ADS-B Communication Security: Assessing the Performance of Format-Preserving Lightweight Block Ciphers

Celdo Souza da Silveira, Juliano B. Lima, and José R. de Oliveira Neto

***Abstract*—In this work, we document an investigation regarding the vulnerabilities of ADS-B (automatic dependent surveillance-broadcast) communication, considering the risks to operational security. The ADS-B technology is a communication protocol to transmit surveillance information, that is, data related to position, altimetry, and speed, from the aircraft's avionics, an evolution to radars. We present an encryption solution for this communication using format-preserving encryption implemented in a microcontroller-embedded system. We also evaluate the use of lightweight symmetric block ciphers for better computational performance. When used as a pseudorandom function, we observe that such ciphers maintain a high entropy output value with low computational cost — up to sixteen times faster for similar entropy values. Finally, the computational performance obtained with the proposed solution is analyzed by processing real-time data from aircraft in the landing and takeoff phase at the international airport of Recife/Guararapes - Gilberto Freyre, based in Recife/PE, Brazil.**

***Index Terms*—ADS-B, Embedded Systems, Encryption, Lightweight.**

I. INTRODUCTION

Aviation plays a crucial role in society, directly resulting from technological advancements and significantly contributing to economic development. Every year, the number of people transported by air travel grows, driven by a globalized society where intercontinental transactions are increasingly necessary. Today, both national and multinational companies depend on this flexible mode of transportation. Aviation's significant role in society's daily life was made possible through technological evolution, primarily focusing on operational safety. In the past, pilots relied on ground references and illuminated beacons on runways for navigation and landing. Communication with pilots was mainly done through radio [1]. Later, in the 1930s, the RADAR (radio detection and ranging) technology, initially employed by the United Kingdom during World War II, allowed for detecting aircraft positions and velocities in flight through electromagnetic wave reflections [2]. This significant advancement considerably improved operational safety, especially in adverse atmospheric conditions [3].

As the number of airlines and air routes increased, enhancing airspace capacity and maintaining operational safety became essential. Therefore, air traffic control requires exact and reliable equipment. Radars played a crucial role in the evolution of air traffic, providing precise position, velocity, and identification information for aircraft ground control [4]. This information is known as surveillance data.

Subsequently, the development of secondary radars enabled the acquisition of more information from aircraft by transmitting a code that served as an inquiry to the plane, requesting data such as identification code and altitude. With technological progress, secondary radars could receive more information from aircraft and individually interrogate each plane. This mode of operation was termed Mode S, where "S" stands for "selective" [5], [6].

Despite radar technology advancements, obtaining more data during flight at shorter intervals was still needed. As a result, developing a new way of transmitting surveillance information to air traffic control allows other aircraft within the same airspace to share this information. This protocol was named ADS-B (automatic dependent surveillance-broadcast), an evolution of Mode S. ADS-B enables aircraft to broadcast various essential information for airspace control, including position, velocity, and identification. Additionally, it can provide call sign information, operational data, precision indicators, and integrity indicators. This type of communication transmits in broadcast without relying on ground systems' requests [5].

The ADS-B communication has many operational advantages, and we can highlight the interval between position reports, which is approximately 0.5 seconds. In contrast, the update interval of a secondary radar in terminal areas is 4 seconds. However, this communication has significant vulnerabilities concerning information security. The messages are transmitted through open radio channels, meaning they lack encryption and do not adopt any security measures to protect data transmission. An intruder can launch attacks against the transmitted information, such as eavesdropping, blocking, and message modification [7]–[9]

The importance and the solid attacking capability of aircraft operational status information, such as position messages, identification, and velocity, make them the primary targets

Submission: 2024-08-05, First decision: 2024-10-09, Acceptance: 2024-11-18, Publication: 2024-11-22.

C. S. da Silva and J. B. Lima are with Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco, Brazil (Orcid: 0000-0003-1737-9144, and 0000-0002-1474-1147; e-mail: {celdo.silveira,juliano.lima}@ufpe.br).

J. R. de Oliveira Neto is with Departamento de Engenharia Mecânica, Universidade Federal de Pernambuco, Brazil (Orcid: 0000-0002-3104-8985; e-mail: joserodrigues.oliveiraneto@ufpe.br).

This work was supported in part by Conselho Nacional de Desenvolvimento Científico e Tecnológico under grants 442238/2023-1, 312935/2023-4, and 405903/2023-5, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior under grant 88881.311848/2018-01, and Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco under grant APQ-1226-3.04/22.

Digital Object Identifier: 10.14209/jcis.2024.21

for malicious intruders. The ADS-B system types of attacks are eavesdropping, interference, message injection, message exclusion, and message modification [7], [8], [10], [11].

Transmission authentication may ensure security by preventing or detecting attacks in unidirectional communication, mitigating potential attacks on the ADS-B system [8]. The solutions proposed in the literature are organized as unencrypted and encrypted schemes [8], [12]. Encryption techniques are a communication protection method widely used today and their study is essential for applications in the ADS-B environment [12]–[14]. There are three potential encryption methods to ensure security in ADS-B communication: hashing, symmetric encryption, and asymmetric encryption. One issue with using hashing and asymmetric encryption is the need to change the ADS-B format [12].

Thus, format-preserving encryption (FPE), which involves encrypting plaintext under the control of a symmetric key while preserving the original plaintext format, appears to be a possible cryptographic solution for ADS-B communication [15]. FPE is highly versatile and provides a high level of security, provided that its pseudorandom function consists of ciphers with high resistance to attacks. Block ciphers developed for lightweight applications, meaning for embedded devices like microcontrollers, can be an excellent choice due to their low computational cost without compromising security [16].

Other authors have been exploring the use of FPE. In [17], Bellare and Rogaway propose a method called FF1. This method is based on small symbol space and short messages without reduced security. In [13], the use of FPE in messages ADS-B is proposed. In this work, simulations were performed with random data and also with fixed bytes and used entropy as a method to evaluate security performance. In [14], based on the study [13], the authors use real data ADS-B of aircraft CESSNA and suggest the use of blockchain for key transmission.

In this study, our goal is to advance research into the potential use of the FPE for securing the ADS-B protocol. We aim to develop a data encryption system specifically tailored for low-cost embedded devices. For this, we analyze the performance and security enhancements achieved by employing modern block ciphers optimized for lightweight applications when implemented in a low-cost microcontroller.

More precisely, we implemented the FF1 algorithm [17] in firmware with the possibility of choosing between four different block ciphers: AES, which is the cipher originally used in FF1; LEA [18], a lightweight standard cipher from South Korea; and ASCON and ASCON-PRF [19], a lightweight cipher recently standardized by the USA National Institute of Standards and Technology (NIST) [20]. In fact, we chose the LEA and ASCON ciphers because they are lightweight ciphers standardized by national agencies, while AES is used for comparison.

We designed the embedded system to communicate with a computer via USB CDC communication; using a software designed for this work, we send commands with the data to be encrypted and indicate the cipher to be used. We verified that all tested ciphers obtained similar results from the entropy calculation and statistical tests of the NIST suit [21]. However, in terms of computation speed, we verified that the FPE using a lightweight cipher function is faster than its version with the AES cipher: sixteen times in the case of LEA, more than three times for ASCON, and more than five times for ASCON-PRF.

The remainder of this paper is organized as follows: in Section II, we present several definitions that are useful in the course of this work; in Section III, we introduce the proposed system and how the experiments are carried out; in Section IV, we present the results and analysis of the tests. The paper closes with concluding remarks in Section V.

II. PRELIMINARIES

A. The ADS-B Protocol

The ADS-B protocol has this terminology because it is:

- ***automatic:*** does not need to be questioned to transmit the information;
- ***dependent:*** depends on onboard equipment to transmit data, such as GPS (global positioning system), altimeter, barometer, etc.
- ***broadcast:*** transmits the information in all directions to everyone within range;

The most common information transmitted by ADS-B is position, altitude, and speed. However, this protocol allows sending various other information to air traffic control on the ground, such as call indicators, accuracy indicators, integrity indicators, and operational status. GPS determines position, the speed is derived from GPS and inertial systems, and altitude is composed of barometric and GPS data, with barometric data obtained from atmospheric pressure sensors. ADS-B works at a frequency of 1090 MHz and can work at a frequency of 978 MHz for altitudes below 18,000 feet in the USA. When operating at 978 MHz, ADS-B can transmit supplemental flight information such as meteorological data [5].

ADS-B offers a position accuracy of 92.6 m and speed accuracy of around 10 m/s, with the update rate of this information every second. This performance standard allows lateral separation to be reduced from 90 NM (nautical mile) to 20 NM and longitudinal separation to be reduced from 80 NM to 5 NM in airspace without radar detection, with separation by level (altitude) performed following the operational models of the control agencies and in compliance with what is established for airways [22].

1) ADS-B system architecture: The ADS-B system has two subsystems: ADS-B IN and ADS-B OUT. ADS-B OUT transmits information periodically in all directions, and the receiving subsystem, ADS-B IN, receives and processes this data. The transmitter on the aircraft gets data from the onboard avionics, encapsulates the information, and transmits it to receivers on the ground, by VHF (very high frequency) stations, or to another aircraft [22]. An aircraft has only the ADS-B OUT or ADS-B IN and ADS-B OUT.

ADS-B OUT packets are made up of 112 bits, with the first 8 bits indicating the data format, the next 24 bits referring to the aircraft's unique indicator determined by the ICAO (International Civil Aviation Organization), the following 56 bits being surveillance data and the final 24 bits are the CRC (cyclic redundancy check) of the packet [22]. The modulation used to transmit the information is pulse position modulation (PPM) with Manchester coding.

ADS-B was designed to be compatible with existing surveillance systems and thus facilitate the transition between systems. ADS-B is very similar to secondary radar mode S, with the main difference being that interrogation by remote equipment is unnecessary, as the aircraft's transponder constantly transmits information [6].

B. Security in the ADS-B Communication

ADS-B technology provides greater operational security, especially in regions without radar detection. Furthermore, the cost to install and maintain is more attractive compared to primary and secondary radars. However, it is an open protocol without encryption, making it vulnerable to ground-to-ground, ground-to-air, and air-to-air attacks [23].

In [23], the authors point out five ADS-B vulnerabilities: spying, interference, message injection, and message deletion.

1) Spying: The vulnerability of spying on aircraft operational status information (aircraft recognition) is characterized by obtaining ADS-B data from the corresponding airspace, using the ADS-B IN device, that is, reading transmitted ADS-B data via aircraft ADS-B OUT transponder. According to [8], since the creation of ADS-B communication, some initiatives have used data capture to provide a legitimate service to users. In this context, we can mention flightradar24.com, which provides real-time information about flights and data from aircraft en route through the ADS-B network, with more than 20,000 receivers worldwide. However, it does not exclude that some malicious attackers could use this vulnerability to launch complex attacks.

2) Interference: Blocking the transmission of an ADS-B message in specific airspace, using an ADS-B transmitting device with sufficiently high transmit power in the relevant frequency band, is an explicit interference technique with ADS-B information with significant operational impact for airspace control affecting the integrity and availability of data.

There are two main types of jamming attacks against ADS-B communication, according to [8]: ground station denial of service and aircraft denial of service. The objective of these attacks is to disrupt the monitoring network by blocking the communication channel. Launching an attack on a ground station is more accessible than directly on an aircraft, requiring less energy.

3) Message Injection: Inputting false aircraft information into specific flight scenarios can confuse air traffic control systems (ghost aircraft target injection). This injection is made using a transmitting device with high enough transmit power in the relevant frequency range and capable of generating the correct modulation that complies with the ADS-B message format. This effect is achieved by not authenticating messages [23].

4) Message Deletion: Deleting some or all of the information in a message (missing aircraft) is an attack implemented at the physical layer through constructive or destructive interference.

Constructive interference causes a large number of bit errors. Since the CRC of transmitted ADS-B messages can correct a maximum of 5 bits per message, the receiver will discard the message as corrupt if it exceeds this limit [23].

On the other hand, deleting the message can impact the surveillance system, temporarily causing the aircraft to disappear from the air traffic control screen. It is essential to note that our surveillance systems, such as primary and secondary radar and multilateration systems, are resilient and capable of identifying such anomalies.

5) Message Modification: Modifying the information contained in a message can be done through obfuscation and bit inversion in the message packet.

Message modification, a typical spoofing attack, can be subtly executed. For instance, if an attacker continually changes aircraft position information in ADS-B messages by small amounts, this is considered a "frog boil" spoofing attack. The subtlety of such attacks makes them particularly dangerous, as they can go undetected by other surveillance technologies and positioning technology due to accuracy issues, leading to incorrect guidance to air traffic controllers or delays in the response of the air traffic prevention system.

C. Format-Preserving Encryption

Due to its importance, researchers have been focusing on the vulnerability problem in ADS-B communication and have proposed solutions to mitigate most of the problems listed [23], intending to maintain the requirements of integrity, availability, and confidentiality [12]. Among the proposed solutions, symmetric encryption solutions stand out [17], because this principle of maintaining the format of data packages is fundamental in aviation, as a change to the protocol already defined and in use would have a huge impact on onboard equipment and ground stations. In this way, the solution appears as a possible cryptographic alternative for ADS-B communication.

Regarding the vulnerabilities presented in Section II-B, encryption can mitigate attacks related to *spying* since if the attacker does not have access to the secret key, one will not access sensitive information. Regarding *message ingestion*, in transmission with message authentication, it is unlikely that a fake aircraft will have access to a valid key to impersonate a real aircraft, minimizing the effects that an unauthenticated attacker can have on the air traffic control system. Also, *message modification* becomes unlikely in a scenario where the modification of a bit in the plaintext generates a significant change in the ciphertext [8], [12]. However, the encryption of ADS-B messages does not make the system more resistant to *interference* and *message deletion* attacks.

Format-preserving encryption (FPE) consists of an encryption technique based on symmetric encryption, in which the cipher text maintains the same format as the original text. If the original message is composed of an alphabet with decimal numbers, the encrypted message will also be composed of an alphabet with decimal numbers. Furthermore, the size of the data packet remains unchanged, with the same number of bits/bytes/characters as the original message. FPE allows a more straightforward migration path when encryption is added to legacy systems [15].

FPE is deterministic; whenever a specific key encrypts a plain text, the result will be the exact cipher text [24]. One advantage is that data does not necessarily need to be binary in FPE. Any finite set of symbols, such as decimal numerals, can be encrypted using this method and will have their set preserved, including the size of the sequence of symbols. In [17], the authors describe the FPE operating modes in detail.

The United States National Institute of Standards and Technology (NIST) is part of the US Department of Commerce. Currently, the NIST approves the encryption algorithms used in the USA. According to [24], three FPE modes were submitted for approval in November 2010: FFX, BPS, and VAES-3. The three algorithms use the Feistel network and are named FF1 (FFX), FF2 (VAES-3) and FF3 (BPS). In [25], each operating mode is detailed and comments on cryptographic security.

The FPE framework provides confidentiality regarding cleartext data, and each mode also takes an additional input called *Tweak*, which does not necessarily need to be secret. The *Tweak* can be considered as a part of the key that can be changed, as it determines the encryption and decryption functions.

The US National Security Agency (NSA) notified the NIST about the security flaw in the FF2 method. This note describes a theoretical attack, with a choice of plain text, that shows that the security strength of FF2 is less than 128 bits [26].

The FF3 algorithm uses a Feistel network with eight rounds. The authors F. Betul Durak and Serge Vaudenay carried out a study proving the existence of a vulnerability in the FF3 method, as described in [27], which allows breaking the encryption of the Feistel network with eight rounds when using a small domain. To do this, they exploit the difference in the domain of the algorithm. They consider the permutation of round functions for the attack and use *Tweak* values with known plaintext.

NIST, in response to the work of Durak and Vaudenay [28], reported its intention to revise the FF3 specification and, to this end, verified the possibility of reducing the size of its *Tweak* parameter from 64 bits to 48 bits or to withdraw approval of FF3. In SP 800-38G Revision 1, the *Tweak* parameter was reduced to 56 bits. The revised FF3 was named FF3-1.

Based on the highlighted safety considerations, we use the FF1 mode in this work, as described in [25].

1) FF1 Mode: The FF1 mode is derived from the FFX algorithm developed by Bellare, Rogaway, and Spies and submitted to NIST in 2010. The purpose of this mode is to enable the encryption of messages with a small symbol space and relatively short messages without compromising security. This algorithm is based on the Feistel network and uses the 128-bit AES block cipher as the round function. A significant advantage of this method is the possibility of working with messages of different sizes and formats, including with an odd number of symbols per message [17], [24].

2) Feistel Network Structure: The structure of the Feistel network consists of several iterations called rounds, which perform the confusion and diffusion functions [17], [25]. This network allows similar encryption and decryption and works as follows:

1) The initial data is divided into two parts, trying to maintain the same number of bits in each half of the divided message;
2) Application of a round function (pseudorandom function), controlled by a cryptographic key, in one of the parts of the plain text;
3) Carrying out a permutation between the initially divided parts to compose the next round.

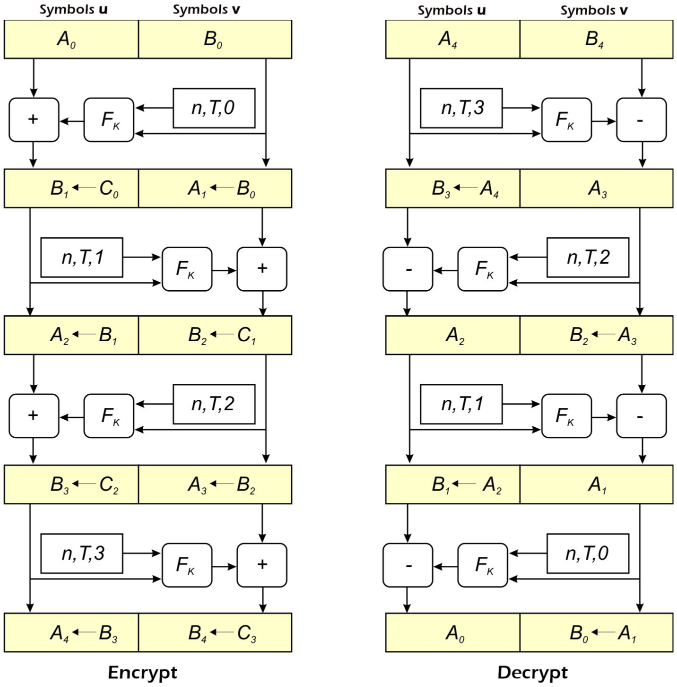


Fig. 1: Structure of the Feistel network.

Figure 1 shows the structure of this method with an example of 4 rounds. As described by [29], Feistel networks allow us to create strong ciphers. One of the significant advantages is that the decryption process is very similar to the process for encrypting plain text. The input data is divided into two numeric strings, named U and V, with u being the size of the string U and v being the size of the string V. The total number of characters is n, $n = u + v$. During round i, the round function, F_k, is applied to one of the input *strings*, called B_i, with size n. The parameters *Tweak* T and round number i are added to the input. The result is used to modify the other *string*, called A_i, via modular addition. The *string* representing the resulting number is called C_i in a temporary variable. The names of the two parties are swapped for the next round, where C_i will be B_{i+1} and B_i will be A_{i+1}, for example. In Figure 1, the rectangles are not the same size to demonstrate that the data can have different sizes. That is, u will differ from v if n is odd.

For decryption, Feistel's Structure is practically identical to encryption, but with small differences, namely:

1) The index orders are reversed;

2) The data change rule in the round function is changed so that at the function's entry, it will always be in the part of A_{i+1} and not in B_i. The output will be combined with B_{i+1}, and no longer with A_i, precisely to produce A_i and not B_{i+1};
3) Modular addition is replaced by modular subtraction.

3) Round Function: Figure 1 shows that each round comprises a function F_k based on some block cipher, such as AES. In the work developed by [24], the AES cipher works with cryptographic keys of 128, 192, and 256 bits. This block function can receive four different parameters: information block (*n*), round number, *tweak* value (*T*), and cryptographic key (*K*). In method FF1, the pseudorandom function PRF calls the block cipher, named $CIPH_k$. The number of block cipher calls is directly related to the size of the block to be encrypted. Since each cipher block is 128 bits, any block larger than this will require more than one call.

Although the AES cipher is secure and widely used in several cryptographic systems, it can be computationally expensive when the algorithm in question needs to be executed by an embedded system. Due to this reason, we tested the replacement of AES as the PRF function in the FF1 round function with the lightweight algorithms ASCON, ASCON-PRF, and LEA [18].

4) LEA: LEA (lightweight encryption algorithm) was developed as a security solution for embedded devices with limited memory and processing speed resources. This cipher was developed in South Korea, and most applications are in microcontrollers operating in IoT solutions [18]. LEA uses status values formed by 32-bit words. The encryption and decryption functions use operations called ARX (addition, rotation, XOR); that is, they involve modular addition in 32 bits, bit rotation, and Exclusive OR. This algorithm focuses on 32- or 64-bit processors but can operate on 8- or 16-bit microcontrollers. The block size defined for this algorithm is 128 bits, and the key length can be 128, 192, or 256 bits. The key length establishes the name of the cipher, that is, LEA-128, LEA-192, or LEA-256.

5) ASCON: ASCON was developed in 2014 by researchers at the Graz University of Technology in Austria, Infineon Technologies, Lamarr Security Research, and Radboud University in the Netherlands. ASCON is a cipher suite that provides authenticated encryption with associated data (AEAD) with *hash* functionality. It was the first choice in the CAESAR competition (Competition for Authenticated Encryption: Security, Applicability, and Robustness) in 2019 for Case 1, referring to *lightweight* applications sized for low-performance devices, and in 2023 the NIST published [20] describing the evaluation criteria and the process for selecting authenticated encryption and *hash* schemes suitable for applications in constrained environments. The ASCON suite is suitable for the following possible applications [19]:

- Authenticated Encryption: ASCON-128 and ASCON-128a
- Hash: ASCON-HASH and ASCON-XOF
- Pseudorandom function: ASCON-PRF

The ASCON structure was developed based on sponge construction. Sponge functions use a fixed amount of permutations and padding rules. The approach that led to the creation of the sponge method is related to the need to perform a function *hash* with a variable input length and output with a determined length [30].

III. COMPUTER EXPERIMENTS

In this section, we describe the experiments carried out by us and analyze the obtained results. The proposed system captures ADS-B OUT signals transmitted by aircraft and performs FPE encryption using four variations of pseudorandom functions, based on AES block ciphers [31], LEA [18], ASCON and ASCON-PRF [19].

In order to implement the system, we use embedded devices to adapt the encryption solution to the transponder installed on aircraft without the need to replace the entire equipment. The FPE/FF1 algorithm and the block ciphers used as pseudorandom functions mentioned in this research can be applied to solutions developed in FPGA (*field-programmable gate array*) or in microcontrollers [32] due to their structure. In this research, we developed the hardware, firmware, and software for evaluating block ciphers in a format-preserving environment. We chose to work with the LPC17xx family of 32-bit microcontrollers, more precisely, the LPC1768 (ARM CORTEX M3) [33]; this is a low-cost and low development time device, even more so with the use of the CooCox CoIDE IDE and GCC compiler (*GNU Compiler*). The 32-bit architecture allows us to implement all block ciphers in this work, mainly with LEA [34].

A. Proposed Model

Figure 2 shows the model for receiving, processing, and encrypting ADS-B data packets. The system comprises an RTL-SDR device (*Realtek software-defined radio*), responsible for receiving ADS-B data transmitted by aircraft, modulated at 1090 MHz, and together with the RTL1090 software. This information is encoded in words in hexadecimal format. Upon receiving this data, the developed hardware performs the encryption and retransmits the encrypted information for analysis through the analyzer software.

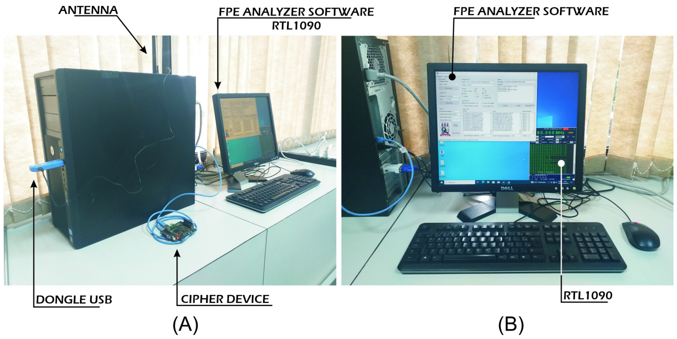


Fig. 2: Proposed model platform.

1) Software Defined Radio: A radio system in which software definitions perform the physical layer functions and all signal processing is called software-defined radio or SDR. The origin of software-defined radios is in the military environment, from which it migrates to the civil environment [35].

These devices' flexibility and low cost have made them widely used today.

The RTL-SDR device is a currently widely used software-defined radio via USB, DVB-T (digital video broadcast-terrestrial) [35]. The device uses the RTL2832U integrated circuit, [36], which has the following technical specifications [37]:

- Bandwidth: 2.4 MHz;
- Frequency Range: 500 kHz - 1766 MHz;
- Input impedance: 50 Ohms.

2) RTL1090 Software: The RTL1090 application, developed by the German company Jetvision, runs on X86 platforms and the Windows operating system, and is capable of processing ADS-B and S mode data when in conjunction with RTL-SDR that operates at 1090MHz [38]. This program receives data from the RTL-SDR and analyzes the preamble of ADS-B messages to extract the information packets later [39]. It allows data to be transmitted via *socket* TCP/IP in ASCII format (*American standard code for information interchange*), functioning as a server in a client-server configuration.

3) Developed Cipher: The developed cipher encrypts ADS-B data captured by the RTL-SDR and extracted by the RTL1090. The format of this information is described in Section II-A1. The data is encrypted with FPE/FF1, using four block cipher options as a pseudorandom function:

- AES-128
- LEA-128
- ASCON-128
- ASCON-PRF

The encrypted data is sent to the computer, where the FPE Analyzer program processes it and stores it for subsequent entropy and performance analysis. Statistical analysis of pseudorandom functions was performed based on the NIST test suite.

4) NIST Test Suite: NIST developed a package composed of 15 statistical tests to test the randomness of random number generators and pseudorandom number generators [21]. The 15 tests are:

- **Frequency Test (Monobit):** This test determines whether the number of "zeros" and "ones" in a sequence is approximately the same as expected for a truly random sequence.
- **Frequency Test within a Block:** This test aims to determine whether the frequency of "ones" numbers in a block of M bits is approximately $M/2$, as expected under an assumption of randomness.
- **Execution Test:** The purpose of the execution test is to determine whether the number of executions of "ones" and "zeros" of various lengths is that expected for a random sequence, that is, the total number of runs in the sequence, where a run is an uninterrupted sequence of identical bits.
- **Longest Run of 1s in a Block Test:** The purpose of this test is to determine whether the length of the longest string of "ones" numbers in a string is consistent with the length of the longest string of "ones" that would be expected in a random sequence. The irregularity in the size of the expected length of the most extended series of "ones" numbers implies that there is also an irregularity in the expected length of the most extended series of "zeros" numbers.
- **Binary Matrix Rank Test:** This test aims to check whether there is a linear dependence between *substrings* of fixed length concerning the original sequence.
- **Discrete Fourier Transform Test (Spectrum):** This test aims to detect periodic characteristics in the tested sequence. These periodic characteristics would indicate a deviation from the randomness assumption.
- **Pattern Non-Overlap Test:** The objective of this test is to check whether there are occurrences of a specific non-periodic pattern in the generated sequences. For this test, a *m* bit window is used to search for a particular *m* bit pattern. The window slides to a one-bit position if the pattern is not found. If the pattern is found, the window resets to the bit after the pattern is found, and the search resumes.
- **Pattern Overlap Test:** This test works very similarly to the Pattern Non-Overlap Test. The difference is that when the pattern is found, the window slides just one bit before resuming the search.
- **Maurer's Universal Statistics:** The objective of the test is to detect whether or not the sequence can be significantly compressed without loss of information since the significantly compressible sequence is considered nonrandom.
- **Linear Complexity Test:** This test aims to determine whether the sequence is complex enough to be considered random since long LFSR and very short LFSR characterize random sequences, implying non-randomness.
- **Serial:** This test aims to determine whether the number of occurrences of bit-overlapping patterns is approximately the same as expected for a random sequence.
- **Approximate Entropy Test:** This test compares the frequency of overlapping blocks of two consecutive lengths with the expected result for a random sequence.
- **Cumulative Sums Test:** The purpose of the test is to determine whether the cumulative sum of the partial sequences that occur in the tested sequence is too large or too small about the expected behavior of this cumulative sum for random sequences. For certain types of nonrandom sequences, the excursions of this random walk from zero will be significant.
- **Random Excursion Test:** This test consists of randomly given unit-length steps that begin and return to the origin. This test aims to determine whether the number of visits to a given state within a cycle deviates from what would be expected for a random sequence. This test is a series of eight tests.
- **Variant Test for Random Tours:** This test aims to detect deviations from the expected number of visits to various states in the random walk. This test is a series of eighteen tests.

According to [21], to run all the tests in the package, a mass of data above 10^6 bits is necessary for each *bitstreaming*. In

TABLE I: The result of the uniformity of p-value and the proportion (PROP.) of processed sequences applying the NIST suite for each cipher used.

STATISTICAL TEST	AES			LEA			ASCON			ASCON-PRF		
	P-VALUE	PROP.	RESULT	P-VALUE	PROP.	RESULT	P-VALUE	PROP.	RESULT	P-VALUE	PROP.	RESULT
Frequency	0,066882	9/10	Random	0,991468	10/10	Random	0,350485	10/10	Random	0,122325	10/10	Random
BlockFrequency	0,739918	10/10	Random	0,213309	9/10	Random	0,122325	10/10	Random	0,002043	8/10	Random
CumulativeSums	0,637032	9/10	Random	0,7227795	10/10	Random	0,2086835	10/10	Random	0,534146	10/10	Random
Runs	0,739918	10/10	Random	0,911413	10/10	Random	0,739918	10/10	Random	0,739918	10/10	Random
LongestRun	0,534146	10/10	Random	0,350485	10/10	Random	0,350485	10/10	Random	0,350485	10/10	Random
Rank	0,350485	10/10	Random	0,122325	10/10	Random	0,534146	10/10	Random	0,350485	10/10	Random
FFT	0,911413	10/10	Random	0,739918	10/10	Random	0,122325	10/10	Random	0,350485	10/10	Random
NonOverlappingTemplate	0,486213	10/10	Random	0,476765	9/10	Random	0,493612	10/10	Random	0,516559	10/10	Random
OverlappingTemplate	0,213309	10/10	Random	0,213309	10/10	Random	0,534146	10/10	Random	0,739918	10/10	Random
Universal	0,739918	10/10	Random	0,991468	9/10	Random	0,911413	10/10	Random	0,739918	10/10	Random
ApproximateEntropy	0,534146	10/10	Random	0,213309	10/10	Random	0,534146	10/10	Random	0,122325	9/10	Random
RandomExcursions	—-	5/5	Random	—-	7/7	Random	—-	6/6	Random	—-	5/5	Random
RandomExcursionsVariant	—-	5/5	Random	—-	6/7	Random	—-	6/6	Random	—-	5/5	Random
Serial	0,4766135	10/10	Random	0,4034	10/10	Random	0,4034	10/10	Random	0,000319	5/10	No-Random
LinearComplexity	0,739918	10/10	Random	0,739918	9/10	Random	0,035174	10/10	Random	0,350485	10/10	Random

this way, random ADS-B data is applied to the specified block ciphers for further evaluation of the pseudorandom functions in the NIST suite. The mass of information extracted from each cipher comprises 10 packets of 1,000,000 bits each, totaling 10^7 bits per cipher. The suite is available on the NIST website at https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software, accessed on July 1, 2024.

Table I displays the results obtained for each of the tests in the NIST suite and compares these results between the ciphers used as pseudorandom functions. The pseudorandom functions generated by the presented block ciphers passed all NIST tests except ASCON-PRF, which did not pass a single test, serial correlation. Therefore, the performance is considered satisfactory for application in a format preservation environment for ADS-B data. We use the statistical tests mainly to validate our implementations of the chosen block cipher since all used block ciphers have already undergone multiple tests before and after standardization.

5) Hardware: The cipher implementation is based on the ARM Cortex M3 microcontroller model LPC1768 from the manufacturer NXP semiconductors. This component operates with a clock frequency of up to 100 MHz and has 512 kB of flash memory and 64 kB of RAM. Additionally, there are several peripherals, such as an 8-channel DMA controller, 4 UART, 2 CAN interfaces, 2 SSP controllers, 12-bit and 8-channel ADC converter, and 10-bit DAC, among others [33]. The extracted data is sent to the cipher via USB CDC communication at a rate of 115200 bps (bits per second), and the USB port provides power to the development board.

The circuit has two USB ports, one for programming and another for communicating with the application. To carry out the USB to UART conversion of the LPC1768, the CP2102 integrated circuit from the manufacturer Silicon Labs [40] is used. The LPC1768 has an integrated physical interface for USB communication through ports P0.29 and P0.30. This interface is used to communicate between the cipher and the application on the computer. The class implemented for this communication was CDC (*communication device class*), which simulates serial communication between devices.

6) Firmware: The code uses C language, and its fundamental function is to encrypt ADS-B data packets. Figure 3 displays the developed firmware flowchart.

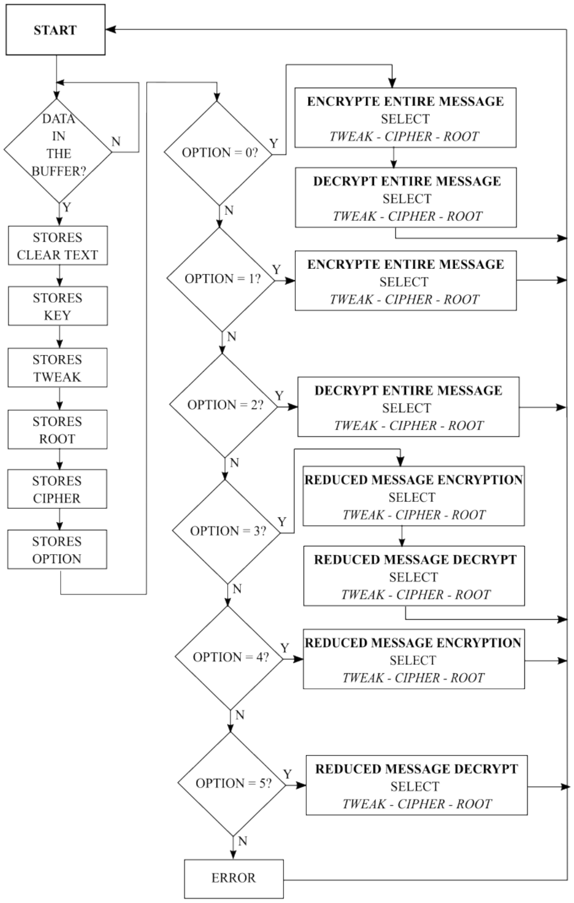

START
DATA IN THE BUFFER?
N
Y
STORES CLEAR TEXT
STORES KEY
STORES TWEAK
STORES ROOT
STORES CIPHER
STORES OPTION
OPTION = 0?
Y
N
ENCRYPTE ENTIRE MESSAGE SELECT TWEAK - CIPHER - ROOT
DECRYPT ENTIRE MESSAGE SELECT TWEAK - CIPHER - ROOT
OPTION = 1?
Y
N
ENCRYPTE ENTIRE MESSAGE SELECT TWEAK - CIPHER - ROOT
OPTION = 2?
Y
N
DECRYPT ENTIRE MESSAGE SELECT TWEAK - CIPHER - ROOT
OPTION = 3?
Y
N
REDUCED MESSAGE ENCRYPTION SELECT TWEAK - CIPHER - ROOT
REDUCED MESSAGE DECRYPT SELECT TWEAK - CIPHER - ROOT
OPTION = 4?
Y
N
REDUCED MESSAGE ENCRYPTION SELECT TWEAK - CIPHER - ROOT
OPTION = 5?
Y
N
REDUCED MESSAGE DECRYPT SELECT TWEAK - CIPHER - ROOT
ERROR

Fig. 3: Flowchart of the developed firmware.

As shown in Figure 3, the algorithm flowchart shows that the system waits for a data package to arrive. Figure 4 shows the package format. After receiving the package, the firmware stores the values of the fields relating to the data to be encrypted and the configuration data. Subsequently, the configuration data determines which cipher and root to use, as well as the operating mode option:

- Encrypt: performs FPE/FF1 encryption with the block cipher selected as a pseudorandom function;

CLEAR TEXT LENGHT (1 byte)	CLEAR TEXT (Variable)	KEY LENGHT (1 byte)	KEY (Variable)	TWEAK LENGHT (1 byte)	TWEAK (Variable)	ROOT (1 byte)	CIPHER (1 byte)	OPTION (1 byte)

Fig. 4: Data packet format.

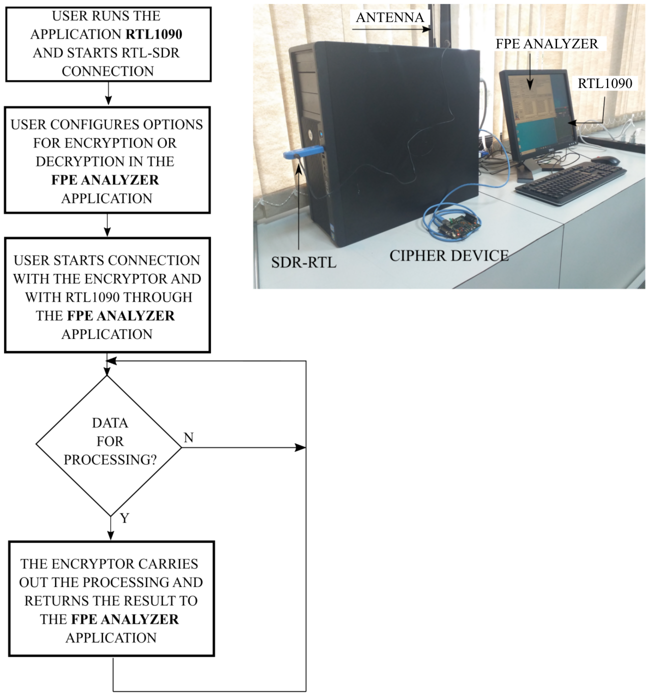

USER RUNS THE APPLICATION RTL1090 AND STARTS RTL-SDR CONNECTION
USER CONFIGURES OPTIONS FOR ENCRYPTION OR DECRYPTION IN THE FPE ANALYZER APPLICATION
USER STARTS CONNECTION WITH THE ENCRYPTOR AND WITH RTL1090 THROUGH THE FPE ANALYZER APPLICATION
DATA FOR PROCESSING?
N
Y
THE ENCRYPTOR CARRIES OUT THE PROCESSING AND RETURNS THE RESULT TO THE FPE ANALYZER APPLICATION
ANTENNA
FPE ANALYZER
RTL1090
SDR-RTL
CIPHER DEVICE

Fig. 5: Flowchart of system operation.

- Decrypt: performs FPE/FF1 decryption with the selected block cipher;
- Encrypt and Decrypt: performs the encryption and, immediately after, performs the FPE/FF1 decryption with the selected block cipher to verify that the algorithms return the original clear text without errors.

For each option above, it is possible to select in advance whether the header and the CRC field of the ADS-B package are part of the message to be encrypted or decrypted. The developed code relies on the four block ciphers treated in this research, with the FPE Analyzer Software selecting which cipher each message transmitted uses.

7) FPE Analyzer Software: We developed an application to configure the cipher and serve as a user interface for obtaining clear text and encrypted data. This program was developed in C# and is called *FPE Analyzer Software*. Communication with the cipher is carried out via a serial port class emulating a serial over a USB port. The connection to the RTL1090 program is made via a TCP socket using the client-server configuration, where the RTL1090 is the server. Figure 5 demonstrates how the system works, obtaining data through the SDR-RTL and interconnections with the developed FPE Analyzer software, cipher, and RTL1090.

Figure 6 displays the program's working screen. In the upper left field, one has the fields corresponding to the communication configuration with the cipher and the RTL1090. On the upper right side, one has the sector named "Static". This function encrypts or decrypts the information in the corresponding *textbox*. In the central field, we see the options that allow the user to configure the cipher to perform encryption, decryption, or both.

Moreover, to enable the choice of the desired block cipher as a pseudorandom function of the FPE, the root option (symbols), whether there will be encryption of the 112 bits of the ADS-B message or just 80 bits (header and CRC discard), use of TWEAK with the aircraft's ICAO code and the option to save the encrypted data in a binary file. The "Real Time" field processes the TCP/IP communication information. It displays the input message, encrypted message, and time between reception of the message by *socket* and reception of the encrypted message by serial when transmitted by the cipher. The "Debug" field provides the number of bytes received in the last ADS-B packet, the 24-bit ICAO code of the detected aircraft, and the entropy of the input information and the encrypted message.

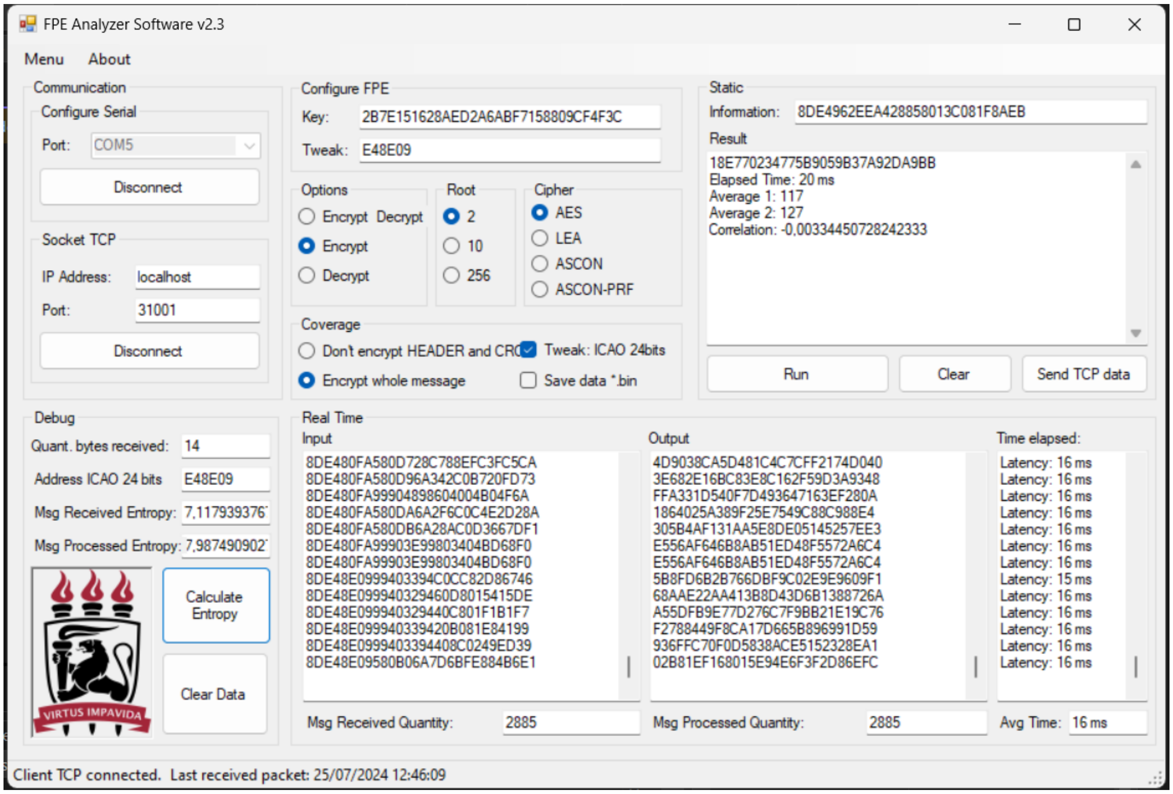
It is worth mentioning that the option not to encrypt the HEADER and CRC causes the firmware to insert the original clear text header into the encrypted message and insert a new recalculated CRC. Regarding TWEAK, the program automatically inserts the value of the aircraft's 24-bit ICAO code as the value for *TWEAK* at the time of encryption.

8) Static Option: In the "Static" configuration, the user enters the information to be encrypted, the cryptographic key, and a *TWEAK*. The root choice must depend on the alphabet of information and the key. In the corresponding *TextBox*, the encryption, decryption, or both, as selected, will be displayed, in addition to the processing time that the microcontroller took to operate.

9) Real-Time Option: In real-time, the program receives all the data sent by the RTL1090 through the TCP socket, retransmits it to the cipher, and receives the encrypted data. The format of the received data and the encryption dynamics will depend on the options the user selects in the FPE configuration. The program calculates the time it took the microcontroller to perform the selected operation and displays the result of the operation after it. The received and processed data are saved in binary files in the application's executable directory when selected by the "Write Data *.bin" option. The program creates a file for each 24-bit ICAO identifier for each day of operation and a file with all aircraft per day. The purpose of making these files is the subsequent data analysis for clear text and encrypted data.

10) ADS-B Server Software: During the research development, it became necessary to process the ADS-B data acquired from the aircraft using the RTL1090 and store it on the computer. Therefore, as a measure of availability and based on the data saved by the previous application, a server program in C# was developed that makes it possible to transfer messages via ADS-B message to the FPE Analyzer Software. Figure 7 shows how it works when using the ADS-B Server application instead of the RTL1090.

As seen in Figure 8, the user selects the IP and communication port through this program and opens the *socket* for the client to connect. The "Open File" option specifies the file



Communication — Configure Serial — Port: COM5

Socket TCP — IP Address: localhost — Port: 31001

Configure FPE — Key: 2B7E151628AED2A6ABF7158809CF4F3C — Tweak: E48E09

Options: ○ Encrypt Decrypt ● Encrypt ○ Decrypt

Root: ● 2 ○ 10 ○ 256

Cipher: ● AES ○ LEA ○ ASCON ○ ASCON-PRF

Coverage: ○ Don't encrypt HEADER and CRC ☑ Tweak: ICAO 24bits ● Encrypt whole message □ Save data *.bin

Static — Information: 8DE4962EEA428858013C081F8AEB

Result

```
18E770234775B9059B37A92DA9BB
Elapsed Time: 20 ms
Average 1: 117
Average 2: 127
Correlation: -0,0033445072824233
```

Debug — Quant. bytes received: 14 — Address ICAO 24 bits: E48E09 — Msg Received Entropy: 7,117939376 — Msg Processed Entropy: 7,987490902

Real Time

```
Input                           Output                          Time elapsed:
8DE480FA580D728C788EFC3FC5CA    4D9038CA5D481C4C7CFF2174D040    Latency: 16 ms
8DE480FA580D96A342C0B720FD73    3E682E16BC83E8C162F59D3A9348    Latency: 16 ms
8DE480FA99904898604004B04F6A    FFA331D540F7D493647163EF280A    Latency: 16 ms
8DE480FA580DA6A2F6C0C4E2D28A    1864025A389F25E7549C88C988E4    Latency: 16 ms
8DE480FA580DB6A28AC0D3667DF1    305B4AF131AA5E8DE05145257EE3    Latency: 16 ms
8DE480FA99903E99803404BD68F0    E556AF646B8AB51ED48F5572A6C4    Latency: 16 ms
8DE480FA99903E99803404BD68F0    E556AF646B8AB51ED48F5572A6C4    Latency: 16 ms
8DE48E0999403394C0CC82D86746    5B8FD6B2B766DBF9C02E9E9609F1    Latency: 15 ms
8DE48E099940329460D8015415DE    68AAE22AA413B8D43D6B1388726A    Latency: 16 ms
8DE48E099940329440C801F1B1F7    A55DFB9E77D276C7F9BB21E19C76    Latency: 16 ms
8DE48E099940339420B081E84199    F2788449F8CA17D665B896991D59    Latency: 16 ms
8DE48E0999403394408C0249ED39    936FFC70F0D5838ACE5152328EA1    Latency: 16 ms
8DE48E09580B06A7D6BFE884B6E1    02B81EF168015E94E6F3F2D86EFC    Latency: 16 ms
```

Msg Received Quantity: 2885 — Msg Processed Quantity: 2885 — Avg Time: 16 ms

Client TCP connected. Last received packet: 25/07/2024 12:46:09

Fig. 6: Working screen of the proposed FPE Analyzer Software.

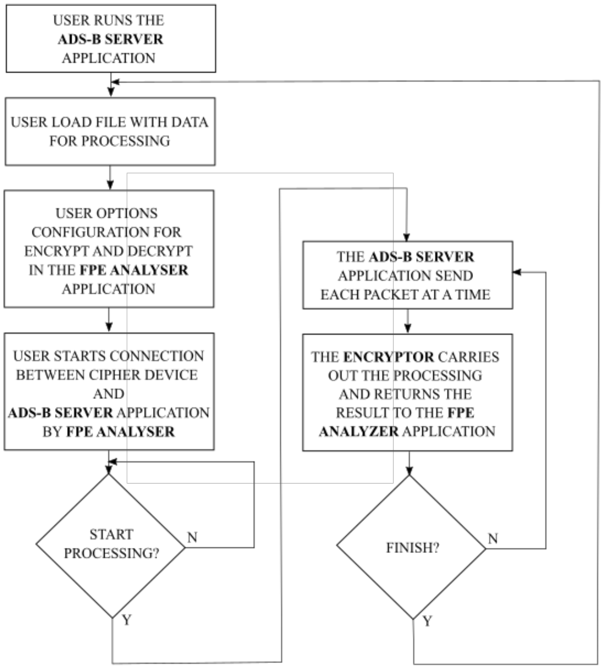
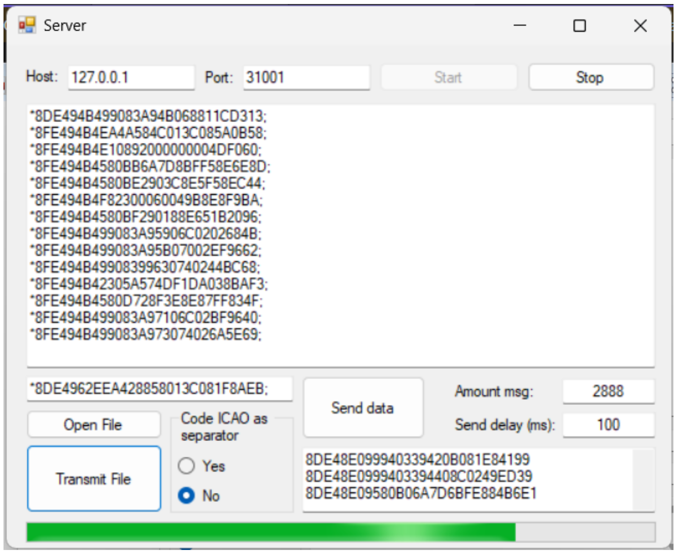

USER RUNS THE ADS-B SERVER APPLICATION
USER LOAD FILE WITH DATA FOR PROCESSING
USER OPTIONS CONFIGURATION FOR ENCRYPT AND DECRYPT IN THE FPE ANALYSER APPLICATION
USER STARTS CONNECTION BETWEEN CIPHER DEVICE AND ADS-B SERVER APPLICATION BY FPE ANALYSER
START PROCESSING?
N
Y
THE ADS-B SERVER APPLICATION SEND EACH PACKET AT A TIME
THE ENCRYPTOR CARRIES OUT THE PROCESSING AND RETURNS THE RESULT TO THE FPE ANALYZER APPLICATION
FINISH?
N
Y

Fig. 7: Flowchart of system operation using the ADS-B Server application.



Host: 127.0.0.1 — Port: 31001

```
*8DE494B499083A94B068811CD313;
*8FE494B4EA4A584C013C085A0B58;
*8FE494B4E10892000000004DF060;
*8FE494B4580BB6A7D8BFF58E6E8D;
*8FE494B4580BE2903C8E5F58EC44;
*8FE494B4F82300060049B8E8F9BA;
*8FE494B4580BF290188E651B2096;
*8FE494B499083A95906C0202684B;
*8FE494B499083A95B07002EF9662;
*8FE494B49908399630740244BC68;
*8FE494B42305A574DF1DA038BAF3;
*8FE494B4580D728F3E8E87FF834F;
*8FE494B499083A97106C02BF9640;
*8FE494B499083A973074026A5E69;
```

*8DE4962EEA428858013C081F8AEB;

Code ICAO as separator: ○ Yes ● No

Amount msg: 2888 — Send delay (ms): 100

```
8DE48E099940339420B081E84199
8DE48E0999403394408C0249ED39
8DE48E09580B06A7D6BFE884B6E1
```

Fig. 8: Working screen of the ADS-B Server Software.

containing the ADS-B messages captured by the RTL1090 in conjunction with the FPE Analyzer Software. Using the “Transmit File” button, the program starts transferring messages, maintaining the time between messages by that selected in “Sending time”. The main *textbox* displays each transferred message. The user can also send individual messages using the “Send data” button. Finally, the program can create other files in the executable directory for each ICAO code; for a file with several ADS-B messages from different aircraft, the program can separate the ADS-B messages that appear in the main file. This functionality enables entropy analysis in messages with slight variations in information.

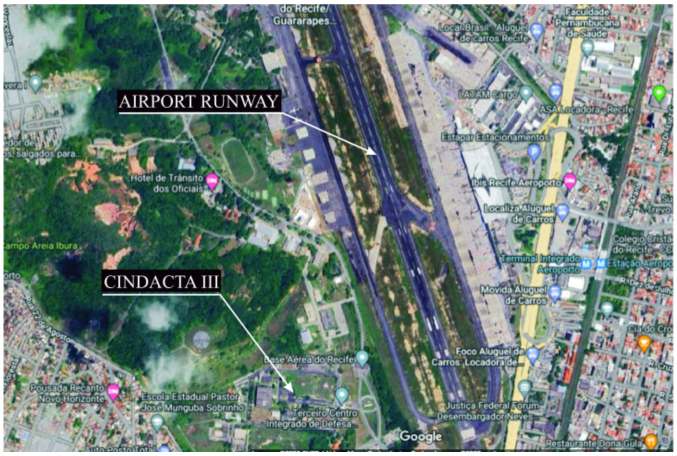


Fig. 9: Platform installation location.

IV. RESULTS

To analyze the behavior of the encryption algorithms, real data transmitted by transponders ADS-B of aircraft in the landing or takeoff phase at the international airport of Recife/Guararapes - Gilberto Freyre, based in Recife/PE, Brazil, were used. According to [41] and [42], the landing and takeoff phases are critical phases of the flight, so this research focused on obtaining data in the airport terminal area. The data acquisition and encryption platform was installed at the Third Integrated Center for Air Defense and Airspace Control (CINDACTA III), as shown in Figure 9, approximately 1 km from the airport, with direct vision.

The acquired data underwent two analyses: entropy and encryption performance. The first analysis aims to verify the level of uncertainty imposed on the information, while the second is related to the encryption/decryption processing time.

The FPE Analyzer and also the ENT program (available at https://www.fourmilab.ch/random/) [13] were used to carry out the entropy calculations. This tool displays the occurrence of each byte in the file and calculates entropy by expressing the result in bits per byte. The methodology for calculating entropy was built based on encryption without any value such as *TWEAK* and also using the 24-bit ICAO code in this field. These options allow us to analyze whether inserting information such as *TWEAK* would significantly change the entropy value. Four scenarios were used for obtaining data:

- Scenario 1: encryption of the complete ADS-B package, without using TWEAK in encryption;
- Scenario 2: encryption of the complete ADS-B package, using TWEAK in encryption;
- Scenario 3: encryption of the ADS-B package without inserting the header and recalculating the CRC, without using TWEAK in the encryption;
- Scenario 4: encryption of the ADS-B package without inserting the header and recalculating the CRC, using TWEAK in the encryption;

For each scenario, the following block ciphers were used as pseudorandom functions in FPE: AES, LEA, ASCON, and ASCON-PRF. The main difference between these scenarios is the amount of data encrypted in each message. When the complete message is encrypted, 112 bits, that is, 14 bytes of information, are used. However, when the header and CRC are not encrypted, this universe decreases to 80 bits or 10 bytes. In all tests, the following parameters were adopted as cryptographic key and TWEAK, and the value adopted for the cryptographic key was taken from the test vector provided by NIST [43]:

- Key: 2B7E151628AED2A6ABF7158809CF4F3C;
- TWEAK: Aircraft identification value in ICAO 24-bit format;

TABLE II: The 10 Aircraft that had the largest amount of ADS-B data collected.

CODE ICAO	AIRLINE	MODEL	REGISTER	QUANT. MESSAGES
E49A47	Voepass	ATR 72	PS-VPA	826
E498E8	Voepass	ATR 72	PR-PDX	724
E495B2	Azul	Airbus A320neo	PR-YSF	447
E49390	Azul	Airbus A320neo	PR-YYF	442
E4910B	Azul	ATR 72	PR-AQV	365
E492CF	Azul	Airbus A320neo	PR-YYJ	343
E495D0	Azul	Airbus A321neo	PR-YJC	340
E49165	Azul	ATR 72	PR-AQW	322
E4953A	Azul	Airbus A320neo	PR-YSB	313
E4917D	Azul	ATR 72	PR-AQZ	282

10,208 ADS-B messages transmitted by 55 different aircraft were acquired. For the research, data from all aircraft were considered in the first analysis, and data from the ten aircraft that had transmitted the most information were subsequently selected. This distinction between analyses is based on the need to evaluate entropy when there is a slight variation in information between ADS-B transmissions. In this way, by separating the data by ICAO code, it is possible to analyze information that kept the DF, CA, and ICAO fields unchanged; that is, 32 bits among the 112 remained fixed. Aircraft information is contained in Table II. Airline, model, and registration data were obtained through the website https://www.planespotters.net/ using aircraft identification in the ICAO 24-bit field.

Entropy calculations were performed for each block cipher per scenario based on the total data collected. Table III contains the entropy values for each scenario, considering the total number of messages observed. The higher entropy values in scenarios 1 and 2 are related to the complete encryption of the 112 bits of each ADS-B message packet, increasing the number of pseudorandom variables in the composition of the encrypted message. In scenarios 3 and 4, the amount of information to be encrypted reduces to 80 bits, as the header is not encrypted, and the CRC is replaced with a new, recalculated value. We observe that the behaviors regarding

TABLE III: Entropy concerning the total number of messages observed for each scenario.

CIPHER	SCEN 1	SCEN 2	SCEN 3	SCEN 4	QUANT. MESSAGES
AES	7,99547	7,99477	7,82683	7,82646	10208
LEA	7,99568	7,99527	**7,83040**	**7,83007**	10208
ASCON	7,99495	7,99556	7,82401	7,82278	10208
ASCON-PRF	**7,99578**	**7,99564**	7,83022	7,82563	10208
Clear Text	7,09332	7,09332	7,09332	7,09332	10208

TABLE IV: Entropy of each cipher for Scenario 1: Complete message encryption without TWEAK.

CIPHER	**ICAO 24 BITS IDENTIFICATOR**					
	E49A47	E498E8	E495B2	E49390	E4910B	E492CF
AES	**7,87399**	7,89808	7,91507	**7,86705**	7,94237	7,86830
LEA	7,86841	7,90093	**7,92048**	7,86356	**7,94728**	7,85114
ASCON	7,86178	7,90193	7,91007	7,85441	7,94030	7,85837
ASCON-PRF	7,86178	**7,90465**	7,91414	7,85528	7,94189	**7,87119**
Clear Text	6,52363	6,58931	6,62835	6,45793	6,51030	6,48920

CIPHER	**ICAO 24 BITS IDENTIFICATOR**				**AVERAGE**	**STAND. DESV.**
	E495D0	E49165	E4953A	E4917D		
AES	7,84878	7,91820	7,88165	7,93189	7,89454	0,02967
LEA	7,86109	7,92359	**7,88632**	7,92804	7,89508	0,03189
ASCON	7,85650	**7,92364**	7,88536	**7,93770**	7,89301	0,03256
ASCON-PRF	**7,88791**	7,92211	7,88170	7,92772	**7,89684**	0,02812
Clear Text	6,50881	6,53171	6,56181	6,52749	6,53285	0,04676

TABLE V: Entropy of each cipher for Scenario 2: Complete message encryption with TWEAK.

CIPHER	**ICAO 24 BITS IDENTIFICATOR**					
	E49A47	E498E8	E495B2	E49390	E4910B	E492CF
AES	7,84875	**7,90421**	7,91352	7,83931	7,93586	7,85844
LEA	7,86491	7,89626	**7,92865**	7,86015	**7,94167**	**7,87571**
ASCON	7,87030	7,89122	7,92085	7,85807	7,93601	7,87483
ASCON-PRF	**7,87151**	7,88848	7,91583	**7,87961**	7,93614	7,87090
Clear Text	6,52363	6,58931	6,62835	6,45793	6,51030	6,48920

CIPHER	**ICAO 24 BITS IDENTIFICATOR**				**AVERAGE**	**STAND**. **DESV**.
	E495D0	E49165	E4953A	E4917D		
AES	7,86873	7,92439	7,88602	7,92457	7,89038	0,03312
LEA	**7,87063**	7,92689	**7,88653**	7,92551	**7,89769**	0,02889
ASCON	7,86227	**7,93323**	7,88063	**7,92995**	7,89574	0,02950
ASCON-PRF	7,86770	7,92041	7,88360	7,92589	7,89601	0,02450
Clear Text	6,50881	6,53171	6,56181	6,52749	6,53285	0,04676

TABLE VI: Entropy of each cipher for Scenario 3: Message without encrypting the header and without using TWEAK.

CIPHER	**ICAO 24 BITS IDENTIFICATOR**					
	E49A47	E498E8	E495B2	E49390	E4910B	E492CF
AES	**7,66817**	**7,71160**	7,71572	7,63225	7,74287	7,64782
LEA	7,65325	7,69371	**7,73382**	**7,66018**	7,74916	7,63881
ASCON	7,63496	7,66505	7,71036	7,63600	7,74784	**7,65429**
ASCON-PRF	7,64432	7,69339	7,72965	7,62242	**7,75250**	7,63964
Clear Text	6,52363	6,58931	6,62835	6,45793	6,51030	6,48920

CIPHER	**ICAO 24 BITS IDENTIFICATOR**				**AVERAGE**	**STAND**. **DESV**.
	E495D0	E49165	E4953A	E4917D		
AES	7,66770	**7,73152**	7,69861	**7,72261**	**7,69389**	0,03567
LEA	7,64187	7,72084	**7,71169**	7,71368	7,69170	0,03816
ASCON	7,64490	7,73121	7,70562	7,71930	7,68495	0,04023
ASCON-PRF	**7,67081**	7,73086	7,69824	7,71279	7,68946	0,04160
Clear Text	6,50881	6,53171	6,56181	6,52749	6,53285	0,04676

TABLE VII: Entropy of each cipher for Scenario 4: Message without encrypting the header and using TWEAK.

CIPHER	**ICAO 24 BITS IDENTIFICATOR**					
	E49A47	E498E8	E495B2	E49390	E4910B	E492CF
AES	**7,64807**	7,67614	7,71355	7,62964	**7,75312**	**7,65534**
LEA	7,63809	**7,70510**	**7,72734**	7,60822	7,74624	7,64152
ASCON	7,63194	7,69511	7,71134	**7,65320**	7,74637	7,63397
ASCON-PRF	7,61927	7,68669	7,71340	7,64848	7,74425	7,64092
Clear Text	6,52363	6,58931	6,62835	6,45793	6,51030	6,48920

CIPHER	**ICAO 24 BITS IDENTIFICATOR**				**AVERAGE**	**STAND**. DESV.
	E495D0	E49165	E4953A	E4917D		
AES	7,66511	7,72089	7,69273	**7,73040**	**7,68850**	0,03811
LEA	7,65104	**7,73815**	**7,70266**	7,71726	7,68756	0,04604
ASCON	**7,66832**	7,72031	7,68471	7,72084	7,68661	0,03712
ASCON-PRF	7,65680	7,72356	7,67444	7,71785	7,68256	0,03916
Clear Text	6,50881	6,53171	6,56181	6,52749	6,53285	0,04676

the uncertainty level caused by the messages' encryption are very similar among the different tested ciphers. However, in absolute values, it is clear that for the encryption of 112 bits (Scenarios 1 and 2), the ASCON-PRF cipher presented the best results, while LEA was the best when 80 bits of the messages were encrypted (Scenarios 3 and 4). As the objective of the developed device is to serve as a cryptographic security element in existing *transponders*, there is a need to evaluate the level of entropy in messages that keep the aircraft identification code (24-bit ICAO code) unchanged.

The Tables IV, V, VI and VII display the results obtained for each scenario of the ten aircraft observed. It is interesting to note that different ciphers produced the best results when analyzed by aircraft. Despite the values being very close, the LEA cipher performed best in most aircraft for Scenarios 1 and 2, while the AES cipher obtained the best overall values for Scenarios 3 and 4.

A. Processing Time Performance Analysis

Table VIII contains the result of the processing time for each block cipher used as PRF in FF1 mode performed for all observed scenarios. The compiler used in the implementation was GCC (*GNU Compiler Collection*) with the -O2 optimization mode enabled. We can note that FPE encryption in FF1 mode with the 128-bit AES block cipher has the highest latency among all ciphers. This aspect can be attributed to the fact that AES uses several steps, including key expansion, replacement, permutation, and shuffling for each round, in addition to consuming more memory with S-Box. The LEA cipher, developed in South Korea, has a much simpler structure, designed for use in *lightweight* devices, presenting the best performance. Therefore, since the entropy results for all ciphers are very similar, the LEA cipher stands out as the best option, given its lower computational cost, which allows greater system scalability.

B. Comparison with Existing Methods

In this section, a comparison is made with other existing encryption methods for ADS-B data using FPE. [17] described the modes of operation for format-preserving encryption, which details the parameters for encryption with binary or decimal alphabet. This work references the solutions proposed later in applying FPE for ADS-B communication.

The work presented in [13] is the first related to applying FPE in ADS-B messages. The authors performed simulations with random data and simulated information with fixed bytes. They assigned information entropy analysis to define the security of the method. In [14], research based on the study by [13] is presented, and similar results are obtained. In this case, the authors used real ADS-B information collected from a CESSNA aircraft and suggested a way of key transmission via blockchain. The works presented applied FPE to the ADS-B communication protocol. They demonstrated the gain related to information obfuscation and the low impact on the current protocol standard, allowing the message format to be maintained. However, the block cipher of the pseudorandom

TABLE VIII: Average encryption latency (ms) for each cipher.

CIPHER	SCENARIO 1	SCENARIO 2	SCENARIO 3	SCENARIO 4
AES	16	16	16	16
LEA	1	1	1	1
ASCON	5	5	5	5
ASCON-PRF	3	3	3	3

TABLE IX: Entropy value of tests with fixed bytes in the input message, using FPE with AES-128 as block cipher in the pseudorandom function.

BYTES QUANT.	Ref. [13]		Ref. [14]		**Proposed**	
	Input	**Encrypted**	**Input**	**Encrypted**	**Input**	**Encrypted**
14 BYTES	7.94772295	7.9964494	6,7036	7,9985	7,10185314	7,99303908
13 BYTES	N/A	N/A	6,7036	7,8988	N/A	N/A
10 BYTES	6.9046341	7.9964559	N/A	N/A	7,10185314	7,82351399
8 BYTES	6.5156282	7.9964274	6,7036	7,714	N/A	N/A

function of these implementations was AES-128. To the best of our knowledge, no study besides this one applies ciphers other than AES-128 in the FPE algorithm for ADS-B communication or presents implementations in embedded systems for the FPE in the ADS-B scenario.

In order to observe the obtained gains, a comparison of clear text and encryption information from the previous works [13] and [14] was carried out concerning the data from this research. It can be seen from Table IX that the values found in this research are in the same range as previous studies, taking into account similar conditions, such as the use of AES-128 to obtain these data.

Although this research did not focus on the treatment of the cryptographic key, we suggest using the aircraft identification obtained in field 7 of the flight plan [44]. This field must be filled in by pilots or airlines. Flight plans are processed digitally through Air Traffic Control's data processing and visualization systems. According to the DECEA website[1], the SIGMA system can receive any flight plan over the internet. It will be processed through DECEA's computerized systems from the origin of the flight to the destination. In the proposed solution, the aircraft's avionics will use the aircraft identification in the TWEAK field to encrypt the ADS-B data. The ground systems will receive the record via the flight plan and can perform decryption together with the cryptographic key.

V. CONCLUSION

In this work, we developed a system for capturing and encrypting ADS-B data, installed near the Recife International Airport, allowed storing and evaluating the performance of the FPE algorithm in FF1 mode with the block ciphers AES, LEA, ASCON, and ASCON-PRF as pseudorandom functions. During the tests, four different scenarios were evaluated, and it was observed that all block ciphers allowed an increase in the entropy of ADS-B information transmitted by aircraft. Future work includes further study of feasible methods for exchanging secret keys that have the least possible impact on legacy infrastructure and the implementations of the proposed solution on reconfigurable hardware.

REFERENCES

[1] G. A. Gilbert, "Historical development of the air traffic control system," *IEEE Transactions on Communications*, vol. 21, no. 5, pp. 364–375, May 1973. doi: 10.1109/TCOM.1973.1091699

[2] M. A. Richards, J. A. Scheer, and W. A. Holm, *Principles of Modern Radar.* SciTech Publishing, Inc., 2010, vol. I. ISBN 978-1-891121-52-4

[3] B. G. Franciscone and P. A. L. Lima, "A consolidação da aviação civil internacional e suas implicações para a implementação do plano global de navegação aérea," *Revista Brasileira de Aviação Civil e Ciências Aeronáuticas*, vol. 01, no. 1, pp. 6–32, 2021, (in Portuguese). [Online]. Available: https://rbac.cia.emnuvens.com.br/revista/article/view/23

[4] R. M. Trim, "A brief history of the development of radar in great britain up to 1945," *Measurement + Control*, vol. 35, pp. 299–301, December 2002. [Online]. Available: https://nonstopsystems.com/radio/pdf-hell/article-MandC-Rdr-2020-12.pdf

[5] J. Sun, *The 1090 Megahertz Riddle. A Guide to Decoding Mode S and ADS-B Signals.* TU Delft OPEN Publishing, 2021. ISBN 978-94-6366-402-8

[6] V. A. Orlando, "The mode S beacon radar system," *Lincoln Laboratory Journal*, vol. 2, no. 3, pp. 345–362, 1989. [Online]. Available: https://www.ll.mit.edu/sites/default/files/publication/doc/mode-s-beacon-radar-system-orlando-ja-6373.pdf

[7] M. Leonardi, "ADS-B anomalies and intrusions detection by sensor clocks tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 5, pp. 2370–2381, 2019. doi: 10.1109/TAES.2018.2886616

[8] Z. Wu, T. Shang, and A. Guo, "Security issues in automatic dependent surveillance - broadcast (ADS-B): A survey," *IEEE Access*, vol. 8, pp. 122 147–122 167, July 2020. doi: 10.1109/ACCESS.2020.3007182

[9] E. Habler and A. Shabtai, "Analyzing sequences of airspace states to detect anomalous traffic conditions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 3, pp. 1843–1857, 2022. doi: 10.1109/TAES.2021.3124199

[10] M. Strohmeier, V. Lenders, and I. Martinovic, "On the security of the automatic dependent surveillance-broadcast protocol," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 1066–1087, 2015. doi: 10.1109/COMST.2014.2365951

[11] S. Khandker, H. Turtiainen, A. Costin, and T. Hämäläinen, "Cybersecurity attacks on software logic and error handling within ADS-B implementations: Systematic testing of resilience and countermeasures," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 4, pp. 2702–2719, 2022. doi: 10.1109/TAES.2021.3139559

[12] S. Amin, T. Clark, R. Offutt, and K. Serenko, "Design of a cyber security framework for ADS-B based surveillance systems," in *2014 Systems and Information Engineering Design Symposium (SIEDS)*, 2014. doi: 10.1109/SIEDS.2014.6829910 pp. 304–309.

[13] C. Finke, J. Butts, R. Mills, and M. Grimaila, "Enhancing the security of aircraft surveillance in the next generation air traffic control system," *International Journal of Critical Infrastructure Protection*, vol. 6, p. 3–11, 03 2013. doi: 10.1016/j.ijcip.2013.02.001

[14] J. Markani, A. Amrhar, J.-M. Gagné, and R. J. Landry, "Security establishment in ADS-B by format-preserving encryption and blockchain schemes," *Applied Sciences*, vol. 13, p. 3105, 02 2023. doi: 10.3390/app13053105

[15] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers, "Format-preserving encryption," in *Selected Areas in Cryptography*, M. J. Jacobson, V. Rijmen, and R. Safavi-Naini, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. ISBN 978-3-642-05445-7 pp. 295–312.

[16] W. J. Buchanan, S. Li, and R. Asif, "Lightweight cryptography methods," *Journal of Cyber Security Technology*, vol. 1, no. 3-4, pp. 187–201, 2017. doi: 10.1080/23742917.2017.1384917

[17] M. Bellare and P. Rogaway, "The FFX mode of operation for format-preserving encryption," 01 2010.

[18] D. Hong, J.-K. Lee, D.-C. Kim, D. Kwon, K. H. Ryu, and D.-G. Lee, "LEA: A 128-bit block cipher for fast encryption on common processors," in *Information Security Applications*, Y. Kim, H. Lee, and A. Perrig, Eds. Cham: Springer International Publishing, 2014. ISBN 978-3-319-05149-9 pp. 3–27. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-05149-9_1

[19] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer, "Ascon v1.2," Submission to Round 1 of the NIST Lightweight Cryptography project, 2019. [Online]. Available: https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/ascon-spec.pdf

[1] https://www.decea.mil.br/?i=midia-e-informacao&p=pg_noticia&materia=sigma-um-jeito-inovador-de-enviar-planos-de-voo-via-internet, accessed on July 3, 2024.

[20] M. S. Turan, K. McKay, D. Chang, J. Kang, N. Waller, J. M. Kelsey, L. E. Bassham, and D. Hong, "Status report on the final round of the NIST lightweight cryptography standardization process," 2023-06-16 04:06:00 2023. [Online]. Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=936814

[21] L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. F. Dray, and S. Vo, "SP 800-22 Rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications," Gaithersburg, MD, USA, Tech. Rep., 2010. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-22r1a

[22] Z. Wu, A. Guo, M. Yue, and L. Liu, "An ADS-B message authentication method based on certificateless short signature," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 3, pp. 1742–1753, June 2020. doi: 10.1109/TAES.2019.2933957

[23] M. Strohmeier, V. Lenders, and I. Martinovic, "Security of ADS-B: State of the art and beyond," *IEEE Communications Surveys e Tutorials*, vol. 17, 07 2013. doi: 10.1109/COMST.2014.2365951

[24] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers, "Format-preserving encryption," in *Selected Areas in Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. ISBN 978-3-642-05445-7 pp. 295–312. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-05445-7_19

[25] M. Dworkin, "Recommendation for block cipher modes of operation: Methods for format-preserving encryption," *Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD*, 2016-03-29 2016. doi: https://doi.org/10.6028/NIST.SP.800-38G

[26] M. Dworkin and R. Perlner, "Analysis of VAES3 (FF2)," Cryptology ePrint Archive, Paper 2015/306, 2015. [Online]. Available: https://eprint.iacr.org/2015/306

[27] F. B. Durak and S. Vaudenay, "Breaking the FF3 format-preserving encryption standard over small domains," in *Advances in Cryptology – CRYPTO 2017*, J. Katz and H. Shacham, Eds. Cham: Springer International Publishing, 2017. ISBN 978-3-319-63715-0 pp. 679–707.

[28] NIST, "Recommendation for block cipher modes of operation: Methods for format-preserving encryption," https://csrc.nist.gov/publications/detail/sp/800-38g/rev-1/draft, 2019, acesso em: 01 de mar. de 2023.

[29] C. Paar and J. Pelzl, *Understanding Cryptography - A Textbook for Students and Practitioners.* Springer, 2010. ISBN 978-3-642-04100-6

[30] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "Cryptographic sponges functions," *online] http://sponge. noekeon. org*, 2011. [Online]. Available: https://keccak.team/files/CSF-0.1.pdf

[31] M. Dworkin, E. Barker, J. Nechvatal, J. Foti, L. Bassham, E. Roback, and J. Dray, "Advanced encryption standard (AES)," 2001-11-26 2001.

[32] R. Agbeyibor, J. Butts, M. Grimaila, and R. Mills, "Evaluation of format-preserving encryption algorithms for critical infrastructure protection," in *IFIP Advances in Information and Communication Technology*, vol. 441, 03 2014. doi: 10.1007/978-3-662-45355-1_16. ISBN 978-3-319-12567-1

[33] NXP, *LPC1769/68/67/66/65/64/63 Product data sheet*, NXP Semiconductors, 04 2011, rev.7. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/LPC1769_68_67_66_65_64_63.pdf

[34] H. Seo, Z. Liu, J. Choi, T. Park, and H. Kim, "Compact implementations of LEA block cipher for low-end microprocessors," in *Information Security Applications*, vol. 9503, 01 2016. doi: 10.1007/978-3-319-31875-2_3. ISBN 978-3-319-31874-5 pp. 28–40.

[35] S. Jahnavi, C. B. Pooja, S. Leeona, and R. Vijayageetha, "Implementation of wide band FM receiver on RTL-SDR," *International Journal of Engineering Research and*, vol. V5, 05 2016. doi: 10.17577/IJERTV5IS050707

[36] Realtek, *RTL2832U data sheet*, Realtek Semiconductor Corp., 11 2010, rev.1.4. [Online]. Available: https://homepages.uni-regensburg.de/~erc24492/SDR/Data_rtl2832u.pdf

[37] H. Mohamed, P. Lazaridis, D. Upton, U. Khan, B. Saeed, A. Jaber, Y. Zhang, P. Mather, M. Queiroz Vieira, K. Barlee, D. Atkinson, A. Mihovska, L. Gavrilovska, and I. Glover, "Partial discharge detection using low cost RTL-SDR model for wideband spectrum sensing," in *2016 23rd International Conference on Telecommunications (ICT)*, 05 2016. doi: 10.1109/ICT.2016.7500353

[38] JETVISION, "RTL1090 software for ADS-B dongles," https://rtl1090.com/, 2023, acesso em: 08 de maio de 2023.

[39] H. V. Kumar, S. G, N. V, S. M, and S. Kumar H, "Tracking of aircrafts using software defined radio (SDR) with an antenna," *International Journal of Scientific Research in Science and Technology*, pp. 660–665, 06 2021. doi: 10.32628/IJSRST2183148

[40] Silicon, *CP2102 data sheet*, Silicon Labs, 01 2017, rev.1.8. [Online]. Available: https://www.silabs.com/documents/public/data-sheets/CP2102-9.pdf

[41] ANAC, "RBAC 135: requisitos operacionais: operações complementares e por demanda," Agência Nacional de Aviação Civil, 2010, (in Portuguese). [Online]. Available: https://www.anac.gov.br/assuntos/legislacao/legislacao-1/boletim-de-pessoal/2010/33s1/rbac-135

[42] ——, "IS 91-001: aprovação de aeronave e operadores para condução de operações PBN," Agência Nacional de Aviação Civil, 2023, (in Portuguese). [Online]. Available: https://www.anac.gov.br/assuntos/legislacao/legislacao-1/iac-e-is/is/is-91-001

[43] C. S. R. C. NIST, "Examples with intermediate values," https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/example-values, 2023, acesso em: 08 de jun. 2023.

[44] DECEA, "MCA 100-11: Preenchimento dos formulários de plano de voo," Departamento de Controle de Espaço Aéreo, 2020, (in Portuguese). [Online]. Available: https://publicacoes.decea.mil.br/publicacao/MCA-100-11

Celdo Souza da Silveira was born in Brazil where he obtained a Bachelor's degree in Electrical Engineering with an emphasis on Industrial Control and Automation from the University of Vale do Rio dos Sinos (UNISINOS) in 2012, and an M.Sc. in Electrical Engineering with a concentration in Communications from the Federal University of Pernambuco (UFPE) in 2023. He has been a member of the Brazilian Air Force since 2001 and worked in radar data processing, data treatment and visualization systems, and datalink technologies from 2013 to 2023 at the Third Integrated Center for Air Defense and Airspace Control (CINDACTA III). He currently works as a Secondee at the International Civil Aviation Organization (ICAO) in Montreal, Canada. His research focuses on surveillance and communication in aviation, as well as additive manufacturing technologies.

Juliano Bandeira Lima was born in Brazil where he studied electrical engineering. He received the M.Sc. and Ph.D. degrees in electrical engineering from Federal University of Pernambuco (UFPE), Brazil, in 2004 and 2008, respectively. Since 2015, he has been a research productivity fellow awarded by the Conselho Nacional de Desenvolvimento Científico e Tecnológico. He is currently an Associate Professor of the Department of Electronics and Systems at UFPE. His main research interests are in the field of discrete and number-theoretic transforms, and their applications in digital signal processing, communications and cryptography.

José Rodrigues de Oliveira Neto was born in Brazil. He received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Federal University of Pernambuco (UFPE), Brazil, in 2013, 2015, and 2019, respectively. He is currently an Assistant Professor with the Department of Mechanical Engineering, UFPE. His main research interests include digital signal processing, embedded systems, and hardware implementations.