Air Force Institute of Technology
AFIT Scholar

Theses and Dissertations

Student Graduate Works

3-14-2014

Secure ADS-B: Towards Airborne Communications Security in the Federal Aviation Administration's Next Generation Air Transportation System

Richard C. Agbeyibor

Follow this and additional works at: https://scholar.afit.edu/etd

Part of the Aviation Safety and Security Commons

Recommended Citation

Agbeyibor, Richard C., "Secure ADS-B: Towards Airborne Communications Security in the Federal Aviation Administration's Next Generation Air Transportation System" (2014). *Theses and Dissertations*. 584.
https://scholar.afit.edu/etd/584

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact richard.mansfield@afit.edu.

SECURE ADS-B: TOWARDS AIRBORNE COMMUNICATIONS SECURITY IN THE FEDERAL AVIATION ADMINISTRATION'S NEXT GENERATION AIR TRANSPORTATION SYSTEM

THESIS

Richard C. Agbeyibor, Second Lieutenant, USAF

AFIT-ENG-14-M-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the Department of Defense, or the United States Government.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENG-14-M-02

SECURE ADS-B: TOWARDS AIRBORNE COMMUNICATIONS SECURITY IN THE FEDERAL AVIATION ADMINISTRATION'S NEXT GENERATION AIR TRANSPORTATION SYSTEM

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Electrical Engineering

Richard C. Agbeyibor, B.S.
Second Lieutenant, USAF

March 2014

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT-ENG-14-M-02

SECURE ADS-B: TOWARDS AIRBORNE COMMUNICATIONS SECURITY IN THE FEDERAL AVIATION ADMINISTRATION'S NEXT GENERATION AIR TRANSPORTATION SYSTEM

Richard C. Agbeyibor, B.S.
Second Lieutenant, USAF

Approved:

//signed// 28 Feb 2014
Maj Jonathan Butts , PhD (Chairman) Date

//signed// 28 Feb 2014
Robert F. Mills, PhD (Member) Date

//signed// 28 Feb 2014
Michael R. Grimaila, PhD, CISM, CISSP (Member) Date

AFIT-ENG-14-M-02

Abstract

The U.S. Congress has mandated that all aircraft operating within the National Airspace System, military or civilian, be equipped with Automatic Dependent Surveillance-Broadcast (ADS-B) transponders by the year 2020. The ADS-B aircraft tracking system, part of the Federal Aviation Administration's NextGen overhaul of the Air Transportation System, replaces Radar-based surveillance with a more accurate satellite-based surveillance system. However, the unencrypted nature of ADS-B communication poses an operational security risk to military and law enforcement aircraft conducting sensitive missions. The non-standard format of its message and the legacy communication channels used by its transponders make the ADS-B system unsuitable for traditional encryption mechanisms. Format-Preserving Encryption (FPE), a recent development in cryptography, provides the ability to encrypt arbitrarily formatted data without padding or truncation. Indeed, three new algorithms recommended by the National Institute of Standards and Technology (NIST), may be suitable for encryption of ADS-B messages. This research assesses the security and hardware performance characteristics of the FF1, FF2, and FF3 algorithms, in terms of entropy of ciphertext, operational latency and resource utilization when implemented on a Field-Programmable Gate Array. While all of the algorithms inherit the security characteristics of the underlying Advanced Encryption Standard (AES) block cipher, they exhibit differences in their performance profiles. Findings demonstrate that a Bump-in-the-Wire FPE cryptographic engine is a suitable solution for retrofitting encryption to ADS-B communication.

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Maj. Jonathan Butts, for his guidance throughout the course of this thesis effort. I would also like to thank Mr. Steve Stokes at the Air Force Research Labs (AFRL) for his indefatigable support and mentorship. Mr. David Prentice at AFRL and Mr. Pranav Patel at AFIT, were instrumental in helping me develop the avionics and hardware understanding necessary to conduct my experiments.

Additionally, I would like to thank my classmates and colleagues, especially the convivial band of Second Lieutenants, who have made these graduate studies most enjoyable. Finally, I would like to thank my parents and my girlfriend for their unwavering emotional support.

Richard C. Agbeyibor

Table of Contents

Page

List of Figures

Figure Page

List of Tables

Table Page

List of Acronyms

Acronym	Definition
AA	Aircraft Address
ADS-B	Automatic Dependent Surveillance-Broadcast
AES	Advanced Encryption Standard
AIS	Automatic Identification System
ANSI	American National Standards Institute
ARTCC	Air Route Traffic Control Center
ATC	Air Traffic Control
ATS	Air Transportation System
BITW	Bump-in-the-Wire
CA	Capability
CI	Confidence Interval
COMSEC	Communications Security
CPR	Compact Position Reporting
CRC	Cyclic Redundancy Check
DF	Downlink Format
DHS	Department of Homeland Security
DoD	Department of Defense
ECB	Electronic Codebook
ES	Extended Squitter
FAA	Federal Aviation Administration
FIS-B	Flight Information Services-Broadcast
FL180	Flight Level 180
FMS	Flight Management System

Acronym	Definition
FPE	Format-Preserving Encryption
FPGA	Field-Programmable Gate Array
GA	General Aviation
GPS	Global Positioning System
ICAO	International Civil Aviation Organization
IFF	Identification Friend or Foe
IFR	Instrument Flight Rules
IMO	International Maritime Organization
ISO	International Organization for Standardization
LUT	Look-Up Table
M5L1	Mode 5 Level 1
M5L2	Mode 5 Level 2
ME	Message Extended Squitter
NAS	National Airspace System
NATO	North Atlantic Treaty Organization
NextGen	Next Generation Air Transportation System
NIST	National Institute of Standards and Technology
NM	nautical miles
NSA	National Security Agency
OPSEC	Operational Security
PI	Parity/Interrogator Identity
PRF	Pseudo-Random Function
PSR	Primary Surveillance Radar
SDR	Software-Defined Radio
SES	Single European Sky

Acronym	Definition
SSR	Secondary Surveillance Radar
STANAG	Standard Agreement
TC	Type Code
TIS-B	Traffic Information Services-Broadcast
TRNG	True Random Number Generator
UAV	Unmanned Aerial Vehicle
VFR	Visual Flight Rules
WAAS	Wide Area Augmentation System
WADS	Western Air Defense Sector
W-AIS	Warship-Automatic Identification System

SECURE ADS-B: TOWARDS AIRBORNE COMMUNICATIONS SECURITY IN THE FEDERAL AVIATION ADMINISTRATION'S NEXT GENERATION AIR TRANSPORTATION SYSTEM

I. Introduction

1.1 Background

The Federal Aviation Administration (FAA) is upgrading the aging National Airspace System (NAS) to a higher capacity Next Generation Air Transportation System (NextGen). A major component of the new Air Traffic Control (ATC) system is Automatic Dependent Surveillance-Broadcast (ADS-B), which upgrades the slow and costly Radar-based surveillance system to a more precise and efficient position reporting system based on the Global Positioning System (GPS) and Wide Area Augmentation System (WAAS) [15].

The U.S. Congress has mandated through the Vision 100 - Century of Aviation Reauthorization Act [64] - that all aircraft, military and civilian, update their equipment to ADS-B capable transponders by the year 2020. Recent research, however, has demonstrated the ease with which ADS-B messages can be spoofed and false traffic injected into the ADS-B domain [38]. In addition to the danger of spoofed or non-existing aircraft appearing in the ATC system, sensitive traffic can be easily tracked with the aid of commercially available equipment. As example of a potentially malicious scenario, an anonymous user with an inexpensive ADS-B In receiver can track the precise latitude, longitude and altitude of Air Force One or other aircraft carrying political dignitaries.

The U.S. military has identified unique applications of ADS-B for its operations, but is concerned with the Communications Security (COMSEC) vulnerabilities of the system [25]. As such, the Department of Defense (DoD) has asked for the development of encryption and jam/spoof proofing mechanisms for ADS-B. The U.S. Navy and Coast Guard use the Advanced Encryption Standard (AES) and Blowfish algorithms to encrypt the Automatic Identification System (AIS) [46], their homologous vessel tracking system. However, the non-standard format of ADS-B messages and the legacy communication channels used by its transponders make it incompatible with traditional encryption mechanisms. Indeed, traditional encryption mechanisms require a message of standard size, such as 128-bit blocks for the AES algorithm, or a message that can be padded or truncated to fit the expected format. ADS-B, however, reuses existing 1090 Mhz Mode S channels and transponders which are limited to transmitting and processing messages that are 112 bits in size.

Format-Preserving Encryption (FPE), a recent development in cryptography, provides the ability to encrypt arbitrarily formatted data without padding or truncation [3]. The National Institute of Standards and Technology (NIST) recently released Draft SP800-38G - Recommendation for Block Cipher Modes of Operation: Methods for FPE [10], which recommends three algorithms for Format-Preserving Encryption. The NIST and members of the cryptography community suggest that FPE algorithms inherit the security characteristics of the underlying block cipher [48]. The FF1, FF2 and FF3 algorithms, recommended by the NIST may be suitable for retrofitting encryption to the ADS-B system.

An alternate solution for maintaining the Operational Security (OPSEC) of sensitive military and law enforcement aircraft is to adapt ADS-B messages for use within the existing military Identification Friend or Foe (IFF) transponders [32]. IFF transponders use a Type-1 [29] algorithm, approved by the National Security Agency (NSA), which is embedded in a programmable cryptographic engine. However, Mode 5 Level 2 (M5L2) IFF transponders lack a well-defined framework for precision tracking, and suffer from high latency which leads to imprecise position messages. The M5L2-B solution for retrofitting encryption to ADS-B trades accuracy for security. This leaves the Air Force with the options of either not complying with the 2020 congressional mandate, using the inaccurate M5L2-B position reporting system, or worst, operating with the unsecured ADS-B system and sacrificing OPSEC.

A more desirable solution is to provide security while maintaining the precision and accuracy of the existing ADS-B system. A Bump-in-the-Wire (BITW) FPE cryptographic engine could be retrofitted to existing ADS-B transponders to accomplish that task. Such a cryptographic engine would have to meet hardware performance requirements established by the FAA for mission-critical avionics equipment [17].

1.2 Motivation

As far as the military is concerned, the NextGen upgrade is insecure as designed, and solutions to its security gaps must be found before moving towards military implementation. However, the FAA maintains that the upgraded system does not subject aircraft to any increased risk compared to that which is already experienced given the current surveillance system [17]. Nevertheless, military aircraft manufacturers have started testing unsecured ADS-B transponders for use in manned and Unmanned Aerial Vehicles (UAVs) [22].

In 2013, Finke proposed the FFX [2] FPE algorithm for use within the Next Generation Air Transportation System [20]. Since then, the NIST has reviewed candidate algorithms for standardization and has officially recommended three algorithms for FPE. At the time of this writing, the NIST has not released details of its internal deliberations nor performance assessments of the FF1, FF2, and FF3 algorithms.

1.3 Research Objectives

The goal of this research is to determine the suitability of the FF1, FF2, and FF3 algorithms for encryption of ADS-B messages, with regards to security and performance.

The first objective is to evaluate the security characteristics of each algorithm within a representative ADS-B environment. Part of the objective is to validate the hypothesis suggested by the algorithm designers and the NIST that the algorithms inherit the strong security characteristics of the block cipher used in the Feistel round function [10, 41].

The second objective is to evaluate the hardware performance of each algorithm by measuring operational latency and resource utilization of a Field-Programmable Gate Array (FPGA) implementation. DO-260B "Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance - Broadcast (ADS-B) and Traffic Information Services - Broadcast (TIS-B)" [55] specifies timing and latency requirements for the ADS-B transponder. The performance of the algorithms is assessed according to the DO-260B standard.

Finally, the research assesses the merits of a BITW FPE cryptographic engine implementing FF1, FF2 or FF3 for retrofitting security to existing ADS-B avionics equipment.

1.4 Approach

The research objectives are approached through modeling and simulation in software, and measurement of a hardware implementation. The methodology used to evaluate the security characterisitics of the FF1, FF2, and FF3 algorithms builds on research conducted on the FFX algorithm by Finke [20]. The algorithms are implemented in C following the pseudocode descriptions provided in [10], with 128-bit AES as the underlying block cipher. Pilot experiments determined that byte alignment and CPU optimization requirements limit the C programming language to the byte as its lowest level of data granularity. Given these limitations, only 104 of the available 107 encrypt-able bits of the ADS-B message are encrypted. The algorithms are tested with a model dataset composed of incrementally deterministic messages in the Fixed Bytes test, a simulated ADS-B message dataset in the Fixed Fields test, and an operational ADS-B dataset extracted from an observed Radar track. The ENT tool [66] is used to measure the Shannon entropy of the resulting ciphertext. Statistical tests are conducted to compare the ability of the FPE algorithms to produce ciphertext with entropy equal to or greater than that of a random sequence.

Once evaluated in software, the algorithms are implemented in VHDL. The hardware designs are simulated and synthesized on the Virtex-6 FPGA using the Xilinx ISE 14.6 design suite. An Iterative Looping architecture is used to implement the Feistel structure of FPE. Behavioral simulation tests, Post-PAR static timing analysis and device utilization analysis are performed on each design. The hardware implementations are compared to each other and to the underlying AES core. An analysis of the research results details the security and performance characteristics of each algorithm and suitability for use in a BITW FPE cryptographic engine for ADS-B avionics equipment.

1.5 Organization

Chapter II reviews the state of the NAS, discusses operating specifications of ADS-B, relevant encryption theory, and describes the FF1, FF2, and FF3 algorithms. Chapter III presents the methodology for evaluating the security and performance of the three algorithms. Chapter IV presents the results of the experiments and an analysis of the findings. In conclusion, Chapter V summarizes the research effort and offers suggestions for future work.

II. Background

This chapter presents necessary background information and examines related research. It assesses the security requirements of ADS-B, a key component of the NextGen ATC system and presents the three methods for Format-Preserving Encryption recommended by the NIST. Finally, it surveys the software and hardware requirements of ADS-B equipment and examines the suitability of FF1, FF2, and FF3.

2.1 The National Airspace System

Following World War II, an increase in air travel in the United States prompted the creation of the Federal Aviation Agency to manage the nation's Air Transportation System (ATS) [15]. The NAS was then created and has evolved into a complex system-of-systems. The NAS consists of a network of air navigation facilities, ATC facilities, airports, radar stations, radio beacons, and the panoply of rules and regulations necessary to provide a safe and efficient flying environment. It is divided into 21 Air Route Traffic Control Centers (ARTCCs), each responsible for a regional sector, which in turn manage more than 690 ATC facilities with associated systems and equipment in order to provide radar and communication services to aircraft transiting the NAS .

In aviation, aircraft operate under two distinct categories of operational flight rules: Visual Flight Rules (VFR) and Instrument Flight Rules (IFR). Under VFR, typically used by General Aviation (GA) aircraft operating under 18,000 feet, the pilot is primarily responsible for seeing other aircraft and maintaining safe separation. This ceiling is also known as Flight Level 180 (FL180). Under IFR, used by commercial and other high-performance aircraft operating above FL180, ATC is primarily responsible for providing aircraft separation in a controlled airspace [16]. Aircraft operating under IFR typically fly along predefined airways and rely on controllers to detect route conflicts and provide navigational direction in order to maintain safe separation. In 2007 alone, FAA towers logged approximately 48,200,000 instrument operations of which 30 percent were air carrier, 27 percent air taxi, 37 percent general aviation, and 6 percent military [15]. The FAA projects a growth in the commercial aviation space from approximately 750 million in 2012 to an unprecedented 1.15 billion enplaned passengers by 2033, as shown in Figure 2.1. Air traffic controllers currently handle 9 to 15 aircraft at any one point [24]. With the projected increase in air traffic, experts believe controllers could be required to handle up to 45 aircraft at any one point, a situation that is completely unsafe and infeasible to manage [27].

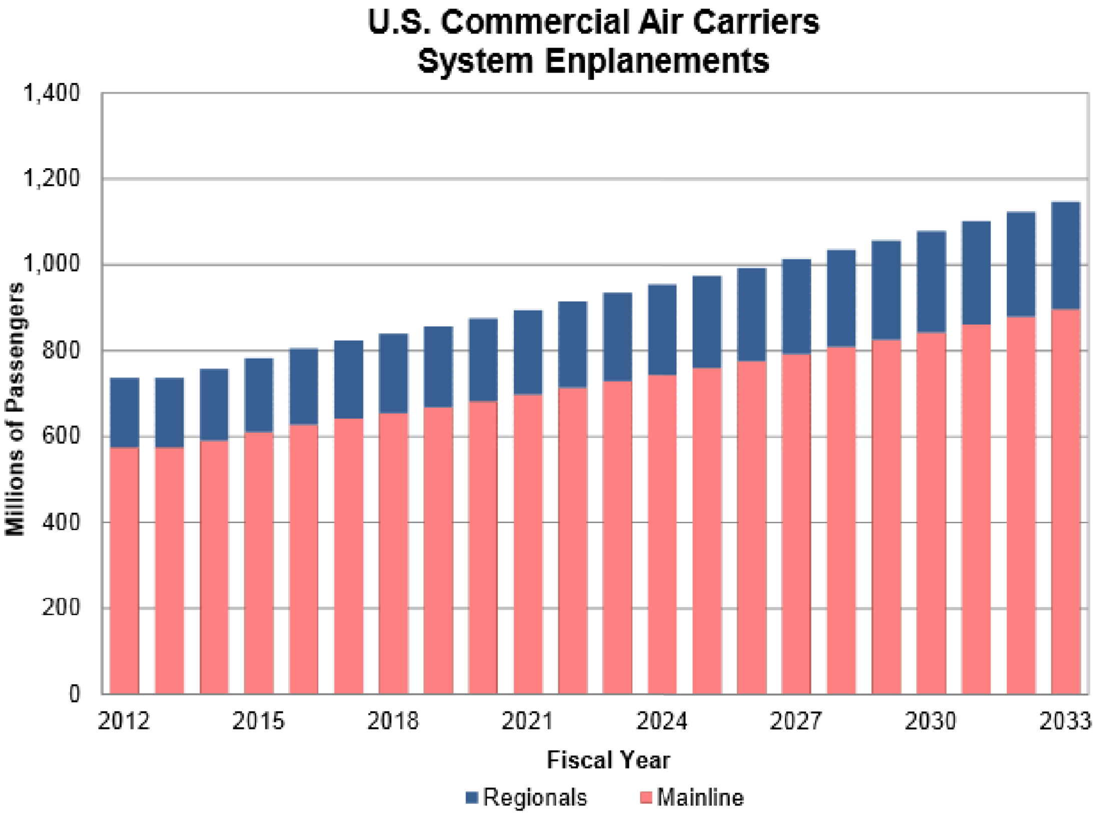


Figure 2.1: FAA Passenger Enplanement Forecast [14].

Since the advent of the FAA in 1958, advances in radar technology, navigation technology, and aircraft avionics have enabled significant expansion of the ATC system. However, the system is now approaching its operating limits and the FAA is looking to improvements in communication and navigation instruments to bring about an evolution towards Free Flight [15]. Free Flight is a concept which minimizes the role of ATC operators and gives responsibility to aircrews to make flight path decisions in a cooperative and distributed decision-making process. Currently, ATC constrains airplanes under its control to fly on fixed airways that are covered by ground-based radar and navigation beacons. Under Free Flight, pilots could file a flight plan and make changes en route without contacting ATC. This freedom would allow the crew to select the shortest, most fuel-efficient route or the most comfortable flight level. Free Flight, however, can only be effective if aircraft are equipped with accurate position determination, collision avoidance and data communications equipment [26].

2.2 The Next Generation Air Transportation System

The current NAS, designed in 1982, relies on legacy infrastructure and antiquated technology [15]. The NextGen is scheduled for implementation across the United States in stages between 2012 and 2025. This transformation aims to enhance safety, reduce delays, save fuel and reduce aircraft exhaust emissions, in addition to its primary mission of enabling sustainment of the increasing demand in air transportation across the country [18]. NextGen was approved in 2003 by the U.S. Congress, and signed into law through the Vision 100 - Century of Aviation Reauthorization Act [64]. NextGen and Europe's upcoming Single European Sky (SES) system, will contribute to the delivery of the International Civil Aviation Organization (ICAO)'s One Sky vision - a seamless, performance-based global air navigation system [56].

The NextGen overhaul to the NAS includes transformational programs for: (i) satellite-based navigation, (ii) collaborative air traffic management, (iii) data communications, (iv) network-enabled weather services, (v) digital voice communication technology, and (vi) improvements to the NAS network infrastructure [18]. One of the most significant changes is the inclusion of the ADS-B system which is intended to improve surveillance capabilities of ATC and enable precision traffic separation and routing. The FAA Reauthorization Bill of 2010 mandates that all aircraft (GA, commercial, and military) operating within the NAS be equipped with ADS-B Out by 2020 [64]. ADS-B Out is the requisite transponder technology which enables aircraft to transmit messages to ground stations and ADS-B In equipped aircraft. ADS-B In technology enables the user to receive and process ADS-B messages from nearby transmitters. Note that lawmakers are considering making ADS-B In mandatory in the near future [18].

The current NAS relies on ground-based Radio Detection and Ranging (Radar) for aircraft surveillance. Primary Surveillance Radar (PSR) uses a network of ground-based stations which can detect targets within a range of approximately 75 nautical miles (NM) [26]. PSR locates a target using the antenna angle at the time of transmission, and the elapsed time before the backscattered signal is received. Note that this information is two-dimensional, while aircraft exist in a three-dimensional world. Secondary Surveillance Radar (SSR) adds two supplemental data points about the target aircraft, and is based on the IFF system introduced in World War II. The SSR emits an interrogation signal, and aircraft in the coverage area equipped with a compatible transponder reply with altitude and identification information. The current ground-based Radar system requires large rotating antennas that are costly to maintain, suffer from significant coverage gaps, and are slow to update [50]. The Radar system has a refresh rate of about 12 seconds, which is slow for aircraft moving at 200+ knots, and can be precise only up to 300 meters. ADS-B employs the same onboard transponder technology and communication channel as SSR, but offers an improved refresh rate of half a second (2 Hz), as well as precision of up to 20 meters [38, 50].

2.3 Automatic Dependent Surveillance-Broadcast

The concept of Automatic Dependent Surveillance (ADS) was first introduced by the ICAO in the 1980s and outlined in the Future Air Navigation System (FANS) plan [8, 20]. ADS-B is *Automatic* in that it does not require interrogation from the ground or other aircraft. It is *Dependent* because it relies on information from aircraft sensors and other onboard equipment to provide *Surveillance* services. Finally, and most critical to this research, ADS-B indiscriminately *Broadcasts* its data to all users within range.

ADS-B enables pilots and ATC to share and display the same information. It relies on the Global Positioning System (GPS) and other satellite navigation tools such as the Wide Area Augmentation System (WAAS) to accurately determine an aircraft's position. The precise location, along with other data such as aircraft identification, airspeed, altitude, and heading gathered from the aircraft's Flight Management System (FMS), are relayed to ground stations and other equipped aircraft as shown in Figure 2.2 [15].

The FAA has identified two options for equipage under the ADS-B mandate: the 978 MHz UAT and the 1090 MHz ES [17]. The 978 MHz Universal Access Transceiver is a new data link designed specifically for GA aircraft which can process ADS-B along with Flight Information Services-Broadcast (FIS-B) and Traffic Information Services-Broadcast (TIS-B). The 1090 MHz Extended Squitter link uses an existing message type supported by Mode S transponders to transmit ADS-B messages. A squitter message or a squawk is a transmitted message not invoked by any interrogation. The 1090 MHz channel is the internationally adopted broadcast frequency, designated for commercial and high-performance aircraft, and is the focus of this research. An ADS-B squitter is 112 bits wide and 120 μs long with an 8 μs preamble. As shown in Figure 2.3, 56 of the 112 bits

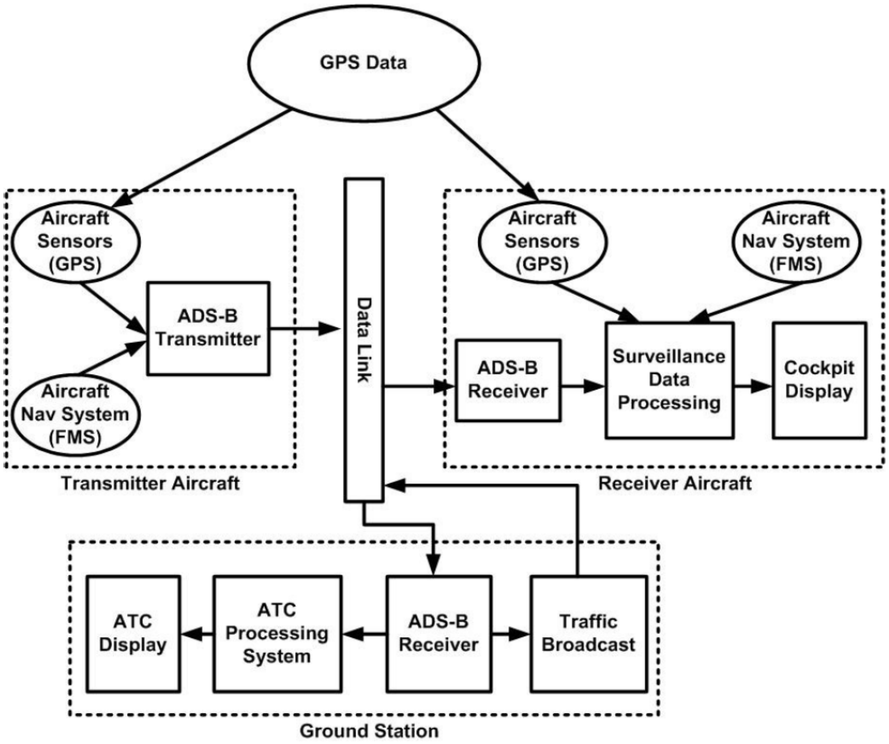


Figure 2.2: Major Components of the ADS-B System [38].

are for ADS-B specific data to include altitude, latitude and longitude. The remaining bits are used for the message format, transponder capability, aircraft address or identifier, and a parity check for data integrity.

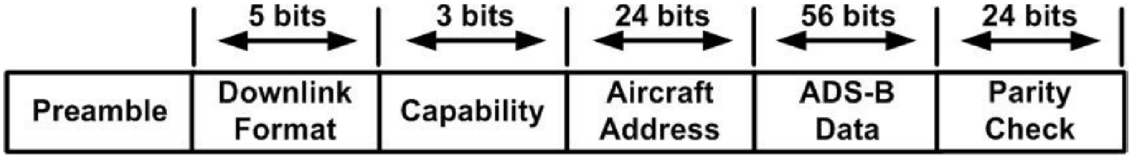


Figure 2.3: ADS-B Message Data Link Layer [38].

2.3.1 Logistical Advantages of ADS-B.

The ADS-B system adds functionality to the NextGen upgrade while reusing the existing 1090 MHz broadcast frequency and Mode S transponder technology.

One of the primary advantages of ADS-B is its ability to provide coverage where Radar is not available. This is particularly relevant in transoceanic flight where viable locations for Radar stations are minimal. Indeed, a few strategically placed ADS-B broadcast stations, in addition to rebroadcasting ADS-B In-equipped aircraft, will enhance transoceanic coverage [37]. Another advantage is the smaller footprint of ADS-B facilities, which allows the FAA to deploy broadcast stations on structures such as oil rigs many miles out from land. In addition to a smaller footprint, operation and maintenance of ADS-B equipment is significantly cheaper than Radar, costing approximately $100-$400 thousand per ADS-B station versus $1-$4 million for a radar station [28, 37].

Alaska's Capstone program, an experiment in testing ADS-B technology and its effect on air traffic controller workload, showed a significant reduction in stress and an increase in efficiency. During the trial period, 208 aircraft were equipped with ADS-B and normal flights in and out of the Alaskan region were monitored. After program completion, surveys of controllers found that 57% said they had spent less time providing IFR separation services, and 79% felt their overall efficiency increased with ADS-B [37, 57]. These advantages, which enable the FAA to accomplish its mission more effectively and at lower cost, have sparked the interests of other prominent actors in the aviation world, notably the United States Armed Forces.

2.3.2 Military Applications.

The U.S. Air Force has identified benefits associated with the transition to NextGen and particularly the ADS-B technology. The Air Force operates three types of missions: Open, Sensitive, and Covert [25]. ADS-B technology could be employed in one or all of these mission types if encryption and jam/spoof resistance features are developed. Specifically, ADS-B could enhance safety and mission capabilities in Air Refueling (AR), Formation Flying, Rendezvous, Fighter Intercept, Air Combat Maneuvering Instrumentation (ACMI) missions, and precision Airdrop [12, 25]. These military-unique applications for ADS-B were identified in 2001 - 13 years ago; however the Air Force has not yet ratified these proposals.

Nevertheless, military aircraft manufacturers have started testing ADS-B technology for use in manned and Unmanned Aerial Vehicle (UAV). General Atomics-Aeronautical Systems Inc, a major defense contractor, has tested a BAE Systems-developed military grade IFF transponder with ADS-B In and ADS-B Out capabilities for use within its Airborne Sense-And-Avoid architecture (ABSAA) [22]. The test was part of a series of demonstrations aiming to prove that UAVs can fly cooperatively and safely in the National Airspace System. Note that the Sense-And-Avoid architecture in development could also be used in the future by autonomous swarms of UAVs for precise formation flight. However, many issues with the ADS-B system must be addressed before it is deployed in such safety critical systems.

The military considers the lack of encryption and jam/spoof resistance features in ADS-B, a significant OPSEC risk [25]. In response, the FAA maintains that the upgraded surveillance system does not subject aircraft to any increased risk compared to that which is already experienced given the current surveillance system [17]. However, GA and military aircraft operate under different risk profiles. A military aircraft conducting an Open mission may accept the same risk profile as a GA aircraft. However, certain missions require much more stringent OPSEC.

2.3.3 Operational Security.

The United States DoD Policy Board on Federal Aviation (PFBA) has stated that the FAA “needs to continue to work with DoD and DHS to ensure that concerns about ADS-

B security are adequately addressed" [5]. Specifically, DoD policy makers are intent on a "requirement to develop operational procedures for special [US Government] flights (such as low observable surveillance aircraft, UASs, combat air patrol missions, counter-drug missions, counter-terrorism missions, VIP transport, law enforcement surveillance, etc)" [5].

A concern with ADS-B is the ability of any individual to purchase commercially available equipment that is capable of receiving ADS-B messages and monitoring air traffic. As an example of the potentially malicious use of such information, the mobile application Plane Finder AR allows a user to aim a smart phone at a passing aircraft, and the application queries an Internet database for flight information including call sign, altitude, current heading, origin/destination and relative distance from the user's current position [20, 38]. Since early warfare, opposing forces have tried to track and maintain an accurate count of one another's forces. Such tracking and targeting capability for such low cost, is a major OPSEC risk for military and law enforcement operations [38].

2.3.4 Communications Security.

COMSEC is the discipline of preventing unauthorized interceptors from accessing telecommunications in an intelligible form, while still delivering content to the intended recipients. The CIA model encompasses the three core security principles of Confidentiality, Integrity, and Availability (CIA) [63]. Confidentiality refers to preventing the disclosure of information to unauthorized parties. Integrity refers to maintaining the accuracy of the data throughout the transmission lifecycle. Availability refers to preventing disruptions to the transmission and that the data remains accessible to authorized parties.

Researchers have analyzed the security vulnerabilities of ADS-B. In [38], McCallie *et al.* provided a taxonomy of attacks against ADS-B. The ADS-B system can be attacked through individual avionics components, during message transmission, and through the backbone network used to share data between ARTCCs. The most probable attack vectors are those aimed at exploiting ADS-B messages being transmitted and received by an aircraft [38]. Specifically, the use of plaintext broadcasts by ADS-B allows messages to be spoofed, replicated, or modified. Of the six various forms of attacks against NextGen outlined in [38], three include the injection of false ADS-B messages. Recent presentations at the Black Hat [37] and Def Con 20 [20] conferences have demonstrated the ability to generate and broadcast false messages with relative ease and at low cost.The DoD has asked for the development of encryption and jam/spoof proofing mechanisms [25] to protect the Confidentiality and Availability of messages being transmitted and received by aircraft [61]. Note that ADS-B already provides Integrity through the use of a parity check.

One approach for adding encryption and jam/spoof proofing to ADS-B is to adapt ADS-B messages for use within the existing military Identification Friend or Foe (IFF) transponders. The United States military and North Atlantic Treaty Organization (NATO) allies are currently equipped with the Mark XIIA Mode 5 system for airborne IFF as defined by NATO STANAG 4193 [45]. Confidentiality in Mode 5 is provided by a NSA-approved Type-1 algorithm embedded in a programmable cryptographic engine [32]. The Mode 5 waveform uses minimum shift keying (MSK) modulation and spread spectrum techniques to realize a processing gain waveform and insure the Availability of the message [32]. Mode 5 Level 1 (M5L1) is currently fielded and offers significant improvements in secure friend determination over the legacy Mode 4. M5L2 is a new asynchronous mode for secure self reporting, with ability to report GPS data in 77 bit tactical data report messages. M5L2 can provide up to 16 message formats, 11 of which are reserved for standard IFF, and 5 are proposed for assignment to Military ADS-B functions [32]. A 112 bit ADS-B message would be reformatted to fit into a 77 bit M5L2 message as shown in Figure 2.4.

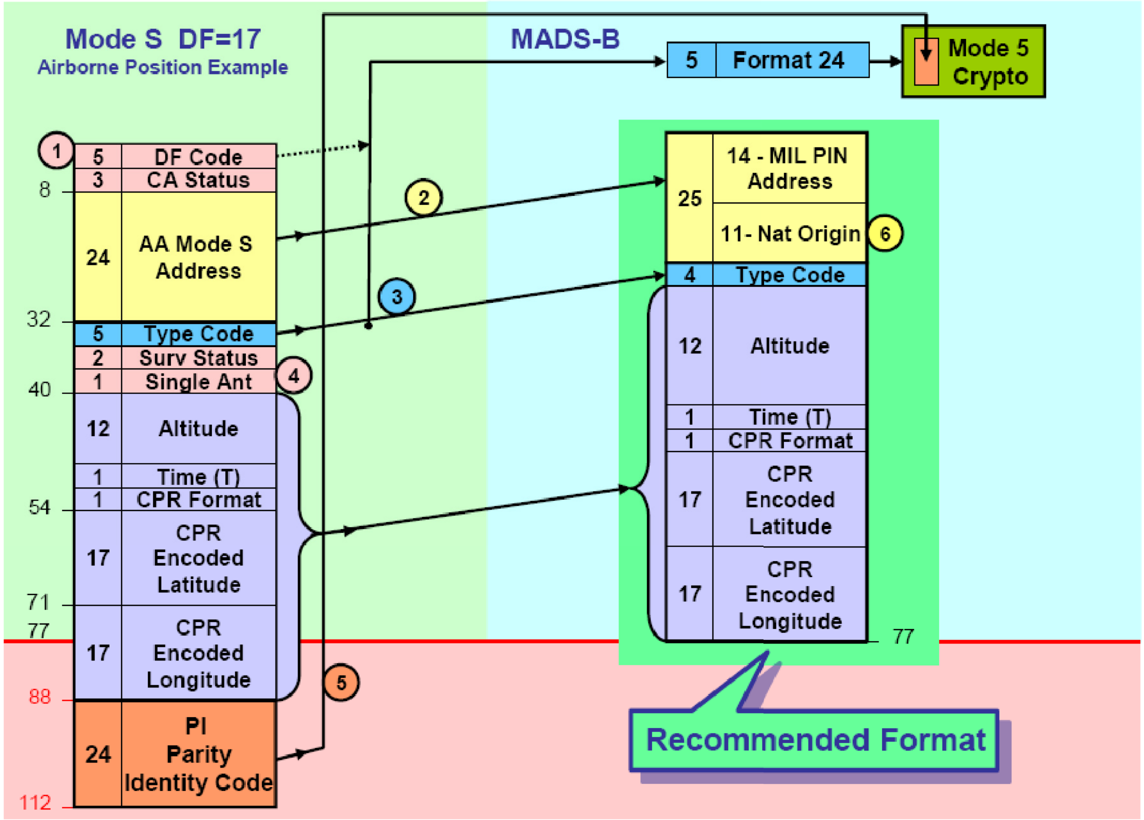


Figure 2.4: An Airborne Position ADS-B message converted into a M5L2 message [32].

Unfortunately, M5L2 does not have a well defined framework for precision tracking. Traditional ADS-B transponders can extrapolate the latest GPS position information every 200 milliseconds to ensure that the broadcasted message is as accurate as possible. M5L2, however, lacks this capability and has a much higher latency which leads to imprecise position messages. Figure 2.5 shows a comparison of the tracking performance of ADS-B and M5L2. The precision of the position broadcast can be critical for aircraft travelling at hundreds and sometimes thousands of knots.

Although M5L2 is a defined standard, there is no identified mandate date for its deployment [62]. M5L2 may not be fielded for several years or decades, leaving the Air

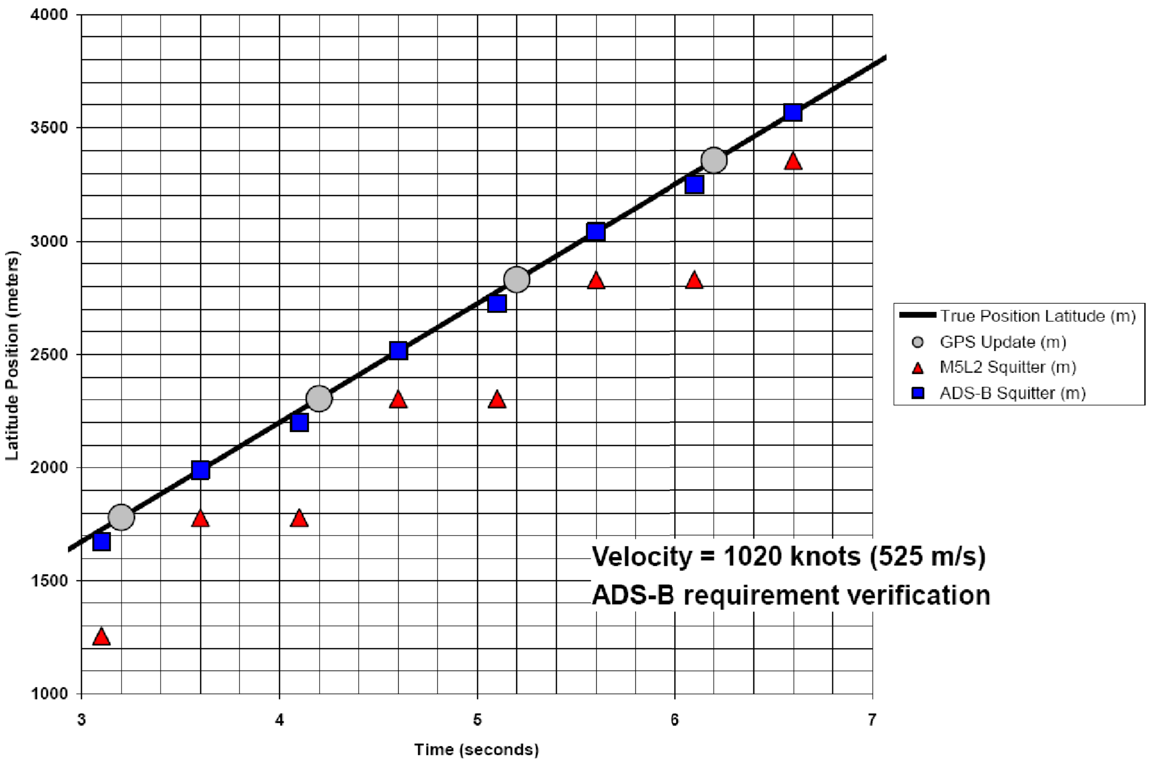


Figure 2.5: Position Extrapolation for Tracking: ADS-B vs. M5L2-B [62].

Force with the options of either not complying with the 2020 congressional mandate, or operating with an insecure ADS-B system and sacrificing OPSEC.

2.4 The Automatic Identification System

The AIS is the naval homologue of ADS-B and is used for collision avoidance, Vessel Traffic Services (VTS), search and rescue, accident investigation and for Aids to Navigation (AtoN). In 2000, the International Maritime Organization (IMO) mandated the fitting of the Automatic Identification System on all international voyaging ships by 1 July 2004. Subsequently, the requirement was expanded to all commercial ships with gross tonnage of 300 or more tons, and all passenger ships regardless of size [46]. To resolve the OPSEC risk, the NATO sought to add encryption to AIS. Standards agencies started devoloping a secure AIS for warships in 2004, resulting in the Warship-Automatic

Identification System (W-AIS) which is defined in the NATO Standard Agreement (STANAG) 4668 [46].

The NATO released a first edition of STANAG 4668 in 2007, and a revised edition in 2010. To reduce the need for a costly acquisition process, W-AIS is based on the commercial AIS transponder specifications defined in ITU-R M.1371 [30] with add-on encryption units. According to STANAG 4668, the W-AIS may be operated in Protected, Active, Passive or Off modes. In the Protected mode of operation, “The W-AIS shall receive and transmit information protected by commercial grade encryption. The W-AIS shall still receive all unencrypted transmissions from commercial AIS equipped ships within range” [46]. The W-AIS may implement the Blowfish open source commercial encryption or the AES algorithm for protection of data, as shown in Figure 2.6. The encrypted content is transmitted in a time slot designated for its specific message format in the AIS Time-Division Multiple Access (TDMA) scheme.

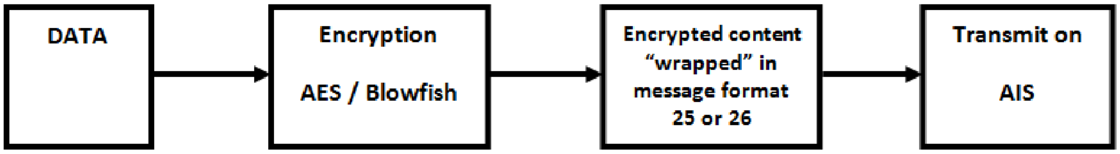


Figure 2.6: W-AIS Block Diagram. Modified from [46].

AES and Blowfish are symmetric encryption schemes requiring each party in the trust ring to know the pre-shared encryption key. With this scheme, warships and other military vessels are able to form trusted networks for sensitive operations, while maintaining situational awareness of other ships in the vicinity. A key-attribute that enables the use of the AES and Blowfish algorithms in W-AIS, is the standard 256-bit size of the AIS message [30].

ADS-B transmits a 112 bit message which is not suitable for encryption with traditional encryption algorithms. Encryption algorithms are typically designed to work with message blocks of size 64 or 128 bits, and pad non-standard length messages to a round multiple of the block size. Padding is not an option with ADS-B because of requirements for compatibility with legacy Mode S transponders.

2.5 Format-Preserving Encryption

Encryption is the mathematical manipulation of data in such a way as to make it unintelligible to unauthorized parties, yet recoverable by the intended recipients. In the basic communication scenario, depicted in Figure 2.7, there are two parties, Alice and Bob, who want to communicate with each other over an unsecured channel. A third party, Eve, is a potential eavesdropper who may gain access to messages sent over the unsecured channel. When Alice wants to send a message to Bob, called the plaintext, she encrypts it using a method prearranged with Bob. When Bob receives the encrypted message, called the ciphertext, he changes it back to the plaintext using a decryption key [63].

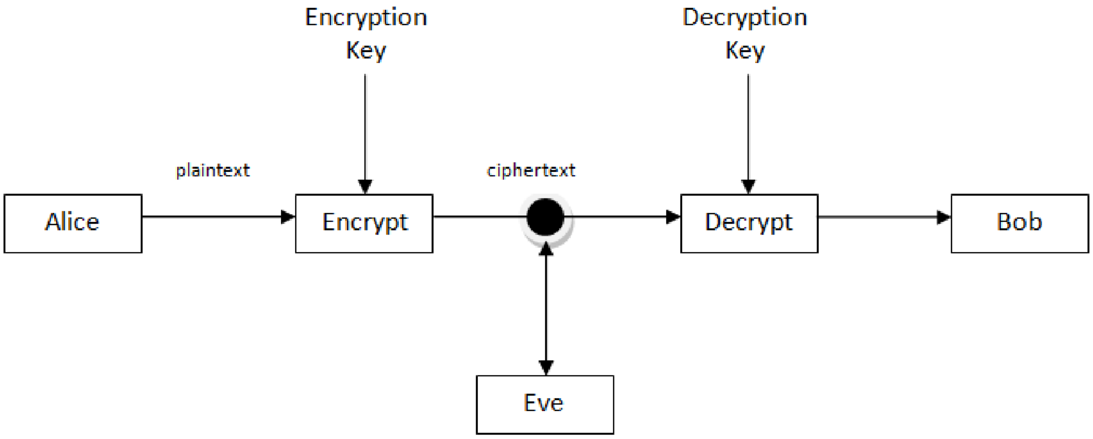


Figure 2.7: The Basic Communication Scenario for Cryptography [63].

Many encryption algorithms are widely available today and used in information security as shown by the hierarchy in Figure 2.8. They can be categorized into symmetric (Private-key) and asymmetric (Public-key) algorithms. In symmetric key encryption, only one key is used for encryption and decryption. The key must be distributed offline before transmission between Alice and Bob. In asymmetric encryption, two keys are used. A public key is used for encryption and a private key is used for decryption, with each party having a unique key set. This resolves the problem of key distribution, but requires more complex and computationally intensive mathematical operations.

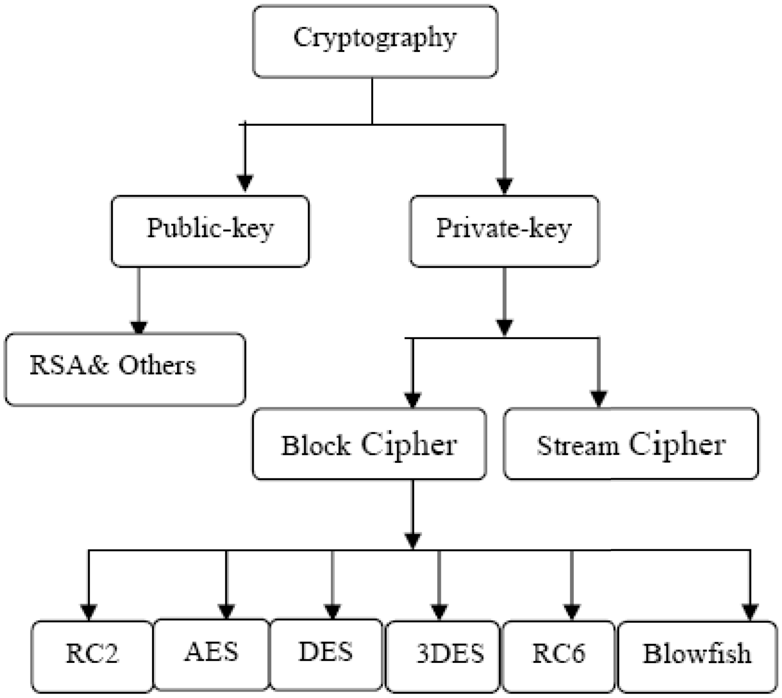


Figure 2.8: Hierarchy of Modern Cryptography [13].

Within symmetric key encryption, there exist block ciphers and stream ciphers [63]. Stream ciphers encipher the plaintext one digit at a time and concatenate these independent encryptions to form the ciphertext. Stream ciphers are fast but are prone to weaknesses in integrity protection and authentication. On the other hand, block ciphers encipher fixed-length groups of plaintext digits. Block ciphers are slower but their mechanism ensures the security properties of confusion and diffusion. Confusion means that the key does not relate in a simple way to the ciphertext, and refers to making the relationship as complex as possible between the key and the ciphertext by using the key non-uniformly throughout the encryption process. Diffusion means that changing a single character in the plaintext causes several characters in the ciphertext to change, and vice versa. For a stream cipher to be secure, its keystream must have a large period; meaning that a complex key management scheme is required. Block ciphers are predominantly used in modern day cryptography [63], and three in particular - AES, 3DES, and Skipjack - are recommended for use by the NIST [10].

In the context of ADS-B, previous research [20, 31] has unanimously supported the use of a symmetric algorithm. Using an asymmetric algorithm in the NAS would require each aircraft to identify and maintain awareness of neighboring traffic and ground stations in order to select the pertinent keys for encrypting each message transmission [31]. The associated overhead would likely negate the benefits that ADS-B affords by impeding the message transmission rate [20]. Symmetric algorithms are computationally more efficient than asymmetric algorithms; however, key management becomes a greater concern. Any compromise of the key at any point compromises the fidelity of the entire security system. While the logistics of key management will need to be addressed, its implementation is beyond the scope of this research.

In determining an appropriate symmetric algorithm, ADS-B system functionality must be considered. Frequent ADS-B broadcasts include only minor changes to data fields. To protect the system from known plaintext attacks, it is necessary that repeated patterns in plaintext be diffused in the ciphertext. As such, a block cipher algorithm is most appropriate for use in the ADS-B operating environment [20, 31].

The aforementioned block cipher algorithms approved by the NIST, such as AES are predicated on encrypting precisely 64, 128 or 256 bit blocks [13, 63] . The 128-bit message space was conventional for the cryptographic community and convenient for the AES designers [52]. Messages that do not fit the prescribed block size are typically padded or truncated. This is incompatible with ADS-B, as the underlying hardware and protocol frameworks are designed specific to the 112 bit fixed data length. Indeed, an encryption scheme that supports arbitrary block size is required.

2.5.1 History of FPE.

FPE is an encryption scheme that supports arbitrary block sizes. Given any finite set of symbols, FPE transforms data that is formatted as a sequence of symbols in such a way that the encrypted form of the data has the same format and length as the original data. Encrypting a 16-decimal-digit plaintext such as a credit-card number results in a ciphertext that is also a 16-decimal-digit number. A shared key K is used to control the encryption. Syntactically, a map $E : K \times X \rightarrow X$ is sought in which X encodes 16-digit strings and $E_K = E(K, \cdot)$ is a permutation for each K $\in K$ [52].

The origins of the FPE problem can be traced back 33 years. In 1981, the US National Bureau of Standards (later to become NIST) published FIPS 74 [42], an appendix of which describes an approach for enciphering arbitrary strings over an arbitrary alphabet. The scheme was subsequently proven to be insecure [2]. It was not until 1997, that Brightwell and Smith clearly and generally described the FPE problem and its utility, which they called at the time “datatype-preserving encryption” [3]. Black and Rogaway brought the problem back to the attention of the cryptographic community in 2002 [3]. In 2003, Terrence Spies proposed the FFSEM [59] FPE algorithm to NIST.

2.5.2 Premise of FPE.

The development of FPE was motivated by the desire to add security to legacy protocols and systems. In such systems, one of the barriers to the adoption of effective encryption methods is the cost of modifying databases and applications to accommodate encrypted information. First, applications often expect input in specific formats, so the encrypted data must retain the data format. Second, data such as Social Security Numbers or personal account numbers are often used as keys or indices in the database, so randomization of these fields requires significant schema changes [52, 60]. Black and Rogaway describe the need for a deterministic FPE algorithm in [3], meaning every time a particular message X is encrypted with a particular key K, the exact same ciphertext Y is created and no additional information is needed to reverse the process.

Black and Rogaway proposed three methods for FPE: a Prefix method, a Cycle-Walking Cipher and a Feistel Construction [3]. The first two methods have strong security bounds, but are targeted for tiny-space and small-small space messages. For tiny-space FPE, the size of the message space $N = |X|$ is so small that it is feasible to spend $O(N)$ time or $O(N)$ space in order to encrypt or decrypt a point [52]. For small-space FPE, the size of the message space $N = |X|$ is at most 2^w where w is the block size of the block cipher underlying the FPE scheme [40]. AES is often used as the block cipher, so $w = 128$ bits and $N = 2^{128} \approx 10^{38.5}$ is the cutoff for “small” . The third method encrypts a much wider variety of data using the Feistel construction first formally examined by Luby and Rackoff in 1988 [36]. The Feistel construction has the desirable property that its ciphers can be proven to reduce to the underlying block cipher used in the round function [48].

FPE schemes are generalizations of block ciphers, and rely on time tested, community-engendered confidence in the underlying cipher for security merit. The Feistel method has been the most well-known approach for making block ciphers for 35 years [41]. It turns a block cipher into a pseudorandom function while maintaining its strong provable-security guarantees, and has been standardized by ANSI, ISO and NIST

[67]. Effective attacks on Feistel-based constructions seldom attempt to attack the Feistel structure itself, but look instead, for defects in the round function used [52].

In 2010, Mihir Bellare, Terence Spies, and Phillip Rogaway submitted to the NIST specifications for FFX [2], a Format-preserving Feistel-based encryption scheme. The X stands for the various implementation forms of the algorithm tailored to suit each particular application. Note that FFX was derived from the previous FFSEM proposal by Spies.

2.5.3 Security of FFX within the ADS-B environment.

A recent study by Finke [20] tested the FFX algorithm as proposed in [2] within the ADS-B environment. The algorithm's ability to encrypt and mask predictable ADS-B messages was measured using classical Shannon entropy. Experimental results demonstrated the utility of FFX encryption based upon its ability to confuse and diffuse ADS-B message content.

In July 2013, NIST released a draft recommendation for format-preserving modes of operation for block ciphers [10]. The release recommended two additional algorithms in addition to a modified version of FFX, along with specified parameters to narrow variances in implementation.

2.6 NIST Recommendations for Format-Preserving Encryption

The NIST is responsible for developing information security standards and guidelines, including minimum requirements for Federal information systems [44]. In July 2013, the NIST released a draft of Special Publication 800-38G (SP800-38G) [10] for public comment, specifying three methods for format-preserving encryption, called FF1, FF2, and FF3. Each of these methods is a mode of operation of the AES algorithm, which is used to construct a round function within the Feistel structure for encryption as shown in Figure 2.9.

The three modes specified, FF1, FF2 and FF3, were submitted to NIST under the names FFX[Radix] [2], VAES3 [65], and BPS-BC [4], respectively. FF1 supports the greatest range of lengths for the protected data and the tweak [10]. FF2 generates a subkey for the block cipher in the Feistel round function, which can help protect the original key from side-channel analysis. FF3 offers the lowest round count, eight, compared to ten for FF1 and FF2, and is the least flexible in the tweaks that it supports.

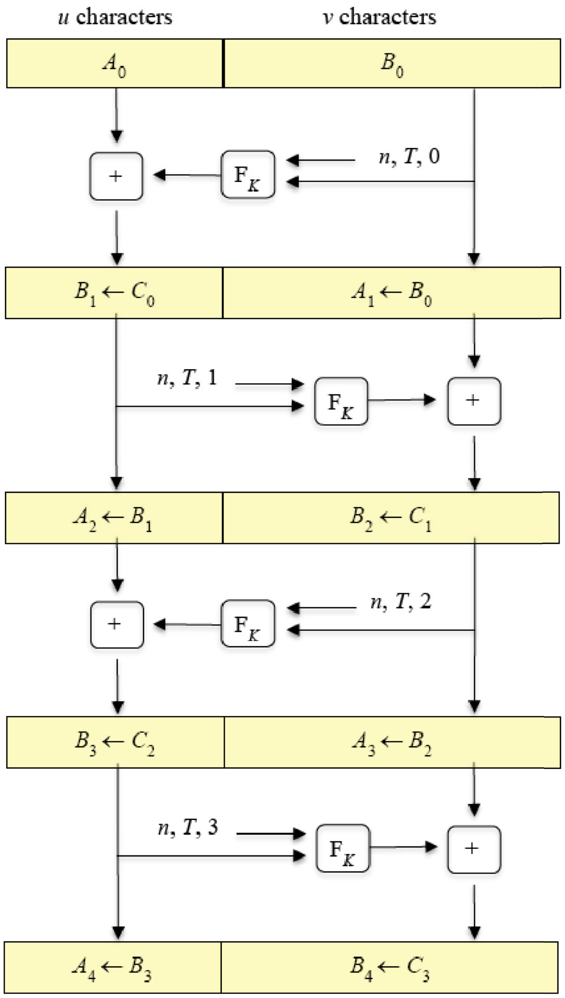


Figure 2.9: Illustration of the Feistel Structure of FPE [10].

2.6.1 FF1.

The FF1 algorithm is derived from Bellare, Rogaway, and Spies' FFX [2] algorithm. The designers of FFX made it customizable with nine alterable parameters. Certain parameter collections such as FFX-A2 and FFX-A10 were specified to encipher binary strings of 8 to 128 bits, and decimal strings of 4 to 36 digits, respectively [2]. In the original algorithm, the user could choose between an arbitrarily unbalanced or alternating Feistel structure. The NIST-specified FF1 narrowed the scope of the algorithm to use 10 rounds of encryption and a maximally-balanced alternating-Feistel structure.

Pseudocode of the FF1 encryption algorithm is provided in Algorithm 1. The parameters *radix, minlen, maxlen,* and *maxTlen* in FF1.Encrypt and FF1.Decrypt shall meet the following requirements:

- $radix \in [2..2^{16}]$;
- $radix^{minlen} \geq 100$;
- $minlen \geq 2$;
- $maxlen < 2^{32}$;
- $maxTlen < 2^{32}$.

FPE algorithms can encrypt finite character strings of arbitrary length and format. Each character or symbol in the character string may be from an arbitrary set of symbols or alphabet. *Radix* represents the number of characters in a given alphabet. *minlen* and *maxlen* represent the number of symbols or length of a character string.

The FF1 algorithm can encrypt alphabets of base 2 to base 2^{16}. The character string must be between 2 and 2^{32} characters in length. There must be at least 100 possible permutations for the chosen base and length. The FF1 algorithm takes a *tweak* in addition to the secret key. The *tweak* is an input parameter to the encryption and decryption functions whose confidentiality is not protected by the mode. It serves to vary the

Algorithm 1 FF1.Encrypt(K,T,X) [10].

Prerequisites:
Approved, 128-bit block cipher, $CIPH$;
Key, K, for the block cipher;
Base, $radix$, for the character alphabet;
Range of supported message lengths, $[minlen..maxlen]$;
Maximum byte length for tweaks, $maxTlen$.

Inputs:
Character string, X, in base $radix$ of length n such that $n \in [minlen..maxlen]$;
Tweak T, a byte string of byte length t, such that $t \in [0..maxTlen]$.

Output:
Character string, Y, such that $LEN(Y) = n$.

Steps:

```
Let $u = \lfloor n/2 \rfloor; v = n - u$.
Let $A = X[1..u]; B = X[u+1..n]$.
Let $b = \lceil \lceil vLOG_2(radix) \rceil /8 \rceil; d = 4\lceil b/4 \rceil + 4$.
Let $P = [1]^1 \, || \, [2]^1 \, || \, [1]^1 \, || \, [radix]^3 \, || \, [10]^1 \, || \, [u \ mod \ 256]^1 \, || \, [n]^4 \, || \, [t]^4$.
for $i \leftarrow 0$ to 9 do
    Let $Q = T \, || \, [0]^{(-t-b-1) mod 16} \, || \, [i]^1 \, || \, [NUM_{radix}(B)]^b$.
    Let $R = PRF(P \, || \, Q)$.
    Let $S$ be the first $d$ bytes of the following string of $\lceil d/16 \rceil$ blocks:
      $R \, || \, CIPH_k(R \oplus [1]^{16}) \, || \, CIPH_k(R \oplus [2]^{16}) \, || \, .. \, || \, CIPH_k(R \oplus [\lceil d/16 \rceil - 1]^{16})$.
    Let $y = NUM_2(S)$.
    If $i$ is even, let $m = u$; Else, let $m = v$.
    Let $c = (NUM_{radix}(A) + y)$ mod $radix^m$.
    Let $C = STR^m_{radix}(c)$.
    Let $A = B$.
    Let $B = C$.
end for
Return $A \, || \, B$.
```

ciphertext for plaintext with small *radix*^{minlen}. The base of the *tweak* is the same as the *radix*. The maximum length of the *tweak* is 2^{32}.

The ADS-B message is represented in binary and has a *radix* value of 2. The message is 112 charcter strings in length which meets the requirements of FF1. The tweak used is also required to be in base 2.

2.6.2 FF2.

The FF2 algorithm is derived from VAES3 [65] submitted to NIST by Joachim Vance. The FF2 algorithm generates a subkey for the blockcipher in the Feistel round function, which can help protect the original key from side-channel analysis [10]. FF2 also has an additional parameter, *tweakradix*, for the choice of the base for tweak strings.

The pseudocode for the FF2 encryption algorithms is provided in Algorithm 2. The parameters *radix, tweakradix, minlen, maxlen,* and *maxTlen* in FF2.Encrypt and FF2.Decrypt shall meet the following requirements:

- $radix \in [2..2^8]$;
- $tweakradix \in [2..2^8]$;
- $radix^{minlen} \geq 100$;
- $minlen \geq 2$;
- $maxlen \leq 2\lfloor 120/LOG_2(radix) \rfloor$ if *radix* is a power of 2;
- $maxlen \leq 2\lfloor 98/LOG_2(radix) \rfloor$ if *radix* is not a power of 2;
- $maxTlen \leq 2\lfloor 104/LOG_2(tweakradix) \rfloor$.

The FF2 algorithm can only encrypt character strings of base less than 2^8. The *tweakradix* must meet the same constraint. There must be at least 100 possible permutations for the chosen base and length. For the ADS-B message with *radix* 2 and *tweakradix* 2, the FF2 algorithm is limited to a maximum plaintext length of 240 and a maximul *tweak* length of 208. The ADS-B message fits within the parameters of the FF2 algorithm.

Algorithm 2 FF2.Encrypt(K,T,X) [10].

Prerequisites:

Approved, 128-bit block cipher, $CIPH$;
Key, K, for the block cipher;
Base, $radix$, for the character alphabet;
Base, $tweakradix$, for the tweak character alphabet;
Range of supported message lengths, $[minlen..maxlen]$;
Maximum supported tweak length, $maxTlen$.

Inputs:

Numeral string, X, in base $radix$ of length n such that $n \in [minlen..maxlen]$;
Tweak numerical string, T, in base $tweakradix$ of length t such that $t \in [0..maxTlen]$.

Output:

Character string, Y, such that $LEN(Y) = n$.

Steps:

```
Let $u = \lfloor n/2 \rfloor; v = n - u$.
Let $A = X[1..u]; B = X[u+1..n]$.
**If** $t > 0, P = [radix]^1 \parallel [t]^1 \parallel [n]^1 \parallel [NUM_{tweakradix}(T)]^{13}$;
**Else** $P = [radix]^1 \parallel [0]^1 \parallel [n]^1 \parallel [0]^{13}$.
Let $J = CIPH_K(P)$.
**for** $i \leftarrow 0$ to 9 **do**
    Let $Q \leftarrow [i]^1 \parallel [NUM_{radix}(B)^{15}$.
    Let $Y \leftarrow CIPH_J(Q)$.
    Let $y \leftarrow NUM_2(Y)$.
    **If** $i$ is even, let $m = u$; **Else**, let $m = v$.
    Let $c = (NUM_{radix}(A) + y)$ mod $radix^m$.
    Let $C = STR^m_{radix}(c)$.
    Let $A = B$.
    Let $B = C$.
**end for**
Return $A \parallel B$.
```

2.6.3 FF3.

The FF3 algorithm is equivalent to the BPS-BC component of BPS [4], instantiated with a 128-bit block and limited to tiny and small space messages [10].

The pseudocode for the FF3 encryption algorithm is provided in Algorithm 3. The parameters *radix, minlen,* and *maxlen* in FF3.Encrypt and FF3.Decrypt shall meet the following requirements:

- $radix \in [2..2^{16}]$;
- $radix^{minlen} \geq 100$;
- $minlen \geq 2$;
- $maxlen \leq 2 \left\lfloor LOG_{radix}(2^{96}) \right\rfloor$.

The FF3 algorithm does not employ a *tweak*. It can encrypt alphabets of base 2 to base 2^{16}. There must be at least 100 possible permutations for the chosen base and length. The character string must be between 2 and $2 \left\lfloor LOG_{radix}(2^{96}) \right\rfloor$ characters in length. For an ADS-B message of *radix* 2, the FF3 algorithm is limited to a maximum plaintext length of 192 characters.

2.7 Software Validation

In related research [20], Finke tested the FFX-A2 encryption algorithm on ADS-B data. That research verified the merits of the algorithm's diffusion characteristics vis-a-vis the incrementally changing nature of ADS-B traffic. She employed Shannon's classical measure of entropy to evaluate the security of the ciphertext.

During the evaluation of candidates for the Advanced Encryption Standard in 1999, one of the criteria used was a demonstrated suitability as random number generators. That is, the evaluation of their output utilizing statistical tests should not provide any means by which to computationally distinguish them from a truly random source. The statistical tests used by the NIST to evaluate the candidates were: frequency test, block

Algorithm 3 FF3.Encrypt(K,T,X) [10].

Prerequisites:

Approved, 128-bit block cipher, $CIPH$;
Key, K, for the block cipher;
Base, $radix$, for the character alphabet;
Range of supported message lengths, [$minlen..maxlen$], such that $minlen \geq 2$ and $maxlen \leq 2\left\lfloor log_{radix}(2^{96})\right\rfloor$.

Inputs:

Numeral string, X, in base $radix$ of length n such that $n \in [minlen..maxlen]$;
Tweak bit string, T, such that $LEN(T) = 64$.

Output:

Character string, Y, such that $LEN(Y) = n$.

Steps:

```
Let $u = \lceil n/2 \rceil; v = n - u$.
Let $A = X[1..u]; B = X[u + 1..n]$.
Let $T_L = T[0..31]$ and $T_R = T[32..63]$;
**for** $i \leftarrow 0$ to 7 **do**
    **If** is even, let $m = u$ and $W = T_R$, **Else** let $m = v$ and $W = T_L$.
    Let $P = REV([NUM_{radix}(REV(B))]^{12}) \mid\mid W \oplus REV([i]^4)$.
    Let $Y = CIPH_K(P)$.
    Let $y = NUM_2(REV(Y))$.
    Let $c = (NUM_{radix}(REV(A)) + y) \bmod radix^m$.
    Let $C = REV(STR^m_{radix}(c))$.
    Let $A = B$.
    Let $B = C$.
**end for**
Return $A \mid\mid B$.
```

*Where $REV(X)$ reverses the order of characters in the character string X frequency test, cumulative sums test, runs test, long runs of ones test, rank test, spectral test, non-periodic templates test, overlapping template test, universal statistical test, random excursion test, random excursion variant test, Lempel-Ziv complexity test, linear complexity test, and an approximate entropy test [58]. The Rijndael candidate was selected as the AES algorithm, and performed satisfactorily on all the tests.

FPE algorithms are modes of operation of the underlying block, thus FF1, FF2, and FF3 benefit from the statistical characteristics of AES [10, 48, 49] such as entropy. Entropy is a measure of unpredictability or information content. Shannon entropy quantifies the expected value of the information contained in a message and is typically measured in bits per byte [63].

In addition to security considerations, the computational performance of the candidate algorithms is an important criterion. Because of the 2Hz frequency of ADS-B traffic, it is important that the encryption mechanism has small latency in order to meet timing requirements. In measuring the performance of encryption algorithms, several performance metrics are used: encryption time, processing time, and total clock cycles per encryption [23].

2.8 Hardware Validation

Stand-alone ADS-B receivers are available for aerial enthusiasts and researchers to experiment with ADS-B equipment outside of the cockpit of an aircraft. There exist commercial grade products such as the Kinetic Avionics SBS-3 dedicated 1090MHz receiver [33], open source Software-Defined Radio (SDR) projects such as the gr-air-modes GNU radio package [21], and Do-It-Yourself (DIY) kits such as Günter Köllner's Mode S Beast kit [34]. The Mode S Beast, shown in Figure 2.10, employs an FPGA to decode received ADS-B messages.

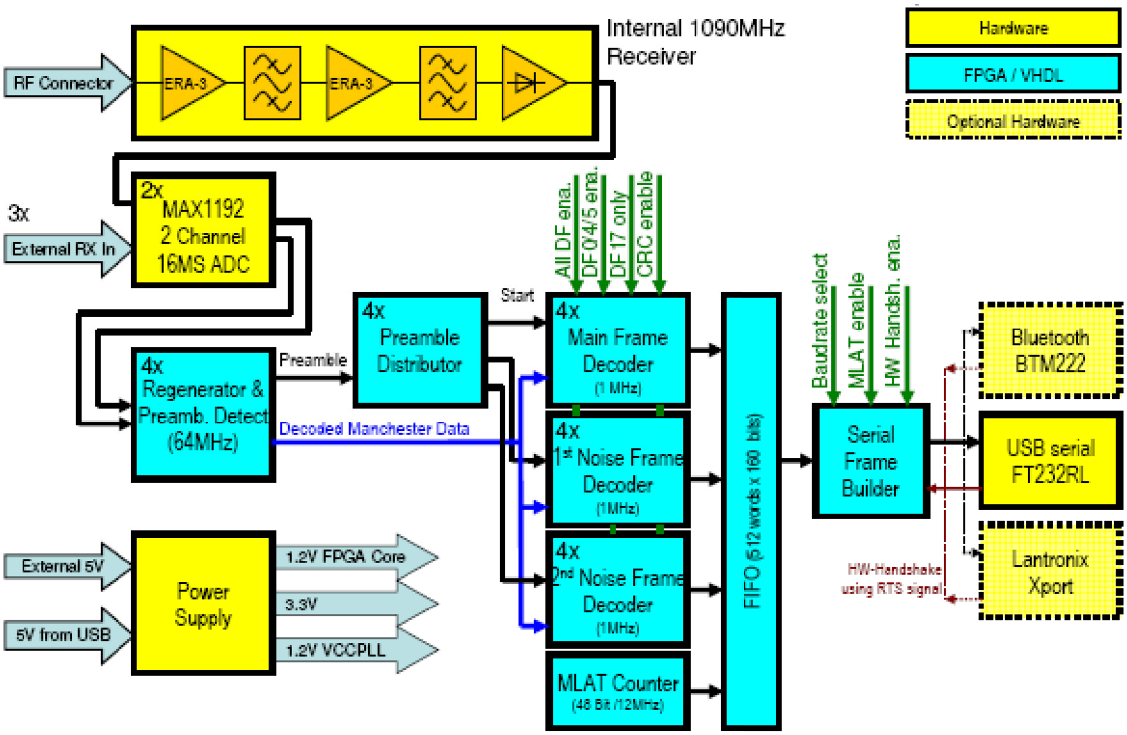


Figure 2.10: Block Diagram of Mode S Beast Receiver by Günter Köllner [34].

Complementarily, researchers have demonstrated tranceivers designed to generate and broadcast spoofed ADS-B messages. For example, an SDR application developed by Magazu, creates and transmits arbitrary ADS-B messages [37]. This application was used to spoof ADS-B messages using Ettus Research's Universal Software Radio Peripheral (USRP) device and the GNU Radio API .

A cryptographic engine implementing FF1, FF2, and FF3 could be used to retrofit security to the ADS-B system and protect the NAS from potentially malicious use of the aforementioned technologies. In order for such a cryptographic engine to be practical, it should integrate seamlessly into the existing infrastructure and cause no adverse changes in performance.

2.8.1 Avionics Requirements.

The term ‘avionics’ is a portmanteau of the words ‘aviation’ and ‘electronics.’ It encompasses the electronic systems used in aircraft to control communications, navigation and flight management systems. The FAA maintains technical standards which regulate the development of safety and mission-critical avionics equipment. The RTCA/DO-254 [53] standard “Design Assurance Guidance For Airborne Electronic Hardware”, regulates hardware and firmware engineering of avionic systems. DO-260A and DO-260B [55] specify “Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance - Broadcast (ADS-B) and Traffic Information Services - Broadcast (TIS-B)” [17]. Among these standards are timing and latency requirements for the ADS-B transponder. DO-260B mandates that the latency of the ADS-B equipment be less than 100ms [55, 62].

The use of FPGAs has been expanding from its traditional role in prototyping to mainstream production. Commercial pressures are driving this change with the intention of reducing design cost and achieving a faster time to market [23]. Major manufacturers of avionic systems are now using FPGAs in their transponders instead of custom ASICs [61].

2.8.2 Performance.

Another criteria the NIST used to evaluate the AES candidate algorithms in 1999 was hardware performance. The Rijndael algorithm was selected partly because it proved to be one of the fastest and most efficient algorithms, and was easily implemented on a wide range of platforms [39]. When evaluating the speed and efficiency of a given hardware implementation, the throughput, latency and hardware resources required are considered the most critical parameters [11].

A number of different architectures can be considered when implementing an encryption algorithm in hardware or on an FPGA. Iterative Looping (IL) is where only one round is designed, hence for an n-round algorithm, n iterations of that round are carried out to perform an encryption. Loop Unrolling (LU) involves the unrolling of multiple rounds. Pipelining (P) is achieved by replicating the round and placing registers between each round to control the flow of data. A pipelined architecture generally provides the highest throughput. Sub-Pipelining (SP) is carried out on a partially pipelined design when the round is complex. It decreases the pipeline's delay between stages but increases the number of clock cycles required to perform an encryption [11, 39].

2.9 Summary

The FAA's NextGen will provide a much needed upgrade to the antiquated ATC system. The ADS-B system will provide enhanced surveillance accuracy, improve situational awareness for ground and aircrew, and further the evolution of Air Traffic Control towards Free Flight. Recent advancements in the field of cryptography have provided tools to encrypt the ADS-B message, and help improve OPSEC for aircraft conducting sensitive operations. The NIST has recommended three algorithms for use as Format-Preserving modes of AES. Using the information gained through this literature review, the FF1, FF2 and FF3 algorithms can be tested for use within the ADS-B environment. The performance of each algorithm will be tested in software and hardware, with representative ADS-B data.

III. Methodology

THIS chapter describes the experimental design and methodology used to test the NIST recommended FPE algorithms for use within the ADS-B environment.

3.1 Experimental Design

The goal of this research is to determine the suitability of the FF1, FF2, and FF3 algorithms for encryption of ADS-B messages, with regards to security and performance.

To attain the first objective, three sets of experiments are designed to test the hypothesis suggested by the algorithm designers and NIST in [10], that the algorithms inherit the strong security characteristics of the underlying block cipher. NIST has not released details of its internal deliberations and performance assessments of the algorithms. As such, statistical tests are conducted to determine the ability of the FPE algorithms to provide confusion and diffusion to plaintext, and output a ciphertext that is computationally indistinguishable from a random process. A dataset of input plaintext is created with varying levels of entropy, and is independently encrypted with the FF1, FF2 and FF3 algorithms. The algorithms are implemented in C using the PolarSSL AES library [47] and the resulting ciphertext is measured for entropy.

The second objective of this research is to evaluate the hardware performance of the three algorithms by measuring the operational latency and resource utilization of an FPGA implementation. The algorithms are implemented in VHDL, simulated and synthesized on a Virtex-6 FPGA (XC6VLX240T) device using the Xilinx ISE 14.6 suite. A hardware-agnostic design is used in order to mitigate the particular effects of the Xilinx CMOS technology and FPGA architecture. Operational latency is estimated by the number of clock cycles elapsed between the input of a plaintext and the output of its ciphertext. Device utilization is assesed by the number of FPGA components used to synthesize the algorithms.

3.2 Evaluating Entropy

The methodology used to evaluate the security characteristics of the FF1, FF2, and FF3 algorithms builds on research conducted on the FFX algorithm by Finke [20]. Similar to Finke, this research employs randomized experiments to allow the greatest reliability in the statistical measurements of entropy and validity of the security analysis. Table 3.1 lists the experiments conducted. One set of experiments, Fixed Bytes, systematically increases the number and distribution of deterministic bytes in the unencrypted ADS-B message and evaluates the effect of these factors on the entropy of the resulting ciphertext. The second set of experiments, Fixed Fields, evaluates the effect of unchanging data in various ADS-B message fields on the entropy of the encrypted message. Finally, ADS-B messages extracted from the radar track of an aircraft are encrypted with the FF1, FF2 and FF3 algorithms and the resulting ciphertexts are evaluated.

The True Random Number Generator (TRNG) service provided by Random.org [51] is used to create the experimental dataset of ADS-B messages with varying levels of random and deterministic data. The dataset is independently encrypted with the FF1, FF2 and FF3 algorithms. The algorithms are implemented in C and tested on a Dell Precision T7500 machine with dual core Intel Xeon 3.46 GHz processors and 48 GB of RAM.

3.2.1 Software Implementation.

The FF1, FF2, and FF3 algorithms are implemented as described in [10]. All three algorithms require a NIST approved 128-bit block cipher. The block cipher algorithm used in this implementation is 128-bit AES. The block cipher serves primarily as a subcomponent for the Pseudo-Random Function (PRF) . The PRF function employs

Table 3.1: Entropy Experiments.

Fixed Bytes	**Fixed Fields**	**Radar Track**
3 Front	*None*	*WADS track*
3 Random	*Position*	
6 Front	*Position, Altitude*	
6 Random	*Position, Altitude, Address*	
9 Front		
9 Random	*Position, Altitude, Address, Type Code*	
12 Front		
12 Random		
All Random		

a block chaining mode of AES to generate the output of the F-block, as shown in Algorithm 4. The 128-bit key ‘$000102030405060708090a0b0c0d0e0f$’ used in test vectors published by NIST in [43], is employed in the following experiments. The tweak is set to ‘88’ in hexadecimal or ‘10001000’ in binary, the standard value for the first byte of the ADS-B message which contains values for the DF and CA fields.

The cryptography community discourages use of unverified implementations of AES. Thus, PolarSSL, a vetted open source library used by the Dutch government to encrypt its official communications [20], is used in the software implementation. The PolarSSL implementation of the 128-bit Electronic Codebook (ECB) variant of AES is validated through comparison with test vectors published in NIST’s Known-Answer Test [43]. While there are many AES operating modes, the ECB variant is the most suitable for FPE [10]. PolarSSL is implemented in the C language and partly motivates the use of C throughout the research. The C programming language offers low-level data manipulation

Algorithm 4 PRF(X) [10].

Prerequisites:
Approved, 128-bit block cipher, $CIPH$;
Key, K, for the block cipher;

Input:
Nonempty bit string, X, such that $LEN(X)$ is a multiple of 128.

Output:
Block, Y

Steps:

```
Let $m = LEN(X)/128$.
Partition $X$ into $m$ blocks $X_1, ...., X_m$, so that $X = X_1 \parallel ... \parallel X_m$ and $LEN(X_i)128$ for all $i$ from 1 to $m$.
Let $Y_0 = 0^{128}$, and for $j$ from 1 to $m$ let $Y_j = CIPH_K(Y_{j-1} \oplus X_j)$.
Return $Y_m$.
```

and rapid implementation of complex mathematical operations. However, it is limited to the byte as its lowest level of data granularity.

3.2.2 Limitations and Assumptions.

The ADS-B message format is 112 bits, of which the first 5 bits or the Downlink Format (DF) field, signal the message type. The DF data field must be left unencrypted in order for the receiver to properly decode the message [31]. The remaining 107 bits are available for encryption, but the non-standard message width is incompatible with the primitive data types of C.

Pilot experiments attempted to construct data structures in C to efficiently store the 107 bits. The underlying language structure relies on byte alignment for CPU optimization, and thus, pads all data types to an even byte width. Given these limitations, this research adheres to the methodology established by Finke in [20], and encrypts only 104 of the 107 encrypt-able bits of the ADS-B message as shown in Figure 3.1. The 104-bit message width allows for a balanced Feistel structure, and can be split into balanced halves.

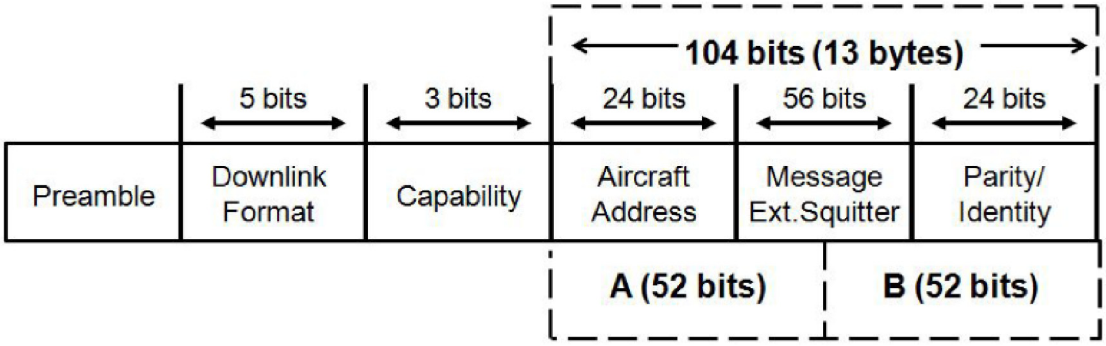


Figure 3.1: Encrypt-able ADS-B fields [20].

The resulting ADS-B ciphertext leaves unencrypted the DF and Capability (CA) fields. The DF field determines which type of message ensues - DF19 or DF17. DF19 is reserved for military use; however, no specifications have been standardized for the ensuing message and it is not currently used in fielded systems [62]. The DF17 message type is exclusively considered because of its prevalence in GA and commercial aviation. The CA field indicates the ability of the emitting transceiver to transmit on the ground or airborne, and whether an emergency or priority alert is active. Of the five available CA codes, code ‘5’ is used indicate an airborne aircraft with full communications capability [55]

The 104-bit encrypted portion of the message contains the Aircraft Address (AA), Message Extended Squitter (ME), and Parity/Interrogator Identity (PI) fields. The AA field contains the 24 bit ICAO address of the aircraft. The ME field contains the 56 bit Extended Squitter (ES) message and reports information such as aircraft position, altitude, and velocity in subfields. The PI field provides data integrity by calculating a Cyclic Redundancy Check (CRC) code based on the value of the preceding fields [37]. The message content is designed to have varying levels of deterministic data, resulting in varying levels of plaintext entropy.

3.2.3 Dataset.

The experimental dataset is generated using data from the Random.org TRNG [51] service. Unlike pseudo-random number generators which use mathematical formulae to generate sequences of numbers that appear random, TRNGs extract randomness from physical phenomena [51]. Random.org generates randomness by measuring atmospheric noise and produces each day one mebibyte (2^{20} bytes) of raw random data. This data is made available to scientists and researchers through their website. The random file of 2013-09-17 was downloaded and used to generate the non-deterministic parts of the plaintext dataset. The dataset contains data for fourteen scenarios, replicated for 20 trials. The plaintext file for each trial of a scenario uses a unique deterministic byte sequence replicated in 4,000 ADS-B message strings. In addition to the generated dataset, 8,866 ADS-B messages are extracted from an observed aircraft track. In total, the experimental dataset contains 1,128,866 unique ADS-B messages. The goal of these experiments is to measure the ability of FPE encryption algorithms to obfuscate ADS-B messages within a representative operational environment.

3.2.3.1 Fixed Bytes.

Consecutive ADS-B messages transmitted by a transiting aircraft contain instances of repeated data since coordinates of the aircraft do not drastically change from one message to the next. Certain data fields such as the Aircraft Address and Type Code (TC) fields may remain constant throughout the duration of a flight. The first set of experiments in this research evaluates the ability of the FPE algorithms to obfuscate arbitrary sequences of repeated data.

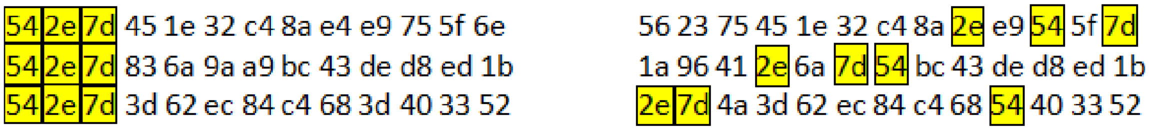
In 1999, NIST tested the ability of candidate algorithms for AES to encrypt a plaintext avalanche constituting of various sequences of random and fixed plaintext *bits* [58]. Given the software limitations, this research tests the ability of the FPE algorithms to encrypt a plaintext avalanche consisting of various sequences of random and fixed plaintext *bytes*. Regardless of the coarser granularity, the fixed bytes methodology provides the desired variation in plaintext entropy and has been employed in research published in a peer-reviewed journal [19]. The Fixed Bytes experiment studies the effect of repetitive and thus predictable input data on the entropy of the ciphertext.

There are two factors in this experiment: the number of deterministic bytes and the distribution of deterministic bytes as shown in Table 3.2. There are four levels for the first factor and two levels for the second. A full factorial experimental design yields eight scenarios. The dataset contains plaintext for eight scenarios in which 3, 6, 9, and 12 bytes of the total 13-byte message are held constant at the front or dispersed randomly throughout the message. The plaintext file for each scenario contains 4000 samples. The deterministic part of the sample ADS-B message replicates the same byte sequence throughout each scenario; however, the non-deterministic part of each message is a unique random sequence extracted from the 2013-09-17 TRNG file. Measurements are taken on the input plaintext and output ciphertext files for each scenario. The experiment is replicated 20 times, consistent with previous research on FFX [20]. Note that the dataset for each trial uses different deterministic and non-deterministic byte sequences.

Table 3.2: Fixed Bytes Levels and Factors.

Factor	**Levels**			
Number of Deterministic Bytes	3 Bytes	6 Bytes	9 Bytes	12 Bytes
Distribution of Deterministic Bytes	Front	Random		

For example, the ‘3 Front’ scenario shown in the left quadrant of Figure 3.2, indicates that the first three bytes are the same for each sample message. The ‘3 Random’ scenario indicates that the three deterministic bytes are randomly dispersed throughout the sample message.



```
54 2e 7d 45 1e 32 c4 8a e4 e9 75 5f 6e          56 23 75 45 1e 32 c4 8a 2e e9 54 5f 7d
54 2e 7d 83 6a 9a a9 bc 43 de d8 ed 1b          1a 96 41 2e 6a 7d 54 bc 43 de d8 ed 1b
54 2e 7d 3d 62 ec 84 c4 68 3d 40 33 52          2e 7d 4a 3d 62 ec 84 c4 68 54 40 33 52
```

Figure 3.2: Sample Plaintext from the Fixed Bytes '3 Front' and '3 Random' Scenarios.

The other six scenarios follow a similar design. The 4,000 messages in each scenario repeat the same deterministic sequence; however, every scenario of the trial uses a unique deterministic byte sequence. The non-deterministic bytes of the sample message are composed of random data extracted from the Random.org sequence of 2013-09-17. Each trial employs new byte sequences in order to insure statistical independence.

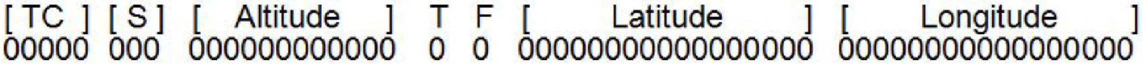
3.2.3.2 Fixed Fields.

The Fixed Fields experiment evaluates the ability of the encryption algorithm to obfuscate ADS-B messages with constant values in certain fields. In this experiment, the values of the Position, Altitude, Address, and Type Code bits are incrementally fixed to reduce entropy in the input message. In flight, these values are often constant or slow to change in messages broadcast by aircraft.

Furthermore, the dataset is restricted to contain ADS-B messages with realistic data in the ME subfields shown in Figure 3.3. In addition to plausible ME data, the PI field contains a valid CRC value. For calculating the CRC, the DF field is set to '10001' in binary or '17' in decimal to indicate a DF17 ES message. The CA field is set to '101' in binary or '5' in decimal to indicate a transponder with "at least Comm-A and Comm-B capability, ability to set code 7, airborne" [55]. These parameters serve to reduce the message space to a subset representative of the ADS-B operating environment [20].

- Altitude

  The altitude component of the ME field consists of 12 bits. The fist 11 bits are used to represent the altitude's numerical value and the final bit indicates whether the



```
[TC ] [S] [  Altitude   ]  T  F  [     Latitude      ]  [     Longitude     ]
00000 000 000000000000  0  0  00000000000000000  00000000000000000
```

Figure 3.3: Structure of Message Extended Squitter (ME) fields [37].

value is expressed in 25 or 100 foot increments [37]. In the decoding process, an additional 1000 feet are added to the indicated altitude. Therefore, this encoding may represent an altitude as high as 205,800 feet which is beyond the operating limit of most aircraft. The dataset for the fixed fields test is limited to represent altitudes commonly used by commercial and high performance aircraft. The 20,000 foot window between FL180 and FL380 is the standard for aircraft operating in the NAS. The value of the altitude field of messages in the fixed fields dataset is restricted to one of 800 values between 18,000 feet and 38,000 feet.

- Position

The geographical position constitutes 34 bits of the ME field and is represented using the Compact Position Reporting (CPR) encoding. CPR was developed for ADS-B messages broadcast on the 1090 MHz Extended Squitter (ES) datalink to reduce the number of bits required to convey participant latitude and longitude while maintaining an accuracy threshold of 5.1 meters. The circumference of the earth is approximately 40,000 kilometers and $(40,000,000m/5.1m) \approx 7,800,000$ discrete position values. Note that 7,800,000 position values would require 23 bits for the longitudinal coordinate but CPR is able to convey position with 17 bits each for latitude and longitude, and 1 format bit . In the CPR coordinate system, the globe is divided into zones. Latitude zones start at the equator and go to both poles. Longitude zones start at the Prime Meridian and proceed eastward around the globe. Latitude and longitude zones are then divided into bins of approximately

5.1 meters in width. Every point on the globe is identified in the CPR coordinate system with a latitude zone index, latitude bin number, longitude zone index, longitude bin number and CPR format (even or odd zone size). This identification number is expressed as a 17 bit sequence. A more detailed explanation can be found in [54].

Messages received by a transceiver necessarily portray a location within its range of reception. According to [17], ADS-B transceivers are required to provide a range of 120NM, and so transmissions decoded by a receiver often originate within a 120NM radius of its position. The Latitude and Longitude ME values are constrained to position coordinates that fit within an area of $120NM^2$.

- Type Code

  The Type Code field consists of the first 5 bits of the ME field, and indicates the type of message that follows. This research focuses solely on airborne position reports for which there are only 14 associate type code values (0, 9-18, 20-22) [20, 55]. One of these values is randomly selected for each simulated ADS-B message.

- Parity/Identity Field

  The PI field is calculated as a Cyclic Redundancy Check (CRC) using the preceding 88 bits and the polynomial shown in Equation (3.1) [37].

$$G(x) = 1+x^3+x^{10}+x^{12}+x^{13}+x^{14}+x^{15}+x^{16}+x^{17}+x^{18}+x^{19}+x^{20}+x^{21}+x^{22}+x^{23}+x^{24} \tag{3.1}$$

3.2.3.3 Radar Track.

The final test uses ADS-B messages generated from real aircraft traffic. An aircraft radar track observed by the Western Air Defense Sector (WADS) was used to create this dataset. The WADS continually monitors the NAS to ensure air sovereignty and strategic air defense. As shown in Figure 3.4, the aircraft took off from Oakland, CA and travelled eastward towards Nebraska. The provided track includes altitude and position information from overlapping radars with 1 to 7 seconds between data points.

The radar coordinates were transformed into ADS-B messages in [20] using code from [37]. The DF and CA fields are held constant similar to the Fixed Fields dataset. The generated plaintext file contains 8,866 unique messages. Given the aircraft's continuous movement, the geographical position varies with each message; however, the altitude changes little due to extended cruise periods at 33,000 and 35,000 feet. This dataset relies on the predictability of the aircraft trajectory to control the message variance factor instead of arbitrary mixtures of deterministic and random data.

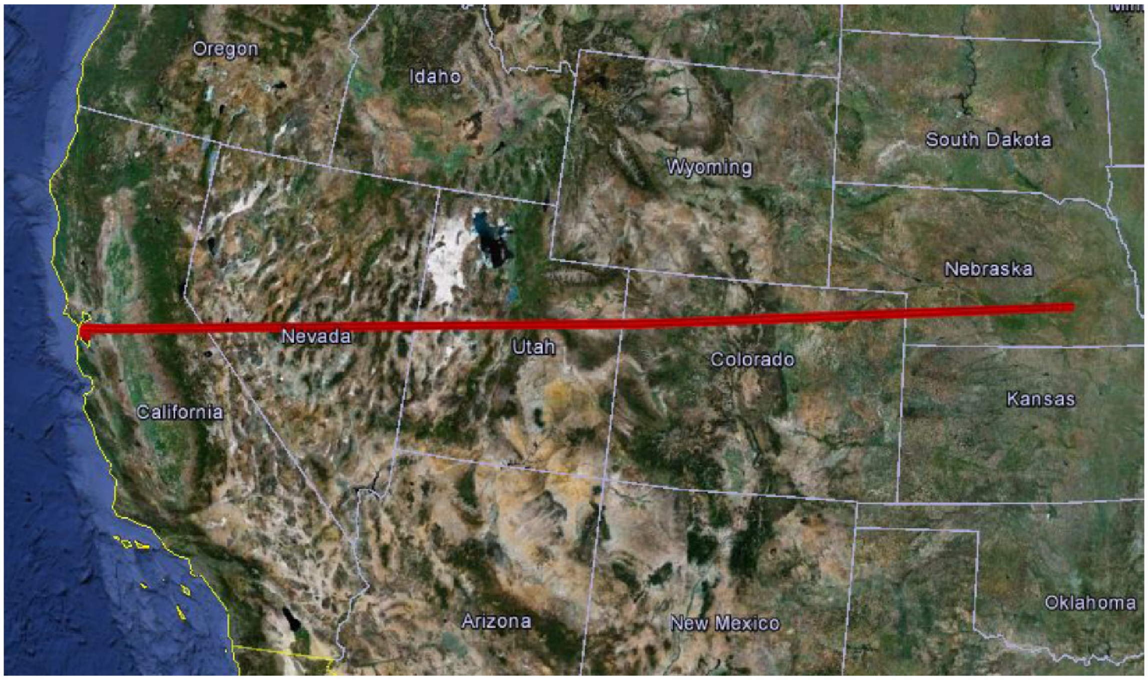


Figure 3.4: Plot of WADS Radar track [20].

3.2.4 Measurements.

When an adversary eavesdrops on secure communication, the encrypted information should appear random. In cryptography, there exist several definitions of security: perfect security, semantic security, and entropic security [6, 63]. An encryption algorithm is perfectly secure if a ciphertext produced using it reveals no information at all about the plaintext. That is, the encryption cannot be broken even when the adversary has unlimited time and computational power. An example of such a cryptanalytically unbreakable cryptosystem is the one-time pad. This theoretical level of security is infeasible to achieve in practice because it requires a key as long as the total length of all messages that are going to be encrypted [63]. On the other hand, semantic security implies that any information revealed about the plaintext cannot be feasibly extracted. That is, any probabilistic, polynomial-time algorithm (PPTA) that is given the ciphertext, and the message length, cannot determine any partial information on the message with non-negligible probability. However, deterministic encryption algorithms such as AES or FPE can never be semantically secure [63]. Entropic security is a weaker definition of security which relaxes the definition to a level where the ciphertext has substantial entropy. The definition of substantial entropy is context-dependent. Nevertheless, random sequences and sequences generated by pseudorandom functions are considered to have high entropy [51, 63]. During the evaluation of candidates for the Advanced Encryption Standard in 1999, one of the criteria used was a demonstrated suitability as a random number generator [58]. Therefore, this research assesses the security of the FPE encryption algorithms by comparison to a random sequence.

3.2.4.1 Shannon Entropy.

Entropy is a measure of unpredictability or information content. It measures both the amount of uncertainty in a distribution before sampling and the amount of information obtained by sampling. This research uses entropy as a measurement of the amount of information that can be gleaned from the encrypted ADS-B message. The entropy $H(X)$ of a variable or distribution is defined in Equation (3.2) [63].

$$H(X) = -\sum_{x \in X} p(x) log_2 p(x) \tag{3.2}$$

Theoretically, a random sequence has perfect entropy because its components are independent. In practice, a pseudorandom sequence generated by a cryptographically secure PseudoRandom Number Generator (PRNG) has substantial but not perfect entropy. The higher the entropy of a sequence, the harder it is to obtain information about the nature of its content. A ciphertext produced by the FF1, FF2, or FF3 algorithm is considered to have high entropic security, if its measure of entropy equals or exceeds that of a random sequence. The entropy of the encrypted ADS-B message is compared to the entropy of an All Random sequence of the same length extracted from the 2013-09-17 TRNG file.

3.2.4.2 ENT Tool.

The ENT tool [66] developed by John Walker at FourmiLab, provides measurements of entropy. The program applies various statistical tests to sequences of bytes stored in files and reports the data's aggregate entropy in bits per byte (bits/byte). The program is useful for evaluating pseudorandom number generators for encryption and other applications where the information density of a file is of interest. As such, this research uses the ENT tool to measure entropy statistics of each trial for every scenario in order to evaluate the pseudo-random characteristics of the FPE algorithms as suggested by [41, 48, 49].

3.3 Evaluating Performance

In 1999, NIST also used hardware performance as a primary criterion for evaluating the AES candidate algorithms. The Rijndael algorithm was selected partly because it proved to be one of the fastest and most efficient algorithms, and was easily implemented on a wide range of platforms [11]. A number of different architectures can be considered when implementing an encryption algorithm in hardware or on a Field Programmable Gate Array (FPGA). This research employs a Pipelined implementation of 128-bit AES and an Iterative Looping (IL) architecture for the Feistel structure of FPE.

3.3.1 Hardware Implementation.

The implementation of the underlying AES cipher follows a pipelined architecture. A pipelined architecture provides distinct hardware for every stage of AES with specific registers between each stage. This allows the system to produce one ciphertext every clock cycle in steady state for a high throughput rate, but utilizes considerable hardware resources. AES is a complex algorithm and improper implementation can cause serious security vulnerabilities. This research makes use of an AES core that was tested and verified in [35]. The core was designed by Pranav Patel and is copyrighted to AFIT. Table 3.3 shows the performance characteristics of the AES core, benchmarked on the Xilinx Virtex-6 FPGA. These size and speed measurements are used as a baseline for comparison of FF1, FF2, and FF3. The high throughput of the pipelined core exceeds the requirements for the 2Hz data rate of ADS-B messages, which ensures that the only factors affecting the operational latency of the design are its maximum frequency and the number of clock cycles per encryption.

The FPE algorithms are implemented according to an Iterative Looping architecture. The IL architecture reuses hardware resources at the cost of overall throughput. Throughput is the average rate of data through a node [39]. For use within the ADS-B environment, the throughput of the cryptographic core must be higher than the 2 Hz message rate. The low data rate of ADS-B does not require an architecture optimized for throughput. As such, only one round of the algorithm is implemented and control logic is used to manage data flow for a complete encryption cycle, as shown in Figure 3.5. A new

Table 3.3: Performance of AES Core.

Algorithm	**AES**
Number of occupied Slices	1,864
Number of Slice Registers	5,801
Number of Slice LUTs	3,452
Number of 18K block RAMs	172
Maximum Frequency (MHz)	336.315
Clock Cycles per Round	3
Clock Cycles per Encryption	31

round does not begin until after data for the previous round has traversed the entire FPE Round block.

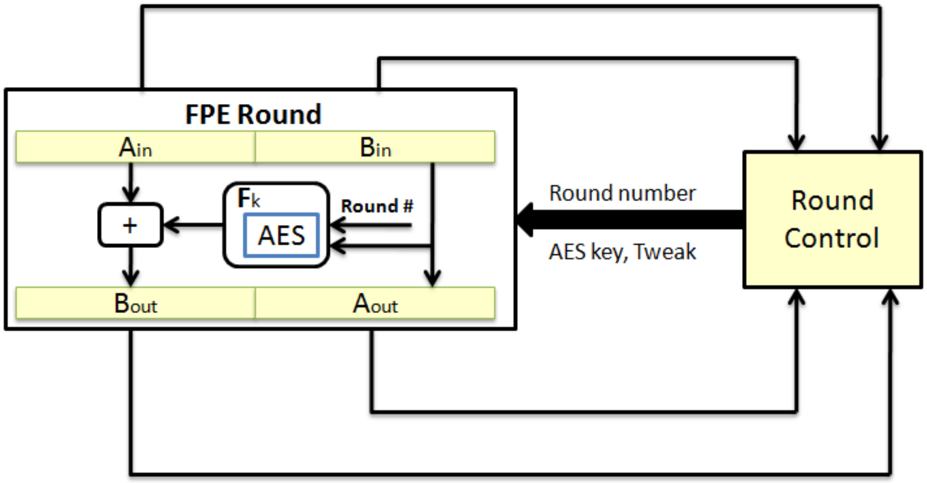


Figure 3.5: Illustration of the Iterative Looping Implementatin of FPE.

The pseudocode description provided by NIST is primarily intended for implementation in software. Certain operations in the pseudocode depend on previous ones, which requires carefully synchronized logic when implemented in hardware. Each algorithm's pseudocode is expanded to identify parallelizable modules and blocks that can be implemented with combinational logic. Function calls to AES or PRF within the F-block of each round must be synchronized to ensure that the output of one block is valid when passed to the next AES block. In this implementation, a shift register is used to delay the start signal of the cascaded AES block until the number of clock cycles required by the first block has expired.

3.3.2 Performance Metrics.

The FF1, FF2, and FF3 algorithms are coded in VHDL, simulated, Placed and Routed (PAR), and synthesized on a Virtex-6 (XC6VLX240T) device using the Xilinx ISE 14.6 design suite. To facilitate comparison with the software implementations, the hardware implementations are designed to process 104-bit messages. No FPGA device-specific features, such as the Virtex-6's DSP48E1 Digital Signal Processing slice, are used that would prevent an equivalent implementation on a different brand or model FPGA. Behavioral simulation tests, Post-PAR static timing analysis and device utilization analysis are performed on each design.

The device utilization analysis provides the following metrics: Number of Slice Registers, Number of Slice LUTs, Number of occupied Slices, and Number of 18Kb block RAMs. Slices are the basic building block components in the Xilinx FPGA fabric. Each slice contains four Look-Up Tables (LUTs) which are used to implement combinatorial logic such as AND gates, OR gates and other boolean functions. In addition to LUTs, slices also contain eight flip-flop registers which hold state and are used to implement sequential logic. In the device utilization report, any slice that is used even partially is counted towards the number of occupied Slices. A design may be fit into fewer slices if necessary, but mapping unrelated logic into the same slice may limit the ability of the placer to meet timing constraints [68]. The Virtex-6 provides 18Kb and 36Kb blocks of RAM that may be used to store data.

The Post-PAR static timing analysis provides the Maximum Frequency (MHz) metric. The maximum frequency is based on the worst path delay found during synthesis, and indicates the fastest frequency at which a signal may be toggled given this constraint.

A behavioral simulation test is conducted using the Xilinx ISE Simulator (ISIM). The results of a behavioral simulation can be replicated on any simulation tool by using the same test bench. The stimuli used in the test bench are a 50 MHz clock, and sample plaintext ADS-B messages taken from the entropy dataset. The operational latency of each algorithm is measured by monitoring input ready and output ready signals in the simulation waveforms. The number of clock cycles elapsed between the input of a plaintext and the output of its ciphertext is counted in the waveform. The numbers of clock cycles required for the completion of one round and for a complete encryption cycle are reported.

3.4 Cryptographic Engine

The Bump-in-the-Wire cryptographic engine intercepts the unencrypted ADS-B Out message at the output of the transponder and encrypts it before transmission. Figure 3.6 and Figure 3.7 show a block diagram of the ADS-B system without and with the proposed encryption engine, respectively. Such a design requires minimal redesign and can be retrofitted to existing transponders. The cryptographic engine also detects and decrypts encrypted ADS-B In messages between the antenna and the legacy transponder.

3.5 Summary

This research evaluates the security and hardware performance profiles of the NIST recommended FPE algorithms. The ability of the algorithms to obfuscate messages is tested with three experimental datasets. The experimental datasets are designed to

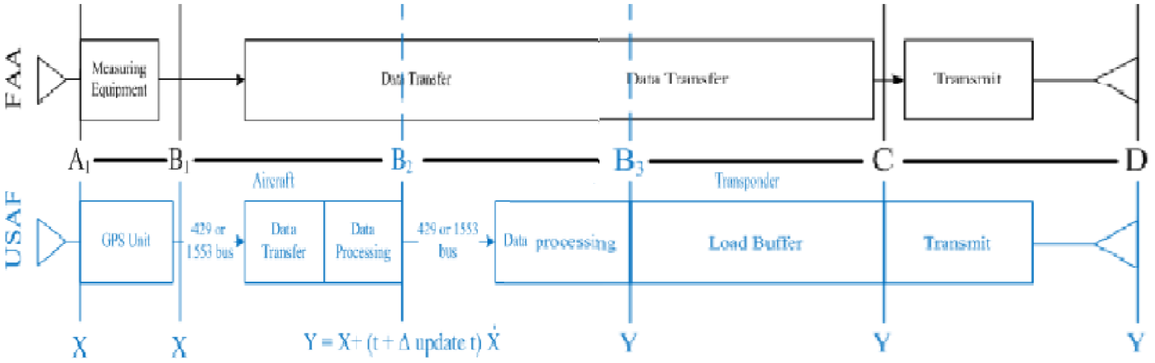


Figure 3.6: Block Diagram of ADS-B avionics [62].

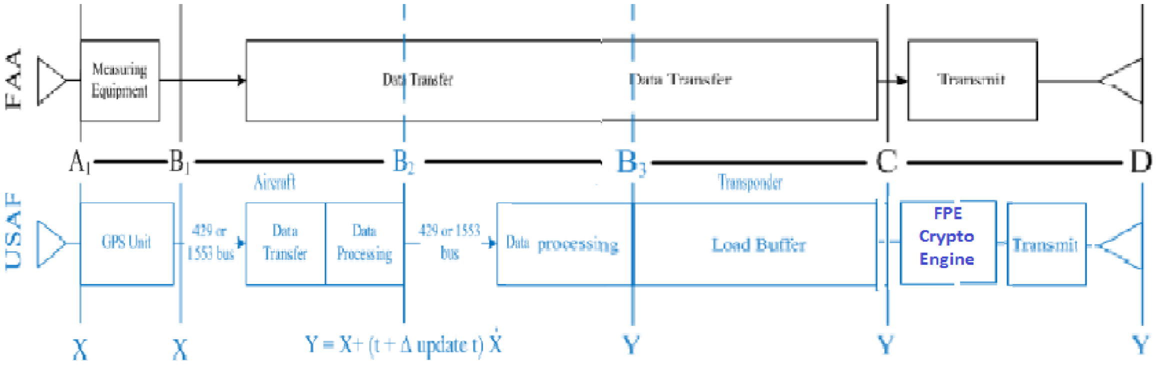


Figure 3.7: Block Diagram of Secure ADS-B avionics. Modified from [62].

challenge the algorithm's ability to obfuscate repeated data in messages. The entropies of the plaintext and resulting ciphertext are measured after encryption with the FF1, FF2, and FF3 algorithms. The ciphertext is considered to have high entropic security, if its measure of entropy equals or exceeds that of a random sequence. After verification of the security characteristics of the algorithms, they are implemented on an FPGA to test their hardware performance. Operational latency and resource utilization are measured for each algorithm. The latency and resource utilization of the underlying AES core are used as a baseline for comparison of FF1, FF2, and FF3. A BITW FPE cryptographic engine placed between the ADS-B transponder and antenna could encrypt and decrypt messages deemed sensitive for enhanced Operational Security.

IV. Results and Analysis

This chapter discusses the security and performance of the FF1, FF2 and FF3 algorithms in software and hardware. Experiments were conducted to test the hypothesis that the FPE algorithms inherit the strong security of the underlying AES block cipher and meet the avionics performance requirements of D0-260b. Results of the entropy and performance experiments are presented below. The chapter concludes with an analyis of the data.

4.1 Entropy Results

The algorithms are implemented in C using the PolarSSL AES library. The dataset is encrypted with each algorithm and the resulting ciphertexts are stored. The ENT tool [66] is used to calculate the entropy of the input plaintext samples and their corresponding ciphertext.

4.1.1 Verification of Software Implementation .

Since FF1, FF2, and FF3 are new algorithms, there exist no Known-Answer Tests or vetted implementations. The implementation is verified through decryption. While implementing the decryption process, errors were discovered in the decryption algorithms published in the Draft Special Publication 800-38G. The decryption algorithms printed in Draft SP800-38G did not properly reverse the Feistel structure of FPE. One of the three erroneous decryption algorithms is shown in Algorithm 5. The NIST was contacted regarding the errors. Morris Dworkin, author of SP800-38G, approved the suggested corrections [9], and plans to revise the three decryption specifications in the next release.

A proper decryption algorithm for FPE should reverse the Feistel structure as shown in Figure 4.1. The appropriate decryption algorithms are designed by reverse engineering

Algorithm 5 Erroneous FF1.Decrypt(K,T,X) [10].

Prerequisites:
Approved, 128-bit block cipher, $CIPH$;
Key, K, for the block cipher;
Base, $radix$, for the character alphabet;
Range of supported message lengths, $[minlen..maxlen]$;
Maximum byte length for tweaks, $maxTlen$.

Inputs:
Numeral string, X, in base $radix$ of length n such that $n \in [minlen..maxlen]$;
Tweak byte string, T, of byte length t, such that $t \in [0..maxTlen]$.

Output:
Numeral string, Y, such that $LEN(Y) = n$.

Steps:

```
Let $u = \lfloor n/2 \rfloor; v = n - u$.
Let $A = X[1..u]; B = X[u+1..n]$.
Let $b = \lceil \lceil vLOG_2(radix) \rceil /8 \rceil; d = 4 \lceil b/4 \rceil + 4$.
Let $P = [1]^1 \| [2]^1 \| [1]^1 \| [radix]^3 \| [10]^1 \| [u \, mod \, 256]^1 \| [n]^4 \| [t]^4$.
for $i \leftarrow 9$ to $0$ do
    Let $Q = T \| [0]^{(-t-b-1) mod 16} \| [i]^1 \| [NUM_{radix}(B)]^b$.
    Let $R = PRF(P \| Q)$.
    Let $S$ be the first $d$ bytes of the following string of $\lceil d/16 \rceil$ blocks:
      $R \| CIPH_k(R \oplus [1]^{16}) \| CIPH_k(R \oplus [2]^{16}) \| .. \| CIPH_k(R \oplus [\lceil d/16 \rceil - 1]^{16})$.
    Let $y = NUM_2(S)$.
    If $i$ is even, let $m = u$; Else, let $m = v$.
    Let $c = (NUM_{radix}(A) - y) \mod radix^m$.
    Let $C = STR^m_{radix}(c)$.
    Let $A = B$.
    Let $B = C$.
end for
Return $A \| B$.
```

the encryption algorithms. The corrected FF1, FF2, and FF3 decryption algorithms are presented in Algorithm 6, Algorithm 7, and Algorithm 8.

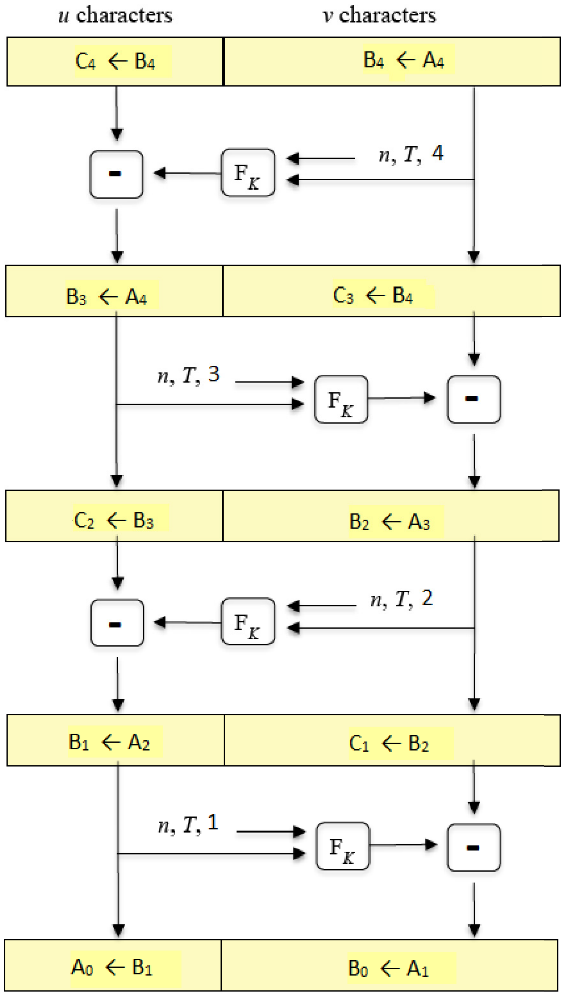


Figure 4.1: Reversed Feistel Structure of FPE for Decryption. Modified from [10]

Correctly decrypted ciphertext provided high confidence that the implementations were accurate, as it is highly unlikely that an error in either the implemented encryption or decryption algorithms would lead to a recovered plaintext. Figure 4.2 shows an example of verification through decryption.

Algorithm 6 Corrected FF1.Decrypt(K,T,X).

Prerequisites:

Approved, 128-bit block cipher, $CIPH$;
Key, K, for the block cipher;
Base, $radix$, for the character alphabet;
Range of supported message lengths, $[minlen..maxlen]$;
Maximum byte length for tweaks, $maxTlen$.

Inputs:

Numeral string, X, in base $radix$ of length n such that $n \in [minlen..maxlen]$;
Tweak byte string, T, of byte length t, such that $t \in [0..maxTlen]$.

Output:

Numeral string, Y, such that $LEN(Y) = n$.

Steps:

```
Let $u = \lfloor n/2 \rfloor; v = n - u$.
Let $A = X[1..u]; B = X[ru + 1..n]$.
Let $b = \lceil \lceil vLOG_2(radix) \rceil /8 \rceil; d = 4 \lceil b/4 \rceil + 4$.
Let $P = [1]^1 \parallel [2]^1 \parallel [1]^1 \parallel [radix]^3 \parallel [10]^1 \parallel [u \bmod 256]^1 \parallel [n]^4 \parallel [t]^4$.
for $i \leftarrow 9$ to $0$ do
    Let $C = B$.
    Let $B = A$.
    Let $Q = T \parallel [0]^{(-t-b-1) mod 16} \parallel [i]^1 \parallel [NUM_{radix}(B)]^b$.
    Let $R = PRF(P \parallel Q)$.
    Let $S$ be the first $d$ bytes of the following string of $\lceil d/16 \rceil$ blocks:
      $R \parallel CIPH_k(R \oplus [1]^{16}) \parallel CIPH_k(R \oplus [2]^{16}) \parallel .. \parallel CIPH_k(R \oplus [\lceil d/16 \rceil - 1]^{16})$.
    Let $y = NUM_2(S)$.
    If $i$ is even, let $m = u$; Else, let $m = v$.
    Let $a = (NUM_{radix}(C) - y) \bmod radix^m$.
    Let $A = STR^m_{radix}(a)$.
end for
Return $A \parallel B$.
```

Algorithm 7 Corrected FF2.Decrypt(K,T,X).

Prerequisites:
Approved, 128-bit block cipher, $CIPH$;
Key, K, for the block cipher;
Base, $radix$, for the character alphabet;
Base, $tweakradix$, for the tweak character alphabet;
Range of supported message lengths, $[minlen..maxlen]$;
Maximum supported tweak length, $maxTlen$.

Inputs:
Numeral string, X, in base $radix$ of length n such that $n \in [minlen..maxlen]$;
Tweak numerical string, T, in base $tweakradix$ of length t such that $t \in [0..maxTlen]$.

Output:
Character string, Y, such that $LEN(Y) = n$.

Steps:

```
Let $u = \lfloor n/2 \rfloor$; $v = n - u$.
Let $A = X[1..u]$; $B = X[u + 1..n]$.
**If** $t > 0, P = [radix]^1 \parallel [t]^1 \parallel [n]^1 \parallel [NUM_{tweakradix}(T)]^{13}$;
**Else** $P = [radix]^1 \parallel [0]^1 \parallel [n]^1 \parallel [0]^{13}$.
Let $J = CIPH_K(P)$.
**for** $i \leftarrow 9$ to $0$ **do**
    Let $C = B$.
    Let $B = A$.
    Let $Q \leftarrow [i]^1 \parallel [NUM_{radix}(B)^{15}$.
    Let $Y \leftarrow CIPH_J(Q)$.
    Let $y \leftarrow NUM_2(Y)$.
    **If** $i$ is even, let $m = u$; **Else**, let $m = v$.
    Let $a = (NUM_{radix}(C) - y) \bmod radix^m$.
    Let $A = STR^m_{radix}(a)$.
**end for**
Return $A \parallel B$.
```

Algorithm 8 Corrected FF3.Decrypt(K,T,X).

Prerequisites:

Approved, 128-bit block cipher, $CIPH$;
Key, K, for the block cipher;
Base, $radix$, for the character alphabet;
Range of supported message lengths, [$minlen..maxlen$], such that $minlen \geq 2$ and $maxlen \leq 2 \left\lfloor log_{radix}(2^{96}) \right\rfloor$.

Inputs:

Numeral string, X, in base $radix$ of length n such that $n \in [minlen..maxlen]$;
Tweak bit string, T, such that $LEN(T) = 64$.

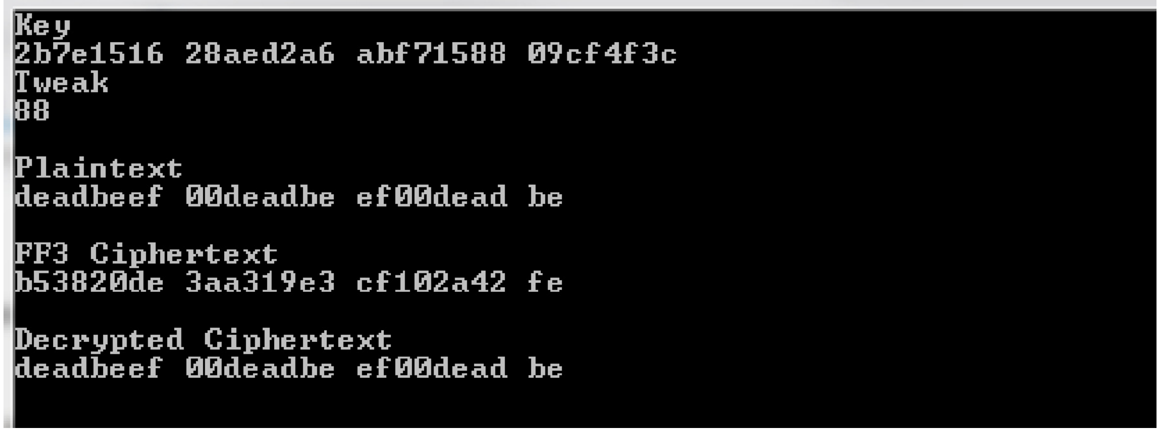
Output:

Numeral string, Y, such that $LEN(Y) = n$.

Steps:

```
Let $u = \lceil n/2 \rceil; v = n - u$.
Let $A = X[1..u]; B = X[u + 1..n]$.
Let $T_L = T[0..31]$ and $T_R = T[32..63]$;
for $i \leftarrow 7$ to 0 do
    Let $C = B$.
    Let $B = A$.
    If is even, let $m = u$ and $W = T_R$, Else let $m = v$ and $W = T_L$.
    Let $P = REV([NUM_{radix}(REV(B))]^{12}) \parallel W \oplus REV([i]^4)$.
    Let $Y = CIPH_K(P)$.
    Let $y = NUM_2(REV(Y))$.
    Let $a = (NUM_{radix}(REV(C)) - y) \bmod radix^m$.
    Let $A = REV(STR^m_{radix}(a))$.
end for
Return $A \parallel B$.
```

*Where $REV(X)$ reverses the order of characters in the Character String X



```
Key
2b7e1516 28aed2a6 abf71588 09cf4f3c
Tweak
88

Plaintext
deadbeef 00deadbe ef00dead be

FF3 Ciphertext
b53820de 3aa319e3 cf102a42 fe

Decrypted Ciphertext
deadbeef 00deadbe ef00dead be
```

Figure 4.2: Software Verification of FF3 Implementation.

4.1.2 Fixed Bytes.

The number and distribution of bytes across the message consistently determines the level of entropy of the plaintext. As expected, the entropy of the unencrypted plaintext samples decreases as the number of deterministic bytes increases. Pilot experiments validate the ENT tool by comparison of its measurements to a theoretical calculation of entropy. According to Equation (3.2), a message composed of identical bytes has zero entropy. The ENT tool successfully measures a plaintext file composed of the same byte repeated in 4,000 messages to have 0 bits/byte of entropy. A 13-byte message with non-repeating byte values has a theoretical entropy of 3.7 bits. Measurement with the ENT tool of a plaintext file composed of 4,000 samples of the same 13 fixed bytes sequence yields an entropy of 3.547 bits/byte. Despite the lack of change from message to message, this entropy measurement reflects the internal byte variation in the message. Note that the definition of entropy contains a logarithmic term.

Unexpectedly, the entropy of plaintexts with consecutive deterministic bytes at the front of the message are not statistically different from their randomly distributed counterparts, within a 95% Confidence Interval (CI). Although the deterministic byte sequence is different for each trial, there is low variance in the resulting entropy as shown by the standard deviation data in Table 4.1. The standard deviation increases as the mean entropy decreases which implies that security differences between the algorithms are more defined with low entropy plaintext.

Table 4.1: Fixed Bytes Entropy (bits/byte).

Scenario	**Plaintext**		**FF1**		**FF2**		**FF3**	
	mean	**std_dev**	**mean**	**std_dev**	**mean**	**std_dev**	**mean**	**std_dev**
All Random	*7.99633*	0.00032	*7.99646*	0.00027	*7.99635*	0.00030	*7.99642*	0.00029
3 Front	*7.24037*	0.00202	*7.99633*	0.00039	*7.99649*	0.00022	*7.99639*	0.00041
3 Random	*7.24022*	0.00208	*7.99654*	0.00028	*7.99650*	0.00030	*7.99650*	0.00030
6 Front	*6.40284*	0.03235	*7.99648*	0.00040	*7.99647*	0.00025	*7.99656*	0.00033
6 Random	*6.40242*	0.03273	*7.99636*	0.00034	*7.99637*	0.00027	*7.99650*	0.00024
9 Front	*5.44880*	0.04575	*7.99637*	0.00036	*7.99634*	0.00028	*7.99651*	0.00026
9 Random	*5.44870*	0.04593	*7.99638*	0.00035	*7.99655*	0.00033	*7.99643*	0.00034
12 Front	*4.25632*	0.05607	*7.94225*	0.00467	*7.93989*	0.00633	*7.94223*	0.00568
12 Random	*4.25617*	0.05592	*7.99652*	0.00034	*7.99656*	0.00025	*7.99640*	0.00029

The entropy of a control message, composed of All Random bytes, is measured to be 7.99633 (bits/byte). The random sequence was extracted from the Random.org TRNG file of 2013-09-17. The TRNG sequence, comparable to a pseudo-random sequence [51], serves as the baseline for evaluation of the security merits of the FF1, FF2 and FF3 algorithms. A ciphertext with entropy equal to or greater than that of the random sequence is considered secure. Figure 4.3 shows the mean entropy of the plaintext and ciphertexts of each Fixed Bytes scenario, averaged for the 20 trials. The ciphertext equals or exceeds the random sequence entropy threshold in all but one scenario.

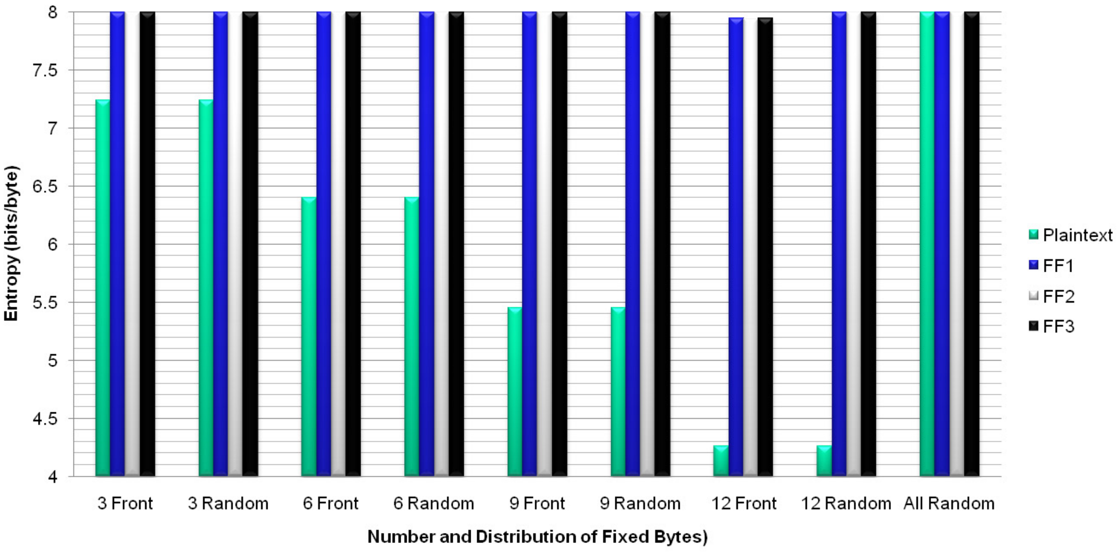


Figure 4.3: Mean Entropy of Fixed Bytes Scenarios.

Note that, the encryption of 12 fixed consecutive bytes in the ‘12 Front’ scenario causes a mean ciphertext entropy consistently below the random sequence threshold of 7.99633 (bits/byte). The mean entropy of the ‘12 Front’ ciphertext is more than two standard deviations smaller than the baseline entropy across all three algorithms. This scenario fails to yield a secure ciphertext. The ‘12 Front’ plaintext has a mean entropy of 4.2563 which is not statistically different from that of the ‘12 Random’ plaintext which has a mean entropy of 4.2562 (bits/byte), with 95% confidence. However, encryption of the 12 randomly distributed bytes in ‘12 Random’ scenario yields a secure ciphertext.

Additionally, the ‘12 Front’ scenario displays the largest differences in the entropy of the FF1, FF2 and FF3 ciphertext. Although, all three algorithms fail to yield secure ciphertext in the ‘12 Front’ scenario, it is important to investigate which algorithm yields the better ciphertext.

4.1.2.1 Comparison.

Pairwise Student’s t-tests are conducted to determine whether there exist statistical differences between the three algorithms. A robust t-test requires the following assumptions: random sampling of population, population normality, independent samples, and similar standard deviations. The use of the entirety of results from the 20 trials satisfies the random sampling requirement. The populations are determined to be approximately normal and of similar distribution through visual analysis of their descriptive statistics graphed in a boxplot. Each trial uses independent deterministic and random byte sequences. Finally, the standard deviations of the ciphertext are similar in all scenarios as shown in Table 4.1. Figure 4.4 shows a boxplot of the ciphertext samples for each algorithm in the worst case scenario. The ‘12 Front’ scenario displays the lowest values and the largest variances in ciphertext entropy; however, the boxplot shows that the populations have similar spreads and skewness, and few outliers.

The R Statistical Computing tool is used to calculate pairwise Welch Two Sample t-tests for each scenario. The p-values shown in Table 4.2 show that within 95% confidence, all t-tests fail to reject the null hypothesis that the algorithms do not have statistically significant differences in their security performance in the Fixed Bytes scenarios. As a result, the three algorithms are statistically the same with regards to security.

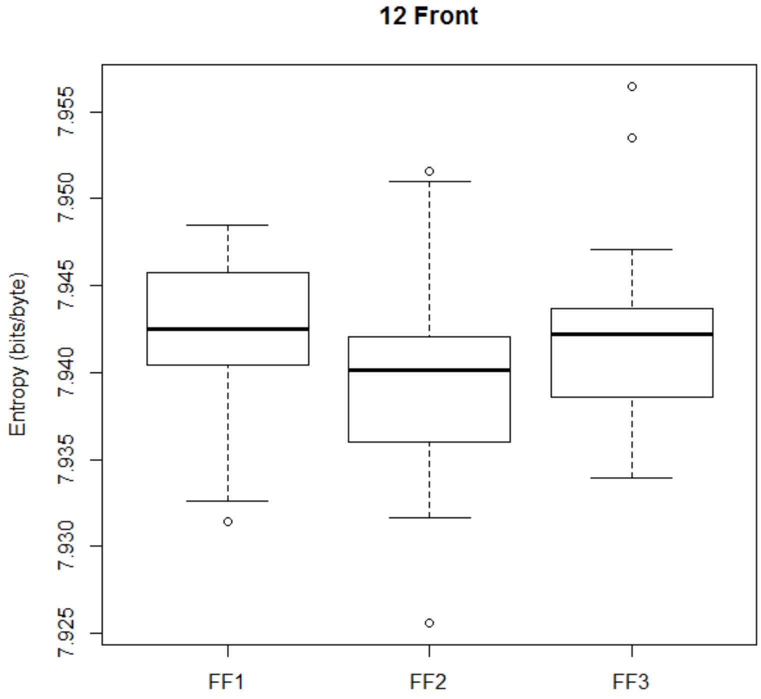


Figure 4.4: Boxplot of ‘12 Front’ Ciphertext Populations.

4.1.3 Fixed Fields.

The random input files were then altered to model plausible ADS-B messages by limiting the data fields to operationally logical values. An increasing number of fields is held constant in the ADS-B message. The results of the Fixed Fields tests are depicted in Table 4.3 and Figure 4.5. As expected, the entropy of the plaintext message decreases as the number of fields with fixed content increases.

In the scenarios with zero, one, or two fixed fields, the ciphertext entropies are above 7.99633 (bits/byte), the threshold of a random sequence. The entropy of the ciphertext falls below the threshold when three data fields are held constant. The input entropy of the scenario in which the Position, Altitude and Address ME fields are held constant is

Table 4.2: Pairwise t-Tests for Fixed Bytes (p-value).

Scenario	**FF1-FF2**	**FF1-FF3**	**FF2-FF3**
3 Front	0.1248	0.6528	0.3504
3 Random	0.6872	0.6505	0.9564
6 Front	0.9267	0.4835	0.3294
6 Random	0.8759	0.1346	0.1251
9 Front	0.794	0.1588	0.0513
9 Random	0.1033	0.6022	0.2593
12 Front	0.1889	0.989	0.2272
12 Random	0.6738	0.2616	0.078

5.55 (bits/byte). Although these data fields are only 6 bytes long, their nearly consecutive emplacement in the structure of the ADS-B message, and the restricted range of their values, causes a failure in the entropic security of the FPE algorithms. The entropy values decrease further when four fields are held constant. In the cases of three and four fixed fields, the message space is reduced to 2^{10} and 2^5 permutations since only 10 or 5 bits of the ME subfield are randomized, respectively.

4.1.3.1 Comparison.

Given the differences between the entropies of the ciphertexts, pairwise two-tailed t-tests are used to evaluate the differences in security of the three algorithms. The t-test assumptions of population normality, independence of samples, and similar variances are satisfied. The R Statistical Computing tool is used to perform Welch Two Sample t-tests. Results are shown in Table 4.4. The p-values suggest that there is no statistical difference between the three algorithms in the Fixed Fields scenarios, within 95% CI.

Table 4.3: Fixed Fields Entropy (bits/byte).

Fixed Fields	**Plaintext**		**FF1**		**FF2**		**FF3**	
	mean	**std_dev**	**mean**	**std_dev**	**mean**	**std_dev**	**mean**	**std_dev**
None	7.94833	0.00167	7.99653	0.00035	7.99634	0.00037	7.99648	0.00036
Position	6.89529	0.03290	7.99651	0.00030	7.99649	0.00028	7.99650	0.00029
Pos & Altitude	6.50837	0.03864	7.99642	0.00023	7.99643	0.00028	7.99659	0.00035
Pos & Alt & Address	5.54999	0.04875	7.96272	0.00339	7.96333	0.00318	7.96412	0.00356
Pos & Alt & Addr &Type Code	5.05440	0.06011	7.47995	0.03859	7.48042	0.03084	7.48164	0.04221

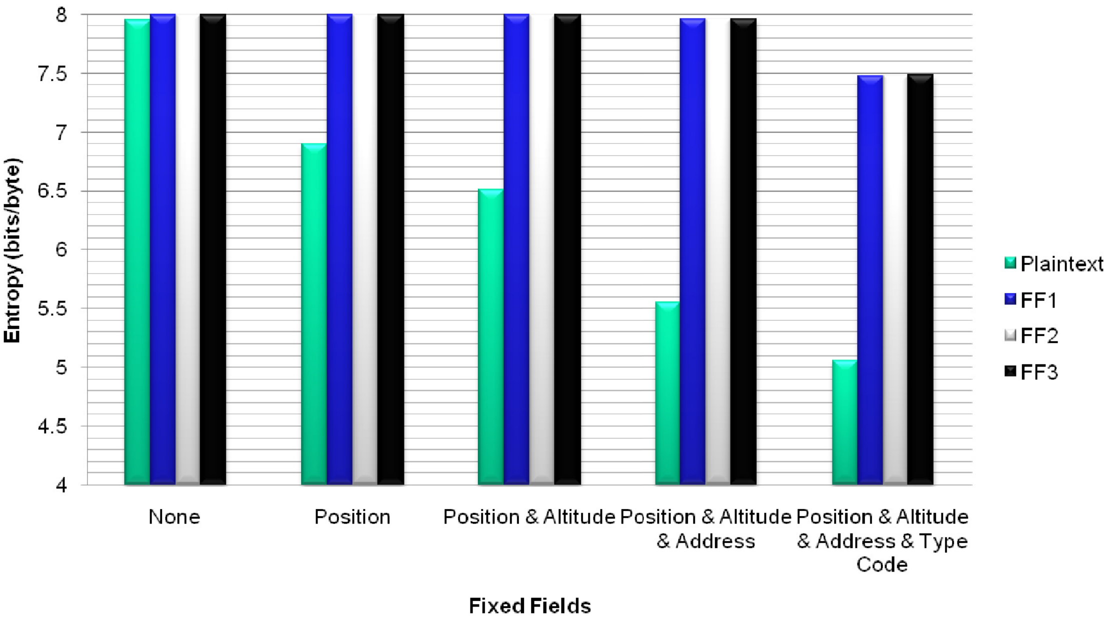


Figure 4.5: Mean Entropy of Fixed Fields Scenarios.

Table 4.4: Pairwise t-Tests for Fixed Fields (p-value).

Fixed Fields	**FF1-FF2**	**FF1-FF3**	**FF2-FF3**
None	0.0985	0.6261	0.2374
Position	0.7978	0.906	0.8921
Pos & Altitude	0.9063	0.0763	0.1154
Pos & Alt & Address	0.5665	0.213	0.464
Pos & Alt & Addr & Type Code	0.9657	0.8956	0.9181

4.1.4 Radar Track.

ADS-B messages were generated for a Radar observed aircraft traveling from California to Nebraska. The aircraft in question takes off from the Californian coast, climbs to an altitude of 35,000 ft and maintains approximately the same course heading all the way to Nebraska. This flight represents one of the worst case scenarios for encryption in the ADS-B environment, in which several data fields are nearly constant from one message to the next. Table 4.5 and Figure 4.6 show the results of the entropy measurements. This track is found to have an aggregate entropy of 6.51 bits/byte which is closest to the entropy of a simulated ADS-B message with two fixed fields.

Table 4.5: Entropy of Radar Track.

	Plaintext	**FF1**	**FF2**	**FF3**
WADS Radar Track	6.513979	7.9983	7.9986	7.9984

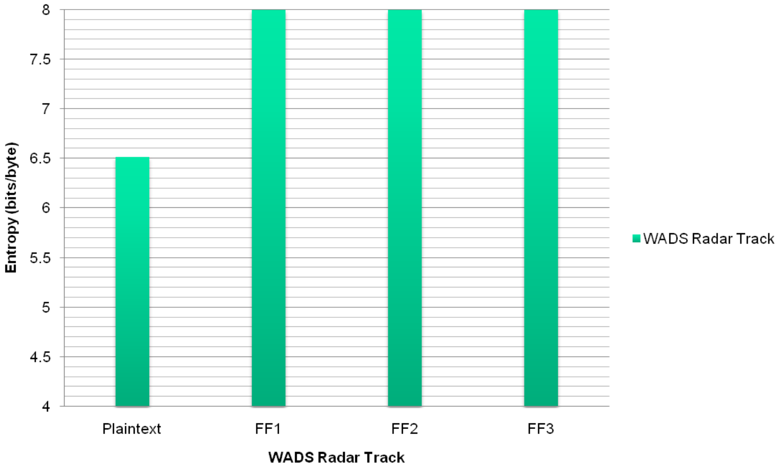


Figure 4.6: Entropy of Radar Track.

4.1.4.1 Comparison.

Encryption of the radar track yields ciphertexts with entropy well above the random sequence threshold of 7.99633 (bits/byte) for all three algorithms. In the Radar Track experiment, the FF2 algorithm produces the ciphertext with the highest entropy. For this particular set of ADS-B messages, the FF3 algorithm has the second highest entropic security among the three algorithms. Note, however, that there are not enough data points to make generalizable inferences.

4.1.5 Assessment.

The FF1, FF2, and FF3 algorithms securely encrypt the majority of plaintext treated in the three sets of experiments. In the Fixed Bytes experiments, the algorithms only fail to securely encrypt plaintext with 12 consecutive deterministic bytes at the front of the message. However, the algorithms successfully encrypt plaintext with lower entropy but with a random distribution of deterministic data. In the Fixed Fields experiments, the algorithms begin to fail when three consecutive data fields are held constant. The Radar Track experiment reproduces a real flight scenario. The algorithms successfully encrypt the ADS-B Out traffic extracted from the WADS Radar Track. The entropy of the encrypted Radar Track messages are higher than all other scenarios.

The entropy analysis demonstrates no statistically significant differences in the security of the FF1, FF2 and FF3 algorithms. This conclusion is supported by tests performed on a total of 1,128,866 unique ADS-B messages from a modeled dataset generated for the Fixed Bytes experiments, to a simulated dataset generated for the Fixed Fields experiments, and an operational dataset measured from a real transiting aircraft.

4.2 Performance Results

The FF1, FF2, and FF3 algorithms are implemented in VHDL and synthesized on a Xilinx Virtex-6. The prevalence of the VHDL hardware description language in US Government research motivated its use in this study. Note that other hardware description languges may be used to implement the FPE algorithms. The performance results discussed in the following sections indicate that the underlying AES core is the principal factor in the latency and resource utilization of the algorithms.

4.2.1 Verification of Hardware Implementation .

The underlying AES block is verified in a behavioral simulation with sample key and plaintext from the NIST's Known-Answer Test [43]. Figure 4.7 shows a screenshot of the behavioral verification of the AES core used to implement the FF1, FF2, and FF3 algorithms. The core produces a ciphertext that matches the test vector.

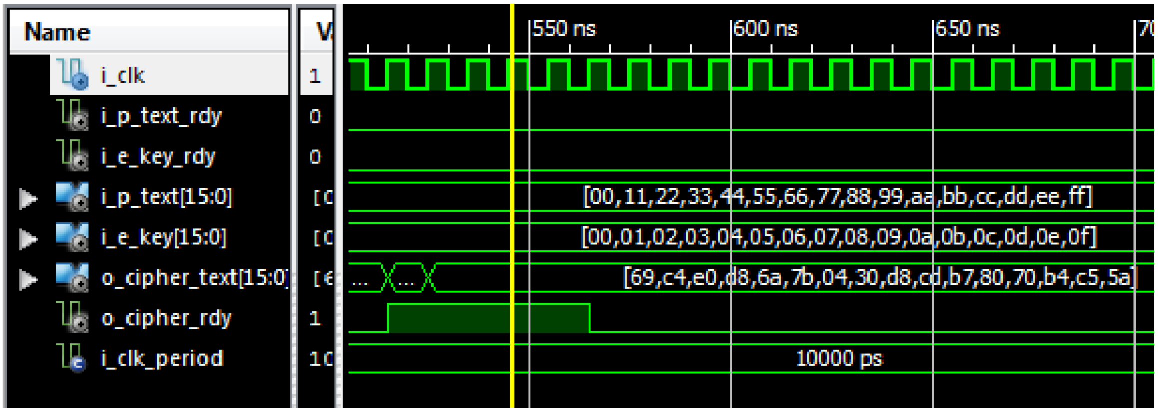

Name
V
550 ns
600 ns
650 ns
i_clk
1
i_p_text_rdy
0
i_e_key_rdy
0
i_p_text[15:0]
[00,11,22,33,44,55,66,77,88,99,aa,bb,cc,dd,ee,ff]
i_e_key[15:0]
[00,01,02,03,04,05,06,07,08,09,0a,0b,0c,0d,0e,0f]
o_cipher_text[15:0]
[69,c4,e0,d8,6a,7b,04,30,d8,cd,b7,80,70,b4,c5,5a]
o_cipher_rdy
1
i_clk_period
10000 ps

Figure 4.7: Behavioral Verification of AES core.

The FF1, FF2, and FF3 implementations are verified in a similar fashion by comparison to known plaintext and ciphertext pairs produced by the software implementation. Figure 4.8 shows as example, the verficiation of the FF3 hardware implementation. Correctness is assessed by comparison to the software verification (see Figure 4.2). The three algorithms are correctly implemented.

4.2.2 Resource Utilization.

The Iterative Looping architecture employed in the design minimizes the hardware resources needed for each algorithm. Only one instance of a round is implemented for each algorithm. A loop counter is used to iterate through the appropriate number of rounds for each algorithm. All other subfunctions are realized with dedicated

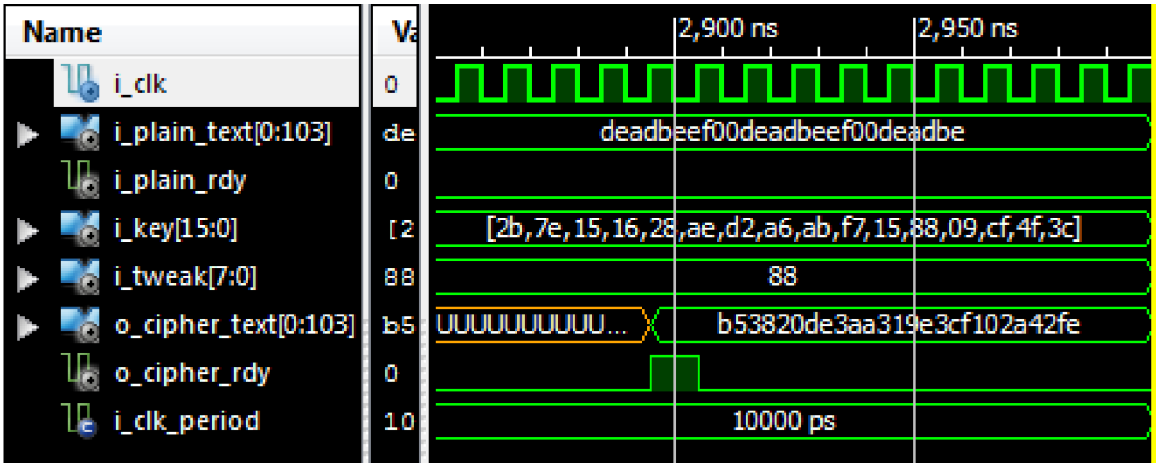


Figure 4.8: Behavioral Verification of FF3 Implementation.

components. Inside the round function, each call to AES is implemented on a dedicated core in order to avoid complexity in the data flow control mechanism.

The AES core employed in these designs occupies 1,864 Slices on the Virtex-6 (XC6VLX240T) FPGA device. Such an implementation is comparable in size to recently published implementations [7]. Table 4.6 shows the results of the device utilization analysis. The size of the AES core is the principal factor in determining the area of the FPE implementations. As expected, the number of resources required increases proportionally to the number of AES components in each design. The exact number of slices, registers, LUTs, and RAM blocks is determined by the default Xilinx ISE 14.6 suite's XST synthesis optimization process. No 36Kb blocks of RAM were used during synthesis.

Unexpectedly, the FF3 implementation consumes less FPGA resources than the AES core. The AES core as benchmarked, includes a packet control mechanism that registers input and output (I/O) signals connected to the core. The AES packet controller is not needed for integration into FF1, FF2, and FF3. Its function is performed by the shift register that is used to synchronize the cascaded AES blocks inside the FPE Round block.

Table 4.6: Resource Utilization of AES and FPE Algorithms.

Algorithm	**AES**	**FF1**	**FF2**	**FF3**
Number of occupied Slices	1,864	3,850	3,728	1,820
Number of Slice Registers	5,801	11,285	11,323	5,592
Number of Slice LUTs	3,452	7,426	6,825	3,587
Number of 18K block RAM	172	343	342	170

The synchronization logic used inside the FF1, FF2, and FF3 designs is more hardware efficient than the AES packet control mechanism.

4.2.2.1 Comparison.

The FF1 implementation requires the most FPGA resources. The FF1 algorithm uses two instances of AES per round which causes the area or number of slices required, to be approximately twice that of one AES core. FF2 uses only once instance of AES in its round design, but requires an additional AES block to generate its subkey. As such, FF1 and FF2 consume approximately twice as many device resources as the AES core. FF3 has the smallest footprint of the three algorithms as it requires only one AES core in its FPE Round block.

4.2.3 Operational Latency.

The post-place and route (post-PAR) static timing report in Xilinx ISE 14.6 provides a comprehensive summary of timing delay information. Table 4.7 shows the results of timing analysis and operational latency measurements for each algorithm. The maximum frequency tolerable for each design is derived from the worst path delay found during routing. According to the Place and Route report, the round control mechanism is the source of the maximum delay in each design. The number of clock cycles per round of FPE and the number of clock cycles required for a complete encryption cycle are obtained through behavioral simulation. The minimum latency for a complete encryption cycle is calculated for each algorithm by dividing the number of clock cycles required per encryption by the maximum frequency.

Table 4.7: Latency of AES and FPE Algorithms.

Algorithm	**AES**	**FF1**	**FF2**	**FF3**
Maximum Frequency (MHz)	336.315	279.587	284.592	283.427
Clock Cycles per Round	3	68	33	32
Clock Cycles per Encryption	31	707	374	269
Minimum Latency (ms)	0.092175	2.528729	1.314162	0.949098

4.2.3.1 Comparison.

The FF1 algorithm makes two calls to AES every round which causes it to have the highest latency of the three algorithms. FF1 takes 68 clock cycles per round, and 707 clock cycles in total to initialize the encryption parameters and complete ten rounds of encryption. FF2 has a lower latency than FF1 because of a single call to AES in the F-block of the Feistel structure versus two in FF1. As such, FF2 takes approximately half as many clock cycles per round and per encryption, as FF1. FF3 has the lowest latency of the three algorithms because it uses only eight rounds compared to ten for FF1 and FF2. The computed minimum latencies are proportional to the operational latencies because the three algorithms have similar maximum frequencies. The latency of an operational system will depend on the system clock frequency and CMOS technology.

4.2.4 Assessment.

The resource utilization of the underlying AES core is the biggest factor in the resource utilization of the FPE algorithms. FF3 consumes the least number of FPGA slices, and has the lowest operational latency of the three algorithms. However, the computed latencies of the FF1, FF2, and FF3 hardware implementations exceed the DO-260B [55] Standard's maximum of 100ms for ADS-B equipment.

4.3 Summary

The FF1, FF2, and FF3 algorithms securely encrypt the majority of plaintext treated in the three sets of entropy experiments. The entropy after encryption of the ADS-B messages extracted from the WADS Radar Track are higher than that of the artificial messages tested in the Fixed Bytes and Fixed Fields experimental scenarios. Statistical analysis reveals no significant differences in the security of FF1, FF2 and FF3. When implemented in hardware, the use of the underlying block cipher by each algorithm is the most significant factor in the performance of the FPGA implementations. The FF3 algorithm has the lowest latency of the three because it uses only eight rounds of encryption, and makes the fewest calls to AES per round. FF2 has slightly higher latency than FF3, and FF1 requires the most clock cycles per encryption. However, all three algorithms benefit from operational latencies that are lower than the DO-260B requirement for ADS-B equipment.

V. Conclusions and Future Work

This chapter summarizes the results of the research effort and provides suggestions for future work. The goal of this research was to determine the suitability of the FF1, FF2, and FF3 algorithms for encryption of ADS-B messages, and the feasibility of a BITW FPE cryptographic engine.

5.1 Research Summary

The NAS is due for a major upgrade to the NextGen Air Transportation System, which includes an evolution from Radar-based surveillange to satellite-based surveillance. NextGen furthers the evolution of the ATC system towards Free Flight, and brings several needed improvements to the GA and commercial aviation sectors. The military has identified multiple operational benefits of ADS-B, but is limited by unresolved security gaps.

The availability of stand-alone ADS-B receivers for aerial enthusiasts, researchers, and anonymous users poses an OPSEC risk to DoD, Department of Homeland Security (DHS), and law enforcement aircraft. A malicious user with an inexpensive ADS-B In receiver can possibly track the precise latitude, longitude and altitude of Air Force One or other aircraft carrying political dignitaries. Furthermore, researchers have demonstrated the ease with which ADS-B messages can be spoofed and false traffic injected into the ADS-B domain. As such, the DoD has asked for the development of encryption and jam/spoof proofing mechanisms for ADS-B to improve COMSEC and mitigate the OPSEC risks.

The U.S. Navy and Coast Guard use the AES and Blowfish algorithms to encrypt the AIS, their homologous vessel tracking system. However, the non-standard format of

ADS-B messages and the legacy communication channels used by its transponders make it incompatible with traditional block ciphers.

One approach for securing ADS-B communication, is to adapt the messages for use within the existing military IFF system. However, the current IFF systems lack the precision-tracking framework needed to maintain the accuracy of ADS-B. A more desirable solution would use FPE to direcly encrypt the ADS-B message.

The proposed solution for securing ADS-B is to retrofit encryption to legacy transponders by adding a BITW FPE cryptographic module to secure ADS-B communications. The goal of this research was to determine the suitability of the FF1, FF2, and FF3 FPE algorithms recommended by the NIST, for encryption of ADS-B messages with regards to security and performance.

The first objective of the research effort was to evaluate the security characteristics of each algorithm using a representative dataset. The algorithms were tested with a model dataset composed of incremental numbers of deterministic bytes in the Fixed Bytes test, a simulated ADS-B message dataset in the Fixed Fields test, and an operational dataset extracted from an observed Radar track. Entropy results in all three sets of experiments, demonstrate that there are no statistical differences in the security of the FF1, FF2 and FF3 algorithms.

The second objective of the research was to evaluate the hardware performance of the three algorithms by measuring operational latency and resource utilization of an FPGA implementation. The FF3 algorithm proved to have the lowest area and latency, due to its small number of encryption rounds and spare use of AES in the Feistel round. The characteristics of the underlying block cipher used in the implementation of the FPE algorithms are the principal factors in determining the resource utilization and latency of the hardware implementation.

The results of this research suggest that FPE is a suitable encryption scheme for encrypting ADS-B communications. The algorithms are able to obfuscate repeated data in plaintext, and output ciphertext with high entropic security. The reliance of the algorithms on AES make them easily implementable on a wide range of platforms, including avionics hardware. The computed latencies of the FF1, FF2, and FF3 FPGA designs exceed the requirements of DO-260B "Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance - Broadcast (ADS-B) and Traffic Information Services - Broadcast (TIS-B)."

5.2 Impact

The use of FPE to encrypt ADS-B messages provides Confidentiality to the system. It prevents the disclosure of aircraft information to unauthorized parties during sensitive military or law enforcement operations. The U.S. military can explore the solution as a viable option for complying with the 2020 congressional mandate for ADS-B equipage, while maintaining OPSEC.

The Air Force can take advantage of the benefits to the transition to NextGen and ADS-B it identified in 2001, without sacrificing security. Secure ADS-B could enhance safety and mission capabilities in Air Refueling (AR), Formation Flying, Rendezvous, Fighter Intercept, Air Combat Maneuvering Instrumentation (ACMI) missions, and precision Airdrop.

Military aircraft manufacturers, such as General Atomics-Aeronautic Systems Inc and BAE Systems, testing ADS-B technology for use within Airborne Sense-And-Avoid architecture (ABSAA), can leverage findings from this research to assure the security of these safety critical systems. A malicious user could potentially derail the trajectory of autonomous swarms of UAVs or disrupt their formation flight by projecting false traffic with spoofed ADS-B messages. By using FPE, precision formation flight can rely on encrypted ADS-B messages private to the formation.

During the course of this research, errors were discovered in the FPE decryption algorithms published in Draft SP800-38G [10]. The decryption algorithms did not properly reverse the Feistel structure of FPE. The error report was submitted to NIST along with corrected decryption algorithms. Morris Dworkin, author of SP800-38G, approved the suggested corrections [9].

5.3 Recommendations for Future Work

The initial findings of this research indicate that FF1, FF2, and FF3 may be used to encrypt ADS-B with high security and low resource cost. Although the three algorithms have the same entropic security, FF3 requires the least amount of hardware resources and demonstrates the lowest operational latency. The research proposes the use of a BITW FPE cryptographic engine to retrofit encryption to legacy ADS-B transponders. Further investigation is necessary before the development and deployment of such a system can be realized.

5.3.1 Characterization of ADS-B Entropy.

This research effort experimented with a Radar track obtained from WADS of an aircraft travelling from California to Nebraska. The entropy of the unencrypted messages was measured to be 6.51 (bits/byte). The steady trajectory and altitude represented one of the expected worst plaintext cases for encryption. However, the subsequent encryption of the messages with the FF1, FF2 and FF3 algorithms, yielded ciphertext with the highest entropies measured in the research. One area of interest is the characterization of the entropy of ADS-B messages for various flight trajectories and aircraft status. The study would systematically evaluate the effect of various combinations of constant ME subfields on the entropy of the FF1, FF2 and FF3 ciphertexts.

5.3.2 Key Management.

FPEs are symmetric encryption algorithm, which means that the key must be distributed offline or through another secure protocol. This research did not consider the challenge of key distribution in its evaluation. Further study is necessary to devise a suitable key distribution scheme. One may look at W-AIS for inspiration or extend the existing key distribution scheme used for military IFF transponders.

5.3.3 NSA Approval of FPE.

The NSA categorizes encryption items into four product types [29]. IFF transponders use a Type-1 algorithm approved by the NSA. A Type-1 Product refers to an NSA endorsed classified or controlled cryptographic item for classified or sensitive U.S. government information when appropriately keyed. AES with a 256-bit key is rated as a Type-1 Product. A Type-2 Product refers to an NSA endorsed unclassified cryptographic equipment for sensitive but unclassified U.S government information. A Type-3 Product refers to NIST endorsed algorithms, registered and FIPS published, for sensitive but unclassified U.S. goverment and commercial information. A Type-4 Product refers to algorithms that are registered by the NIST but are not FIPS published. FF1, FF2, and FF3 currently qualify as Type-4 Products. The reclassification of FF1, FF2 and FF3 as Type-1 or the development of a Type-1 FPE algorithm would facilitate adoption by the DoD and DHS community.

5.3.4 Channel Interference.

While in encrypted mode, a W-AIS transponder can still receive all unencrypted transmission from commercial AIS equipped ships within range [46]. This allows military vessels to communicate with their trusted networks, while maintaining situational awareness of other ships in the vicinity. The impact of injecting encrypted messages into the ADS-B domain must be quantitatively evaluated.

In the W-AIS system, encrypted content is transmitted in a time slot designated for its specific message format. ADS-B does not currently use TDMA or any other channel multiplexing technique. The ICAO is conducting research [1] on phase modulation of the 1090 MHz ES channel to increase data capacity without adding interference. This multiplexing technique may enable military and law enforcement aircraft to transmit encrypted ADS-B messages on a reserved portion of the channel.

5.3.5 Prototype Transponder with Cryptographic Engine.

This research proposed a BITW FPE cryptographic engine. A detailed systems engineering study is necessary to evaluate the integration of such a cryptographic engine into existing ADS-B transponders. The algorithms are demonstrated to have lower latency than the maximum indicated by DO-260B [55] for ADS-B equipment. However, it is not possible to evaluate the impact of the latency of the cryptographic engine on that of the overall transponder without detailed specifications on commercial ADS-B transponders. These component level system specifications are regarded as proprietary information by avionics manufacturers.

A prototype Secure ADS-B transponder built with a DIY kit such as Günter Köllner's Mode S Beast [34], can help estimate the overall latency of an ADS-B transponder with an add-on FPE cryptographic engine. Such a prototype, will best ressemble a production transponder by adhering to the DO-254 [53] Standard "Design Assurance Guidance For Airborne Electronic Hardware."

5.3.6 Standardization of Secure ADS-B.

The first edition of the NATO STANAG for W-AIS was released in 2007, three years after the IMO mandate to fit AIS on all international voyaging ships. The W-AIS is based on existing AIS transponder specificiations defined in ITU-R M.1371 [30] with add-on encryption units, in order to reduce acquisition costs. A standard for Secure ADS-B based on DO-260B [55] with add-on FPE encryption units would significantly expedite the acquisition process and reduce costs to the taxpayer.

5.4 Conclusions

The FF1, FF2 and FF3 FPE algorithms adequately secure ADS-B communications. Although the three algorithms have statistically identical security, the FF3 algorithm stands out as the most efficient in hardware. A BITW cryptographic module employing FF3 or any of the other FPE algorithms may be used to retrofit encryption to legacy ADS-B equipment.

Bibliography

[1] Aeronautical Surveillance Panel (ASP). *Update to the 1090ES Phase Overlay Project Progress*. Technical Report ASP TSG WP14-11, International Civil Aviation Organization (ICAO), 2013.

[2] Bellare, Mihir, Phillip Rogaway, and Terence Spies. "The FFX Mode of Operation for Format-Preserving Encryption", February 2010. Report to NIST describing FFX algorithm.

[3] Black, John and Phillip Rogaway. "Ciphers with arbitrary finite domains". *Topics in CryptologyCT-RSA 2002*, 114–130. Springer, 2002.

[4] Brier, Eric, Thomas Peyrin, and Jacques Stern. "BPS: a format-preserving encryption proposal". *Submission to NIST, available from their website*, 2010.

[5] DoD Policy Board on Federal Aviation. "ADS-B Implementation". URL events.aviationweek.com/html/adsb09/Allan%20Storm.pdf, 2010. Aviation Week: ADS-B Management Forum.

[6] Dodis, Yevgeniy and Adam Smith. "Entropic security and the encryption of high entropy messages". *Theory of Cryptography*, 556–577. Springer, 2005.

[7] Dogan, Ahmet, S Berna Ors, and Gokay Saldamli. "Analyzing and comparing the AES architectures for their power consumption". *Journal of Intelligent Manufacturing*, 1–9, 2012.

[8] Drouilhet Jr, Paul R, George H Knittel, and Vincent A Orlando. "Automatic dependent surveillance air navigation system", October 29 1996. US Patent 5,570,095.

[9] Dworkin, Morris. "FPE decryption algorithms". Private Communication, 21 Nov 2013.

[10] Dworkin, Morris. "Draft Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption". *NIST Special Publication*, 800:38G, 2013.

[11] Elbirt, Adam J, Wei Yip, Brendon Chetwynd, and Christof Paar. "An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists". *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 9(4):545–557, 2001.

[12] Electronic Systems Center, USAF 853 ELSG/NT. "Automated Dependent Surveilance - Broadcast Military (ADS-M)". URL http://www.afceaboston.com/documents/events/cnsatm2007/presentations/2007%20CNSATM%20Presentations/

04-24-2007%20Tuesday/Session%20Rm%202/4%20-%20Military%20ADS-B%20Briefing%20for%20CNS-ATM%20Conference%20-%20McMath%2020070418.pdf, 2007.

[13] Elminaam, Diaa Salama Abdul, Hatem Mohamed Abdul Kader, and Mohie Mohamed Hadhoud. “Perfomance evaluation of symmetric encryption algorithms”. *International Journal of Computer Science and Network Security*, 8(12):280–285, December 2008.

[14] FAA Aerospace Forecast. “Fiscal Years 2013–2033”. URL http://www.faa.gov/about/office_org/headquarters_offices/apl/aviation_forecasts/aerospace_forecasts/2013-2033/media/Faa_Aerospace_Forecasts_Fy20132033.pdf, 2013.

[15] Federal Aviation Administration. “Chapter 1: IFR Operations in the National Airspace System”. *FAA-H-8261-1A Instrument Procedures Handbook*, 1–32. URL http://www.faa.gov/regulations_policies/handbooks_manuals/aviation/instrument_procedures_handbook/media/CH-01.pdf, 2007.

[16] Federal Aviation Administration. “National Airspace System Overview”. URL http://www.faa.gov/air_traffic/nas_redesign/regional_guidance/eastern_reg/nynjphl_redesign/documentation/feis/appendix/media/Appendix_A-National_Airspace_System_Overview.pdf, 2007.

[17] Federal Aviation Administration. “Automatic Dependent Surveillance-Broadcast (ADS-B) Out Performance Requirements to Support Air Traffic Control (ATC) Service; Final Rule”. *Federal Register*, 75(103):30106 – 30195, May 2010.

[18] Federal Aviation Administration. “NextGen Implementation Plan”. *Washington, DC, URL http://www.faa.gov/nextgen/implementation/media/NextGen_Implementation_Plan_2013.pdf*, 2013.

[19] Finke, Cindy, Jonathan Butts, and Robert Mills. “ADS-B encryption: confidentiality in the friendly skies”. *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, 9. ACM, 2013.

[20] Finke, Cindy D. *Format Preserving Encryption: Evaluating FFX for Use Within the NextGen Air Traffic Control System*. Master’s thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, March 2013.

[21] Foster, Nick. “GNU Radio Air Mode S”. URL https://www.cgran.org/wiki/gr-air-modes, September 2013.

[22] General Atomics Aeronutical Systems, Inc. “GA-ASI Successfully Tests ADS-B Surveillance System Aboard Guardian”. Press Release. URL http://www.ga-asi.com/news_events/index.php?read=1&id=402, October 25 2012.

[23] Good, Tim and Mohammed Benaissa. "AES on FPGA from the fastest to the smallest". *Cryptographic Hardware and Embedded Systems–CHES 2005*, 427–440. Springer, 2005.

[24] Harrison, Michael J. "ADS-X the next Gen approach for the next generation air transportation system". *25th Digital Avionics Systems Conference, 2006 IEEE/AIAA*, 1–8. IEEE, 2006.

[25] Headquarters U.S. Air Force, USAF XOR-GANS. "Military Unique Applications for ADS-B". URL http://adsb.tc.faa.gov/WG6_Meetings/Meeting4/242A-WP-4-10%20Military%20Apps.pdf, 2001.

[26] Helfrick, Albert D. *Principles of avionics, Fourth Edition*. Avionics Communications, 2007.

[27] Huang, Wing-Shih, Ram M Narayanan, and A Feinberg. "Multiple targets estimation and tracking for ADS-B radar system". *Digital Avionics Systems Conference, 2008. DASC 2008. IEEE/AIAA 27th*, 3–C. IEEE, 2008.

[28] IATA, Manny Gongora. "Air Traffic Surveillance Views and Expectations of Airspace Users". URL http://legacy.icao.int/nacc/meetings/2007/SURV_SEMI/Day03_IATA_Gongora.pdf, 2009.

[29] Instruction, CNSS. "4009,National Information Assurance Glossary, Committee on National Security Systems, May 2003". *Formerly NSTISSI*, 4009, 2010.

[30] ITU, RECOMMENDATION. "ITU-R M.1371: Technical Characteristics for an Automatic Identification System using Time-Division Multiple Access in the VHF Maritine Mobile Band". *ITUlS J*, 2001.

[31] Jochum, John R. *Encrypted Mode Select ADS-B for Tactical Military Situational Awareness*. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, April 2002.

[32] Kenney, Larry, Joe Dietrich, and Jerry Woodall. "Secure ATC surveillance for military applications". *Military Communications Conference, 2008. MILCOM 2008. IEEE*, 1–6. IEEE, 2008.

[33] Kinetic Avionics Limited. "SBS-3". URL http://www.kinetic-avionics.com, September 2013.

[34] Köllner, Günter. "Mode S Beast". URL http://www.modesbeast.com/systemdesign.html, September 2013.

[35] Koziel, Eric A. *Effects of Architecture on Information Leakage of a Hardware Advanced Encryption Standard Implementation*. Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, September 2012.

[36] Luby, Michael and Charles Rackoff. “How to construct pseudorandom permutations from pseudorandom functions”. *SIAM Journal on Computing*, 17(2):373–386, 1988.

[37] Magazu, Dominic. *Exploiting the Automatic Dependent Surveillance-Broadcast System via False Target Injection*. Master’s thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, March 2012.

[38] McCallie, Donald L. *Exploring Potential ADS-B Vulnerabilities in the FAA’s NextGen Air Transportation System*. Master’s thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, OH, June 2011.

[39] McLoone, Máire and John V McCanny. “High performance single-chip FPGA Rijndael algorithm implementations”. *Cryptographic Hardware and Embedded SystemsCHES 2001*, 65–76. Springer, 2001.

[40] Morris, Ben, Phillip Rogaway, and Till Stegers. “How to encipher messages on a small domain”. *Advances in Cryptology-CRYPTO 2009*, 286–302. Springer, 2009.

[41] Naor, Moni and Omer Reingold. “On the Construction of Pseudorandom Permutations: LubyRackoff Revisited”. *Journal of Cryptology*, 12(1):29–66, 1999.

[42] National Bureau of Standards. “Guidelines for Implementing and Using the NBS Data Encryption Standard”. *Federal Information Processign Standards Publication*, 74, 1981.

[43] National Institute of Standards and Technology. “FIPS-197, Advanced encryption standard (AES)”. *Federal Information Processing Standards Publication 197*, 2001.

[44] National Institute of Standards and Technology. “Critical Infrastructure Protection”. URL http://www.itl.nist.gov/ITLCIPBrochure.pdf, 2002.

[45] NATO Standardization Agency. *Technical Characteristics of IFF Mk XA and Mk XII Interrogators and Transponders*. Technical Report STANAG 4193 (Part I), North Atlantic Treaty Organization (NATO), 1998.

[46] NATO Standardization Agency. *Warship - Automatic Identification System (W-AIS)*. Technical Report STANAG 4668 (Edition 2), North Atlantic Treaty Organization (NATO), 2010.

[47] Offspark. “PolarSSL: Straightforward, Security Communication”. Company Website. URL https://polarssl.org/about-us, 2014.

[48] Patarin, Jacques. “Luby-Rackoff: 7 rounds are enough for 2 n (1- ε) security”. *Advances in Cryptology-CRYPTO 2003*, 513–529. Springer, 2003.

[49] Patarin, Jacques. “Security of random Feistel schemes with 5 or more rounds”. *Advances in Cryptology–CRYPTO 2004*, 106–122. Springer, 2004.

[50] Purton, L, Hussein Abbass, and Sameer Alam. "Identification of ADS-B system vulnerabilities and threats". *Australian Transport Research Forum, Canberra*. 2010.

[51] Randomness and Integrity Services Limited. "Random.org Pregenerated Random Numbers". Random binary file 2013-09-18. http://www.random.org/files. 24 October 2013.

[52] Rogaway, Phillip. "A Synopsis of Format-Preserving Encryption". Accessed 12 September 2013.

[53] RTCA Special Committee 180. "DO-254, Design Assurance Guidance For Airborne Electronic Hardware". *DO-254/ED-80*, 2000.

[54] RTCA Special Committee 186. "An Expanded Description of the CPR Algorithm". *1090-WP30-12*, 2009.

[55] RTCA Special Committee 186. "DO-260B with Corrigendum 1, Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance - Broadcast (ADS-B) and Traffic Information Services - Broadcast (TIS-B)". *ADS-B 1090 MOPS*, 2011.

[56] Single European Sky ATM Research. "SESAR at the ICAO Symposium: Towards One Sky". Press Release. URL http://www.sesarju.eu/news-press/news/sesar-icao-symposium-towards-one-sky-901, August 29 2011.

[57] Smith, AP and AD Mundra. "Impact of ADS-B on Controller Workload: Results from Alaska's Capstone Program". *25th Digital Avionics Systems Conference, 2006 IEEE/AIAA*, 1–9. IEEE, 2006.

[58] Soto, Juan. "Randomness testing of the AES candidate algorithms". *NIST. Available via csrc.nist.gov*, 1999.

[59] Spies, Terence. "Feistel finite set encryption mode". *NIST Proposed Encryption Mode. Available online at http://csrc. nist. gov/groups/ST/toolkit/BCM/documents/proposedmodes/ffsem/ffsem-spec. pdf*, 2008.

[60] Spies, Terence. "Format preserving encryption". *Unpublished white paper, www.voltage.com Database and Network Journal*, 2008.

[61] Stokes, Steven. "Air Force Research Laboratory Avionics Vulnerability Mitigation Branch (AFRL/RYWA) Research Portfolio". Private Communication, 27 Aug 2013.

[62] Thedford, Wiliam A. "USAF AFLCMC/HBAI (Global Traffic Management) ADS-B Specs". Private Communication, 09 Apr 2013.

[63] Trappe, Wade and Lawrence C. Washington. *Introduction to Cryptography: with Coding Theory*. Pearson Prentice Hall, Upper Saddle River, New Jersey, 2 edition, 2006.

[64] US Congress. “Public Law 108-176: Vision 100-Century of Aviation Reauthorization Act”. *Washington, DC: US Congress*. 2003.

[65] Vance, Joachim. “VAES3 scheme for FFX: An addendum to The FFX Mode of Operation for Format Preserving Encryption”, May 2011. A parameter collection for encipher strings of arbitrary radix with subkey operation to lengthen life of the enciphering key.

[66] Walker, John. “ENT: A Pseudorandom Number Sequence Test Program”. Tool description, http://www.fourmilab.ch/random/, January 2008. 24 October 2013.

[67] Williams, Nicolas. “A Pseudo-Random Function (PRF) for the Kerberos V Generic Security Service Application Program Interface (GSS-API) Mechanism”. *IETF RFC4401*, 2006.

[68] Xilinx Data Sheet. “Virtex-6 Family Overview”. URL http://www.xilinx.com/support/documentation/data_sheets/ds150.pdf, 2012.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD–MM–YYYY)* 27–03–2014	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* Oct 2012–Mar 2014

4. TITLE AND SUBTITLE

Secure ADS-B: Towards Airborne Communications Security in the Federal Aviation Administration's Next Generation Air Transportation System

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Agbeyibor, Richard C., Second Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-14-M-02

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RYWA Avionics Vulnerability Mitigation Branch
2241 Avionics Cir
WPAFB, OH 45433-7334
POC: Mr. Steven E. Stokes, Program Manager
steven.stokes@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RYWA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Distribution Statement A:
Approved For Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES

This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

The U.S. Congress has mandated that all aircraft operating within the National Airspace System, military or civilian, be equipped with ADS-B transponders by the year 2020. The ADS-B aircraft tracking system, part of the Federal Aviation Administration's NextGen overhaul of the Air Transportation System, replaces Radar-based surveillance with a more accurate satellite-based surveillance system. However, the unencrypted nature of ADS-B communication poses an operational security risk to military and law enforcement aircraft conducting sensitive missions. The non-standard format of its message and the legacy communication channels used by its transponders make the ADS-B system unsuitable for traditional encryption mechanisms. FPE, a recent development in cryptography, provides the ability to encrypt arbitrarily formatted data without padding or truncation. Indeed, three new algorithms recommended by the NIST, may be suitable for encryption of ADS-B messages. This research assesses the security and hardware performance characteristics of the FF1, FF2, and FF3 algorithms, in terms of entropy of ciphertext, operational latency and resource utilization when implemented on a Field-Programmable Gate Array. While all of the algorithms inherit the security characteristics of the underlying AES block cipher, they exhibit differences in their performance profiles. Findings demonstrate that a Bump-in-the-Wire FPE cryptographic engine is a suitable solution for retrofitting encryption to ADS-B communication.

15. SUBJECT TERMS

ADS-B Security, Format Preserving Encryption, Avionics, AES

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Maj. Jonathan Butts (ENG)
					19b. TELEPHONE NUMBER
U	U	U	UU	104	(937) 255-3636 x4332 Jonathan.Butts@afit.edu