POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Securing the ADS-B Protocol: A Comprehensive Framework Combining Bandwidth-Efficient Broadcast Authentication and Deep Learning-Based Intrusion Detection

MIKAËLA STÉPHANIE NGAMBOÉ MVOGO
Département de génie informatique et génie logiciel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie informatique

Décembre 2025

© Mikaëla Stéphanie Ngamboé Mvogo, 2025.

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Securing the ADS-B Protocol: A Comprehensive Framework Combining Bandwidth-Efficient Broadcast Authentication and Deep Learning-Based Intrusion Detection

présentée par **Mikaëla Stéphanie NGAMBOÉ MVOGO**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Martine BELLAÏCHE , présidente
Gabriela NICOLESCU, membre et directrice de recherche
Samuel PIERRE, membre
Kamel ADI, membre externe

DEDICATION

To the glory of the Most Holy Trinity,
in gratitude for all the graces bestowed throughout this journey.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank You, God, for the love, support, and comfort You have given me throughout this journey. You know all things, and You know how deeply I love You.

I also wish to thank the Blessed Virgin Mary (Seat of Wisdom) and Saint Joseph (Model of Workers), under whose protection I have placed this PhD and my professional career.

On a more academic note, I am grateful to the members of my jury, Professors Martine Bellaïche, Samuel Pierre, and Kamel Adi, for accepting to evaluate this work and for being part of my committee.

I am particularly grateful to my thesis advisor, Professor Gabriela Nicolescu, for her invaluable guidance and the trust she placed in me. Her encouragement to propose ideas and take the initiative has been an essential lesson in independence and professional development.

I am equally thankful to my mentor, Professor José Manuel Fernandez, for instilling in me a sense of scientific rigor—though perhaps more in spirit than in practice. His unwavering support throughout this process, whether answering my questions or reviewing my texts even after his retirement, has been invaluable.

I thank the industrial partners of the CyberSA project, particularly those I had the privilege to collaborate more directly with: Rémi Benito, Greg Rice, Benoit Joly, Steven Biegler, and Paul Berthier. Thank you for your time and for sharing your expertise with me so generously.

My deepest thanks go to my family, my parents, and three brothers. Words cannot fully express my love and gratitude. Your unconditional support and affection, alongside God, have been the driving forces behind my perseverance and determination to give my best every day. My family is my rock.

I would also like to acknowledge the members of my laboratory. In particular, I am grateful to Professor Felipe and to Jean-Yves for their kindness, advice, and guidance during these years. I also thank all my lab colleagues (there are many of you!), especially Jean-Simon, Nuno and Jimmy. I truly miss the work sessions at the Caravan Café.

To my friends, thank you for your patience and understanding. Even when I was less available, you never left me in the lurch. On the contrary, you continued to encourage me and remind me that life extends beyond Polytechnique and the PhD. I would like to thank Ana, Antoine L., Archibaldo, Chantal, Danilo, Désirée, Fritz, Hoa, Kim, Louise, Marielba, Narcisse, Popi, Vicky and Yoanna (listed alphabetically).

A few months ago, you told me, "*battles are won by tired soldiers*," to encourage me to give my all until the end, just like Him. These words have since become my *mantra*. Thank you, Don Salva, for your unwavering support and encouragement.

For the same reason, I also wish to thank Brother René, Sor Natalia and the Fathers of the Angus Residence: Abbé Desorcy (RIP), Magnan, Lachapelle, Quintal, and Labossière.

Last but not least, a heartfelt thank you to Toni Z.

RÉSUMÉ

L'ADS-B (*Automatic Dependent Surveillance–Broadcast*) est un protocole de communication aéronautique, qui permet aux aéronefs de diffuser en continu leur position ainsi que d'autres données de navigation essentielles. Aujourd'hui largement déployé par les prestataires de services de navigation aérienne, tels que NAV CANADA, son utilisation est obligatoire dans plusieurs espaces aériens.

Cependant, ce protocole présente une vulnérabilité majeure : l'absence de mécanismes d'authentification et de vérification de l'intégrité des messages transmis. Cette faille expose l'ADS-B à des cyberattaques, notamment via des radios logicielles, permettant à des acteurs malveillants d'émettre de faux messages, de rejouer des messages authentiques ou de brouiller les communications.

Les conséquences potentielles d'une attaque sont graves : augmentation des risques pour la sûreté aérienne, perturbations opérationnelles et financières pour les compagnies aériennes (retards, annulations de vols, perte de confiance des passagers), ainsi que des enjeux géopolitiques liés à la sécurité et à la souveraineté des espaces aériens.

Les travaux visant à sécuriser l'ADS-B ont révélé deux limites principales. D'une part, les schémas d'authentification proposés peinent à concilier les exigences de cybersécurité avec les contraintes opérationnelles imposées par la norme. En effet, la compatibilité descendante — indispensable pour que tous les récepteurs puissent décoder les messages — et les limites strictes de bande passante, qui imposent d'éviter toute augmentation du trafic sur un canal déjà saturé, rendent cette conciliation particulièrement difficile.

D'autre part, bien que prometteurs, les systèmes de détection d'intrusion (SDI) fondés sur l'apprentissage profond montrent des limites face à des attaques sophistiquées ou inédites. Leur latence, inadaptée aux contraintes de temps réel de la gestion du trafic aérien, pose un défi supplémentaire.

Pour répondre à cette problématique, cette thèse propose une approche intégrée combinant le protocole CABBA (*Compatible Authenticated Bandwidth-efficient Broadcast for ADS-B*) et un système de détection d'intrusions sophistiquées basé sur l'architecture xLSTM (*extended long-short term memory*).

La solution CABBA repose sur le schéma d'authentification TESLA (*Timed Efficient Stream Loss-tolerant Authentication*), une modulation par superposition de phase et une infrastructure à clé publique basée sur des certificats. Elle garantit l'authentification des émetteurs,

l'intégrité des messages, la compatibilité avec les récepteurs existants et une efficacité spectrale optimale.

Afin de valider CABBA, nous l'avons implémenté et testé avec des radios logicielles. Nous avons mené des tests de compatibilité descendante avec des récepteurs ADS-B commerciaux et des équipements d'aviation générale. Nous avons également évalué le taux d'occupation du canal 1 090 MHz (pour les messages CABBA) et analysé le taux d'erreur binaire (BER, *bit-error rate*). Les résultats montrent que CABBA assure la compatibilité descendante, n'engendre qu'une surcharge minime du canal de communication et atteint un BER acceptable pour des valeurs de E_b/N_0 supérieures à 14 dB.

En complément, nous avons développé le premier système de détection d'intrusions pour l'ADS-B fondé sur une architecture xLSTM, optimisé par apprentissage par transfert. Par rapport à un SDI basé sur un transformeur, ce modèle atteint un score F1 de 98,9 % (contre 94,3 %) et s'avère efficace pour détecter les attaques graduelles, qui altèrent progressivement la conscience de la situation des pilotes et des contrôleurs aériens. Bien que sa latence d'inférence (7,26 s) soit supérieure à celle du transformeur (2,1 s), elle reste adaptée aux intervalles de rafraîchissement des radars secondaires (5 à 12 s).

Ainsi, la solution CABBA et le système de détection d'intrusion basé sur l'architecture xLSTM constituent un cadre de sécurité complémentaire : CABBA assure la prévention grâce à l'authentification des diffusions, tandis que le système de détection pallie les risques résiduels liés aux limites de CABBA.

ABSTRACT

Automatic Dependent Surveillance–Broadcast (ADS-B) is an aeronautical communication protocol that enables aircraft to periodically broadcast their position and other navigation data. By providing continuous and precise information about aircraft location, ADS-B functions as a surveillance technology employed by Air Navigation Service Providers (ANSP), such as NAV CANADA, to support Air Traffic Control (ATC) services. The use of ADS-B is mandated in many countries; however, it remains vulnerable to cyberattacks. The system lacks built-in mechanisms for authenticating messages or ensuring their integrity. Using readily available and inexpensive equipment, such as software-defined radios (SDR), malicious actors can inject false messages into the ADS-B system, replay legitimate messages, or jam communications. If left unaddressed, the exploitation of this vulnerability could compromise the effectiveness of ADS-B and lead to wide-ranging consequences. These include heightened risks to aviation safety, operational and financial disruptions for airlines, such as delays, flight cancellations, and loss of passenger confidence, as well as broader geopolitical implications related to the security and sovereignty of national airspaces.

Previous efforts to secure the ADS-B protocol have revealed significant limitations. Broadcast authentication schemes have struggled to balance security with operational requirements, such as backward compatibility and bandwidth efficiency. For data integrity verification, deep learning-based intrusion detection systems (IDS) have shown limited effectiveness against sophisticated or novel attacks, raising safety concerns. These concerns are often compounded by inference delays resulting from the complexity of modern deep-learning (DL) architectures. Given these challenges, the question of how to secure ADS-B while satisfying the technical and operational requirements that underpin its safety remains unresolved.

To address this gap, we propose the Compatible Authenticated Bandwidth-efficient Broadcast protocol for ADS-B (CABBA), an enhanced broadcast authentication scheme that integrates Timed Efficient Stream Loss-tolerant Authentication (TESLA), phase-overlay modulation techniques, and certificate-based public key infrastructure (PKI). CABBA provides entity authentication, data origin authentication, and data integrity. To validate compliance with existing standards, we developed an SDR-based implementation and conducted backward compatibility tests with both commercial and general aviation ADS-B receivers. We also measured the 1 090 MHz channel occupancy rate following the ITU-R SM.2256-1 recommendation and performed a bit-error rate analysis. The results indicate that CABBA preserves backward compatibility, introduces negligible communication overhead, and achieves accept-

able error rates for E_b/N_0 values above 14 dB.

Complementing this preventive solution, we designed the first extended long short-term memory (xLSTM)-based intrusion detection system for ADS-B. Using a transfer learning strategy — pre-training on benign traffic and fine-tuning with tampered messages — the system was benchmarked against a transformer-based IDS. The xLSTM model demonstrated superior performance, achieving an F1-score of 98.9 % compared to 94.3 % for the transformer. It also proved effective in detecting gradual attacks that progressively undermine situational awareness. Latency analysis showed that the 7.26-second inference time of the xLSTM model remains within the Secondary Surveillance Radar refresh interval (5–12 s), although it may pose constraints in time-critical scenarios, such as those in aircrew situational awareness and collision avoidance. The transformer-based IDS achieved a lower latency (2.1 s) but at the expense of reduced detection accuracy.

Together, CABBA and the xLSTM-based IDS provide a complementary security framework: CABBA provides prevention through broadcast authentication, whereas the IDS mitigates the residual risks associated with the limitations of CABBA.

TABLE OF CONTENTS

DEDICATION iii

ACKNOWLEDGEMENTS iv

RÉSUMÉ vi

ABSTRACT viii

LIST OF TABLES xiv

LIST OF FIGURES xv

LIST OF SYMBOLS AND ACRONYMS xvi

LIST OF APPENDICES xix

CHAPTER 1 INTRODUCTION 1
- 1.1 ADS-B: A paradigm shift in air traffic control 1
- 1.2 How can the security of ADS-B be strengthened? 3
- 1.3 Research objective and associated research questions 7
- 1.4 Thesis outline 9

CHAPTER 2 FROM VISUAL SIGNALS TO ADS-B: EVOLUTION OF ATC SURVEILLANCE SYSTEMS 10
- 2.1 Historical background and early techniques 10
- 2.2 Radar-based surveillance systems 11
  - 2.2.1 Primary Surveillance Radar (PSR) 12
  - 2.2.2 Secondary Surveillance Radar (SSR) 13
- 2.3 Mode S 14
- 2.4 ADS-B: A satellite-enabled surveillance system 14
  - 2.4.1 Operational uses of ADS-B 15
  - 2.4.2 Components of the ADS-B system 15
  - 2.4.3 ADS-B Messages: Packet Structure, Types and Transmission Rates . 16
  - 2.4.4 Message transmission process 18
  - 2.4.5 Message reception and integration in ATC system 18
- 2.5 Summary of the chapter 19

CHAPTER 3 ADS-B SECURITY ANALYSIS AND PERFORMANCE REQUIREMENTS . . . 20
3.1 Security Properties . . . 20
3.2 ADS-B security analysis . . . 21
3.2.1 Attack taxonomy . . . 21
3.2.2 Threat Model . . . 22
3.2.3 Risk to Surveillance . . . 22
3.2.4 Risk to Traffic Awareness . . . 24
3.3 Technical and operational criteria for secure ADS-B solutions . . . 25
3.3.1 Safety Criteria . . . 25
3.3.2 Security Criteria . . . 27
3.3.3 Rapid Deployment Principles . . . 28
3.4 Summary of the chapter . . . 29

CHAPTER 4 CRYPTOGRAPHIC APPROACHES FOR ADS-B BROADCAST AUTHENTICATION . . . 30
4.1 Symmetric cryptography-based techniques . . . 30
4.1.1 Format-preserving encryption . . . 30
4.1.2 Message authentication code . . . 31
4.2 Asymmetric Cryptography-Based Techniques . . . 32
4.2.1 Certificate-based signature . . . 32
4.2.2 Identity-based signature . . . 35
4.2.3 Certificateless signature . . . 38
4.3 Hybrid cryptography-based techniques . . . 40
4.3.1 TESLA . . . 40
4.4 Positioning of our approach . . . 42
4.5 Summary of the chapter . . . 44

CHAPTER 5 MACHINE LEARNING TECHNIQUES FOR ADS-B DATA INTEGRITY VERIFICATION . . . 45
5.1 Deep Learning Architectures for Intrusion Detection . . . 45
5.1.1 Autoencoders . . . 45
5.1.2 Variational autoencoders . . . 46
5.1.3 Recurrent Neural Networks . . . 46
5.1.4 Long Short-Term Memory . . . 46
5.1.5 Convolutional LSTM . . . 47
5.1.6 Transformer . . . 48

5.2 Deep Learning Applications in ADS-B Intrusion Detection 49
5.3 Emerging Approaches in Deep Learning-Based Intrusion Detection 53
5.3.1 Extended LSTM 53
5.3.2 Transfer learning 55
5.4 Positioning of our approach 56
5.5 Summary of the chapter 56

CHAPTER 6 ARTICLE 1 - CABBA: COMPATIBLE AUTHENTICATED BANDWIDTH-EFFICIENT BROADCAST PROTOCOL FOR ADS-B 57
6.1 Introduction 58
6.2 Overview of cryptographic solutions for ADS-B 61
6.2.1 Symmetric cryptography-based protocols 62
6.2.2 Asymmetric cryptography-based protocols 62
6.2.3 Hybrid cryptography-based protocols 64
6.3 Background 66
6.3.1 Timed Efficient Stream Loss-tolerant Authentication (TESLA) . . . 66
6.4 Phase overlaid modulation techniques 67
6.5 CABBA : Compatible Authenticated Bandwidth-efficient Broadcast protocol for ADS-B 69
6.5.1 CABBA on the sender side 71
6.5.2 CABBA on the receiver side 74
6.6 Backward compatibility experiments 79
6.7 Operational Viability of CABBA 80
6.7.1 Comparative BER analysis of CABBA with D8PSK vs. D16PSK . . 81
6.7.2 Channel occupancy rate (COR) analysis 83
6.7.3 Safety Impact of Unauthenticated Messages 87
6.8 Conclusion 91

CHAPTER 7 ARTICLE 2 - NEW MACHINE LEARNING APPROACHES FOR INTRUSION DETECTION IN ADS-B 93
7.1 Introduction 94
7.2 ADS-B Threat Model 97
7.3 Previous works 98
7.4 Background on xLSTM 100
7.5 Methodology 100
7.5.1 IDS Implementation 100
7.5.2 Data Acquisition and Dataset Implementation 102

7.6 Experiments 103
7.6.1 Experimental Setup 104
7.6.2 Performance Evaluation Metrics 104
7.6.3 Hyperparameter Optimization 106
7.7 Results 107
7.7.1 Binary Classification: Classical vs Deep Learning Models 107
7.7.2 Fine-Tuning and Multiclass Classification 108
7.7.3 Generalization to a Novel Attack 110
7.7.4 Inference Time Analysis and System Performance 110
7.8 Conclusion 111

CHAPTER 8 CONCLUSION 113
8.1 Summary of Works 113
8.1.1 CABBA 113
8.1.2 xLSTM-based IDS 115
8.2 Limitations 116
8.2.1 CABBA 116
8.2.2 xLSTM-based IDS 117
8.3 Future Research 118

REFERENCES 120

APPENDICES 136

LIST OF TABLES

Table 1.1 Overview of Thesis Chapters and Contributions 9
Table 2.1 Transmission rates of ADS-B messages depending on aircraft state. . 17
Table 4.1 Domain Parameters of ECDSA 33
Table 4.2 Parameter Set of IBS-1 Scheme from ISO/IEC 14888-3 35
Table 4.3 Parameter set of CL-PKS Scheme from the original paper [1] 37
Table 4.4 Positioning our ADS-B broadcast authentication methodology with respect to existing literature 43
Table 5.1 Works on deep learning-based ADS-B intrusion detection 50
Table 6.1 Cryptographic techniques for enhancing ADS-B security 61
Table 6.2 Comparison of CABBA packet structure with earlier TESLA-based solutions. 70
Table 6.3 Transmission period parameters for each of the four scenarios for which we computed the COR values. 85
Table 6.4 Packet loss probability and uncertainty times for TCAS. 89
Table 6.5 Packet loss probability and uncertainty times for ATC. 91
Table 7.1 Best hyperparameter configurations for pre-trained models 105
Table 7.2 Best hyperparameter configurations for fine-tuned models 105
Table 7.3 Performance results for the binary classification task consisting of distinguishing between genuine and tampered messages. 107
Table 7.4 Performance results of the four fine-tuned binary classifiers implemented. 109
Table 7.5 Performance results of the multiclass classifier when evaluated on unseen data containing known attacks. 109
Table 7.6 Performance results of the multiclass classifier when evaluated on unseen data containing unknown attacks. 110
Table A.1 Approaches for ADS-B data integrity verification 137

LIST OF FIGURES

Figure 1.1 Operational overview of ADS-B. Adapted from [2]. 2
Figure 2.1 Early air traffic control operations 11
Figure 2.2 Collocated PSR and SSR antennas 12
Figure 2.3 Structure of a Mode S downlink message 13
Figure 2.4 1 090 MHz ADS-B System 16
Figure 2.5 ADS-B packet structure 16
Figure 4.1 ECDSA scheme adapted from FIPS 186-5 [3] 34
Figure 4.2 ISO/IEC 14888-3 IBS-1 Scheme as interpreted in [4] 36
Figure 4.3 CL-PKC Signature Scheme as proposed in [1] 38
Figure 4.4 One-way keychain and authenticathion keys generation in TESLA . . 40
Figure 5.1 Comparison between LSTM and ConvLSTM cell structures. 47
Figure 5.2 Transformer architecture, reproduced from [5] 49
Figure 5.3 Comparison between LSTM and xLSTM memory cells, reproduced from [6]. 54
Figure 5.4 Comparison of update equations in scalar LSTM (left) and matrix LSTM (right) formulations within xLSTM. 54
Figure 6.1 Phase overlay method for embedding additional data in 1090ES messages. 68
Figure 6.2 Structure of the Type A packets in CABBA. 74
Figure 6.3 Structure of Type B1 and B2 packets 75
Figure 6.4 Structure of Type C packets. 76
Figure 6.5 Authentication process for ADS-B messages 77
Figure 6.6 Backward compatibility test with the Stratus II receiver 81
Figure 6.7 Backward compatibility test with the TSS-4100 transponder 82
Figure 6.8 BER analysis of CABBA 83
Figure 6.9 Mean COR for ADS-B transmissions 85
Figure 6.10 Mean cor for ADS-B and CABBA transmissions 86
Figure 7.1 Architecture of the original LSTM memory cells and the new xLSTM variants (sLSTM and mLSTM), based on the illustration in paper [6] 96
Figure 7.2 Methodology for pre-training and fine-tuning. 101
Figure 7.3 Overview of the experimental methodology. 103
Figure 7.4 Comparison of performance metrics across six classifiers. 108

LIST OF SYMBOLS AND ACRONYMS

ACC	Area Control Centers
ACAS	Airborne Collision Avoidance System
AD	Airworthiness Directives
AE	Autoencoder
AFD	Adaptive Flight Display
ADS-B	Automatic Dependent Surveillance–Broadcast
ANSP	Air Navigation Service Provider
ARTCC	Air Route Traffic Control Center
ASL	Above Sea Level
ATIS	Automatic Terminal Information Service
ATM	Air Traffic Management
AWGN	Additive White Gaussian Noise
BCE	Binary Cross-Entropy
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CA	Certification Authority
CABBA	Compatible Authenticated Bandwidth-efficient Broadcast protocol for ADS-B
CAE	Contextual Autoencoder
CL-PKC	Certificateless Public Key Cryptography
COST	Commercial Off-The-Shelf
ConvLSTM	Convolutional LSTM
D16PSK	Differential 16-Phase-Shift Keying
D8PSK	Differential 8-Phase-Shift Keying
DL	Deep Learning
DNN	Deep Neural Network
DT	Decision Tree
ECDSA	Elliptic Curve Digital Signature Algorithm
EdDSA	Edwards-Curve Digital Signature Algorithm
EFB	Electronic Flight Bag
eMRTD	Electronic Machine Readable Travel Documents
ES	Extended Squitter
ESI	Extra Security Information

FAA	Federal Aviation Administration
FAR	False Alarm Rate
FIS-B	Flight Information Service–Broadcast
FN	False Negative
FP	False Positive
FPE	Format-Preserving Encryption
FPR	False Positive Rate
GA	General Aviation
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HMAC	Hash-Based Message Authentication Code
IBS	Identity-Based Signature
ICAO	International Civil Aviation Organization
IDS	Intrusion Detection System
IF	Intermediate Frequency
KGC	Key Generation Center
LEO	Low Earth Orbit
LOS	Line of Sight
LSTM	Long Short-Term Memory
MAC	Message Authentication Code
ML	Machine Learning
MLAT	Multilateration
MLP	Multilayer Perceptron
mLSTM	Matrix LSTM
MFD	Multifunction Display
MOPS	Minimum Operational Performance Standard
MSE	Mean Squared Error
MTOW	Maximum Takeoff Weight
NextGen	Next Generation Air Transportation System
NAA	National Aviation Authority
NIST	National Institute of Standards and Technology
NN	Neural Networks
NSI	Non-Security Information
NOTAM	Notice to Air Missions
PKD	Public Key Directory
PKG	Private Key Generator

PKI	Public Key Infrastructure
PPM	Pulse Position Modulation
RAT	Remote Access Tools
RF	Radio Frequency
RF	Random Forest
RA	Resolution Advisory
RNN	Recurrent Neural Network
RPM	Revolutions Per Minute
RSA	Rivest–Shamir–Adleman
RTCA	Radio Technical Commission for Aeronautics
SB	Service Bulletins
sLSTM	Scalar LSTM
SDR	Software-Defined Radio
SESAR	Single European Sky ATM Research
SSR	Secondary Surveillance Radar
SVM	Support Vector Machine
SVDD	Support Vector Data Description
TA	Traffic Advisory
TCAS	Traffic Alert and Collision Avoidance System
TCN	Temporal Convolutional Network
TDOA	Time Difference of Arrival
TMA	Terminal Maneuvering Area
TN	True Negative
TP	True Positive
TESLA	Timed Efficient Stream Loss-tolerant Authentication
UAT	Universal Access Transceiver
VDL-4	VHF Data Link Mode 4
VAE	Variational Autoencoder
VFR	Visual Flight Rules
VHF	Very High Frequency
xLSTM	Extended LSTM

LIST OF APPENDICES

Appendix A Existing verification techniques and motivation for machine learning-based approaches 136

CHAPTER 1 INTRODUCTION

1.1 ADS-B: A paradigm shift in air traffic control

In the 1990's, severe congestion and costly delays, along with forecasts of increasing demand for aviation services, prompted a global push to modernize air traffic management (ATM) systems [7]. Consequently, modernization programs were initiated worldwide, including the Next Generation Air Transportation System (NextGen) in the United States, the Single European Sky ATM Research (SESAR) in Europe, and similar initiatives in other regions [8]. These efforts aim to enhance the safety, capacity, efficiency, and environmental sustainability of air traffic flow [9, 10]. Achieving these goals necessitated a major shift from ground-based radar surveillance to satellite-enabled systems. Automatic Dependent Surveillance–Broadcast (ADS-B) is the key technology driving this transition, replacing radar as the primary surveillance source in Canada, the European Union, the United States, and many other airspaces globally [11].

To fully grasp the impact of ADS-B on air traffic management, understanding its basic operation is crucial. As depicted in Figure 1.1, aircraft equipped with ADS-B Out capability consistently broadcast messages that contain their GPS-based location, identity, ground speed, operational status, and other data collected from onboard sensors [2, 12]. These messages are intercepted by ADS-B In receivers, which process and use the information primarily for two purposes: air traffic surveillance and aircrew situational awareness.

In the realm of surveillance, ground-based ADS-B antennas are the prevalent type of ADS-B In receivers. They intercept broadcast messages and transmit them to the air traffic control service through the designated Air Navigation Service Provider (ANSP). This facilitates precise real-time monitoring from the ground [13, 14]. In areas where such infrastructure is unfeasible, such as remote, polar, or oceanic regions, low Earth orbit (LEO) communication satellites equipped with ADS-B In receivers extend surveillance by intercepting broadcasts and relaying them to ground stations. This space-based ADS-B system extends coverage to airspaces that would otherwise remain unsupervised. For example, NAV CANADA was the first ANSP in the world to implement space-based ADS-B surveillance, launching the system over the North Atlantic airspace in 2019 [15].

Another application of ADS-B is its use to increase the situational awareness of aircrews to surrounding traffic, that is, as an alternate means of providing an Airborne Collision Avoidance System (ACAS). The ADS-B reports sent by other aircraft within range are received by

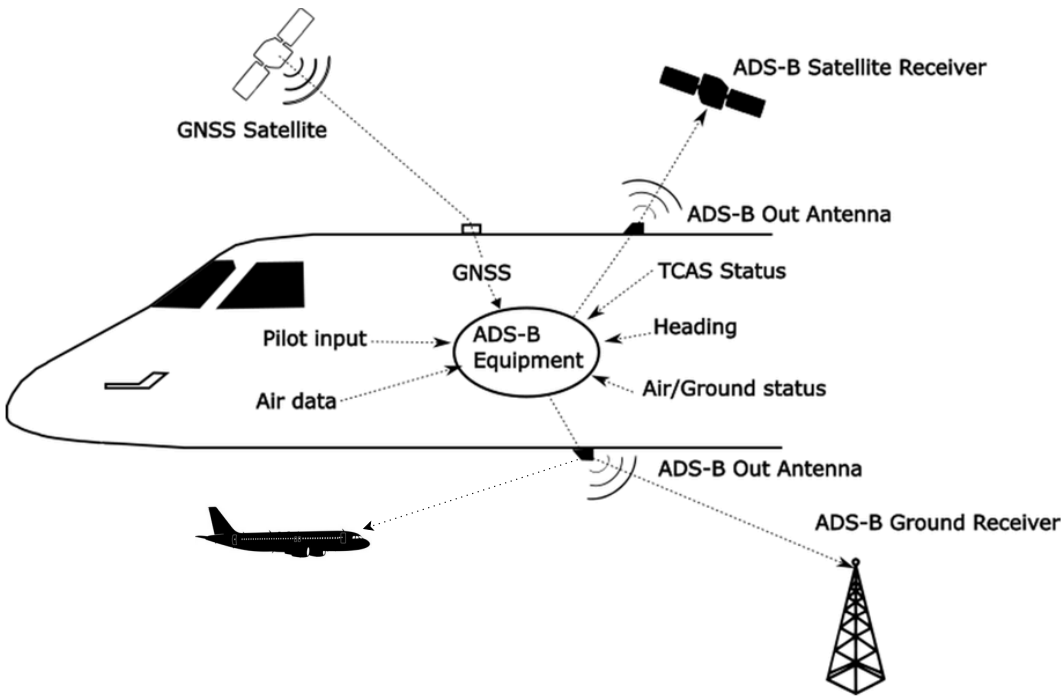


Figure 1.1 Aircraft equipped with ADS-B Out capability consistently broadcast their GPS-derived position along with other onboard sensor data. These transmissions are intercepted by ADS-B In systems on nearby aircraft, ground stations, and satellites, thereby supporting air traffic surveillance and aircrew situational awareness. Adapted from [2].

an onboard ADS-B In receiver and then processed and presented to the aircrew through a visual display and aural warnings [16]. In commercial and business aircraft, such ACAS capability is provided through Traffic Alert and Collision Avoidance System (TCAS)-compliant hardware. Airborne TCAS systems actively interrogate the transponders of nearby aircraft, whose replies are used to compute their relative position, and if needed, issue Traffic Advisory (advisory information) or Resolution Advisory (traffic information requiring immediate and mandatory action) messages. TCAS deployments are mandatory in certain types of aircraft and are subject to strict airworthiness standards [17].

In contrast, ADS-B-based ACAS solutions are only advisory and not certified; however, they provide an attractive and affordable alternative for light to medium General Aviation (GA) aircraft (where the cost of a TCAS solution is often more than the value of the aircraft itself). In ADS-B-based ACAS solutions, the reception and processing of ADS-B messages are typically achieved through a compatible transponder or a dedicated ADS-B In receiver, and traffic information is typically displayed on the transponder screen, a compatible GPS unit, a multifunction display (MFD), or a handheld tablet, for example, one used by the

crew as an Electronic Flight Bag (EFB) [16]. In flights conducted under Visual Flight Rules (VFR), where traffic avoidance is the sole responsibility of the aircrew (i.e., a significant portion of flights conducted by light and medium aircraft), the use of such an affordable alternative provides an important and non-negligible safety benefit. This is especially true for VFR flights conducted in congested low-level airspace with a high density of light aircraft, such as training airports and GA fly-ins.

In addition to enhancing safety, ADS-B offers a wide range of operational benefits. First, by providing precise and continuous position data, ADS-B enables more efficient routing and reduced separation between aircraft. This, in turn, allows for optimized flight paths that lower fuel consumption and increase the overall airspace capacity. Second, compared to traditional radar infrastructure, ground-based ADS-B systems are significantly less expensive to deploy and maintain, making them a cost-effective surveillance solution [10, 18, 19]. In particular, they offer a practical alternative for countries and regions in which radar systems are not economically viable. Third, ADS-B supports additional services in the United States. Through the Flight Information Service–Broadcast (FIS-B), ground stations transmit various types of aviation information, including NEXRAD weather radar images, Automatic Terminal Information Service (ATIS), and Notices to Air Missions (NOTAM). To date, FIS-B remains the only service of its kind worldwide [20]. Finally, ADS-B has provided an open and incredibly rich source of fine-grained data on air traffic, which has enabled many non-operational applications, such as flight tracking by the public and analysis of air traffic trends.

Owing to its significant benefits, ADS-B use has been mandated in many airspaces. In Canada, ADS-B Out is required for operations in Class A and B airspace above 12,500 feet above sea level (ASL), as outlined by NAV CANADA [21, 22]. Within the European Union, ADS-B is required for all aircraft operating under IFR if the aircraft has a maximum takeoff weight (MTOW) exceeding 5,700 kg or a maximum cruising true airspeed (TAS) greater than 250 knots [23, 24]. In the United States, the Federal Aviation Administration (FAA) mandates ADS-B Out in Class A, B, and C airspace, as well as in Class E airspace at and above 10,000 ft mean sea level (MSL), excluding airspace at or below 2,500 ft above ground level (AGL) [25].

1.2 How can the security of ADS-B be strengthened?

ADS-B is mandated in many countries, yet it remains vulnerable to cyberattacks. The system lacks built-in mechanisms to authenticate messages or at least ensure their integrity [26–29]. Using readily available and inexpensive equipment, such as software-defined radio (SDR), malicious actors can effortlessly inject false messages into the ADS-B system [30, 31]. If left

unaddressed, the exploitation of such a vulnerability could compromise the effectiveness of ADS-B in all its intended applications. Should this risk materialize, it could lead to a wide range of impacts, including economic, safety-related, and geopolitical.

From an economic perspective, the injection of tampered position reports may cause flight delays or cancellations, resulting in substantial financial losses for airlines and airports. From a safety standpoint, this could create hazardous situations, such as separation conflicts, unnecessary maneuvers, or misleading ATC instructions, all of which are recognized contributing factors in aviation incidents and accidents. From a geopolitical perspective, targeted spoofing attacks near national borders or in politically sensitive regions can provoke diplomatic tensions or lead to precautionary measures such as airspace restrictions or military alertness [32].

At first glance, addressing the security shortcomings of ADS-B might seem straightforward: adding an authentication layer to the protocol to overcome its fundamental weakness, namely, the absence of an authentication mechanism in its design. This is typically achieved through cryptographic techniques. However, implementing such a solution in practice is complex and challenging. In addition to ensuring the authenticity of the message and its originator, any proposed approach must comply with strict operational and technical constraints, which we classify into two main categories: safety and rapid deployment requirements.

From a safety standpoint, the ADS-B Minimum Operational Performance Standard (MOPS) requires that all receivers be capable of parsing ADS-B messages [12]. This implies that any proposed security solution must be backward compatible, ensuring that legacy ADS-B In receivers — those that do not support the secure version of the protocol — can still interpret broadcasts from ADS-B Out systems that implement the new mechanism. Additionally, the solution must be bandwidth efficient to minimize its impact on channel occupancy. Currently, ADS-B messages are primarily transmitted over the 1 090 MHz frequency using the Extended Squitter (ES) format in mode S [33]. This frequency, also known as 1090ES, is used not only for ADS-B but also by other surveillance systems such as radar, multilateration, and TCAS. In the United States, an alternative frequency, the Universal Access Transceiver (UAT) at 978 MHz, is used for TIS-B and FIS-B services [33]. Both channels already experience congestion in high-density airspace, which poses a major constraint on the integration of authentication into ADS-B, as security comes at the cost of an increased payload size.

Regarding the rapid deployment, changes in aviation require extended implementation timelines owing to aircraft longevity, stringent safety regulations, and international standardization processes. Unlike IT systems, where security patches can be deployed in a matter of weeks, updates in aviation may take years to be fully adopted. Consequently, any ADS-B security solution must streamline the certification procedures and enable timely deployment.

Software-based updates that avoid hardware changes are a practical path forward, as updating firmware on existing avionics or ground stations is faster and less costly than replacing components, making it a more viable operational option.

Unfortunately, none of the cryptographic schemes proposed thus far for authenticating ADS-B broadcasts simultaneously satisfy all three requirements mentioned above. To ensure backward compatibility, the authentication mechanism must preserve the original ADS-B message in cleartext, which can be achieved by appending an authentication code or a signature to the message. One such method is the use of Message Authentication Codes (MAC), a symmetric cryptography approach in which the sender and all receivers share a common symmetric key: the sender generates the MAC using this key, and each receiver verifies it upon reception [34, 35]. The main advantage of this approach is the short length of the authentication codes, which supports bandwidth efficiency. However, MAC-based authentication is ill-suited to the open nature of ADS-B broadcasts and the security of the system [36]. Because all receivers possess the same symmetric key, any of them can impersonate the sender by forging and retransmitting messages, rendering this approach inherently insecure [36].

In contrast, digital signatures rely on asymmetric cryptography, where two keys are used: a private key, known only to the sender and used to produce the signature, and a public key, known to all receivers and used to verify the message's authenticity [3, 37]. However, there is no consensus on how to manage aircraft public keys. Some authors advocate for the use of a Public Key Infrastructure (PKI), which is the standard solution in IT, where public keys are distributed and authenticated via certificates [38]. Although secure and well established, the absence of a PKI for managing aircraft public keys in aviation poses a challenge. While its implementation is both feasible and necessary, it requires a global organizational effort, which hinders rapid deployment. As an alternative, other authors have proposed certificateless digital signature schemes, such as Identity-Based Signatures (IBS) [39] and Certificateless Public Key Cryptography (CL-PKC) [1]. These approaches eliminate the need for certificates but introduce other risks: identity-based schemes are vulnerable to key escrow [40], while certificateless schemes may be compromised by malicious Key Generation Centers (KGC) [41, 42].

Hybrid schemes, particularly those based on the Timed Efficient Stream Loss-tolerant Authentication (TESLA) protocol [43], offer an improved trade-off between security and bandwidth efficiency compared to purely symmetric or asymmetric approaches. In TESLA, each message is transmitted with a MAC, and the corresponding authentication key is disclosed only after both the message and MAC are received. This delayed key disclosure thwarts message forgery and enhances security, while the use of MAC helps preserve the 1090ES band-

width. Although previously proposed hybrid schemes have not fully overcome the inherent trade-off between security and bandwidth efficiency, recent updates to the MOPS specifications provide promising development. Specifically, the possibility of using phase overlay modulation techniques at the physical layer of ADS-B to convey additional data. This capability offers a possible path toward implementing TESLA-based authentication within the bandwidth constraints of the ADS-B system.

In addition to broadcast authentication, previous studies have suggested methods for verifying the integrity of ADS-B messages. Although cryptographic solutions are essential, they are not foolproof; authentication keys can be stolen, leaked, or misused. Therefore, a defense-in-depth strategy that integrates both preventive and detective security measures is crucial. In this regard, it is equally important to implement detection mechanisms on the receiver side to address the residual risks associated with preventive approaches, such as broadcast authentication.

The proposed detection methods use a variety of tools and approaches, including multilateration (MLAT), data fusion, Kalman filtering, ontologies, and machine learning. Although MLAT and data fusion approaches are implemented in certain operational settings, they still face limitations. The MLAT is impractical in remote areas, and data fusion can yield inaccurate results owing to timing discrepancies, inconsistent data formats, and varying noise levels across the fused data sources [29, 44]. In recent years, there has been growing interest in deep learning (DL), which has shown strong promise in cybersecurity applications such as malware and network intrusion detection [45–47].

DL is particularly well-suited for ADS-B, as messages form multivariate time series describing an aircraft's position and movement over time [48]. This trajectory must adhere to the physical laws, aviation regulations, and operational capabilities of the aircraft. Consequently, any intrusion detection model for ADS-B must retain relevant information from past observations to evaluate the consistency of aircraft motion. In addition, detecting gradual and sophisticated attacks, where subtle deviations are introduced progressively across successive messages, requires models capable of capturing long-term temporal dependencies.

Early approaches, particularly those using Long Short-Term Memory (LSTM) [49] based autoencoders, faced challenges in this regard. Although LSTMs are designed to process sequential data and maintain internal states through gated mechanisms, their fixed-size memory limits their capacity to preserve the long-term context. Once information is gated out or overwritten, it is effectively lost [6]. As the American painter Kenneth Noland once said, *"Context is the key—from that comes the understanding of everything"*. Indeed, the loss of past information reduces the model's ability to accurately detect gradual attacks conducted

over long periods of time.

To overcome these limitations, recent IDS architectures for ADS-B have focused on improving the context modeling. Contextual autoencoders (CAE) [50] employ a shared encoder with multiple decoders, each tailored to a specific flight phase, whereas Transformers [5,51] leverage self-attention mechanisms to capture long-term dependencies more effectively than the gating mechanism of LSTM [49]. Although both models achieved similar detection performance for subtle gradual attacks — an F1 score of 93.9 for CAE [50] and 94.0 for the transformer-based model [51] — this level is still insufficient from a safety perspective. Moreover, it is well known that transformers scale quadratically with the sequence length, resulting in increased computational costs [52]. This can delay the response time of the IDS, raising further safety concerns.

Consequently, it is essential to explore architectures that balance detection performance and scalability. One promising approach is the Extended LSTM (xLSTM), a recent variant of the traditional LSTM [6]. xLSTM introduces two key enhancements: exponential gating and matrix-valued memory cells. Whereas conventional LSTMs use rigid sigmoid gates that irreversibly discard information [49], xLSTM employs exponential gates that allow the memory to decay gradually. This facilitates the dynamic reweighting of past information, thereby enabling improved modeling of long-term dependencies. In parallel, xLSTM replaces vector-valued memory cells with matrix-valued ones, enriching the internal state and capturing more complex temporal patterns in multivariate data such as ADS-B. Despite its recurrent nature, xLSTM benefits from structured memory and vectorized operations that improve stepwise efficiency. More importantly, its computational complexity scales linearly with the sequence length, in contrast to the quadratic growth associated with attention-based models. These properties make xLSTM a compelling candidate for ADS-B intrusion detection, where both inference speed and detection accuracy are critical factors [6].

1.3 Research objective and associated research questions

Previous efforts to secure the ADS-B protocol have revealed significant limitations. Broadcast authentication schemes have failed to achieve an effective trade-off between safety and security. In terms of data integrity verification, DL-based IDS have shown limited ability to detect sophisticated or novel attacks, raising safety concerns. These concerns can be further exacerbated by inference delays, which often result from the complex operations of recent DL architectures used in some IDS implementations.

Given these challenges, the question of how to properly secure ADS-B while meeting the

technical and operational requirements that underpin its safety remains unanswered.

The research objective of this thesis is to propose and implement effective solutions to secure ADS-B, while also assessing their operational viability. To support a defense-in-depth strategy, we adopt two complementary approaches: broadcast authentication as a preventive measure and DL-based intrusion detection as a detective measure.

The first line of defense involves implementing a cryptographic scheme that combines the TESLA authentication protocol (for message authentication and backward compatibility) with a PKI (for sender identity authentication), while leveraging physical-layer optimization techniques recently defined in the MOPS DO-260C, primarily to ensure bandwidth efficiency. Our approach aims to achieve a more balanced trade-off between safety and security. To evaluate the practicality and effectiveness of this approach, the following research questions were investigated:

RQ1: To what extent can a TESLA-based authentication scheme be integrated into ADS-B while maintaining backward compatibility and ensuring efficient bandwidth utilization?

RQ2: Are the delays introduced by the proposed solution compliant with the timing thresholds defined by TCAS and consistent with the update delays observed in ATC surveillance systems?

RQ3: Is the bit error rate of the carrier's quadrature component, which conveys security information, within acceptable limits to guarantee that the proposed solution maintains overall quality of service?

The second line of defense introduces a DL-based IDS built on the novel xLSTM architecture, which leverages exponential gating and matrix-valued memory cells to better capture long-term dependencies while maintaining fast stepwise inference through linear scaling with sequence length [53, 54]. We train the model using transfer learning to enhance generalization and improve the detection of sophisticated or previously unseen attacks. To assess whether it offers both faster inference and better detection performance, we implemented a comparative IDS using a transformer architecture, which previously yielded the best results for detecting sophisticated attacks on ADS-B. Both models share the same transfer learning setup, allowing a fair evaluation of their contribution to safety. Therefore, this thesis addresses the following research question:

RQ4: What are the most effective architectures and learning strategies for designing and developing innovative, high-performance IDS solutions for ADS-B?

1.4 Thesis outline

Table 1.1 Overview of Thesis Chapters and Contributions

Chapter	Main Contribution	Description
2	Literature review	Evolution of ATC surveillance systems.
3	Original research	Analysis of ADS-B security and key performance requirements.
4	Literature review	Comparison of cryptographic approaches for ADS-B authentication.
5	Literature review	Survey of ML techniques for ADS-B data integrity.
6	Original research published work	Proposal and evaluation of the CABBA protocol.
7	Original research published work	ML-based intrusion detection for ADS-B.
8	Synthesis of works	Summary of contributions, limitations, and future research.

The remainder of this thesis is organized as follows. Chapter 2 offers an in-depth review of the progression of ATC surveillance technologies, tracing the shift from early visual cue-based methods to the implementation of ADS-B. Chapter 3 examines ADS-B security, its vulnerabilities, how they can be exploited, and the resulting risks to ATC and situational awareness. Furthermore, it specifies the security, safety, and deployment criteria that any proposed security solution must meet to be considered for its adoption. Chapter 4 investigates existing cryptographic approaches to ADS-B broadcast authentication and assesses symmetric, asymmetric, and hybrid methods in terms of security, backward compatibility, and communication overhead. Chapter 5 reviews the machine learning techniques for ADS-B data integrity verification, highlighting the strengths and limitations of the current anomaly detection and classification methods in terms of detection performance and inference times. Chapter 6, based on our published work [55], introduces the bandwidth-efficient broadcast authentication protocol, CABBA, designed for ADS-B and evaluates its performance relative to existing solutions. Chapter 7, also based on our published work [56], explores innovative machine learning approaches for intrusion detection in ADS-B, presenting the proposed models, experimental design, and results. Finally, Chapter 8 concludes the thesis with a summary of the contributions, limitations and directions for future research.

CHAPTER 2 FROM VISUAL SIGNALS TO ADS-B: EVOLUTION OF ATC SURVEILLANCE SYSTEMS

This chapter reviews the evolution of air traffic control (ATC) surveillance technologies from early visual techniques to the implementation of ADS-B.

2.1 Historical background and early techniques

In 1903, the Wright brothers completed their first sustained, powered flight, an achievement that lasted only twelve seconds but marked the dawn of modern aviation. This milestone paved the way for the creation of the first practical airplane in 1905 and sparked a worldwide surge of aviation innovation. Over the subsequent decades, aircraft design and performance improved steadily. Aviation also began to diversify in its applications, with World War I accelerating military advancements and early airmail services demonstrating its potential for civilian use [57].

In the early years, civil aviation was fraught with significant risk. Pilots navigated under challenging conditions, relying mainly on magnetic compasses for guidance. They flew at low altitudes to follow roads and railways, usually ranging from 200 feet to 500 feet. During periods of poor visibility, bonfires were lit on airfields to assist with landing [57]. Despite frequent fatal accidents, the commercial aviation industry expanded, with 30 airlines operating worldwide in the early 1920's [58].

In 1920, Croydon Airport near London was the first to implement air traffic control. The control tower was a 15-feet (5 m) wooden hut with windows on all sides. The controllers provided pilots with traffic, weather, and position updates via radio when available onboard. In 1922, formal traffic control procedures were introduced following two incidents [58–60]. A minor ground collision at Croydon led to the requirement for pilots to receive sequence numbers and takeoff clearance, as indicated by a red flag waved from the control tower [60]. A second fatal mid-air collision near Paris during poor weather resulted in seven deaths [61]. The British Air Ministry reacted by instituting regulations that continue to be enforced today. These include the creation of designated air corridors connecting major European cities, the requirement for pilots to remain within these corridors and maintain a right-hand flight path, the need for improved cockpit visibility in newly designed aircraft, and the mandatory installation of radio stations on all aircraft [58–60].

In the United States, significant changes also occurred in air navigation infrastructure. Au-

Figure 2.1 At left: controllers at Newark Air Route Traffic Control Center (1936), managing en-route flights without direct radio contact with pilots. At right: controller at the Cleveland Airport Control Tower (1937), where two-way radio communication with aircraft was already established for local air traffic control. Sources: [62, 63].

thorities constructed towers equipped with rotating beacons for night flights, and by 1923, a fully illuminated route spanned the country. The first American control tower featuring two-way radio communication was established at the Cleveland Airport in 1930. By 1932, the beacon system was upgraded to radio beacons, creating a network of 83 stations. This advancement allowed pilots to navigate using radio signals instead of visual cues, thereby enhancing safety and enabling basic triangulations. In 1936, the first three Air Route Traffic Control Centers (ARTCC) were established in Chicago, Cleveland, and Newark [57]. As depicted in Figure 2.1, these centers monitored aircraft traveling along the airways using paper maps and blackboards. They relied on reports from towers and radio operators because they lacked direct communication with pilots [57, 62]. Direct radiotelephone communication between aircraft and an ARTCC began in 1949 at the Chicago ARTCC [62].

In 1952, a new era in ATC began as radar systems were formally integrated into operational procedures, enabling controllers to track aircraft positions with a level of precision far surpassing that of visual observations or position reports conveyed through radiotelephony [64].

2.2 Radar-based surveillance systems

Radar technology, initially developed by the British and first employed for military applications during World War II, was subsequently integrated into the civil aviation industry. Two primary types of radar are utilized in the field of surveillance: Primary Surveillance Radar (PSR) and Secondary Surveillance Radar (SSR) [65].

Figure 2.2 Above the white support structure are two collocated radar antennas: the lower one, with a curved profile, is a PSR, and the upper one, flat and rectangular, is an SSR. Source : [66].

2.2.1 Primary Surveillance Radar (PSR)

Primary surveillance radar operates as a non-cooperative system, designed to detect aircraft without necessitating their active participation. This radar functions based on the principle of echolocation. The rotating antenna of the PSR emits short electromagnetic pulses within the 1–2 GHz or 3–4 GHz frequency bands. Upon striking an aircraft, a portion of the signal is reflected back to the radar. By analyzing this reflection, the radar determines the position of the target by measuring both the slant range (distance) and the azimuth (direction). The azimuth is derived from the angular orientation of the rotating antenna at the time the reflection is received, while the slant range is calculated based on the time delay between the pulse emission and the reception of the reflected signal. Once these measurements are obtained, the radar system processes the data and transmits them to the ATC service, where controllers can visualize. Although PSR systems are effective in determining aircraft positions, they do not provide identification or altitude data. To address this limitation, secondary surveillance radar (SSR) was developed [65].

2.2.2 Secondary Surveillance Radar (SSR)

Secondary Surveillance Radar operates as a cooperative system by transmitting a coded interrogation pulse at 1 030 MHz to aircraft. In response, the transponder (transmitter–responder) onboard the aircraft sends a coded reply at 1 090 MHz. SSR operates in three modes or communication protocols: Mode A, Mode C, and Mode S. Mode A enables the acquisition of aircraft identification codes. Mode C builds upon this by including the barometric altitude in the response. Mode S, the most advanced, allows for selective interrogations and supports the retrieval of additional data such as velocity, position and operational status. Despite the enhanced monitoring capabilities offered by SSR, the system has several limitations. It is costly to deploy and maintain, and its coverage is inherently confined to regions equipped with secondary radar infrastructure. Consequently, large portions of the globe, particularly maritime areas, which represent approximately 71 percent of the Earth's surface, fall outside its surveillance range owing to the impracticality of installing radars in such environments. Likewise, remote or inaccessible terrestrial regions may lack adequate SSR coverage. In addition, the accuracy of the transmitted data, such as the barometric altitude, is limited by the precision of the sensors onboard the aircraft. The introduction of Automatic Dependent Surveillance–Broadcast (ADS-B) was intended to address these limitations [65].

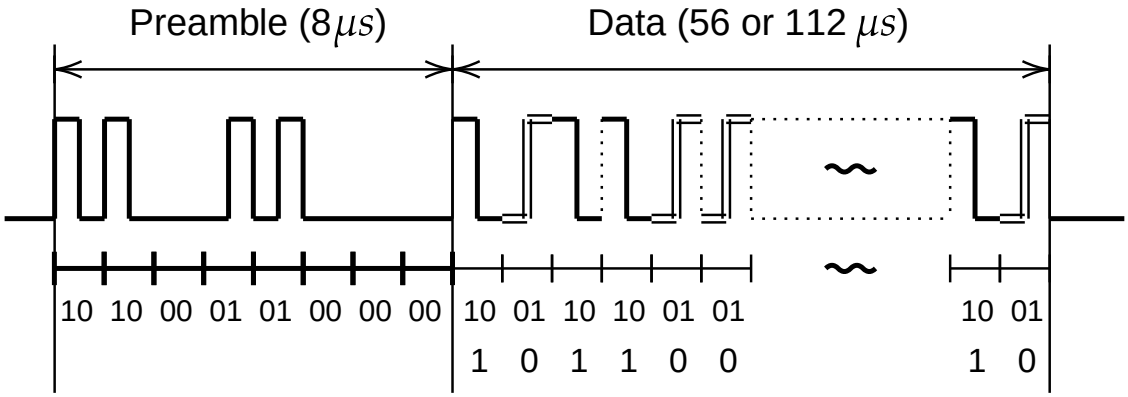


Figure 2.3 Mode S downlink message pulses begin with a preamble, followed by a 56- or 112-bit message encoded using Manchester encoding and transmitted at 1 Mbps. Reprinted from [65], with permission.

2.3 Mode S

Before exploring ADS-B, it is crucial to first understand the Mode S communication protocol. Developed by MIT Lincoln Laboratory in the late 1970's [67], Mode S supports two downlink message formats: short 56-bit messages and extended 112-bit messages (see Figure 2.3, [65]). This protocol is used in SSR, TCAS, and ADS-B systems. As previously mentioned, SSR employs the Mode S communication protocol to identify and track aircraft by sending interrogations at 1030 MHz and receiving aircraft replies at 1 090 MHz. TCAS uses the same interrogation/reply mechanism. However, in TCAS, Mode S is employed to evaluate collision risks and provide avoidance advisories [65, 67]. In ADS-B systems, Mode S is used in the Extended Squitter format, meaning that messages are 112 bits long (extended) and are automatically broadcast (squitter) without requiring interrogation from another system [12].

2.4 ADS-B: A satellite-enabled surveillance system

ADS-B technology plays a crucial role in the ongoing global modernization of air traffic management systems. It serves as the backbone for major initiatives, such as NextGen in the United States and SESAR in Europe, both of which are designed to improve the safety, efficiency, and capacity of air navigation services [9]. A key aspect of these initiatives is the transition from traditional radar-based to satellite-enabled surveillance systems [11].

The first step in this transition involved the use of Global Navigation Satellite Systems (GNSS), such as GPS, to enhance the accuracy and reliability of aircraft position data. GNSS satellites provide precise positioning information that forms the foundation of ADS-B surveillance and enables more accurate aircraft tracking compared to conventional radar [9, 11].

More than a decade later, the development of space-based ADS-B further extended the surveillance capabilities. Unlike GNSS, which supports positioning, space-based ADS-B relies on communication satellites to receive ADS-B signals transmitted by aircraft flying over remote or oceanic regions, thereby extending coverage beyond the range of ground-based receivers [15]. A notable example of this capability is the Aireon service developed by Aireon LLC, a joint venture established in 2012 by NAV CANADA, Iridium Communications Inc., and other partners. Aireon employs the Iridium constellation, a network of 66 Iridium NEXT satellites equipped with space-qualified ADS-B receivers, to enable real-time global aircraft monitoring [68].

2.4.1 Operational uses of ADS-B

Surveillance by ATC At the operational level, ADS-B enables each aircraft to periodically broadcast its navigation data — including position, altitude, velocity, identity, and operational status — without requiring interrogation by SSR or other sensors. These ADS-B Out transmissions are captured by ground-based receivers and, in remote or oceanic regions, by satellite receivers that relay the data to the ground stations.

ADS-B serves two major functions in ATC. First, it complements SSR and multilateration (MLAT) systems through data fusion, providing a more accurate and redundant surveillance picture in regions with radar infrastructure. Second, it extends surveillance coverage to areas where SSR or MLAT are unavailable, such as oceanic or sparsely equipped regions [12, 14]. This expanded coverage supports the safe management of air traffic in regions that were previously outside radar range, marking a major advance in global airspace surveillance capabilities.

Flight crew situational awareness Beyond ATC surveillance, ADS-B also enhances situational awareness for flight crews. Aircraft equipped with ADS-B In receivers can intercept nearby ADS-B Out transmissions and use this information to visualize the surrounding traffic in real time [2, 12]. This capability is particularly valuable for light and medium general aviation aircraft that lack TCAS-compliant hardware, as it offers a cost-effective means of improving traffic awareness.

In aircraft equipped with a hybrid TCAS (TCAS II) [17], ADS-B can further optimize operations through hybrid surveillance. In these systems, ADS-B data supplement radar-based TCAS surveillance, thereby reducing the need for frequent active transponder interrogations. When nearby aircraft remain outside conflict thresholds, the TCAS can passively monitor them using ADS-B broadcasts, resorting to active interrogation only when a potential conflict is detected based on range, closure rate, or altitude separation [17, 69].

2.4.2 Components of the ADS-B system

The ADS-B system consists of three components, as shown in Figure 2.4: transmitting subsystems with ADS-B Out capability that broadcast ADS-B messages, receiving subsystems with ADS-B In capability that receive ADS-B messages from nearby aircraft, and the communication channel or propagation medium [12].

The International Civil Aviation Organization (ICAO) originally considered three candidate data links for ADS-B: the 1 090 MHz extended squitter (1090ES), the VHF Data Link Mode 4

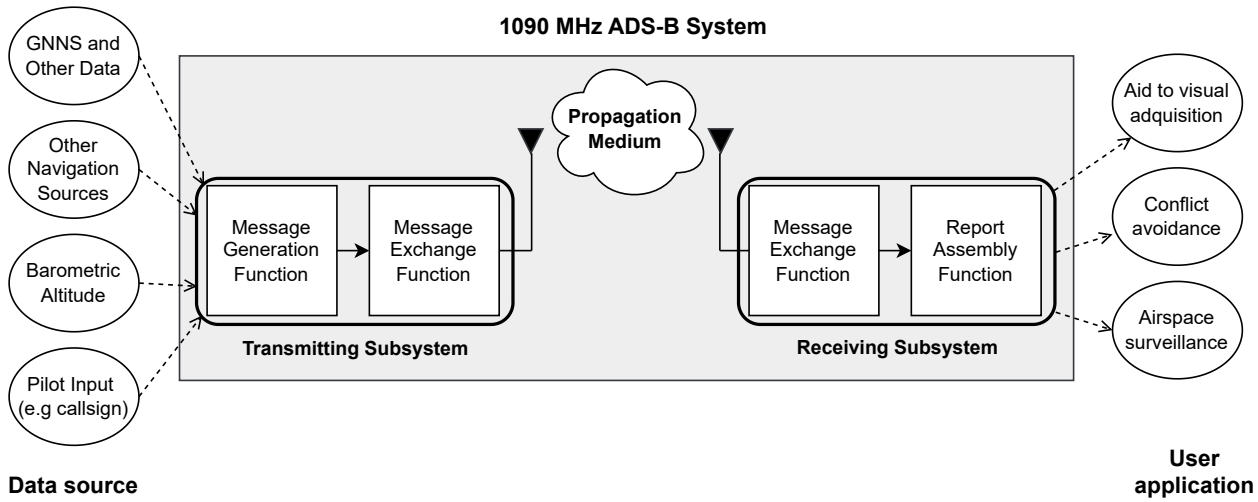


Figure 2.4 Overview of the 1 090 MHz ADS-B system, which comprises the transmitting subsystem, receiving subsystem, and the propagation medium, specifically 1 090 MHz. For other propagation media, such as UAT and VDL-4, the internal functions within the subsystems may exhibit slight variations.

(VDL-4), operating in the 108–136.975 MHz range, and the Universal Access Transceiver (UAT), operating at 978 MHz. Currently, only 1090ES and UAT remain in operational use, as VDL-4 has been largely superseded by 1090ES. UAT is primarily used in the United States for Traffic Information Service–Broadcast (TIS-B) and Flight Information Service–Broadcast (FIS-B), and ADS-B Out transmission on UAT is not mandatory. In contrast, 1090ES is the mandated data link for ADS-B Out in most regions worldwide [12, 33].

This thesis therefore, focuses exclusively on the 1 090 MHz ADS-B system. Besides being the globally mandated data link, it also represents the most congested and bandwidth-limited channel [12]. Consequently, any security or safety solution validated under these conditions can be expected to remain effective on the less congested UAT or VDL-4 data links.

2.4.3 ADS-B Messages: Packet Structure, Types and Transmission Rates

DF (5 bits)	CA (3 bits)	ICAO (24 bits)	ME (56 bits)	PI (24 bits)

Figure 2.5 ADS-B packet structure

An ADS-B packet is composed of 112 bits and is divided into five distinct fields, as depicted in Figure 2.5. These fields include:

`DF (Downlink Format)` — `5 bits`
Indicates the source of the ADS-B message.
DF = 17 indicates a Mode S transponder message.
DF = 18 indicates a message from a non-transponder-based ADS-B system.

`CA (Capability)` — `3 bits`
Specifies the transponder level. Values range from 0 to 7.

`ICAO Address` — `24 bits`
Unique identifier assigned to each aircraft.

`ME (Message Field)` — `56 bits`
Contains the message `Type Code (TC)` and the payload.
`TC (5 bits)`: indicates the nature of the information.
There are seven types of ADS-B messages, each with its own transmission rate (see Table 2.1).

`PI (Parity Identifier)` — `24 bits`
Used for error detection and message validation.

The transmission rate of ADS-B messages varies depending on whether the aircraft is airborne or on the ground, and whether it is stationary or moving. The rates are listed in Table 2.1

Messages	**TC**	**Ground (still)**	**Ground (moving)**	**Airborne**
Aircraft identification	1–4	0,1 Hz	0,2 Hz	0,2 Hz
Surface position	5–8	0,2 Hz	2 Hz	-
Airborne position	9–18, 20–22	-	-	2 Hz
Airborne velocity	19	-	-	2 Hz
Aircraft status	28	0,2 Hz *(no TCAS RA and Squawk Code change)*		
		1,25 Hz *(change in TCAS RA or Squawk Code)*		
Target states and status	29	-	-	0,8 Hz
Operational status	31	0,2 Hz	0,4 Hz *(no NIC/NAC/SIL change)*	
			1,25 Hz *(change in NIC/NAC/SIL)*	

Table 2.1 Transmission rates of ADS-B messages depending on aircraft state.

2.4.4 Message transmission process

The transmitting subsystem is responsible for generating and broadcasting the ADS-B messages. It consists of two main functions: message generation and message exchange. Initially, navigation data collected from onboard sensors are processed by the message generation function, where they are converted into a 56-bit binary code and embedded into a 112-bit ADS-B packet. Subsequently, the message exchange function applies pulse position modulation (PPM) to the packet. In PPM, each bit period is divided into two equal halves: Bit 1 is represented by a high signal in the first half and a low signal in the second half; for Bit 0, the sequence is reversed. After the PPM modulation is finalized, the resulting signal is used to modulate a 1 090 MHz carrier wave, which is subsequently broadcast over the air at a data rate of 1 [12, 65].

Notably, since 2020, MOPS [12] has incorporated the concept of *phase overlay capability*, which enables the use of alternative modulation techniques in the physical layer of ADS-B to increase data throughput without raising channel activity. This feature is optional; however, the Radio Technical Commission for Aeronautics (RTCA) [1] introduced it in the MOPS so that stakeholders could begin designing and testing equipment with this functionality [55, 70]. This approach is detailed in a patent [71]. The RTCA recommends using 8-phase differential phase shift keying (D8PSK) modulation together with error correction codes, such as Reed–Solomon (RS) or Low-Density Parity-Check (LDPC), to encode additional information within a standard 1090ES message. Of the 336 extra bits provided by D8PSK, 204 are available for supplementary data.

2.4.5 Message reception and integration in ATC system

The receiving subsystem captures, demodulates, and decodes the ADS-B messages transmitted by a nearby aircraft. It comprises two main functions: message exchange and report assembly. Reception begins with an antenna tuned to 1 090 MHz that captures the incoming modulated signal. In the message exchange function, this signal is processed by a radio frequency (RF) front-end, filtered, amplified, and down-converted to an intermediate frequency (IF) or baseband. The demodulator then recovers the original PPM-encoded bitstream by detecting signal transitions within each bit period, and reconstructs the 112-bit ADS-B packet. In the report assembly function, the packet is decoded to extract navigation data, such as position. The decoded information is then formatted into a standard surveillance report and made available to onboard systems or relayed to air traffic control units

[1] RTCA is a U.S. nonprofit organization that develops performance standards and guidelines for aviation systems, including communication, navigation, surveillance, and air traffic management.

through an appropriate communication interface [12].

For systems employing the phase overlay capability, the received signal must be processed through two parallel demodulation paths: one to extract the standard ADS-B information as described above, and a second to recover the additional data encoded using D8PSK and the associated error-correction scheme, which may include security-related information.

2.5 Summary of the chapter

In this chapter, the evolution of air traffic control systems is explored, beginning with early visual observation methods and progressing to radar-based technologies, including both Primary Surveillance Radar (PSR) and Secondary Surveillance Radar (SSR). The chapter then provides a comprehensive overview of the ADS-B protocol, describing its message structure and operational principles. Finally, it highlights the optional phase overlay capability introduced in the latest version of the ADS-B MOPS (DO-260C, 2020) [12]. This feature was made optional by the RTCA to enable stakeholders to design and test equipment supporting higher data throughput without increasing channel activity, thereby preparing ADS-B for future enhancements [70].

CHAPTER 3 ADS-B SECURITY ANALYSIS AND PERFORMANCE REQUIREMENTS

This chapter delves into the security of ADS-B, exploring its vulnerabilities and outlining the essential criteria that any solution must satisfy to ensure its safe implementation.

3.1 Security Properties

Before delving into the security analysis of ADS-B, it is crucial to revisit the core principles that form the foundation of cybersecurity [1].

Computer security focuses on protecting data, software, hardware, communication networks, and the physical systems they control from unauthorized actions. This protection can be achieved either by preventing such actions or by detecting and recovering from them. The overall objective is to ensure that computer-based services maintain the security properties that allow them to operate reliably and as expected [72].

These security properties are generally defined as follows [73]:

- **Confidentiality:** is the assurance that data cannot be viewed by an unauthorized user.
- **Integrity:** is the assurance that data has not been altered in an unauthorized manner.
- **Data origin authentication :** is the assurance that a given entity was the original source of received data.
- **Entity authentication:** is the assurance of the identity of a given entity interacting with a system.
- **Non-repudiation:** is the assurance that an entity cannot deny a previous commitment or action.
- **Availability:** is the assurance that services and resources remain accessible when needed.

To fulfill operational and safety requirements, ADS-B must primarily ensure entity and message origin authentication, confirming that each broadcast originates from a legitimate sender

[1] In this thesis, the terms cybersecurity and computer security are used interchangeably.

(aircraft or any authorized system) and remains unaltered during its transmission. This condition inherently implies that ADS-B must also guarantee data integrity, as data origin authentication represents a stronger notion that encompasses integrity.

Availability is equally important, as the continuous reception of these broadcasts is vital for safe air traffic monitoring. As previously mentioned, confidentiality is not a primary concern for ADS-B because the system is designed to broadcast information openly so that all receivers can read the information.

3.2 ADS-B security analysis

3.2.1 Attack taxonomy

The primary attacks that threaten the ADS-B system have been summarized in prior work [26, 28, 30], such as:

- **Eavesdropping:** The passive interception of genuine ADS-B messages.
- **Jamming:** The disruption of RF channels to prevent the transmission of genuine ADS-B messages.
- **Message deletion:** Suppression or removal of genuine ADS-B messages.
- **Message modification:** The alteration of genuine ADS-B messages, potentially falsifying aircraft position, velocity, or identity.
- **Message injection:** The transmission of fabricated ADS-B messages, potentially introducing spoofed aircraft or falsified flight data into the surveillance network.

In this research, we intentionally omit eavesdropping, jamming, and message deletion attacks from consideration. Eavesdropping is not viewed as a direct threat unless it is paired with active attacks [28]. Jamming is a common issue in radio frequency (RF) systems and is generally straightforward to detect. Likewise, message deletion attacks, when executed on their own, are easily identified due to the obvious gaps they create in aircraft tracking data [74].

Our main emphasis is on message modification and message injection attacks. In the subsequent paragraphs, we refer to these two types of attacks collectively as spoofing.

3.2.2 Threat Model

To evaluate the security risks to ADS-B in its present context, we define a threat model centered on spoofing. In surveillance applications, there are three methods by which spoofed information can be inserted into the ATC system.

Compromise of IT infrastructure Competent malicious actors with knowledge and capacity could hack into the ANSP IT infrastructure used to operate the ATC system. By installing malware and remote access tools (RAT) the attacker could inject false information about aircraft position straight into the ATC applications. Note that this threat is not specific to ADS-B, and we will not consider it further in this thesis.

Spoofing of ADS-B network packets Without hacking into the ATC system *per se*, resourceful attackers could change the content of ADS-B position report information *in transit*, for example, by compromising telecommunication network links between ADS-B ground stations and ATC centers, or equivalently from Iridium ground-stations to ATC centers. While this is not an ADS-B specific threat either, and there are adequate IT cybersecurity countermeasures to address this (at the physical and network layers), the introduction of message authentication in ADS-B could provide an additional layer of protection, e.g. by forwarding authentication information to the final destination for independent verification.

Spoofing of ADS-B radio signals As described in the introduction, the basic threat is that of a malicious actor generating a perfectly compatible spoofed ADS-B message and transmitting it using readily available hardware such as an SDR. There already exists readily available open-source software that can be downloaded and easily installed onto an SDR that can receive and display ADS-B information. A competent programmer can easily modify this software to transmit as well. An adequate SDR and antenna for ADS-B frequencies can be acquired for a few hundred dollars. This threat is real. It is specific to ADS-B and no generic IT cybersecurity measures can effectively prevent it.

Throughout the remainder of this chapter, our threat model will focus exclusively on this last type of threat.

3.2.3 Risk to Surveillance

The risk that ADS-B spoofing represents for ATC depends on whether other surveillance technologies are deployed and the method of ADS-B deployment. Three different scenarios must be considered, discussed here from better to worse.

Hybrid surveillance

In regions where other surveillance methods are available, the impact of spoofing can be greatly reduced by employing *data fusion*. For example, where radar is available, contrasting SSR information (azimuth, range, pressure altitude) with ADS-B position reports (latitude/longitude, GPS altitude) can allow the detection of spoofed messages that do not correspond to actual aircraft. Of course, this benefit would be lost if the original NextGen aim of replacing radar stations with ADS-B receivers is maintained. In that case, a similar benefit can be obtained by combining multilateration information and ADS-B. Data fusion does not completely eliminate the risk of spoofing but greatly reduces it, as it forces the attacker to transmit the spoofed signal from locations close to that of the actual aircraft.

Ground-based ADS-B only

Since ADS-B receivers typically use omnidirectional antennas they provide no azimuth information. In terms of detection, the most that can be done is to verify whether the reported position is within the range of the corresponding receiver, that is, within its line of sight (LOS); this assumes, of course, that the detecting algorithm has access to and uses the corresponding receiver information for each position report (its exact location). If that is the case, to defy this countermeasure, all the attacker has to do is place the spoofing transmitter within the range of a receiving station, typically a circular region several kilometers in radius. Alternatively, this radius can be dramatically improved to several tens of kilometers by placing the transmitter on a cheap remote-controlled drone at a height of just a few hundred feet.

Space-based ADS-B only

With its 66 low Earth-orbit satellites, the Iridium constellation provides global coverage, allowing transmitters anywhere on Earth to send signals that will eventually be received by Iridium ground stations and retransmitted to its intended recipient, in our case, the corresponding ANSP [68, 75]. Theoretically, this means that an ADS-B spoofing message indicating the presence of an aircraft in the airspace of a targeted ANSP could be sent from *anywhere on Earth*. This is worse than in the ground-based scenario, in which the spoofing transmitter must be in the same region as the ADS-B receiver.

Detection of this kind of spoofing is difficult because, unlike geostationary satellites, Iridium satellites do not have a fixed position with respect to the ground (they travel at 27 000 km/h) [68, 75]. Nonetheless, the orbit of a particular satellite is known, and its precise

location in time can be inferred, as can the region of Earth that is within the reception range at that time, that is, its *footprint*. Thus, by carefully contrasting the reception time of an ADS-B message at the satellite with the ADS-B-reported position, it is possible to verify whether this position is within the satellite's footprint at that time. This would theoretically allow the satellite operator, or even the ANSP (if it receives the satellite and precise timing information), to filter out spoof messages sent from an unrelated far-away location, since the satellite having received the message would not be the same as those overflying the reported position at that time.

To bypass such detection, the attacker must locate the transmitter within the footprint of such a satellite. Unfortunately, the orbit of Iridium satellites has an altitude of 781 km [68, 75][2], which generates a 4–480 km wide footprint within which to position the transmitter. In the case where more than one satellite receives the signal, this area would be reduced to the intersection of footprints by fusing information from all receiving satellites. However, given the current configuration of the Iridium constellation, multi-satellite reception seems to be the exception rather than the rule, a situation that is further complicated by the fact that Iridium functions as a "cell" network where transmitters are handed off from one satellite to another as they fly by [75].

In summary, the risk of spoofing is much greater in space-based ADS-B applications because 1) it is typically employed in situations where no other surveillance technology is available, and 2) the area within which the spoofing transmitter can be located could be as large as a whole continent.

3.2.4 Risk to Traffic Awareness

In broad terms, the potential impact of ADS-B spoofing in the context of ACAS is to force pilots to make incorrect decisions on collision avoidance, due to unreliable traffic situational awareness. This could lead, at the very least, to unnecessary maneuvers and, in the worst case, become a contributing factor to mid-air collisions. The associated risk is relatively low for commercial aviation (at least for the time being) for two reasons. First, the vast majority of commercial flights are conducted under Instrument Flight Rules (IFR) where traffic separation is the responsibility of ANSP, who implement separation by either using positive surveillance or greater separation standards. Second, because larger business and transport aircraft are required to have TCAS hardware, which does not depend on ADS-B to create traffic situational awareness. However, the risk is non-negligible to Visual Flight Rules (VFR) GA flights in congested areas, where sole reliance on the "see and be seen"

[2]This is the altitude above the Earth's surface.

paradigm has proven over time to provide inadequate prevention of near-miss incidents and mid-air collisions. Therefore, there is an uncomfortable trade-off between the added safety benefit of using ADS-B In as an affordable alternative to TCAS for VFR GA traffic and the security risk of having malicious actors create a confusing and potentially dangerous traffic awareness picture for such pilots.

3.3 Technical and operational criteria for secure ADS-B solutions

As stated in Chapter 1, any secure ADS-B solution must embrace requirements of *safety*, *security* and *rapidity of deployment*. In this section, we review these high-level requirements in the light of the operational and technical reality of the current use of ADS-B and associated security risks and extract detailed criteria that such a solution must meet.

3.3.1 Safety Criteria

The most important requirement is that of safety. Whether we are talking about surveillance or traffic awareness, ADS-B messages are meant to be freely received and interpreted by all concerned. This is indeed mandated in the Minimum Aviation System Performance Standard for ADS-B,

> [the] continuous and common view of the surveillance situation from the perspective of all users greatly enhance coordination, communication, and safety and is a major goal of ADS-B [76]

In other words, all receivers must be capable of reading ADS-B messages, without constraints.

This requirement of *accessibility* of data represents a formidable conundrum given the current situation. On the one hand, one desires to introduce security features that would make it possible to authenticate ADS-B messages, but not in a way that would prevent existing hardware to send and receive critical safety information in the interim period where both security-compliant and legacy systems would have to co-exist. In other words, in the context of the gradual introduction of more secure solutions, the accessibility requirement translates into the necessity for *backward compatibility* criteria for such a solution. This backward compatibility goes both ways:

- Legacy equipment must be able to continue to function as before, which includes both a) the ability to receive unauthenticated ADS-B messages, and b) the ability to correctly receive and parse authenticated ADS-B messages.

- New security-compliant equipment must still be able to see and adequately process unauthenticated ADS-B messages, albeit allowing for filtering or alternate modes of displaying it.

Both of these criteria apply to both ADS-B applications. In surveillance, it is critical that aircraft with security-compliant ADS-B equipment continue to be able to fly into airspace controlled by ANSP using legacy ADS-B equipment. Conversely, ANSP having transitioned to secure ADS-B solutions should continue to be able to see information from aircraft still flying with legacy ADS-B equipment. The same is true for traffic awareness. Air crews in aircraft with legacy ADS-B will still want to see aicraft with security-compliant ADS-B, and vice-versa.

As we will see in Chapter 4, the most viable proposed solutions separate the current (legacy) ADS-B message content from the security information that allows for authentication of the message. For clarity purposes, we will denominate them *Non-Security Information* (NSI) and the *Extra Security Information* (ESI), respectively.

As explained in Chapter 2, the most widely used transmission channel for ADS-B is 1 090 MHz, commonly referred to as 1090ES. This frequency is also used for non-ADS-B ES messages by radar stations, TCAS, and multilateration. The alternate channel is the UAT at 978 MHz, which is used in the United States for TIS-B and FIS-B information and is being considered for similar use in Europe. In both cases, these channels can become significantly congested in dense airspace, introducing additional constraints for ADS-B authentication.

Taking this into consideration, we rephrase the two backward compatibility criteria above more precisely as follows:

BC1 The transmittal of ESI must not impede the *reception* of NSI by legacy equipment.

BC2 The transmittal of ESI must not impede the extraction and interpretation (parsing) of NSI.

BC3 The transmittal of ESI must have minimal impact on channel occupation.

BC4 Secure ADS-B equipment must be able to receive and interpret NSI transmitted by legacy equipment.

Beyond backward compatibility criteria, one must consider that there may be non-malicious circumstances thay may prevent security-compliant ADS-B aircraft to transmit security information, or that may prevent security-compliant ADS-B receivers from correctly receiving

and interpreting this information. In such circumstances, and for the sake of safety, it should be possible for ATC controllers (in surveillance) and air crews (traffic awareness) to view the non security information, if needed/desired.

F1 The non availability of ESI should not prevent Secure ADS-B equipment from receiving and interpreting NSI.

3.3.2 Security Criteria

Beyond detection solutions, the general idea common to all prevention proposals to secure ADS-B involves message authentication schemes. In such schemes, the receiver of a message must be able to verify that

1. the message has not been modified in transit (data integrity), and
2. the message was indeed sent by its advertised originator (originator or source authenticity).

In the context of ADS-B and our threat model, these criteria can be rephrased as follows:

S1 The attacker must not be able to *modify* an authenticated ADS-B message previously sent by an aircraft using security-compliant ADS-B equipment.

S2 The attacker must not be able to *create* a new ADS-B message that appears to be authenticated by an aircraft *already in flight*.

S3 The attacker must not be able to *create* a new ADS-B message that appears to be authenticated by *any* aircraft.

The most typical method for achieving message authentication is for the originator to generate a Message Authentication Code (MAC) based on the message and a shared secret key (HMAC, see Section 4.1.2). Such schemes require that all entities responsible for verifying authenticity also know this shared secret. In ADS-B for surveillance, this could include all ground-based ADS-B equipment operated by ANSP worldwide. For airborne traffic awareness, it may extend to all ADS-B In–equipped aircraft globally. This widespread distribution of the shared key represents a significant and unacceptable security risk. For example, an attacker could compromise a remote ADS-B ground station in the first case or gain control of a registered aircraft in the second case, thereby obtaining the shared secret key. Accordingly, it is judicious to add a third security criterion, which is a consequence of the high-level safety requirement.

S4 Verification of message authenticity must not require the knowledge of any secret information.

This criterion is indeed also a consequence of the application of the *Key Segmentation Principle* in cybersecurity, where the aim is to minimize the impact of a compromise of secret information.

3.3.3 Rapid Deployment Principles

Changes in aviation technology take a long time to be implemented. One reason is the long lifetime of aircraft and aviation equipment. Another is the necessity to ensure that new technologies do not compromise safety. The international nature of aviation requires that solutions be consensually approved before being deployed worldwide. All of these reasons make the process of designing, approving, and deploying new technology long and complex, often taking decades. This is in stark contrast with IT, for example, where a common approach to address security vulnerabilities is to quickly deploy new versions of software that patch them, typically within days or weeks.

Therefore, security solutions that address known vulnerabilities in aviation must be designed from the onset to simplify the process of getting them approved and deploying them. Generically speaking, the risk of protracted approval and deployment can be mitigated by applying the following principles:

RD1 Avoid changes to existing standards. This can be accomplished by deploying backward-compatible solutions and unused features of existing standards,for example.

RD2 Implement changes by updating software and minimizing changes to hardware, whenever possible. This is due to the fact that it is easier and cheaper to update firmware on existing hardware (avionics, ground stations, etc.) than changing the whole component. This principle is actually coded in Radio Technical Commission for Aeronautics (RTCA) standards for ADS-B [12, 76].

RD3 Decouple urgent security-oriented changes from other changes in technology or paradigm shifts. This principle is in fact already applied in aviation with respect to urgent safety-oriented changes, that have their own approval, dissemination and mandatory deployment schedules, e.g. Service Bulletins (SB) and Airworthiness Directives (AD). If security solutions can be decoupled from other slower-advancing technological changes, then they could benefit from a similar approach.

RD4 Utilize existing organizational structures and processes, adapting them minimally if necessary.

In the context of ADS-B these generic principles translate into the backward compatibility criteria described above, which are key to enable rapid deployment of a security solution.

3.4 Summary of the chapter

In this chapter, the security of ADS-B is examined, beginning with the identification of its principal vulnerability: the absence of an authentication mechanism in its design. The discussion then investigates how this weakness can be exploited through various attack scenarios, emphasizing the spoofing of ADS-B signals as the most critical threat. The chapter further analyzes the potential risks such attacks pose to the two primary applications of ADS-B: air traffic surveillance and aircrew situational awareness. Finally, it outlines the fundamental criteria that any proposed security solution must satisfy, including safety, security, and rapid deployability, to ensure practical implementation.

CHAPTER 4 CRYPTOGRAPHIC APPROACHES FOR ADS-B BROADCAST AUTHENTICATION

This chapter presents and analyzes the cryptographic techniques proposed in the existing literature, providing a structured examination of their underlying principles, practical applications in the ADS-B context, and inherent limitations in terms of security, backward compatibility, and communication overhead.

4.1 Symmetric cryptography-based techniques

Symmetric cryptography refers to a cryptographic algorithm that uses the same key for an operation and its complement, for example, encryption and decryption. The key is kept secret and is known as a secret or symmetric key [35, 77]. Studies that use symmetric cryptography to secure ADS-B rely on cryptographic primitives such as format-preserving encryption or message authentication codes.

4.1.1 Format-preserving encryption

Format-preserving encryption (FPE) is a cryptographic technique in which the ciphertext maintains the same format as the plaintext. For example, a 16-digit credit card number remains 16 digits post-encryption. A principal method for implementing the FPE is the format-preserving Feistel-based encryption algorithm, known as FFX, which is constructed using a Feistel network. A Feistel network is a general cipher structure that divides an input into two segments and processes them through multiple rounds. Each round employs a round function, which is a reversible transformation that integrates one segment of data with a subkey. The FFX algorithm extends this structure to preserve the format by using a variable radix and input length. This enables it to operate on strings of any size and character set, such as digits, letters, or alphanumeric values, while maintaining the length and format of the plaintext. It also incorporates a tweak as an auxiliary input to vary the encryption output without requiring a new key [78].

The FFX algorithm has been employed in several studies [79–82] to secure ADS-B messages, as it preserves the original message length, thereby avoiding any increase in communication overhead and complying with the MOPS requirement of preserving the 1090ES bandwidth. However, encrypting the messages compromises backward compatibility since navigation data are no longer available in clear text. To address this limitation, other researchers [83, 84] have

proposed using message authentication codes instead.

4.1.2 Message authentication code

A Message Authentication Code (MAC) functions as a cryptographic checksum or tag designed to ensure both the integrity of a message and the authentication of its originator, a property known as message (or data) origin authentication [34]. A MAC is computed using the secret key K shared between the sender and the receiver. The authentication process based on a MAC proceeds as follows: the sender generates a tag for the message m using Equation 4.1 and transmits the pair (m, $\mathrm{MAC}_K(m)$) to the receiver.

$$\mathrm{MAC}_K(m) = h(m, K) \tag{4.1}$$

Upon receiving (m', $\mathrm{MAC}_K(m)$), the receiver computes a tag from the received message m' using the same key, as shown in Equation 4.2:

$$\mathrm{MAC}_K(m') = h(m', K) \tag{4.2}$$

The receiver then compares the computed tag with the received one, as specified in Equation 4.3:

$$\mathrm{MAC}_K(m) \stackrel{?}{=} \mathrm{MAC}_K(m') \tag{4.3}$$

If the equality in Equation 4.3 holds, the message m' is accepted as authentic. To standardize MAC generation, the United States National Institute of Standards and Technology (NIST) has approved three algorithms: CMAC (SP 800-38B [85]), which is based on block ciphers such as AES; HMAC (SP 800-224 [86]), which uses cryptographic hash functions; and KMAC (SP 800-185 [87]), which is built on the SHA-3 KECCAK function.

In contrast to encryption methods, the MAC technique allows message content to remain accessible to all recipients. Samuelson and Kacem [83,84] employed this approach to verify the authenticity of ADS-B packets. However, Samuelson *et al.* [83] provide only limited technical details about their solution, whereas Kacem *et al.* [84] further developed this concept by examining the practicality of using a 24-bit MAC as a substitute for the CRC, ensuring compliance with the MOPS constraints on packet size. Nonetheless, this method may still present challenges in terms of backward compatibility. Replacing the parity identifier field with a MAC could cause legacy transponders to reject the message by interpreting the CRC as

invalid. Therefore, an alternative placement for the MAC outside the standard ADS-B packet fields should be considered. Ultimately, relying solely on a MAC for ADS-B authentication may be insecure, as the keys (K) used to generate it are symmetric. To meet the MOPS operational requirement that all receivers can access the data, these keys would need to be publicly known. Consequently, any adversary with knowledge of K could forge and broadcast false messages. This limitation underscores the need to integrate asymmetric mechanisms into the authentication and verification processes of ADS-B communications. Accordingly, several authors have proposed using digital signatures — an asymmetric cryptographic primitive — rather than a MAC.

4.2 Asymmetric Cryptography-Based Techniques

Asymmetric cryptography uses a pair of keys, one public and one private, to enable secure communication and data authentication. The private key is kept secret by its owner, whereas the public key is shared with other entities to support secure interactions [37]. For example, the private key can decrypt data encrypted with the public key, or it can sign a message that others can verify using the public key.

In the context of ADS-B, most of the proposed asymmetric solutions emphasize digital signatures to authenticate broadcast messages. The key difference between these solutions lies in the implementation and distribution of public keys, which directly affects the signature generation and verification processes. Some authors have proposed traditional Public Key Infrastructure (PKI) schemes, where public keys are certified by trusted third parties [30,88,89]. Others have explored identity-based cryptography, in which public keys are derived from unique identifiers, eliminating the need for certificates [40, 90–95]. More recent studies have focused on certificateless cryptography [40, 96–98], which combines the advantages of both approaches while avoiding the key escrow problem associated with identity-based schemes.

Each of these techniques is discussed in detail below.

4.2.1 Certificate-based signature

Traditional digital signature schemes such as Rivest–Shamir–Adleman (RSA), Edwards-curve Digital Signature Algorithm (EdDSA), and Elliptic Curve Digital Signature Algorithm (ECDSA), as defined in FIPS 186-5 [3], are widely used to ensure data origin authentication and data integrity. For instance, studies [88,89] use ECDSA signatures to authenticate ADS-B messages. In ECDSA, all cryptographic operations are performed over a specific elliptic curve defined by a set of agreed-upon domain parameters. These parameters ensure interop-

Table 4.1 Domain Parameters of ECDSA

Domain Parameters	Description
q	Field size
FR	Field representation (polynomial or normal basis)
a, b	Coefficients defining the curve equation
G	Base point generating a large prime order subgroup
n	Order of G, where $n \cdot G = O$ and O is the identity element
h	Cofactor, ratio of curve order to n

erability by defining a common mathematical context that enables the signer to generate key pairs and produce signatures, and the verifier to validate them, in a secure and consistent manner. The domain parameters and the detailed steps of the ECDSA signature generation and verification processes are summarized in Table 4.1 and illustrated in Figure 4.1.

Before the signing process begins, each signer generates their key pair by selecting a private key d, a randomly chosen integer within the range $[1, n-1]$. The corresponding public key Q is obtained by multiplying the private key d with the base point G, resulting in $Q = d \cdot G$. The private key must remain confidential, while the public key can be distributed to verifiers.

The ECDSA signing process starts by computing the hash of the message to be signed, $e = H(m)$, where e denotes the integer representation of the entire hash output. The signer then generates a random scalar k and computes the elliptic curve point $k \cdot G = (x_1, y_1)$. Using this point, along with e and the signer's private key d, the signature pair (r, s) is calculated as:

$$r = x_1 \bmod n \quad \text{and} \quad s = k^{-1}(e + r \cdot d) \bmod n \tag{4.4}$$

To verify the signature, the receiver hashes the received message to obtain e, then computes the values u_1 and u_2 as shown in Equation (4.5) and reconstructs the elliptic curve point as shown in Equation (4.6).

$$u_1 = e \cdot s^{-1} \bmod n \quad \text{and} \quad u_2 = r \cdot s^{-1} \bmod n \tag{4.5}$$

$$(x_1', y_1') = u_1 \cdot G + u_2 \cdot Q \tag{4.6}$$

The signature is valid if $r = x_1' \bmod n$, which confirms that it was generated using the sender's private key.

Entities: A – Signer, B – Verifier

System Setup: The domain parameters are (q, FR, a, b, G, n, h)

Key Pair Generation: To generate the public-private key pair (d, Q), A performs the following steps:

- Pick an integer $d \in [1, n-1]$.
- Compute $Q = d \cdot G$.

Sign: To sign a message m, A performs the following steps:

- Compute $e = H(m)$.
- Pick an integer $k \in [1, n-1]$, compute $k^{-1} \bmod n$.
- compute $(x_1, y_1) = k \cdot G$, and set $r = x_1 \bmod n$.
- Compute $s = k^{-1}(e + r \cdot d) \bmod n$.
- If $r = 0$ or $s = 0$, repeat the process with a new random k.
- Output (r, s) as the signature.

Verify: To verify a signature (r, s) on message m, B performs:

- If $r, s \notin [1, n-1]$, reject.
- Compute $e = H(m)$.
- Compute $s^{-1} \bmod n$, $u_1 = e \cdot s^{-1} \bmod n$, and $u_2 = r \cdot s^{-1} \bmod n$.
- Compute $(x_1', y_1') = u_1 \cdot G + u_2 \cdot Q$; if $(x_1', y_1') = \mathcal{O}$, reject.
- If $r = x_1' \bmod n$, accept; otherwise, reject.

Figure 4.1 ECDSA scheme adapted from FIPS 186-5 [3]

ECDSA signature schemes cannot ensure security on their own; they rely on mechanisms that guarantee the authenticity of the public keys used in the verification process. This assurance is typically provided by a PKI, which employs digital certificates to link public keys to specific entities. Each certificate includes the entity's identity, public key, and a digital signature issued by a trusted Certification Authority (CA) [99]. The PKI framework oversees the issuance, validation, and revocation of these certificates, enabling recipients to confirm that a public key belongs genuinely to the claimed sender [3, 38, 100].

In the context of ADS-B, this model requires each aircraft to append a digital signature to its broadcast messages. The receivers authenticate the signed messages using the sender's public key provided in a digital certificate. Trust in the public key is established through the CA hierarchy, where ICAO could serve as the global root authority, delegating trust to each country's National Aviation Authority (NAA).

Although this approach offers a standardized solution for broadcast authentication, its implementation in the ADS-B system presents challenges. These challenges involve managing certificate distribution and revocation within a dynamic, global system as well as minimizing communication overhead to comply with the stringent bandwidth constraints of the 1090ES data link.

To address these issues, several adaptations of traditional PKI-based schemes have been explored. For instance, Feng *et al.* [88] introduced an ECDSA-based scheme using X.509 certificates; however, their solution appears not to specify how certificates are distributed or revoked and may not fully meet ADS-B MOPS regarding communication overhead. To reduce this overhead, Costin *et al.* [30] proposed a lightweight PKI in which digital signatures are split across multiple ADS-B messages, with key distribution handled during routine aircraft maintenance. To tackle the challenge of certificate revocation, Buchholz *et al.* [89] developed a dual-path PKI scheme in which each aircraft carries both a long-term certificate issued by its home country's NAA and a session certificate provided by the local ATC center. While this approach enables global authorization to operate in international airspace and local validation to access a given controlled airspace, it may introduce operational complexity due to the need to manage multiple certificates during flight.

Overall, certificate-based authentication in ADS-B faces two main hurdles: (1) additional communication overhead — ECDSA signatures add 256 bits per message in the study [88] and 320 bits in the study [89], on top of the 120-bit standard — and (2) reliance on a global PKI infrastructure that has not yet been established.

4.2.2 Identity-based signature

Table 4.2 Parameter Set of IBS-1 Scheme from ISO/IEC 14888-3

Public parameters	Description
q	Prime order of the groups G_1 and G_T
G_1	Additive cyclic groups of prime order q
G_T	Multiplicative cyclic group of prime order q
$\hat{e}$	Bilinear pairing, $\hat{e} : G_1 \times G_1 \rightarrow G_T$
n	Bit-length of the message to be signed
P	Generator of the group G_1
R	Public key: $R = sP$ where s is the master private key
H_1	Hash function, $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$
H_2	Hash function, $H_2 : \{0,1\}^* \times G_T \rightarrow \mathbb{Z}_q$

Identity-based cryptography eliminates the key distribution challenge of certificate-based PKI by allowing public keys to be derived from known identifiers such as email addresses [39]. This removes the need for certificates because a trusted Private Key Generator (PKG) computes each user's private key using the identity-derived public key and its master key [39].

Since the initial work [39], several identity-based signature (IBS) schemes have been proposed. Among these, three have been standardized: the IBS-1 scheme [101], included in ISO/IEC 14888-3; the IBS-2 scheme [102], a part of ISO/IEC 14888-3; and the BLMQ scheme [103], which was standardized in IEEE 1363.3 [104] from 2013 until its deactivation by 2024. In this study, we focus on the IBS-1 scheme from ISO/IEC 14888-3, as it remains active. Readers interested in the other active schemes can consult the works of Choon *et al.* [102] and the standard IEEE 1363.3 [104].

The system parameters outlined in Table 4.2 constitute the mathematical basis for the IBS-1 scheme. This scheme involves three entities — the PKG, signer, and verifier — and is organized around four fundamental algorithms: Setup, Extract, Sign, and Verify. Figure 4.2 illustrates the detailed procedure for each phase. In the Setup phase, PKG determines the domain parameters of the system and selects a random integer s to function as the master private key. During the Extract phase, for a signer identified by ID_A, the PKG utilizes this identity along with the master key s to calculate the private key as $d_A = sH_1(ID_A)$. When signing a message, the signer uses its private key d_A and random value k to generate the signature pair (h, S). Finally, the Verify algorithm allows the verifier to authenticate the signature using the system parameters, along with the signer's identity.

Entities: A – Signer, B – Verifier, T – Private Key Generator (PKG)

Setup: The system parameters are $(q, G_1, G_T, \hat{e}, n, P, R, H_1, H_2)$, and the master private key is $s \in_R \mathbb{Z}_q^*$.

Extract: Given ID_A representing A's identity, T computes A's private key as $d_A = sQ_A$, where $Q_A = H_1(ID_A)$.

Sign: To sign a message $m \in \{0,1\}^n$, A performs the following steps:

- Pick $k \in_R \mathbb{Z}_q^*$.
- Compute $U = \hat{e}(sQ_A, P)^k$, $h = H_2(m \,\|\, U)$, and $S = (k - h)d_A$.
- Output (h, S) as the signature.

Verify: To verify the signature (h, S), B performs the following steps:

- Compute $U = \hat{e}(S, P) \cdot \hat{e}(Q_A, R)^h$.
- If $h \equiv H_2(m \,\|\, U)$, accept; otherwise, reject.

Figure 4.2 ISO/IEC 14888-3 IBS-1 Scheme as interpreted in [4]

Incorporating identity-based signatures into ADS-B involves assigning the PKG role to a trusted authority, such as ICAO, a NAA, or an airline. In this model, the aircraft's identity — typically its ICAO address or flight number — serves as the basis for the public key. The PKG then uses this identity-derived public key to generate the corresponding private key and securely provisions it to the aircraft. Each ADS-B message is subsequently transmitted with a digital signature. Although this model eliminates the need for certificates, its practical implementation in an ADS-B environment presents certain challenges, including the need to improve computational efficiency, support large-scale deployments, and minimize communication overhead. These challenges have led to the tailor-made solutions described below [40, 90–95].

Research efforts have primarily focused on optimizing cryptographic operations. For example, Baek *et al.* [90] proposed a two-stage signature scheme (online and offline) to improve the efficiency of signature generation. Yang *et al.* [91] focused on reducing verification delay and computation by introducing batch verification of digital signatures, while a subsequent work [92] added message recovery capabilities to the ID-based broadcast authentication. Recognizing the limitations of a single PKG in large-scale systems, Yang *et al.* [93] adopted a hierarchical ID-based signature (HIBS) scheme with batch verification, though its reliance on hash-to-point operations impacted deployability. To mitigate this, He *et al.* [94] proposed a three-level HIBS using only general hash functions, aiming to simplify operations and reduce computational cost. Further improvement came from the work by Thumbur *et al.* [95], who eliminated bilinear pairings entirely, achieving a modest reduction in overhead.

Despite these advances, these solutions tend to increase communication overhead, which may conflict with ADS-B 1090ES bandwidth constraints, and they remain potentially vulnerable to key escrow, where a compromised PKG could misuse the private keys of the users [40].

Table 4.3 Parameter set of CL-PKS Scheme from the original paper [1]

Public parameters	Description
q	Prime order of the groups G_1 and G_T
G_1	Additive cyclic groups of prime order q
G_T	Multiplicative cyclic group of prime order q
$\hat{e}$	Bilinear pairing, $\hat{e} : G_1 \times G_1 \rightarrow G_T$
n	Bit-length of the message to be signed.
P	Generator of the group G_1
P_0	$P_0 = sP$ where s is the master private key.
H	Hash function, $H : \{0,1\}^* \times G_T \rightarrow \mathbb{Z}_q$

4.2.3 Certificateless signature

Certificateless public key cryptography (CL-PKC) addresses the key escrow issue of identity-based signatures (IBS) by splitting trust between the user and a third party called the Key Generation Center (KGC). In these schemes, users combine their own secret with a partial private key provided by the KGC, eliminating the need for traditional digital certificates.

A CL-PKS scheme is generally defined through seven algorithms: Setup, Partial-Private-Key-Extract, Set-Secret-Value, Set-Private-Key, Set-Public-Key, Sign, and Verify. These algorithms are detailed in Figure 4.3. During the Setup phase, the KGC generates the public system parameters along with a master secret key s; the list of system parameters is presented in Table 4.3.

Players: A – Signer, B – Verifier, T – Key Generation Center (KGC)

System Setup: The system parameters are $(q, G_1, G_2, n, \hat{e}, P, P_0, H)$, and the master secret key is s.

Partial Private Key Extraction: Given ID_A, T computes $Q_A = H_1(ID_A)$ and outputs $D_A = sQ_A$.

Secret Value Generation: A selects $x_A \in_R \mathbb{Z}_q^*$ as their secret value.

Private Key Construction: A computes $S_A = x_A D_A = x_A s Q_A$.

Public Key Construction: A sets $X_A = x_A P$, $Y_A = x_A P_0$, and publishes $P_A = \langle X_A, Y_A \rangle$.

Signing: To sign a message M, A performs the following steps:

- Pick $a \in_R \mathbb{Z}_q^*$.
- Compute $r = \hat{e}(P, P)^a$, $v = H(M, r)$.
- Compute $U = vS_A + aP$.
- Output (U, v) as the signature.

Verification: To verify signature (U, v) on message M using ID_A and $P_A = \langle X_A, Y_A \rangle$, B performs the following steps:

- Verifies $e(X_A, P_0) = e(Y_A, P)$.
- Computes $Q_A = H_1(ID_A)$.
- Computes $r = \hat{e}(U, P) \cdot \hat{e}(Q_A, -Y_A)^v$.
- Accepts if $v = H(M, r)$; otherwise, rejects.

Figure 4.3 CL-PKC Signature Scheme as proposed in [1]

In the Partial-Private-Key-Extract step, for a signer identified by ID_A, the KGC computes the corresponding partial private key $D_A = sH_1(ID_A)$, which is then securely transmitted to the signer. Next, in the Set-Secret-Value phase, the signer randomly selects a secret value x_A. This value is used to construct the full private key as $S_A = x_A D_A$. In parallel, the signer also generates it public key P_A by computing $X_A = x_A P$ and $Y_A = x_A P_0$, and then publishes $P_A = \langle X_A, Y_A \rangle$, completing it key pair setup. The signature generation process relies on the private key S_A and a random value a to produce the signature tuple (U, v). When verifying the signature, the verifier uses the sender's public key P_A and identity ID_A to check the validity of the received message.

Studies [40, 96–98] have proposed certificateless authentication schemes to secure ADS-B. Wu *et al.* [96] introduced a certificateless short signature solution, while Braeken *et al.* [97] proposed a certificateless generalized signcryption scheme that flexibly supports signature generation, encryption, or both, depending on the security requirements of the scenario. Subsequent certificateless-based approaches focused on incorporating optimization techniques aimed at improving the efficiency of the signature generation and verification process. Asari *et al.* [40] proposed signature aggregation to reduce communication overhead by producing shorter signature chains. Subramani *et al.* [98] introduced batch verification to decrease computational overhead by enabling faster verification of multiple signatures.

The concept of certificateless short signatures is relatively new and, although it shows promising potential, it is not yet mature enough for widespread adoption. Certificateless cryptography faces two major security challenges. The first challenge is the key replacement attack. In CL-PKC, public keys are not certified by a trusted authority, which makes it possible for an attacker to replace a legitimate user's public key with a forged one. If this attack succeeds, users may unknowingly rely on counterfeit public keys, allowing attackers to impersonate others or decrypt sensitive communications [42].

The second challenge is the risk of a malicious KGC. Although CL-PKC removes the full key escrow problem found in identity-based cryptography, the KGC still generates a partial private key for each user. In 2007, Au *et al.* [41] introduced the notion of a malicious KGC and emphasized that if the KGC is untrustworthy, it can deliberately embed trapdoors during the master key generation process. These trapdoors are hidden vulnerabilities that can later be exploited by the KGC to break the system's security, for example, by forging signatures or decrypting messages without users' knowledge. This risk was overlooked in earlier certificateless schemes, some of which may be vulnerable to such attacks. Despite ongoing research, developing certificateless cryptographic systems that can fully resist both key replacement attacks and malicious KGC threats remains an open research problem [41, 42].

4.3 Hybrid cryptography-based techniques

Upon evaluating the strengths and weaknesses of both symmetric and asymmetric cryptographic techniques, it becomes clear that neither approach alone can fully satisfy the security and operational needs of ADS-B. Symmetric solutions, such as MAC, are bandwidth-efficient but they are not well suited to the open environments in which ADS-B operates. On the other hand, asymmetric techniques, such as digital signatures, provide robust security but result in excessive communication overhead. To overcome this trade-off, researchers have investigated hybrid cryptographic techniques, such as the TESLA protocol [43, 105], which aims to integrate the strengths of both methods.

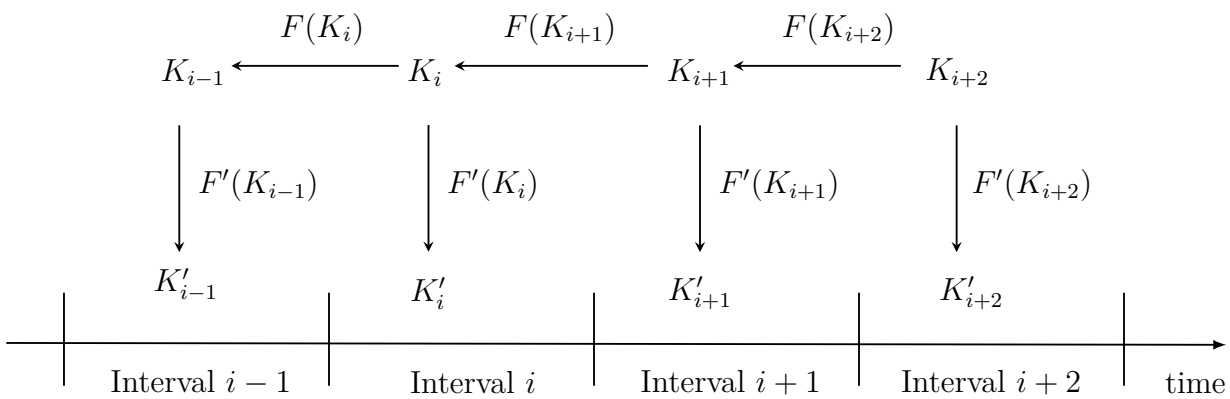


Figure 4.4 Process for generating the one-way keychain and the authentication keys for each time interval, as depicted and explained in the original TESLA paper [105]. The broadcast time is partitioned into N+1 intervals. A keychain is created by iteratively applying a one-way function F, to the preceding key in the chain. Subsequently, another function F' is applied to the elements within the keychain to derive the authentication key for each time interval.

4.3.1 TESLA

The TESLA protocol [43, 105] integrates asymmetric and symmetric cryptography to enable lightweight broadcast authentication. It appends a MAC to each transmitted message, using symmetric keys that are initially secret and disclosed after a fixed delay. Upon reception, receivers temporarily store the messages along with their MAC and can only verify their authenticity once the corresponding keys are disclosed.

TESLA relies on pre-generated keychains and synchronized timing between the sender and the receiver. As illustrated in Figure 4.4, the sender divides the broadcast timeline into N intervals and generates a one-way keychain $\{K_0, K_1, \ldots, K_N\}$ such that each key satisfies $K_0 = F^i(K_i)$, where K_0 serves as the keychain commitment. This structure ensures that

future keys cannot be derived from previously disclosed ones. Each key K_i is then processed through a second one-way function F' to obtain the authentication key $K'_i = F'(K_i)$ used to compute the MAC.

TESLA operates as follows. Before the transmission begins, the sender securely shares the keychain commitment K_0 and the key disclosure delay d with the receiver. To broadcast a message m_j during interval i, the sender constructs the TESLA packet:

$$P_j = m_j \parallel \text{MAC}(K'_i, m_j) \parallel K_{i-d} \tag{4.7}$$

where $\text{MAC}(K'_i, m_j)$ is the authentication tag and K_{i-d} is the disclosed key from a previous interval.

Upon reception, the receiver stores the packet until the disclosed key can be verified by checking whether $K_0 = F^v(K_{i-d})$ for some integer v. If the key is valid, the receiver computes $K'_i = F'(K_{i-d})$ and authenticates the message by verifying its MAC.

TESLA's key advantage lies in enabling short symmetric tags while protecting against real-time forgery through delayed key disclosure. However, its deployment in ADS-B faces several challenges, including the stringent bandwidth constraints of the 1090ES datalink, the need for precise time synchronization, and the inherent authentication delay, which may not fully satisfy the real-time operational requirements of air traffic surveillance.

TESLA-based solutions

To address these challenges, several TESLA-based adaptations have been proposed. Berthier *et al.* [106] modified TESLA to better fit ADS-B by replacing the standard synchronization mechanism with GPS-based timekeeping and introducing a certificate-based PKI to authenticate aircraft and their messages. This scheme employs three types of packets: Type A packets (136 bits) carrying authenticated ADS-B messages with a truncated 16-bit MAC, Type B packets (184 bits) transmitting TESLA keys every 5 seconds, and Type C packets (1520 bits) distributing aircraft certificates every 30 seconds. This design increases bandwidth usage by approximately 35%. Furthermore, while truncating the MAC is a standard and accepted security practice, the choice of a 16-bit MAC in this scheme [106] may slightly increase the probability of MAC guessing and the potential for flooding attacks. Nevertheless, this design represents an important step toward adapting TESLA for the unique constraints of ADS-B.

Sciancalepore *et al.* [107] adopt a different strategy, using TESLA for batch authentication rather than authenticating each message individually. This approach replaces MAC-based verification with digital signatures computed over sets of messages, which significantly reduces

bandwidth consumption. While this solution is efficient, it is inherently sensitive to packet loss and message injection attacks. If a single packet in the authenticated batch is missing or altered, the entire set may become unverifiable, which could pose operational challenges.
Finally, Yang *et al.* [108] propose a hybrid approach that combines TESLA with format-preserving, Feistel-based encryption to provide both confidentiality and integrity for ADS-B messages. This method requires the transmission of five ADS-B messages per navigation data update, resulting in increased bandwidth usage. Moreover, encrypting the ICAO code may affect backward compatibility, potentially limiting the practical deployment of this scheme in operational settings.
Although the reviewed TESLA-based adaptations partially mitigate the challenges of securing ADS-B, the inherent trade-offs between bandwidth efficiency and security remain. Interestingly, the recent introduction of phase overlay capacity in the MOPS provides a potential pathway for future solutions that could reconcile these constraints.

4.4 Positioning of our approach

The preceding sections examined both traditional and emerging cryptographic techniques for authenticating ADS-B broadcasts. Below is a summary of the main strengths and weaknesses identified in previous studies, which serve as the basis for positioning our approach.
Symmetric cryptography methods, such as Message Authentication Codes, offer low communication overhead but are ill-suited to the open nature of ADS-B broadcast communications. As the same symmetric key must be shared among all parties, any adversary with access to that key can forge and transmit malicious messages.
To overcome the limitations of symmetric cryptography, asymmetric techniques have been introduced, including certificate-based, identity-based, and certificateless digital signatures. However, these approaches present several challenges. Certificate-based methods require the management of digital certificates and incur significant communication overheads. Identity-based and certificateless schemes eliminate the need for certificates but introduce their own risks: identity-based schemes are vulnerable to key escrow, whereas certificateless schemes can be compromised by malicious Key Generation Centers.
Hybrid schemes, particularly those based on the TESLA protocol, partially balance security and bandwidth efficiency by combining the advantages of symmetric and asymmetric cryptography. Although promising, these adaptations still face challenges related to this trade-off.
Building on these observations, our work in Chapter 6 addresses the remaining gap by introducing the Compatible Authenticated Bandwidth-efficient Broadcast (CABBA) protocol for ADS-B [55]. CABBA integrates the TESLA authentication mechanism [43, 105] with

a PKI and phase overlay modulation in the physical layer of ADS-B. This integration simultaneously meets security objectives — data integrity, origin authentication, and identity authentication — while preserving the bandwidth of the 1090ES communication channel and maintaining backward compatibility. As summarized in Table 4.4, CABBA effectively positions itself among existing approaches by achieving a balance between security and safety, thus filling the gaps left by previous symmetric, asymmetric and hybrid schemes.

Table 4.4 Positioning our ADS-B broadcast authentication methodology with respect to existing literature

	Cryptographic primitive	Security goals			Operational performances	
		Origin Auth	Integrity	Entity Auth	Backward compatibility	Bandwidth preservation
Symmetric	Encryption [79–82, 109]	✗	✗	✗	✗	✓
	MAC [83, 84]	✗	✗	✗	✓	✗
Asymmetric	Digital signature using certificate-based PKI [30, 88, 89]	✓	✓	✓	✓	✗
	Digital signature using identity-based PKI [90–95]	✗	✗	✗	✓	✗
	Digital signature using certificateless PKI [40, 96–98]	✓	✓	✗	✓	✗
Hybrid	Encryption using TESLA with certificate-based PKI [108]	✓	✓	✓	✗	✗
	MAC or Digital signature using TESLA with certificate-based PKI [106, 110]	✓	✓	✓	✓	✗
	CABBA solution [55] MAC using TESLA with certificate-based PKI and phase overlay techniques	✓	✓	✓	✓	✓

4.5 Summary of the chapter

This chapter explores the existing methods for authenticating ADS-B broadcasts, focusing on their strengths and limitations. Our contribution, the cryptographic solution CABBA [55], is then positioned within prior research, showing how it addresses the gaps and challenges identified in earlier studies.

CHAPTER 5 MACHINE LEARNING TECHNIQUES FOR ADS-B DATA INTEGRITY VERIFICATION

Although cryptographic approaches directly address the core vulnerability of ADS-B, the absence of native authentication, they do not offer complete protection. In particular, these methods can fail if key material is compromised or misused. To ensure defense-in-depth, additional mechanisms are required on the receiving end to verify the consistency of incoming data. These detection controls act as a complementary layer, helping to mitigate the residual risks associated with cryptographic defenses. Accordingly, this chapter explores machine learning-based techniques designed to verify the integrity of ADS-B messages. Other techniques for verifying message integrity have been explored in the literature. We define them in Appendix A, outline their strengths and limitations, and justify the choice of machine learning-based methods over these approaches.

5.1 Deep Learning Architectures for Intrusion Detection

Deep learning is a subfield of machine learning that uses artificial neural networks (NN) to extract meaningful patterns from data through a series of nonlinear transformations [111]. One of the most basic forms of NN is the Multilayer Perceptron (MLP), which consists of an input layer, one or more hidden layers, and an output layer. When an MLP contains several hidden layers, it is referred to as a deep neural network (DNN). Each unit, or "neuron", in these layers computes a weighted sum of its inputs, applies an activation function, and passes the result to the next layer. The model learns by minimizing the error between its predictions and the expected outputs through backpropagation, which adjusts the weights in the network to improve performance [111, 112].

5.1.1 Autoencoders

Autoencoders (AE) are a type of neural network specifically designed to learn compact representations of input data, often for purposes such as dimensionality reduction or anomaly detection. They comprise two primary components: an encoder that compresses the input into a latent representation and a decoder that reconstructs the original input from this compressed form. The network is trained to minimize the reconstruction error, which is the difference between the input and reconstructed output. By learning to represent the normal patterns present in the training data, AE typically produces higher reconstruction errors for inputs that deviate from the training distribution [113, 114]. This characteristic is a key reason why AE are frequently used to implement intrusion detection systems.

5.1.2 Variational autoencoders

Variational Autoencoders (VAE) [115] are a probabilistic extension of traditional autoencoders that aim to reconstruct input data and learn the underlying distribution that generates it. Instead of mapping inputs to fixed latent vectors, the VAE encodes them into a probability distribution — typically a Gaussian — parameterized by a mean vector μ and a standard deviation vector σ. Given an input $\mathbf{x}$, the encoder network outputs the two vectors. Then, a latent representation $\mathbf{z}$ is generated using the reparameterization trick: $\mathbf{z} = \mu + \sigma \odot \epsilon$, where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The decoder then uses the sampled $\mathbf{z}$ to reconstruct the original input. Training a VAE involves minimizing a loss function that combines two terms: (1) the reconstruction loss, which measures how close the output is to the original input, and (2) a regularization term (the Kullback–Leibler divergence), which ensures that the learned latent distribution remains close to a standard normal distribution. This structure helps the VAE generalize well, produce meaningful latent representations, and detect anomalies by measuring how well new data fit the learned distribution.

5.1.3 Recurrent Neural Networks

Autoencoders are commonly implemented using MLP, the basic architecture of neural networks. However, when the data have a sequential nature, as is the case with ADS-B time series, Recurrent Neural Networks (RNN) [116] are often preferred to enhance the anomaly detection performance.

RNN are a class of neural networks specifically designed to process sequential data. Unlike MLP, which assume that inputs are independent, RNN incorporate temporal context by maintaining a hidden state that is updated at each time step based on the current input and previous hidden state. This architecture allows the network to model dependencies over time, making it suitable for applications involving time series, speech, or text. However, standard RNN often struggle to learn long-term dependencies because of issues such as vanishing or exploding gradients during training. To address these limitations, Long Short-Term Memory (LSTM) networks [49] were introduced as an enhanced type of RNN.

5.1.4 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks [49] were introduced as a specialized form of RNN capable of learning long-term dependencies in sequential data. LSTM use memory cells to retain information over time and include three gates — input, forget, and output — that control the flow of information into, within, and out of each cell. These gates are governed by learned weights and nonlinear activation functions, typically the sigmoid (σ) and hyperbolic tangent (tanh), to determine what information should be remembered or discarded at each

time step. The computations within an LSTM cell at time step t are as follows:

$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) && \text{(forget gate)} \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) && \text{(input gate)} \\
\tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c) && \text{(candidate cell state)} \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t && \text{(new cell state)} \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) && \text{(output gate)} \\
h_t &= o_t \odot \tanh(c_t) && \text{(hidden state)}
\end{aligned}
\tag{5.1}
$$

Here, x_t is the input vector, h_{t-1} the previous hidden state, and c_{t-1} the previous cell state. The weights W, U, and biases b are learned during training. This gated mechanism allows LSTMs to effectively capture both short- and long-term patterns, making them particularly well-suited for tasks like anomaly detection in sequential data such as ADS-B. The structure of an LSTM cell is illustrated in Figure 5.1.

5.1.5 Convolutional LSTM

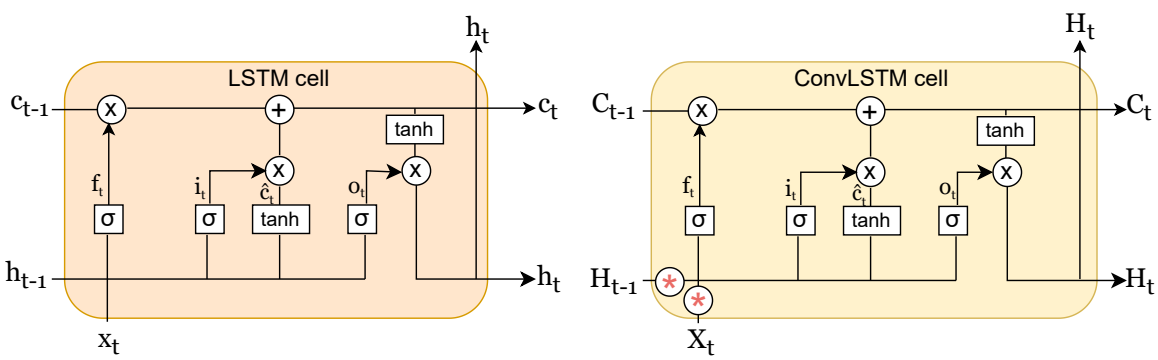


Figure 5.1 Comparison between LSTM and ConvLSTM cell structures.

Convolutional LSTM (ConvLSTM) networks [117] are a variant of the standard LSTM, specifically designed to model spatio-temporal data, such as sequences of 2D spatial grids over time (e.g., precipitation maps). While traditional LSTM use fully connected layers to process vector inputs, ConvLSTM replace these operations with convolutional layers, preserving the spatial structure of the input data. This design makes the ConvLSTM particularly suitable for tasks involving both spatial and temporal correlations, such as weather prediction.
In ConvLSTM, the input X_t, hidden state H_{t-1}, and cell state C_{t-1} are all 3D tensors. The core computations at time step t are as follows:

$$
\begin{aligned}
f_t &= \sigma(W_f * X_t + U_f * H_{t-1} + b_f) && \text{(forget gate)} \\
i_t &= \sigma(W_i * X_t + U_i * H_{t-1} + b_i) && \text{(input gate)} \\
\tilde{C}_t &= \tanh(W_c * X_t + U_c * H_{t-1} + b_c) && \text{(candidate cell state)} \\
C_t &= f_t \odot C_{t-1} + i_t \odot \tilde{C}_t && \text{(new cell state)} \\
o_t &= \sigma(W_o * X_t + U_o * H_{t-1} + b_o) && \text{(output gate)} \\
H_t &= o_t \odot \tanh(C_t) && \text{(new hidden state)}
\end{aligned}
\tag{5.2}
$$

Here, $*$ denotes the convolution operator, $\odot$ the Hadamard (element-wise) product, and σ the sigmoid activation function. The convolutional structure allows ConvLSTM to capture both temporal dynamics and spatial patterns in spatio-temporal data.

5.1.6 Transformer

Unlike RNN [116], LSTM [49], or ConvLSTM [117], the Transformer [5] handles sequential data without relying on recurrence. It uses *self-attention mechanisms* to process entire sequences in parallel and efficiently capture long-range dependencies, making it highly scalable. At the core of the Transformer is the *scaled dot-product attention* mechanism, which operates on three matrices — queries (Q), keys (K), and values (V) — derived from the input embeddings:

$$
\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{5.3}
$$

Here, d_k denotes the dimensionality of the keys. The attention mechanism assigns dynamic relevance scores to tokens in the sequence, enabling each token to selectively focus on the most informative parts of the input regardless of its position. This enhances the model's ability to capture the contextual relationships and long-range dependencies.

To increase its representational capacity, the Transformer uses *multi-head attention*, where several attention heads run in parallel. Each head learns independent projections of Q, K, and V, enabling the model to capture diverse types of dependencies within the same sequence.

$$
\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{5.4}
$$

Each head is computed as $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$, with learned projection matrices W_i^Q, W_i^K, W_i^V, and W^O representing the output projection matrix used to combine the results of all heads into a single vector.

As illustrated in Figure 5.2, the architecture consists of an *encoder* and a *decoder*, each made up of stacked layers. Encoder layers contain a multi-head self-attention sublayer and a feed-

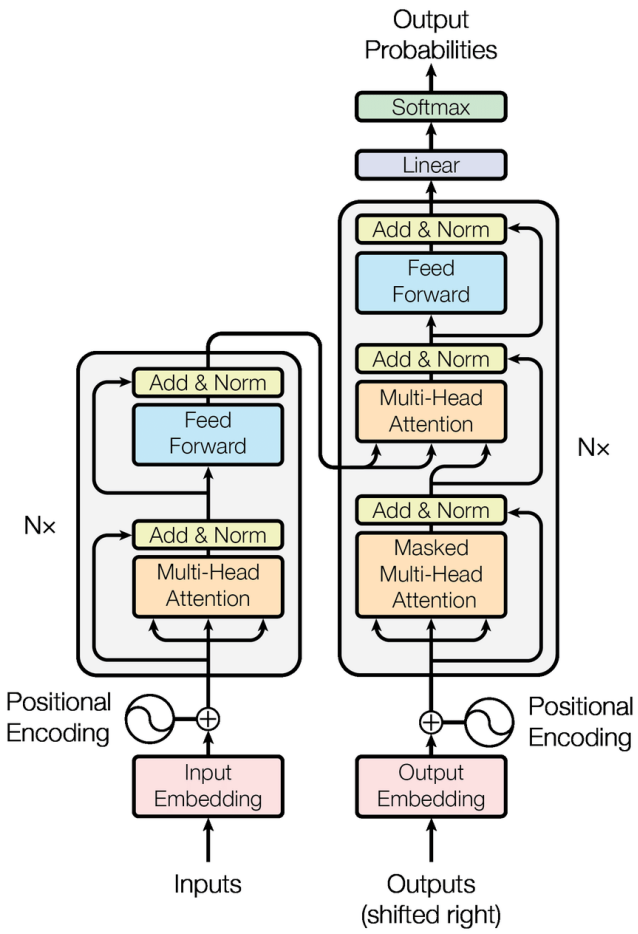


Figure 5.2 Transformer architecture, reproduced from [5]

forward network, both followed by residual connections and layer normalization. Decoder layers add an encoder-decoder attention sublayer, enabling the decoder to incorporate encoder outputs during generation.

Because the Transformer lacks recurrence, it uses *positional encodings* to capture the token order based on sinusoidal functions:

$$\mathrm{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right), \quad \mathrm{PE}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{5.5}$$

where pos is the token position and i is the dimension index. These encodings allow the model to learn position-aware representations.

5.2 Deep Learning Applications in ADS-B Intrusion Detection

Recent advancements in deep learning architectures have enabled the development of various models for detecting anomalies and intrusions in ADS-B data. Many of these models employ

reconstruction-based methods, a class of anomaly-detection techniques in which a model is trained to recreate its input data. During training, the model learns the normal patterns and structure of the data, allowing it to accurately reconstruct similar inputs. Inputs that deviate from these learned patterns are poorly reconstructed, and the resulting differences, known as reconstruction errors, are used to identify anomalies [47, 48, 118]. The following overview presents key representative studies on ADS-B intrusion detection systems in chronological order, highlighting the increasing complexity of these architectures.

Table 5.1 Works on deep learning-based ADS-B intrusion detection, organized chronologically. The last paper [56] is our approach explained in Chapter 6

Paper	**Architecture**	**Technique**
[74]	LSTM-based AE	Sequential reconstruction of individual flight trajectories; anomaly detection via reconstruction error.
[119]	ConvLSTM-based AE	Image-based reconstruction of aggregated airspace traffic; anomaly detection via reconstruction error.
[120]	Clustering + AE	Cluster Mode-S trajectories into major traffic flows; Train the AE on clustered trajectories with regularization; Anomaly detection via reconstruction error.
[121]	Non-recurrent AE	Feature-based reconstruction from statistical descriptors; Anomaly detection via reconstruction error.
[122]	VAE + SVDD	Probabilistic reconstruction; Anomaly detection via hypersphere on reconstruction error.
[50]	LSTM-based contextual AE (CAE)	Flight-phase conditioned reconstruction; Anomaly detection via 3-sigma threshold on reconstruction error.
[51]	TCN + Transformer + SVDD (TTSAD)	Short-term prediction + full-sequence reconstruction; Anomaly detection via SVDD threshold on reconstruction error.
[56]	xLSTM	Transfer learning: pre-training and fine-tuning; Anomaly detection via classification.

Early work by Habler and Shabtai [74] introduced an LSTM-based autoencoder trained on sequences of ADS-B messages corresponding to a specific flight from takeoff to its landing. The model learns a flight route by encoding sequences of ADS-B messages into a fixed-size latent vector, which the decoder then uses to reconstruct the original input data. During testing, anomalies were flagged when the reconstructed output deviated significantly from the actual data. The approach was evaluated separately on different flight routes using

datasets containing artificially injected anomalies and was shown to outperform traditional baseline methods, including the One-Class Support Vector Machine (SVM) [123] and Isolation Forest [124], in detecting anomalous sequences. However, the proposed model [74] evaluates each aircraft's trajectory in isolation, overlooking the spatio-temporal dependencies among multiple aircraft sharing the same airspace [56]. Consequently, its ability to identify anomalies may be hindered by a lack of situational awareness, a limitation that subsequent research has sought to address.

In a later study, Akerman and Shabtai [119] analyzed air traffic at a collective level by aggregating ADS-B messages from all aircraft operating within a defined geographical area. The messages were transformed into image sequences that represented the spatial distribution of the aircraft over time. Each sequence was then processed using a Convolutional LSTM (ConvLSTM) encoder-decoder network designed to capture both spatial and temporal correlations in air traffic patterns. The encoder extracts high-level features and predicts subsequent frames, while the decoder reconstructs the corresponding input images. Anomalies were identified when the reconstructed frames exhibited significant deviations from the original frames. To enhance interpretation, the model integrates an explainability module that generates heatmaps, highlights areas associated with potential anomalies, and aids pilots in making informed decisions.

Olive and Basora [120] proposed another approach that combined trajectory clustering with autoencoder-based reconstruction to detect anomalies in major air traffic flows. Initially, Mode S trajectories were grouped into clusters representing dominant air routes through the airspace, with each cluster containing multiple flights following similar paths. An autoencoder was then trained for each cluster to model the typical flight patterns within that group. To enhance robustness, a custom regularization term was incorporated into the autoencoder's loss function, mitigating overfitting in clusters with limited samples and improving generalization across the sector. For each trajectory, the model produced a reconstruction error score, which served as an indicator of potential anomalies and aided in the identification of unusual events in air traffic operations. Although effective, this approach depends on the availability of sufficient historical data for each cluster, which may not always exist for infrequently used routes or for training flights.

To address the reliance on extensive historical data required by location-specific approaches, such as those of Akerman and Shabtai [119] and Olive and Basora [120], Fried and Shabtai [121] proposed methods capable of generalizing across flights with limited prior records. This study applied a non-recurrent autoencoder to ADS-B data, following a series of preprocessing steps designed to extract meaningful features. First, the geodetic coordinates were converted into a three-dimensional Cartesian system. Next, K-lag and K-order dif-

ferencing removed trends and seasonal patterns, and statistical descriptors, including the mean, variance, maximum, minimum, median, and standard deviation, were calculated for each temporal window. These descriptors, which capture flight dynamics without relying on sequential modeling, were used as inputs to the non-recurrent autoencoder for anomaly detection. For comparison, a recurrent LSTM-based autoencoder was also evaluated, directly processing the sequences after the pre-processing step that removes seasonality and trends, thereby learning temporal dependencies.

Luo *et al.* [122] later combined a Variational Autoencoder (VAE) with a Support Vector Data Description (SVDD) model. The VAE first reconstructs the input messages, and the differences between the reconstructed and actual values are used to define the SVDD hypersphere, which encloses normal data in the feature space. During testing, any message whose reconstruction error falls outside the hypersphere is flagged as anomalous. The results indicate that by leveraging both the temporal patterns captured by the VAE and the distributional boundaries defined by SVDD, this hybrid method enhances anomaly detection in flight trajectories, particularly for identifying irregularities, such as random or constant positional deviations. Furthermore, the approach demonstrated superior performance compared to conventional machine learning techniques.

Building upon the LSTM autoencoder design, Chevrot *et al.* [50] introduced a Contextual Autoencoder (CAE), which incorporates flight-phase awareness to enhance anomaly detection in ADS-B data. A shared bidirectional LSTM encoder extracts temporal dependencies from short time windows, whereas multiple decoders, each dedicated to a specific flight phase, such as climbing, cruising, or descending, reconstruct the corresponding segment. During training, the model learns normal flight patterns and computes reconstruction errors for each time window. The mean and standard deviation of these errors are then used to define a threshold based on the 3-sigma rule, so that windows with errors exceeding this threshold are considered anomalous. By conditioning reconstruction on flight context, the CAE increases sensitivity to phase-specific irregularities.

Finally, Luo *et al.* [51] proposed the TTSAD framework, which combines Temporal Convolutional Networks (TCN), Transformers, and Support Vector Data Description (SVDD) to detect anomalies in ADS-B sequences.. The transformer module then reconstructs the entire sequence, including the predicted values. Finally, the SVDD module evaluates the reconstruction error, which is the difference between the reconstructed sequence from the transformer and the actual observed dataand defines a threshold based on the distribution of normal errors. Sequences exceeding this threshold were flagged as anomalous sequences. This hybrid architecture slightly improves the detection of sophisticated gradual attacks compared to the study in [50].

This improvement reflects a broader trend observed in the literature, where F1-scores for detecting subtle and gradual attacks on ADS-B data progressively increased across studies. For example, in their comparative analysis, the authors of [50] reported the following F1-scores for the detection of velocity drift attacks, a kind of gradual attack: 0,886 using an LSTM-AE, 0,926 using the VAE-SVDD, and 0,939 for CAE architectures. In the same detection context, the TTSAD framework [51] achieved an F1-score of 0,94. These results highlight two key observations: (1) context-aware architectures, such as CAE [50] and transformer-based models [5, 51], currently demonstrate the highest effectiveness in identifying gradual and subtle ADS-B attacks; and (2) despite these improvements, the detection accuracy of current state-of-the-art models [50, 51] remains insufficient for safety-critical systems such as ADS-B [56].

5.3 Emerging Approaches in Deep Learning-Based Intrusion Detection

Transformers have shown promise in identifying subtle gradual attacks on ADS-B data, as evidenced by the performance of the TTSAD [51]. However, their quadratic computational complexity with respect to the sequence length poses scalability challenges. This has led to growing interest in alternatives such as the extended LSTM (xLSTM) [6], which builds upon traditional LSTMs [49] while potentially overcoming some of the limitations associated with Transformers.

5.3.1 Extended LSTM

The extended Long Short-Term Memory (xLSTM) [6] network enhances the classical LSTM [49] through two key innovations: exponential gating and redesigned memory structures. As shown in Figure 5.3, the exponential gating mechanism replaces the conventional sigmoid activation, improving gradient flow and training stability. In addition, xLSTM introduces two memory variants, the scalar LSTM (sLSTM) and matrix LSTM (mLSTM), which expand the representational capacity of the model. While the sLSTM retains scalar memory states with an optimized update scheme, the mLSTM generalizes these states to matrices, enabling richer internal dynamics and more efficient parallel computation. Both variants share the exponential gating mechanism but differ in the dimensionality and structure of their internal operations. Their update equations at time step t are as follows:

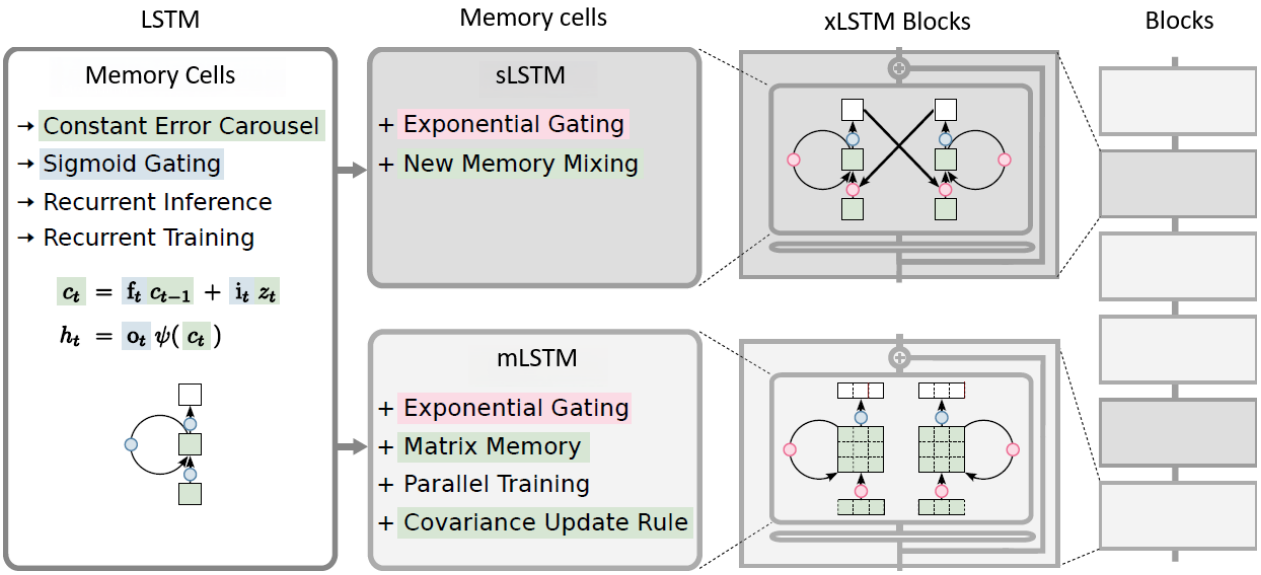


Figure 5.3 Comparison between LSTM and xLSTM memory cells, reproduced from [6].

Scalar LSTM (sLSTM):

$$f_t = \exp(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \exp(W_i x_t + U_i h_{t-1} + b_i)$$
$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$
$$o_t = \exp(W_o x_t + U_o h_{t-1} + b_o)$$
$$h_t = o_t \odot \tanh(c_t)$$

Matrix LSTM (mLSTM):

$$F_t = \exp(W_F x_t + U_F H_{t-1} + B_F)$$
$$I_t = \exp(W_I x_t + U_I H_{t-1} + B_I)$$
$$\tilde{C}_t = \tanh(W_C x_t + U_C H_{t-1} + B_C)$$
$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t$$
$$O_t = \exp(W_O x_t + U_O H_{t-1} + B_O)$$
$$H_t = O_t \odot \tanh(C_t)$$

Figure 5.4 Comparison of update equations in scalar LSTM (left) and matrix LSTM (right) formulations within xLSTM.

Here, the sLSTM operates on vector-valued hidden and cell states (h_t, c_t), while the mLSTM generalizes these to matrices (H_t, C_t), allowing for parallel processing and improved modeling of feature interactions. The exponential activation function $\exp(\cdot)$, used across all gates, provides more expressive gating dynamics compared to the classical sigmoid. These advanced memory mechanisms are organized into residual xLSTM blocks, which are stacked to form deep architectures that outperform standard LSTMs in modeling long-range dependencies and complex sequential data.

5.3.2 Transfer learning

The performance of intrusion detection systems depends not only on selecting an appropriate model architecture but also on the model's ability to learn and generalize normal system behavior. Transfer learning has emerged as a promising approach in this regard [125], as it enables a model to first capture fundamental patterns of normal behavior and then be fine-tuned for anomaly detection tasks [125,126]. By reusing these pretrained representations, IDS models can generalize more effectively and improve their ability to detect novel or previously unseen attacks [112].

Reconstruction-based methods, which remain predominant in the current state of the art, face two major limitations. First, as shown in the study [127], their underlying assumption can fail. Indeed, autoencoders may reconstruct anomalous inputs as accurately as normal data, leading to missed detections. This happens through three mechanisms in the network's latent space: interpolation, where an anomalous input falls between normal data points and the decoder reconstructs it by using patterns from nearby normal points; extrapolation, where an input lies outside the range of normal data but the decoder still reconstructs it in a way that looks normal based on the patterns it learned from normal data; and boundary effects, where anomalies near the edges of normal data are reconstructed as if they were normal because the network cannot precisely distinguish the boundary of normal patterns. This issue is exacerbated when the attacker's strategy is to make subtle modifications to real data, allowing anomalies to remain undetected.

Second, reconstruction-based methods rely on a static notion of normality, making them susceptible to concept drift and high false-alarm rates when legitimate behavior evolves [128], as in ADS-B, where flight dynamics naturally vary across flight phases.

An alternative to reconstruction-based anomaly detection is offered by deep learning–based classification techniques, where models are directly trained to discriminate between normal and attack samples. However, these approaches face two main challenges: they require large, well-labeled datasets and, in practice, labeled attack data are often scarce.

In contrast, transfer learning offers several advantages over both reconstruction-based and classification approaches:

1. **Improved generalization**: the model can detect previously unseen attacks by learning general patterns of aircraft behavior rather than memorizing specific datasets.

2. **Faster training and adaptation**: fine-tuning a pretrained model is substantially faster than training one from scratch, and the IDS can be updated incrementally as new attack data become available, without retraining the entire system.

3. **Data efficiency**: fewer labeled attack samples are needed since the model already understands the temporal and multivariate structure of ADS-B data.

Given these advantages, we believe that transfer learning is well-suited for ADS-B IDS, where gradual, subtle attacks and evolving flight patterns challenge traditional methods.

5.4 Positioning of our approach

Most previous studies on ADS-B IDS used LSTM-based autoencoders, which suffer from limited memory and thus struggle to capture the long-term dependencies needed to detect subtle gradual attacks. Later approaches employed context-aware architectures such as Contextual Autoencoders [50] and Transformers [51]. Although these models slightly improve detection, the gains remain insufficient for safety-critical applications.
We propose a novel xLSTM-based architecture that provides augmented memory compared to the standard LSTM and is more computationally efficient than transformers. xLSTM is faster because its computational complexity scales linearly with sequence length, whereas the complexity of transformers scales quadratically [52].
Furthermore, instead of relying on reconstruction-based training, as in previous studies (see Table 5.1), we applied transfer learning: pre-training the xLSTM on normal ADS-B sequences allowed it to learn general temporal patterns, enabling faster adaptation, better generalization to unseen attacks, and reduced dependence on large labeled attack datasets.
Our methodology consists of three main steps:

1. **Pre-training**: The model learns the temporal dynamics of normal ADS-B communications from a large dataset.

2. **Fine-tuning**: The pretrained model is fine-tuned to identify specific types of gradual attacks on ADS-B data, resulting in four specialized models corresponding to different attack types.

3. **Ensembling**: The fine-tuned classifiers are combined into a unified IDS capable of detecting multiple intrusion types.

5.5 Summary of the chapter

This chapter reviews deep learning–based techniques, including architectures and training strategies, proposed for verifying the integrity of ADS-B messages, highlighting their strengths and limitations. Our contribution, an xLSTM-based IDS [56] trained using transfer learning, is then positioned within this prior research, demonstrating how it addresses the challenges faced by previous IDS.

CHAPTER 6 ARTICLE 1 - CABBA: COMPATIBLE AUTHENTICATED BANDWIDTH-EFFICIENT BROADCAST PROTOCOL FOR ADS-B

Mikaëla Ngamboé[a], Xiao Niu[a], Benoit Joly[b], Steven P. Biegler[b], Paul Berthier[d], Rémi Benito[c], Greg Rice[b], José M. Fernandez[a], Gabriela Nicolescu[a]

[a] Department of Computer and Software Engineering, Polytechnique Montréal Technological University, Montréal, Québec, Canada

[b] Collins Aerospace, Cedar Rapids, Iowa, United States

[c] Bombardier, Montréal, Québec, Canada

[d] Rhea Group, Montréal, Québec, Canada

Published in the International Journal of Critical Infrastructure Protection. *Available online December 7, 2024.*

Abstract

The Automatic Dependent Surveillance-Broadcast (ADS-B) is a surveillance technology mandated in many airspaces. It improves safety, increases efficiency and reduces air traffic congestion by broadcasting aircraft navigation data. Yet, ADS-B is vulnerable to spoofing attacks as it lacks mechanisms to ensure the integrity and authenticity of the data being supplied. None of the existing cryptographic solutions fully meet the backward compatibility and bandwidth preservation requirements of the standard. Hence, we propose the Compatible Authenticated Bandwidth-efficient Broadcast protocol for ADS-B (CABBA), an improved approach that integrates TESLA, phase-overlay modulation techniques and certificate-based PKI. As a result, entity authentication, data origin authentication, and data integrity are the security services that CABBA offers. To assess compliance with the standard, we designed an SDR-based implementation of CABBA and performed backward compatibility tests on commercial and general aviation (GA) ADS-B in receivers. Besides, we calculated the 1090ES band's activity factor and analyzed the channel occupancy rate according to ITU-R SM.2256-1 recommendation. Also, we performed a bit error rate analysis of CABBA messages. The results suggest that CABBA is backward compatible, does not incur significant communication overhead, and has an error rate that is acceptable for Eb/No values above 14 dB.

Keywords

ADS-B, security, authentication, backward compatibility, bandwidth efficiency, TESLA, PKI.

6.1 Introduction

Automatic Dependent Surveillance-Broadcast (ADS-B) is an aircraft surveillance technology [129] that allows aircraft to broadcast information about their identification, position, speed, and other data acquired from onboard sensors [12, 130, 131]. It supports many airborne and ground safety applications [10]. For example, Air Traffic Control (ATC) can use ADS-B information as an alternate means of surveillance, complementary to radar, to improve efficiency of controlled airspace [13, 14]. Furthermore, ADS-B provides an alternate source of information to allow airborne aircraft to maintain traffic separation.

ADS-B was originally designed in the early 2000s to replace radars as part of the United States Federal Aviation Administration (FAA) NextGen initiative [9]. It has since been adopted worldwide. Indeed, there are three certified ADS-B data links: the Universal Access Transceiver (UAT), which operates only in the United States at the 978 MHz frequency and uses 420-bit messages (272 bits for the payload); the 1 090 MHz Extended Squitter (1090ES), an internationally adopted link with 112-bit messages (56 bits for the payload); and the VHF Digital Link (VDL) MODE 4, which operates in the 108-136 975 MHz range with a message structure similar to that of the 1090ES, is most adopted in Northern Europe but is also rarely used due to the requirement for equipment upgrades [33].

Unfortunately, the ADS-B was conceived without any communication security mechanisms [29, 30], which represents a significant threat to aviation safety. Indeed, by using low-cost equipment such as a Software Defined Radio (SDR), attackers could easily transmit false ADS-B messages [28–30] to create a confused and false picture of traffic for controllers and pilots. This can potentially lead to flight delays, separation conflicts between aircraft or unnecessary maneuvers by pilots. In addition, spoofed ADS-B messages received and processed by Traffic Collision and Avoidance Systems (TCAS) in the cockpit could affect the decision-making ability of air crews [132]. Therefore, the use of ADS-B in ATC and traffic avoidance can represent a security risk.

Based on the foregoing, it is necessary to secure ADS-B. In particular, to prevent spoofing attacks, there must be a method to ensure *identity authentication* of the senders and *message authentication* of transmitted ADS-B messages. This is achieved through the simultaneous fulfillment of these three security goals:

1. *Data integrity*, is the assurance that data has not been altered in an unauthorized manner.

2. *Entity authentication*, also known as *Identity Authentication*, is the assurance of the identity of a given entity interacting with a system.

3. *Data origin authentication*, also known as *Message Authentication*, is the assurance that a given entity was the original source of received data.

Furthermore, any solution to secure ADS-B must adhere to the operational requirements delineated in the Minimum Operational Performance Standard (MOPS) for the 1 090 MHz frequency [12], which serves as the primary channel for ADS-B communications. Specifically, the solution must be backward compatible with current receivers, ensuring their ability to accurately receive, interpret, and display position information for nearby traffic. Additionally, the solution is mandated to minimize the utilization of the congested 1090ES, which is extensively utilized by Secondary Surveillance Radar (SSR) and Extended Squitter (Mode S) transmitters, such as radar, multi-lateration, and airborne TCAS [13, 14].

Several cryptographic solutions have been proposed to secure ADS-B. None of them appear to meet all the security goals and operational requirements listed above. Therefore, we consider that the question of how to secure ADS-B while meeting security and operational constraints is still open.

To that effect, in this paper, we introduce a solution called Compatible Authenticated Bandwidth-efficient Broadcast protocol for ADS-B (CABBA). CABBA integrates the Time Efficient Stream Loss-tolerant Authentication (TESLA) mechanism [43, 105] with phase overlay modulation techniques and a Public Key Infrastructure (PKI). By leveraging TESLA and PKI, CABBA fulfills all the security objectives specified earlier to prevent ADS-B spoofing attacks. This includes data integrity, data origin authentication, and identity authentication. Furthermore, by integrating the phase overlay modulation in CABBA's physical layer, we aim to align our solution with the operational requirements outlined by the MOPS.

Given the consequences of a potential attack exploiting the ADS-B vulnerability, one would hope that ADS-B be replaced as quickly as possible by a secure alternative. Unfortunately, such a one-to-one replacement will be lengthy and difficult in the context of aviation. First, it will likely take several years for an accepted standard to be drawn, discussed, approved, and then made mandatory by civil aviation authorities — at least 5 to 10 years. Second, considering the long lifetime of aircraft and their avionics, it is very likely that CABBA-capable and ADS-B legacy avionics would have to co-exist and use the same communication channels during the long transition period from initial deployment to full worldwide adoption. While it is paramount that CABBA-capable receivers be able to authenticate messages from CABBA-capable transmitters, it is equally important in terms of aviation safety that in the transition period both CABBA-capable and legacy ADS-B receivers be able to receive and interpret ADS-B messages from legacy ADS-B transmitters.

In light of these operational requirements, the two most important questions regarding any secure ADS-B solution, in particular CABBA, that needs to be answered are:

1. Could CABBA be gradually deployed while ensuring that legacy ADS-B equipment continues to operate?

2. What would be the viability of deploying CABBA in terms of communication channel saturation?

To evaluate the backward compatibility of CABBA, we have constructed an SDR-based implementation. We have used this implementation to test backward compatibility with two different suites of commercial-off-the-shelf (COTS) ADS-B In solutions: One is used in General Aviation (GA), and the other is used in business jets and airline transport aircraft. We also used this lab implementation of CABBA to test and analyze its bit error rate (BER). A channel occupancy rate (COR) analysis was also undertaken to quantify the channel occupancy overhead of CABBA in a likely real-world scenario. Besides, a safety impact assessment of unauthenticated messages was conducted to evaluate the effect of CABBA on the situational awareness of pilots and air traffic controllers.
Considering the above discussion, the contributions of this work can be summarized as follows:

1. We introduce CABBA, a secure variant of ADS-B technology that is bandwidth-efficient, backward compatible, and offers an adequate level of security by providing simultaneously two security services: aircraft identity authentication and ADS-B message authentication.

2. We use the D8PSK phase overlaid modulation technique, as defined in the MOPS, to support the transmission of additional security information required by CABBA while preserving bandwidth usage. To the best of our knowledge, this is the first proposal to use the phase overlay technique as specified in the MOPS.

3. We performed tests on a commercial aviation avionics suite and with a general aviation ADS-B in receiver to check whether our solution would be backward compatible with legacy equipment.

4. We carried out a channel occupancy analysis to verify the operational viability of our solution, in terms of channel occupancy.

5. We conducted a safety impact of unauthenticated messages to assess their effects on the situational awareness of pilots and controllers.

6. We provide a detailed specification of the CABBA protocol, including the structure of the different packet types (in-phase and quadrature), the authentication mechanism and the decision logic used to discriminate between genuine and false packets. This

specification is sufficiently detailed to allow anyone to implement the CABBA solution, and serve as the basis for subsequent standardization and adoption by the aviation industry.

The remainder of the paper is structured as follows. Section 6.2 reviews prior works on cryptographic approaches for securing ADS-B. Section 6.3 outlines the operational details of the TESLA protocol. Section 6.4 describes how phase overlay modulation techniques can be applied to ADS-B to increase data throughput while keeping the channel activity rate constant. Section 6.5 introduces CABBA, a cryptographic approach for securing ADS-B that integrates the TESLA authentication protocol with phase overlay modulation techniques. Section 6.6 details the experimental procedures used to assess CABBA backward compatibility and Section 6.7 the methodology for evaluating bit error, channel occupancy, and uncertainty delays. We conclude in Section 6.8 with a summary of our findings, describing their consequences in terms of possible real-world deployment of CABBA and highlight necessary future work in this direction.

Table 6.1 An overview of cryptographic techniques for enhancing ADS-B security. The approaches are categorized into three groups based on their use of symmetric, asymmetric, or hybrid cryptography.

	Cryptographic primitive	Security goals			Operational performances	
		Origin Auth	Integrity	Entity Auth	Backward compatibility	Bandwidth preservation
Symmetric	Encryption [79–82, 109]	✗	✗	✗	✗	✓
	MAC [83, 84]	✗	✗	✗	✓	✗
Asymmetric	Digital signature using certificate-based PKI [30, 88, 89]	✓	✓	✓	✓	✗
	Digital signature using Identity-based PKI [90–95]	✗	✗	✗	✓	✗
	Digital signature using certificateless PKI [40, 96–98]	✓	✓	✗	✓	✗
Hybrid	Encryption using TESLA with Certificate-based PKI [108]	✓	✓	✓	✗	✗
	MAC or Digital signature using TESLA with certificate-based PKI [106, 110]	✓	✓	✓	✓	✗
	MAC using TESLA with certificate-based PKI and phase overlay techniques	✓	✓	✓	✓	✓

6.2 Overview of cryptographic solutions for ADS-B

In this section, we review previous works and characterize the security goals and operational performance requirements they did not meet. We group these works into three categories, they use symmetric, asymmetric, and hybrid cryptography.

6.2.1 Symmetric cryptography-based protocols

The studies that use symmetric cryptography to secure the ADS-B rely on cryptographic primitives such as encryption or message authentication code.

Format-preserving encryption, or FPE, involves encrypting data in a manner such that the resulting *ciphertext* preserves the format of the original *plaintext* [133]. Some studies employ this approach because it aligns with the technological requirements of the ADS-B standard in preserving the bandwidth of the 1090ES channel [79–82, 109]. However, encryption schemes fall short of meeting the backward compatibility criteria of the ADS-B standard, primarily because navigation data are not transmitted in *plaintext*. To overcome that limitation, it has been suggested to use instead message authentication codes or MAC [83, 84]. For the MAC approach to be effective, there must be symmetric trust assurance between the communicating parties. However, it is challenging to achieve in open communications such as that of the ADS-B because it is often impossible to manage and master the parties involved in the broadcast. In such a scenario, knowing that when employing symmetric cryptography every receiver must know the symmetric key, a malicious actor can impersonate a sender and forge messages to other receivers.

To ensure authenticated broadcast, ADS-B requires an asymmetric process enabling every receiver to ascertain the genuineness of received messages, devoid of the ability to produce genuine messages from received ones [134]. Asymmetric cryptography, particularly digital signature, is the standard technique to achieve this [135].

6.2.2 Asymmetric cryptography-based protocols

In *asymmetric or public-key cryptography*, a pair of keys (public and private) is used for encryption and digital signatures. The *private key* is kept secret, while the *public key* is shared for secure communication. To guarantee authenticity, the public key must be confirmed by a certification authority (CA) through a public key certificate that links the key to an entity [99]. The system responsible for issuing, maintaining, and revoking certificates is known as a PKI [3, 38, 100]. For ADS-B, the literature proposes three types of PKI to manage aircraft certificates: certificate-based, identity-based, and certificateless.

Among the certificate-based PKI solutions proposed to secure ADS-B is an authentication scheme that relies on Elliptic Curve Digital Signature Algorithm (ECDSA) signatures and X.509 certificates [88]. Although this solution might fulfill the demands of the ADS-B protocol regarding security, it fails to meet the standard's technological performance criteria. In addition, the authors leave open the issue of certificate distribution and do not address that of certificate revocation. To address the weaknesses of [88], a lightweight PKI solution is recommended in [30], where the ADS-B message is signed, and its signature is partitioned across

N messages. It is suggested that keys distribution occurs during the routine maintenance of the aircraft. Furthermore, still to address the limitations of [88], a dual path PKI solution that aims to handle the certificate revocation problem by using session certificates is proposed in [89]. According to this scheme, an aircraft should have certificates from both their home country's National Aviation Authority (NAA) and the local ATC center where they are currently located. Thus, the dual certification is evidence that the aircraft has been granted permission to fly, as well as validated as a safe and current entity within the local center from which it is flying. We argue that the adoption of the PKI proposed in [89] will raise the operational expenses of the ADS-B system and render its use cumbersome, especially for international flights. In general terms, using certificate-based PKI for ADS-B security has two limitations. First, it significantly increases communication costs, conflicting with the bandwidth preservation criteria of the ADS-B standard, proof of which is that none of the above-mentioned solutions [30, 88, 89] meet this need. Second, establishing and operating a PKI for these solutions is impractical in the current state of global coordination among ICAO and NAAs.

Identity (ID)-based PKI attempts to eliminate the key distribution problem of certificate-based PKI. In ID-based PKI, public keys are derived from easily identifiable user attributes, such as email addresses, eliminating the need for traditional certificates and the complex infrastructure that supports them [39]. This is achieved through a central entity called a private key generator (PKG), tasked with computing each user's private key based on their corresponding public key [39, 136]. Several studies have used the ID-based authentication approach to secure ADS-B communications. For instance, a scheme that signs ADS-B messages in two stages, online and offline, has been developed to increase the efficiency of the signature generation process [90]. Furthermore, a broadcast authentication technique incorporating batch verification of digital signatures [1] has been proposed to reduce the time and computational expense involved in the signature verification process for ADS-B messages [91]. Subsequently, a broadcast authentication protocol that relies on ID-based signature and enables message recovery has been designed [92]. Aware that working with a single PKG in large-scale endeavors is not viable [137], the authors of the contribution [93] took inspiration from the hierarchical ID-based cryptosystems [2] presented in [137, 138] and implemented an authentication framework that relies on hierarchical ID-based signature (HIBS) and performs signature batch verification. However, the need for intricate hash-to-point operations

[1] Batch verification allows to simultaneously verify multiple digital signatures, whether they were produced by one signer or several.

[2] In a hierarchical ID-based cryptosystem, multiple PKGs create a tree-like structure [137, 138]. The primary PKG generates private keys for its subordinates, who, in turn, produce private keys for PKGs beneath them [137, 138]. PKGs at the edges generate private keys for users [137, 138].

during signature and verification processes renders the scheme [93] non-lightweight, reducing its deployability. To overcome this limitation, a three-level hierarchical ID-based signature scheme (TLHIBS) that relies solely on general hash functions has been introduced in [94]. Despite this effort, the issue of computational overhead persisted. In response, an alternative scheme that avoids employing any intricate bilinear pairing operations over elliptic curves has been implemented in [95]. This approach slightly reduces the computational overhead when compared to the previous works [90–94]. Besides, all these solutions have two additional drawbacks. First, they increase communication overhead, violating ADS-B bandwidth requirements for the 1090ES channel, which makes them unimplementable. Second, they are vulnerable to key escrow, a privacy issue in ID-based cryptosystems, allowing an untrustworthy PKG to decrypt messages and forge signatures by accessing users' secret keys [40]. This vulnerability raises substantial concerns about the overall security of these proposed solutions.

Certificateless PKI eliminates the key escrow problem by splitting the private key generation process between the PKG and the user. The PKG generates a portion of the private key, while the user creates a random value for the remaining portion, which is kept confidential. This approach has been used to implement ADS-B messages authentication schemes that rely on certificateless short signatures [96,97]. These schemes were subsequently enhanced by integrating privacy-preserving and aggregate signature methods to ensure sender anonymity and reduce the computational cost of signature verification [40,98]. The concept of certificateless short signature is new, and while it appears promising, it is not yet mature enough to be adopted. Indeed, a significant challenge in certificateless cryptography lies in the establishment of security schemes that can ensure a satisfactory level of protection against attackers attempting to manipulate users into employing counterfeit public keys. This difficulty arises from the absence of digital certificates to unequivocally verify the authenticity of a public key [42].

6.2.3 Hybrid cryptography-based protocols

So far, we have seen that there are two approaches to secure ADS-B while adhering to the standard's backward compatibility criteria. Through MACs using symmetric cryptography or through digital signatures using asymmetric cryptography, notably that based on certificate-based PKI. The digital signature approach is secure, however, the generated signatures are too long, which causes problems if we consider the requirement of preserving the 1090ES's bandwidth. On the other hand, the MAC approach allows generating short signatures, nevertheless, it is not secure since symmetric trust cannot be ensured between communicating parties. As a result, some authors have proposed using hybrid cryptography, particularly the

Timed Efficient Stream Loss-tolerant Authentication (TESLA) protocol [105]. Details of how TESLA operates can be found in Section 6.3.

Security in the Air using TESLA or SAT, is an authentication protocol that adapts TESLA to the requirements of ADS-B [106]. It replaces TESLA's synchronization protocol with onboard GPS clock time and employs certificate-based PKI for aircraft and message authentication. SAT, tested on `gr-air-modes`, shows potential backward compatibility with existing ADS-B receivers. However, it has two limitations. Firstly, it increases bandwidth usage by requiring three types of packets for message authentication. For simplicity of explanation, we refer to them as Type A, B and C packets. Standard 112-bit ADS-B packets are replaced with Type A packets that include a 16-bit MAC code and 8-bit sequence number, increasing to 136 bits (a 14% increase). Type B packets, containing TESLA authentication keys, are 184 bits long, and Type C packets containing aircraft certificates are 1520 bits long. Let Δ_B be the time between transmission of Type B packets (originally set to 5 seconds), and Δ_C be the time between transmission of certificate packets (originally set to 30 seconds). Assuming a mean transmission rate $\bar{f}_A$ of 6.2 ADS-B messages per second per aircraft, the use of SAT results in an additional transmission overhead per aircraft per minute given by:

$$\begin{aligned} O_{\min} &= (\bar{f}_A \cdot 60 \cdot 24) + \left(\frac{60}{\Delta_B} \cdot 184\right) + \left(\frac{60}{\Delta_C} \cdot 1520\right) \\ &= 14752. \end{aligned} \tag{6.1}$$

This results in a total overhead of 245.8 bps over the normal bit rate of 694.4 bps for standard ADS-B message transmission, a total 35% increase in bandwidth usage. The second limitation is related to security. To limit bandwidth usage, the authors of SAT limited the size of the MAC to 16 bits. Truncating the MAC like this is a standard described and accepted by FIPS standard 198-1 [139] and described in FIPS Standard Publication 800-107 [140]. In this case, the residual attack risk is two-fold:

1. The attacker is lucky and guesses the right MAC for a spoofed message he desires to send. This will happen with probability 2^{-16}.

2. The attacker floods the channel with spoofed messages with all MAC possibilities, i.e. he sends $65536 = 2^{16}$ messages hoping that the ADS-B receivers ignore the ones with a wrong MAC and process and accept the one with the correct.

In an ideal scenario where bandwidth is not constrained, we believe a larger MAC size would provide better security, ideally with a minimum of 32 bits, forcing the attacker to be extremely lucky or have to send an astronomical number of messages (over 4 billion messages) for his attack to be successful.

Securing Open Skies or SOS, is a solution that integrates TESLA with a mechanism for collectively verifying all messages transmitted by an aircraft within a specified timeframe [110]. Unlike the SAT method, which authenticates messages individually using MAC, SOS opts for batch authentication through digital signatures. This strategy is designed to effectively tackle the bandwidth consumption limitations of SAT and the broader challenge of bandwidth constraints in the 1090ES band. However, although transmitting one digest per message pool takes less bandwidth, the SOS technique can be troublesome in some instances. In the case of message injection, for example, the receivers must get the set of genuine messages. The authors propose a community server-based majority voting filtering stage. To identify the correct message sequence, servers try various message combinations as well as hash operations and comparisons. We argue that if an attacker injects false messages at a high rate, it will result in computation and a time-consuming task. Furthermore, should any of the ground receivers fail to receive a single packet, all packets delivered during that interval cannot be validated, posing a serious safety issue.

The solution presented in [108], combines Format-preserving, Feistel-based encryption, and TESLA to ensure the confidentiality and integrity of ADS-B messages. However, due to the lengthy security parameters required in their authentication technique, their solution necessitates the transmission of five ADS-B messages for every navigation data sent by an aircraft. This results in significant bandwidth consumption, thereby failing to meet the bandwidth preservation requirement outlined in the standard. Additionally, their proposed encryption of the ICAO code contradicts backward compatibility criteria. Consequently, this solution fails to comply with any of the operational requirements specified in the MOPS for ADS-B.

6.3 Background

In this section, we give a detailed explanation of how TESLA [105] protocol operates.

6.3.1 Timed Efficient Stream Loss-tolerant Authentication (TESLA)

The TESLA protocol combines asymmetric and symmetric cryptography to capitalize on their respective advantages. The core concept underlying TESLA is that the sender, Alice, adds to every packet a MAC computed with a secret authentication key K' known only by her. The receiver, Bob, buffers the packet when it arrives because he lacks the key to authenticate it. Only when Alice sends it to him, a while later, will he be able to verify the authenticity of the packet. To function properly, TESLA requires time synchronization of senders and receivers and, a trustworthy method for producing keys at the sender and authenticating them at the receiver.

The authentication keys are generated by the sender, Alice, before the broadcast begins. First, she divides the broadcast period into N time intervals. Second, she constructs a one-way keychain of length N, the last key generated K_0, serves as a pledge spanning the whole chain and may be used to verify any of the keys in the chain using the formula $K_0 = F^i(K_i)$. Third, she applies a one-way function F' to the keys of the keychain. This process generates the TESLA *authentication keys* $K'_i = F'(K_i)$, which are used to calculate the MAC of the messages to be broadcast.

Before starting to broadcast, Alice communicates the key disclosure delay d and the pledge to the keychain K_0 to Bob, the receiver, via a secure channel. Then, to broadcast a message m_j at time interval i, Alice must first compute the MAC = MAC(K'_i, m_j), then build the TESLA packet P_j which is then broadcast.

$$P_j = m_j \parallel \mathrm{MAC}(K'_i, m_j) \parallel K_{i-d} \tag{6.2}$$

When Bob receives P_j, he stores the triplet (i, m_j, MAC(K'_i, m_j)) in a buffer while waiting for the TESLA interval key that will allow him to deduce the authentication key K'_i and validate the MAC of the message m_j. Furthermore, Bob checks the authenticity of the origin of the interval key K_{i-d} by determining whether there exists a small integer v (i.e. of size commensurate with the number of intervals in a typical flight) such that $K_0 = F^v(K_{i-d})$. In such an event, Bob computes the authentication key $K'_{i-d} = F'(K_{i-d})$ and then validates the integrity of the messages broadcast within the time interval $i - d$ by computing their MAC and comparing them with the stored ones.

6.4 Phase overlaid modulation techniques

In this section, we focus on phase overlay modulation techniques and describe how they can be applied at the physical layer of ADS-B to increase data throughput while keeping the channel activity rate constant.

Systems that are currently envisioned by avionics system designers will most likely require more data transmission than the 6.2 messages per second restriction allowed by the ADS-B standard [12, 71]. Furthermore, increasing data throughput is a *sine qua non* condition for securing the protocol. Both industry and academia are aware of this need and have begun to look for methods to increase data throughput while meeting the standard's requirements of preserving the 1090ES band [12, 71, 142–144]. There are three versions of ADS-B, with the most recent (Version 3) released in the 2020 MOPS [12]. This version of the MOPS incorporates the notion of phase overlay capacity, which involves using alternate modulation techniques to increase data throughput without increasing channel activity rate. Although

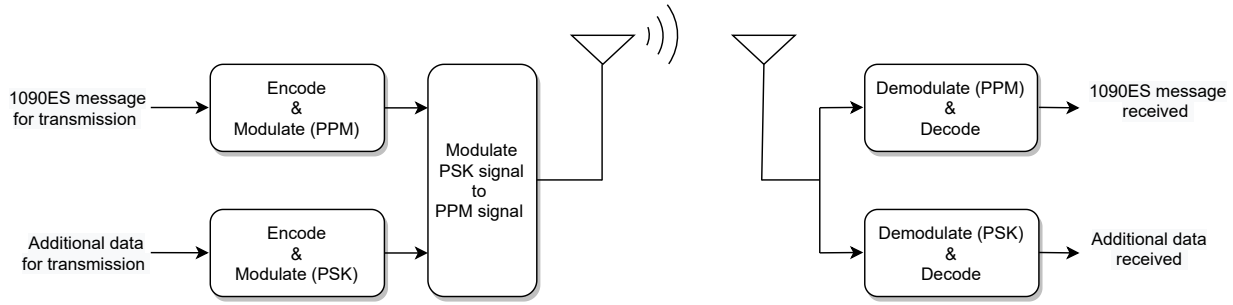


Figure 6.1 Block diagram for the use of phase overlay method to add more data to a 1090ES message, as explained in the patents [71, 141]. To perform an SDR-based implementation using an I-Q modulator at the input of the transmitting antenna and an I-Q demodulator at the output of the receiving antenna is the most practical way to proceed. In this way, the 1090ES message is conveyed in the in-phase (I) component of the carrier and the additional data in its quadrature (Q) component.

phase overlay is not required in this version of the MOPS, it is included so that stakeholders can begin designing, manufacturing and testing equipments and systems with the capability [70].

The MOPS proposes the use of the phase overlay functionality to encode additional bits of information into a conventional 1090ES message beyond the original 112 bits. The phase overlay method proposed is that described in a patent [71]. As depicted in Figure 6.1, this can be done by performing a pulse position modulation (PPM) on the 1090ES message to be transmitted, then performing a phase shift keying (PSK) modulation on the additional data to be transmitted. To complete the process, the PSK signal resulting from the previous step has to be modulated to the PPM signal resulting from the first step.

PSK is a modulation technique in which data is transmitted by altering the phase of the carrier wave. It was chosen as the overlay modulation method because it can be individually demodulated (Figure 6.1) and is non-destructive to the original message sent by amplitude modulation [71,141]. In principle, changing the phase of the carrier signal should not affect the older hardware's ability to decode the original 1090ES message [71, 141]. However, although there is agreement on the usage of PSK modulation as the overlay modulation technique, stakeholders are still divided on which combination to adopt. The study [144] suggest using the binary phase shift keying (BPSK) method, which allows doubling the throughout, sending a total of 224 (112*2) bits. However, this amount of bits is insufficient when we consider that the smallest digests produced by SHA-2 (SHA-224) and SHA-3 (SHAKE 128) are 224 bits and 128 bits, respectively [145,146]. Furthermore, the smallest ECDSA signature is 256 bits [135]. It is precisely in order to allow signing of ADS-B messages that researchers [142, 143] have

advocated using the Differential 16-Phase-Shift Keying (D16PSK) as it allows quintupling the throughput. However, as is widely known, increasing the modulation order increases BER. This tradeoff in transmission reliability is probably one of the reasons why the RTCA [12] advocates using D8PSK modulation, in combination with error correction codes such as Reed-Solomon or Low-Density Parity-Check. As a result, only 204 of the 336 extra bits provided by D8PSK can be used to convey extra information. Thus, while phase overlay techniques increase data capacity, there are still certain constraints when it comes to securing ADS-B communications. Traditional digital signatures using certificate-based PKI continue to be a concern regarding the communication cost, and this despite the increase in data capacity. In contrast, short signatures employing hybrid cryptography appear to be a more attractive option.

6.5 CABBA : Compatible Authenticated Bandwidth-efficient Broadcast protocol for ADS-B

The CABBA solution is presented in this section. CABBA seamlessly integrates phase-overlay modulation into the ADS-B physical layer and TESLA authentication into its application layer, while using a certificate-based public key infrastructure. CABBA is based on a new approach in which the fundamental TESLA concept of MAC-based message authentication and key disclosure delay remains intact, while introducing a significant transformation in the way information is transmitted and the type of information transmitted to enhance security. In the following lines, we delve into how CABBA distinguishes itself from previously proposed TESLA-based solutions, i.e. 1) the integration of phase overlay modulation and 2) enhanced packet structure.

Integration of Phase Overlay Modulation First, we enhance the physical layer of ADS-B by incorporating the phase overlay modulation technique proposed in the patent [71] recently promoted by the RTCA in the most recent version of the ADS-B MOPS [12]. As detailed in Figure 6.1, part of the information is conveyed in the in-phase component of the carrier, and the remaining information is in its quadrature component. For this purpose, we consider the two phase overlay modulation techniques mentioned earlier:

1. the D8PSK method advocated by RTCA [12], and
2. the D16PSK method proposed by academia [142, 143].

To determine which of these techniques is most appropriate for CABBA, we first implemented and conducted backward compatibility tests with both of them. Most importantly, we set to identify which of these modulation techniques provides an optimum trade-off between

higher data throughput and acceptable quality of signal, i.e. a lower BER, by conducting a simulation study. This is described in Section 6.6. Nonetheless, since D8PSK is the proposed standard and for the sake of simplicity of explanation, in the rest of this section we describe only the implementation with D8PSK. This configuration allows for an additional 336 (3*112) bits to be sent together with the ADS-B 112-bit original message.

Type				Content	Period	Size (bits)			
TESLA	SAT	SOS	CABBA			TESLA	SAT	SOS	CABBA
A				Message, MAC	-	-	-	-	-
	A			ADS-B message, MAC, sequence no.	-	-	136	-	-
		A1		ADS-B message	-	-	-	112	-
		A2		MAC	T_{A2}	-	-	128	-
			A	ADS-B message, MAC, sequence no.	-	-	-	-	112
B				Interval key	T_B	128	-	-	-
	B			Interval key	T_B	-	184	-	-
		B		Interval key	T_B	-	-	128	-
			B1	Interval key	T_{B1}	-	-	-	128
			B2	Interval key and signature	T_{B2}	-	-	-	210
	C			Interval key and signature; aircraft public key and signature	T_C	-	1520	-	-
			C	aircraft public key and signature	T_C	-	-	-	242

Table 6.2 A Comparison of CABBA's packet structure with that of earlier TESLA-based solutions. In CABBA, Type B packets are replaced with Type B1 Packets at the beginning of each interval (each $T_B = T_{B1}$ seconds) and by Type B2 packets every T_{B2} seconds. Type C packets are shorter than in SAT and sent with period T_C.

Enhanced Packet Structure The second distinction between CABBA and previous works relates to the content and structure of the packets to be transmitted, as highlighted in Table 6.2.

In TESLA, the security information in Type A packets (the MAC) and the interval key subsequently received via Type B packets allows the receiver to verify *data integrity* of the message. *Data origin authentication* is achieved by cross-referencing the information in Type B packets with additional data shared from sender to receiver via a secure communication channel. Both protocols, TESLA [105] and SOS [107], assume pre-existing trust between communicating parties, presupposing that the receiver possesses the sender's certificate beforehand.

In real-life operation, however, an aircraft cannot anticipate precisely which other planes it will encounter along its flight path. Consequently, the authors of SAT [106] propose a practical solution: distributing certificates through type C packets—an approach we endorse. Nonetheless, there are at least three aviation scenarios in which Bob (the receiver) may not require the certificate, as he may already possess the sender's public key and have duly authenticated its legitimacy. These scenarios are:

1. Bob, as an ADS-B ground station, receives messages from aircraft either directly (via

Line-of-Sight RF signal) or indirectly (via satellite). To authenticate these messages, Bob's ground station can access a PKI containing public keys of worldwide aircraft, indexed by their ICAO ID. This setup enables instant entity authentication without relying on aircraft to transmit certificates.

2. Alice, another ADS-B ground station, transmits information to airborne aircraft like Bob through LOS signal. Here, it's reasonable for the aircraft's receiver to hold a small, infrequently updated database of public-key certificates provided by the NAA (e.g., FAA) for authentication.

3. Air-to-air ADS-B transmissions pose challenges as it's impractical to preload worldwide aircraft public keys into each aircraft's receiver, let alone update them frequently due to aircraft turnover. However, future adoption of integrated digital communications like ATN could enable real-time access to remote PKIs, allowing aircraft to cache recently encountered aircraft's public keys for authentication.

To accommodate for such situations where the transmission of certificates might not be needed or not be needed as often, we propose the packet structure that follows :

Type A. Contain the ADS-B message its MAC and sequence number under the interval

Type B1. Contain the interval key K_i

Type B2. Contain K_i and the digital signature of K_i.

Type C. Contain the aircraft public key K_{pub} and its signature by the CA

Unlike SAT [106], Type C packets in CABBA do not contain interval key information or interval key signatures, and are therefore shorter. Signed interval keys are transmitted in a new type of packet, Type B2, which contains only an interval key and its signature. This has the advantage that signed keys can be sent with a lower frequency than certificates, resulting in a better use of bandwidth. In addition, this CABBA packet structure has the advantage of reducing bandwidth usage by eliminating the redundant transmissions of the interval keys, as it was the case in SAT [106].

6.5.1 CABBA on the sender side

CABBA requires airplanes to have a private-public key pair (K_{pr}, K_{pub}) and a certificate issued by a well-known and trusted certification authority. Before the flight begins, Alice, the sender, divides its duration into equal intervals of d seconds, and generates an authentication key for each interval. The process for generating these keys is the same as that used in

TESLA and SAT. During the flight, the ADS-B messages and their MAC, the authentication keys of the intervals, and the certificate of Alice's aircraft are sent as described below.

Sending a message and its MAC

To send an ADS-B message m at time interval i, Alice first produces the security data σ for message m. This includes:

1. The message MAC, formed by the λ leftmost bits of the message $\text{HMAC}(m, K'_i)$
2. The message sequence number s for m within that interval i

In other words $\sigma = \text{MAC} \,\|\, s$. As before, this information will continue to be encoded into the in-phase component of the RF signal. We denote by $P_{A-I} = m$ the message information sent in-phase. The security information σ will be sent using the quadrature component of the RF signal and is thus denoted $P_{A-Q} = \sigma$.
With the same packet length (112 bits) and containing the same information encoded in the same manner as standard ADS-B packets, P_{A-I} packets are intended to be fully intelligible by legacy ADS-B receivers. A logical packet P_{A-Q}, on the other hand, will in principle only be intelligible with CABBA-compliant receivers. With the choice of D8PSK, the highest quantity of bits that can be encoded in the quadrature component is 336 bits. Nonetheless, not all of these bits are available to encode the security information σ. The RTCA recommends using 12 bits to encode a reference phase and 120 parity bits to support the `RS(54,34)` error-correcting code, which must be applied to the σ security data. This means a maximum size of 204 bits for σ, which with the 8-bit sequence number s results in a maximum size of 196 bits for the MAC, i.e. $\lambda \leq 196$.
Alice uses the logical packet P_{A-I} to perform PPM on a pulse train to generate the signal $S_{A-I}(t)$ as follows:

$$\begin{aligned} S_{A-I}(t) &= \sum_{k=0}^{111} g(t - t_k) \quad ; \\ t_k &= kT_S + m_t(1 - P_{A-I}(k)) \end{aligned} \tag{6.3}$$

where $T_S = 1\,\mu s$ is the symbol period for 1090ES transmissions, $m_t = T_S/2$ is the PPM time-modulation index and $P_{A-I}(k)$ is the value of the k-th bit which will be transmitted at time $k * T_S$. Alice simultaneously uses logical packet P_{A-Q} to perform D8PSK modulation

on a sine wave to generate the signal $S_{A-Q}(t)$ as follows:

$$\begin{aligned} S_{A-Q}(t) &= \sum_{k=0}^{111} \sin\left(\omega_c t - \theta_k\right) \quad ; \\ \theta_k &= \frac{2\pi}{8}\,\text{symbol}_{P_{A-Q}}(k) \end{aligned} \tag{6.4}$$

where ω_c represents the carrier signal frequency, θ_k is the phase associated to the 8PSK symbol $\text{symbol}_{P_{A-Q}}(k) \in [0,7]$, which is computed from the three bits from P_{A-Q} to be transmitted at time $k * T_S$.

$$\begin{aligned} \text{symbol}_{P_{A-Q}}(k) =& 2^2 P_{A-Q}(3k) + 2^1 P_{A-Q}(3k+1) + \\ & 2^0 P_{A-Q}(3k+2) \end{aligned} \tag{6.5}$$

These two signals S_{A-I}(t) and S_{A-Q}(t) are then used by Alice to I-Q modulate (Equation 6.6) the 1090ES carrier and so, produce the radio signal S_A to be broadcast.

$$S_A(t) = S_{A-I}(t)\cos(\omega_c t) + S_{A-Q}(t)\sin(\omega_c t) \tag{6.6}$$

Sending authentication keys and their signatures

In order to allow the receiver to authenticate the Type A messages sent in interval i, the sender must later disclose the corresponding interval keys and their signatures. This is done by sending Type B1 and B2 packets in subsequent intervals.

Type B1 packets contain the TESLA interval K_i (128 bits) from which the authentication key $K'_i = F'(K_i)$ of the interval i is calculated. The corresponding packet P_{B1} will be transmitted during the next time interval $i+1$. These packets are sent at the beginning of each interval, i.e. every $T_{B1} = T_{int}$ seconds.

The signature of the authentication keys is added in Type B2 packets. B2 packets replace B1 packets at the beginning of the interval, every fixed number k of intervals. Their transmission period T_{B2} is thus a multiple of T_{B1}, with $T_{B2} = kT_{B1}$. A typical packet P_{B2} of this type will contain:

$$P_{B2} = K_i \,\|\, \text{sig}_{K_{pr}}(K_i) \tag{6.7}$$

where the $\text{sig}_{K_{pr}}$ represents the chosen signature-generating function with private key K_{pr}.

For Type B1 packets, the logical information P_{B1} is split between packets P_{B1-I} and P_{B1-Q} that will be transmitted through the in-phase and quadrature components of the RF signal. The in-phase packet P_{B1-I} contains the 50 leftmost bits of the K_I and the quadrature packet

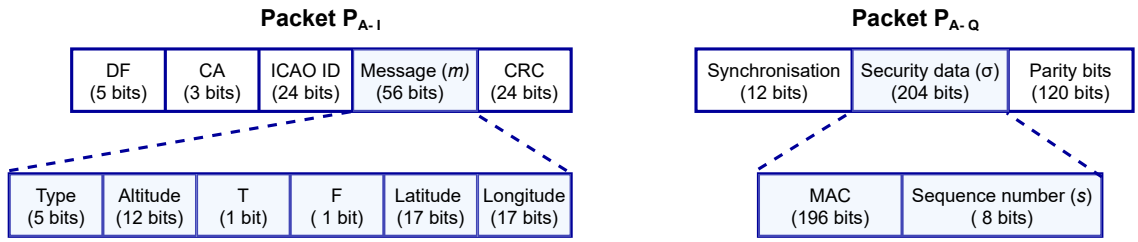


Figure 6.2 Structure of the Type A packets in CABBA. The ADS-B message m encoded in the in-phase component P_{A-I} (in this example an airborne position report) and the security data σ in the quadrature component P_{A-Q}. These two logical packets are then used to generate the in-phase and quadrature signal components S_{A-I} and S_{A-Q} of the RF signal to be transmitted S_A.

P_{B1-Q} contains the remaining 78 bits, as indicated in Figure 6.3a.
For Type B2 packets, the information is similarly split into packets P_{B2-I} and P_{B2-Q}. The in-phase component P_{B2-I} contains the entire interval key K_i and the leftmost 14 bits of the signature, while the quadrature component P_{B2-Q} contains the remaining 498 bits of the 512-bit signature.
The signal components S_{B1-I}, S_{B1_Q}, S_{B2-I} and S_{B2-Q} are then generated similarly as for Type A packets (Equations 6.3, 6.4 and 6.6).

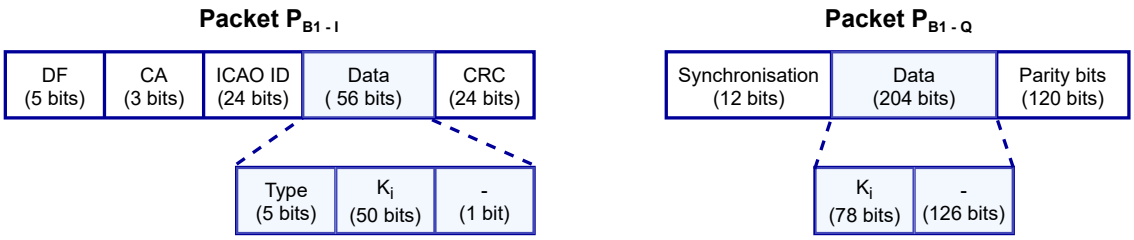
Sending the certificate of the transmitting aircraft

Alice will broadcast the certificate of aircraft every T_C seconds. The Type C packet P_C contains the public key of the aircraft K_{pub} and the signature of this key $\text{sig}_{K_{prCA}}(K_{pub})$. With a security strength of 128 bits, an ECDSA public key size of 256 bits is required, resulting in a signature size of 512 bits [3]. The first 181 bits of the public key K_{pub} are encoded in the in-phase packet P_{C-I} and the remaining 75 bits at the beginning of the quadrature packet P_{C-Q}. The 512 bits of the signature sig are also encoded into P_{C-Q}.
After encoding P_{C-I} and P_{C-Q}, the sender generates the signals S_{C-I}, S_{C-Q}, and finally the signal S_C which she broadcast. The procedure for producing these signals is the same as for producing signals for Type A and B packets.

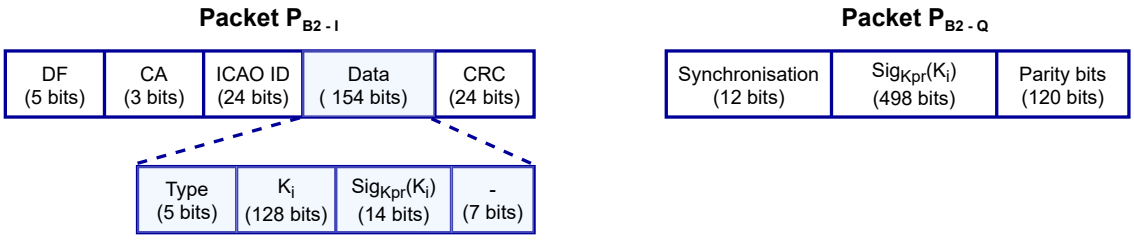
6.5.2 CABBA on the receiver side

Reception and demodulation of signals

The process of receiving and demodulating messages by Bob (the receiver) is the same for all packet types. The received signal S is first demodulated with quadrature local oscillators to obtain the in-phase component S_I and the quadrature component S_Q. Bob then performs



(a) Structure of Type B1 packets, 112 bits for P_{B1-I} and 336 bits P_{B2-Q}



(b) Structure of Type B2 packets, 210 bits for P_{B2-I} and 630 bits P_{B2-Q}

Figure 6.3 Structure of Type B1 and B2 packets, conveying only the authentication key or the authentication key and its signature, respectively.

PPM demodulation onto S_I and D8PSK demodulation onto S_Q, to generate logical packets P_I and P_Q, respectively. Depending on the format of these packets, Bob determines the original packet type (A, B1, B2 or C) and processes them accordingly. The processing of the CABBA messages contained within these logical packets is detailed below and depicted in the state diagram in Figure 6.5.

Processing the ADS-B message and its security data

As described above, the ADS-B message information and its security data is contained in Type A packet P_{A-I} and P_{A-Q}, respectively. Bob thus extracts the message m' from P_{A-I} and the security data σ (its sequence number s and the MAC) from P_{A-Q}. Lastly, he stores the triplet (m', s, MAC) in a buffer until he can verify the integrity of the message.

Verification of security properties

CABBA is an asynchronous protocol and there is no guarantee that messages corresponding to a particular aircraft will be received in any particular order. The state diagram in Figure 6.5 describes the various states in which the receiver of CABBA could be depending on what security information, i.e. what CABBA packet types, have been received so far. Note that such a state diagram is used for all messages received with the same ICAO ID, i.e. purportedly corresponding to the same aircraft.

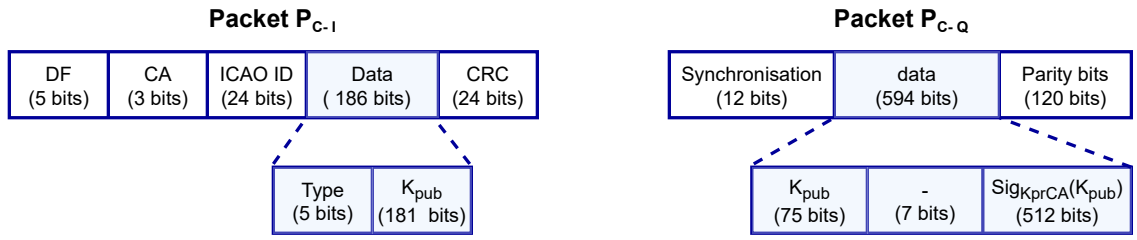


Figure 6.4 Structure of Type C packets. They contain the public key K_{pub} of the aircraft and the signature of said key $\text{sig}_{K_{prCA}}(K_{pub})$. These packets are then used to generate the transmitted signal S_C.

The state machine is initialized at state S_0 when the first packet for a given ICAO ID is received. If it is a Type A packet, it will be stored and the machine stays in the same state. Reception of type B1 packet containing an interval key will generate a transition to State S_1. Reception of a Type B2 packet, a *signed* interval key, will make the state machine transition to State S_2. Finally, the (unlikely but possible) reception of a certificate in a Type C packet before a Type B1 or B2 packet will transition to State S_3. In all of these states (S_1, S_2 and S_3), subsequent reception of ADS-B messages in Type A packets and further interval keys in Type B packets causes no transitions.
At this point, the receiver is unable to perform either entity authentication or data origin authentication of any messages received because some security information is missing (has not been received), i.e. either a validate certificate in the case of State S_2, a signed interval key in the case of State S_3, or both in State S_1. Nonetheless, the receiver is able to perform data integrity verification of the ADS-B messages received in previous intervals.

Data Integrity In order to validate the message integrity of a message m' received during interval i, Bob must have already received Type B1 packets P_{B1-I} and P_{B1_Q} at the beginning of the next interval $i+1$. From these packets, he will be able to reconstruct the interval key K_i by concatenating the first 50 bits contained in P_{B1-I} and the remaining 78 bits contained in P_{B1-Q} (as shown in Figure 6.2). The next step is to calculate the authentication key $K'_i = F'(K_i)$. Then, Bob calculates the "correct" HMAC of the received message m' with this authentication key K'_i as follows $\text{HMAC}' = \text{HMAC}(m', K'_i)$. Finally, Bob compares the λ leftmost bits of HMAC' with the received MAC. If they coincide, Bob will accept the message m', otherwise, he will ignore it.
Note that this verification only meets the goal of *data integrity* of the message m', i.e. that the message has not been modified after its MAC was computed by its originator, whomever the originator might be (friend or foe, real aircraft or hacker).

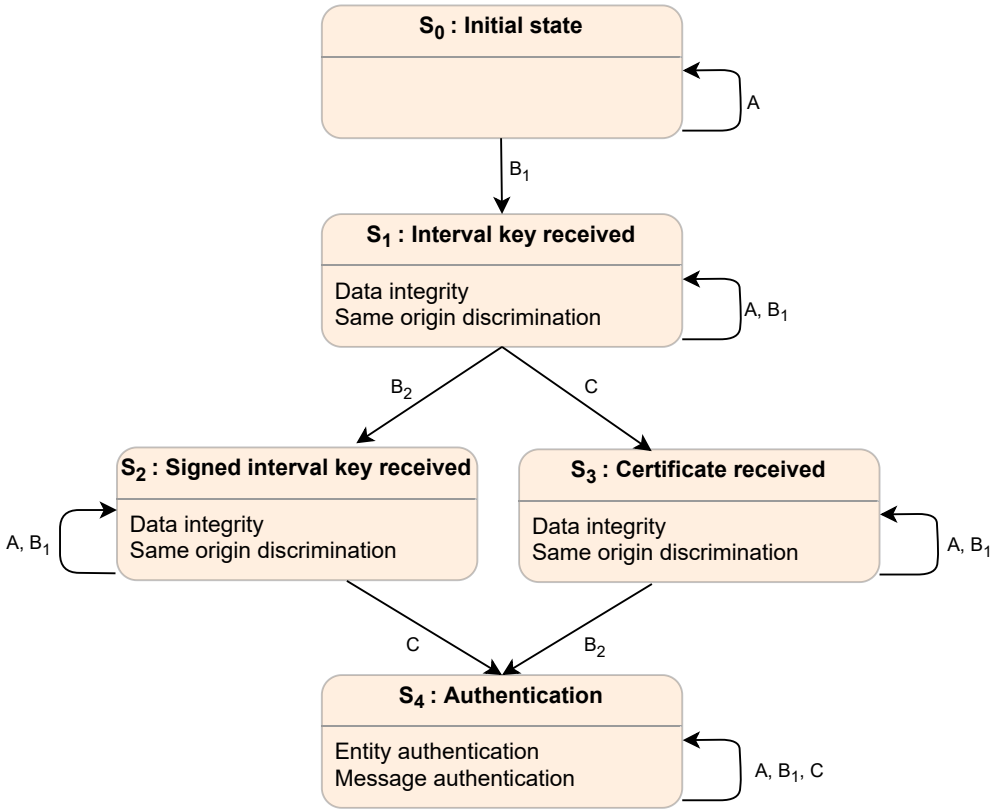


Figure 6.5 State diagram illustrating the authentication process for ADS-B messages on the receiver side.

Same-origin discrimination While in States S_1, S_2 and S_3 the receiver is unable to perform data origin authentication, there is an important security property that can be asserted at this point: *same-origin discrimination*, i.e. the ability to determine which message was sent by which sender, without necessarily having authenticated them.

To better understand this property, consider the following spoofing scenario. The attacker is aware that Alice's aircraft with ICAO ID X_A is currently broadcasting ADS-B messages that Bob is receiving. The intent of the spoofer is to send counterfeit ADS-B messages bearing the same ICAO ID X_A. Aware that the aircraft ADS-B transmitter and the receiver have implemented CABBA, the spoofer can generate his own interval key sequence and use it to authenticate his fake messages. More precisely, let $K^* = K_0^*, \ldots, K_N^*$ be the interval sequence generated by the aircraft and let $K^\dagger = K_0^\dagger, \ldots, K_N^\dagger$ be the sequence generated by the spoofer. The spoofer then generates his own messages, including hashes computed with his own key sequence.

If both spoofer and Alice's aircraft are in Bob's reception range, Bob would then receive these two message streams corresponding to the same ICAO ID X_A, potentially contradictory.

Thus, while Bob may not be able to determine which messages came from the spoofer and which came from Alice's aircraft, he will still be able to correctly associate a new message m' with either stream. He does this by identifying the TESLA keychain to which the interval key used to compute the MAC of m' belongs, as described in Formula 6.8.
Consider two messages m_1 and m_2 received by Bob at intervals i_1 and i_2, $i_2 > i_1$, respectively, and whose integrity was verified by Bob in the subsequent intervals $i_1 + 1$ and $i_2 + 1$ with interval keys K_{i_1} and K_{i_2}, respectively. Then if:

$$K_{i_1} = F^{(i_2 - i_1)}(K_{i_2}) \tag{6.8}$$

then Bob knows that m_1 and m_2 were sent by the same sender. In other words, in the above scenario Bob will be able to detect that there are two different senders A^* and $A^\dagger$ sending messages with the same ICAO code, and further know which message corresponds to which sender. He will not, however, know which one corresponds to the real aircraft A.

Authentication When Bob has received all required security information, i.e. a signed interval key and a certificate, he can then perform both identity authentication of the sender and message authentication (i.e. data origin authentication) of previously received messages. This will be possible when the state machine transitions to State S_4.

Identity Authentication Upon receiving Type C packet P_{C-I} and P_{C-Q}, the public key K_{pub} is extracted by concatenating the 181 bits in P_{C-I} and the first 75 bits in P_{C-Q}. The key signature $\text{sig}_{K_{prCA}}(K_{pub})$ is extracted from the remaining 512 bits of P_{C-Q}, as shown in Figure 6.4. Since Bob knows the public key K_{pubCA} of the Certificate Authority, he is able to verify the validity of the signature of the aircraft key the corresponding signature verification procedure `Verify`, that returns a boolean of `true` if the signature is valid. In other words, let v_1 be the result of this first signature verification:

$$v_1 = \texttt{Verify}(K_{pubCA}, \text{sig}_{K_{prCA}}(K_{pub}), K_{pub}) \tag{6.9}$$

Message authentication Finally, having received a Type B2 packet P_{B2-I} and P_{B2-Q}, Bob proceeds to extract the key K_i from P_{B2-I}. To obtain the signature $\text{sig}_{K_{pr}}(K_i)$, he concatenates the 14 bits from P_{B2-I} with the 498 bits from P_{B2-Q}. Using this information, Bob verifies the authenticity of the interval key K_i. Let v_2 be the result of this verification procedure, as defined by the equation:

$$v_2 = \texttt{Verify}(K_{pub}, \text{sig}_{K_{pr}}(K_i), K_i) \tag{6.10}$$

If v_2 evaluates `true`, it means that the key K_i is authentic, i.e. that it has been generated by Alice. Furthermore, this outcome also implies that all ADS-B messages for which the MAC has been computed using K_i can also be deemed authentic.

6.6 Backward compatibility experiments

We conducted backward-compatibility tests to verify that the phase overlay capability, as implemented in CABBA, does not affect the ability of existing hardware to decode the original ADS-B messages that are transmitted in the in-phase component of the 1090ES carrier.
To do so, we built an SDR-based implementation of CABBA and tested its backward compatibility with two distinct COTS ADS-B solutions:

1. The Appareo Stratus II ADS-B receiver, a non-certified portable device used in general aviation (GA) aircraft. The Stratus II was connected via Wi-Fi to an IPad running the ForeFlight application displaying ADS-B traffic information.

2. The Collins TSS-4100, a certified avionics device integrating TCAS, transponder and ADS-B traffic surveillance capabilities, used in business jets and airline transport aircraft. This equipment was connected to a Collins AFD-6520 Adaptative Flight Display to render the traffic information.

The experimental setup we employed involved the following steps:

1. Generate ADS-B messages and corresponding CABBA packets using custom-made scripts.

2. Generate and transmit the corresponding RF signals using the HackRF One SDR.

3. Receive these RF signals with the corresponding COTS receiver.

4. Check that the transmitted ADS-B information is received and correctly interpreted. We consider the test successful if the transmitted traffic information is displayed with the correct information (call sign, position, etc.).

In our experimental setup, the ADS-B messages and the corresponding packets were generated using custom-made scripts. These scripts are based on the *ADSB_Encoder.py* [147] scripts. This original script only generates ADS-B messages of the position report type, when given the ICAO, latitude, longitude, and altitude of an aircraft as inputs. However, the logic of ADS-B receivers is such that in order for them to consider a given aircraft's traffic information, they must receive all required ADS-B message types, i.e. identity, speed, status, and

operating status, at the frequency prescribed by the protocol. Thus, to conduct these tests, we built scripts that generate the remaining types of ADS-B messages[3].

The scripts we constructed further added the functionality required to generate CABBA messages (keys and certificates) and the corresponding packets. This includes among others functions to compute the MAC of ADS-B packets, to apply DPSK modulation to data to be transmitted in quadrature, to I-Q modulate the in-phase data in PPM with the quadrature data in DPSK.

For these backward-compatibility experiments, we only constructed and transmitted type A messages, which carry the ADS-B message and its security data. We did not transmit the other types of CABBA messages (B1, B2 and C), as these would be ignored by legacy receivers since they do not contain ADS-B data.

Our tests reveal that CABBA is backward-compatible with the two ADS-in receivers under test. Figure 6.6 shows the display of the ForeFlight application on the IPAD connected to the Stratus II receiver. The information displayed corresponds exactly with the information sent from the Hacker RF One SDR. The same is true for the information displayed on the AFD-6520 connected to the TSS-4100 transponder, as shown in Figure 6.7. The results suggest that using CABBA with legacy equipment will not compromise safety during the transitional period, where some aircraft would not yet have CABBA-capable ADS-B receivers. This finding supports the assumptions behind the MOPS used in CABBA and provides encouraging evidence for our implementation of it. However, further analysis and testing with a wider range of equipment in laboratory settings is needed, including packet reception analysis and, interoperability and stability tests. Once these tests are satisfactory, in-flight tests should follow, ideally in environments with high ADS-B channel usage and with sources of interference, such as multi-path transmissions due to terrain (mountains, water surface) or man-made obstacles (buildings, antennae, etc.). While we do not think that the use of the MOPS would affect backward compatibility in such real-world conditions, we do believe that it is important to study how transmission and bit error rate for the quadrature signal would be affected by such sources of interference and in high-channel usage.

6.7 Operational Viability of CABBA

While CABBA as proposed could provide a high level of security in terms of message authentication, there are some open questions regarding the viability of employing it in real-world situations due to operational and technological constraints.

First, we must determine which modulation scheme is most appropriate, D8PSK or D16PSK. Second, we must evaluate the bandwidth overhead of CABBA. Even with the use of PSK

[3]For this purpose the book *The 1090 Megahertz Riddle* [65] was an invaluable resource.

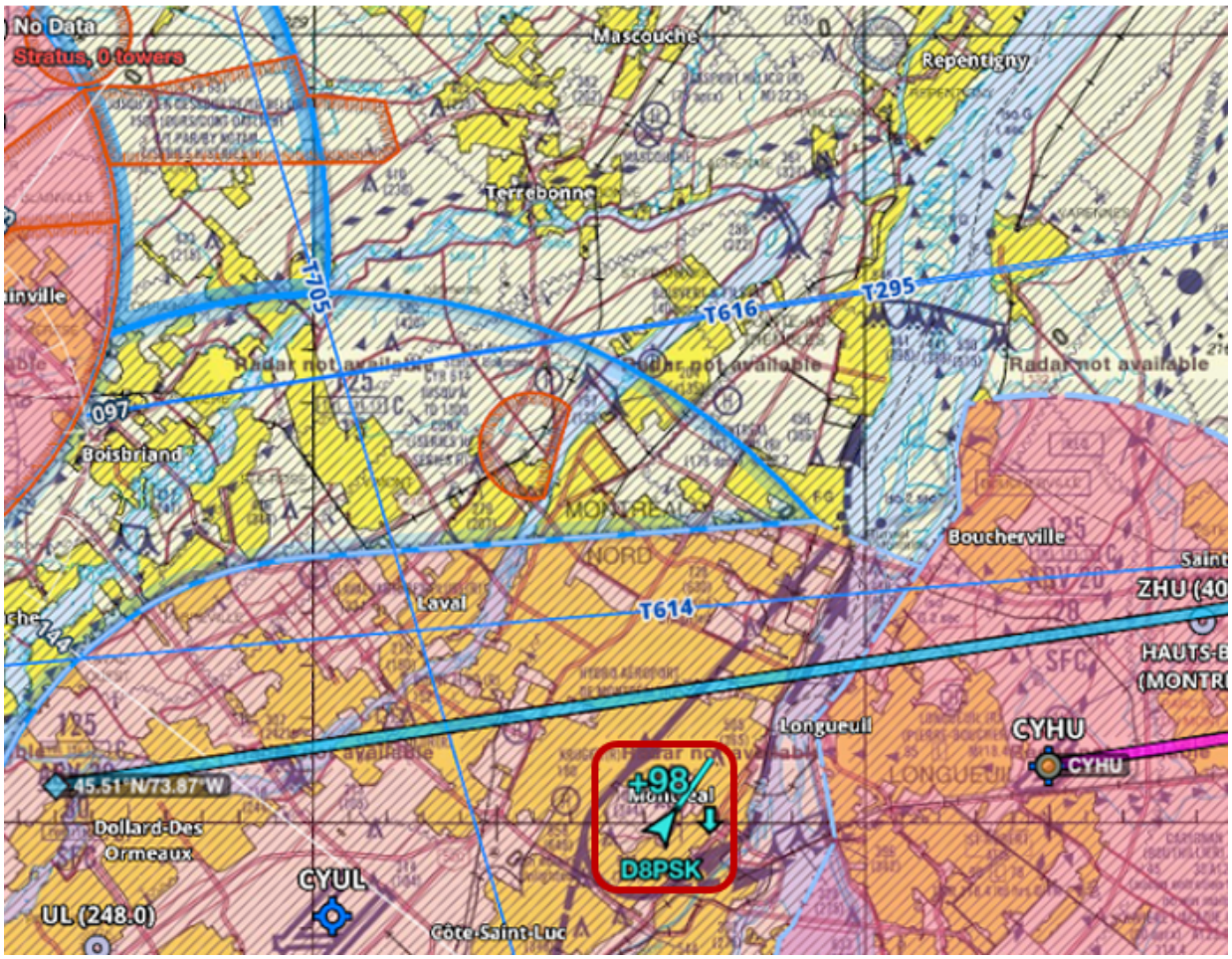


Figure 6.6 Screen capture of the ForeFlight Maps display with traffic option activated, showing the correct information for the "synthetic" aircraft with call sign "D8PSK", obtained from an IPad connected to the Stratus II receiver.

modulation, CABBA still requires more transmission time than plain ADS-B. It thus remains to be seen whether the resulting bandwidth overhead challenges its use in the already congested 1 090 MHz frequency. In this section, we describe our preliminary analysis of these questions using simulations to evaluate BER and real-world ADS-B data to conduct COR analysis.

6.7.1 Comparative BER analysis of CABBA with D8PSK vs. D16PSK

The aim of this BER analysis, is to determine which of the two phase overlay modulation schemes, D8PSK or D16PSK, provides the best balance between higher data throughput and acceptable signal quality, i.e. ADS-B service quality.

We used Simulink [148] to model the communication link of the CABBA protocol. Then, we

Figure 6.7 Here is a picture of the display AFD-6520 Adaptive Flight Display showing the correct information for the "synthetic" aircraft with call sign "D16PSK".

used the MATLAB program `bertool` to perform Monte-Carlo simulations to determine the BER across an Additive White Gaussian Noise (AWGN) channel.
A lower BER indicates a better performance; for ADS-B, the standard establishes a maximum BER of $10^{-}6$ [12]. The BER curves of the two implementations of CABBA that we wanted to compare, as well as the BER curves of the D8PSK, D16PSK, and D32PSK modulations, are depicted in Figure 6.8. By observing these curves, we notice that:

1. When implemented with D8PSK, CABBA fulfills the requirements of the standard for normalized signal-to-noise values (Eb/No) greater than or equal to 15 dB. For these values, the BER is equal to zero, indicating that the transmission is error-free.

2. When implemented using D16PSK, CABBA fails to meet the requirement of the standard.

Based on these results, we find that the D8PSK technique is the best method for implementing phase overlay functionality in avionics systems operating in the 1090ES band. In the ADS-B context, the D16PSK technique has a significant impact on data quality and reliability. Given the high error rates provided by D16PSK, the increase in throughput may not be worth it.

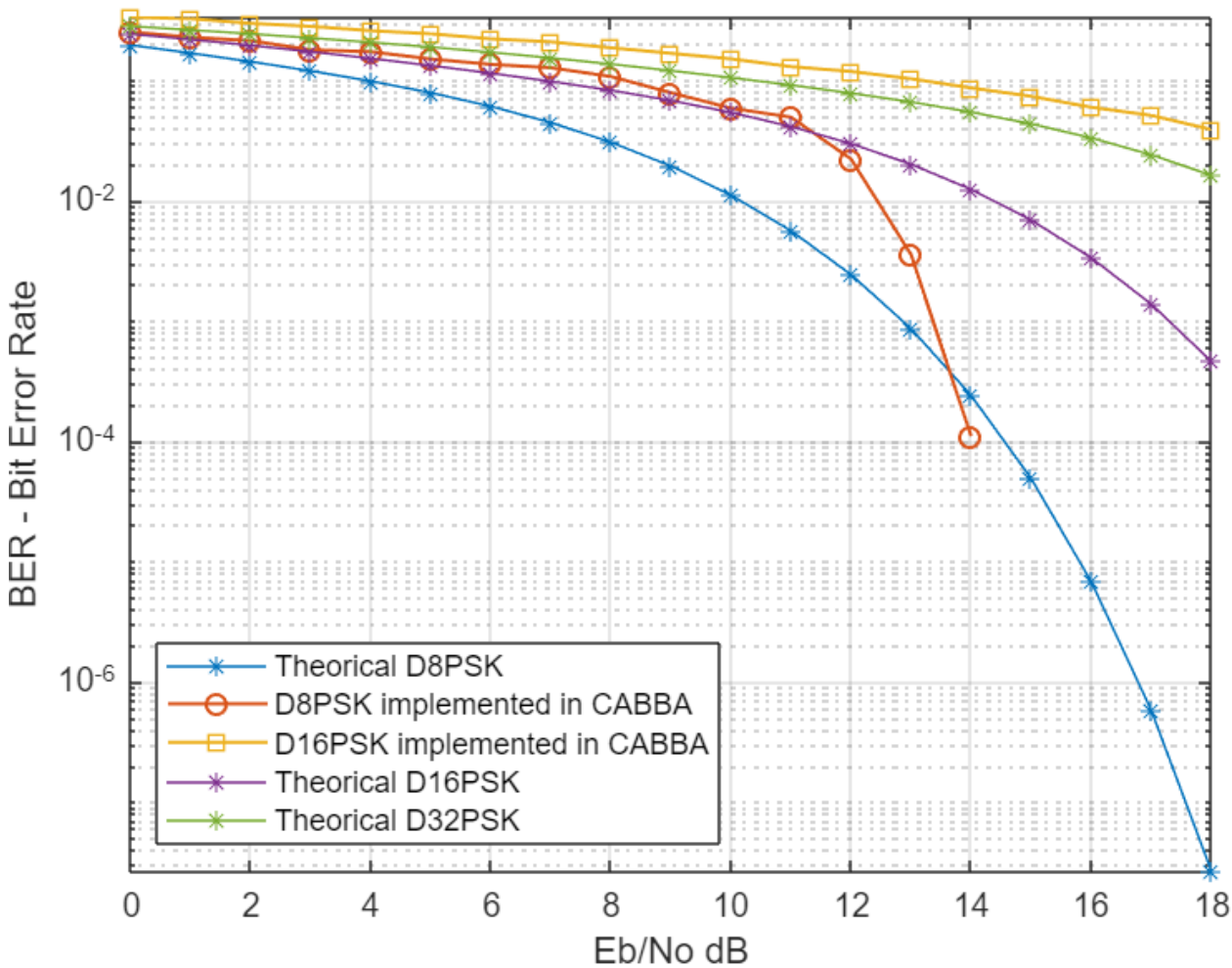


Figure 6.8 BER analysis shows that CABBA meets the standard's requirements for normalized signal-to-noise values (Eb/No) greater than or equal to 15 dB when implemented with D8PSK. Indeed, the standard requires a BER $< 10^{-6}$ and from Eb/No=15 the BER is equal to 0.

6.7.2 Channel occupancy rate (COR) analysis

We conducted a COR analysis to determine to what extent the transmission of non-standard ADS-B information, which are essential for CABBA support, decreases the available bandwidth.

The ITU report ITU-R SM.2256-1 [149] provides a detailed discussion on different approaches for measuring and evaluating spectrum occupancy, i.e. a methodology to conduct COR analyses. We used it as a guide to conduct our analysis. Indeed, the activity factor (γ) reflects how active the communication channel is. It is defined as follows [150]:

$$\gamma = \frac{\sum_{i=1}^{n} \Delta t_i}{\Delta t} \tag{6.11}$$

where Δt_i represents the channel occupation time for the i-th active transmission and Δt represents the total duration of the period being considered.
We created a baseline of normal 1090ES channel occupancy levels using real ADS-B data retrieved from the OpenSky Network database [151]. Since 2013, the OpenSky Network has been gathering continuous air traffic surveillance data as a non-profit community-based receiver network [152]. All unfiltered raw data is kept by OpenSky and made available to academic and institutional researchers.
To collect data for our research, we chose a receiver near Paris Orly airport (IATA code `ORY`, ICAO code `LFPO`). We chose this receiver because of the high density of aircraft traffic that can come within its reception range, including:

1. Aircraft transiting through the Northern France airspace, i.e. Paris Area Control Center (ACC), one of the busiest aerial corridors in the World. The ADS-B station could receive signals from aircraft at cruise altitude (30-35,000 feet) up to 200 nautical miles (360 km).

2. Aircraft transiting through the Paris Terminal Maneuvering Area (TMA) that are landing or departing from Paris Charles de Gaulle, Orly or Le Bourget, some of the busiest airports in Europe.

3. Aircraft on the ground at the Orly airport taxiing with transponders on.

We obtained a data capture of all traffic for this station for a 24-hour period on 3 August 2023. Obviously, aircraft traffic varies during the day, and hence so does 1090ES transmissions. We sampled the traffic within each 1-hour period and observed the transmission rate within 30 second-long periods within that hour. Taking six such samples for every hour, we observe quite a bit of variation in the number of transmissions within each hour; the corresponding confidence intervals are included in our results below.

In CABBA, we transmit four different types of packets, i.e. packet types A, B1, B2 and C. All packets of the same type have the same length and occupy the channel for the same duration. Let Δt_A, Δt_{B1}, Δt_{B2} and Δt_C be the transmission times for each of these packet

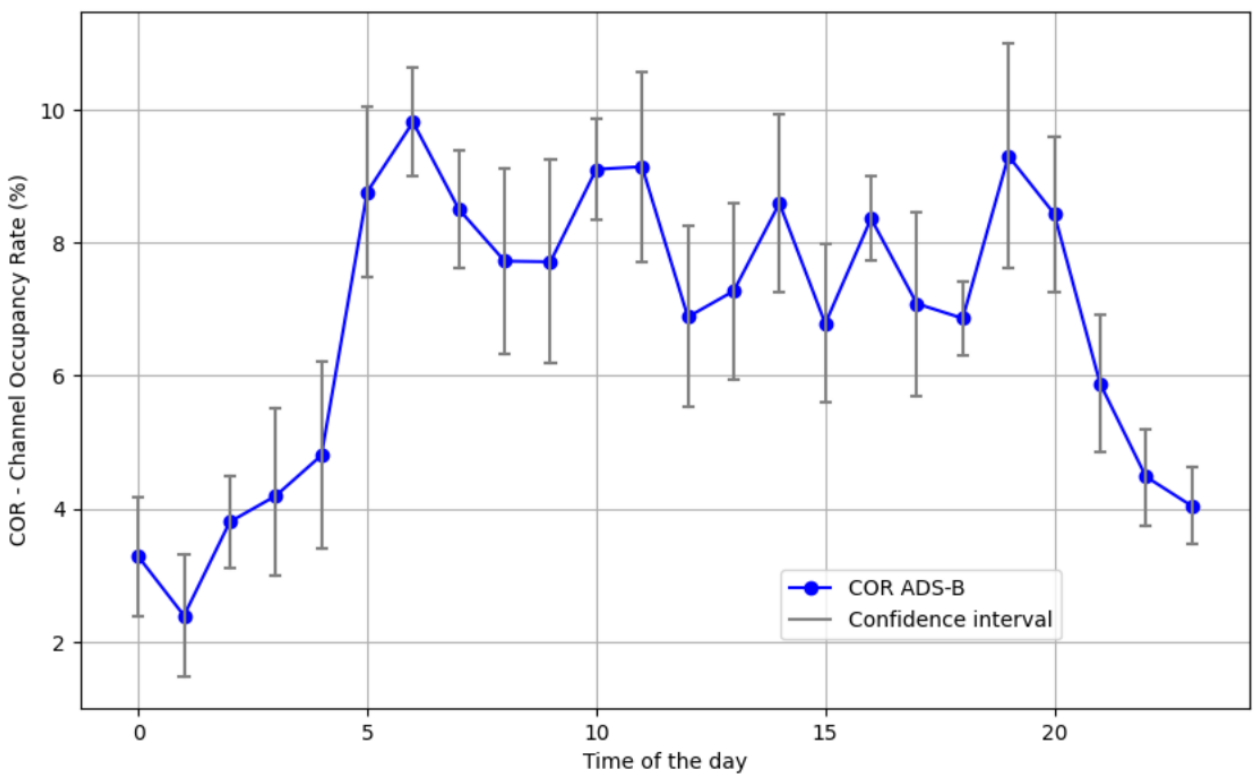


Figure 6.9 Mean COR for ADS-B transmissions with confidence intervals, computed with a sampling of six 30-second periods for every hour of the day, on 3 August 2023.

types. These values are proportional to the bit length of these packets (Table 6.2) *plus* the fixed 8-bit preamble. Given the 1090ES channel bit rate of 1 megabit/s, this results in 120 ms, 120 ms, 218 ms and 250 ms, respectively.

For each sampled time interval, let n_A, n_{B1}, n_{B2} and n_C be the number of packets of each that would be transmitted with CABBA. The resulting COR value is given by

$$\gamma = \frac{n_A \, \Delta t_A + n_{B1} \, \Delta t_{B1} + n_{B2} \, \Delta t_{B2} + n_C \, \Delta t_C}{\Delta t} \tag{6.12}$$

For our analysis, we conservatively consider that all aircraft in the dataset are CABBA-

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
T_{B1}	5 s	5 s	5 s	5 s
T_{B2}	5 s	10 s	10 s	15 s
T_C	5 s	15 s	20 s	30 s

Table 6.3 Transmission period parameters for each of the four scenarios for which we computed the COR values.

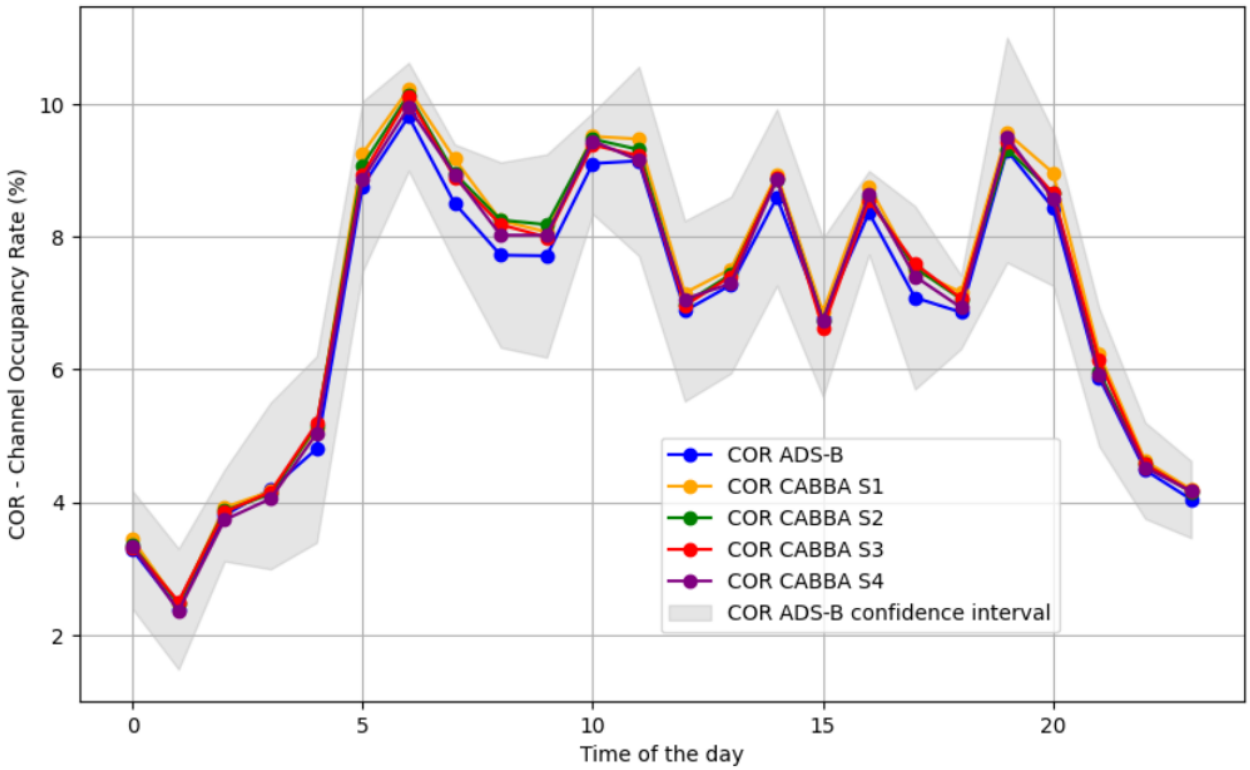


Figure 6.10 Estimated mean COR values per hour for every hour of 3 August 2023, for ADS-B and for CABBA in each of the three possible parameter settings described in Table 6.3.

capable and are sending all CABBA packet types as described in the protocol description in Section 6.5. Of course, our dataset only includes standard ADS-B packets and does not provide us with packet counts for non-standard CABBA packet type, except for Type A packets for which the count number n_A is the same as the number of ADS-B packets. For the other packet types, we estimate the number of transmissions to be equal to the number of different aircraft seen in the previous T seconds [153], where T is the transmission period of that type of packet. For example, let us consider Type B1 packets, i.e. packets containing unsigned interval keys. Each aircraft will send such a packet every T_{B1} seconds, i.e. with a T_{B1}-second period. At a given time t, let x be the number of different aircraft (with different ICAO ID) we have seen in our data set in the previous T_{B1} seconds. During that time period, some aircraft will have arrived in range and some others will have departed. If we assume that within the sampling interval (30 seconds) both the arrival and departure rate of aircraft are relatively small and similar to each other, we then can safely approximate the number of Type B1 packets that will be sent during that period to be x, i.e. $n_{B1} \approx x$. The same can be said for the counts n_{B2} of Type B2 and n_C of Type C packets which we approximate to be the number of different ICAO ID received in the previous T_{B2} and T_C seconds, respectively.

With these approximations, we are then able to compute the COR value γ from Equation 6.12. We consider four different parameter settings, described in Table 6.3. In all scenarios, we keep the same 5-second Tesla interval duration set in SAT. In the first scenario, Type C packets are sent at every Tesla interval along with Type B2 packets; Type B1 packets are thus never sent. This is the "safest" scenario in which CABBA-compatible receivers must wait at most 5 seconds until being able to authenticate messages, in the sense that the uncertainty period for CABBA receivers where originators cannot be authenticated is minimized. The fourth scenario is the most bandwidth efficient, with Type B2 packets being sent every other Tesla interval and Type C packet every six intervals.

The first finding of this study is that COR values for standard ADS-B vary between 2.4% and 9.8%, for the quietest and busiest hours, 01h00 and 06h00 UTC, i.e. 03h00 and 08h00 Paris local time, respectively. Second, with respect to CABBA we observe that the maximum overhead corresponds, obviously, to Scenario 1, with the lowest interarrival times for Type B1, B2 and C packets. In Scenario 1, the average overhead in terms of packets transmitted is 3.76% per 30-second period, with an observed maximum of 5.76% during that day. In comparison, for the most bandwidth-efficient Scenario 4, the average overhead is 1.56% and maximised at 2.38%. With respect to channel occupancy levels observed on that day, this results in a maximum COR increase of 0.43% of total bandwidth capacity for the most bandwidth-consuming Scenario 4. These results are represented graphically in Figure 6.10.

In summary, whatever the parameter setting scenario we choose, the impact of implementing CABBA in terms of bandwidth is negligible. The overhead in terms of packets sent is capped at less than 6%. This is in sharp contrast with the original SAT proposal, which has an estimated overhead of 35%.

At the observed channel occupation levels, i.e. COR values between 2% and 10%, the COR increase for CABBA even its most bandwidth-consuming setting is less than 1%. Even in more congested airspaces with hypothetical COR values of up to 40%, implementing CABBA would increase COR to less than 43%, a very acceptable compromise.

6.7.3 Safety Impact of Unauthenticated Messages

The key idea in TESLA that enables authenticated broadcast is the *delayed* key disclosure. However, this feature introduces a safety trade-off: messages cannot be immediately authenticated, resulting in an *uncertainty delay* Δ_u between message arrival and authentication.

As discussed in Section 6.5.2, to authenticate a message the following information must be in possession of the receiver: 1) the interval key for that message, 2) a signed interval key from this or a previous interval, and 3) the sender's certificate. This information may have already been received in Type B1, B2 or C packets. While the delay of arrival of such information

is bounded by their respective interarrival periods, their order of arrival at the receiver is non-deterministic and thus the actual value of Δ_u is also non-deterministic.
Further analysis of what these uncertainty delays are and what is their impact on aviation safety depends on in what application and context the ADS-B information is being used. We analyze various scenarios in TCAS and ATC applications below.

Impact of packet loss on uncertainty delay

In addition, we have to consider packet loss from noise, interference, or collisions due to congestion, which could introduce a further delay due to having to wait an additional 1 or 2 subsequent periods. The following equation describes the expected value of Δ_u as a function of the probability of packet loss p and the interarrival period T of the required information, where the first term corresponds to the packet being successfully received in the first interval, the second term in the second interval, etc.

$$\begin{aligned} \overline{\Delta_u} &= (1-p-p^2)\frac{T}{2} + p\left(T+\frac{T}{2}\right) + p^2\left(2T+\frac{T}{2}\right) \\ &= \frac{T}{2}(1+2p+4p^2) \end{aligned} \tag{6.13}$$

Previous works [152,154] have explored the packet loss ratio for ADS-B. They apply a conservative definition, where packet loss occurs if *any* receiver within the aircraft's range is missing the corresponding reception record. In the rest of this section, we approximate the probability of packet loss at a particular receptor, by interpolating from the empirical cumulative distribution function between packet loss and distance in [154].

Uncertainty Delays in TCAS

TCAS technology is essential in two environments: 1) non-radar environments, like oceanic regions, where ATC may be unable to provide separation services, and 2) dense traffic areas, such as busy airport terminals, where separation conflicts are likely to occur. The TCAS standards and supporting equipment ensure pilots have sufficient situational awareness of nearby aircraft within predefined *protection volumes*. These volumes are defined to address potential separation conflicts within specific *time* periods: 20–48 seconds for traffic advisories (TA) and 15–35 seconds for resolution advisories (RA) [155]. These intervals provide pilots with adequate time to enhance their situational awareness of incoming traffic (TA) and to execute evasive maneuvers (RA). It is thus very important that uncertainty delays do not significantly reduce the reaction time for pilots.
For aircraft in flight, the line-of-sight range corresponds to the distance to the horizon. For

typical altitudes that are small in comparison with the radius of the Earth, this can be approximated as follows:

$$\text{LOS range (in NM)} = 1.06 * \sqrt{\text{altitude (in feet)}} \tag{6.14}$$

In remote and oceanic areas, TCAS establishes a maximum lateral closure rate of 1 200 knots (1 200 nautical miles (NM) per hour = 2 222 km/h) [155], resulting in protected volumes with radii of 16 NM for TA and 11,6 NM for RA (distances corresponding to this closure velocity and the specified time periods for TA and RA). At a cruise altitude of 35 000 feet, the line-of-sight (LOS) range between aircraft is approximately 396,6 NM, providing 19,8 minutes of transmission time before the aircraft enters the protected volume. In terminal areas, arriving aircraft typically operate between 10 000 and 3 000 feet, resulting in a worst-case minimum LOS range of 116,1 NM (both at 3 000 feet). With a speed limit of 250 knots and a maximum closure rate of 500 knots, these aircraft will enter LOS range at least 13,9 minutes before entering each other's protected volumes. In both cases, this is significantly larger than the proposed T_{B2} and T_C periods, which guarantees reception of the required Type B1 and C packets, with high probability, even accounting for the packet loss rates at those distances. Thus, the only significant delay to be considered is the one due to the delayed transmission of the interval keys (T_{B1}).

At distances commensurate with the radii of protected volumes, the probability of packet loss for Type B1 packets is modest and the resulting expected value of Δ_u is less than 3,0 s, for both TA and RA. In other words, the use of CABBA reduces reaction time by at most 3 s, which we deem acceptable compared with the above-mentioned overall reaction times for TA and RA protected volumes.

TCAS	**radius**	p (%)	$\Delta_u(T_{B1})$ (s)	**time** (s)	**LOS** (min)	$\Delta_u(T_C)$ (s)
TA	6,6–16 NM (12,2–29,6 Km)	8,9	3,0	20–48	13,9–19,8	18,1
RA	5–11,6 NM (9,3–21,5 Km)	6,4	2,9	15–35	13,9–19,8	17,2

Table 6.4 Packet loss probability and uncertainty times for TCAS. We give the *radius* for the corresponding protected volume, the derived probability p of packet loss at that distance (as per [154]), the resulting expected authentication delay due to interval key transmission $\Delta_u(T_{B1})$, and compare it with the total reaction time as per the TCAS standard [155]. In addition, we compare the time in the LOS range with the expected uncertainty delay for transmission of Type C packets $\Delta_u(T_c)$. Here, we use the parameters for the "worst-case" Scenario 4 of Table 6.3, i.e. $T_{B1} = 5$ s and $T_{B2} = T_C = 30$ s.

Uncertainty Delays in ATC

In ATC, air traffic controllers utilize interconnected air traffic management (ATM) systems that can communicate with airborne aircraft via digital channels (e.g. ACARS, CPDLC). It is reasonable to expect that these systems will have access to a PKI with aircraft certificates or receive them along with flight plan information. The same can be said of signed keys for the initial interval of a flight, that could be transmitted by the aircraft at the gate. In this case, the primary factor affecting uncertainty delays would be the interarrival time of interval keys, i.e. the parameter T_{B1}.

If the certificate or signed key is unavailable, then the ATM system may have to rely on their transmission by CABBA through Type B1 and C2 packets, with the corresponding authentication uncertainty delay. The impact of this uncertainty will depend on the type and size of airspace being controlled.

Airport control zones (aka "Tower"). A sector centered on an airport, with a typical radius of 5 NM and a maximum altitude of 3 000 feet.

Terminal areas. A inverted-cone shaped sector above the control zone, with up to 40 NM in radius and up to 12 500 feet.

Area Control Center. A large sector, covered by surveillance from a single surveillance installation (radar or ADS-B receiver), typically with a 100–150 NM radius, to ensure that all aircraft in the area are well within minimum LOS range (approx. 140 NM for typical cruise altitudes18–60 000 feet in class A controlled airspace).

For tower and terminal areas, the LOS range is 58 and 118,5 NM, respectively. With a maximum speed of 250 knots, this results in minimum time in LOS for approaching aircraft of 13,9 and 28,4 min, respectively. For ACC, the LOS range is 140 NM for aircraft in Class A airspace, which at an unrestricted max cruise speed of 450 knots, results in a minimum time in LOS range of 18,6 min. As was the case with TCAS, in all of these cases, these LOS range transition times are widely sufficient for the ADS-B receiver to receive Type B2 and C packets with high probability, even considering the packet loss probabilities for those distances, as shown in Table 6.5.

As for the uncertainty delay created by the reception of the interval keys, in the case of tower and terminal areas, these are 3,0 and 4,1 s respectively, which compares very favorably with the refresh rate of typical Secondary Surveillance Radar (SSR) installations used for ATC (6 antenna rotations per minute, resulting a 10 s refresh period). The worst case here is that of the ACC, where the delay can be up to 14 sec, which remains comparable to the delay found in traditional radar-based infrastructures.

ATC	**radius**	p (%)	$\Delta_u(T_{B1})$ (s)	**update** (s)	**LOS** (min)	$\Delta_u(T_C)$ (s)
Tower	5 NM (9,3 Km)	2,8	3,0	10	13,9	16,3
Terminal	5–40 NM (9,3–74,1 Km)	22,2	4,1	10	28,4	25,0
ACC	100–150 NM (185,2–277,8 Km)	83,3	14,0	10	18,6	82,1

Table 6.5 Packet loss probability and uncertainty times for ATC. We give the *radius* for the corresponding control area, the derived probability p of packet loss, the expected authentication delay for interval key reception $\Delta_u(T_{B1})$, and compare it with the typical radar update rate. We also compare the time in the LOS range with the expected uncertainty delay for Type C packet reception $\Delta_u(T_c)$. We use the same parameters as before, i.e. $T_{B1} = 5$ s and $T_{B2} = T_C = 30$ s.

6.8 Conclusion

In this paper, we have explored the Compatible Authenticated Bandwidth-efficient Broadcast protocol for ADS-B (CABBA), a proposal designed to secure the Automatic Dependent Surveillance - Broadcast (ADS-B) protocol used in aviation. CABBA integrates the TESLA authentication protocol into the application layer of ADS-B. It also incorporates the phase overlay modulation techniques outlined in the Minimum Operational Performance Standard (MOPS) [12] into the physical layer of ADS-B. With these enhancements, CABBA strengthens ADS-B security and ensures the safety of the protocol by complying with the rigorous operational standards set by MOPS.

From an *operational and technical* point of view, CABBA shows promise. On one hand, preliminary tests indicate that the use of phase-overlay modulation techniques (D8PSK) proposed in the MOPS does not affect the capacity of legacy receivers to correctly interpret ADS-B messages. This would enable CABBA-compliant ADS-B hardware to co-exist with legacy ADS-B equipment without compromising safety. On the other hand, simulations using real ADS-B traffic data from high-traffic environments suggest a tolerable channel occupancy rate overhead when deploying CABBA. The results indicate that the bandwidth overhead when using CABBA is very reasonable and should not impede its deployment, even in congested airspace. Moreover, given the hardware and software architecture of most modern avionics systems, transforming legacy ADS-B equipment to support CABBA could probably be done with a firmware and software upgrade (e.g. in avionics using FPGA for signal processing). In such situations, the cost of upgrades and time to availability and certification would be less than a full avionics replacement.

From an *organizational* point of view, however, a robust international public-key infras-

tructure (PKI) must be established and operated. While there are ICAO ID databases in operations, they do not currently support certificate-based PKI sharing of aircraft public keys. Implementing such a PKI would require global agreement on trustworthy organizations to manage, share, and store aircraft public keys and their certificates. Although this poses a significant challenge, we believe it is doable in the relatively short term. A similar infrastructure exists for the sharing and storage of public keys for electronic Machine Readable Travel Documents (eMRTDs), including biometric passports [156]. It is currently supported by the ICAO Public Key Directory (PKD) with 90 participating countries [157]. We believe this framework could be expanded to include aircraft certificates for authenticating CABBA messages.

In conclusion, for all of these reasons supported by our experimental work and analysis, we believe that CABBA offers the best choice for a quicker deployment of a secure ADS-B solution that meets operational and technological requirements, while simultaneously achieving security and aviation safety objectives.

Acknowledgment

The authors would like to thank the partners of the CyberSA project: Queen's University, Bombardier, Collins Aerospace, Rhea Group, Carillon Information Security, the International Air Transport Association (IATA), the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Consortium for Research and Innovation in Aerospace in Québec (CRIAQ). Their support has been pivotal in bringing this research to fruition. We sincerely appreciate their collaborative efforts to advance research in the field of avionics systems cybersecurity.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used DeepL, Quillbot, ChatGPT, Grammarly and Turnitin in order to : 1) translate from French and Spanish to English (DeepL), 2) rephrase some sentences from previous work to write the SOTA (Quillbot and ChatGPT), 3) enhance the quality and clarity of specific sentences (ChatGPT), 4) check grammar and spelling mistakes (Grammarly), check for plagiarism (Turnitin). After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

CHAPTER 7 ARTICLE 2 - NEW MACHINE LEARNING APPROACHES FOR INTRUSION DETECTION IN ADS-B

Mikaëla Ngamboé, Jean-Simon Marrocco, Jean-Yves Ouattara,
José M. Fernandez (Retired), Gabriela Nicolescu

Computer and Software Engineering
Polytechnique Montréal
Montréal, Canada

Published in the Proceedings of the 2025 AIAA/IEEE 44th Digital Avionics Systems Conference (DASC). *Accepted on July 22, 2025.*

Abstract

With the growing reliance on the vulnerable Automatic Dependent Surveillance–Broadcast (ADS-B) protocol in air traffic management (ATM), ensuring security is critical. This study investigates emerging machine learning models and training strategies to improve AI-based intrusion detection systems (IDS) for ADS-B. Focusing on ground-based ATM systems, we evaluate two deep learning IDS implementations: one using a transformer encoder and the other an extended Long Short-Term Memory (xLSTM) network, marking the first xLSTM-based IDS for ADS-B. A transfer learning strategy was employed, involving pre-training on benign ADS-B messages and fine-tuning with labeled data containing instances of tampered messages. Results show this approach outperforms existing methods, particularly in identifying subtle attacks that progressively undermine situational awareness. The xLSTM-based IDS achieves an F1-score of 98.9%, surpassing the transformer-based model at 94.3%. Tests on unseen attacks validated the generalization ability of the xLSTM model. Inference latency analysis shows that the 7.26-second delay introduced by the xLSTM-based IDS fits within the Secondary Surveillance Radar (SSR) refresh interval (5–12 s), although it may be restrictive for time-critical operations. While the transformer-based IDS achieves a 2.1-second latency, it does so at the cost of lower detection performance.

Keywords

ADS-B, Intrusion detection systems, IDS, Deep learning, Transfer learning, xLSTM, Transformer.

7.1 Introduction

Automated Dependent Surveillance-Broadcast (ADS-B) technology is essential for air traffic management and broadcasting real-time aircraft navigation data [12]. Its adoption has significantly enhanced flight safety and improved airspace efficiency by enabling better situational awareness for pilots and air traffic controllers. However, ADS-B is vulnerable to cyberattacks [26, 28, 30]. This vulnerability stems from the absence of entity authentication, data authentication, and data-integrity verification mechanisms in its design.

To address these vulnerabilities, researchers have proposed a series of countermeasures that fall into two main categories: (1) adding an authentication layer to the ADS-B protocol, primarily through cryptographic methods, and (2) detecting altered messages or signals using non-cryptographic techniques such as multilateration, Kalman filtering, physical layer analysis, and machine learning.

One might wonder: If cryptographic methods can effectively prevent intrusions, why is there still a need for non-cryptographic detection techniques? Although cryptographic methods are essential for ensuring message authenticity and are generally effective, they are not infallible. For instance, the theft or misuse of a secret key can compromise the entire protection scheme. In such cases, preventive mechanisms may fail silently, allowing malicious messages to be accepted as legitimate. This is where detection controls become indispensable. Operating on the receiving side, they monitor system behavior to detect signs of data tampering and uncover attacks that bypass or exploit weaknesses in preventive countermeasures. By doing so, detection techniques address residual risks that persist despite strong authentication, thereby complementing cryptographic solutions. Accordingly, we advocate a defense-in-depth strategy that combines multiple layers of security to enhance the resilience of ADS-B against cyber attacks.

In this study, we focus on machine learning strategies for intrusion detection, particularly deep learning methods, due to their effectiveness in addressing anomalies that affect ADS-B data. According to Chandola *et al.* [118], an anomaly can be classified as a *point anomaly*, which is a single data point deviating from the expected behavior; a *collective anomaly*, involving data points that together show abnormal behavior; and a *contextual anomaly*, which is anomalous only within specific temporal or operational contexts [118, 158]. In time-series data, such as ADS-B, anomalies are typically contextual. For example, an aircraft at 10,000 feet may be normal for domestic flights, but anomalous over the Atlantic, where cruise altitudes exceed 30,000 feet.

Traditional clustering methods (e.g., DBSCAN [159]) and ensemble methods (e.g., Isolation Forests [124]) effectively detect point anomalies, but struggle with contextual anomalies. Statistical methods rely on distributional assumptions that often fail in practice, and the

selection of appropriate test statistics remains challenging [118]. Deep learning methods, particularly neural networks trained on normal behavioral patterns, better capture the temporal dependencies and multivariate dynamics of ADS-B messages. Studies have highlighted the potential of autoencoders and other deep learning architectures to build robust intrusion detection systems for ADS-B [47, 48, 160].

Autoencoders, when combined with recurrent neural networks (RNN [116]), particularly Long Short-Term Memory (LSTM [49]) networks, excel in countering coarse attacks, such as jamming [161]. However, they struggle with subtle message injections, such as gradual attacks, in which a specific feature of the ADS-B message is subtly altered over time. The inherent limitations of LSTM, including irreversible storage decisions and limitations in memory and computational capacity, make it challenging to enhance LSTM-based autoencoder models in this context. This has spurred interest in context-aware architectures such as contextual autoencoders and transformers. Transformers [5] with their self-attention mechanism, enhance contextual awareness by capturing long-term dependencies more effectively than LSTM. However, self-attention scales quadratically with sequence length, increasing computational and environmental costs. By contrast, extended LSTM (xLSTM) [6] introduces efficient memory architectures that maintain long-term dependencies through recurrent operations. This reduces reliance on global attention [52] and suits time-series data, such as ADS-B, where tracking the order and timing of events is more relevant than accessing the entire context at once. Consequently, xLSTM emerges as a suitable architecture for intrusion detection in ADS-B.

Beyond deep-learning architecture innovations, recent advances in intrusion detection systems (IDS) design have focused on how learning is structured and transferred across tasks [125]. One promising direction is using transfer learning, where a model is first pre-trained to capture important characteristics of the normal behavior of the system. This learned knowledge is then applied to a downstream anomaly detection task to help the system distinguish between benign and malicious activities more effectively [125, 126]. By leveraging these pre-trained representations, IDS models can improve generalization and enhance their ability to detect novel or previously unseen attacks [112].

With recent advancements in deep learning architectures and IDS implementation strategies, it is essential to assess the potential of emerging solutions to address the ongoing challenge of implementing an efficient IDS for ADS-B. In this study, we propose and evaluate two deep learning-based IDS specifically designed for ADS-B data. Our focus is on ground-based detection systems, such as those used by air traffic management (ATM), where the application of machine learning is less limited by computational constraints than in regular avionics systems. Regarding our IDS, the first implementation utilizes the encoder component

of a transformer architecture, whereas the second is based on the xLSTM architecture. To the best of our knowledge, this is the first implementation of an xLSTM-based IDS specifically tailored to ADS-B. Furthermore, to the best of our knowledge, this is the first application of a transfer learning approach for implementing an IDS for ADS-B.

Our transfer learning strategy uses a two-stage process. The models were pre-trained on benign ADS-B traffic to learn contextual patterns and temporal dependencies in ADS-B communications, thereby enhancing generalization across flight trajectories. The pre-trained models were then fine-tuned using labeled datasets to develop specialized models for detecting different types of gradual attacks. These specialized models were subsequently integrated into a unified multiclass classifier capable of accurately identifying and categorizing various types of ADS-B intrusions. We evaluated the classifier for both known and unknown attacks, and the results show that the xLSTM architecture outperforms the transformer and demonstrates robust generalization to novel threats.

The remainder of this paper is organized as follows. Section II outlines ADS-B threat models and details the specific one examined in this study. Section III reviews recent deep-learning-based approaches for intrusion detection in ADS-B. Section IV provides the background of the extended LSTM (xLSTM) architecture. Section V details the proposed methodology. Section VI describes the experimental setup and Section VII discusses the results. Finally, Section VIII concludes the paper and outlines directions for future research.

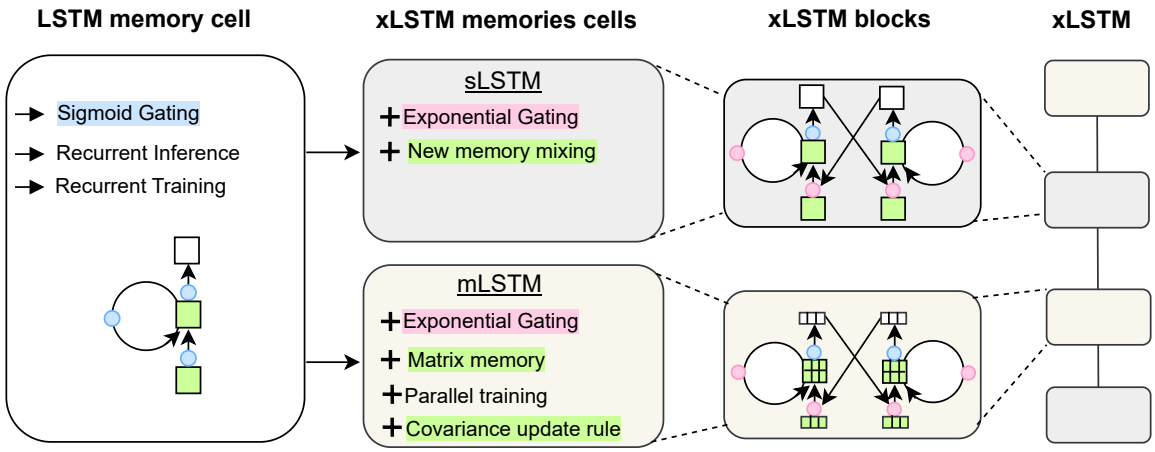


Figure 7.1 Architecture of the original LSTM memory cells and the new xLSTM variants (sLSTM and mLSTM), based on the illustration in paper [6]

7.2 ADS-B Threat Model

The lack of security controls in ADS-B allows malicious actors to exploit the system by injecting, altering, or suppressing messages without detection [26, 28]. An overview of the main attacks targeting the ADS-B system is provided in [26, 28, 30], including:

- **Eavesdropping:** Passive interception of genuine ADS-B messages.
- **Jamming:** Disruption of RF channels to prevent the transmission of genuine ADS-B messages.
- **Message deletion:** Suppression or removal of genuine ADS-B messages.
- **Message modification:** Alteration of genuine ADS-B messages, potentially falsifying aircraft position, velocity, or identity.
- **Message injection:** Transmission of fabricated ADS-B messages, potentially introducing spoofed aircraft or falsified flight data into the surveillance network.

In this study, we deliberately exclude eavesdropping, jamming, and message deletion attacks. Eavesdropping is not considered a direct threat unless combined with active attacks [26]. Jamming is generic to radio frequency (RF) systems and is typically easy to detect. Similarly, message deletion attacks, when performed independently, are easily flagged due to noticeable gaps in aircraft tracking data.

Our primary focus is on message modification and message injection attacks, which allow adversaries to discreetly alter surveillance data. A significant subset of these attacks is known as gradual attacks, which involve the subtle and continuous alteration of specific ADS-B message features, such as altitude, latitude, etc. These attacks can be executed by either modifying intercepted messages or injecting crafted ones. The insidious nature of gradual attacks enables them to undermine situational awareness over time without triggering immediate alarms, making them particularly difficult to detect and justifying their selection as the central focus of this study.

In this context, the adversary is conceptualized as an unauthorized individual operating from the ground or the air with full control over the communication channel (1090 MHz). This control enables the adversary to suppress legitimate ADS-B traffic, ensuring that the victim (the controller) receives only the information the adversary wishes to convey. This scenario was selected for its plausibility and minimal resource requirements. Tools such as software-defined radios (SDRs) are affordable and widely available, allowing attackers to broadcast falsified ADS-B messages over long distances. Although internal threats from insiders, such

as aircraft and airport maintenance technicians, are relevant, this study focuses on external ground-based attacks.

7.3 Previous works

An overview of the deep learning-driven IDS introduced in the literature to enhance the security of ADS-B is provided below.

Habler and Shabtai [161] were pioneers in employing machine learning techniques to detect anomalous ADS-B messages through an LSTM encoder-decoder model, which was trained on legitimate flight sequences from takeoff to landing. This model processes new sequences by transforming them into fixed-dimensional vectors using an encoder, followed by reconstruction through the decoder. Anomalous sequences are indicated by higher reconstruction errors. Their approach focuses on data from individual aircraft, overlooking the spatio-temporal correlations among multiple aircraft sharing the same airspace, which compromises accuracy due to limited situational awareness. Akerman *et al.* [119] and Olive *et al.* [120] address this limitation by considering broader traffic flow.

Akerman *et al.* [119] aggregates ADS-B messages from multiple aircraft within geographical areas as image streams, using a ConvLSTM encoder-decoder to detect anomalies. The model analyzes image sequences and identifies anomalies when the reconstructed output deviates significantly from the input. An explainability technique provides visual indicators of anomalies to assist in pilot decision-making. Olive *et al.* [120] integrated trajectory clustering with autoencoders to detect anomalies within traffic flows by introducing a custom regularization term based on the distribution distance to optimize the training for sparse clusters. The model generates reconstruction error scores for trajectories, thereby facilitating the identification of anomalous situations in air-traffic operations.

Fried *et al.* [121] contend that training distinct models for each location, as demonstrated in [119,120], restricts solutions to flights with sufficient historical data, which is often lacking in business aviation, instructional flying, and aerial work. To address this, they proposed transforming ADS-B data before classification using a non-recurrent autoencoder. These transformations include converting geodetic coordinates to 3D Cartesian coordinates, applying K-lag and K-order differencing to eliminate trends, and extracting time-series characteristics such as maximum, minimum, mean, median, and variance. These properties serve as inputs for a non-recurrent autoencoder. They compared their approach to recurrent autoencoders, noting that their method extracts time-series characteristics, a step omitted by the recurrent autoencoders.

Although the approach in [121] addresses location-specific model constraints, traditional autoencoders, whether recurrent or non-recurrent, map inputs to fixed points within the latent

space, limiting their capacity to capture the full variability of the data. In contrast, models such as variational autoencoders (VAE) present a more flexible alternative by sampling from a distribution defined by the encoder's output, thereby enhancing the representation of uncertainty and variability, which is an advantage for anomaly detection. Luo *et al.* [122] proposed a model that integrates a VAE with Support Vector Data Description (SVDD) to detect anomalies in ADS-B data. The VAE is utilized to reconstruct ADS-B messages, and the reconstruction error is employed to train the SVDD model, which establishes a threshold around normal data. During the testing phase, messages that exceeded this threshold were identified as anomalous.

Chevrot *et al.* [50] argue that autoencoder architectures employing LSTM and VAE inadequately account for temporal dependencies and assume a Gaussian distribution, resulting in suboptimal performance. Their proposed contextual autoencoder (CAE) employs a single encoder to capture time-dependent patterns and multiple decoders for specific flight phases. The CAE learns normal patterns and calculates anomaly scores for time windows, establishing thresholds based on the 3-sigma rule to distinguish between normal and anomalous data.

Luo *et al.* [51] propose another context-aware architecture using a transformer for sequence reconstruction in their TTSAD model. This model comprises three modules: the temporal convolutional network (TCN) prediction module, which predicts the next value using temporal correlations; the transformer reconstruction module, which reconstructs the sequence to capture long-range dependencies; and the SVDD threshold determination module, which compares reconstructed to real data to detect anomalies.

In general, the F1-scores for detecting subtle, gradual attacks have increased across successive studies. For instance, [50] documented scores of 0.886 with LSTM-AE, 0.926 with VAE-SVDD, and 0.939 with CAE for the velocity drift attack, whereas TTSAD achieved a score of 0.94 in the same context. These results underscore two significant insights: (1) context-aware architectures currently offer the best performance for detecting sophisticated ADS-B attacks, such as gradual drifts, and (2) despite these gains, the performance of the context-aware approaches proposed in the state-of-the-art remains insufficient for safety-critical systems such as ADS-B.

Among context-aware models, transformers show promise, as evidenced by TTSAD results. However, their quadratic complexity with sequence length limits scalability, prompting interest in alternatives such as extended LSTM (xLSTM), which enhances LSTM while potentially addressing transformer limitations. In this study, we evaluate the effectiveness of the xLSTM architecture in detecting subtle gradual attacks on ADS-B and compare its performance with that of a transformer-based model; specifically, we use the encoder component of the transformer. Our work differs from previous studies in two ways. First, we implement an IDS

using the xLSTM architecture, marking its first reported application to ADS-B intrusion detection. Second, we apply transfer learning to train both xLSTM and transformer-based models. Both models are pre-trained to capture normal system behavior, and this knowledge is then transferred to the anomaly detection task to better distinguish benign from malicious activities [125, 126]. By leveraging these pre-trained representations, the IDS systems can improve generalization and enhance their ability to detect novel or previously unseen attacks [112].

7.4 Background on xLSTM

A background on the xLSTM architecture [6] is provided to support understanding of the proposed approach; readers already familiar with this architecture may skip this section without loss of continuity.

The xLSTM architecture has been designed to elevate the sequence modeling capabilities of LSTM through two main innovations, as depicted in Fig.7.1: exponential gating and new memory structures. Exponential gating enhances the control of information flow within the network by employing more adaptable and stable gating mechanisms, thereby strengthening the ability of the xLSTM model to process and retain relevant data.

Furthermore, xLSTM introduces two advanced memory cell types: scalar LSTM (sLSTM), which employs a refined scalar-based memory update and mixing strategy, and matrix LSTM (mLSTM), which arranges memory cells into matrices, enabling parallel computation and a covariance-based update rule. This matrix-based approach not only expands the memory capacity but also improves the handling of long-range dependencies and complex data patterns. These innovations are embedded within residual block backbones, referred to as xLSTM blocks, and are stacked to form deep xLSTM architectures.

7.5 Methodology

In this section, we outline the procedure for implementing the proposed IDS for ADS-B, followed by a detailed explanation of how the datasets were constructed and attacks were injected into them to train and evaluate the proposed models.

7.5.1 IDS Implementation

To implement the IDS for ADS-B, we adopted a three-step methodology: pre-training, fine-tuning, and multiclass classification.

First, deep learning models were pre-trained in an unsupervised manner to improve their ability to generalize across diverse ADS-B message sequences (*ergo* diverse flight trajectories).

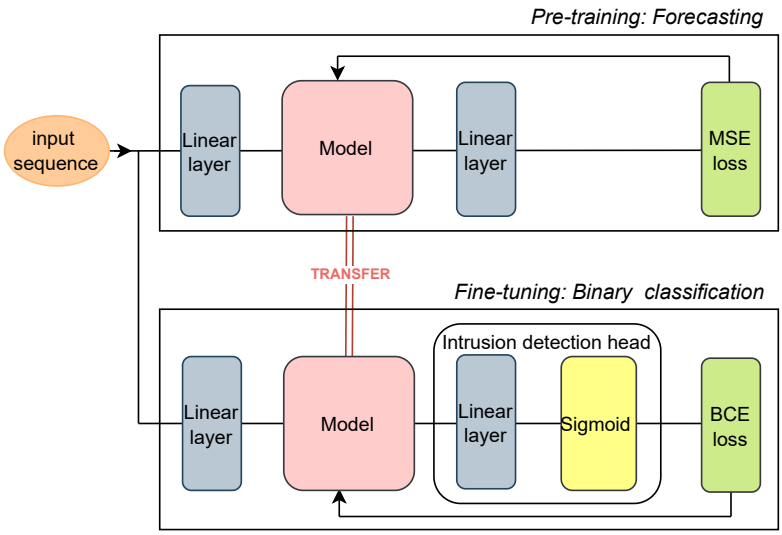


Figure 7.2 Methodology for pre-training and fine-tuning. Models are first pre-trained to predict future ADS-B messages by minimizing the mean squared error (MSE) loss. They are then fine-tuned using transfer learning for binary classification tasks, learning to distinguish between benign and malicious traffic by minimizing the binary cross-entropy (BCE) loss.

This step enabled the model to learn the contextual patterns of ADS-B communications, which is essential for detecting anomalous behavior in dynamic airspace environments. As illustrated in Fig. 7.2, the input sequence underwent a linear transformation before being passed to the core model. The model was then trained to perform a forecasting task, where it predicted future message values. The prediction passed through another linear layer, and the training objective was to minimize the mean squared error (MSE) loss. As demonstrated in prior studies, unsupervised pre-training positions deep architectures within favorable regions of the parameter space, leading to improved convergence and generalization during supervised learning [125].

Second, the pre-trained models were fine-tuned in a supervised manner on specific binary classification tasks, each aimed at detecting a particular class of ADS-B attack. In this phase, we adopted a transfer learning approach, where the model to be fine-tuned was initialized with pre-trained weights and retained the same architecture. As shown in Fig. 7.2, the input sequence followed a similar preprocessing path to that used during the pre-training phase; however, the output was now fed into an intrusion detection head consisting of a linear layer and a sigmoid activation. The model was then trained to classify traffic as either benign or malicious by minimizing a binary cross-entropy (BCE) loss, which measured the discrepancy between the predicted probabilities and the true labels. This process allowed each fine-tuned

model (or binary classifier) to specialize in recognizing the characteristics of a specific attack type.

Finally, we integrated the fine-tuned models into a multiclass classifier capable of simultaneously detecting and categorizing different ADS-B intrusions. Indeed, this final step is crucial for real-world applicability, where network traffic is subject to a variety of intrusion types. By enabling fine-grained threat identification, the multiclass approach supports timely and targeted mitigation strategies, which are vital for maintaining the integrity and safety of air traffic surveillance systems.

7.5.2 Data Acquisition and Dataset Implementation

In this work, we used state vector data collected by the OpenSky Network [162], a community-based receiver network that continuously gathers air traffic surveillance data for research purposes. State vectors provide an abstraction of tracking information. This data, available in 10-second update intervals, is derived from ADS-B and Mode S messages, offering a summary of the state of an aircraft at a given moment.

We constructed three datasets: Dataset A, Dataset B, and Dataset C, each of which corresponded to a different day and time of data collection to reflect varying flight and navigation conditions. Each dataset was used at a different stage of our methodology: Dataset A for unsupervised pre-training (forecasting), Dataset B for supervised fine-tuning (binary classification), and Dataset C for multiclass classification. The following paragraphs describe the dataset construction process.

Initially, each dataset contained a mix of ADS-B messages from multiple flights. To ensure flight-level coherence, we grouped messages by flight identifier (*callsign*) so that each message sequence corresponded to a single flight. We then discarded flights with missing or incomplete data. Finally, we removed unnecessary fields and retained only the most relevant features for our use case. Namely, aircraft ICAO ID, latitude, longitude, groundspeed, heading, vertical rate, and altitude.

Then, we introduced gradual attacks in Datasets B and C. We focused on this category of attacks because they represent a subtler and more dangerous threat model than abrupt or disruptive attacks (e.g., jamming or replay). While existing models are generally effective at detecting high-noise disruptions, they often fail to detect low-profile message injection attacks that gradually alter flight parameters. Such attacks that may go unnoticed by human operators could have severe consequences.

In a gradual attack, a specific ADS-B message feature is modified incrementally over time: the first message is altered by Δx, the second by $2\Delta x$, the third by $3\Delta x$, and so forth. In our implementation, we applied three gradual attack types: +82 feet per message on altitude,

+1.9 knots per message on groundspeed, and +1 degree per message on heading.
For binary classification, Dataset B was prepared using a one-vs-rest (OvR) strategy. After dividing the dataset into training and test sets, four distinct subsets were derived from each split: *altitude-vs-rest*, *groundspeed-vs-rest*, *heading-vs-rest*, and *benign-vs-rest*. In each subset, 50% of the flights were subjected to a gradual attack on the target feature and labeled as 1, while the remaining 50%—including flights affected by other types of attacks or containing only genuine messages—were labeled as 0. This setup enabled each binary classifier to focus on distinguishing a specific attack type from all other conditions, laying the groundwork for the final multiclass classification.
The strategy for constructing Dataset C involved applying each of the previously defined gradual attacks to a portion of the flights while leaving others unaltered. Each class—altitude, groundspeed, heading, and benign—was assigned a unique label. Care was taken to balance the number of samples across all classes to prevent bias during the multiclass training process.

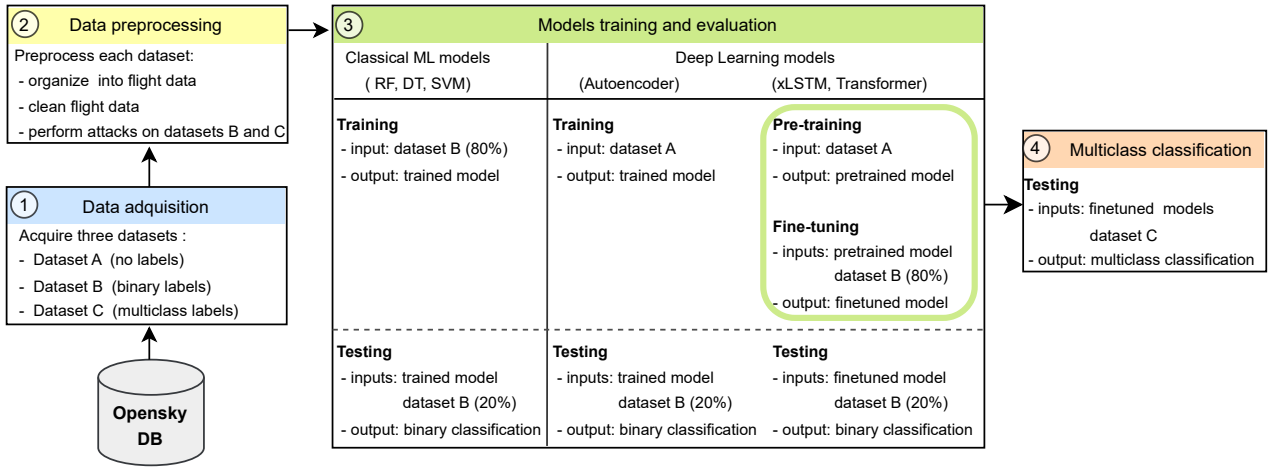


Figure 7.3 Overview of the experimental methodology. Dataset A contains genuine data; Dataset B includes binary-labeled genuine and tampered data; Dataset C has multiclass labels. Classical ML models are trained on Dataset B. The autoencoder is trained on Dataset A and tested on B using reconstruction error. xLSTM and transformer models are pre-trained on A and fine-tuned on B. An ensemble of the fine-tuned models performs multiclass classification on Dataset C.

7.6 Experiments

Here, we describe the experimental procedure and explain the evaluation methodology used to assess the performance of the proposed IDS. We also describe the hyperparameter optimization process.

7.6.1 Experimental Setup

Fig. 7.3 illustrates the experimental setup used to implement both the xLSTM-based and transformer-based IDS. The process started with the acquisition of three datasets from the OpenSky Network [162]. Dataset A contained benign, unlabeled ADS-B messages. Dataset B is a hybrid dataset with binary labels indicating either attack or benign traffic. Dataset C is also hybrid, but labeled for multiclass classification, with each attack assigned a distinct label. After the acquisition, the data were preprocessed. This includes structuring the messages into flight sequences and injecting simulated attacks into Datasets B and C, as explained in Section 7.5.2. The training pipeline for the xLSTM and transformer models was organized into three main stages.

First, the models were pre-trained on Dataset A using a sequence prediction task. The goal is to predict the next ADS-B message based on the sequence of previous messages. This step helps the models capture the temporal dependencies in benign traffic sequences. Next, the pre-trained models were fine-tuned on Dataset B for binary classification. Four separate binary classifiers were trained. Each is specialized for detecting a specific attack or recognizing benign traffic. Dataset B was divided into 80% for training and 20% for testing. Finally, a multiclass classifier was built and tested on Dataset C. This dataset includes all attack types, each of which is labeled with a distinct class. When a new sequence is received, it passes through all four binary classifiers. The prediction with the highest probability is selected as the final output of the multiclass classifier.

In parallel, we trained additional models for performance comparisons. These models serve as benchmarks for xLSTM and transformer-based IDS. We selected three classical machine learning algorithms: Random Forest (RF [163]), decision tree (DT [164]), and support vector machine (SVM [123]). Each was trained on 80% of Dataset B and tested on the remaining 20% for binary classification. In addition, we trained an autoencoder-based model. It uses Dataset A and follows a forecasting objective similar to the pre-training step of our deep learning models. During the testing, 20% of Dataset B was used. Sequences were classified based on the reconstruction error.

7.6.2 Performance Evaluation Metrics

We assessed the effectiveness of our IDS across multiple attack scenarios using the confusion matrix, which provides the basis for five key evaluation values:

- **True positive (TP)**: Malicious messages correctly identified as intrusions.
- **False positive (FP)**: Benign messages incorrectly classified as intrusions.
- **True negative (TN)**: Benign messages correctly identified as non-intrusions.

Table 7.1 Best hyperparameter configurations for pre-trained models

Category	Hyperparameter	xLSTM Model	Transfomer Model
Optimizer	Optimizer	Adam	Adam
	Learning rate	8.4×10^{-4}	1.3×10^{-4}
Model	Embedding size	64	64
	Number of heads	1	1
	Number of blocks	4	–
	Encoder layers	–	4
	slstm block at	1	–
	Dropout	–	0.005

Table 7.2 Best hyperparameter configurations for fine-tuned models

Category	Hyperparameter	xLSTM Models				Transformer Models			
		ALT	**GS**	**HDG**	**GN**	**ALT**	**GS**	**HDG**	**GN**
General	Epochs	5	10	10	15	15	10	10	15
	Batch size	50	40	50	30	50	40	40	30
	Sequence length	50	50	50	50	50	50	20*	50
Optimizer	Learning rate	6×10^{-5}	2×10^{-4}	5×10^{-5}	1×10^{-4}	8.5×10^{-5}	1.5×10^{-5}	4×10^{-4}	1×10^{-4}
Model	Dropout	–	–	–	–	0.14	0.056	0.028	0.24

- **False negative (FN)**: Malicious messages that were not detected as intrusions.

These values enable the computation of several standard performance metrics, which collectively offer a comprehensive view of the behavior of the models:

- **Precision** measures the fraction of correctly detected intrusions among all intrusion predictions.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7.1}$$

- **Recall**, also referred to as the true positive rate (TPR), indicates the fraction of actual intrusions that were correctly identified.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7.2}$$

- **F1-score** is the harmonic mean of precision and recall. A high F1-score reflects the ability of the model to accurately detect intrusions while maintaining a low rate of false alarms.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7.3}$$

- **False alarm rate (FAR)**, also known as the false positive rate (FPR), reflects the proportion of benign messages mistakenly flagged as intrusions. This metric is especially critical for intrusion detection in aviation systems, as excessive false alarms can overwhelm operators and compromise decision-making.

$$\text{FAR} = \frac{FP}{FP + TN} \tag{7.4}$$

By considering these metrics together, we obtained a well-rounded evaluation of the ability of our IDS to accurately detect tampered ADS-B messages while minimizing false alarms.
In addition to the detection performance, we also report the inference time of the IDS, defined as the time taken by the IDS to classify a message. It is a key factor in real-world deployments, particularly in time-sensitive environments, such as air traffic surveillance. Although our implementation was not specifically optimized for speed, the same experimental script was used across all the models to ensure fairness. The only variable that changed between the runs was the model being evaluated.

7.6.3 Hyperparameter Optimization

Tables 7.1 and 7.2 present the optimal hyperparameter configurations obtained for the pre-training and fine-tuning phases, respectively. These configurations were derived by following the experimental protocol described in Subsection 7.6.1 and by using the Optuna hyperparameter optimization framework [165]. Optuna leverages Bayesian optimization techniques to explore the hyperparameter space efficiently and identify high-performing combinations. The optimization process was performed on the training sets, with 80% of the data used for training and 20% reserved for validation.
In both tables, the hyperparameters are grouped into three main categories: training and evaluation-related parameters (*general*), optimizer-related parameters (*optimizer*), and architecture-specific parameters (*model*). During the pre-training phase, the search focused on identifying the best model and optimizer hyperparameters, whereas general parameters such as batch size (32), sequence length (10), and number of epochs (20) were manually set. As shown in Table 7.1, the search led to comparable architectural choices for both the xLSTM and transformer models, particularly in terms of the embedding dimension and attention heads.
In the fine-tuning phase, model-specific hyperparameters identified during pre-training were reused, and the search concentrated on optimizing general parameters. As illustrated in

Table 7.2, the optimal sequence length was 50 for all models except the transformer HDG (heading) model, which achieved the best results with a sequence length of 20. However, to ensure consistency during the subsequent multiclass classification task, all the models were fine-tuned using a sequence length of 50.

7.7 Results

In this section, we present the experimental results. We begin by comparing the performance of classical machine learning and deep learning models in distinguishing between genuine and tampered ADS-B messages in a binary classification task. Next, we assess the effectiveness of xLSTM- and transformer-based classifiers, particularly after they have been fine-tuned for specific types of attacks. We then examine how well these models adapt to unknown attacks, meaning attacks that the model was not trained to recognize. Finally, we analyze the inference time of the model and its impact on the situational awareness of controllers.

Table 7.3 Performance results for the binary classification task consisting of distinguishing between genuine and tampered messages.

Metric	**SVM**	**DT**	**RF**	**AE**	**Tx**	**xLSTM**
Accuracy	0.649	0.854	0.888	0.893	0.919	0.982
Precision	0.613	0.856	0.881	0.890	0.913	0.980
Recall	0.811	0.852	0.897	0.901	0.926	0.984
F1-score	0.698	0.854	0.889	0.891	0.920	0.982
FPR	0.511	0.143	0.119	0.012	0.087	0.018
FNR	0.189	0.147	0.102	0.099	0.074	0.016

7.7.1 Binary Classification: Classical vs Deep Learning Models

Table 7.3 and Fig. 7.4 show the outcomes of binary classification, where the models are tasked with distinguishing genuine from anomalous ADS-B messages. Deep learning models, particularly the xLSTM and transformer, consistently achieved superior scores across all evaluation metrics. For example, xLSTM achieves a precision of 0.980, a recall of 0.984, and an F1-score of 0.982, whereas the transformer records less impressive but still commendable values of 0.913, 0.926, and 0.920, respectively. Both models, along with the autoencoder, exhibited low false positive rates (FPR). The autoencoder achieved the lowest FPR at 0.012, followed by xLSTM at 0.018, and transformer at 0.087. These findings suggest that deep learning models, particularly xLSTM, are highly robust in identifying sophisticated or stealthy attacks while minimizing false alarms.

The classical machine learning model, Random Forest (RF), also demonstrated respectable performance. With a precision of 0.881 and an F1-score of 0.889, Random Forest appears well

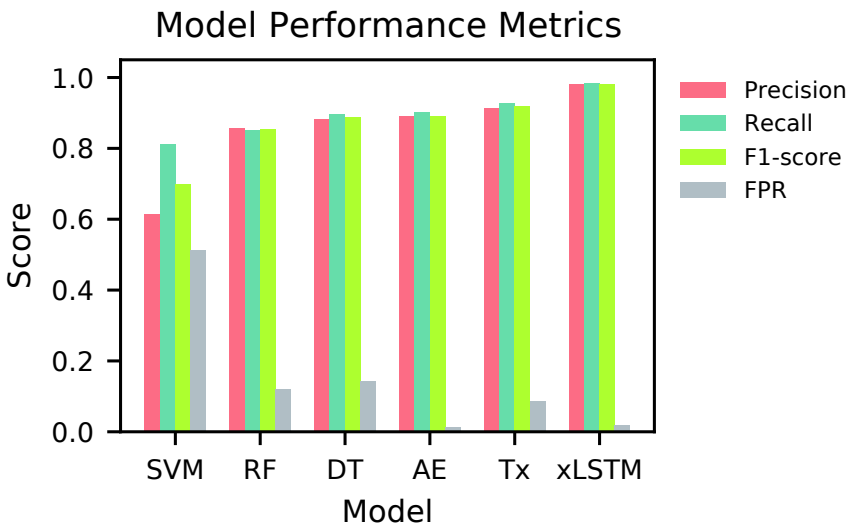


Figure 7.4 Comparison of performance metrics across six classifiers applied to ADS-B intrusion detection. The xLSTM and transformer models consistently outperform traditional methods, while the SVM exhibits the highest false positive rate.

suited to scenarios where the nature of attacks is more static or where clear distinctions exist between normal and abnormal patterns. The relatively low FPR of 0.119 further supported this observation. However, not all classical models perform equally well in this regard. For example, the support vector machine (SVM) recorded a significantly higher FPR of 0.511, which may lead to an unmanageable number of false alarms in practice.

These results underscore the strengths of deep learning models in handling complex and nuanced attack scenarios, particularly when the boundary between normal and malicious behavior is subtle. At the same time, they acknowledge the continued relevance of classical machine learning models in more controlled or well-characterized environments. This performance gap ultimately justifies our choice of adopting deep learning architectures for the implementation of the IDS.

7.7.2 Fine-Tuning and Multiclass Classification

Following the binary classification results, we constructed multiclass classifiers using the xLSTM and transformer models. This involved three steps: (1) pre-training the models on genuine ADS-B data, (2) fine-tuning them on labeled samples of specific attacks, and (3) implementing a multiclass classifier using the fine-tuned models.

Table 7.4 presents the performance of the xLSTM and transformer (Tx) models after fine-tuning. xLSTM consistently outperformed the transformer across all four binary classifiers,

Table 7.4 Performance results of the four fine-tuned binary classifiers implemented.

Model	Classifier	Target class	Accuracy	Precision	Recall	F1-score	FPR	FNR
xLSTM	ALT	altitude	0.995	0.994	0.996	0.996	0.006	0.004
	GS	groundspeed	0.989	0.987	0.990	0.989	0.013	0.010
	HDG	heading	0.993	0.995	0.990	0.993	0.005	0.010
	BN	benign	0.982	0.980	0.984	0.982	0.018	0.016
Tx	ALT	altitude	0.980	0.979	0.982	0.981	0.021	0.018
	GS	groundspeed	0.987	0.987	0.998	0.987	0.012	0.002
	HDG	heading	0.966	0.960	0.972	0.966	0.040	0.028
	BN	benign	0.919	0.913	0.926	0.920	0.087	0.074

Table 7.5 Performance results of the multiclass classifier when evaluated on unseen data containing known attacks.

Metric	xLSTM	Transformer
Accuracy	0.989	0.9432
Precision	0.988	0.9434
Recall	0.990	0.9432
F1-score	0.989	0.9433
FPR	0.012	0.056
FNR	0.010	0.056
Time (s)	7.26	2.1

achieving high accuracy, precision, recall, and F1-scores, with low false positives and false negatives. Notably, xLSTM achieves an F1-score of 0.982 for benign (BN) samples, indicating a reliable discrimination between normal and malicious behaviors. In contrast, the transformer shows a drop in performance for benign messages, with an F1-score of 0.920 and higher error rates. These results suggest that while both models can capture subtle anomalies, xLSTM is more robust, particularly in identifying benign traffic.

These results highlight the effectiveness of the pretraining and fine-tuning approach for intrusion detection. In [50], the authors report F1-scores of 0.886 for LSTM-AE [161], 0.926 for VAE-SVDD [122], and 0.939 for CAE [50] when detecting velocity drift attacks. In comparison, the TTSAD [122] method achieves a slightly higher score of 0.94 under the same conditions. In our study, we refer to the variable called velocity in previous works as ground speed (GS). Focusing on the results of the GS classifier in Table 7.4, the models based on xLSTM and transformers achieve F1-scores of 0.989 and 0.987, respectively, outperforming previous studies. These findings further confirm the value of combining pretraining with targeted fine-tuning to improve detection performance in ADS-B intrusion detection.

Table 7.5 lists the results of full multiclass classification. xLSTM achieves an accuracy of 0.989 and an F1-score of 0.989, maintaining its superior performance. However, the transformer

Table 7.6 Performance results of the multiclass classifier when evaluated on unseen data containing unknown attacks.

Metric	**xLSTM**	**Transformer**
Accuracy	0.911	0.840
Precision	0.920	0.853
Recall	0.912	0.842
F1-score	0.910	0.832
FPR	0.036	0.055
FNR	0.056	0.080
Time (s)	7.49	2.1

suffers from higher FPR and FNR. This comparatively lower performance with respect to xLSTM suggests that further feature engineering or data preprocessing, such as the encoding embedding technique proposed by the authors of [112], may be beneficial to improve the classification capabilities of the transformer model.

7.7.3 Generalization to a Novel Attack

To evaluate the robustness of our multiclass classifiers or IDS, we introduced a new standing still attack that was not included during training. This attack sets the ground speed (velocity) of the aircraft to zero and freezes its position for a short period of time. Table 7.6 shows that the xLSTM-based IDS performs adequately, achieving an F1-score of 0.910 and correctly identifying the majority of samples from this previously unseen attack. In contrast, the transformer-based IDS struggles to generalize, with a sharp decline in F1-score.

These findings confirm the capacity of the xLSTM-based IDS to generalize to new threats, making it a reliable candidate for real-time anomaly detection in dynamic airspace environments.

7.7.4 Inference Time Analysis and System Performance

Incorporating a security mechanism into ADS-B, whether cryptographic or non-cryptographic, introduces a safety trade-off: messages are not validated instantly, resulting in an uncertainty delay between their reception and verification. In [55], the authors assess this delay by comparing it with the refresh time of Secondary Surveillance Radar (SSR) systems to understand how it could affect the situational awareness of air traffic controllers. Following this approach, we use the SSR refresh time as a reference point to evaluate the operational impact of the inference delays introduced by our IDS.

In ATC, radar systems are essential for tracking aircraft positions and maintaining a safe flight separation. ATC service integrates radar data with other surveillance sources, such

as ADS-B, to perform data fusion and build a more accurate and reliable picture of the airspace. Rotating radar systems, including SSR and certain Primary Surveillance Radars (PSR), typically operate at 5–12 revolutions per minute (RPM), yielding refresh intervals between 12 and 5 seconds.

As shown in Table 7.6, our xLSTM-based multiclass classifier introduces an uncertainty delay of 7.26 seconds. This means that the controllers must wait for more than 7 seconds after receiving an ADS-B message to assess its truthworthiness. In contrast, the transformer-based IDS significantly reduces this delay to approximately 2.1 seconds.

These uncertainty delays have different operational implications, depending on the airspace context. In airports and terminal areas, where controllers often have direct line-of-sight (LOS) to aircraft, longer verification delays may be partially mitigated through visual confirmation. However, controllers at area control centers (ACC), which manage en-route traffic without visual contact, depend entirely on sensor data and are therefore more exposed to the risks introduced by delayed message authentication.

Although the xLSTM-based IDS provides a higher detection rate, its longer uncertainty delay poses limitations in time-sensitive ATC environments. While the 7.26-second delay technically falls within the SSR refresh interval range, it is less suitable where faster decisions are critical. The transformer-based IDS, with its shorter delay, improves timeliness, but does not achieve the same detection performance. As such, the xLSTM model may still be viable in low-density or LOS-supported settings. Recent optimizations of the original xLSTM architecture [166] have aimed to reduce inference time, and future work should assess whether these updated models can preserve detection performance while improving responsiveness to modern ATC needs.

7.8 Conclusion

This study evaluates emerging solutions for implementing efficient intrusion detection systems (IDS) for ADS-B surveillance technology. We investigate two deep learning-based IDS implementations: a transformer architecture and an extended Long Short-Term Memory (xLSTM) model. Both models are trained using transfer learning to evaluate its effect on their performance and generalization, particularly in detecting subtle and previously unseen attacks. Results show that pretraining and fine-tuning improve detection rates. The xLSTM-based model outperforms the transformer-based model, especially in identifying benign traffic and generalizing to new threats, making it well-suited for real-time anomaly detection. These findings emphasize the importance of low-latency architectures for air traffic control decisions. Although the xLSTM-based IDS achieves higher detection rates, its 7.26-second delay limits its applicability in crowded environments, though it remains suit-

able for low-density settings with visual confirmation. In contrast, the transformer-based IDS offers shorter inference times but lower detection performance. Future research should explore recent xLSTM optimizations [166] to improve responsiveness while maintaining accuracy. Additionally, quantum-inspired algorithms may enhance computational efficiency and inference speed.

CHAPTER 8 CONCLUSION

8.1 Summary of Works

8.1.1 CABBA

In this study, we introduce CABBA (Compatible Authenticated Bandwidth-efficient Broadcast protocol for ADS-B) [55], a secure replacement for ADS-B. CABBA enhances the system by adding an authentication layer that combines the TESLA protocol with a Public Key Infrastructure. To meet the stringent operational requirements of ADS-B, we employed phase overlay modulation at the physical layer, allowing authentication data to be embedded in the signal carrier without disrupting standard transmissions. This design ensures message integrity, source authentication, and aircraft identity verification while maintaining the safety requirements of ADS-B concerning backward compatibility and bandwidth efficiency.

To evaluate backward compatibility with existing avionics, we set up a testbed using the HackRF One software-defined radio. Custom scripts were developed to broadcast standard ADS-B messages and CABBA-augmented messages. These transmissions were directed to two commercial ADS-B In receivers: the Appareo Stratus II, a non-certified, portable receiver commonly used in general aviation, and the Collins TSS-4100, a certified avionics system integrating TCAS and a transponder. This setup allowed us to verify whether legacy equipment could still decode standard ADS-B messages within CABBA packets.

Additionally, we leveraged our lab implementation of CABBA to evaluate its bit error rate (BER), focusing specifically on the quadrature component of the carrier, which carries the MAC and other security-related information. To model the CABBA communication link, we used Simulink [148] and employed the MATLAB tool `bertool` to measure the BER over an Additive White Gaussian Noise (AWGN) channel.

To further assess the impact of CABBA on bandwidth usage, we conducted a channel occupancy rate (COR) analysis of the 1090ES. We first established a baseline using real ADS-B traffic data obtained from the OpenSky Network database by selecting a receiver located near a high-traffic airport (Paris Orly). From this dataset, we extracted 24 hours of continuous traffic and segmented it into 1-hour blocks. Within each hour, we selected six random 30-second intervals. For each sampled interval, we computed the activity factor as the ratio of the total number of CABBA packets transmitted — covering packet types A, B1, B2, and C — to the interval duration, in line with ITU-R SM.2256-1 recommendations.

Furthermore, a safety impact assessment of unauthenticated messages was conducted to evaluate the effect of CABBA on the situational awareness of pilots and air traffic controllers.

This assessment involved estimating the time aircraft remained within line-of-sight (LOS) communication range before entering protection volumes in TCAS and controlled sectors in ATC. Using typical sector geometries, aircraft altitudes, and maximum closure speeds, the minimum LOS durations were calculated for each operational context. These durations were then compared against the expected reception times of CABBA packet types B1, B2, and C. Finally, a comprehensive protocol specification is presented, detailing the packet structure, modulation techniques, authentication processes, and decision-making logic for differentiating between genuine and spoofed traffic. This design facilitates reproducibility and lays the groundwork for standardization and future implementation.
Three research questions guided this study:

RQ1: To what extent can a TESLA-based authentication scheme be integrated into ADS-B while maintaining backward compatibility and ensuring efficient bandwidth utilization?

RQ2: Are the delays introduced by the proposed solution compliant with the timing thresholds defined by TCAS and consistent with the update delays observed in ATC surveillance systems?

RQ3: Is the bit error rate of the carrier's quadrature component, which conveys security information, within acceptable limits to guarantee that the proposed solution maintains overall quality of service?

In relation to RQ1, our tests confirm that CABBA maintains backward compatibility with the two ADS-B In receivers assessed. This outcome indicates that the integration of CABBA into existing systems would not compromise safety during the transitional phase, when certain aircraft may not yet be equipped with CABBA-capable receivers.
Furthermore, the COR analysis revealed that CABBA introduced only minimal communication overhead. When the channel occupancy ranges from 2 % to 10 %, even the most bandwidth-demanding configuration of CABBA results in an increase of less than 1 % in COR. In situations with greater congestion, where COR might theoretically reach 40 %, CABBA would only raise channel occupancy to just below 43 %, which is deemed an acceptable trade-off.
In addressing RQ2, the line-of-sight durations between aircraft—ranging from 13.9 to 28.4 minutes for ATC and up to 19.8 minutes for TCAS—ensure that B2 and C messages are received and authenticated within the acceptable timeframe. However, authentication delays may occur when B1 packets, which contain TESLA keys for authenticating Type A (ADS-B) messages, are either lost or not yet received. Our analysis indicates that these delays can extend up to 3 seconds for TCAS and 14 seconds for ATC. For TCAS, where response

times range from 15 to 48 seconds, such authentication delays have a negligible operational impact, as pilots still have time for avoidance maneuvers. Regarding ATC, although ATC systems operate with a 10-second radar refresh interval, the authentication delays associated with CABBA do not significantly compromise controllers' situational awareness. We argue that ATC relies on continuous trajectory updates rather than the integrity of an individual message. As most preceding messages are authenticated, controllers still maintain reliable situational awareness, even during these brief verification delays.

For RQ3, the BER analysis indicated that CABBA met the MOPS requirements when used with D8PSK for normalized signal-to-noise ratios (E_b/N_0) of 15 dB or higher. The target BER is less than 10^{-6}, and CABBA achieves a BER of 0 starting at $E_b/N_0 = 15$ dB.

8.1.2 xLSTM-based IDS

As part of this project, we designed a deep learning-based intrusion detection system (IDS) built on the novel xLSTM architecture, which leverages exponential gating and matrix-valued memory cells to better capture long-term dependencies, while maintaining fast, stepwise inference through linear scaling with sequence length. The model was trained using transfer learning, leveraging pre-trained temporal patterns to accelerate convergence and enhance generalization to previously unseen attack patterns with ADS-B messages sourced from the OpenSky Network database.

During pre-training, only the original, unaltered messages were used to ensure that the xLSTM model learned normal communication patterns. For fine-tuning and testing, a hybrid dataset comprising authentic and tampered messages was employed, allowing for the evaluation of the xLSTM-based IDS's capability to detect both known and previously unseen attacks.

To benchmark its performance, we trained several alternative models. These included three classical machine learning algorithms: Random Forest (RF) [163], Decision Tree (DT) [164], and Support Vector Machine (SVM) [123], as well as two deep learning-based approaches: an LSTM-based autoencoder and a transformer-based classifier using the encoder portion of the architecture. These models were selected to reflect the main approaches in the literature. We first evaluated all the models in a binary classification setting, distinguishing between genuine and malicious ADS-B messages. This initial comparison verified the commonly reported superiority of deep learning over classical machine learning for ADS-B intrusion detection. It also highlighted the limited effectiveness of LSTM-based autoencoders in detecting subtle or complex intrusions, particularly when compared with more context-aware architectures.

Subsequently, we assessed the xLSTM and Transformer-based IDS in a multiclass classification setup. Both models were trained using an one-vs-rest (OvR) strategy, in which a series

of binary classifiers were employed to distinguish each attack type or benign traffic from all other classes. The Transformer-based model, such as xLSTM, was trained using transfer learning to improve generalization. This evaluation enabled a more fine-grained analysis of each model's ability to recognize specific attack patterns, differentiate them from normal traffic, and even identify previously unseen attack behaviors.
Finally, we measured the inference time of both IDS implementations and compared it to the typical refresh interval of commercial SSR systems, in order to assess whether the detection latency could impact real-time situational awareness for air traffic controllers.
One research question guided this study:

RQ4: What are the most effective architectures and learning strategies for designing and developing innovative, high-performance IDS solutions for ADS-B?

The results indicate that the xLSTM-based IDS outperforms the benchmark models in terms of detection performance but falls short in terms of inference delay.
In the multiclass classification setting, the xLSTM-based model achieved an F1-score of 98,9 %, compared with 94,3 % for the transformer-based model. It also demonstrated higher accuracy in identifying benign traffic (98,2 % vs. 91,9 % F1-score), thereby reducing false positives, which is a key factor for maintaining operator trust and ensuring that attention remains focused on genuine threats. Additionally, the xLSTM-based IDS showed improved detection of previously unseen attack patterns, highlighting its superior adaptability to evolving threats, an essential capability given the continuous emergence of novel intrusion strategies.
However, these performance gains come at the cost of increased latency. The xLSTM-based IDS exhibited a significantly higher inference delay of 7.26 s compared to just 2.1 s for the transformer-based model. This delay raises concerns about its suitability for time-critical surveillance applications, where timely threat detection is crucial for maintaining controller situational awareness.
Besides, although the xLSTM-based IDS demonstrates superior detection capabilities, its 7.26-second inference delay may hinder its deployment in high-density or time-sensitive airspace, where rapid anomaly detection is critical. However, this level of latency could be acceptable in lower-density environments or in scenarios with line-of-sight (LOS) support, where delayed verification can be supplemented by visual confirmation of the results.

8.2 Limitations

8.2.1 CABBA

Initial tests confirmed CABBA's backward compatibility with legacy receivers from two vendors, Appareo and Collins, suggesting a promising level of interoperability. However, these

preliminary results alone are insufficient to guarantee broad backward compatibility across the diverse range of ADS-B In receivers that are currently in use. Further testing in controlled environments is necessary to ensure that various types of receivers, featuring different software and hardware, can accurately parse CABBA messages.

Moreover, while the bit error rate (BER) was assessed through simulations using MATLAB and Simulink, these theoretical models fall short of fully capturing the complexity of real-world signal propagation. In-flight tests are crucial for accurately evaluating the robustness of CABBA, particularly in environments with high ADS-B traffic and potential interference sources, such as multipath effects from terrain (e.g., mountains, water surfaces) or urban structures. Such testing would enable us to assess the transmission reliability under realistic conditions, offering a clearer view of CABBA's quality of service of CABBA.

One limitation of our current solution is that, although the receiver logic was fully designed, it has not yet been implemented. Developing a prototype CABBA receiver would enable precise latency measurements and allow for a more accurate estimation of its impact on the situational awareness of pilots and controllers in the future. It would also support stability testing under stress conditions, such as continuous message transmission over long durations, to assess system robustness and determine the point at which degradation or failure might occur. Additionally, such testing would help determine how the receiver handles the long-term storage of TESLA keys and certificates from multiple aircraft and whether this could lead to excessive memory use or saturation.

Finally, a fundamental limitation of CABBA is its reliance on a PKI for proper authentication. Currently, no PKI exists within the aviation domain for authenticating and distributing aircraft public keys. While we consider the development of such infrastructure essential for enabling secure ADS-B communications in the long term, we recognize that the current lack of a standardized PKI is a major barrier to CABBA's near-term deployment.

8.2.2 xLSTM-based IDS

The xLSTM model exhibited an inference latency of 7.26 seconds, which is notably high for safety-critical scenarios requiring real-time responsiveness. This latency is primarily attributed to the use of the vanilla backend for executing the sLSTM block during experimentation. The vanilla backend relies on standard PyTorch operations (e.g., Linear, matmul, LayerNorm), which, although GPU-compatible, are not as performance-optimized as the CUDA backend that leverages fused kernels for faster execution. Although the model and its stacked blocks were correctly transferred to the GPU, we were unable to activate the CUDA backend for sLSTM due to a runtime error. At the time of testing, the root cause of this issue remained unclear, and no official documentation provided a solution or a workaround.

It was only after the experimentation phase that the problem was diagnosed by the developers [1], who identified it as an architecture mismatch in the CUDA kernel compilation—specifically related to unsupported GPU features such as bfloat16. A fix was later documented, which involved the manual specification of the TORCH_CUDA_ARCH_LIST environment variable to ensure compatibility. Consequently, the inability to use the optimized CUDA backend during experimentation significantly affected the model inference speed.

A further limitation is the interpretation of our findings from a safety perspective. While the detection capabilities of our models show promise, it remains uncertain whether these results align with safety standards. At present, there is no defined benchmark for what constitutes a "safe" level of F1-score, accuracy, precision, or false positive rate in the realm of aviation anomaly detection systems.

This concern was raised during discussions with FAA representatives. They explained that, at this point, AI/ML practitioners in the aviation industry are still trying to understand these metrics. Standard organizations have not yet reached the stage of establishing safety thresholds for these metrics. One reason is that the thresholds depend on the specific functionality and how the system is integrated into a larger system or aircraft for failure analysis, where stringent requirements apply (for example, under DAL-A, the failure rate must be less than 10^{-9}). Currently, their approach is to ask developers to put the metrics on the table for consideration on a case-by-case basis.

Until standardized evaluation criteria are established, the interpretation of model performance requires caution. Further validation is necessary to evaluate how the model's outputs impact the operational risk of ATC service, specifically in terms of increasing or reducing that risk. In other words, it is crucial to understand how the IDS will influence the decision-making process of air traffic controllers, as well as the behavior and reliability of the broader system into which it is integrated.

8.3 Future Research

Future research should address the following key areas:

1. **Extended backward compatibility and interoperability tests for CABBA**
   Broaden backward compatibility and interoperability assessments by testing CABBA across a wider range of legacy ADS-B In receivers and avionics systems from multiple vendors.

2. **Development of a CABBA prototype receiver**

[1] https://github.com/NX-AI/xlstm/issues/74

Develop and implement a functional receiver to facilitate accurate real-world latency measurements. This prototype should also serve to evaluate the robustness of the system during continuous long-duration message transmissions and under high-traffic conditions.

3. **Optimize xLSTM-based IDS inference speed**
   Address the CUDA kernel compatibility issue hindering sLSTM acceleration. If latency remains problematic, investigate alternative architectures and optimization strategies to meet real-time operational requirements.

4. **Definition of safety and performance thresholds for aviation AI-based IDS**
   Collaborate with aviation stakeholders to define clear benchmarks and acceptable limits for detection accuracy and latency in operational airspace environments.

REFERENCES

[1] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *International conference on the theory and application of cryptology and information security*. Springer, 2003, pp. 452–473.

[2] T. Canada, "Advisory Circular AC No. 500-029: Certification of Automatic Dependent Surveillance – Broadcast (ADS-B)," Online advisory circular, 2024, accessed: 2025-08-10. [Online]. Available: https://tc.canada.ca/en/aviation/reference-centre/advisory-circulars/advisory-circular-ac-no-500-029

[3] NIST, "Digital Signature Standard (DSS)," U.S. Department of Commerce, Gaithersburg, MD, Standard FIPS PUB 186-5, 2023. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf

[4] L. Chen, "An Interpretation of Identity-Based Cryptography," in *Foundations of Security Analysis and Design IV*. Springer, 2007, pp. 183–208.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[6] M. Beck, K. Pöppel, M. Spanring, A. Auer, O. Prudnikova, M. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter, "xLSTM: Extended Long Short-Term Memory," *Advances in Neural Information Processing Systems*, vol. 37, pp. 107 547–107 603, 2024.

[7] Federal Aviation Administration, "NextGen Background," https://www.faa.gov/nextgen/background, 2025, accessed August 7, 2025.

[8] P. Brooker, "SESAR and NextGen: Investing in new paradigms," *The Journal of Navigation*, vol. 61, no. 2, p. 195, 2008.

[9] "Next Generation Air Transportation System (NextGen)," Federal Aviation Administration (FAA), May 2025. [Online]. Available: https://www.faa.gov/nextgen

[10] S. Thompson, D. Spencer, and J. Andrews, "An Assessment of the Communications, Navigation, Surveillance (CNS) Capabilities Needed to Support the Future Air Traffic Management System," Massachusetts Institute of Technology, Lincoln Laboratory, Cambridge, MA, Technical report, 2001.

[11] Federal Aviation Administration (FAA). (2024, Jul.) Nextgen. [Online]. Available: https://www.faa.gov/newsroom/nextgen

[12] RTCA, "Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance – Broadcast (ADS-B) and Traffic Information Services – Broadcast (TIS-B)," Radio Technical Commission for Aeronautics, Washington, DC, Technical report DO-260C, 2020.

[13] EUROCAE, "Technical Specification for the ADS-B Ground Station," European Organisation for Civil Aviation Equipment, Paris, FR, Technical report ED-129, 2010.

[14] ——, "Technical Specification for a 1090 MHZ Extended Squitter ADS-B Ground Station," European Organisation for Civil Aviation Equipment, Paris, FR, Technical report ED-129, 2015.

[15] G. Sirigu, J. Dolan, and M. A. Garcia, "Independent Estimation of Aircraft Positions Using Space-Based ADS-B Data for GNSS Anomaly Identification and Investigation," in *2025 Integrated Communications, Navigation and Surveillance Conference (ICNS)*. IEEE, 2025, pp. 1–9.

[16] RTCA, "Minimum Operational Performance Standards for Airborne Collision Avoidance System X (ACAS X) (ACAS Xa and ACAS Xo) Volume I and II," Radio Technical Commission for Aeronautics, Washington, DC, Technical report DO-385A, 2023.

[17] ——, "Minimum Operational Performance Standards (MOPS) for Traffic Alert and Collision Avoidance System II (TCASII) Hybrid Surveillance," Radio Technical Commission for Aeronautics, Washington, DC, Technical report DO-300A, 2015.

[18] S. Hyang, M. Dong, and H. Chan, "Analysis of CNS/ATM technology trend," *Current Industrial and Technological Trends in Aerospace*, vol. 8, no. 2, pp. 113–123, 2010.

[19] R. Abeyratne, "Evolution from FANS to CNS/ATM and products liability of technology providers in the united states," *ZLW*, vol. 43, p. 156, 1994.

[20] R. Strain and D. Stapleton, "Defining aerodrome and airspace FIS-B products for delivery to aircraft via data link," in *Proceedings. The 21st Digital Avionics Systems Conference*, vol. 1. IEEE, 2002, pp. 3A5–3A5.

[21] NAV CANADA, "New ADS-B Mandate to Enhance Aircraft Operations in Canada," 2023, accessed July 22, 2025. [Online]. Available: https://www.navcanada.ca/en/news/news-releases/new-ads-b-mandate-to-enhance-aircraft-operations-in-canada.aspx

[22] ——, "Notice of Change – Canadian ADS-B Out Performance Requirements Mandate," Aeronautical Information Circular (AIC 17/23), 2023, accessed July 22, 2025. [Online]. Available: https://realaviation.ca/wp-content/uploads/2023/11/ADS-B-OUT-TCCA-AIC-aiceng202317.pdf

[23] European Commission, "Commission Implementing Regulation (EU) No 1207/2011 on surveillance performance and interoperability requirements," 2011, accessed July 22, 2025. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32011R1207

[24] European Union Aviation Safety Agency (EASA), "ADS-B Certification FAQs," 2020, accessed July 22, 2025. [Online]. Available: https://www.easa.europa.eu/en/the-agency/faqs/ads-b-certification

[25] Federal Aviation Administration, "14 CFR §91.225 - Automatic Dependent Surveillance-Broadcast (ADS–B) Out equipment and use," 2020, accessed July 22, 2025. [Online]. Available: https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-91/subpart-C/section-91.225

[26] D. McCallie, J. Butts, and R. Mills, "Security analysis of the ADS-B implementation in the next generation air transportation system," *International Journal of Critical Infrastructure Protection*, vol. 4, no. 2, pp. 78–87, 2011.

[27] M. Strohmeier, V. Lenders, and I. Martinovic, "Security of ADS-B: State of the Art and Beyond." DCS, 2013.

[28] 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, no. 5, pp. 111–118, 2014.

[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, pp. 16–31, 2017.

[30] A. Costin and A. Francillon, "Ghost in the air (traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices," in *Proceedings of Black Hat USA*, 2012.

[31] B. Haines, "Hacker + Airplanes = No Good Can Come Of This," in *Presentation at DEFCON Hacking conference 20, July 26*, vol. 29, 2012.

[32] N. Marinos and H. Krause, "Aviation CYBERSECURITY FAA Should Fully Implement Key Practices to Strengthen Its Oversight of Avionics Risks," United States Government Accountability Office, Washington, DC, Rapport technique GAO-21-86, 2020.

[33] ICAO, "Surveillance of remotely piloted aircraft systems (RPAS) and cybersecurity," Technical Commission Russian Federation, Tech. Rep. A39-WP/296, 2016. [Online]. Available: https://www2023.icao.int/Meetings/a39/Documents/WP/wp_296_en.pdf

[34] NIST Computer Security Resource Center CSRC, "Message Authentication Codes MAC," 2024. [Online]. Available: https://csrc.nist.gov/Projects/message-authentication-codes

[35] ——, "Symmetric Cryptography," 2014, Glossary terms and definitions last updated: July 21, 2022. [Online]. Available: https://csrc.nist.gov/glossary/term/symmetric_cryptography

[36] M. Strohmeier, V. Lenders, and I. Martinovic, "On the Security of the Automatic Dependent Surveillance-Broadcast Protocol," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1066–1087, 2014.

[37] NIST Computer Security Resource Center CSRC, "asymmetric cryptography," 2024. [Online]. Available: https://csrc.nist.gov/glossary/term/asymmetric_cryptography

[38] NIST, "Recommendation for Key Management: *Part 2 – Best Practices for Key Management Organizations*," U.S. Department of Commerce, Gaithersburg, MD, Standard NIST SP 800-57 PT . 2 R EV . 1, 2019. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt2r1.pdf

[39] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," in *Workshop on the theory and application of cryptographic techniques*. Springer, 1984, pp. 47–53.

[40] A. Asari, M. R. Alagheband, M. Bayat, and M. R. Asaar, "A new provable hierarchical anonymous certificateless authentication protocol with aggregate verification in ADS-B systems," *Computer Networks*, vol. 185, p. 107599, 2021.

[41] M. H. Au, Y. Mu, J. Chen, D. S. Wong, J. K. Liu, and G. Yang, "Malicious KGC attacks in certificateless cryptography," in *Proceedings of the 2nd ACM symposium on Information, computer and communications security*, 2007, pp. 302–311.

[42] A. W. Dent, *Certificateless Cryptography*. Boston, MA: Springer US, 2011, pp. 192–193. [Online]. Available: https://doi.org/10.1007/978-1-4419-5906-5_314

[43] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The TESLA broadcast authentication protocol," *Rsa Cryptobytes*, vol. 5, no. 2, pp. 2–13, 2002.

[44] T. Yong, W. Honggang, X. Zhili, and H. Zhongtao, "ADS-B and SSR data fusion and application," in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 2. IEEE, 2012, pp. 255–258.

[45] Z. Yuan, Y. Lu, Z. Wang, and Y. Xue, "Droid-sec: deep learning in android malware detection," in *Proceedings of the 2014 ACM conference on SIGCOMM*, 2014, pp. 371–372.

[46] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, and S. Venkatraman, "Robust Intelligent Malware Detection Using Deep Learning," *IEEE access*, vol. 7, pp. 46 717–46 738, 2019.

[47] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019. [Online]. Available: https://arxiv.org/pdf/1901.03407

[48] L. Basora, X. Olive, and T. Dubot, "Recent advances in anomaly detection methods applied to aviation," *Aerospace*, vol. 6, no. 11, p. 117, 2019.

[49] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[50] A. Chevrot, A. Vernotte, and B. Legeard, "CAE: Contextual auto-encoder for multivariate time-series anomaly detection in air transportation," *Computers & Security*, vol. 116, p. 102652, 2022.

[51] P. Luo, B. Wang, and J. Tian, "TTSAD: TCN-Transformer-SVDD Model for Anomaly Detection in air traffic ADS-B data," *Computers & Security*, vol. 141, p. 103840, 2024.

[52] M. Noquer I Alonso, "The Mathematics of Sequence Architectures: Transformers, xLSTM, Titan, and Transformer-Squared," https://ssrn.com/abstract=5137903, 2025, sSRN Working Paper.

[53] T. Schmied, T. Adler, V. Patil, M. Beck, K. Pöppel, J. Brandstetter, G. Klambauer, R. Pascanu, and S. Hochreiter, "A Large Recurrent Action Model: xLSTM enables Fast Inference for Robotics Tasks," *arXiv preprint arXiv:2410.22391*, 2024.

[54] P. Haller, J. Golde, and A. Akbik, "Empirical Evaluation of Knowledge Distillation from Transformers to Subquadratic Language Models," *arXiv preprint arXiv:2504.14366*, 2025.

[55] M. Ngamboé, X. Niu, B. Joly, S. P. Biegler, P. Berthier, R. Benito, G. Rice, J. M. Fernandez, and G. Nicolescu, "CABBA: Compatible Authenticated Bandwidth-efficient Broadcast protocol for ADS-B," *International Journal of Critical Infrastructure Protection*, vol. 48, p. 100728, 2025.

[56] M. Ngamboé, J.-S. Marrocco, J.-Y. Ouattara, J. M. Fernandez, and G. Nicolescu, "New Machine Learning Approaches for Intrusion Detection in ADS-B," *arXiv preprint arXiv:2510.08333*, 2025.

[57] Federal Aviation Administration, "A Brief History of the FAA," https://www.faa.gov/about/history/brief_history, 2021, accessed: 2025-08-08.

[58] S. Golstein, "History of air traffic control," https://www.highlander.io/post/history-of-air-traffic-control, Aug 2023, updated Nov 6, 2023.

[59] Historic Croydon Airport Trust, "Genesis of Air Traffic Control," https://www.historiccroydonairport.org.uk/interesting-topics/air-traffic-control/, n.d., accessed: 2025-08-08.

[60] S. Flying, "History: The Story of the world's First Air Traffic Control Tower," Nov 2024, accessed: 2025-08-08. [Online]. Available: https://simpleflying.com/first-air-traffic-control-tower-london-history/

[61] W. contributors, "1922 Picardie mid-air collision," 2025, accessed: 2025-08-08. [Online]. Available: https://en.wikipedia.org/wiki/1922_Picardie_mid-air_collision

[62] Federal Aviation Administration, "Air Traffic Control," https://www.faa.gov/about/history/photo_album/air_traffic_control, 2021, accessed: 2025-08-08.

[63] M. Rotman, "Cleveland Hopkins International Airport," 2011, accessed: 2025-08-08. [Online]. Available: https://clevelandhistorical.org/items/show/150

[64] Civil Aeronautics Administration / Federal Aviation Administration, "When Radar Came to Town: Radar Departure Control Procedures," https://www.faa.gov/sites/faa.gov/files/about/history/milestones/radar_departure_control.pdf, n.d., accessed: 2025-08-08.

[65] J. Sun, *The 1090 Megahertz Riddle: A Guide to Decoding Mode S and ADS-B Signals*, 2nd ed. TU Delft OPEN Publishing, 2021.

[66] ELDIS Pardubice, s.r.o., "Rl-2000/mssr-1 collocated primary and secondary surveillance radar," https://www.eldis.cz/en/rl-2000-mssr-1, 2025, accessed: 2025-08-10.

[67] M. Lincoln Laboratory, "A GLOBAL MODE OF TRACKING AIRCRAFT," Online article by MIT Lincoln Laboratory, 2025, accessed: 2025-08-10. [Online]. Available: https://www.ll.mit.edu/impact/global-mode-tracking-aircraft

[68] Thales Group, "Iridium NEXT constellation built by Thales Alenia Space now completely deployed," Online press release, 2019, accessed: 2025-08-10. [Online]. Available: https://www.thalesgroup.com/en/worldwide/space/press-release/iridium r-next-constellation-built-thales-alenia-space-now-completely

[69] "Traffic collision avoidance system," 2005, accessed: 2025-08-08. [Online]. Available: https://en.wikipedia.org/wiki/Traffic_collision_avoidance_system

[70] A. Doug, "Future ADS-B applications," ICAO, Technical Report Technical On-Line Workshop for the NAM/CAR Regions (ADS-B/OUT/W), 2021. [Online]. Available: https://www.icao.int/NACC/Documents/Meetings/2021/ADSB/P05-FutureADS-B -ENG.pdf

[71] S. Gregory T., "Systems and methods for enhanced ATC overlay modulation," 20 Apr. 2016. [Online]. Available: https://patents.google.com/patent/EP2661039B1

[72] P. van Oorschot, *Cryptographic Building Blocks*. Springer International Publishing, 2020, pp. 29–53.

[73] K. Martin, *Cryptographic protocols*. Oxford University Press, 2017, pp. 325–368.

[74] E. Habler and A. Shabtai, "Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages," *Computers & Security*, vol. 78, pp. 155–173, 2018.

[75] Wikipedia contributors, "Iridium satellite constellation," https://en.wikipedia.org/w iki/Iridium_satellite_constellation, 2025, accessed: 2025-08-10.

[76] RTCA, "Minimum Aaviation System Performance Standards for Automatic Dependent Surveillance – Broadcast (ADS-B)," Radio Technical Commission for Aeronautics, Washington, DC, Technical report DO-242A, 2002.

[77] NIST Computer Security Resource Center CSRC, "secret key (symmetric) cryptographic algorithm," 2014, Glossary terms and definitions last updated: July 21, 2022. [Online]. Available: https://csrc.nist.gov/glossary/term/secret_key_cryptogra phic_algorithm

[78] M. Bellare, P. Rogaway, and T. Spies, "The FFX mode of operation for format-preserving encryption," *NIST submission*, vol. 20, p. 19, 2010.

[79] C. Finke, J. Butts, and R. Mills, "ADS-B encryption: confidentiality in the friendly skies," in *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*. ACM, 2013, pp. 1–4.

[80] C. Finke, J. Butts, R. Mills, and M. Grimaila, "Enhancing the security of aircraft surveillance in the next generation air traffic control system," *International Journal of Critical Infrastructure Protection*, vol. 6, no. 1, pp. 3–11, 2013.

[81] R. S. Huang, H. M. Yang, and H. G. Wu, "Enabling confidentiality for ADS-B broadcast messages based on format-preserving encryption," in *Applied Mechanics and Materials*, vol. 543. Trans Tech Publ, 2014, pp. 2032–2035.

[82] R. Agbeyibor, J. Butts, M. Grimaila, and R. Mills, "Evaluation of format-preserving encryption algorithms for critical infrastructure protection," in *International Conference on Critical Infrastructure Protection*. Springer, 2014, pp. 245–261.

[83] K. Samuelson, E. Valovage, and D. Hall, "Enhanced ADS-B research," in *2006 IEEE/AIAA 25TH Digital Avionics Systems Conference*, 2006, pp. 1–7.

[84] T. Kacem, D. Wijesekera, and P. Costa, "Integrity and authenticity of ADS-B broadcasts," in *2015 IEEE Aerospace Conference*. IEEE, 2015, pp. 1–8.

[85] M. Dworking, "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication," National Institute of Standards and Technology (NIST), Gaithersburg, MD, Technical Report NIST Special Publication (SP) 800-38B, 2016.

[86] M. Sönmez and L. Brandão, "Keyed-Hash Message Authentication Code (HMAC) : Specification of HMAC and Recommendations for Message Authentication," National Institute of Standards and Technology (NIST), Gaithersburg, MD, Technical Report NIST SP 800-224 ipd, 2024.

[87] J. Kelsey, S.-J. Chang, and R. Perlner, "SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash and ParallelHash," National Institute of Standards and Technology (NIST), Gaithersburg, MD, Technical Report NIST SP 800-185, 2016.

[88] Z. Feng, W. Pan, and Y. Wang, "A data authentication solution of ADS-B system based on X. 509 certificate," in *27th International Congress of the Aeronautical Sciences, ICAS*, 2010, pp. 1–6.

[89] A. K. Buchholz, "DPP: Dual Path PKI for Secure Aircraft Data Communication," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2013.

[90] J. Baek, Y.-J. Byon, E. Hableel, and M. Al-Qutayri, "An authentication framework for automatic dependent surveillance-broadcast based on online/offline identity-based signature," in *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. IEEE, 2013, pp. 358–363.

[91] H. Yang, H. Kim, H. Li, E. Yoon, X. Wang, and X. Ding, "An efficient broadcast authentication scheme with batch verification for ADS-B messages," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 7, no. 10, pp. 2544–2560, 2013.

[92] H. Yang, R. Huang, X. Wang, J. Deng, and R. Chen, "EBAA: An efficient broadcast authentication scheme for ADS-B communication based on IBS-MR," *Chinese Journal of Aeronautics*, vol. 27, no. 3, pp. 688–696, 2014.

[93] A. Yang, X. Tan, J. Baek, and D. S. Wong, "A new ADS-B authentication framework based on efficient hierarchical identity-based signature with batch verification," *IEEE Transactions on Services Computing*, vol. 10, no. 2, pp. 165–175, 2015.

[94] D. He, N. Kumar, K. K. R. Choo, and W. Wu, "Efficient Hierarchical Identity-Based Signature With Batch Verification for Automatic Dependent Surveillance-Broadcast System," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 454–464, 2016.

[95] G. Thumbur, N. Gayathri, P. V. Reddy, M. Z. U. Rahman *et al.*, "Efficient Pairing-Free Identity-Based ADS-B Authentication Scheme With Batch Verification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 5, pp. 2473–2486, 2019.

[96] 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, no. 3, pp. 1742–1753, 2019.

[97] A. Braeken, "Holistic Air Protection Scheme of ADS-B Communication," *IEEE Access*, vol. 7, pp. 65 251–65 262, 2019.

[98] J. Subramani, A. Maria, R. B. Neelakandan, and A. S. Rajasekaran, "Efficient anonymous authentication scheme for automatic dependent surveillance-broadcast system with batch verification," *IET Communications*, vol. 15, no. 9, pp. 1187–1197, 2021.

[99] NIST, "Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms," National Institute of Standards and Technology (NIST), Gaithersburg, MD, Technical Report NIST Special Publication (SP) 800-175B Rev. 1, 2020.

[100] H. C. Van Tilborg and S. Jajodia, Eds., *Encyclopedia of Cryptography and Security*, 2nd ed. New York, NY: Springer New York, 2011.

[101] F. Hess, "Efficient identity based signature schemes based on pairings," in *International workshop on selected areas in cryptography*. Springer, 2002, pp. 310–324.

[102] J. C. Choon and J. Hee Cheon, "An identity-based signature from gap Diffie-Hellman groups," in *2003 Public Key Cryptography—PKC 6th International Workshop on Practice and Theory in Public Key Cryptography*. Springer, 2002, pp. 18–30.

[103] P. S. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," in *2005 Advances in Cryptology-ASIACRYP 11th International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2005, pp. 515–532.

[104] *IEEE Standard for Identity-Based Cryptographic Techniques using Pairings*, IEEE Std. IEEE Std 1363.3-2013, 2013. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6662370

[105] A. Perrig and J. Tygar, "TESLA broadcast authentication," in *Secure Broadcast Communication*. Springer, 2003, pp. 29–53.

[106] P. Berthier, J. M. Fernandez, and J.-M. Robert, "SAT: Security in the Air using Tesla," in *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*. IEEE, 2017, pp. 1–10.

[107] S. Sciancalepore and R. Di Pietro, "SOS: Standard-compliant and packet loss tolerant security framework for ADS-Bcommunications," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 4, pp. 1681 –1698, 2019.

[108] H. Yang, M. Yao, Z. Xu, and B. Liu, "LHCSAS: A Lightweight and Highly-Compatible Solution for ADS-B Security," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–7.

[109] J. Habibi Markani, A. Amrhar, J.-M. Gagné, and R. J. Landry, "Security Establishment in ADS-B by Format-Preserving Encryption and Blockchain Schemes," *Applied Sciences*, vol. 13, no. 5, 2023.

[110] S. Sciancalepore and R. Di Pietro, "SOS-Securing Open Skies," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, 2018, pp. 15–32.

[111] S.-W. Lee, H. Mohammed sidqi, M. Mohammadi, R. Shima, , M. Amir, M. Mohammad, and H. Mehdi, "Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review," *Journal of Network and Computer Applications*, vol. 187, p. 103111, 2021.

[112] J.-S. Marrocco, "TRIPT-IDS: Triplet Loss Pre-trained Transformer for Avionic Intrusion Detection System," mémoire de maîtrise, Dép. de génie informatique et génie logiciel, École Polytechnique de Montréal, Montréal, QC, 2023. [Online]. Available: https://publications.polymtl.ca/56994/

[113] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[114] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological Cybernetics*, vol. 59, no. 4, pp. 291–294, 1988.

[115] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations (ICLR2014)*, 2014. [Online]. Available: https://doi.org/10.48550/arXiv.1312.6114

[116] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[117] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[118] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[119] S. Akerman, E. Habler, and A. Shabtai, "VizADS-B: Analyzing sequences of ADS-B images using explainable convolutional LSTM encoder-decoder to detect cyber attacks," *arXiv preprint arXiv:1906.07921*, 2019.

[120] X. Olive and L. Basora, "Identifying anomalies in past en-route trajectories with clustering and anomaly detection methods," in *ATM Seminar 2019*, 2019.

[121] A. Fried and M. Last, "Facing airborne attacks on ads-b data with autoencoders," *Computers & Security*, vol. 109, p. 102405, 2021.

[122] P. Luo, B. Wang, T. Li, and J. Tian, "ADS-B anomaly data detection model based on VAE-SVDD," *Computers & Security*, vol. 104, p. 102213, 2021.

[123] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[124] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[125] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, "Why does unsupervised pre-training help deep learning?" in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 201–208.

[126] Y. N. Kunang, S. Nurmaini, D. Stiawan, and B. Y. Suprapto, "Attack classification of an intrusion detection system using deep learning and hyperparameter optimization," *Journal of Information Security and Applications*, vol. 58, p. 102804, 2021.

[127] R. Bouman and T. Heskes, "Autoencoders for anomaly detection are unreliable," *arXiv preprint arXiv:2501.13864*, 2025.

[128] Y. Yuan, Y. Huang, and J. Wang, "Adaptive NAD: Online and Self-adaptive Unsupervised Network Anomaly Detector," *arXiv preprint arXiv:2410.22967*, 2024.

[129] X. Yang, J. Sun, and R. T. Rajan, "Aircraft Trajectory Prediction using ADS-B Data," in *Pre-Proceedings of the 2022 Symposium on Information Theory and Signal Processing in the Benelux*, 2022, p. 113.

[130] RTCA, "Minimum Operational Performance Automatic Dependent Surveillance-Broadcast (ADS-B) and Traffic Information Services-Broadcast (TIS-B)," Radio Technical Commission for Aeronautics, Washington, DC, Technical report DO-260B, 2011.

[131] ICAO, "Doc 4444, Procedures for Air Navigation Services — Air Traffic Management," International Civil Aviation Organization, Montréal, QC, Technical report Doc 4444 PANS-ATM, 2016.

[132] L. Ryon and G. Rice, "A safety-focused security risk assessment of commercial aircraft avionics," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. IEEE, 2018, pp. 1–8.

[133] NIST, "Recommendation for Block Cipher Modes of Operation:*Methods for Format-Preserving Encryption*," National Institute of Standards and Technology (NIST), Gaithersburg, MD, Technical Report NIST Special Publication (SP) 800-38G Rev. 1, 2019.

[134] Y. Challal, H. Bettahar, and A. Bouabdallah, "A taxonomy of multicast data origin authentication: Issues and solutions," *IEEE Communications Surveys & Tutorials*, vol. 6, no. 3, pp. 34–57, 2004.

[135] NIST, "Digital Signature Standard (DSS)," U.S. Department of Commerce, Gaithersburg, MD, Standard FIPS PUB 186-4, 2013. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

[136] X. Hu, T. Wang, and H. Xu, "Cryptanalysis and improvement of a hibe and hibs without random oracles," in *2010 International Conference on Machine Vision and Human-machine Interface*. IEEE, 2010, pp. 389–392.

[137] S. S. Chow, L. C. Hui, S. M. Yiu, and K. Chow, "Secure Hierarchical Identity Based signature and Its Application," in *International Conference on Information and Communications Security*. Springer, 2004, pp. 480–494.

[138] C. Gentry and A. Silverberg, "Hierarchical ID-based Cryptography," in *International conference on the theory and application of cryptology and information security*. Springer, 2002, pp. 548–566.

[139] NIST, "The Keyed-Hash Message Authentication code (HMAC)," U.S. Department of Commerce, Gaithersburg, MD, Standard FIPS PUB 198-1, 2008. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.198-1.pdf

[140] ——, "Recommendation for Applications Using Approved Hash Algorithms," U.S. Department of Commerce, Gaithersburg, MD, Standard NIST SP 800-107 Rev. 1, 2012. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-107r1.pdf

[141] S. Gregory T., "Systems and methods for providing an advanced atc data link," 1 Apr. 2010. [Online]. Available: https://patents.google.com/patent/US20100079329

[142] O. Yeste-Ojeda and R. Landry, "ADS-B Authentication Compliant with Mode-S Extended Squitter Using PSK Modulation," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC). Proceedings*, 2015, pp. 1773 – 1778.

[143] A. Nguyen, A. Amrhar, J. Zambrano, G. Brown, J. Landry, R., and O. Yeste, "Application of Phase Modulation Enabling Secure Automatic Dependent Surveillance-broadcast," *Journal of Air Transportation*, vol. 26, no. 4, pp. 157 – 170, 2018.

[144] M. Leonardi and M. Maisano, "Backward Compatible Physical Layer Protocol Evolution for ADS-B Message Authentication," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 5, pp. 16 – 26, 2020.

[145] NIST, "Secure Hash Standard (SHS)," U.S. Department of Commerce, Gaithersburg, MD, Standard FIPS PUB 180-4, 2015. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

[146] ——, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," U.S. Department of Commerce, Gaithersburg, MD, Standard FIPS PUB 202, 2015. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

[147] Linar Yusupov, "ADSB_Encoder.py," 2017. [Online]. Available: https://github.com/lyusupov/ADSB-Out

[148] C. Moler, "MATLAB version 9.11.0.1873467 (R2021b) Update 3," 2021. [Online]. Available: https://www.mathworks.com/products/matlab.html

[149] ITU-R, "Spectrum occupancy measurements and evaluation," ITU, Technical report Report SM.2256-1(06/2016), 2016. [Online]. Available: https://www.itu.int/pub/R-REP-SM.2256-1-2016

[150] J. Sun and J. M. Hoekstra, "Analyzing Aircraft Surveillance Signal Quality at the 1090 Megahertz Radio Frequency," in *Proceedings of the 9th International Conference for Research in Air Transportation*, 2020.

[151] OpenSky, "OpenSky Raw Data," 2018. [Online]. Available: https://opensky-network.org/datasets/raw/protected

[152] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, "Bringing Up Opensky: A Large-scale ADS-B Sensor Network for Research," in *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IPSN, 2014, pp. 83–94.

[153] T. Kistan, A. Gardi, R. Sabatini, S. Ramasamy, and E. Batuwangala, "An evolutionary outlook of air traffic flow management techniques," *Progress in Aerospace Sciences*, vol. 88, pp. 15–42, 2017.

[154] S. Sciancalepore, S. Alhazbi, and R. Di Pietro, "Reliability of ADS-B communications: Novel insights based on an experimental assessment," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2019, pp. 2414–2421.

[155] FAA, *Introduction to TCAS II Version 7.1*, 2011. [Online]. Available: https://www.faa.gov/documentLibrary/media/Advisory_Circular/TCAS%20II%20V7.1%20Intro%20booklet.pdf

[156] ICAO PKD, "ICAO Public Key Directory ICAO PKD White Paper – System Specification for participants," ICAO, Montréal, QC, White paper, 2020. [Online]. Available: https://www.icao.int/Security/FAL/PKD/Documents/PKDTechnicalDocuments/ICAO%20PKD%20White%20Paper_2020-07.pdf

[157] ICAO Security and Facilitation, "ICAO PKD Participants," 2022. [Online]. Available: https://www.icao.int/Security/FAL/PKD/Pages/ICAO-PKDParticipants.aspx

[158] X. Song, M. Wu, C. Jermaine, and S. Ranka, "Conditional Anomaly Detection," *IEEE Transactions on knowledge and Data Engineering*, vol. 19, no. 5, pp. 631–645, 2007.

[159] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *KDD*, vol. 96, no. 34, 1996, pp. 226–231.

[160] V. M. Janakiraman and D. Nielsen, "Anomaly detection in aviation data using extreme learning machines," in *2016 international joint conference on neural networks (IJCNN)*. IEEE, 2016, pp. 1993–2000.

[161] E. Habler and A. Shabtai, "Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages," *Computers & Security*, vol. 78, pp. 155–173, 2018.

[162] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, "Bringing up OpenSky: A large-scale ADS-B sensor network for research," in *IPSN-14 proceedings of the 13th international symposium on information processing in sensor networks*. IEEE, 2014, pp. 83–94.

[163] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[164] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[165] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[166] M. Beck, K. Pöppel, P. Lippe, R. Kurle, P. M. Blies, G. Klambauer, S. Böck, and S. Hochreiter, "xLSTM 7B: A Recurrent LLM for Fast and Efficient Inference," *arXiv preprint arXiv:2503.13427*, 2025.

[167] F. Shang, B. Wang, F. Yan, and T. Li, "Multidevice False Data Injection Attack Models of ADS-B Multilateration Systems," *Security and Communication Networks*, vol. 2019, 2019.

[168] J.-Y. De Micelli, "Détection d'attaques informatiques sophistiquées contre les communications ADS-B en aviation," mémoire de maîtrise, Dép. de génie informatique et génie logiciel, École Polytechnique de Montréal, Montréal, QC, 2020.

[169] Z. Wu, T. Shang, and A. Guo, "Security Issues in Automatic Dependent Surveillance-Broadcast (ADS-B): A Survey," *IEEE Access*, vol. 8, pp. 122 147–122 167, 2020. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2020.3007182

[170] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, 2016, pp. 21–26.

[171] C. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21 954–21 961, 2017.

APPENDIX A EXISTING VERIFICATION TECHNIQUES AND MOTIVATION FOR MACHINE LEARNING-BASED APPROACHES

A variety of techniques have been proposed to detect forged ADS-B messages by assessing the plausibility of received data. Traditional ADS-B data integrity verification techniques include signal-level measurements, probabilistic estimations, and rule-based reasoning. Notable examples include multilateration, Kalman filtering, data fusion, distance bounding and ontology-based expert systems. Below, we briefly define each technique, with its strengths and limitations summarized in Table A.

- **Multilateration (MLAT):** A surveillance method that estimates the position of an aircraft by measuring the Time Differences of Arrival (TDOA) of a signal broadcast by the aircraft at multiple spatially separated ground receivers [36].
- **Kalman filtering:** An algorithm that improves the estimation of the state of an aircraft, such as its position, velocity, etc., by combining noisy sensor measurements with predictive motion models to produce smoother and more accurate tracking [167].
- **Data fusion:** The process of integrating information from multiple sources, such as ADS-B, radar, MLAT, and flight plans, to construct a comprehensive and reliable representation of air traffic [44].
- **Distance bounding:** A verification technique whereby ground stations confirm the actual distance of an aircraft by measuring response times to rapid challenge signals, thus validating the authenticity of the reported ADS-B position [29].
- **Ontology-based Expert Systems:** Systems that utilize an ontology[1] to formally define key aviation concepts (e.g., aircraft, flight plans, message types) and their interrelations; an expert system then applies rules to this structured knowledge to detect suspicious or inconsistent ADS-B messages through semantic reasoning [168].

Although these methods contribute to ADS-B data integrity verification, they exhibit important limitations. Multilateration requires a specialized and synchronized ground infrastructure. Although deployment is feasible in well-covered regions, extending the coverage to remote areas remains technically and logistically challenging [28, 29]. Distance bounding is particularly difficult to implement in aviation owing to the high speed of aircraft. This

[1] An ontology is a structured knowledge base that formally represents concepts and their relationships within a specific domain.

Method	Strengths	Weaknesses
MLAT	Determines aircraft position independently of ADS-B data using the Time Difference of Arrival (TDOA) of the signals.	Sensitive to signal reflections; requires synchronized ground stations; difficult to deploy in remote areas.
Kalman Filtering	Combines motion predictions with noisy measurements (ADS-B data) to provide accurate and smooth aircraft state estimation.	High computational cost; delays in state updates; vulnerable to subtle (gradual) attacks.
Data Fusion	Combines multiple surveillance sources to enhance situational awareness.	Inconsistent data formats, timing, and noise levels across sources degrade tracking performance.
Distance Bounding	Provides a physical-layer method to verify whether the aircraft is located where it claims to be.	Aircraft motion can affect measurement accuracy; protocol modifications are required.
Expert System	Applies domain knowledge and predefined rules to identify inconsistencies in ADS-B data; decisions are interpretable.	Labor-intensive to build and maintain; scalability issues slow processing; cannot adapt to new threats automatically.

Table A.1 Strengths and weaknesses of traditional approaches for ADS-B data integrity verification.

technique relies on measuring the round-trip time of signals exchanged with the aircraft, and even a slight motion during the response interval can introduce errors in the estimated distance [169]. Additionally, distance bounding requires precise control over signal-processing delays, which is technically demanding and often necessitates modifications to the ADS-B protocol, potentially affecting its compatibility with existing systems [29]. Kalman filtering techniques improve tracking accuracy, but their computational complexity increases significantly when handling multivariate data and complex motion models. As the system becomes more complex, the filtering process slows, which can delay state updates and negatively impact the situational awareness of air traffic controllers. Kalman filtering is also susceptible to "frog boiling" attacks, where adversaries gradually inject false data within the filter's tolerance, allowing the trajectory to be manipulated without raising alarms [27]. Data fusion enhances situational awareness by combining multiple surveillance sources; however, inconsistencies in data formats, timing, and noise levels across these sources can degrade the performance [44]. Expert systems provide transparent and interpretable decisions but require

continuous manual maintenance and struggle to adapt to new or evolving threats.

Machine learning methods, particularly deep learning, overcome many of the limitations of traditional approaches for verifying the integrity of ADS-B data, most notably by avoiding the need for protocol modifications and adapting to emerging threats. These methods learn patterns from large volumes of historical data, enabling them to generalize to unseen behaviors, capture temporal dependencies, and detect subtle anomalies without relying on handcrafted rules. Their effectiveness has been demonstrated in areas such as malware and network intrusion detection [45, 46, 170, 171], and these benefits extend naturally to ADS-B systems.