, position, and other information derived from onboard systems periodically without the need for interrogation. Besides the ground station, other aircraft also can receive the broadcast to have situational awareness and self-separation. The most important part of the ADS-B is position information, which is determined by GNSS. There are two main functionalities in ADS-B: ADS-B IN and ADS-B OUT. ADS-B IN refers to receiving, processing, and displaying the ADS-B signals from the ATC, aircraft, and other ADS-B OUT-equipped vehicles. ADS-B OUT refers to transmitting an aircraft's position, identity, velocity, and additional flight-related information. For data transmission, two datalink solutions are used as the physical layer for the ADS-B: 1090 MHz Extended Squitter (1090ES) and Universal Access Transceiver at 978 MHz (UAT978). Figure 1 depicts the ADS-B protocol in SSR.

A. 1090ES

1090ES uses the 1090 MHz radio frequency to transmit ADS-B OUT via a Mode-S transponder. Squitter refers to a burst or broadcast of aircraft-tracking data transmitted periodically by a Mode S transponder without interrogation

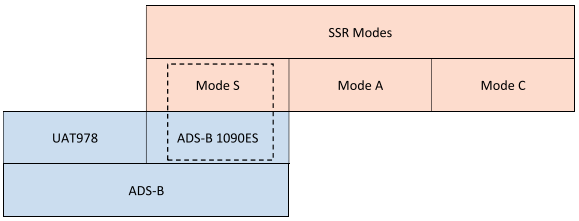


FIGURE 1. ADS-B protocol within SSR.

from the controller's radar. There are two types of squitters: short squitter and extended squitter. As short squitter includes downlink format capability, ICAO24 address, and cyclic redundancy check (CRC). An extended squitter contains all the information of a short squitter but it also includes altitude, position, heading, and velocity. To analyze all the information, we have focused on the extended squitter in this study. The ADS-B 1090ES signal is modulated using pulse position modulation (PPM), which is 112 bits long. A 0.8μs preamble should precede the data block.

B. UAT978

UAT978 applies to aircraft that fly below 18,000 feet in the US, mainly focusing on GA. If an aircraft flies above 18,000 feet, it must be equipped with an ADS-B 1090ES transmitter. Besides navigation, UAT978 also provides services such as flight information system-broadcast (FIS-B) and traffic information system-broadcast (TIS-B). UAT978 uses continuous phase frequency shift keying (CPFSK) modulation with a modulation index of 0.6 and a data rate of 1.041667 Mbps. There are two types of UAT ADS-B downlink messages: basic and long. A basic message contains 144 bits, while a long message has 272 bits. Forward error correction (FEC) is performed using a systematic Reed–Solomon error correction code. For the basic message, the FEC should be 96 bits long, and for the long message, the FEC should be 112 bits long. *1110101011001101110110100100111100010* is the default synchronization bit pattern for both types of messages in UAT978.

C. MOBILE COCKPIT INFORMATION SYSTEM

Compared to SSR, ADS-B is very handy and lightweight. With this simplified version of the air navigation technique, many manufacturers offer portable ADS-B transceivers. Some of these transceivers can fit in the plane's cockpit; some are hung on the window. They transmit and receive the ADS-B signals with a built-in antenna or via the aircraft's antenna port. EFB applications hosted on smartphones or tablets are connected to the transceiver device via WiFi. EFB application displays all the necessary navigation data to the pilot. These portable transceivers are programmable via computer or mobile application. A needed change in the static information (e.g., ICAO24 address, flight number, squawk code) can be done via the nominated program. In contrast,

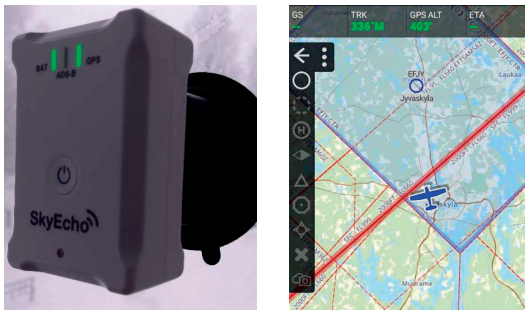


FIGURE 2. SkyEcho2 with OzRunways EFB application.

dynamic data (e.g., location, altitude) are changed automatically via the GNSS receiver of the device. Figure 2 shows a MCIS setup, where data ADS-B data from the SkyEcho2 transceiver is displayed on OzRunways EFB application via WiFi network.

D. TERMINOLOGY CLARIFICATIONS

During different phases of the experiment in this study, we observed different behavior from various tested configurations and MCISs. Below we explain the terminology and meaning of states, as used throughout this paper:

- **Crash:** If a MCIS totally shuts down unexpectedly or ungracefully due to software misbehaviour from the Denial-of-Service (DoS) attack inputs, we classify that as a crash. Commonly, this is the first and immediate step before an attacker can perform remote code execution (RCE) or arbitrary code execution (ACE) attacks- This means the attacker can execute their own code (e.g., ransomware, malware) on the affected system (e.g., device, software, MCIS).
- **Unresponsive:** Some setups did not crash but they could not handle the overwhelming amount of data. As a result, they hang on, which is described as unresponsive.
- **Output clogged:** Some setups, perhaps to avoid a system crash or due to design limitations, can decode or display a limited number of aircraft. The setups cyclically show the ADS-B message within that capacity. Sometimes the ADS-B messages from new aircraft replace the old ones within that limit, or new aircraft from valid ADS-B signals do not appear at all. This situation is called output clogged.
- **Unreadable screen:** When the system is flooded with attacker ADS-B signals, the very large number of data fields and aircraft icons make it impossible to read the screen. However, the system keeps functioning without crashing or becoming unresponsive, though most of the time, the system becomes slower.
- **No effect:** Despite the attack, if the system behaves normally without any visible/observable DoS or side-effects, we called that no effect. However, none of the tested MCISs were able to handle a massive amount of

ADS-B messages (e.g., 200,000 or more). For instance, we observed in many MCISs a significant amount of valid messages being dropped (i.e., the number of processed/displayed ADS-B messages is significantly lower than the number of input ADS-B messages we send). In such cases, we called this no effect but make a side comment that messages were dropped.

III. RELATED STUDIES

Securing the ADS-B has drawn massive attention from researchers due to its direct connection to aircrafts' safe navigation and the effects that failures have on passengers' life. As early as 2004, Krozel and Andrisani [15] reported that data dropouts, erroneous inputs, and deception might degrade data integrity from ADS-B-equipped aircraft. They proposed verification and validation techniques to ensure data integrity using a Kalman filter. The filter would smooth out noise in measured ADS-B signals, identify and suppress erroneous data, coast between data dropouts, and provide the current best state estimates. Since then, there have been many kinds of studies to enhance ADS-B communication's authenticity, security, confidentiality, and integrity [16]–[18].

Sampigethaya [19] focused on the security of ADS-B and proposed a framework for broadcast data link-based navigation and surveillance for the ADS-B-enabled aircraft. Costin and Francillon [9] presented the first public implementation and results of launching ADS-B message injection and spoofing attacks. Strohmeier *et al.* [20] analyzed the 1090 MHz communication channel to understand the behavior of ADS-B 1090ES under increasing traffic load and security challenges. They concluded that the cheap and easily available SDRs posed a significant threat to ADS-B communication and could be used for practical RF-based attacks. Schäfer *et al.* [11] implemented attacks on ADS-B 1090 using USRP N210 as the transmitter and SBS-3 as the receiver. They showed that active attacks such as ghost aircraft injection, ghost aircraft flooding, ground station flooding, and virtual trajectory modification are easily implemented using low-cost devices.

McCallie *et al.* [21] analyzed the security vulnerabilities associated with ADS-B implementations. They classified the attacks and examined the potential damage that the attacks may have on air transportation operations. They stated that ADS-B exploitation could cause disastrous consequences, confusion, aircraft groundings, and in the worst case even plane crashes. Manesh *et al.* [22] used Piccolo autopilot and a portable ground station to observe the autopilot's ghost aircraft injection response. They injected fake ADS-B messages causing ghost aircraft to appear in the vicinity of the Piccolo autopilot (ownership). This caused the autopilot to take evasive measures to avoid the collision. Subsequently, they pushed the ghost aircraft very close to the autopilot. The sudden appearance of false aircraft caused the pilot to execute a steep turn and start descending to regain well-clear as soon as possible. Eskilsson *et al.* [23] demonstrated ADS-B and controller–pilot data link communications (CPDLC) attacks using HackRF. They used freely available *ADSB_Encoder.py* Python script [24] to encode ICAO, latitude, longitude, and altitude information into an IQ file. Later the file was transmitted over the air using a HackRF device and decoded by *dump1090* software. They stated that simple implementation, systematic documentation, and relatively inexpensive equipment could also result in an increasing number of people carrying out an attack. The acquisition of more attacking devices can lead to a large-scale attack.

Tabassum *et al.* [25], [26] concluded ADS-B systems are prone to message and payload loss. In their exploratory analysis, they found that message contents are sometimes inconsistent with nominal conditions. They spotted message dropout, partial message content losses, data drift from the nominal value, and discrepancies between geometric and barometric altitude. They suggested that prior to the complete implementation of ADS-B, it is important to address, understand and monitor these deficiencies.

Air communication modes are also a significant source of big data that must be handled securely and effectively. Mink *et al.* [27] analyzed the unaddressed big data issues for NextGen. They evaluated the NextGen system using five differentiated qualitative characteristics of big data: volume, velocity, variety, veracity, and value. They estimated that all modes (Mode A, C, and S) combined would generate 41 TiB data per year at a velocity of 13 messages per millisecond with no encryption. These findings indicate that the NextGen system has several big data challenges that must be addressed it it is to obtain its maximal potential. However, no such study in Europe has been conducted yet.

Wu *et al.* [10] did a survey of the security issues of ADS-B. They noted that the attack intention could be for economic benefit, terrorism, cyber warfare, or personal interest. The authors modeled the attacker as professional hacking groups, terrorist organizations, military organizations, or amateurs. The survey showed that a single solution does not fully protect the ADS-B system's security. The public key infrastructure or spread spectrum technology can resist most attacks, but there are still deficiencies. They proposed a multi-layered security framework.

Most recently, Leonardi *et al.* [28] studied the effect of jamming attacks in crowd-sourced air traffic surveillance. They found that ground-based communication link jamming can disrupt ADS-B communication more easily and effectively than an air-based jammer and it is easy to implement the attack from the ground. Their work complements our study in the sense that it analyzes DoS attacks on air traffic surveillance (including ADS-B). However, they performed the DoS on the communication link where we performed it on the datalink ADS-B layers.

Dave *et al.* [29] reviewed the cybersecurity challenges in aviation communication, navigation, and surveillance. According to them, as the aviation sector becomes digitized and increasingly reliant on wireless technology, cyberattackers in this sector are also increasing. From old VHF, CPDLC, and PSR to today's ADS-B technology, all are proven to

TABLE 1. Comparison with related work shows different types of ADS-B attacks *demonstrated* throughout the state of the art.

Reference	Injection	Spoofing	Flooding Display (operational DoS)	Software Attacks (application DoS, RCE)	Logical Vulnerabilities	Coordinated Attackers N-to-1	Coordinated Attackers N-to-N	ADS-B UAT978 (any attacks)	Unified Attack Platform (1090ES and UAT978)
Costin et al. [9]	✓	✓	–	–	–	–	–	–	–
Schäfer et al. [11]	✓	✓	✓	–	–	–	–	–	–
Shang et al. [13]	–	–	–	–	–	–	✓(theoretical, simulated)	–	–
Khandker et al. [14]	✓	✓	–	–	✓	✓	✓(supported, untested)	–	–
This work	✓	✓	✓	✓	–	–	–	✓	✓

be vulnerable to cyberattacks. Moreover, the unencrypted nature of ADS-B opens many other attack paradigms. SDR availability is one of the most technical advantages for attackers. Many GA pilots use MCIS, which is very handy and easy to install. Lundberg *et al.* [30] found that this type of mobile setup is not a part of the onboard systems. Thus, its reliability does not meet the standards applied to traditional avionics such as radio technical commission for aeronautics, aeronautical radio incorporated, and the European organisation for civil aviation equipment. They tested three sets of hardware and applications: Appareo Stratus2 receiver with the ForeFlight app, Garmin GDL 39 receiver with the Garmin Pilot app, and SageTech Clarity CL01 with the WingX Pro7 application. They reported that all of them were vulnerable, allowing an attacker to manipulate information presented to the pilot. They recommended a device should sign the data sent from the receiver to the app and vice versa. They also recommend regularly updating the firmware, implementing EFB updates, and to following security-aware software development in order to enhance the security of such mobile cockpit information systems.

Even though the security of ADS-B is heavily researched, Lundberg *et al.* [30] have provided as the sole contribution to MCIS security. However, technology and the demand for MCIS have drastically changed since that study. Many new ADS-B transceivers and software have been developed. Attackers have new tools and ideas as well. Therefore, evaluating the attacks on MCIS against current technology is essential. In comparison with Lundberg *et al.* [30], our work provides comprehensive qualitative and quantitative security feature testing of MCIS. Last but not least, the present paper complements our research work and the results in [14], [31].

Table 1 compares this article's attacks and contributions against the relevant attacks presented in the literature.

IV. EXPERIMENT SETUP

In this section, we describe our approach for attacking MCIS. We performed the experiments in well-controlled lab environments using low power, placing the receivers and transmitters in close proximity, and employing signal attenuators.

A. ATTACK PLATFORM

We used Python programming language to generate the attack payloads. Then a program called GNU radio companion (GRC) was used to produce the IQ values, subsequently transmitted on the air using transmission-enabled SDR. Three transmission-enabled SDRs were used: HackRF, BladeRF, and PlutoSDR. One type of device was sufficient for the attacks in this study. However, we tested three of them to check the feasibility of attacks by heterogeneous devices. To encode the position and altitude into the ADS-B 1090ES signal, we used Yusupov's example script [24]. Later we extended the software's service by writing the codes for other necessary data fields of the ADS-B 1090ES, such as flight information, velocity, and squawk. Yusupov also provided a UAT978 long-message generator [32], and we used that script to experiment with UAT978 data encoding. We slightly modified Larroque's Reed–Solomon codec to generate the FEC [33]. Later, by adding synchronization bits and proper serialization, we generated the final UAT978 attack payload. We used GRC's CPFSK block to transmit the UAT978 signal over the air. Our written software can send *1-to-N* 1090ES and UAT978 messages by *1-to-N* transmitters. It is controlled by several arguments in a command-line interface or with a graphical user interface. To generate a fake ADS-B radio signal, we generate N messages to a CSV file. Then, we create the IQ file of those messages for the 1090ES or UAT978 signal. We duplicate each message 5–10 times to ensure that the tested receiver caught each one. In the end, we transmitted all the N messages very quickly to push the receiving software to its limits. Figure 3 shows how a Python-generated attack payload reaches the MCIS through a radio link. Below we present the ADS-B fields and other parameters that can be set in our software to send individual or multiple messages using 1090ES or UAT978 protocols.

- **icao24:** set an ICAO address to the message.
- **squawk:** set a squawk code to the message.
- **flightnum:** set a flight number.
- **velocity:** set airspeed of the aircraft.
- **lat:** set GPS latitude coordinate.
- **lon:** set GPS longitude coordinate.
- **alt:** set GPS altitude.
- **gain:** set the transmit gain.
- **modetx** set the protocol 1090ES or UAT978.
- **devtx:** set a specific transmitter device.
- **file** set the paths of attack file.
- **ts:** set a timestamp in milliseconds.
- **crc:** set a CRC checksum. For UAT978, this argument refers to the Reed–Solomon FEC.
- **multiprocessing:** set the number of parallel transmitters to use at once.

B. MOBILE COCKPIT INFORMATION DEVICES

We tested six mobile cockpit information devices from different manufacturers. Some of them had ADS-B transmission capability, while others were limited to receive only. Because

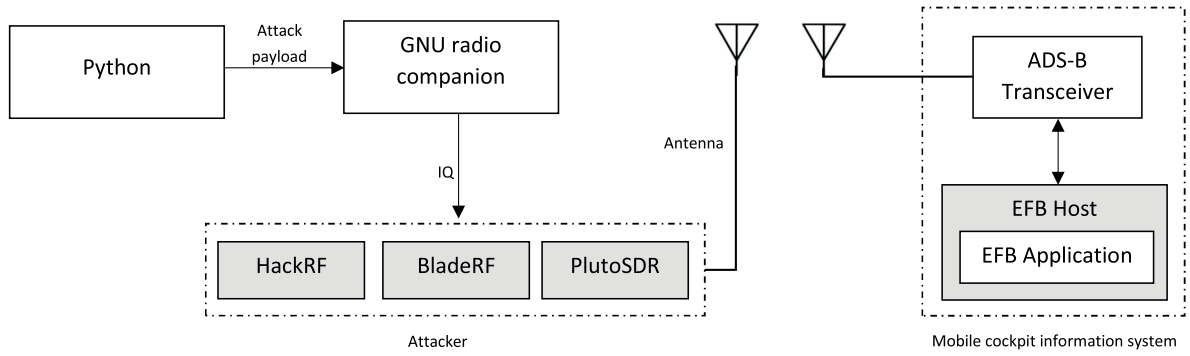


FIGURE 3. Attacker model.

TABLE 2. List of tested mobile cockpit information devices.

Device Name	Receive mode	Transmit mode
uAvionix SkyEcho2	1090ES UAT978	1090ES
uAvionix echoUAT	1090ES UAT978	UAT978
ForeFlight Sentry	1090ES UAT978	No
Garmin GDL 52	1090ES UAT978	No
ADL 180	1090ES	No
Helios Avionics SensorBox	1090ES	No

ADS-B is not fully functional in many parts of the world, some devices did not support transmission. Table 2 shows the list of tested devices.

C. ELECTRONIC FLIGHT BAG APPLICATIONS

A variety of EFB applications support the devices listed in Table 2. However, not all the applications were compatible with all the devices, because some devices use proprietary protocols to exchange data with applications. The most popular protocol is GDL-90 [31], which most applications, such as AvPlan, FLyQ, OzRunways, use. However, the GarminPilot application worked only with the Garmin GDL-52 device, while SensorBox worked with their developed Horizon application. Table 3 shows the list of tested EFB applications. We included the world-wide installation number for Android platform applications (reliably available) to get an idea of how many users could be affected by an application failure. Some applications were not available for a specific platform, device, or our region. We cross-marked if we could not test it on a platform. Tested applications per platform are check-marked. Missing information was marked NA (not available). All the EFB applications did not support all the tests (see VI-D).

TABLE 3. List of tested EFB applications.

Application name	Tested platform		Installation number
	Android	iOS	Android
ADSB Flight Tracker	✓	×	100k+
ADSB Flight Tracker Lite	✓	×	10k+
ADS-B for Pilot PRO	✓	×	10k+
ADL Connect	✓	✓	1k+
AirMate	×	✓	50k+
Avare ADSB	✓	×	100k+
Avare ADSB Pro	✓	×	5k+
AvPlan	✓	✓	1k+
EasyVFR4	✓	✓	1k+
FlyQ	×	✓	10k+
ForeFlight	×	✓	NA
Garmin Pilot	✓	✓	100k+
Horizon	✓	✓	10k+
iFlightPlanner	×	✓	NA
Levil Aviation	×	✓	NA
Naviator	✓	×	100k+
OzRunways	✓	✓	50k+
Pilots Atlas	×	✓	50+
SkyDemon	✓	✓	100k+
Stratus Insight	×	✓	NA
Xradio ADS-B Receiver	✓	×	10k+

V. ATTACKS ON MOBILE COCKPIT INFORMATION SYSTEMS

We implemented RF-link-based attacks on the MCISs. Being portable and lightweight, MCISs have limited computation power, memory capacity, and screen size. Therefore, an ADS-B packet-level DoS attack would be a good choice to check their resilience under a cyberattack. In this study, we primarily focused on DoS attacks on the MCISs. DoS attacks disrupt the availability of services by clogging or

shutting down service entities or networks. The intention is to prevent legitimate users from accessing the service or to prevent legitimate data from reaching its destination. This is accomplished by crashing the service with malicious data or by flooding its input with garbage data or fake messages beyond its capabilities. Because ADS-B does not use authentication or encrypted wireless traffic, it is virtually impossible for it to block a malicious source of fake signals. Therefore, identifying and properly handling the messages is the key to defending against these attacks. The effects of DoS attacks on wireless traffic and wireless sensor networks have been the subject of extensive and prolific research. Osanaiye *et al.* [34] and Ghildiyal *et al.* [35] concluded that DoS attacks could be detrimental to the operation of the system, and defending against them is not trivial. Strohmeier *et al.* [20] addresses the security issues of ADS-B broadcasts, stating that the system is sensitive to RF attacks. DoS attacks can lead to an unresponsive or disabled system, which can lead to poor decision-making within ATC or the malfunction of automated systems because of the authentic information. Our attack system operated on a click-and-run approach, where we first generated random yet valid ADS-B messages and transmitted those messages via transmission-enabled SDRs in a rapid burst. While attacking, we visually observed the effect of the attack and recorded the observations. We noted if the software had any crashes, errors, malfunctions, or unresponsiveness. If not, we noted if the output of the software was clogged enough to miss ADS-B messages.

We also tested coordinated attacks on the MCISs. In these attacks, multiple attackers targeted a single aircraft (or ICAO24 address). Multiple attackers continuously sent sporadic information about the targeted aircraft. To a receiver, this seems like the targeted aircraft is erratically changing its location or other important flight-relevant information. We showed that such attacks lead to logical vulnerabilities [14].

Finally, we conducted fuzz testing for the EFB applications. This is an automated software testing method for finding implementation and input sanitization bugs using intentionally malformed or randomized inputs. The ADS-B devices communicated to the mobile application following some protocols. Among them, GDL-90 is the most popular. By following this protocol but using malformed input, we conducted fuzz tests of the EFB applications [31].

VI. RESULTS AND EVALUATION

We identified many candidate EFB applications for various tests. After a few trial-and-error setups (e.g., successful installation and configuration with hardware), 17 applications were selected for RF-link-based attacks, and 15 applications were selected for fuzz tests. Some applications supported both types of tests, while some were limited to only one. In total, 21 distinct EFB applications were tested in this study. Furthermore, six MCIS devices were tested. Of them, four supported UAT978, while all six supported the 1090ES protocol.

In the receive mode, compared with other MCIS devices such as SkyEcho2 or Sentry, the echoUAT receives and processes both 1090ES and UAT978 messages at a considerably lower ($\approx 100 \times -140\times$ less) number of messages per minute. Subsequently, it forwards a significantly smaller number of ADS-B messages to the decoding application. Therefore, we construe this as being the main reason that none of the tested mobile apps crashed during the DoS attack tests while using echoUAT hardware. SkyEcho2 and Sentry can receive up to 55k distinct ICAO24 addresses per minute, but echoUAT surprisingly has a *hardware limitation* that processes approximately 400 distinct ICAO24 addresses per minute. We are not sure about the core reasons for this functional discrepancy. The maximum transmission rates of messages per second for 1090ES and UAT978 were 6.2 and 1, respectively [36], [37]. However, we have not found the maximum or minimum receiving resolution of the ADS-B system. In our experiment, we found that SensorBox and Garmin GDL-52's decoding capacity was approximately 10,000 and 30 distinct ICAO24 addresses, respectively. Because these two devices work with their proprietary application only, we could not find out whether the limitation was in the hardware or the software. During the test, we found the the ADL 180 device displayed approximately 75 aircraft at a time in both Android and iOS applications.

A. DoS ATTACK RESULTS FOR UAT978

DoS attacks on UAT978 were tested on a number of hardware and software combinations:

- 4 MCIS devices
- 2 mobile operating systems
- 9 EFB applications
- 24 different setup combinations

Overall, our DoS attack affected 9 out of 24 tested configuration for UAT978. The configurations crashed, clogged, or were unresponsive. Some applications were not affected during the DoS attack. Instead they dropped a significant number of legitimate messages and displayed only a tiny portion of the transmitted signal. In practice, with the limited memory, computational power, and display capacity, it is nearly impossible for the MCISs to display and update the ADS-B data for a huge number of distinct aircraft flawlessly (e.g., attack payload of 200,000 ICAO24 address or more). Despite the applications not crashing, we believe that clogging the system and disabling the capability of the system to show all required signals to the user was a successful DoS attack as it disrupted the availability of required data. We marked these situations as non-impacted to distinguish the systems that showed even some resilience to the attacks from the ones that crashed consistently. Therefore, we believe that the non-impacted setups are also not adequate for safety and mission-critical systems. Table 4 presents a summary of the results of the attacks.

TABLE 4. DoS attack results for UAT978.

EFB application	Receiving device	EFB host and operating system	Effect	Time to DoS (seconds)	Observation note
AirMate 2.3	SkyEcho2	iPhone 11 & iOS 14.4	**Crash**	90	
AirMate 2.3	echoUAT	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
AvPlan 7.10.7	SkyEcho2	iPhone 11 & iOS 14.4	**Crash**	30	
AvPlan 7.10.7	echoUAT	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
AvPlan 1.3.14	SkyEcho2	Samsung A21s & Android 10	**Crash**	60	
AvPlan 1.3.14	echoUAT	Samsung A21s & Android 10	No effect	NA	Valid message dropped
EasyVFR4 4.0.866	SkyEcho2	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
EasyVFR4 4.0.866	echoUAT	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
EasyVFR4 4.0.870	SkyEcho2	Samsung A21s & Android 10	No effect	NA	Valid message dropped
EasyVFR4 4.0.870	echoUAT	Samsung A21s & Android 10	No effect	NA	Valid message dropped
FlyQ EFB 5.0	SkyEcho2	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
FlyQ EFB 5.0	echoUAT	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
ForeFlight 13.0.1	Sentry	iPhone 11 & iOS 14.4	**Crash**	15	
ForeFlight 13.0.1	SkyEcho2	iPhone 11 & iOS 14.4	**Crash**	15	
ForeFlight 13.0.1	echoUAT	iPhone 11 & iOS 14.4	No effeact	NA	Valid message dropped
Garmin pilot 10.5.7	GDL 52	iPhone 11 & iOS 14.4	**Output clogged**	10	Maximum 30 aircraft display
Garmin pilot 8.0.0	GDL 52	Samsung A21s & Android 10	**Output clogged**	10	Maximum 30 aircraft display
OzRunways 10.10	SkyEcho2	iPhone 11 & iOS 14.4	**Crash**	420	
OzRunways 10.10	echoUAT	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
OzRunways 4.4.1	SkyEcho2	Samsung A21s & Android 10	**Unresponsive**	NA	
OzRunways 4.4.1	echoUAT	Samsung A21s & Android 10	No effect	NA	Valid message dropped
Pilots Atlas 5.11.10	SkyEcho2	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
Stratus Insight 5.17.3	SkyEcho2	iPhone 11 & iOS 14.4	**Crash**	60	
Stratus Insight 5.17.3	echoUAT	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped

B. DoS ATTACK RESULTS FOR 1090ES

The attacks on ADS-B 1090ES were tested on a number of hardware and software combinations:

- 6 MCIS devices and 1 RTL-SDR
- 2 mobile operating system
- 15 EFB applications
- 44 total configuration combinations

We found that some EFBs worked with the RTL-SDR through SDR driver v.3.10 in the Android platform. Thus, we used RTL-SDR as the RF front-end for EFBs. Overall, out of 44 tested configurations for DoS attacks on ADS-B 1090ES, 28 were affected. Table 5 presents a summary of the results of the attacks.

C. ADS-B OUT IMPACT

We also investigated the impact of DoS attacks on the performance of ADS-B OUT. Among the MCIS devices, SkyEcho2 transmits the 1090ES signals and echoUAT transmits the UAT978 signals. Table 6 shows the results of the ADS-B OUT impact experiment. In all the ADS-B OUT scenarios, we performed the attacks with a burst of 10k unique ICAO ADS-B messages. However, changing attack intensity numbers (i.e., increasing to bursts of 20k or 30k ICAO24 address) did not significantly change the impact. Each test was were carried out 15 times for each scenario. The results show that the DoS attack on ADS-B IN reduced the ADS-B OUT capacity of SkyEcho2 by approximately 15%, while no significant impact was observed on the echoUAT. However, it is still unclear whether the described impact on SkyEcho2 ADS-B OUT also had a *qualitative impact*. In other words, it remains for future work to investigate if the decline was due to some critical ADS-B OUT messages being dropped or being sent with unacceptable delay. For example, if some ADS-B OUT packets are delayed or dropped altogether, this could dramatically impact the effectiveness of the traffic collision avoidance system.

D. FUZZING

The communication between the MCIS devices and the EFB was mostly conducted via WiFi using the GDL-90 protocol by Garmin. However, the MCIS devices used insecure WiFi connections through which malformed data can be passed to

TABLE 5. DoS attack results for 1090ES.

EFB application	Receiving device	EFB host and operating system	Effect	Time to DoS (seconds)	Observation note
ADL Connect 8.95	ADL 180	iPhone 11 & iOS 14.4	**Unresponsive**	300	
ADL Connect 8.90	ADL 180	Samsung A21s & Android 10	**Output clogged**	60	Maximum 75 aircraft display
ADSB Flight Tracker v 30.9	RTL-SDR	Samsung A21s & Android 10	**Output clogged**	600	
ADSB Flight Tracker Lite v 8.4.1	RTL-SDR	Samsung A21s & Android 10	**Output clogged**	600	
ADS-B for Pilot PRO v 1.8	RTL-SDR	Samsung A21s & Android 10	**Output clogged**	600	
Avare ADSB v 4.9.1	RTL-SDR	Samsung A21s & Android 10	**Output clogged**	600	CRC errors reported
Avare ADSB Pro v 4.9.1	RTL-SDR	Samsung A21s & Android 10	**Output clogged**	600	CRC errors reported
AirMate v 2.3	SkyEcho2	iPhone 11 & iOS 14.4	**Crash**	1200	
AirMate v 2.3	echoUAT	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
AvPlan v 7.10.7	SkyEcho2	iPhone 11 & iOS 14.4	**Crash**	120	
AvPlan v 7.10.7	echoUAT	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
AvPlan v 7.10.7	ADL 180	iPhone 11 & iOS 14.4	**Output clogged**	60	Maximum 75 aircraft display
AvPlan v 1.3.14	SkyEcho2	Samsung A21s & Android 10	**Crash**	180	
AvPlan v 1.3.14	echoUAT	Samsung A21s & Android 10	No effect	NA	Valid message dropped
AvPlan v 1.3.14	ADL 180	Samsung A21s & Android 10	**Output clogged**	60	Maximum 75 aircraft display
EasyVFR4 v 4.0.866	SkyEcho2	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
EasyVFR4 v 4.0.866	echoUAT	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
EasyVFR4 v 4.0.866	ADL 180	iPhone 11 & iOS 14.4	**Output clogged**	60	Maximum 75 aircraft display
EasyVFR4 v 4.0.870	SkyEcho2	Samsung A21s & Android 10	No effect	NA	Valid message dropped
EasyVFR4 v 4.0.870	echoUAT	Samsung A21s & Android 10	No effect	NA	Valid message dropped
EasyVFR4 v 4.0.870	ADL 180	Samsung A21s & Android 10	**Output clogged**	60	Maximum 75 aircraft display
FlyQ EFB v 5.0	SkyEcho2	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
FlyQ EFB v 5.0	echoUAT	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
FlyQ EFB v 5.0	ADL 180	iPhone 11 & iOS 14.4	**Output clogged**	60	Maximum 75 aircraft display
ForeFlight v 13.0.1	Sentry	iPhone 11 & iOS 14.4	**Crash**	120	
ForeFlight v 13.0.1	SkyEcho2	iPhone 11 & iOS 14.4	**Crash**	120	
ForeFlight v 13.0.1	echoUAT	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
ForeFlight v 13.0.1	ADL 180	iPhone 11 & iOS 14.4	**Output clogged**	60	Maximum 75 aircraft display
Garmin pilot 10.5.7	GDL 52	iPhone 11 & iOS 14.4	**Output clogged**	10	Maximum 30 aircraft display
Garmin pilot 8.0.0	GDL 52	Samsung A21s & Android 10	**Output clogged**	10	Maximum 30 aircraft display
Horizon v 3.1	SensorBox	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
Horizon v 3.1	SensorBox	Samsung A21s & Android 10	No effect	NA	Valid message dropped
OzRunways v 10.10	SkyEcho2	iPhone 11 & iOS 14.4	**Crash**	120	
OzRunways v 10.10	echoUAT	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
OzRunways v 10.10	ADL 180	iPhone 11 & iOS 14.4	**Output clogged**	60	Maximum 75 aircraft display
OzRunways v 4.4.1	SkyEcho2	Samsung A21s & Android 10	**Unreadable screen**	600	
OzRunways v 4.4.1	echoUAT	Samsung A21s & Android 10	No effect	NA	Valid message dropped
OzRunways v 4.4.1	ADL 180	Samsung A21s & Android 10	**Output clogged**	60	Maximum 75 aircraft display
Pilots Atlas v 5.11.0	SkyEcho2	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
Pilots Atlas v 5.11.0	ADL 180	iPhone 11 & iOS 14.4	**Output clogged**	60	Maximum 75 aircraft display
Stratus Insight v 5.17.3	SkyEcho2	iPhone 11 & iOS 14.4	**Crash**	180	
Stratus Insight v 5.17.3	echoUAT	iPhone 11 & iOS 14.4	No effect	NA	Valid message dropped
Stratus Insight v 5.17.3	ADL 180	iPhone 11 & iOS 14.4	**Output clogged**	60	Maximum 75 aircraft display
Xradio ADS-B Receiver v 5.106	RTL-SDR	Samsung A21s & Android 10	**Crash**	20	

TABLE 6. Impact on ADS-B OUT when executing a DoS attack.

Device	ADS-B OUT support	ADS-B IN support	Numbers of test runs	ADS-B OUT at normal operation (avg. msg/min)	ADS-B OUT at DoS attack (avg. msg/min)	Performance Impact (%)
uAvionics SkyEcho2	1090ES	1090ES	15	268	227	– 15%
		UAT978	15	NA	228	– 15%
uAvionix echoUAT	UAT978	1090ES	15	NA	60	0%
		UAT978	15	60	60	0%

the application, and this may affect the integrity and security of the overall system.

We performed extensive fuzz-testing for the EFBs by using the American fuzzy lop (AFL) Python implementation. AFL was set up to send malformed data to the IP address of the EFB application host. Table 7 highlights our fuzz-testing results. To compare the result with the DoS tests, we also show in Table 7 the corresponding ADS-B DoS test result on corresponding EFBs. In the table, we marked as NA whenever we could not configure an EFB for the test (e.g., due to unavailability, or some other limitation). In addition, some applications (e.g., EFB apps, desktop software) did not work with our MCIS devices. However, they worked with GDL-90 and the fuzzing setup, which can also be seen in Table 7 in their corresponding rows. The results show that 3 out of 7 or approximately 42% applications were affected by the fuzzing test on the Android platform. On the iOS platform, the impact rate was around 53% for 7 affected EFB applications out of 13. Some EFBs applications, such as AvPlan, were crashed by both tests. Some EFB were crashed by one of the tests, while only EasyVFR4 and Pilot Atlas survived both tests.

One particular observation from Table 7 is as follows. If the tested application is vulnerable to ADS-B DoS attacks (e.g., crash), it is extremely likely that it will be found vulnerable by GDL-90 fuzzing with very likely the same consequences (e.g., crash). Examples include AirMate, AvPlan, OzRunways, and Stratus Insight. Likewise, if an application did not present any major issues during ADS-B DoS attacks, it will very likely pass the GDL-90 fuzzing tests. Although, exceptions to this rule are iFlightPlanner and Levil Aviation. This shows strong efficiency and correlation of cybersecurity testing by ADS-B DoS and/or GDL-90 fuzzing. This means that insufficiently secured applications (e.g., see Table 7) that result in serious consequences (e.g., software/EFB crash) will eventually be discovered with sufficient testing when using the methodology and the pentesting platform design that we propose in this paper and in our related works [14], [31].

E. LOGICAL VULNERABILITIES

For an aircraft, ADS-B traffic information in the MCIS updates with the reference of the ICAO24 address. If multiple sources of ADS-B signal containing the same ICAO24 address emit the position information from different places, it appears that aircraft is changing its position erratically.

TABLE 7. Comparing ADS-B DoS results with GDL-90 fuzzing results [31].

Fuzzed EFB (has GDL-90)	Application platform		Result during ADS-B DoS (for comparison)
	Android	iOS	
AirMate	NA	**Crash**	**Crash**
AvPlan	**Crash**	**Crash**	**Crash**
EasyVFR4	No effect	No effect	No effect
FlyQ EFB	NA	**Unresponsive**	No effect
ForeFlight	NA	No effect	**Crash**
Horizon	No effect	No effect	**Clogged**
iFlightPlanner	NA	**Crash**	NA
Levil Aviation	NA	**Crash**	NA
Naviator	**Unresponsive**	NA	NA
OzRunways	**Crash**	**Crash**	**Crash**
Pilots Atlas	NA	No effect	No effect
SkyDemon	No effect	No effect	NA
Stratus Insight	NA	**Crash**	**Crash**
Traffic	No effect	NA	NA
Xavion	NA	No effect	NA

Also, we found that none of the tested MCIS setups check the received data's integrity. For example, many aircraft might have the same flight number or irrational altitude and speed relationship [14]. Such a situation may raise logical vulnerabilities for the MCIS user.

VII. DISCUSSION

We did not observe any hardware crashes. However, this does not mean the devices we tested do not have potential bugs or security vulnerabilities at their hardware or firmware level. In fact, firmware vulnerabilities are quite common in IoT and embedded devices [5], [6] and as Muench *et al.* [38] demonstrated, when memory is corrupted in embedded devices, the results are different from desktop systems. Looking more into the future, we argue that the possibilities of the presented attacks may have impacts beyond the ground-, and aircraft-based ADS-B systems and well into the aerospace domain. The emergence and deployment of satellite-based ADS-B surveillance and receivers [39]–[41] and the increase of ADS-B application in unmanned aerial vehicles (UAVs) [42],

[43] could increase the attack sphere and severely amplify the potential impact of attacks [44].

To address the security vulnerabilities of MCISs demonstrated in this study, we present some solutions. First, the hardware, firmware, and software should be rigorously and continuously tested through automated means such as our platform. The testing should start from the development environment and extend to the operational environment, because development environments do not fully represent the proper use cases. Kacem *et al.* [45] proposed a crypto and radio-location-based hybrid solution to thwart ADS-B attacks. Their proposed framework called ADS-Bsec provides authenticity and integrity for ADS- B packets by using a keyed-hash message authentication code (HMAC). The minimum size of an HMAC is 128 bits which need to be distributed among several ADS-B messages. Although their proposed framework supports backward compatibility with the current ADS-B protocol; however, the CRC checks must be disabled.

Kassab [46] surveyed safety-critical software development and concluded that although safety-critical applications are tested more frequently, quality assurance testing is mostly performed in the very late stages of software development. According to him, the software development practices must be of a higher standard. Possible attack vectors must be identified during software development, and mitigation must be implemented. The iterative development cycle between testing and mitigation implementation should be enforced. For example, a subset of DO-178B (Software Considerations in Airborne Systems and Equipment Certification) could be developed and explicitly required for MCISs. Furthermore, proper memory management must be implemented in the software. The software that crashed, hung up, or went unresponsive does not have appropriate memory management implemented. Therefore, it can be assumed that the EFBs were not tested against DoS attacks during the software development. Researchers have proposed several defense strategies against attacks on ADS-B. However, the effectiveness of the proposed attack detection and prevention methods are yet to be tested in academia and industry. Nonetheless, some defense strategies are available.

Li and Wang [47] proposed a sequential collaborative attack detection strategy based on ADS-B data. According to them, time series and position, the law of motion, historical data, etc., can be used to detect injection, DoS, replay, and ghost attacks. However, the authors did not consider the physical or signal pattern of the attacks. They solely trusted the data. The position-related data of a commercial aircraft change a bit within 30 seconds. However, our study shows that a successful DoS attack can be performed within this short time. In contrast, it may take much more time to apply their proposed method to establish collaboration among the nodes such as ground stations and aircraft in the vicinity to detect the DoS attack. Ying *et al.* [48] proposed a deep neural network (DNN)-based spoofing detector. That method allows a ground station to examine each incoming message based on physical layer features such as IQ samples and phases to flag suspicious messages. The classifier predicts the ICAO24 address of the received ADS-B message and compares it against the claimed ICAO24 address. The rate of the change in the signal phase indicates the carrier frequency offset, which is a sum of frequency offsets and the Doppler shift. They used this feature for classification purposes. However, the main limitation of their method is the supervised learning method for a dynamic environment. An unknown legitimate aircraft flying over the region can initiate a false alarm. Moreover, radio propagation, receiver characteristics, and measurement noise also can affect the system. Our attacking approach can generate any ICAO24 address, which can be regarded as an aircraft flying for the first time in the air space with no historical data, thus bypassing the security or generating a false alarm. Jansen *et al.* [49] proposed a non-invasive trust evaluation system to detect attacks on ADS-B-based air-traffic surveillance. They used a ''Wireless Witnessing'' method to detect the attacks, which is essentially sharing the observations of geographically distributed sensors. An ADS-B receiving sensor should always receive the signals within its coverage. During a spoofing or an injection attack, sensors may receive such ADS-B signals that the signal's encoded position information exceeds the sensor's range. Multiple sensors' wireless witnessing would increase the probability of attack detection. By collecting scores from all the sensors, they calculated a total that indicated and ADS-B attack. Their proposed method is a post-processing method. It is not suitable for a real-time attack. As our study has shown, an attack can be made within a few minutes. A quick DoS attack may cause substantial negative consequences.

VIII. CONCLUSION

This work performed the largest and the most comprehensive cybersecurity assessment of DoS availability attacks on popular MCIS setups by modelling the attacker via remote unauthenticated and unauthorized RF-link. We developed a cybersecurity pentesting platform consisting of a large and comprehensive list of ADS-B transceivers, SDRs, and different EFB applications. Furthermore, we developed a flexible software suite that allows us to perform cybersecurity tests. We tested 44 1090ES and 24 UAT978 MCIS setups, for a total of 68 test configurations. Our ADS-B packet-level DoS attack affected availability on approximately 63% and 37% of 1090ES and UAT978 setups, respectively. The most concerning finding of this study was the very high number of MCISs and ADS-B software that crashed as a result of the performed attacks, where such crashes further expose the affected systems to potential ACE attacks.

The test results show that many, if not most, popular MCISs are vulnerable to many types of cyberattacks, including attacks on availability with resulting software crashes. Relevant overseeing and regulatory bodies (such as FAA, EASA, and ICAO) should investigate these issues further,

and propose practical steps and approaches to ensure further resilience of MCISs to cyberattacks.

ACKNOWLEDGMENT

The authors acknowledge the grants of computer capacity from the Finnish Grid and Cloud Infrastructure (persistent identifier urn:nbn:fi:research-infras-2016072533). Major parts of this research supported by cascade funding from the Engage Consortium's Knowledge Transfer Network (KTN) project ''Engage-204-Proof-of-concept: practical, flexible, affordable pentesting platform for ATM/avionics cybersecurity'' (SESAR Joint Undertaking under the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 783287). All and any results, views, and opinions presented herein are only those of the authors and do not reflect the official position of the European Union (and its organizations and projects, including Horizon 2020 program and Engage KTN). The authors thank Dr. Andrei Costin for facilitating and managing a partially-supporting grant Decision of the Research Dean on research funding within the Faculty (07.04.2021) of the Faculty of Information Technology, University of Jyväskylä (JYU). Hannu Turtiainen also thanks the Finnish Cultural Foundation/Suomen Kulttuurirahasto (https://skr.fi/en) for supporting his Ph.D. dissertation work and research (under grant decision no. 00221059) and the Faculty of Information Technology, JYU, in particular, Prof. Timo Hämäläinen, for partly supporting and supervising his Ph.D. work at JYU in 2021–2022. *(Syed Khandker and Hannu Turtiainen are co-first authors.)*

REFERENCES

[1] The Federal Aviation Administration. (2013). *FAA Aerospace Forecast, Fiscal Years 2013–2033*. Accessed: Feb. 24, 2021. [Online]. Available: https://www.faa.gov/data_research/aviation/aerospace_forecasts/media/2013_forecast.pdf

[2] EUROCONTROL. (2019). *European Aviation in 2040*. Accessed: Mar. 2, 2021. [Online]. Available: https://www.eurocontrol.int/sites/default/files/2019-07/challenges-of-growth-2018-annex1_0.pdf

[3] EASA. (2018). *EASA Seasonal Technical Commission*. Accessed: Mar. 2, 2021. [Online]. Available: https://www.easa.europa.eu/sites/default/files/dfu/EASA_STC_NEWS_JUNE_2018.pdf

[4] Z. Wu, A. Guo, M. Yue, and L. Liu, "An ADS-B message authentication method based on certificateless short signature," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 3, pp. 1742–1753, Jun. 2020.

[5] A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti, "A large-scale analysis of the security of embedded firmwares," in *Proc. 23rd USENIX Secur. Symp.*, 2014, pp. 95–110.

[6] A. Costin, A. Zarras, and A. Francillon, "Automated dynamic firmware analysis at scale: A case study on embedded web interfaces," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, May 2016, pp. 437–448.

[7] R. Mark. (2017). *FAA Warns of ADS-B False Alerts*. Accessed: Jun. 6, 2021. [Online]. Available: https://www.flyingmag.com/faa-warns-ads-b-false-alerts

[8] N. Morgan and G. D. Vynck. (2015). *WestJet Says it Never Sent Hijack Alarm, Wasn't in Danger*. Accessed: Jun. 4, 2021. [Online]. Available: https://www.bloomberg.com/news/articles/2015-01-10/westjet-hijack-signal-called-false-alarm

[9] A. Costin and A. Francillon, "Ghost in the air (traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices," in *Proc. Black Hat USA*, 2012, pp. 1–12.

[10] Z. Wu, T. Shang, and A. Guo, "Security issues in automatic dependent surveillance—Broadcast (ADS-B): A survey," *IEEE Access*, vol. 8, pp. 122147–122167, 2020.

[11] M. Schäfer, V. Lenders, and I. Martinovic, "Experimental analysis of attacks on next generation air traffic communication," in *Applied Cryptography and Network Security*. Berlin, Germany: Springer, 2013, pp. 253–271.

[12] M. R. Manesh and N. Kaabouch, "Analysis of vulnerabilities, attacks, countermeasures and overall risk of the automatic dependent surveillance-broadcast (ADS-B) system," *Int. J. Crit. Infrastruct. Protection*, vol. 19, pp. 16–31, Dec. 2017.

[13] F. Shang, B. Wang, F. Yan, and T. Li, "Multidevice false data injection attack models of ADS-B multilateration systems," *Secur. Commun. Netw.*, vol. 2019, pp. 1–11, Mar. 2019.

[14] S. Khandker, H. Turtiainen, A. Costin, and T. Hamalainen, "Cybersecurity attacks on software logic and error handling within ADS-B implementations: Systematic testing of resilience and countermeasures," *IEEE Trans. Aerosp. Electron. Syst.*, early access, Dec. 31, 2021, doi: 10.1109/TAES.2021.3139559.

[15] J. Krozel, D. Andrisani, M. Ayoubi, T. Hoshizaki, and C. Schwalm, "Aircraft ADS-B data integrity check," in *Proc. AIAA 4th Aviation Technol., Integr. Oper. (ATIO) Forum*, Sep. 2004, p.6263.

[16] K. Samuelson, E. Valovage, and D. Hall, "Enhanced ADS-B research," in *Proc. IEEE Aerosp. Conf.*, Mar. 2006, pp. 1–7.

[17] R. G. Wood, "A security risk analysis of the data communications network proposed in the NextGen air traffic control system," Ph.D. dissertation, School Educ. Found., Leadership Aviation, Oklahoma State Univ., Stillwater, OK, USA, 2009.

[18] L. Purton, H. Abbass, and S. Alam, "Identification of ADS-B system vulnerabilities and threats," in *Proc. 33rd Australas. Transp. Res. Forum (ATRF)*, 2010, pp. 1–16.

[19] K. Sampigethaya, R. Poovendran, and L. Bushnell, "A framework for securing future e-enabled aircraft navigation and surveillance," in *Proc. AIAA Infotech@Aerosp. Conf.*, Apr. 2009, pp. 1–10.

[20] M. Strohmeier, M. Schafer, V. Lenders, and I. Martinovic, "Realities and challenges of NextGen air traffic management: The case of ADS-B," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 111–118, May 2014.

[21] D. McCallie, J. Butts, and R. Mills, "Security analysis of the ADS-B implementation in the next generation air transportation system," *Int. J. Crit. Infrastruct. Protect.*, vol. 4, no. 2, pp. 78–87, Aug. 2011.

[22] M. R. Manesh, M. Mullins, K. Foerster, and N. Kaabouch, "A preliminary effort toward investigating the impacts of ADS-B message injection attack," in *Proc. IEEE Aerosp. Conf.*, Mar. 2018, pp. 1–6.

[23] S. Eskilsson, H. Gustafsson, S. Khan, and A. Gurtov, "Demonstrating ADS-B AND CPDLC attacks with software-defined radio," in *Proc. Integr. Commun. Navigat. Surveill. Conf. (ICNS)*, Sep. 2020, pp. 1B2-1–1B2-9.

[24] L. Yusupov. (2021). *Lyusupov/ADSB-Out*. Accessed: Feb. 24, 2021. [Online]. Available: https://github.com/lyusupov/ADSB-Out

[25] A. Tabassum, N. Allen, and W. Semke, "ADS-B message contents evaluation and breakdown of anomalies," in *Proc. IEEE/AIAA 36th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2017, pp. 1–8.

[26] A. Tabassum and W. Semke, "UAT ADS-B data anomalies and the effect of flight parameters on dropout occurrences," *Data*, vol. 3, no. 2, p. 19, Jun. 2018.

[27] D. Mink, W. B. Glisson, R. Benton, and K.-K. R. Choo, "Manipulating the five V's in the next generation air transportation system," in *Security and Privacy in Communication Networks*, X. Lin, A. Ghorbani, K. Ren, S. Zhu, and A. Zhang, Eds. Cham, Switzerland: Springer, 2018, pp. 271–282.

[28] M. Leonardi, M. Strohmeier, and V. Lenders, "On jamming attacks in crowdsourced air traffic surveillance," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 36, no. 6, pp. 44–54, Jun. 2021.

[29] G. Dave, G. Choudhary, V. Sihag, I. You, and K.-K.-R. Choo, "Cyber security challenges in aviation communication, navigation, and surveillance," *Comput. Secur.*, vol. 112, Jan. 2022, Art. no. 102516.

[30] D. Lundberg, B. Farinholt, E. Sullivan, R. Mast, S. Checkoway, S. Savage, A. C. Snoeren, and K. Levchenko, "On the security of mobile cockpit information systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 633–645.

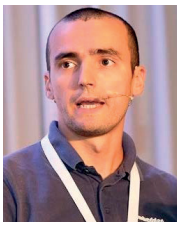
[31] H. Turtiainen, A. Costin, S. Khandker, and T. Hamalainen, "GDL90fuzz: Fuzzing—GDL-90 data interface specification within aviation software and avionics devices—A cybersecurity pentesting perspective," *IEEE Access*, vol. 10, pp. 21554–21562, 2022.

[32] L. Yusupov. (2021). *Lyusupov/UAT-Test-Signal*. Accessed: Feb. 24, 2021. [Online]. Available: https://github.com/lyusupov/UAT-test-signal

[33] S. Larroque. (2020). *Tomerfiliba/Reedsolomon*. Accessed: Mar. 15, 2021. [Online]. Available: https://github.com/tomerfiliba/reedsolomon/blob/master/reedsolo.py
[34] O. A. Osanaiye, A. S. Alfa, and G. P. Hancke, "Denial of service defence for resource availability in wireless sensor networks," *IEEE Access*, vol. 6, pp. 6975–7004, 2018.
[35] S. Ghildiyal, A. K. Mishra, A. Gupta, and N. Garg, "Analysis of denial of service (DOS) attacks in wireless sensor networks," *Int. J. Res. Eng. Technol.*, vol. 3, no. 22, pp. 140–143, Jun. 2014.
[36] P. Prakash, A. Abdelhadi, and M. Pan, "Secure authentication of ADS-B aircraft communications using retroactive key publication," 2019, *arXiv:1907.04909*.
[37] ICAO. (2003) *Manual for the Universal Access Transceiver (UAT)*. Accessed: Mar. 2, 2021. [Online]. Available: https://www.icao.int/safety/acp/Inactiveworkinggroupslibrary/ACP-WG-C-UA T-2/UAT-SWG02-WP04-DraftTechManualV0-1.pdf
[38] M. Muench, J. Stijohann, F. Kargl, A. Francillon, and D. Balzarotti, "What you corrupt is not what you crash: Challenges in fuzzing embedded devices," in *Proc. NDSS*, 2018, pp. 1–15.
[39] M. A. Garcia, J. Stafford, J. Minnix, and J. Dolan, "Aireon space based ADS-B performance model," in *Proc. Integr. Commun., Navigat. Surveill. Conf. (ICNS)*, 2015, pp. C2-1–C2-10.
[40] M. Garcia, J. Dolan, and A. Hoag, "Aireon's initial on-orbit performance analysis of space-based ADS-B," in *Proc. Integr. Commun., Navigat. Surveill. Conf. (ICNS)*, Apr. 2017, pp. 1–28.
[41] S. Kaul, "Smallsats, hosted payload, aircraft safety, and ADS-B navigation services," in *Handbook of Small Satellites: Technology, Design, Manufacture, Applications, Economics and Regulation*. Cham, Switzerland: Springer, 2020, pp. 1011–1027.
[42] A. Pahsa, P. Kaya, G. Alat, and B. Baykal, "Integrating navigation & surveillance of unmanned air vehicles into the civilian national airspaces by using ADS-B applications," in *Proc. Integr. Commun., Navigat., Surveill. Conf.*, May 2011, pp. J7-1–J7-7.
[43] M. Consiglio, B. J. Duffy, S. Balachandran, L. Glaab, and C. Munoz, "Sense and avoid characterization of the independent configurable architecture for reliable operations of unmanned systems," NASA, Washington, DC, USA, Tech. Rep., 2019. [Online]. Available: https://www.nasa.gov/sites/default/files/atoms/files/2019_consiglio_isaac_atm2019_tpsas_v9-508_0.pdf
[44] J. Pavur, M. Strohmeier, V. Lenders, and I. Martinovic, "In the same boat: On small satellites, big rockets, and cyber trust," in *Proc. 13th Int. Conf. Cyber Conflict (CyCon)*, May 2021, pp. 151–169.
[45] T. Kacem, A. Barreto, D. Wijesekera, and P. Costa, "ADS-Bsec: A novel framework to secure ADS-B," *ICT Exp.*, vol. 3, no. 4, pp. 160–163, Dec. 2017.
[46] M. Kassab, "Testing practices of software in safety critical systems: Industrial survey," in *Proc. 20th Int. Conf. Enterprise Inf. Syst.*, 2018, pp. 359–367.
[47] T. Li and B. Wang, "Sequential collaborative detection strategy on ADS-B data attack," *Int. J. Crit. Infrastruct. Protection*, vol. 24, pp. 78–99, Mar. 2019.
[48] X. Ying, J. Mazer, G. Bernieri, M. Conti, L. Bushnell, and R. Poovendran, "Detecting ADS-B spoofing attacks using deep neural networks," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Jun. 2019, pp. 187–195.
[49] K. Jansen, L. Niu, N. Xue, I. Martinovic, and C. Pöpper, "Trust the crowd: Wireless witnessing to detect attacks on ADS-B-based air-traffic surveillance," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2021, pp. 1–17.

SYED KHANDKER received the M.Sc. degree in web intelligence and service engineering from the University of Jyväskylä, Finland, in 2016, where he is currently pursuing the Ph.D. degree with the Faculty of Information Technology. Since his childhood, he has been a Radio Enthusiast and holds an Amateur Radio Operator License. His research interests include the field of RF fingerprint positioning, automatic dependent surveillance-broadcast, automatic identification systems, wireless communications, and artificial intelligence.

HANNU TURTIAINEN received the B.Sc. degree in electronics engineering from the University of Applied Sciences, Jyväskylä, Finland, and the M.Sc. degree in cybersecurity, in 2020. He is currently pursuing the Ph.D. degree in software and communication technology with the University of Jyväskylä. His research interests include machine learning and artificial intelligence in the cybersecurity and digital privacy field. He is also working in the IoT field as a Cybersecurity Engineer and a Software Engineer at Binare.io, a deep-tech cybersecurity spin-off from the University of Jyväskylä.



ANDREI COSTIN received the Ph.D. degree from EURECOM/Telecom ParisTech, in 2015, under co-supervision of Prof. Francilon and Prof. Balzarotti. He is currently a Senior Lecturer/an Assistant Professor in cybersecurity with the University of Jyväskylä, Finland, with a particular focus on IoT/firmware cybersecurity and digital privacy. He has been publishing and presenting at more than 45 top international cybersecurity venues, both academic (Usenix Security and ACM ASIACCS) and industrial (BalckHat, CCC, and HackInTheBox). He is the author of the first practical ADS-B attacks (BlackHat 2012) and has literally established the large-scale automated firmware analysis research areas (Usenix Security 2014)—these two works are considered seminal in their respective areas, being also most cited at the same time. He is also the CEO/Co-Founder of Binare.io, a deep-tech cybersecurity spin-off from the University of Jyväskylä, focused on innovation and tech-transfer related to IoT cybersecurity.

TIMO HÄMÄLÄINEN has over 25 years of research and teaching experience related to computer networks. He has lead tens of external funded network management related projects. He has launched and leads Master Programs with the University of Jyväskylä (currently SW and Communications Engineering) and teaches network management related courses. He has more than 200 internationally peer-reviewed publications and he has supervised 36 Ph.D. theses. His current research interests include wireless/wired network resource management (the IoT, SDN, and NFV) and network security.

PVIII

CYBERSECURITY ATTACKS ON SOFTWARE LOGIC AND ERROR HANDLING WITHIN AIS IMPLEMENTATIONS: A SYSTEMATIC TESTING OF RESILIENCE

by

S Khandker, H Turtiainen, A Costin, T Hamalainen 2022

IEEE Access, 10, 29493-29505

https://doi.org/10.1109/access.2022.3158943

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 21, 2021, accepted February 23, 2022, date of publication March 11, 2022, date of current version March 21, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3158943

Cybersecurity Attacks on Software Logic and Error Handling Within AIS Implementations: A Systematic Testing of Resilience

SYED KHANDKER, HANNU TURTIAINEN, ANDREI COSTIN, AND TIMO HÄMÄLÄINEN
Faculty of Information Technology, University of Jyväskylä, 40014 Jyväskylä, Finland

Corresponding author: Andrei Costin (ancostin@jyu.fi)

This work was supported in part by the Finnish Grid and Cloud Infrastructure (FGCI) (persistent identifier urn:nbn:fi:research-infras-2016072533), in part by the Decision of the Research Dean on Research Funding within the Faculty of Information Technology of the University of Jyväskylä, and in part by the Finnish Cultural Foundation under Grant Decision 00211119.

ABSTRACT To increase situational awareness of maritime vessels and other entities and to enable their exchange of various information, the International Maritime Organization mandated the use of the Automatic Identification System (AIS) in 2004. The AIS is a self-reporting system that uses the VHF radio link. However, any radio-based self-reporting system is prone to forgery, especially in situations where authentication of the message is not designed into the architecture. As AIS was designed in the 1990s when cyberattacks were in their infancy, it does not implement authentication or encryption; thus, it can be seen as fundamentally vulnerable against cyberattacks. This paper demonstrates and evaluates the impact of multiple cyberattacks on AIS via remote radio frequency (RF) links using transmission-enabled software-defined radio (SDR). Overall, we implemented and tested a total of 11 different tests/attacks on 19 AIS setups, using a controlled environment. The tested configurations were derived from heterogeneous platforms such as Windows, Android, generic receivers, and commercial transponders. Our aim is to enhance the early discovery of new vulnerabilities in AIS to effectively address AIS attacks in the nearest future. The results showed that approximately 89% of the setups were affected by Denial-of-Service (DoS) attacks at the AIS protocol level. Besides implementing some existing attack ideas (e.g., spoofing, DoS, and flooding), we showed some novel attack concepts in the AIS context such as a coordinated attack, overwhelming alerts, and logical vulnerabilities, all of which have the potential to cause software/system crashes in the worst-case scenarios. Moreover, an implementation/specification flaw related to the AIS preamble was identified during the experiments, which may affect the interoperability of different AIS devices. The error-handling system in AIS was also investigated. Unlike the aviation sector's Automatic Dependent Surveillance-Broadcast (ADS-B), the maritime sector's AIS does not effectively support any error correction method, which may contribute to RF pollution and less effective use of the overall system. The consistency of our results for a comprehensive range of hardware-software configurations indicated the reliability of our approach, test system, and evaluation results.

INDEX TERMS AIS, attacks, cybersecurity, DoS, maritime, resiliency, ship.

I. INTRODUCTION

TO facilitate the growing world trade, the number of commercial cargo carriers is increasing. Also, many other vessels share the same waterways, such as leisure boats, fishing boats, and passenger ships. To improve the safety of navigation and to avoid collisions, the International Maritime Organization (IMO) announced in 2004 the mandatory use of the Automatic Identification System [1]. AIS is an automatic tracking system that periodically transmits a ship's type, name, position, speed, and other data to nearby vessels and other maritime entities. It is a prevalent maritime situational awareness system used by approximately 570,000 vessels [2]. AIS uses a VHF radio link to transmit and receive signals. It is a self-reporting system wherein the trustworthiness of information depends on the data being reported by the vessel

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks.

rather than measured by radar. However, any self-reporting system over a radio link is exposed to security vulnerabilities due to the possibility of spoofing. To prevent spoofing, security measures such as authentication or encryption are necessary. However, the current AIS protocol does not utilize any authentication or encryption methods; therefore, it is exposed to serious cybersecurity threats.

There have been several incidents of exploitation of the AIS system. On June 19, 2021, an online ship tracking site showed that a British warship and a Dutch frigate were close to Sevastopol in Crimea, which escalated tension between Russia and Britain. However, the on-board cameras of both ships revealed that they were roughly 300 km away [3]. In another case, North Korean vessels altered the Maritime Mobile Service Identity (MMSI) of their fishing ship to evade sanctions. Under their false identity, they were found fishing near the coastal area of China [4]. Finally, unknown entities were falsely claiming to be the US or coalition warships near the strait of Hormuz [5]. Such reports indicate that AIS has already been exploited at the national or military level. Such sensitive dangerous security flaws have not yet been thoroughly investigated in academia. Only a handful of studies practically investigated the such flaws [6]. In the meantime, malicious attacks on AIS threaten to spread at the ordinary hacker level due to the proliferation of low-cost, transmission-enabled software-defined radio (SDR) technology that has made it possible to produce any radio signal at a low cost and effort. For example, our laboratory had two types of transmission-enabled SDR, HackRF and BladeRF. Each of them, though costing less than $500, was able to produce fake AIS signals. Missing basic security measures and the evolution of transmission-enabled SDR technology have forced this three-decades-old AIS technology to face unprecedented challenges from cybersecurity attacks. Nonetheless, all vessels in the vast waterways have to follow the current AIS protocol, which is insecure by default.

AIS receivers are also diversifying day by day. Besides the traditional on-board AIS setup, smartphone-based navigation applications are also broadly used. The 7 smartphone-based navigation applications used in this study were downloaded approximately 43,000 times from the Google play store, leaving alone other non-tested applications and iOS platform's download numbers aside. The navigation data are fed to the mobile application from the receiver through a WiFi connection. These smartphone-based receiving setups, due to their attractive graphical user interface, low cost, and ease of installation, are gaining popularity among private users. However, these types of portable receivers remain untested against cyberattacks, as shown in current literature. Lack of extensive study of AIS exploitation, insufficient study on the impact of cyberattacks on modern AIS setups, and our previous security experience on a similar aviation service (Automatic Dependent Surveillance-Broadcast (ADS-B) [7], [8]) have motivated us to conduct this study. The main contributions of this study are:

1) Some novel (and existing) attacking concepts on AIS – such as spoofing, jamming, Denial-of-Service (DoS), coordinated attack, collision alert, overwhelming alerts, logically invalid data encoding, man overboard, etc., – were practically implemented and evaluated over the radio link;
2) Logic vulnerabilities, error handling and coordinated attacks in AIS were studied for the first time (to the best of our knowledge); and
3) An important AIS preamble-related implementation flaw was identified and investigated.

The rest of this article is organized as follows. Details of the AIS technology are described in Section II. Related studies are discussed in Section III. Details of our test platform, attack implementation, and experimental setup are presented in Section IV. Our attacks, results, and analysis are explained in Section V. Some countermeasures to attacks are discussed in Section VI. Finally, possible workarounds, future studies, and conclusions are presented in Section VII.

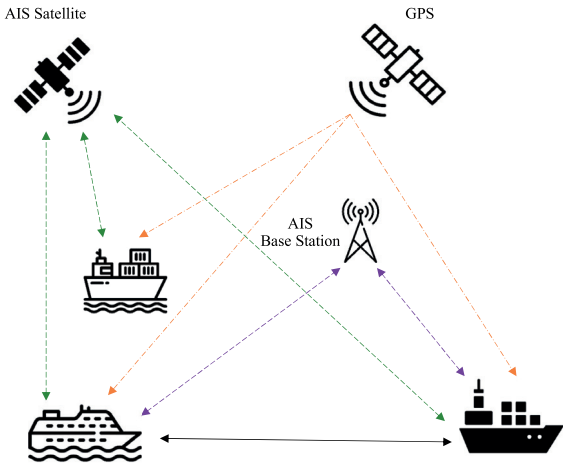


FIGURE 1. AIS communication architecture.

II. OVERVIEW OF AUTOMATIC IDENTIFICATION SYSTEM

The AIS is a worldwide automatic ship tracking system that based on fits into vessel as small transponders that periodically transmit the ship name, type, MMSI, speed, navigation status, and other useful information using the VHF radio signal. The most important information in the AIS data is the location information, which is generally obtained from the Global Navigation Satellite System (GNSS) such as the Global Positioning System (GPS). AIS uses two radio channels: channel A at 161.975 MHz (denoted as 87B) and channel B at 162.025 MHz (denoted as 88B). A dual-channel system is used to increase the link capacity and to minimize the RF interference. Using these two channels, ships can communicate with other ships, base stations, or other entities (e.g., stationary boys and men overboard). Due to the earth's

curvature and antenna height, the typical range of AIS signals is limited to approximately 40 nautical miles. If two ships cannot communicate directly, they can still exchange information via a base station or satellite called SAT-AIS. However, SAT-AIS is not yet fully operational across the globe. To date, countries such as USA, Canada, Norway, and India launched a few satellites to conduct full-scale research on SAT-AIS. Figure 1 shows the AIS communication concept. It enables different types of vessels to exchange information directly or via a base station or a satellite. There are mainly five types of AIS devices:

- Class A uses the Self-Organized Time Division Multiple Access (SOTDMA) scheme. Its nominal transmission power is 12.5 watts. It is mainly used by large commercial vessels;
- Class B uses the Carrier Sense Time Division Multiple Access (CSTDMA) scheme. It is used for lighter commercial and leisure vessels with its 2 watts transmission power;
- Base station, situated at the shore side, provides AIS channel management, time synchronization, text messages, navigation information, and meteorological and hydrological information;
- Aids to Navigation (AtoN), a shore- or buoy-based transceiver, is designed to collect and transmit data related to sea and weather conditions and to relay AIS messages so as to extend the network coverage;
- Search and rescue transceiver (SART), an emergency distress beacon that assists in homing to itself (i.e., lifeboats and life rafts). It transmits a text broadcast using message type 14.

The AIS uses the Time Division Multiple Access (TDMA) channel access method, which means the time unit is divided into many slots [9]. Generally, each time slot or frame can accommodate a single AIS message. A frame is 256 bits, and the standard data transmission rate is 9,600 bits/second. Therefore, each frame has a timing limit of 26.66 milliseconds, which results in 2,250 slots per minute per channel or 4,500 slots per minute in both channels. Sometimes, multiple frames can be used for a single message. AIS frames are transmitted into the air using Gaussian Minimum Shift Keying (GMSK) modulation with the bandwidth-time (BT) product set at 0.4. The data must be encoded using the Non-Return to Zero Inverted (NRZI) format before transmission. Figure 2 shows the basic structure of an AIS frame.

Ramp up 8 bit	Preamble 24 bit	Start flag 8 bit	Payload 168 bit	CRC 16 bit	Stop flag 8 bit	Buffer 24 bit

FIGURE 2. The basic structure of an AIS frame.

There are 64 types of AIS messages, of which 27 are currently in use. The rest (37) are reserved for the future. Some of the most important AIS message types are listed below.

- 1 = Position report of class A
- 4 = Base station report
- 14 = Safety-related broadcast message
- 18 = Standard class B position report
- 20 = Data link management message
- 21 = Aid-to-Navigation report
- 22 = Channel management
- 23 = Group assignment command
- 24 = Static data report

The full list and details of the AIS message types are available in [10]. Despite offering many valuable services for ship navigation, AIS falls short in security measures. Its main problem is that it does not utilize authentication or encryption; therefore, any attacker with the proper knowledge of generating valid AIS protocol signals can impact the AIS communication.

III. RELATED STUDIES

Mathapo [11] implemented an SDR-based AIS receiver as a proposed payload of the South African ZA-002 satellite in 2007. The proposed receiver can be used to track and store the movement of ships at sea and then forward this information to the ground station upon request. C++ programming language was used to implement the AIS receiver on the SDR architecture. The SDR AIS receiver was capable of high-pass filtering, amplifying, symbol synchronizing, decoding, bit destuffing, error checking, translating, and interpreting the AIS messages in real time. The results showed that the AIS messages were decoded correctly. Larsen *et al.* [12] also demonstrated their SDR-based AIS receiver for the Danish AAUSAT3 satellite. The receiver down-converts a 200KHz-wide frequency spectrum of around 162 MHz to a 200 KHz intermediate frequency (IF) signal, then samples it to an in-phase and quadrature components (IQ) signal at a speed of 1 mega-samples-per-second. Later, the IQ data is filtered into the two AIS channels. Then each channel is demodulated using matched filter implementation. The transmitted bits are estimated by recovering the bit-synchronization using a correlator to find the training sequence. The authors tested their receiver using a stratospheric balloon flight at a 24km altitude. The test results showed that AIS can be received from approximately 500km away. The first SDR-based AIS attacks were demonstrated in 2014 by Balduzzi *et al.* [6]. The authors developed a Python language-based program called *AISTX* [13] to create an AIS payload according to the protocol. GNU Radio Companion (GRC) was used to generate the IQs of the signal, which were transmitted into the air using the Universal Software Radio Peripheral (USRP). Three different receivers verified the reception of the counterfeit transmission. The AIS protocol specifications were affected by several threats and were vulnerable to cyberattacks such as spoofing, false collision threat, and service availability disruption.

Marques *et al.* [14] built a low-cost AIS transponder using a HackRF. The authors encoded the message with [15] and used *AISTX* to construct the final AIS frame. They tested the transmission of type 1 messages via an RF link. Another SDR-based receiver with a chart-plotting software called OpenCPN was used to test the reception. The main

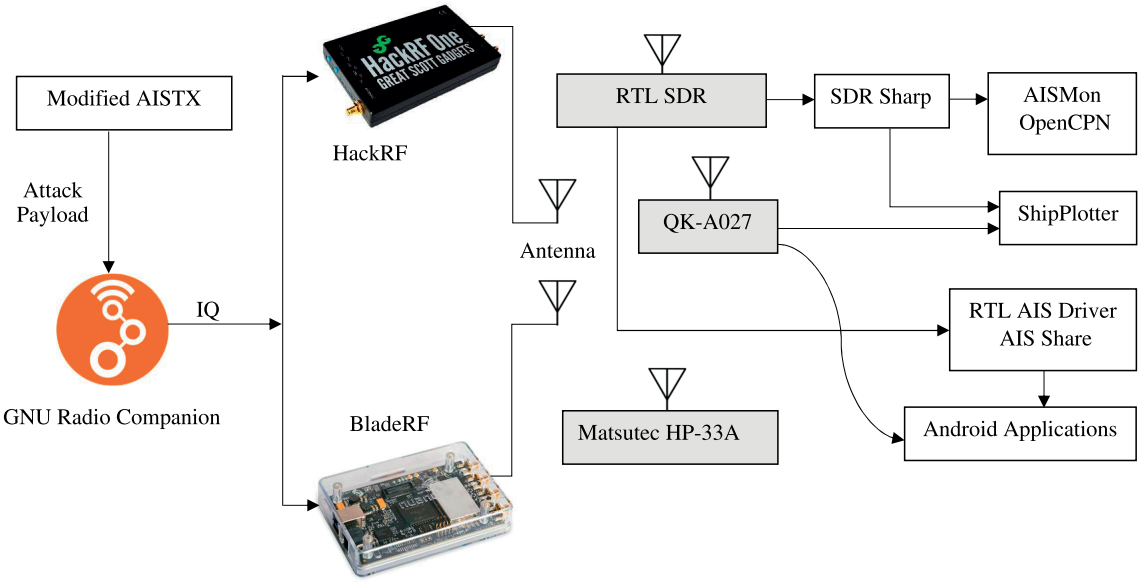


FIGURE 3. Experimental attack setup.

intention of their research was to develop a low-cost (around 150 euros) AIS transponder. A similar test was conducted by Cruz *et al.* [16]. Using transmission-enabled SDR, GRC, and *AISTX*, the authors transmitted the position of a ship. At the receiver end, they calculated the difference between the transmitted data and the received data. They found on average 7-meter difference in the ship's location between the original data and the received data. They found that the technical standard requires four decimal places in latitude and longitude, but the GPS device that they used provided three decimal places, which caused a small calculation error. Foster [17] developed an AIS decoder based on the Python programming language to report the shipborne position called "gr-ais". It can be used with GRC and other chart-plotting software. The software can be used with any SDR that provides IQ data, such as RTL-SDR and HackRF. Some other decoders were also developed [18], [19]. Attacks on ships are not limited to fake AIS transmissions. A research team from the University of Texas hijacked an $80 million yacht with cheap GPS spoofing [20]. In another test, the GPS spoofing resulted in an unbelievable ship speed [21]. Androjna *et al.* [22] studied AIS vulnerabilities and challenges. They thoroughly analyzed a spoofing event that happened near Elba in December 2019. The fake signals created a dangerous situation for a real ship. More than a dozen fake ships were found on a collision course of that ship. They concluded that the maritime industry is neither immune to cyberattacks nor fully prepared for the risks associated with the use of modern digital systems.

So far, only Balduzzi *et al.* [6] in 2014 have demonstrated attacks using AIS packet data. Other studies focused on the SDR implementation of AIS, decoding, or possibilities for cheap transponder build-up. Technology has drastically changed since the study of Balduzzi *et al.* [6] study. Many new types of receivers, software, chart-plotting applications using smart devices have been developed. Attackers have new tools and ideas as well. Therefore, evaluating the attacks on modern AIS setups against current technology is essential.

TABLE 1. List of hardware.

Hardware	Functionality
Matsutec HP-33A	Stand alone AIS transponder with GPS reception
Quark-elec QK-A027	AIS receiver. Share AIS data via USB and WiFi
RTL-SDR	AIS signal receiver
HackRF	Generate fake AIS RF signal
BladeRF	Generate fake AIS RF signal
Samsung A 21s	Navigation applications' host

IV. EXPERIMENT SETUP

Many types of AIS hardware and software are available in the market. Some hardware has built-in display functionality, while others display the AIS data through external software and additional accessories. In most cases, we found the hardware and software are interoperable. For the purpose of our study, a compatible combination of hardware and software makes an AIS setup for the evaluation. In this study, a total of 19 combinations of AIS setups were tested against AIS attacks. The hardware and software were selected

TABLE 2. List of software.

Software	Platform	Functionality
OpenCPN v 5.2.4	Windows	Displaying AIS data
iRegatta v 4.07	Android	Displaying AIS data
Ships v 4.07	Android	Displaying AIS data
Boating v 17.0.2	Android	Displaying AIS data
iBoating v 190.0	Android	Displaying AIS data
Boat Beacon v 2.53	Android	Displaying AIS data
AF track v 12.7	Android	Displaying AIS data
OpenCPN v 1.0.5	Android	Displaying AIS data
SDR sharp v 1.0.0.1732	Windows	Receiving AIS signal
RTL AIS driver v 1.1.8	Android	Decoding AIS data
AIS share v 1.2.2	Android	Sharing AIS data
ShipPlotter v 12.5.4.6	Windows	Decoding and displaying AIS data
AISmon v 2.2.0	Windows	Decoding and sharing AIS data

comprehensively to test the attacks on diverse AIS setups, yet at the same time the diversity of the setups was limited by the budget limitations, as well as market availability of certain products at the time of the experiments. The list of hardware and software used and their functionalities are presented in Tables 1 and 2, respectively. For this experiment, we have also acquired an official MMSI number from TRAFICOM (Finland). For privacy and safety reasons, we blurred the MMSI number in some figures. Some past studies used *AISTX* as an AIS payload generator. The original version of *AISTX* produced a single AIS frame at a time, when testing of some attacks such as DoS or flooding it was very slow. For the demands of this study, we modified *AISTX* so that it could produce N number of AIS frames from a single command in a single file. Linux-based GRC was used to produce the IQ samples based on the data of the file. Finally, the IQ samples were provided to the HackRF and BladeRF for the transmission of an AIS RF signal. To verify the reception and impact of the attacks, we used the Matsutec HP-33A AIS transponder, Quark-elec QK-A027 AIS receiver, Windows-based ShipPlotter [23] and OpenCPN [24] software, and several Android mobile applications. Apart from the HP-33A transponder which is an all-in-one complete setup, all other setups used QK-A027 and RTL-SDR as the RF front-end. In Windows, we used SDR Sharp to tune the AIS frequency and provided the resulting audio to AISMon [25], which decoded the AIS signal. The signal was subsequently fed to the OpenCPN using a UDP port in the National Marine Electronics Association (NMEA) format. ShipPlotter had a built-in decoder. In Android, the decoding task was done by the RTL AIS driver application. The decoded messages were shared by AIS Share with different navigation applications [26]. The QK-A027 receiver has a built-in decoder and could provide the decoded AIS data to the other application using a TCP port. Therefore, while QK-A027 was used, the Android applications did not need other decoding software. *Ships v 4.07* did not work with QK-A207 because that application does not support a TCP connection. Figure 3 shows the experimental attack setup and how the AIS payload reaches the receiver over the air. Modified *AISTX* supplied the payload to GRC, where base-band IQs of the signal are generated according to the GMSK modulation. The IQs are transmitted into the air using HackRF and BladeRF. AIS receivers receive the signal from the air and demodulate, decode, and display the AIS data. Figure 4 shows the GRC transmission script. We maintained the 9,600 bits-per-second rate according to the AIS protocol. Therefore, 2 mega-samples-per-second resulted in around 208 samples/symbol. The bandwidth-time is the product of the duration of a signal and its spectral width, which was set at 0.4. By changing the channel frequency, we were able to transmit on both AIS channels. Moreover, through multiple transmission-enabled SDRs, we were able to simultaneously transmit the AIS signal in both channels.

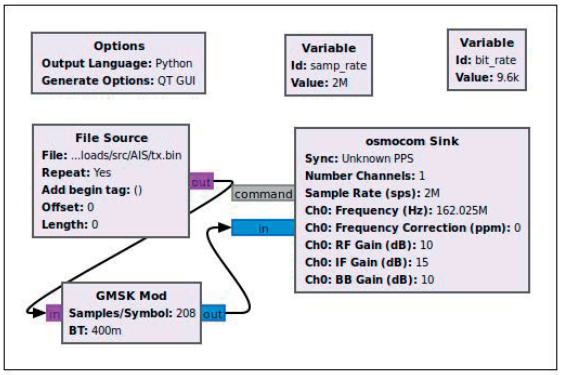


FIGURE 4. GRC flow-graph for AIS injections.

V. ATTACKS AND RESULTS

This section describes different types of attacks on AIS and the observed impact on the receiving devices and applications.

A. SPOOFING

We were able to produce spoofed or fake ships, as has already been done before [6]. The spoofed ship was visible on all the receivers, including the commercial AIS transponder. A spoofed ship may have severe consequences [3] and may jolt the safety of navigation in waterways. Figure 5 shows a spoofed ship.

B. MAN OVERBOARD

Man Overboard (MOB) is a survivor recovery alert system used when a crew member or a passenger falls off the ship into the water or used by rescue workers in any ship evacuation operation. It indicates that a human is in the water and needs immediate rescue. It is a small beacon that transmits to the neighboring vessels a distress AIS signal containing the beacon's GPS location (if available). The MOB signal

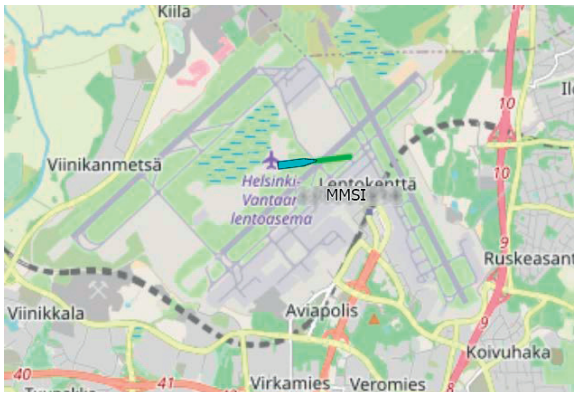


FIGURE 5. A spoofed maritime vessel as-if present on the runway of Helsinki airport in an android ship application.

uses a type 1 message, navigation status 14, and a 9-digit MMSI starting with 972. It alerts the ship's radar system that there is an emergency. Using our aviation and maritime pentesting platform, we were able to produce a fake MOB alert. Attackers using this type of counterfeit signal can waste a significant amount of time of any ship by engaging the ship in a vain and costly rescue operation. Figure 6 shows a MOB alert in the Matsutec HP-33A transponder due to the fake distress signal.

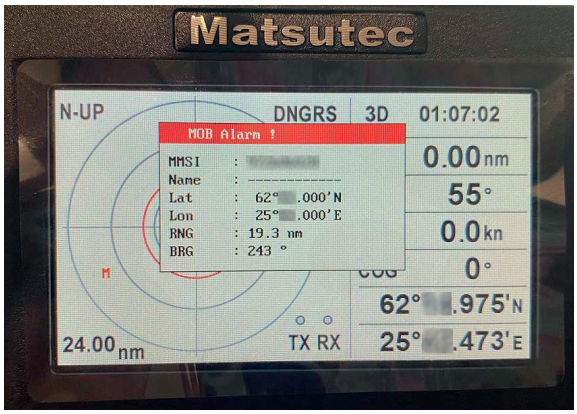


FIGURE 6. MOB alert in the Matsutec HP-33A AIS transponder.

C. COLLISION ALERT

Since it is possible to fake a ship's location, we placed a fake ship near another ship to observe the reaction. We observed that when the closest point of approach (CPA) and the time to the closest point of approach (TCPA) values fell behind the threshold, the ship was alerted about a possible collision. Attackers may use this type of attack to change the course of a target vessel. Figure 7 shows a collision alert.

FIGURE 7. Collision alert in ShipPlotter.

D. JAMMING

Jamming in AIS can be divided into two categories: RF jamming, and (display) flooding. RF jamming is the transmission of overpowered white noise to the AIS channels in such a way that the valid packets cannot be transmitted through the channel. We tested RF jamming in our laboratory. Although it worked, the effectiveness of this type of attack in vast water areas would be limited. We were able to jam the channels with valid AIS packets. According to the AIS's TDMA scheme, each AIS channel has 2,250 available time slots to receive AIS signals from a maximum of 2,250 different ships every minute. Each unique MMSI is counted as a different ship. Using the modified *AISTX*, we generated two files that contained a huge amount of AIS frames with different MMSI. Using two transmission-enabled SDRs, we transmitted those two files to both AIS channels. Figure 8 shows a screenshot of the AIS reception statistics in the AIS Share application. Two consequent attackers using SDRs consumed approximately 96% of channel A's capacity and 100% of channel B's capacity. Thus, the channels can be jammed by the attacker with valid messages.

FIGURE 8. Signal receiving statistics of both channels for one minute in AIS Share.

E. OVERWHELMING ALERTS

Usually, AIS receivers raise an alert when there is a possibility of collision with another ship based on the CPA and TCPA thresholds or in a distress situation like MOB. We took

advantage of this feature to trigger a huge number of alerts in a short period of time. We set 1,000 fake ships very near the receiver's *Own ship* location to trigger the collision alert. Similarly, in another test, we transmitted 1,000 MOB distress signals. We found that a large number of simultaneous alerts led to a crash of the ShipPlotter software, which did not crash under normal circumstances when exposed to the AIS DoS attack (see V-I). When an overwhelming number of collision alerts, similar to Figure 7, started to pile up on the user interface, the software eventually collapsed, hence affecting the availability of a mission-critical software and display. Some applications do not present any alert at all, which is quite dangerous and exposes the ship to unnecessary risk. Relying on this type of poor software may lead to accidents and other unexpected scenarios. We also noticed that the overwhelming number of alerts lead to a situation in some software where thousands of audio-visual alerts create a chaotic situation called "alert fatigue" [27]. A true positive alert may go unnoticed in such a chaotic situation. Table 3 summarizes the result of our alert-handling tests. Certainly, similarly to traffic collision avoidance system (TCAS) and cockpit systems [28], the AIS users (e.g., port and ship crews) may switch off the AIS receivers in case of malicious attacks via overwhelming alerts. However, such switching off, in essence, means the AIS is under DoS, and the entire benefits of AIS (e.g., navigational awareness and communication) is completely lost.

F. COORDINATED ATTACK

Among the data fields in an AIS message, the MMSI number is used as the main reference by the receiving software. In successive messages, information on a particular ship is updated against this MMSI number. To avoid a conflict with other ships' MMSIs, some commercial transponders allow the MMSI to be set only once. However, since our pentesting platform can use any MMSI number, multiple emitters can be used to transmit different AIS signals containing the same MMSI number. During this type of attack, the attackers coordinate among themselves to send multiple signals that contain the same reference (the MMSI number) but differing values in some of the AIS data fields. This is called a "coordinated attack." Since the reference point is the same, the data fields will be updated according to the encoded message of multiple signals in the receiving software. However, some fields in AIS should be updated by following a standard or a common pattern. For example, the position of the ship should be updated smoothly with a clear course, possibly with a historical fading-out path. However, in a coordinated attack, the attackers, using multiple emitters, can change the position of the ship from one place to another in an instant. Cargo carriers may turn into passenger carriers. The ship's name, call sign, dimension, and other data may differ in a second. These can lead to confusion in Vessel Traffic Services (VTS) or among other ships, thus producing dangerous consequences. We focused on the most important data fields of AIS, for example, the ship name, call sign, position, speed, navigation status, vessel type, and dimension. We observed that in all the receivers, the ship's information fluctuates every second, that is, alternates between different transmitters' signal values. Table 4 shows the results of the coordinated attack. In practice, it is possible to carry out a coordinated attack even with a single attacking emitter since the legitimate ship itself can be the second emitter. In this case, the attacking emitter mimics the target ship's MMSI number and alters the other values, thus achieving effects similar to those when there are two coordinated attackers.

TABLE 3. Test results: overwhelming alerts.

Configuration		Observation	
Hardware	Software	Collision alert	MOB alert
Matsutec HP-33A	Matsutec Firmware v 1.0	All alerts in a list with one audio alert	All alerts in a list with one audio alert
RTL-SDR	ShipPlotter v 12.5.4.6	Many audio and visual alerts prompted to a software **crash**	**No alert**
	OpenCPN v 5.2.4	**Alert fatigue**	**Alert fatigue**
	iRegatta v 4.07	**No alert**	**No alert**
	Ships v 4.07	**No alert**	**No alert**
	Boating v 17.0.2	**Alert fatigue**	Red color MOB locations and one audio alert
	iBoating v 190.0	**No alert**	**No alert**
	Boat Beacon v 2.53	**Alert fatigue**	**Alert fatigue** Red color MOB locations
	AF track v 12.7	**No alert**	Red color MOB locations
	OpenCPN v 1.0.5	**Alert fatigue**	**Alert fatigue**
QK-A027	ShipPlotter v 12.5.4.6	Many audio and visual alerts prompted to a software **crash**	**No alert**
	OpenCPN v 5.2.4	**Alert fatigue**	**Alert fatigue**
	iRegatta v 4.07	**No alert**	**No alert**
	Ships v 4.07	Did not work	Did not work
	Boating v 17.0.2	**Alert fatigue**	Red color MOB locations and one audio alert
	iBoating v 190.0	**No alert**	**No alert**
	Boat Beacon v 2.53	**Alert fatigue**	**Alert fatigue** Red color MOB locations
	AF track v 12.7	**No alert**	Red color MOB locations
	OpenCPN v 1.0.5	**Alert fatigue**	**Alert fatigue**

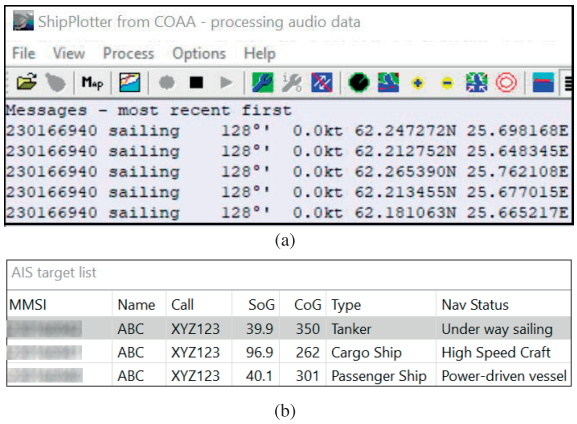
G. LOGICALLY INVALID DATA ENCODING

We noticed that there is no data validity checking in AIS. It is possible to form technically correct but logically invalid messages. For example, in Figure 9(a), ShipPlotter decodes that a ship (with the same MMSI number) went from one place to another place, but its speed remained zero even though it was sailing. This type of irrationality among data may open the opportunity for an attack. For example, if the speed remains zero, a TCPA alert would not be triggered. Figure 9(b) shows that multiple ships engaged in different activities have different speeds and courses, but all of them have the same name and call sign. Such a situation could be confusing for VTS or other ships if the MMSI is not checked carefully. However, maintaining the same MMSI number through a coordinated attack can result in a more complex situation, which we describe in V-F. Nonetheless, for these logically invalid data, no receiver provided any alarm during our experiment.

TABLE 4. Test results: coordinated attack.

Configuration		Effects						
Hardware	Software	Ship name	Call sign	Vessel type	Navi. status	Speed	Position	Ship dimension
Matsutec HP-33A	Matsutec Firmware v 1.0	FLC	FLC	FLC	FLC	FLC	FLC	FLC
RTL-SDR	ShipPlotter v 12.5.4.6	FLC	FLC	FLC	FLC	FLC	FLC	FLC
	OpenCPN v 5.2.4	FLC	FLC	FLC	FLC	FLC	FLC	FLC
	iRegatta v 4.07	FLC	INA	INA	INA	FLC	FLC	INA
	Ships v 4.07	FLC	FLC	FLC	FLC	FLC	FLC	FLC
	Boating v 17.0.2	FLC	FLC	INA	FLC	FLC	FLC	FLC
	iBoating v 190.0	FLC	FLC	FLC	FLC	FLC	FLC	FLC
	Boat Beacon v 2.53	FLC	INA	FLC	FLC	FLC	FLC	FLC
	AF track v 12.7	FLC	FLC	FLC	FLC	FLC	FLC	FLC
	OpenCPN v 1.0.5	FLC	FLC	FLC	FLC	FLC	FLC	FLC
QK-A027	ShipPlotter v 12.5.4.6	FLC	FLC	FLC	FLC	FLC	FLC	FLC
	OpenCPN v 5.2.4	FLC	FLC	FLC	FLC	FLC	FLC	FLC
	iRegatta v 4.07	FLC	INA	INA	INA	FLC	FLC	INA
	Ships v 4.07	DNW	DNW	DNW	DNW	DNW	DNW	DNW
	Boating v 17.0.2	FLC	FLC	INA	FLC	FLC	FLC	FLC
	iBoating v 190.0	FLC	FLC	FLC	FLC	FLC	FLC	FLC
	Boat Beacon v 2.53	FLC	INA	FLC	FLC	FLC	FLC	FLC
	AF track v 12.7	FLC	FLC	FLC	FLC	FLC	FLC	FLC
	OpenCPN v 1.0.5	FLC	FLC	FLC	FLC	FLC	FLC	FLC

Note: FLC = Fluctuates (i.e., displays alternate values from different attackers); INA = Information Not Available in the application; DNW = Did Not Work.



(a)

AIS target list

MMSI	Name	Call	SoG	CoG	Type	Nav Status
[redacted]	ABC	XYZ123	39.9	350	Tanker	Under way sailing
[redacted]	ABC	XYZ123	96.9	262	Cargo Ship	High Speed Craft
[redacted]	ABC	XYZ123	40.1	301	Passenger Ship	Power-driven vessel

(b)

FIGURE 9. (a) Zero speed ship moving position in ShipPlotter. (b) Different ships having the same Name and call sign in OpenCPN.

H. VISUAL NAVIGATION DISRUPTION

Ships generally navigate in water by following an AIS-supported plotted chart or radar screen. We found that this visual navigation can be significantly disrupted by fake AIS transmissions. For example, message type 4 is reserved for the AIS base station, which is usually stationary on the shore. However, in some type 4 messages, we changed the coordinate value linearly but kept the same MMSI, so it appeared that the base station was moving towards the ship along the navigation line of the ship itself. In another setting, using several different MMSI and geo-coordinate values, we surrounded a ship with stationary base stations. Figure 10 shows a scenario wherein a ship is surrounded by base stations. This type of situation may cause the operator of the ship to experience serious situational awareness confusion.

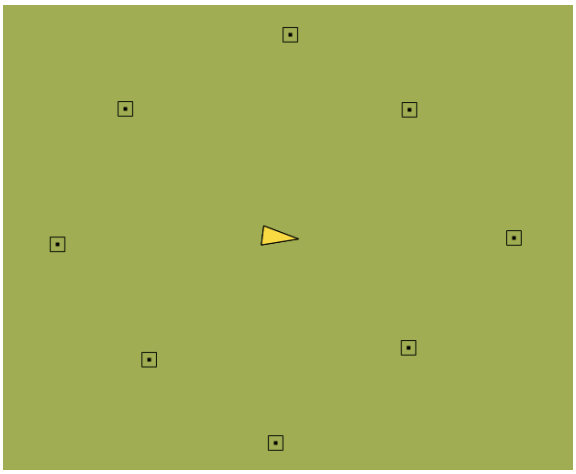


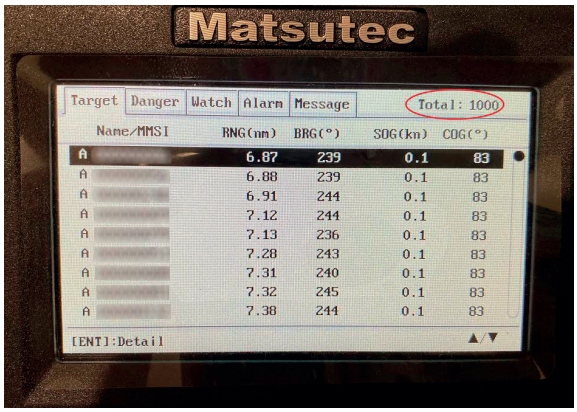
FIGURE 10. Ship surrounded by fake base stations in OpenCPN.

I. DENIAL OF SERVICE

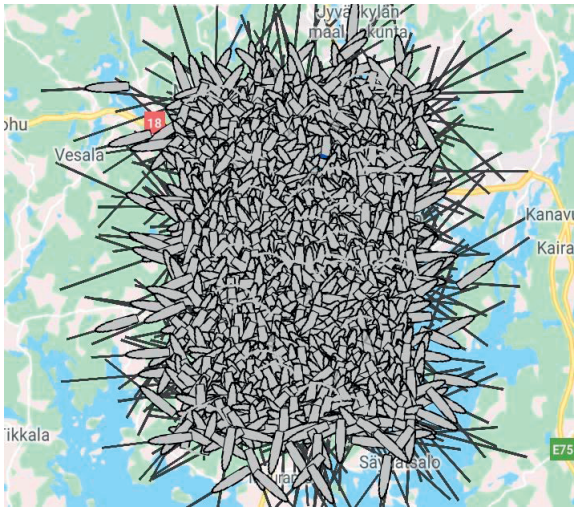
To test the system resiliency, we performed our DoS attack by trying to "overpower" the receivers with AIS signals. Our modified *AISTX* could produce *N* number of AIS frames at a time. We produced 200K type 1 frames in a single file and transmitted it through an SDR. The results in Table 5 show that approximately 89% of the configurations were impacted by the DoS attack (e.g., the output was clogged, unresponsive, or crashed). Some software did not crash during the AIS DoS attack because they decoded only a limited number of distinct MMSIs (e.g., up to 1,000 for Matsutec HP-33A). However, this type of behavior indicates the possibility of a legitimate message drop. Some setups follow a moderate decoding cycle to optimize memory and displaying capacity. However, the attack floods the display, so it becomes nearly impossible to read the screen. Figure 11(a) shows that the Matsutec transponder was clogged at its maximum decoding capability of 1,000 ships. Figure 11(b) shows the flooded screen of an Android application.

J. ERROR HANDLING TEST

According to the technical characteristics of AIS, cyclic redundancy check (CRC) is used for error detection [9]. If an error is detected, that message is dropped, assuming possible corruption. To test the error detection system, we deliberately flipped a message bit and repeatedly transmitted that file with a HackRF. We observed that all the receivers dropped the erroneous message. Figure 12 shows the error detection in AISMon, where all the messages were detected as erroneous messages. Error detection worked well across all the tested setups. However, none of them alerted the user regarding the erroneous packets. Error correction and alerting are optional in the standard. For many reasons, an error can occur; for example, we found that sometimes, due to a slight frequency offset in the receiver, a valid AIS frame was regarded as invalid. Alerting users upon error detection could help them to



(a)



(b)

FIGURE 11. (a) Clogged screen as effect of AIS DoS attack on Matsutec HP-33A. (b) Flooded screen as effect of AIS DoS attack on boat beacon.

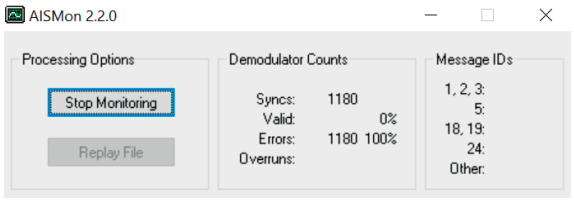


FIGURE 12. Detection of errors in AIS frames by AISMon.

calibrate the system for better-quality service. Moreover, in a congested RF channel, signal distortion is a regular incident, which can induce an error. Therefore, besides error detection, ADS-B uses an error correction system. An error correction scheme can save significant RF pollution. Thus we suggest at least a 1-bit error correction system in AIS.

K. AIS PREAMBLE TEST

During some early experiments, we noticed that one of our testbed devices (QK-A027) misbehaved. It displayed data from some transponders (e.g., Matsutec) but did not display valid non-attack AIS signals from our pentest platform. At the same time, the same signal was displayed perfectly fine in the Matsutec-based and RTL-SDR-based setups. This prompted us to investigate further this root cause of the QK-A027's misbehavior, which led us to discover what we call the "AIS preamble-related implementation flaw".

AIS uses a 24-bit preamble consisting of alternating zeros and ones (0101….). ITU-R M.1371-5 [9] in Annex 2 titled "Technical characteristics of an automatic identification system using time division multiple access techniques in the maritime mobile band" (in Figure 13(a)) specifies that the preamble (or training sequence) can start with 1 or 0 when transmitting. At the same time, the same document in Annex 7 titled "Class B automatic identification system using carrier sense time division multiple access technology" (in Figure 13(b)) specifies that the preamble always starts with 0 when transmitting. Annex 2 focuses on the SOTDMA channel access scheme, which is mainly used by class A vessels, while Annex 7 focuses on the CSTDMA scheme

TABLE 5. Test results: DoS attack.

Configuration		Observation
Hardware	Software	
Matsutec HP-33A	Matsutec Firmware v 1.0	**Output clogged** Maximum 1,000 MMSI displaying capacity
RTL-SDR	ShipPlotter v 12.5.4.6	No impact
	OpenCPN v 5.2.4	**Unresponsive** After approx. 30 minutes
	iRegatta v 4.07	**Crashed**
	Ships v 4.07	**Unresponsive** After approx. 20 minutes
	Boating v 17.0.2	**Output clogged** Maximum 100 MMSI displaying capacity
	iBoating v 190.0	**Output clogged** Maximum 1,200 MMSI displaying capacity
	Boat Beacon v 2.53	**Unresponsive** After approx. 20 minutes. Crashed sometimes.
	AF track v 12.7	**Output clogged** Approx. 11 minutes decoding cycle. Remove old decoded data when new cycle starts.
	OpenCPN v 1.0.5	**Unresponsive** After approx. 10 minutes. Crashed sometimes.
QK-A027	ShipPlotter v 12.5.4.6	No impact
	OpenCPN v 5.2.4	**Unresponsive** After approx. 30 minutes
	iRegatta v 4.07	**Crashed**
	Ships v 4.07	Did not work
	Boating v 17.0.2	**Output clogged** Maximum 100 MMSI displaying capacity
	iBoating v 190.0	**Output clogged** Maximum 1,200 MMSI displaying capacity
	Boat Beacon v 2.53	**Unresponsive** After approx. 20 minutes. Crashed sometimes.
	AF track v 12.7	**Output clogged** Approx. 11 minutes decoding cycle. Remove old decoded data when new cycle starts.
	OpenCPN v 1.0.5	**Unresponsive** After approx. 25 minutes. Crashed sometimes.

primarily used for class B vessels. Nonetheless, according to the requirement [9], regardless of class, all the vessels (in the range) should be visible to each other via AIS. Therefore, the International Electrotechnical Commission (IEC) standards IEC 62287-1 [29], IEC 62287-2 [30], and IEC 61993-2 [31] require all *Equipment Under Test* to receive the signal regardless of where the preamble starts with 0 or 1. Now, allowing the transmission preamble to start with 1 and expecting the reception preamble to start with 0 may induce confusion. The QK-A027 case can be an example. We observed that the QK-A027 receiver did not receive messages when the preamble in the messages started with 1. However, it detected the message when the preamble started with 0. The details of the AIS preamble test are presented in Table 6.

2.5 Training sequence

Data transmission should begin with a 24-bit demodulator training sequence (preamble) consisting of one segment synchronization. This segment should consist of alternating zeros and ones (0101....). This sequence may begin with a 1 or a 0 since NRZI encoding is used.

2.6 Data encoding

The NRZI waveform is used for data encoding. The waveform is specified as giving a change in the level when a zero (0) is encountered in the bit stream.

(a)

4.2.1.4 Training sequence

Data transmission should begin with a 24-bit demodulator training sequence (preamble) consisting of one segment synchronization. This segment should consist of alternating zeros and ones (0101....). This sequence always starts with a 0.

4.2.1.5 Data encoding

The NRZI waveform is used for data encoding. The waveform is specified as giving a change in the level when a zero (0) is encountered in the bit stream.

(b)

FIGURE 13. (a) AIS preamble instruction in ITU-R M.1371-5, Annex 2. (b) AIS preamble instruction in ITU-R M.1371-5, Annex 7.

Certainly, the way the specification is drafted and the lack of cautionary notes can easily mislead system designers, developers, and integrators. This finding would mean in practice that there is a very high probability that a ship equipped with a QK-A027 may not detect some other ships on the AIS displays. Therefore, this could lead to a possible collision or a similar accident, especially in low-visibility situations. Moreover, this problem is not necessarily isolated to QK-A027. In fact, we fear that similar misinterpretations and implementation flaws could have been made by other vendors or in other models similar to QK-A027 from the same vendor. In practice, this means that extensive retesting and revalidation of a large number of devices are required, with particular application of the methodology proposed in this article. While investigating this issue, we also studied the impact of NRZI conversion on the AIS data and preamble. We present those results in Appendix A for technical completeness.

VI. DEFENCE AGAINST AIS ATTACKS

Attacks on AIS can affect information confidentiality, authenticity, integrity, availability, possession, and utility [32], which need to be prioritized. However, implementing a proper defense strategy requires more systematic research and development, which is beyond the core focus of this paper. Nonetheless, in a similar service in aviation (ADS-B), we showed a received signal strength (RSS) and distance relationship model [8] that reached up to 90% accuracy in the best case. This strategy can easily be transferred to AIS, as the concept is protocol agnostic. We cannot create yet an RSS-Distance model for the AIS service because we are not in the close vicinity of a real-world port where realistic AIS traffic is seen. In fact, from our location (the central part of Finland), we do not receive any AIS messages. In the current literature, researchers have suggested some solutions.

To identify erroneous transmissions or falsified data, Ray *et al.* [33] proposed checking the integrity of the AIS data. According to them, the integrity can be assessed by comparing the AIS data with long-term historical data on the message with respect to other messages, and on the signal itself with its physical characteristics. Their proposed method is supposed to provide an integrity-based confidence coefficient on data that can be used to take further action on that data.

Junior *et al.* [34] showed a triggering mechanism that uses an image processing template matching technique to detect specific patterns transmitted by an attacker. Chart plotter software (e.g., OpenCPN) plots the ship on the map as it receives the data through AIS. This plotting changes the mean

TABLE 6. AIS signal detection status depending on the preamble start.

Configuration		Preamble	
Hardware	Software	Starts with 1	Starts with 0
Matsutec HP-33A	Matsutec Firmware v 1.0	✓	✓
RTL-SDR	ShipPlotter v 12.5.4.6	✓	✓
	OpenCPN v 5.2.4	✓	✓
	iRegatta v 4.07	✓	✓
	Ships v 4.07	✓	✓
	Boating v 17.0.2	✓	✓
	iBoating v 190.0	✓	✓
	Boat Beacon v 2.53	✓	✓
	AF track v 12.7	✓	✓
	OpenCPN v 1.0.5	✓	✓
QK-A027	ShipPlotter v 12.5.4.6	×	✓
	OpenCPN v 5.2.4	×	✓
	iRegatta v 4.07	×	✓
	Ships v 4.07	–	–
	Boating v 17.0.2	×	✓
	iBoating v 190.0	×	✓
	Boat Beacon v 2.53	×	✓
	AF track v 12.7	×	✓
	OpenCPN v 1.0.5	×	✓

intensity of the pixels of an area on the map. The authors set a threshold and compared the display pixel intensity; if it exceeded the threshold, it triggered the alerting mechanism. The authors reported a 93% success rate.

Goudossis and Katsikas [35] proposed identity-based public cryptography and symmetric cryptography to enhance the security of AIS. For asymmetric cryptography, they proposed that IMO generate a private key for the corresponding public keys of the National Maritime Authorities (NMA). In contrast, NMA could generate private keys for the corresponding keys at the national level. They also proposed symmetric cryptography for insecure sea areas such as the coast of Somalia. In this case, the presence of at least one trusted third party (e.g., a military patrol boat or a micro-satellite) is necessary to create and distribute the keys.

Bothur *et al.* [36] analyzed the security vulnerabilities and countermeasures in a smart ship system. According to them, most of the electronics systems in a ship, such as information technology networks, control systems, electronic chart display information system, very small aperture terminals, and AIS, are vulnerable to cyberattacks. They listed the possible weakness of all the subsystems. They mentioned that proper policy, and ensuring the security of data, application, host, network, etc., could help counter the attacks.

Su *et al.* [37] proposed a digital certificate-based identity authentication scheme to ensure the authenticity of the AIS data. According to them, a ship should generate its public and private keys. The public key should be distributed by an trusted official institution. They further proposed a mixed-zone and blind-signature-based trajectory privacy protection scheme to protect the vessel identity and the trajectory privacy. They suggested using pseudonyms instead of the actual MMSI to safeguard the true identity of a ship under the supervision of a trusted party called a ''certification authority'' (CA). It knows the relationship of every pseudonym to the real identity. If ships want to bar the CA from knowing the identity of the ship, the selection of pseudonyms must be executed by the vessel. In this case, the vessel sends many digital signatures to the CA. After the CA signs and returns the signatures, the ship chooses one randomly.

Sciancalepore *et al.* [38] proposed a secure, flexible, standard-compliant, and backward-compatible authentication framework to secure AIS broadcast messages. They contextualized a broadcast authentication protocol called Timed Efficient Stream Loss-tolerant Authentication (TESLA) and a space- and time-efficient probabilistic data structure called ''Bloom filter data structure'' in their authenticated AIS called ''Auth-AIS.'' Their proposed system required transmission of cryptography-related data via a type 8 message, which increases the AIS overhead. The difference with other cryptography solutions and Auth-AIS is the start time of the AIS transmission for a vessel, which is hidden but shared with other vessels by a trusted third party. An attacker is unlikely to know the precise AIS transmission timing of a vessel; thus, the fake transmission would not be authenticated by other ships.

VII. CONCLUSION

In this paper, we practically demonstrated and evaluated the impact of multiple attacks on AIS (both novel and known ones), primarily achievable via an RF link and with effects on the various network, processing, and display subsystems used within the AIS ecosystem. We developed a heterogeneous testbed that consisted of a commercial transponder, dual-channel AIS receiver, several SDRs, and software from different platforms (e.g., Android, Windows, and embedded OS), which resulted in a total of 19 tested configurations. Overall, within a controlled environment we performed 11 different tests/attacks, most of which represented either novel attack concepts or novel implementation of existing ideas in the AIS context. We demonstrated that the navigation security of multi-million dollar ships can be affected by a low-cost setup. Almost all the test configurations were impacted by some sort of attack. A coordinated attack, flooding, visual navigation disruption, and others indicate that attackers can be potentially harmful due to their ingenious attacks and state-of-the-art equipment. Software crashes due to DoS or an overwhelming number of alerts showed the most worrying scenario. Mobile applications are mostly unreliable and more vulnerable to attacks than desktop solutions, very likely due to memory, display, and computational constraints. During the experiments, we identified an AIS preamble-related implementation flaw. When the preamble started with 0, it worked fine across all the devices; but when it started with 1, it had the potential to affect the interoperability of some AIS devices. We urge the relevant stakeholders to pay attention to this issue to avoid further mishaps. Test results throughout this study reveal that, apart from the preamble-related issue, the other attack consequences belong to the software limitation. The identified issues are in the process of being reported to respective vendors according to responsible disclosure policy. The consistency of our results for a comprehensive range of hardware-software configurations indicates the reliability of our approach and test results. We hope researchers and industry can positively use our approach and outcomes to improve the cybersecurity of today's ever-growing AIS deployments. In the future, we plan to design, implement, and test different defense strategies against attacks on AIS. We also leave the jamming of specific messages for any targeted receiver as our future work.

APPENDIX A
AIS PREAMBLE AND NRZI

AIS uses NRZI encoding to encode the data. NRZI has many variants, and the one where the waveform changed due to the occurrence of 0 is called Non-Return-to-Zero Space (NRZS). More precisely, AIS uses NRZS variant of NRZI. Therefore, we also investigated the resulting differences between the scenarios where the preamble starts with 0, and 1 respectively, when using the NRZI conversion. Before the preamble starts according to the AIS message structure in Figure 2 there should be an 8-bit ramp up (*000000000*). Since NRZI depends on the previous level, we draw the signal with

ramp-up bits. The Figure 14 shows that after NRZI encoding, the two signals are not exactly the same. We further checked the NRZI conversion of AIS data with the Matlab code in [39]. That also resulted that preamble start with 1 and preamble start with 0 do not result exactly the same after NRZI conversion.

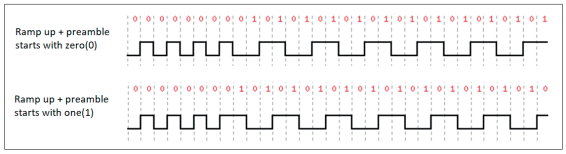


FIGURE 14. Comparison of NRZI conversion of ramp up and preamble bits – scenarios when starting with 0, and 1, respectively.

ACKNOWLEDGMENT

Some graphic materials in Figures 1 and 3 are under the fair-use courtesy of: GNU Radio, Nuand/Antratek, and HackRF/GreatScottGadgets; and royalty-free icons from www.flaticon.com (icons by: Freepik, Mattbadal, and Xnimrodx).

REFERENCES

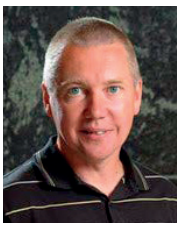
[1] IMO. *AIS Transponders*. Accessed: Dec. 10, 2021. [Online]. Available: https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx
[2] *Vessel Finder*. Accessed: Dec. 11, 2021. [Online]. Available: https://www.vesselfinder.com/vessels
[3] T. Bateman. *HMS Defender: AIS Spoofing is Opening up a New Front in the War on Reality*. Accessed: Nov. 16, 2021. [Online]. Available: https://www.euronews.com/next/2021/06/28/hms-defender-ais-spoofing-is-o%pening-up-a-new-front-in-the-war-on-reality
[4] C. Zwirko. *North Korean Vessels Exploiting Tracking System Flaws to Evade Sanctions: Report*. Accessed: Nov. 15, 2021. [Online]. Available: https://www.nknews.org/2019/06/north-korean-vessels-exploiting-tracking%-system-flaws-to-evade-sanctions-report
[5] S. Chambers. *U.S. Warns Merchant Shipping of Iranian GPS Spoofing Threat*. Accessed: Nov. 15, 2021. [Online]. Available: https://splash247.com/us-warns-merchant-shipping-of-iranian-gps-spoofin%g-threat
[6] M. Balduzzi, A. Pasta, and K. Wilhoit, "A security evaluation of AIS automated identification system," in *Proc. 30th Annu. Comput. Secur. Appl. Conf.*, Dec. 2014, pp. 436–445.
[7] A. Costin and A. Francillon, "Ghost in the air (traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices," *Black Hat USA*, vol. 1, pp. 1–12, Jul. 2012.
[8] S. Khandker, H. Turtiainen, A. Costin, and T. Hamalainen, "Cybersecurity attacks on software logic and error handling within ADS-B implementations: Systematic testing of resilience and countermeasures," *IEEE Trans. Aerosp. Electron. Syst.*, early access, Dec. 31, 2021, doi: 10.1109/TAES.2021.3139559.
[9] *Technical Characteristics for an Automatic Identification System Using Time-Division Multiple Access in the VHF Maritime Mobile Band*, International Telecommunication Union, Geneva, Switzerland, 2014. [Online]. Available: https://www.itu.int/rec/R-REC-M.1371-5-201402-I
[10] United States Coast Guard. *AIS Messages*. Accessed: Dec. 19, 2021. [Online]. Available: https://www.navcen.uscg.gov/?pageName=AISMessages
[11] K. F. Mathapo, "A software-defined radio implementation of maritime AIS," M.S. thesis, Dept. Elect. Electron. Eng., Univ. Stellenbosch, Stellenbosch, South Africa, 2007.
[12] J. A. Larsen, H. P. Mortensen, and J. D. Nielsen, "An SDR based AIS receiver for satellites," in *Proc. 5th Int. Conf. Recent Adv. Space Technol. (RAST)*, Jun. 2011, pp. 526–531.
[13] M. Balduzzi. *AISTX*. Accessed: Aug. 25, 2021. [Online]. Available: https://github.com/trendmicro/ais
[14] M. M. Marques, D. Teles, V. Lobo, and G. Capela, "Low-cost AIS transponder using an SDR device," in *Proc. OCEANS MTS/IEEE SEATTLE*, Oct. 2019, pp. 1–4.
[15] E. S. Raymond. *AIVDM/AIVDO Protocol Decoding*. Accessed: Dec. 28, 2021. [Online]. Available: https://gpsd.gitlab.io/gpsd/AIVDM.html
[16] F. R. G. Cruz, R. C. M. Gania, B. W. C. Garcia, and J. C. R. Nob, "Implementing automatic identification system transmitter on software defined radio," in *Proc. IEEE 10th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM)*, Nov. 2018, pp. 1–4.
[17] N. Foster. *Automatic Information System Decoder for Shipborne Position Reporting for the Gnuradio Project*. Accessed: Dec. 13, 2021. [Online]. Available: https://github.com/bistromath/gr-ais
[18] H. Hannikainen. *Open Source Software for Demodulating and Decoding AIS Messages*. Accessed: Dec. 19, 2021. [Online]. Available: https://github.com/rubund/gnuais
[19] A. D. Cruz. *Automatic Identification System (AIS) using GNU Radio*. Accessed: Dec. 30, 2021. [Online]. Available: https://www.funwithelectronics.com/?id=9
[20] B. Brewin. (2013). *University of Texas Team Hijacks $80 Million Yacht With Cheap GPS Spoofing Gear*. Accessed: Oct. 10, 2021. [Online]. Available: https://www.nextgov.com/cio-briefing/2013/07/university-texas-team-hija%cks-80-million-yacht-cheap-gps-spoofing-gear/67625
[21] M. L. Psiaki, T. E. Humphreys, and B. Stauffer, "Attackers can spoof navigation signals without our knowledge. Here's how to fight back GPS lies," *IEEE Spectr.*, vol. 53, no. 8, pp. 26–53, Aug. 2016.
[22] A. Androjna, M. Perkovič, I. Pavic, and J. Mišković, "AIS data vulnerability indicated by a spoofing case-study," *Appl. Sci.*, vol. 11, no. 11, p. 5015, May 2021.
[23] COAA. *ShipPlotter*. Accessed: Aug. 27, 2021. [Online]. Available: http://www.coaa.co.U.K./shipplotter.htm
[24] OpenCPN. *Chart Plotter and Navigational Software*. Accessed: Aug. 20, 2021. [Online]. Available: https://opencpn.org
[25] MarineTraffic. *AISMon*. Accessed: Nov. 17, 2021. [Online]. Available: https://help.marinetraffic.com/hc/en-us/articles/205339707-AISMon
[26] ebcTech. *AIS Share*. Accessed: Aug. 20, 2021. [Online]. Available: https://play.google.com/store/apps/details?id=eu.ebctech.ais_share&hl=e%n&gl=U.S
[27] W. U. Hassan, S. Guo, D. Li, Z. Chen, K. Jee, Z. Li, and A. Bates, "NoDoze: Combatting threat alert fatigue with automated provenance triage," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019, pp. 1–40.
[28] M. Smith, M. Strohmeier, J. Harman, V. Lenders, and I. Martinovic, "A view from the cockpit: Exploring pilot reactions to attacks on avionic systems," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2020, pp. 1–18.
[29] *Maritime Navigation and Radio Communication Equipment and Systems Class B. Shipborne Equipment of the Automatic Identification System (AIS)—Part 1: Carrier-Sense Time Division Multiple Access (CSTDMA) Techniques*, International Electrotechnical Commission, Geneva, Switzerland, 2017.
[30] *Maritime Navigation and Radiocommunication Equipment and Systems-Class B. Shipborne Equipment of the Automatic Identification System (AIS)—Part 2: Self-Organising Time Division Multiple Access (SOTDMA) Techniques*, International Electrotechnical Commission, Geneva, Switzerland, 2017.
[31] *Maritime Navigation and Radiocommunication Equipment and Systems Automatic Identification Systems (AIS). Part 2: Class a Shipborne Equipment of the Automatic Identification System (AIS) Operational and Performance Requirements, Methods of Test and Required Test Results*, International Electrotechnical Commission, Geneva, Switzerland, 2012.
[32] D. B. Parker, *Toward a New Framework for Information Security*. Hoboken, NJ, USA: Wiley, 2012.
[33] C. Ray, C. Iphar, and A. Napoli, "Methodology for real-time detection of AIS falsification," in *Proc. Maritime Knowl. Discovery Anomaly Detection Workshop*, 2016, pp. 74–77.
[34] W. C. Leite Junior, C. C. de Moraes, C. E. P. de Albuquerque, R. C. S. Machado, and A. O. de Sá, "A triggering mechanism for cyber-attacks in naval sensors and systems," *Sensors*, vol. 21, no. 9, p. 3195, May 2021.
[35] A. Goudossis and S. K. Katsikas, "Towards a secure automatic identification system (AIS)," *J. Mar. Sci. Technol.*, vol. 24, no. 2, pp. 410–423, Jun. 2019.

[36] D. Bothur, G. Zheng, and C. Valli, "A critical analysis of security vulnerabilities and countermeasures in a smart ship system," in *The Proc. 15th Austral. Inf. Secur. Manage. Conf.*, 2017, pp. 81–87.
[37] P. Su, N. Sun, L. Zhu, Y. Li, R. Bi, M. Li, and Z. Zhang, "A privacy-preserving and vessel authentication scheme using automatic identification system," in *Proc. 5th ACM Int. Workshop Secur. Cloud Comput.*, Apr. 2017, pp. 83–90.
[38] S. Sciancalepore, P. Tedeschi, A. Aziz, and R. Di Pietro, "Auth-AIS: Secure, flexible, and backward-compatible authentication of vessels AIS broadcasts," *IEEE Trans. Depend. Sec. Comput.*, early access, Mar. 30, 2021, doi: 10.1109/TDSC.2021.3069428.
[39] A. Dembovskis, "AIS message extraction from overlapped AIS signals for SAT-AIS applications," Ph.D. dissertation, Dept. Math. Comput. Sci., Univ. Bremen, Bremen, Germany, 2015.

SYED KHANDKER received the M.Sc. degree in web intelligence and service engineering from the University of Jyväskylä, Finland, in 2016, where he is currently pursuing the doctoral degree with the Faculty of Information Technology. Since his childhood, he has been a radio enthusiast and holds an amateur radio operator license. He has authored five journal articles. His research interests include in the field of RF fingerprint positioning, automatic dependent surveillance-broadcast, automatic identification systems, wireless communications, and artificial intelligence.

HANNU TURTIAINEN received the B.Sc. degree in electronics engineering from the University of Applied Sciences, Jyväskylä, Finland, and the M.Sc. degree in cybersecurity from the University of Jyväskylä, Finland, in 2020, where he is currently pursuing the Ph.D. degree in software and communication technology. He is also working in the IoT field as a Cybersecurity and Software Engineer at Binare.io, a deep-tech cybersecurity spin-off from the University of Jyväskylä. His research interests include machine learning and artificial intelligence in the cybersecurity and digital privacy field.

ANDREI COSTIN received the Ph.D. degree from EURECOM/Télécom ParisTech, in 2015, under co-supervision of Prof. Francilon and Prof. Balzarotti. He is currently a/an Senior Lecturer/Assistant Professor in cybersecurity with the University of Jyväskylä, Central Finland, with a particular focus on the IoT/firmware cybersecurity and digital privacy. He has been publishing and presenting at more than 45 top international cybersecurity venues, both academic (Usenix Security and ACM ASIACCS) and industrial (BalckHat, CCC, and HackInTheBox). He is the author of the first practical ADS-B attacks (BlackHat 2012) and has literally established the large-scale automated firmware analysis research areas (Usenix Security 2014)—these two works are considered seminal in their respective areas—being also most cited at the same time. He is also the CEO/Co-Founder of Binare.io, a deep-tech cybersecurity spin-off from the University of Jyväskylä, focused on innovation and tech-transfer related to the IoT cybersecurity.



TIMO HÄMÄLÄINEN has over 25 years of research and teaching experience related to computer networks. He has lead tens of external funded networks management related projects. He has launched and leads master's programs in the University of Jyväskylä (currently SW and Comm. Eng.) and teaches networks management related courses. He has more than 200 internationally peer-reviewed publications, and he has supervised 36 Ph.D. theses. His current research interests include wireless/wired networks resource management (the IoT, SDN, and NFV) and networks security.