UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

CELDO SOUZA DA SILVEIRA

SEGURANÇA NA COMUNICAÇÃO ADS-B:

Avaliação de Cifras de Bloco *Lightweight* num Ambiente de Preservação de Formato

Recife
2023

CELDO SOUZA DA SILVEIRA

SEGURANÇA NA COMUNICAÇÃO ADS-B:

Avaliação de Cifras de Bloco *Lightweight* num Ambiente de Preservação de Formato

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração: Comunicações.

Orientador: Prof. Dr. José Rodrigues de Oliveira Neto.
Coorientador: Prof. Dr. Juliano Bandeira Lima.

Recife
2023

Catalogação na fonte:
Bibliotecária Sandra Maria Neri Santiago, CRB-4 / 1267

S587s Silveira, Celdo Souza da.
Segurança na comunicação ADS-B: avaliação de cifras de bloco *lightweight* num ambiente de preservação de formato / Celdo Souza da Silveira. – 2023.
87 f.: il., fig., tab., abrev. e siglas.

Orientador: Prof. Dr. José Rodrigues de Oliveira Neto.
Coorientador: Prof. Dr. Juliano Bandeira Lima.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Elétrica. Recife, 2023.
Inclui referências e apêndice.

1. Engenharia elétrica. 2. ADS-B. 3. FPE. 4. Cifragem. 5. *Lightweight*. 6. Entropia. I. Oliveira Neto, José Rodrigues de (Orientador). II. Lima, Juliano Bandeira (Coorientador). III. Título.

UFPE

621.3 CDD (22. ed.) BCTG/2023-219

CELDO SOUZA DA SILVEIRA

SEGURANÇA NA COMUNICAÇÃO ADS-B:
Avaliação de Cifras de Bloco *Lightweight* num Ambiente de Preservação de Formato

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

Área de Concentração: Comunicações.

Aprovado em: 28 / 07 / 2023 .

Prof. Dr. José Rodrigues de Oliveira Neto. (Orientador)
Universidade Federal de Pernambuco

Prof. Dr. Juliano Bandeira Lima.(Coorientador)
Universidade Federal de Pernambuco

Prof. Dr. Daniel Pedro Bezerra Chaves (Examinador Interno)
Universidade Federal de Pernambuco

Prof. Dr. Vitor de Andrade Coutinho - UFRPE (Examinador Externo)
Universidade de Pernambuco

Dedico esse trabalho com todo amor e carinho para a minha querida esposa Ana Cristina, a qual sempre esteve ao meu lado proporcionando força e determinação, além de ser pilar fundamental de nossa família.

AGRADECIMENTOS

Pude contar, ao longo desse trabalho, com o auxílio de várias pessoas que fizeram ser possível o sucesso desse estudo.

A essas pessoas dedico os mais sinceros agradecimentos:

Aos meus pais Celdo Soares da Silveira *(in-memoriam)* e Nilda Gessi Souza da Silveira, por proporcionarem toda a formação necessária para que me tornasse um cidadão capaz e digno, por todo o amor e ternura e por sempre estarem ao meu lado em todos os momentos da minha vida, sendo os pilares de minha existência;

A minha esposa Ana Cristina, companheira de todos os momentos, sempre presente dando seu amor e proporcionando força e motivação;

Aos meus amados filhos Celdo, Isaac e Giovanna por serem minhas fontes de energia e motivo de viver, dando amor e carinho;

A minha irmã Joicy Souza da Silveira pelo apoio incondicional em todas as horas, mesmo estando distante fisicamente, mas sempre presente em minha vida;

Ao Prof. Dr. José Rodrigues de Oliveira Neto, orientador desse trabalho, pela amizade, orientação e paciência;

Ao Prof. Dr. Juliano Bandeira Lima, co-orientador, pela amizade, ensinamentos e apoio;

Ao meu amigo Tenente Coronel Marcos Aurélio dos Santos, pela confiança depositada e pelo apoio incondicional durante o período em que trabalhamos juntos.

RESUMO

Neste trabalho, apresenta-se um estudo referente às vulnerabilidades da comunicação ADS-B (*Automatic Dependent Surveillance – Broadcast*), considerando sobretudo, os riscos à segurança operacional. Essa tecnologia foi desenvolvida para transmitir informações de vigilância, ou seja, dados relativos a posição, altimetria e velocidade, a partir dos aviônicos da própria aeronave, sendo uma complementação aos radares. Nesse trabalho, propõe-se uma solução de criptografia para essa comunicação com a utilização da cifragem com preservação de formato, avaliando-se também o emprego de cifras de bloco simétricas com o objetivo de obter melhor desempenho em soluções aplicadas em ambiente *lightweight*, ou seja, aplicação em sistemas com baixo consumo energético, dimensões reduzidas e recurso computacional limitado. Observa-se que tais cifras, quando utilizadas como função pseudoaleatória, mantêm elevado valor de entropia com baixo custo computacional atendendo aos níveis de segurança desejados. Além disso, esse estudo com pesquisas semelhantes com o objetivo de confrontar as diferentes abordagens e complementar os estudos. Por fim, analisa-se o desempenho computacional obtido com a solução proposta processando dados reais de aeronaves em fase de pouso e decolagem do aeroporto internacional do Recife/Guararapes - Gilberto Freyre, sediado em Recife/PE.

Palavras-chave: ADS-B; FPE; cifragem; *lightweight*; entropia.

ABSTRACT

In this work, a study is presented regarding the vulnerabilities of ADS-B communication (*Automatic Dependent Surveillance – Broadcast*), considering, above all, the risks to operational security. This technology was developed to transmit surveillance information, that is, data related to position, altimetry and speed, from the avionics of the aircraft itself, complementing the radars. In this work, an encryption solution is proposed for this communication using format-preserving encryption, also evaluating the use of symmetric block ciphers in order to obtain better performance in solutions applied in a *lightweight environment* , that is, application in systems with low energy consumption, reduced dimensions and limited computational resources. It is observed that such ciphers, when used as a pseudorandom function, maintain a high entropy value with low computational cost, meeting the desired security levels. In addition, this study with similar research in order to confront the different approaches and complement the studies. Finally, the computational performance obtained with the proposed solution is analyzed by processing real data from aircraft in the landing and takeoff phase at the international airport of Recife/Guararapes - Gilberto Freyre, based in Recife/PE.

Keywords: ADS-B; FPE; encryption; lightweight; entropy.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACAS	Airborne Collision Avoidance System
ADS-B	Automatic Dependent Surveillance – Broadcast
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
ANAC	Agência Nacional de Aviação Civil
ARSR	Air Route Surveillance Radar
ARX	Addition, Rotation, XOR
ASCII	American Standard Code for Information Interchange
ASR	Airport Surveillance Radar
ATC	Air Traffic Control
ATCRBS	Air Traffic Control Beacon System
CAESAR	Competition for Authenticated Encryption: Security, Applicability, and Robustness
CAN	Controller Area Network
CBC	Cipher-Block Chaining
CDC	Communication Device Class
CRC	Cyclic Redundancy Check
DECEA	Departamento de Controle do Espaço Aéreo
DES	Data Encryption Standard
DF	Downlink Format
DMA	Direct Memory Access
DPSK	Differential Phase-Shift Keying
DVB-T	Digital Video Broadcast-Terrestrial
FF1	Format-preserving, Feistel-based encryption mode 1
FF2	Format-preserving, Feistel-based encryption mode 2
FF3	Format-preserving, Feistel-based encryption mode 3
FFX	Format-preserving, Feistel-based encryption
FIPS	Federal Information Processing Standards
FPE	Format Preserving Encryption
FPGA	Field-Programmable Gate Array
FRUIT	False Replies Unsynchronized with Interrogator Transmissions
GCC	GNU Compiler Collection
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
ICAO	International Civil Aviation Organization
IoT	Internet of Things
ISP	In-System Programming

LEA	Lightweight Encryption Algorithm
LNA	Low-Noise Amplifier
MAC	Message Authentication Code
MN	Milhas Náuticas
NEXGEN	Next Generation Air Transportation System
NIST	National Institute of Standards and Technology
NSA	National Security Agency
PPM	Pulse Position Modulation
PRF	Pseudorandom Function
RADAR	Radio Detection and Ranging
RPM	Rotações Por Minuto
RTL	Realtek
SDR	Software-Defined Radio
SESAR	Single European Sky ATM Research
SIGMA	Sistema Integrado de Gestão de Movimentos Aéreos
SISCEAB	Sistema de Controle do Espaço Aéreo Brasileiro
SSR	Secondary Surveillance Radar
TCAS	Traffic Collision Avoidance System
TDOA	Time Difference Of Arrival
UART	Universal Asynchronous Receiver/Transmitter
UF	Uplink Format
USB	Universal Serial Bus
XOR	Exclusive OR

LISTA DE SÍMBOLOS

$\oplus$	Função OU Exclusivo
$\boxplus$	Soma modular
$\|\|$	Concatenação
$\lceil\ \rceil$	Função Teto
$\lfloor\ \rfloor$	Função Piso

SUMÁRIO

1 INTRODUÇÃO

O transporte aéreo tem papel fundamental na sociedade, sendo resultado direto da evolução tecnológica e contribuindo significantemente para o desenvolvimento econômico. Anualmente, a quantidade de pessoas transportadas por meio da aviação tem crescido, motivado por uma sociedade globalizada, em que as transações intercontinentais são cada vez mais necessárias. Por ser um meio de transporte flexível, hoje as empresas nacionais e multinacionais se tornam dependentes dele (GUIMARÃES et al., 2019; FRANCISCONE; LIMA, 2021).

Esse papel relevante da aviação na vida cotidiana da sociedade só foi possível com a evolução da tecnologia e, principalmente, com a segurança operacional. No passado, em relação à navegação, os pilotos utilizavam referências em solo para navegação e balizas luminosas nas pistas para que fosse possível o pouso. No campo das comunicações, o rádio era o meio de comunicação com os pilotos (GILBERT, 1973). Posteriormente, com o advento do RADAR (*Radio Detection and Ranging*) na década de 30, tecnologia cujo uso foi iniciado pelo Reino Unido na segunda guerra mundial, foi possível detectar o posicionamento e velocidade das aeronaves em pleno voo através da reflexão de ondas eletromagnéticas, (RICHARDS; SCHEER; HOLM, 2010), permitindo aumento considerável na segurança operacional, principalmente em condições atmosféricas não favoráveis (FRANCISCONE; LIMA, 2021).

Com o aumento da quantidade de companhias aéreas e, consequentemente, de linhas aéreas, foi necessário aumentar a capacidade do espaço aéreo mantendo a segurança operacional. Dessa forma, tornou-se fundamental que o controle de tráfego aéreo possuísse equipamentos com elevado nível de precisão e confiabilidade. Nesse tocante, os radares foram primordiais para a evolução do tráfego aéreo, pois permitem obter informações precisas de posição, velocidade e identificação das aeronaves sob controle terrestre (TRIM, 2002). Essas informações são chamadas de dados de vigilância.

Os radares secundários, desenvolvidos posteriormente, permitiram obter uma quantidade maior de informações das aeronaves, pois efetuavam a transmissão de um código que servia como interrogação para as aeronaves com a solicitação de dados, tais como código de identificação e altimetria. Com a evolução da tecnologia, os radares secundários permitiram receber mais informações das aeronaves, bem como adquiriram a possibilidade de interrogar individualmente cada avião. Essa modo de operação do radar secundário foi chamado de modo S, onde o S está relacionado a “seletivo” (SUN, 2021; ORLANDO, 1989).

Mesmo com o avanço da tecnologia radar, ainda havia necessidade de obtenção de mais dados durante o voo em intervalos menores de tempo. Com isso, foi desenvolvida uma nova forma de transmitir as informações de vigilância para o controle de tráfego, além de permitir que outras aeronaves, sob o mesmo espaço aéreo, pudessem compartilhar essas mesmas informações. Esse protocolo foi denominado ADS-B (*Automatic Dependent Surveillance – Broadcast*) e é uma evolução do modo S. Com ele, as aeronaves transmitem diversas informações fundamentais para o controle do espaço aéreo, tais como: posição, velocidade e identificação. Além desses

dados, pode fornecer informações de código de chamada, dados operacionais, indicadores de precisão e indicadores e integridade. Esse tipo de comunicação é transmitido como *broadcast*, sem depender de solicitação dos sistemas de solo (SUN, 2021).

1.1 JUSTIFICATIVA

A comunicação ADS-B possui muitas vantagens operacionais, dentre elas pode-se destacar o intervalo entre envio de mensagens de posição, que é de aproximadamente 0,5 segundos, enquanto que o intervalo de atualização de um radar secundário de terminal é de 4 segundos (SUN, 2021). No entanto, essa comunicação possui grande vulnerabilidade relacionada a segurança da informação. As mensagens são transmitidas por canais de rádio abertos, ou seja, sem criptografia e não adotam nenhuma medida de segurança para proteger a transmissão de dados. Um invasor pode efetuar diversos tipos de ataques contra as informações enviadas, tais como espionagem, bloqueio e modificação de mensagens (WU; SHANG; GUO, 2020). A importância e vulnerabilidade das informações de status operacional da aeronave, tais como mensagens de posição, identificação, velocidade, dentre outros, as tornam o principal alvo de invasores mal-intencionados, em um contexto onde o principal fator está relacionado a criticidade, pois um ataque nesse tipo de comunicação coloca em risco a segurança operacional. Atualmente, os tipos de ataques que existem para o sistema ADS-B são divididos, conforme (STROHMEIER; LENDERS; MARTINOVIC, 2014), em:

- *Espionagem*;
- *Interferência*;
- *Injeção de mensagem*;
- *Exclusão de mensagem*;
- *Modificação de mensagem*;

Dessa forma, a cifragem com preservação de formato, conhecido como FPE (*format preserving encryption*), a qual consiste na técnica de cifrar o texto claro, sob controle de uma chave simétrica, sendo que o texto cifrado mantém o mesmo formato do texto original. Esse princípio de manter o formato dos pacotes de dados é fundamental na aviação, visto que a alteração do protocolo já definido e em uso teria enorme impacto nos equipamentos de bordo e nas estações de solo. Dessa forma, a solução mostra-se como uma possível alternativa criptográfica para a comunicação ADS-B (BELLARE et al., 2009).

A FPE é extremamente versátil e proporciona elevado nível de segurança, desde que sua função pseudoaleatória seja composta por cifras que possuam elevada resistência a ataques. A versatilidade está diretamente relacionada a possibilidade de aplicação em um número ilimitado de formatos. Nesse sentido, cifras de bloco desenvolvidas para aplicações *lightweight*, ou seja,

para dispositivos com baixo poder computacional e baixo consumo energético, podem ser uma excelente escolha, uma vez que foram dimensionados para realizar a cifragem de pacotes de dados com número reduzido de operações por ciclo sem comprometer a segurança (BUCHANAN; LI; ASIF, 2017).

1.2 OBJETIVOS

Objetivo geral:

O objetivo da pesquisa é avaliar a viabilidade do uso da cifragem com preservação de formato para o protocolo ADS-B, implementando um sistema de cifragem dos dados desse protocolo em dispositivos embarcados de baixo custo e avaliar o ganho de desempenho e segurança com a utilização de cifras de bloco atuais, desenvolvidas para aplicações *lightweight*.

Objetivos específicos:

Os objetivos específicos desta proposta são:

1. Estudar a bibliografia necessária para o conhecimento das questões de segurança da informação relacionada a comunicação ADS-B;
2. Destacar as principais vulnerabilidades da comunicação ADS-B;
3. Propor uma solução criptográfica para essa comunicação com vistas a manter o formato atual dos pacotes de dados;
4. Avaliar o ganho de segurança e desempenho com diferentes cifras de bloco como função pseudoaleatória;
5. Propor hardware e software para operar como elemento criptográfico e possibilitar a avaliação de desempenho.

1.3 METODOLOGIA

Para alcançar os objetivos mencionados na Seção 1.2, o trabalho foi estruturado com base nas etapas mencionadas a seguir.

- ETAPA 1- *Revisão da literatura:* realizar a revisão da literatura relacionada aos apontamentos, análises e discussões sobre as vulnerabilidades da comunicação ADS-B, como elencado no estudo de (WU; SHANG; GUO, 2020), o qual aborda as características de cada ponto fraco e suas implicações. Para evitar possíveis ataques criptográficos, será tratado o estudo de (FINKE et al., 2013), sobre a aplicação da cifragem com preservação de formato aplicada à comunicação ADS-B, assim como realizar a revisão com detalhes do formato desse algoritmo como descrito por (BELLARE; ROGAWAY, 2010). Por fim,

realizar a revisão da literatura relacionada as cifras de bloco para cifragem de informação aplicadas a dispositivos embarcados, denominadas de *lightweight*.

- ETAPA 2- *Proposição:* na fase inicial do projeto, avaliar as principais vulnerabilidades da comunicação ADS-B, seu impacto para a segurança operacional no contexto do controle de tráfego aéreo e propor solução de cifragem que permita evitar ataques criptográficos.
- ETAPA 3 - *Avaliação:* nesta etapa, observar a possível solução criptográfica que pode mitigar as vulnerabilidades apresentadas mantendo a compatibilidade no formato das mensagens ADS-B, a fim de não alterar significativamente os equipamentos de bordo e dos sistemas de solo. Como critério de avaliação, considerar a análise de entropia de Shannon aos dados cifrados e comparar com estudos anteriores.
- ETAPA 4 - *Concepção:* a execução desta etapa está concentrada no desenvolvimento de hardware e software que permita a avaliação de segurança para a solução criptográfica apresentada, assim como analisar o desempenho computacional da ferramenta através do tempo de latência.
- ETAPA 5 - *Emprego:* com base nas observações realizadas, após os testes de avaliação, chegar a uma conclusão sobre a eficiência do algoritmo e das cifras de blocos apresentadas.

1.4 ESTRUTURA DO DOCUMENTO E CONTRIBUIÇÕES

O documento está estruturado da seguinte forma:

- No Capítulo 1, é introduzida a temática do estudo, a justificativa e os objetivos gerais e específicos pretendidos para a realização desse trabalho.
- No Capítulo 2, é realizada revisão na literatura sobre a evolução do controle de tráfego aéreo, incluindo o desenvolvimento tecnológico dos equipamentos de detecção e vigilância do espaço aéreo.
- No Capítulo 3, são debatidas as vulnerabilidades da comunicação ADS-B e pontuadas as possíveis soluções criptográficas para mitigação;
- No Capítulo 4, é detalhado o funcionamento da cifragem com preservação de formato e suas vantagens na aplicação de sistemas legados;
- No Capítulo 5, é proposta uma solução criptográfica com quatro cifras de blocos distintas para avaliação de segurança e desempenho através do desenvolvimento de hardware e software;
- No Capítulo 6, são revisitadas de forma resumida as principais contribuições do trabalho, elencados possíveis desdobramentos desta dissertação em trabalhos futuros relacionadas às investigações realizadas;

- No Apêndice A, está ilustrado o esquema elétrico do circuito desenvolvido para avaliar as soluções de criptografia aplicada à comunicação ADS-B.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo tem o objetivo de contextualizar o cenário relativo ao controle de tráfego aéreo com base no passado, presente e a expectativa de futuro, com vistas à segurança e eficiência operacional, demonstrando a importância do tema desse trabalho para solucionar as questões de vulnerabilidade apresentadas. O foco principal é na tecnologia ADS-B como tecnologia emergente nos principais programas de modernização no tráfego aéreo, tais como: SIRIUS (DECEA, 2022), NEXTGEN (FAA, 2021) e SESAR (SESAR, 2022).

2.1 HISTÓRIA DO TRÁFEGO AÉREO

A aviação utilizava regras semelhantes às utilizadas no tráfego terrestre até meados de 1910, sendo que cada país possuía as suas próprias regras. No entanto, havia um consenso na Europa que deveria existir um mínimo de padronização. Após a primeira guerra mundial, na Conferência de Paz de Versailles em 1919, foi acordada a criação da Convenção Internacional de Navegação Aérea com o objetivo de desenvolver, entre outros regulamentos, as normas gerais para o tráfego aéreo (GILBERT, 1973).

Nesse período, os pilotos utilizavam referências em solo para navegação, além de balizas luminosas no entorno de pistas para pouso. O rádio, como meio de comunicação com o piloto, só foi possível entre 1930 e 1935. O aumento constante da aviação exigiu que técnicas e equipamentos fossem desenvolvidos para melhoria da segurança, principalmente em situações adversas como as meteorológicas (GILBERT, 1973).

2.2 HISTÓRIA DO RADAR

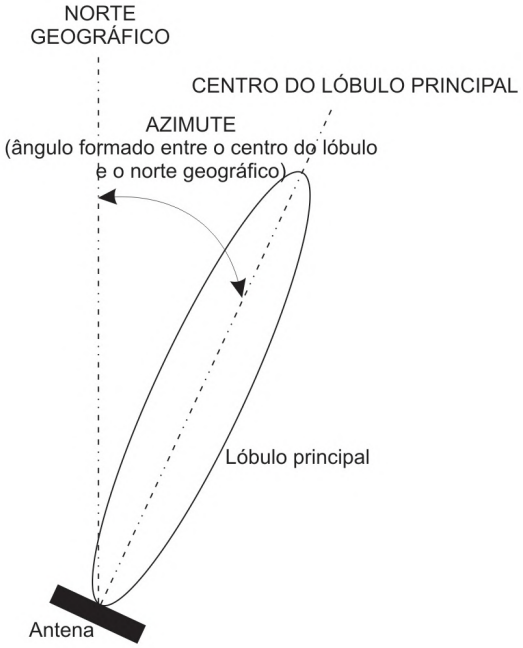
O RADAR, acrônimo que significa detecção e alcance por rádio, é um equipamento destinado à aquisição de informações de distância e azimute para aeronaves detectadas através de ondas eletromagnéticas. Ele foi desenvolvido praticamente de forma simultânea pelo Reino Unido, EUA, Japão, Alemanha, França, Itália, Holanda e União Soviética, de maneira isolada entre esses países e com enorme segredo durante os anos de 1930 (TRIM, 2002).

O conceito de azimute está relacionado ao ângulo formado entre a posição da aeronave detectada e o ponto de início da contagem de varredura da antena rotativa, chamado de norte. Geralmente, o ponto de início da contagem de azimute está relacionado ao norte geográfico, conforme ilustra a Figura 1.

Os radares modernos são sofisticados sistemas que não apenas detectam alvos e determinam o alcance do alvo, mas também podem rastrear, identificar, criar imagens e classificar alvos, além de poder suprimir forte interferência indesejada, como ecos do ambiente e contramedidas (RICHARDS; SCHEER; HOLM, 2010).

O Reino Unido foi a nação que apontou na liderança do desenvolvimento do RADAR, tendo instalado uma cadeia de radares para alerta aéreo visando à defesa aérea em 1939, durante

Figura 1 – Azimute em relação ao norte geográfico.



Fonte: O Autor (2023).

a segunda guerra mundial. Era composta por antenas fixas que transmitiam na faixa de frequência entre 20MHz e 55MHz e possuíam limitação para detecção em baixa altitude. Com a evolução da guerra, os radares tiveram seu desenvolvimento acelerado chegando a transmissores na faixa dos 200MHz, ou seja, maior precisão em distância, uma vez que a largura de banda era maior (TRIM, 2002).

Após a segunda guerra mundial, os radares foram utilizados no gerenciamento do tráfego aéreo, servindo como um divisor entre a primeira e a segunda geração de tecnologias para a aviação na década de 50 (GILBERT, 1973). Atualmente, esses equipamentos são largamente utilizados no controle do espaço aéreo mundialmente permitindo separações (distância entre aeronaves) na faixa de 3 a 5 MN (milhas náuticas) e uma precisão de posição na ordem de 2 MN (WESSON; HUMPHREYS; EVANS, 2014), onde 1 MN corresponde a 1852 metros. Há basicamente dois tipos de radares: primário e secundário; descritos nas seções subsequentes.

2.3 RADAR PRIMÁRIO

O radar primário utiliza a reflexão das ondas eletromagnéticas propagadas no espaço pela superfície metálica das aeronaves. A informação de distância é obtida com base no tempo entre a emissão e recepção do sinal transmitido, desde que conhecida a velocidade de propagação da onda. A Figura 2 exibe a densidade de potência que retorna ao radar como efeito da reflexão das ondas eletromagnéticas. A equação radar, $Q_r = Q_i\sigma/4\pi R^2 = P_t G_t \sigma/(4\pi)^2 R^4$ associada à figura, expõe os termos que influenciam na detecção, onde:

Q_r é a densidade de potência que retornou do alvo;

Q_i é a densidade de potência transmitida;

P_t é a potência de transmissão;

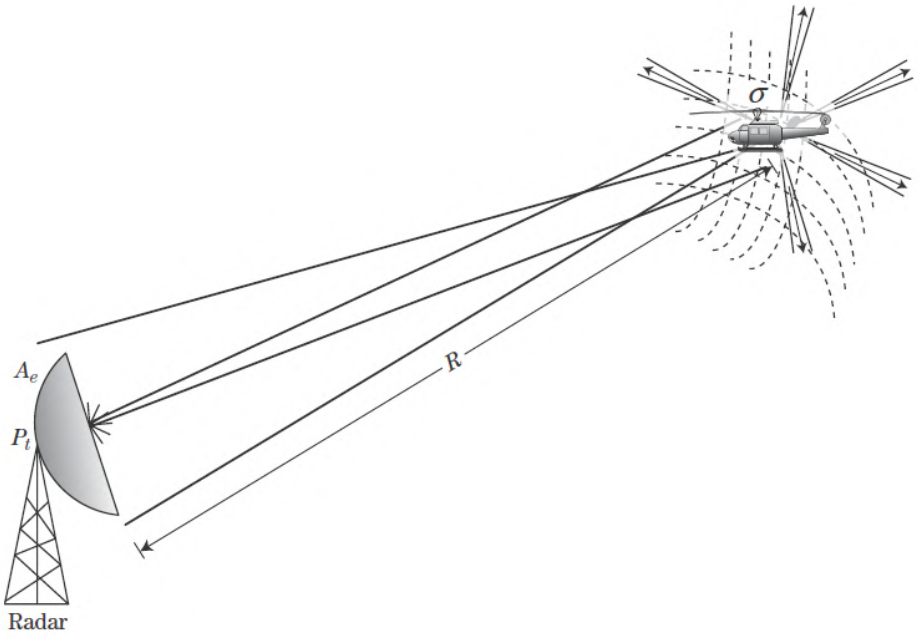
G_t é o ganho da antena;

R é a distância entre a antena e o alvo;

σ é a RCS (*radar cross section*) ou seção reta radar, a qual é determinada pelo tamanho físico, formato e material do alvo (RICHARDS; SCHEER; HOLM, 2010).

O azimute, ou seja o ângulo de detecção em relação ao norte do equipamento, é obtido com base na rotação da antena, conforme exibido anteriormente na Figura 1.

Figura 2 – Densidade de potência recebida pelo radar, após reflexão pelo alvo.



Fonte: (RICHARDS; SCHEER; HOLM, 2010).

Os radares primários são divididos em ASR (*Airport Surveillance Radar*) e ARSR (*Air Route Surveillance Radar*). O primeiro está relacionado à necessidade de detecção de aeronaves próximas às pistas e, devido a isso, a velocidade de rotação da antena tem que ser maior, permitindo uma atualização mais rápida para o controlador de tráfego aéreo. Esses radares operam com uma velocidade de rotação por volta de 12 RPM e uma frequência de portadora em banda S. A Tabela 1 exibe as bandas de frequência dos serviços móveis. Os radares ARSR são dimensionados para detectar aeronaves em rota com uma distância de detecção muito maior do que os radares ASR, chegando a 200 MN contra 50 MN dos radares de terminal. A frequência de operação está na banda L (950 a 2150 MHz) e a velocidade de rotação da antena fica em torno de 6 RPM (TRIM, 2002). Esses radares são muito utilizados em defesa aérea, visto que não dependem de qualquer sinal a ser transmitido pela aeronave de forma colaborativa.

Tabela 1 – Largura de banda do espectro de frequências.

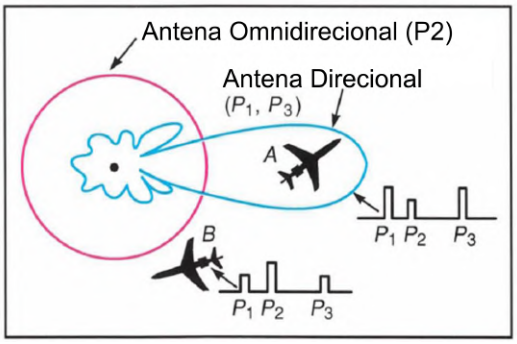
BANDA DE FREQUÊNCIAS	
Banda	*Faixa de Frequência*
L	1 a 2 GHz
S	2 a 4 GHz
C	4 a 6 GHz
Ku	11 a 14 GHz
Ka	20 a 30 GHz

Fonte: O Autor (2023).

2.4 RADAR SECUNDÁRIO

O radar secundário foi uma evolução do radar primário, quando se trata de controle de tráfego aéreo. Originou-se na década de 1960 e foi batizado de ATCRBS (*Air Traffic Control Beacon System*) ou sistema de baliza para o controle de tráfego aéreo, mas também é conhecido mundialmente como SSR (*Secundary Surveillance Radar*). Esse tipo de radar possui uma antena rotativa, assim como o primário, mas a diferença reside no fato que ele não depende da reflexão do sinal transmitido, mas sim de uma resposta emitida pela aeronave. O equipamento a bordo da aeronave que responde ao radar secundário é chamado transponder. O radar secundário envia uma "interrogação" para aeronave que responde com a informação solicitada (TRIM, 2002). O sinal transmitido é composto de 3 sinais chamados de P1, P2 e P3, em que a diferença de tempo entre P1 e P3 define qual o modo de interrogação que está sendo realizado, como, por exemplo, o código da aeronave ou sua altitude em relação ao solo. O sinal de P2 serve para que o equipamento saiba se está sendo utilizado o lóbulo principal da antena ou algum lóbulo secundário, o qual existe devido a aspectos físicos da construção das antenas (ORLANDO, 1989), como pode ser observado na Figura 3. Quando o sinal de P2 for maior que P1 e P3, o extrator do radar descarta a resposta da aeronave, visto que não pode garantir que a detecção tenha ocorrido através do lóbulo principal ou de algum lóbulo secundário.

Figura 3 – Supressão de lóbulos secundários.



Fonte: adaptado de (ORLANDO, 1989)

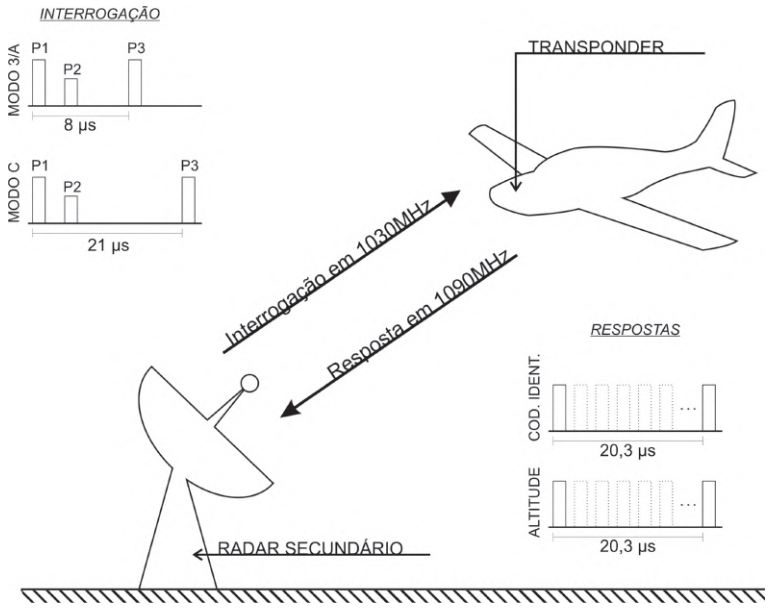
A forma de comunicação descrita anteriormente é conhecida como Modo A/C, onde "A"

é a identificação da aeronave e “C” é a altitude. A frequência da portadora de transmissão do radar é 1030 MHz e a portadora do transponder da aeronave é 1090 MHz (WESSON; HUMPHREYS; EVANS, 2014). A Figura 4 ilustra a comunicação entre o radar SSR e a aeronave, onde é possível observar que a diferença de tempo entre P1 e P3 em 8 µs significa que o radar está solicitando à aeronave informação do código “3/A”, ou seja, sua identificação. Quando o tempo entre P1 e P3 for de 21 µs, a interrogação é para o código “C”, altitude. A resposta da aeronave para o radar terá sempre a duração de 20,3 µs com a informação solicitada dentro da palavra digital transmitida.

Os radares secundários possuem algumas limitações, tais como:

- Limite de 4096 códigos diferentes de identificação para aeronaves, visto que a mensagem de identificação é composta de 12 bits;
- FRUIT (*False Replies Unsynchronized with Interrogator Transmissions*) que é a resposta de uma aeronave relativa à interrogação de um radar próximo;
- *Garbling* que é o efeito quando duas aeronaves respondem ao mesmo tempo a uma determinada interrogação, ocasionando sobreposição de sinal no espaço e modificando a resposta recebida pelo receptor do radar.

Figura 4 – Comunicação entre o radar secundário e a aeronave.



Fonte: O Autor (2023).

2.4.1 Modo S

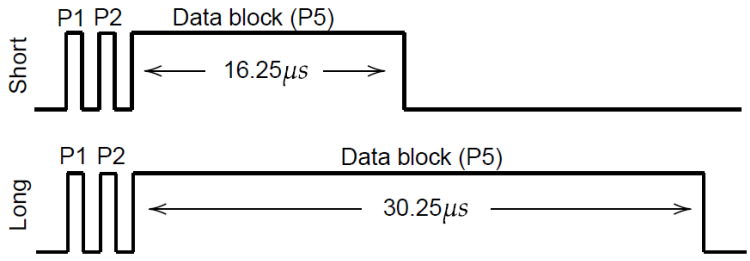
Para minimizar as deficiências dos radares secundários, foi desenvolvido o modo S, em que “S” significa seletivo. Essa nova tecnologia interroga as aeronaves individualmente, visto

que cada aeronave possui um registro formado por 24 bits, possibilitando mais de 16 milhões de códigos distintos (ORLANDO, 1989).

2.4.1.1 Interrogações no modo S

As interrogações no modo S indicam qual a informação desejada pelo controle de tráfego aéreo, como: identificação e altimetria, por exemplo. Elas podem ser de dois tipos: (I) mensagem curta, com 56 bits de informação; (II) mensagem longa, com 112 bits. O pulso P2 serve como supressão de lóbulo lateral para o modo A/C (SUN, 2021). A Figura 5 ilustra os dois tipos de pulso de interrogação no modo S, onde a informação com 56 bits está localizada em P5 com duração de 16,25 µs e a informação com 112 bits possui duração de 30,25 µs.

Figura 5 – Mensagens de Interrogação em Modo S.



Fonte: (SUN, 2021).

A modulação do bloco de mensagens transmitidas é DPSK (*differential phase shift keying*), ou seja, modulação por desvio de fase diferencial. Esse tipo de modulação é caracterizado por alterar a fase da portadora quando o nível do sinal modulante for “0” e mantem a fase quando for “1” (PROAKIS; SALEHI, 2002).

2.4.1.2 Respostas no modo S

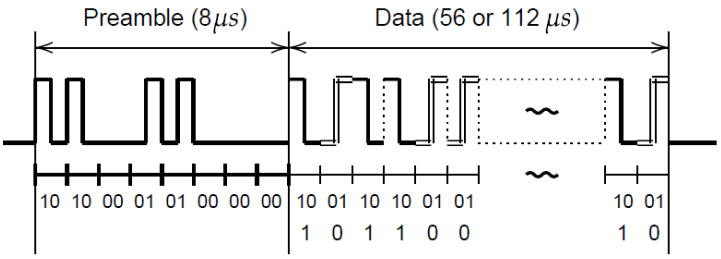
Há dois tipos de mensagem de resposta, as quais correspondem a mensagem curta e mensagem longa de interrogação. Na resposta, há um preâmbulo fixo com duração de 8 µs que precede a mensagem de resposta que poderá ser de 56 bits ou 112 bits. A Figura 6 ilustra um exemplo de mensagem de resposta em modo S.

A modulação do pacote de resposta é em PPM (*pulse position modulation*) ou modulação por posição de pulso, em que o nível “1” é representado por nível alto nos primeiros 0,5 µs, seguido de nível baixo por mais 0,5 µs. O nível “0” é o inverso, ou seja, nível baixo por 0,5 µs, seguido de nível alto por mais 0,5 µs.

2.4.1.3 Formato das mensagens

As mensagens trafegadas em modo S possuem diversos formatos separados para cada necessidade específica. Os primeiros cinco bits da mensagem definem qual o formato, tanto

Figura 6 – Exemplo de mensagem resposta em Modo S.



Fonte: (SUN, 2021).

para a mensagem de interrogação, quanto para a resposta. A Tabela 2 exibe as mensagens com tamanho de 56 bits e a Tabela 3 exibe as mensagens longas, ou seja, com 112 bits. É possível observar nas mensagens UF0/DF0 e UF16/DF16, correspondentes a primeira linha de informação de cada tabela, que essas mensagens são responsáveis por permitir a prevenção de colisão entre aeronaves que estejam voando em rotas convergentes, visto que o sistema ACAS (*Airborne Collision Avoidance System*) atua com base nesses dados.

Tabela 2 – Formato de mensagens de interrogação e resposta curto.

Formato Interrogação (UF) / Formato Resposta (DF)	Tipo de interrogação -Vigilância-	Tipo de resposta -Vigilância-
0	ar-ar curto (ACAS)	ar-ar curto (ACAS)
4	Requisição de altitude	Resposta de altitude
5	Requisição de identificação	Resposta de identificação
11	Requisição All-Call	Resposta All-Call

Fonte: adaptado de (SUN, 2021).

Tabela 3 – Formato de mensagens de interrogação e resposta longo.

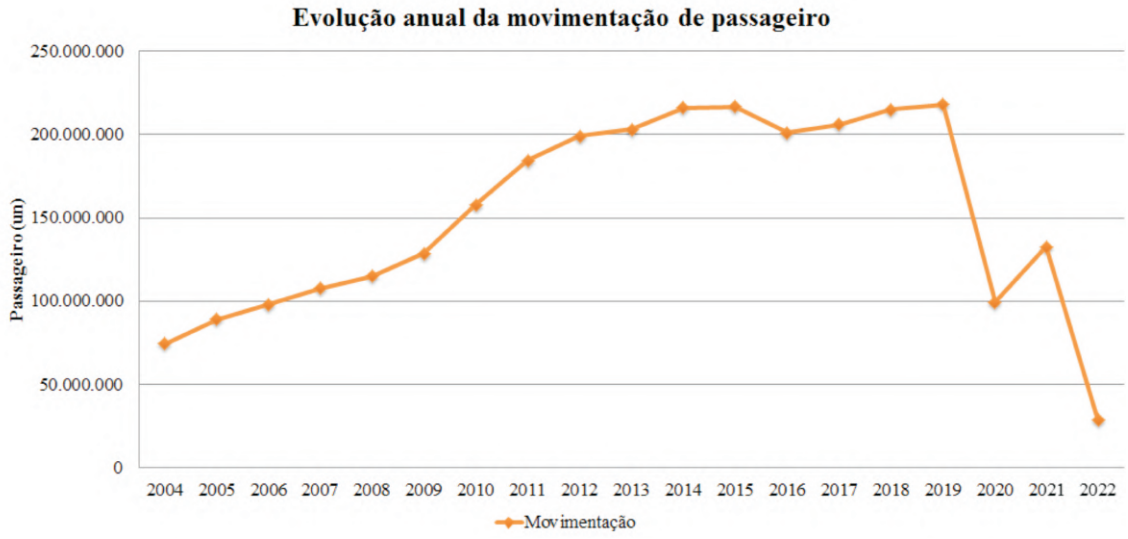
Formato Interrogação (UF) / Formato Resposta (DF)	Tipo de interrogação -Vigilância-	Tipo de resposta -Vigilância-
16	ar-ar longo (ACAS)	ar-ar longo (ACAS)
17	-	*Extended squitter*
18	-	*Extended squitter (Non transponder)*
19	-	*Military extended squitter*
20	Requisição de altitude Comm A	Resposta de altitude Comm B
21	Requisição de identificação Comm A	Resposta de identificação Comm B
24	Comm C (ELM)	Comm D (ELM)

Fonte: adaptado de (SUN, 2021).

2.5 EVOLUÇÃO DO TRÁFEGO AÉREO

A quantidade de aeronaves tem aumentado todos os anos no mundo inteiro. No Brasil, conforme dados da Agência Nacional de Aviação Civil, pode-se comprovar essa evolução, o que exige muito planejamento e também tecnologias que suportem esse avanço. A Figura 7 mostra como a movimentação de passageiros do setor aéreo cresceu até 2015 e manteve estabilidade até final de 2019, quando houve a pandemia, o que ocasionou uma abrupta queda no transporte aéreo no mundo. Os radares primários e secundários possuem uma limitação em relação à separação mínima entre aeronaves, visto que há uma taxa de atualização de informação para o controlador de tráfego aéreo que está diretamente relacionada à velocidade de rotação da antena. Nesse contexto, surge uma nova tecnologia que permite uma separação menor e, consequentemente, aumenta a capacidade do espaço aéreo, proporcionando um maior volume de aviões. Além disso, essa nova tecnologia possui maior precisão de posição, velocidade e permite a troca de um maior número de informações entre piloto e controlador, sem que haja a necessidade de interrogação pelos sistemas de controle de tráfego aéreo. O protocolo ADS-B, apresentado na próxima seção, permite essas vantagens almejadas para o gerenciamento aéreo e para a segurança na aviação (WESSON; HUMPHREYS; EVANS, 2014).

Figura 7 – Evolução anual da movimentação de passageiro.



Fonte: adaptado de (ANAC, 2022).

2.6 PROTOCOLO ADS-B

O protocolo ADS-B possui essa nomenclatura por ser:

- *AUTOMÁTICO:* não precisa ser questionado para transmitir as informações;
- *DEPENDENTE:* depende de equipamentos de bordo para transmitir os dados, tais como GPS, altímetro, barômetro, etc;

- *BROADCAST:* as informações são transmitidas em todas as direções para todos que estiverem no alcance;

As informações mais comuns transmitidas pelo ADS-B são posição, altitude e velocidade. No entanto, esse protocolo permite enviar várias outras informações para o controle de tráfego aéreo em solo, como: indicador de chamada, indicadores de precisão, indicadores de integridade e status operacional. A posição é determinada pelo GPS, a velocidade é derivada do GPS e dos sistemas inerciais e a altitude é composta de dados barométricos e de GPS, sendo os dados barométricos obtidos dos sensores de pressão atmosféricos (SUN, 2021). O ADS-B trabalha na frequência de 1090 MHz, sendo que nos EUA pode trabalhar na frequência de 978 MHz para altitudes abaixo de 18.000 pés. O ADS-B operando em 978 MHz pode transmitir dados complementares de informações de voo, tais como dados meteorológicos. O ADS-B oferece precisão de posição de 0,05 MN e precisão de velocidade em torno de 19,4 MN/h (10 m/s), sendo a taxa de atualização dessas informações a cada segundo. Esse padrão de desempenho permite que a separação lateral seja reduzida de 90 MN para 20 MN e a separação longitudinal seja reduzida de 80 MN para 5 MN em espaço aéreo sem detecção radar (WESSON; HUMPHREYS; EVANS, 2014), sendo a separação por nível (altitude) executada conforme modelos operacionais dos Órgãos de controle e obedecendo ao estabelecido para as aerovias.

2.6.1 Arquitetura do sistema ADS-B:

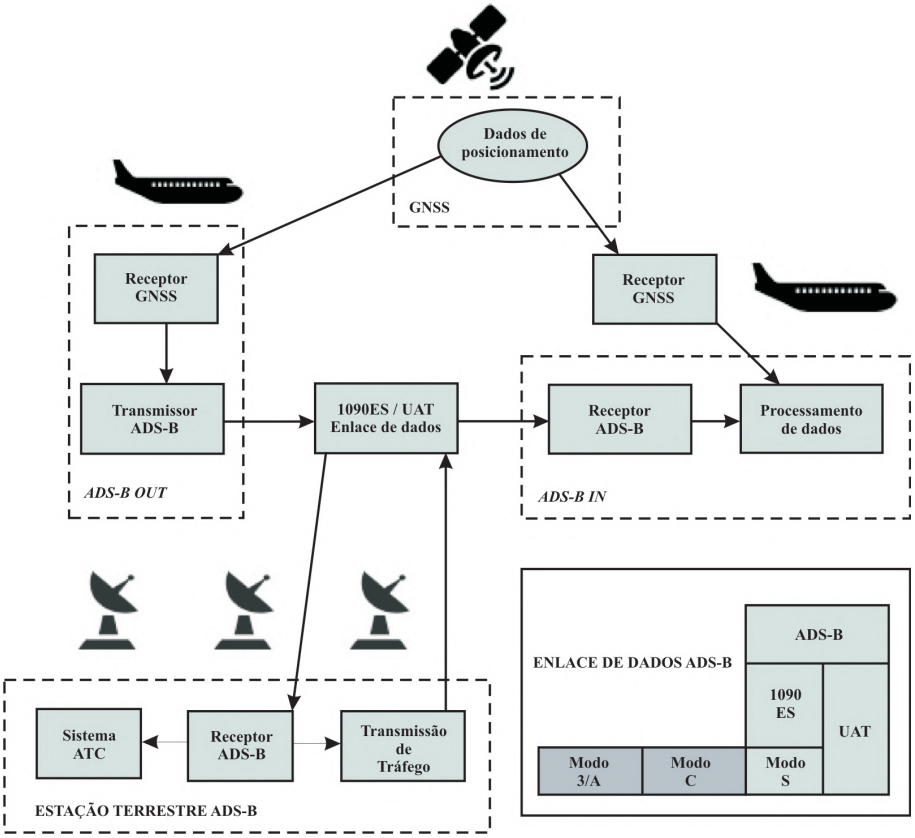
O sistema ADS-B possui dois subsistemas: ADS-B IN e ADS-B OUT. O ADS-B OUT transmite as informações periodicamente em todas as direções e o subsistema de recepção, ADS-B IN, recebe e processa esses dados. O transmissor na aeronave recebe os dados dos aviônicos embarcados, encapsula as informações e transmite para os receptores em solo, os quais são compostos por estações VHF, ou para outra aeronave (WU et al., 2020). As aeronaves podem estar equipadas apenas com ADS-B OUT ou com ambos, ADS-B IN e ADS-B OUT.

A Figura 8 ilustra a arquitetura do sistema, onde é perceptível que a aeronave, com capacidade ADS-B OUT, utiliza os dados de posicionamento do sistema global de navegação por satélite e também dos sistemas inerciais da própria aeronave para transmitir as informações de posição. Os receptores em solo, compostos por estações terrestres que fazem parte do sistema de tráfego aéreo, utilizam essas informações, assim como as aeronaves com capacidade ADS-B IN, as quais utilizam essas informações no ACAS para comparar os dados de posição da aeronave transmissora com sua própria localização. Observa-se na figura que a camada de enlaces são os diferentes níveis de transmissão de dados de vigilância, ou seja, um transponder ADS-B pode responder interrogações no modo 3/A, C, S e 1090 ES.

Os pacotes ADS-B OUT são formados por 112 bits, sendo que os 8 primeiros bits indicam o formato de dados, os próximos 24 bits se referem o indicador único da aeronave determinado pela Organização Internacional de Aviação Civil (OACI), os 56 bits seguintes são dados de vigilância e os 24 bits finais são o CRC (*Cyclic Redundancy Check*) do pacote.

Durante o voo, a aeronave transmite no formato de dados DF 17, como descrito na Tabela 3, com as informações de tempo, latitude, longitude e altitude. Quando a aeronave está na pista, outros formatos de dados são transmitidos com dados operacionais (WESSON; HUMPHREYS; EVANS, 2014).

Figura 8 – Arquitetura ADS-B.



Fonte: adaptado de (WU et al., 2020).

O ADS-B foi criado para ser compatível com os sistemas de vigilância existentes e, dessa forma, facilitar a transição entre sistemas. O ADS-B é muito semelhante ao modo S do radar secundário, tendo como principal diferença o fato de não ser necessária a interrogação por equipamento remoto, visto que o transponder da aeronave transmite constantemente as informações. Como pode ser observado na Figura 9, o pacote ADS-B possui 112 bits, sendo que 56 bits são destinados a informação de posição e altitude da aeronave e os demais bits fazem referência ao formato da mensagem, endereço da aeronave, verificação de paridade e um campo destinado a capacidade de comunicação do transponder.

Figura 9 – Campos do pacote ADS-B.

FORMATO DE DOWNLINK (5 bits)	CAPACIDADE (3 bits)	ENDEREÇO DA AERONAVE (24 bits)	DADOS ADS-B (56 bits)	VERIFICAÇÃO DE PARIDADE (24 bits)

Fonte: O Autor (2023).

A modulação utilizada para transmitir a informação é a modulação por posição de pulso (PPM) com codificação Manchester.

2.6.2 ADS-B na bacia de Campos/RJ

O Departamento de Controle do Espaço Aéreo (DECEA) conduz o programa SIRIUS, o qual é voltado para a evolução do Sistema de Controle do Espaço Aéreo Brasileiro (SISCEAB), em resposta às demandas provenientes do crescimento e do aumento da diversidade do tráfego aéreo previstos para as próximas décadas e das evoluções tecnológicas no campo da aviação. Nesse contexto e devido ao aumento da aviação *offshore* no Brasil, o DECEA resolveu aplicar o gerenciamento de tráfego aéreo na Bacia de Campos/RJ com a utilização do ADS-B. Essa região é responsável por mais de 80 por cento da extração de petróleo no País, com uma abrangência de 115 mil quilômetros quadrados de área com dezenas de plataformas marítimas. O fluxo aéreo de pessoal e mercadorias chega a 120 voos diários, composto exclusivamente de asas rotativas. São ao todo seis estações receptoras de sinais ADS-B. Quatro no mar, instaladas sobre plataformas marítimas, e duas em terra firme. Elas estão integradas ao sistema de tratamento e visualização de dados para controle pelos controladores de voo do órgão operacional. É possível monitorar uma série de informações como identificação, altitude, velocidade, direção e localização. O programa SIRIUS prevê a extensão da vigilância ADS-B em todo o território nacional com foco nas áreas que possuem pouca sobrecobertura de radares (DECEA, 2022).

2.7 CONSIDERAÇÕES

O transporte aéreo tem papel fundamental na vida das pessoas e na economia. A demanda por esse tipo de transporte vem aumentando no mundo inteiro e a segurança na aviação precisa acompanhar essa evolução. Desde os primórdios da aviação, a tecnologia vem desempenhando esse papel fundamental para garantir a segurança ao mesmo passo que otimiza o fluxo aéreo. Dessa forma, os radares foram uma revolução e permanecem essenciais até hoje, mas a comunicação ADS-B foi desenvolvida para cobrir espaços desguarnecidos sem reduzir a precisão e com custos relativamente baixos em comparação aos radares convencionais. No entanto, precisa-se discutir e avaliar a segurança da informação nesse novo sistema.

3 SEGURANÇA DA COMUNICAÇÃO ADS-B

A tecnologia ADS-B proporciona maior segurança operacional, principalmente nas regiões onde não há detecção por radares. Além disso, o custo para instalar e manter é mais atrativo em comparação aos radares primários e secundários. No entanto, por ser um protocolo aberto e sem qualquer tipo de cifragem, torna-se vulnerável a ataques terra-terra, terra-ar e ar-ar. Um ataque terra-terra, por exemplo, seria um ataque sobre uma estação VHF que realiza o bloqueio da recepção dos pacotes de dados. Esse capítulo trata das vulnerabilidades do protocolo ADS-B para a aviação, além de realizar análise sobre propostas de segurança para contornar as fragilidades apontadas.

3.1 VULNERABILIDADES DO ADS-B

A importância das informações de status operacional da aeronave, tais como mensagens de posição, identificação, velocidade, dentre outros, devem estar protegidas de invasores mal-intencionados, por representarem grande importância na segurança operacional.

3.1.1 Espionagem

A vulnerabilidade de espionar informações de status operacional de aeronaves (reconhecimento de aeronaves) se caracteriza pela obtenção de dados ADS-B do espaço aéreo correspondente, com uso do dispositivo ADS-B IN, ou seja, efetua a leitura de dados ADS-B transmitido por transponder ADS-B OUT das aeronaves. Segundo (WU; SHANG; GUO, 2020), desde a criação da comunicação ADS-B, algumas iniciativas têm utilizado a captura de dados para prestar um serviço legítimo para usuários. Nesse contexto, pode-se citar o <flightradar24.com>, o qual presta informações em tempo real sobre voos e dados de aeronaves que estão em rota através da rede ADS-B, contando com mais de 20.000 receptores espalhados pelo mundo. No entanto, não exclui que alguns invasores mal-intencionados possam usar essa vulnerabilidade para lançar ataques complexos.

3.1.2 Interferência

O bloqueio na transmissão de uma mensagem ADS-B em um espaço aéreo específico, utilizando dispositivo de transmissão ADS-B com potência de transmissão suficientemente alta na banda de frequência relevante, é uma técnica de interferência explícita nas informações ADS-B com grande impacto operacional para o controle do espaço aéreo afetando a integridade e disponibilidade das informações.

Há dois tipos principais de ataques de interferência contra a comunicação ADS-B, segundo (WU; SHANG; GUO, 2020): negação de serviço das estações de solo e negação de serviço das aeronaves. O objetivo desses ataques é interromper a rede de monitoramento

bloqueando o canal de comunicação. É mais fácil lançar um ataque a uma estação terrestre do que diretamente a uma aeronave, pois exige uma quantidade menor de energia.

3.1.3 Injeção de Mensagem

A injeção de informações de falsas aeronaves em cenários de voo específicos, confundindo os sistemas de controle de tráfego aéreo (injeção fantasma de alvo de aeronave), usando um dispositivo de transmissão com potência de transmissão alta o suficiente na faixa de frequência relevante e capaz de gerar a modulação correta e em conformidade com o formato de mensagem ADS-B, é outro tipo de ataque que explora a vulnerabilidade da comunicação ADS-B. Esse efeito é conseguido por não haver autenticação nas mensagens (STROHMEIER; LENDERS; MARTINOVIC, 2013). A Figura 10 ilustra a criação de 100 alvos falsos. Esse fator impacta na integridade das informações.

Figura 10 – Injeção de alvos falsos.

Fonte: (SCHÄFER; LENDERS; MARTINOVIC, 2013).

3.1.4 Exclusão de Mensagem

Excluir algumas ou todas as informações contidas em uma mensagem (desaparecimento de aeronaves) é um ataque implementado na camada física através de interferência construtiva ou destrutiva.

A interferência construtiva, como meio de interferência, causa um grande número de erros de bits. Como o CRC (*Cyclic Redundancy Check*) das mensagens transmitidas de ADS-B podem corrigir um máximo de 5 bits por mensagem, caso uma mensagem exceda esse limite, o receptor a descartará como corrompida (STROHMEIER; LENDERS; MARTINOVIC, 2013).

A exclusão da mensagem terá impacto no sistema de vigilância, fazendo com que a aeronave desapareça temporariamente da tela do controle de tráfego aéreo, mas pode ser identificada por sistemas de vigilância como radar primário, secundário e sistemas de multilateração.

3.1.5 Modificação da Mensagem

A modificação das informações contidas em uma mensagem, pode ser realizada através de ofuscamento e inversão de bits no pacote da mensagem.

A modificação da mensagem é um ataque de falsificação típico. Por exemplo, se um invasor altera continuamente as informações de posição da aeronave nas mensagens ADS-B em pequenas quantidades, isso é considerado um ataque de falsificação do tipo “fervura de sapo”. Esse termo se refere a analogia com um sapo que quando inserido em um recipiente com água fervente, irá instantaneamente pular para fora. No entanto, se inserido em um recipiente com água em temperatura ambiente e a temperatura for sendo elevada aos poucos, esse sapo morrerá sem sentir a fervura da água, como descrito por (CHAN-TIN et al., 2011). Neste momento, outras tecnologias de vigilância (como sistemas de vigilância por radar) e tecnologia de posicionamento terão dificuldade em detectar essas pequenas diferenças devido a problemas de precisão, resultando em orientação incorreta aos controladores de tráfego aéreo ou atraso na resposta do sistema de prevenção de colisões. Isso tem um grande impacto no sistema ATC, (WU; SHANG; GUO, 2020), principalmente em na integridade e confidencialidade.

3.2 ESTUDOS ATUAIS SOBRE SEGURANÇA NA COMUNICAÇÃO ADS-B

Alguns pesquisadores se debruçaram sobre o problema da vulnerabilidade da comunicação ADS-B e propuseram soluções para mitigação da maioria dos problemas elencados, com vistas a manter os requisitos de integridade, disponibilidade e confidencialidade. Dentre os trabalhos realizados, pontua-se as soluções de cifragem simétrica e autenticação de mensagens. Cada solução possui pontos positivos e negativos que merecem destaque. A revisão conduzida por (WU; SHANG; GUO, 2020) explana as análises conduzidos até o presente momento e os subdivide em:

1. Verificação de local seguro:
    - Multilateração;
    - Filtro de Kalman;
    - Certificação em grupo;

- Fusão de dados;
- Distância limite;
- Modelagem de tráfego.

2. Autenticação de transmissão segura:

Esquemas não-cifrados:

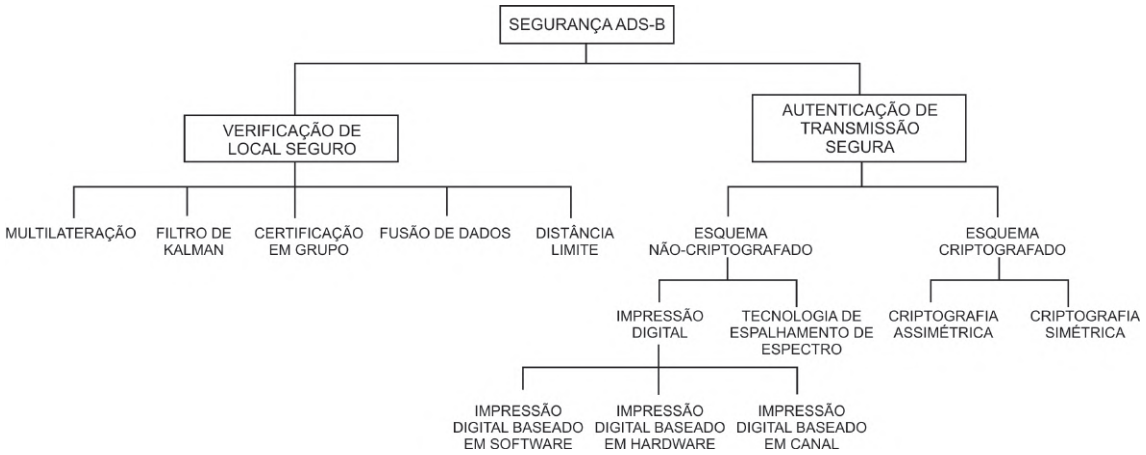
- Impressão digital baseada em software;
- Impressão digital baseada em hardware;
- Impressão digital baseada em canal; e
- Tecnologia de espalhamento de espectro.

Esquemas cifrados:

- cifragem assimétrica;
- cifragem simétrica.

Cada método possui vantagens e desvantagens para implementação ou utilização pelo controle de tráfego aéreo. A Figura 11 exibe a distribuição das soluções apresentadas.

Figura 11 – Classificação das soluções de segurança.



Fonte: Adaptado de (WU; SHANG; GUO, 2020).

3.2.1 Verificação de Local Seguro

O objetivo da verificação de local seguro é garantir que a informação recebida de posição foi transmitida por uma aeronave que estava sobrevoando o local informado na mensagem. Para isso, há diferentes abordagens para confirmação dessas mensagens. A verificação de local seguro não garante a autenticação da mensagem propriamente dita, apenas afirma que a informação de posição está condizente com o local onde ocorreu a transmissão. Segue abaixo vários métodos distintos para obtenção da verificação de local seguro:

- Multilateração: a tecnologia de multilateração se baseia no cálculo da diferença de tempo de chegada do sinal transmitido pela aeronave em relação a recepção nas antenas, com a finalidade de aproximar a posição de transmissão, também chamado de TDOA (*Time Difference Of Arrival*). O sistema recebe sinais periódicos ADS-B, sinais de resposta do transponder ao questionamento de radar secundário ou sinais de resposta TCAS (*Traffic Collision Avoidance System*).

  Desvantagens: a ICAO (*International Civil Aviation Organization*) lista algumas deficiências conhecidas dessa técnica, como: susceptibilidade a efeitos multicaminho, dependência de múltiplas estações terrestres para definir a localização da aeronave, unidade central de processamento deve estar conectada a várias estações simultaneamente e a dificuldade e alto valor para instalar estações em locais remotos.

- Filtro de Kalman: o filtro de Kalman é uma técnica que utiliza o algoritmo de filtragem de Kalman para excluir dados inválidos, prever posição futura e filtrar ruído de sinal. O filtro de Kalman utiliza sistemas lineares de equações de estado para observar a entrada do sistema e os dados de saída para estimar o estado do sistema.

  Desvantagens: o filtro de Kalman pode ser afetado por dois tipos de ataques, sendo que o primeiro está relacionado a injeção de um alvo falso com baixa atualização de posição, sendo que depois de um tempo o filtro não saberá distinguir. O segundo é o ataque de negação de serviço, como o filtro de Kalman processa as informações por longo período a complexidade aumenta com o passar do tempo e a probabilidade de sofrer um ataque de negação aumenta proporcionalmente.

- Certificação de grupo: a certificação de grupo utiliza a técnica de multilateração entre integrantes do mesmo grupo para determinar a posição de cada integrante, além de certificação entre os membros para estabelecer uma comunicação segura.

  Desvantagens: para que seja possível as aeronaves obterem dados de multilateração, todas precisarão estar capacitadas com ADS-B IN, sendo que essa funcionalidade não é obrigatória na especificação ADS-B. A comunicação ADS-B foi concebida para ser de caminho único.

- Fusão de dados: a fusão de dados utiliza várias fontes de informação para determinar a localização, velocidade e demais informações de performance das aeronaves. Dentre as fontes de informação, cita-se radares primários, radares secundários, ADS-B, multilateração, etc. Além disso, técnicas como modelagem probabilística, *machine learning* e lógica fuzzy, podem ser usadas para tratar esses dados.

  Desvantagens: embora a solução seja muito precisa, ela é estritamente dependente da sincronização entre todas as informações. No foco da segurança da comunicação ADS-B, essa solução necessita de várias fontes disponíveis para garantir que a mensagem ADS-B recebida é realmente daquela fonte.

- Distância limite: a distância limite é uma técnica que efetua a interação de tempo entre dois dispositivos em um meio. Há um dispositivo de verificação e um dispositivo de prova. A determinação da distância entre eles é baseada na velocidade de propagação do sinal, acrescido o tempo de processamento do dispositivo de prova. A implementação da limitação de distância se baseia no fato de que as ondas eletromagnéticas viajam perto da velocidade da luz. A distância dos dois dispositivos é então calculada e provada para ver se a aeronave está na mesma região informada pelos dados ADS-B.

  Desvantagens: os cálculos para computar e validar os dados levam um tempo significante quando comparado com a velocidade das aeronaves, principalmente aeronaves comerciais, pois ao término dos cálculos, no momento da decisão, a aeronave já terá percorrido grande distância.

3.2.2 Autenticação de Transmissão Segura

A autenticação de transmissão possui por objetivo garantir a segurança prevenindo ou detectando ataques na comunicação unidirecional. As soluções desenvolvidas estão organizadas como esquemas não-cifrados e esquemas cifrados.

- Esquemas não-cifrados
  - Impressão digital baseada em software: a impressão digital baseada em software está relacionada a identificação de determinados dispositivos através do comportamento de software.

    Desvantagens: atualmente muitas companhias aéreas utilizam os mesmos modelos de equipamentos ou modelos muito semelhantes, dificultando a distinção.
  - Impressão digital baseada em hardware: esse método utiliza variações de comportamento de hardware para determinar dispositivos. Um exemplo está na variação de transientes ou diferenças na modulação dos sinais de rádio transmitidos, como assinaturas digitais desses dispositivos.

    Desvantagens: esse método é utilizado atualmente em equipamentos não móveis, sendo um desafio aplicar para uma solução ao ADS-B, visto que a mobilidade das aeronaves poderiam causar efeitos que diminuiriam a precisão do esquema.
  - Impressão digital baseada em canal: esse tipo de análise leva em consideração características específicas encontradas na propagação de sinais em canais, tais como nível do sinal recebido, resposta ao impulso e fase da portadora.

    Desvantagens: do ponto de vista da segurança ADS-B, o fato da comunicação ser unidirecional, acaba por limitar a utilização da impressão digital baseada em canal.
  - Tecnologia de espalhamento espectral: a tecnologia de espalhamento espectral é usada principalmente em comunicação sem fio para combater interferência e espionagem, incluindo espalhamento espectral de sequência direta e espalhamento

espectral de salto de frequência. Nesse método, o transmissor e o receptor precisam compartilhar previamente o código de espalhamento ou o modo de salto de frequência durante o uso.

Desvantagens: a tecnologia de espalhamento espectral também apresenta problemas de gerenciamento e distribuição de chaves.

- Esquemas cifrados

  As técnicas de cifragem são um método de proteção das comunicações amplamente utilizado atualmente, inclusive nas comunicação sem fio, sendo o seu estudo importante para aplicações no ambiente ADS-B. O método de cifragem pode ser dividido em cifragem simétrica e cifragem assimétrica, dependendo do tipo de chave criptográfica utilizada. Conforme (AMIN et al., 2014), há três formas de garantir segurança na comunicação ADS-B:

  – Autenticação (Hashing): o objetivo da autenticação é confirmar que a mensagem foi transmitida por uma fonte confiável. Esse método acrescenta um valor ao final da mensagem chamado de *hash*. O receptor calcula o valor *hash* e compara, independentemente, com o valor *hash* anexado ao pacote principal. Caso sejam iguais, a mensagem é considerada verdadeira e de fonte confiável.

    Desvantagens: para a utilização desse método na comunicação ADS-B, há necessidade de alteração na quantidade de bits do formato ADS-B especificado.

  – cifragem simétrica: a utilização da cifragem serve para transformar um texto claro em texto cifrado para torná-la ilegível. O principal motivo da utilização dessa técnica é manter a confidencialidade, não repúdio, autenticidade e integridade. A autenticidade garante que apenas o remetente e o destinatário pretendido possam ver a mensagem. O não repúdio é a capacidade do algoritmo de cifragem de fornecer prova da origem da mensagem. A Integridade se refere ao conteúdo da mensagem e à precisão das informações enviadas na mensagem. Para cifragem simétrica, cada entidade possui uma chave secreta para cifrar as mensagens ADS-B. As entidades receptoras também têm acesso a essas chaves e usam as chaves para descifrar o conteúdo.

    Desvantagens: problema de segurança das trocas de chaves.

  – cifragem assimétrica: há duas chaves, chamadas de chave privada e pública. A chave pública é divulgada ostensivamente, enquanto a chave privada deve ser do conhecimento apenas da entidade específica. Essas chaves são matematicamente dependentes uma da outra, visto que o texto pleno é cifrado pela chave privada e decifrado pela chave pública.

    Devantagens: esse método necessita de uma forma de compartilhar as chaves públicas entre as entidades.

3.3 CONSIDERAÇÕES

A comunicação ADS-B trouxe uma solução de baixo custo para a vigilância aérea em locais onde a instalação de um radar é impraticável do ponto de vista técnico ou econômico. Além disso, a quantidade de informações fornecidas por esse protocolo possibilita ao sistema de controle de tráfego aéreo efetuar o gerenciamento do fluxo com grande precisão, aumentando a capacidade operacional e garantido alto nível de segurança. No entanto, existe uma vasta gama de vulnerabilidades que permite ataque direto a esse protocolo causando sério risco à aviação.

4 CIFRAGEM COM PRESERVAÇÃO DE FORMATO

A cifragem com preservação de formato, conhecido como FPE (*Format Preserving Encryption*), consiste na técnica de cifragem baseada em criptografia simétrica, sendo que o texto cifrado mantém o mesmo formato do texto original, ou seja, caso a mensagem original seja composta por alfabeto com números decimais, a mensagem cifrada também será composta por alfabeto com números decimais. Além disso, o tamanho do pacote de dados permanece inalterado, com a mesma quantidade de bits/bytes/caracteres da mensagem original. O FPE permite um caminho de migração mais simples quando a cifragem é adicionada a sistemas legados, uma vez que os custos para alterar as bases de dados e aplicações desses sistemas, no caso de utilização de técnicas de cifragem que alteram o formato dos dados, são elevados (BELLARE et al., 2009).

4.1 HISTÓRIA DA FPE

A idéia inicial da FPE surgiu em 1981, quando o *National Bureau of Standards*, posteriormente denominado NIST (*National Institute of Standards and Technology*), publicou o FIPS 74 (NIST, 1981), que descreve uma abordagem para cifrar uma string arbitrária sobre um alfabeto não-binário baseada em DES (*Data Encryption Standard*) (ROGAWAY, 2010).

Os autores Brightwell e Smith, em 1997 (BELLARE et al., 2009), consideraram um cenário mais geral e denominaram como cifragem de preservação de tipo de dados o método para cifrar as entradas do banco de dados, levando em conta algum tipo de dado específico, sem alterar o formato desse tipo de informação.

Os pesquisadores Black e Rogaway estudaram uma forma de segurança com FPE, cujas soluções focaram em um domínio arbitrário X relacionado ao domínio $\mathbb{Z}_N$, $X = \mathbb{Z}_N$ para os inteiros $\{0, 1, ..., (N-1)\}$ (BLACK; ROGAWAY, 2002). Eles propuseram três métodos de cifragem de preservação de formato, são eles:

- Cifra de Prefixo: na cifra de prefixo, são atribuídos pesos pseudoaleatórios em cada número inteiro em um texto simples, pertencente a $\mathbb{Z}_N$. Esses pesos são definidos pela aplicação de uma cifra de bloco existente a cada número inteiro. Para cifrar dados com esse método, primeiro se constrói uma tabela que armazena uma permutação sobre o conjunto completo de texto simples e, posteriormente, procura-se o valor do texto cifrado usando o texto simples. Isso significa que a cifragem e decifragem são muito rápidas, no entanto, a aplicabilidade desse método fica restrito a conjuntos pequenos.

- Cifra *Cycle-Walking*: a construção do *Cycle-Walking* funciona cifrando o texto claro com uma cifra de bloco existente repetidamente até que a cifra fique em um intervalo aceitável. Através de um texto claro, cria-se um algoritmo FPE utilizando a cifra de bloco, aplicando repetidamente cifras de bloco até que o resultado satisfaça o intervalo FPE necessário. O desempenho desse método está diretamente relacionado à diferença do tamanho do texto

claro e do tamanho do formato da saída. Como exemplo, pode-se citar a cifragem de um texto claro formado por 6 dígitos decimais que deverão ser representados por um número de 20 bits. A relação dos domínios remete a seguinte relação: $2^{20}/10^6 \approx 1{,}05$, o que não seria uma relação ruim para o desempenho. No entanto, caso fosse necessário cifrar um número de 16 dígitos representado por um número de 128 bits, elevaria essa relação para algo não prático (BLACK; ROGAWAY, 2002).

- Cifra de Feistel: a rede de Feistel foi criada por Horst Feistel na década de 1960. A rede Feistel é caracterizada pela divisão de sua entrada em duas partes que são embaralhadas e combinadas com uma sequência de chaves e uma função F_k. Uma rede Feistel precisa de uma fonte de valores pseudoaleatórias para as subchaves de cada rodada. Essa rede é uma estrutura de cifragem denominada "Cifragem/Decifragem Similar".

Os dois primeiros métodos funcionam bem para pequenos domínios, perdendo desempenho e segurança com o aumento no número de informações das mensagens.

4.1.1 Sinopse da FPE

Conforme Claude Shannon, autor da Teoria da Informação, existem duas operações primitivas que permitem construir algoritmos com forte segurança criptográfica. São elas: confusão e difusão, (PAAR; PELZL, 2010; SHANNON, 1948).

- Confusão: considera-se confusão uma operação que torna a relação entre chave criptográfica e texto cifrado obscura. No caso da cifra de bloco AES, essa operação é realizada por substituição de bytes.

- Difusão: a difusão tem por objetivo mascarar a influência de algum símbolo do texto claro em relação ao texto cifrado. Dessa forma, a técnica realiza o espalhamento da informação de um símbolo do texto claro em vários símbolos do texto cifrado. Com isso, oculta-se as propriedades estatísticas do texto claro. O AES utiliza uma operação chamada Mixcolumn para realizar a difusão (PAAR; PELZL, 2010).

Atualmente as cifras de bloco mais modernas realizam diversas vezes as operações de confusão e difusão. Cada conjunto de operação de confusão/difusão é chamada de rodada ou *round*. Cifras que utilizam somente confusão ou somente difusão não são seguras. Pode-se citar a Máquina Enigma, desenvolvida pela Alemanha na Segunda Guerra Mundial, a qual era baseada em uma Cifra de deslocamento que realizava apenas a operação de confusão (PAAR; PELZL, 2010).

A FPE é determinística, ou seja, sempre que um texto claro for cifrado por uma chave específica, o resultado será o mesmo texto cifrado (ROGAWAY, 2010). Uma vantagem é que os dados não precisam ser necessariamente binários no FPE. Qualquer conjunto finito de símbolos, como os numerais decimais, podem ser cifrados por esse método e terão seu conjunto preservado,

incluindo o tamanho da sequência de símbolos. O trabalho de (BELLARE; ROGAWAY, 2010), descreve com detalhes os modos de operação da FPE.

O Instituto Nacional de Padrões e Tecnologia dos Estados Unidos, conhecido como NIST, faz parte do Departamento de Comércio dos EUA. Atualmente o NIST é responsável por realizar a aprovação dos algoritmos de cifragem a serem utilizados nos EUA. Conforme (ROGAWAY, 2010), foram submetidas para aprovação três modos de FPE em novembro de 2010, denominados FFX, BPS e VAES-3. Eles utilizam a rede de Feistel e foram nomeados como FF1 (FFX), FF2 (VAES-3) e FF3 (BPS). Em (DWORKIN, 2016) é detalhado cada modo de funcionamento e realiza comentários sobre a segurança criptográfica.

A estrutura do FPE fornece confidencialidade em relação aos dados de texto claro e cada modo também recebe uma entrada adicional chamada *tweak*, que não precisa ser necessariamente secreta. O *tweak* pode ser considerado como uma parte da chave que pode ser alterada, visto que determinam as funções de cifragem e decifragem.

A Agência Nacional de Segurança dos EUA (*National Security Agency - NSA*), notificou o NIST sobre falha de segurança do método FF2. Esta nota descreve um ataque teórico, com a escolha de um de texto simples, que mostra que a força de segurança do FF2 é inferior a 128 bits, (DWORKIN; PERLNER, 2015).

O algoritmo FF3 é baseado em uma rede de Feistel com 8 (oito) rodadas. Os autores F. Betul Durak e Serge Vaudenay realizaram um estudo em que se prova a existência de vulnerabilidade no método FF3, como descrito em (DURAK; VAUDENAY, 2017), a qual permite quebrar a cifragem da rede Feistel com 8 rodadas, quando utilizado pequeno domínio. Para isso, eles exploram a diferença de domínio do algoritmo. Para o ataque, eles levam em consideração a permutação de funções de rodadas e utilização de valores de *tweak* com texto claro conhecido.

O NIST em resposta ao trabalho de Durak e Vaudenay (NIST, 2019), informou sobre a intenção de revisar a especificação do FF3 e, para isso, verificou a possibilidade de reduzir o tamanho de seu parâmetro de ajuste (tweak) de 64 bits para 48 bits ou para retirar a aprovação do FF3. No SP 800-38G Revisão 1, o parâmetro tweak foi reduzido para 56 bits. O FF3 revisado foi nomeado como FF3-1.

Com base nas considerações de segurança apontados, optou-se por utilizar o modo FF1, como descrito em (DWORKIN, 2016), para realização do estudo desse trabalho.

4.1.2 Modo FF1

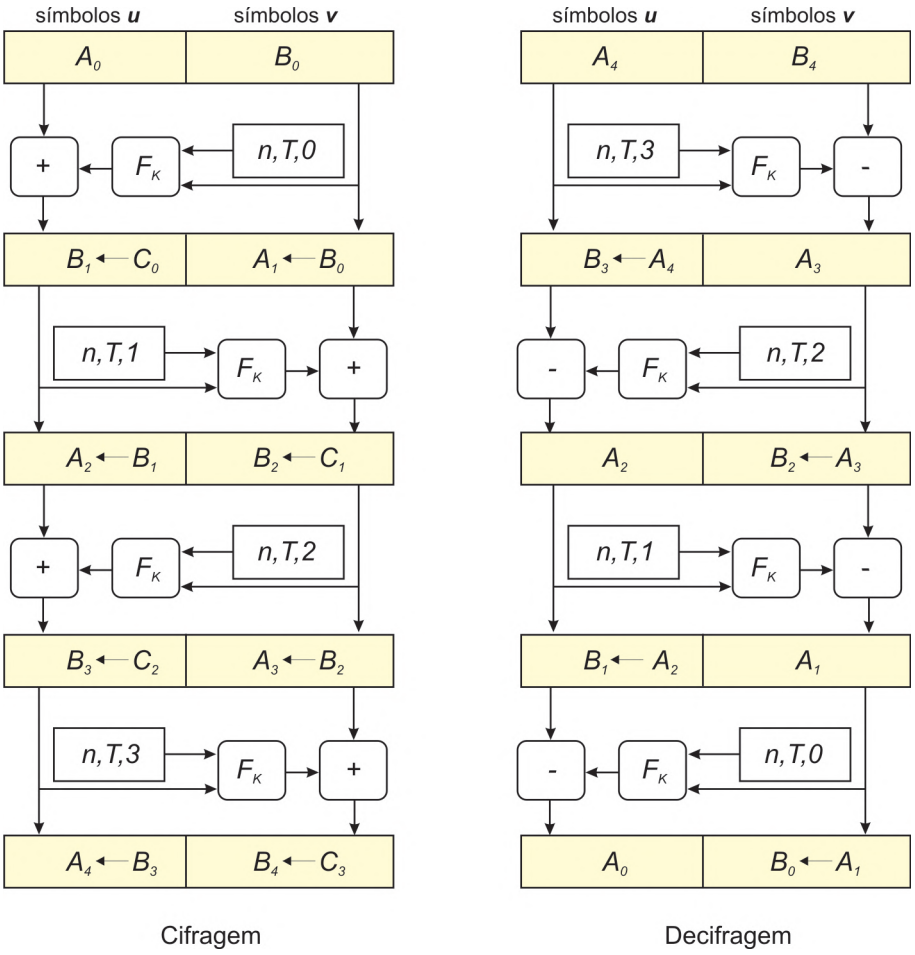
O modo FF1 é derivado do algoritmo FFX desenvolvido por Bellare, Rogaway e Spies, o qual foi submetido ao NIST em 2010. A finalidade desse modo é possibilitar a cifragem de mensagens com pequeno espaço de símbolos e mensagens relativamente curtas, sem comprometer a segurança. Esse algoritmo é baseado na rede de Feistel e usa o cifra de bloco AES de 128 bits como função de rodada. Uma grande vantagem desse método é a possibilidade de trabalhar com mensagens de diferentes tamanhos e formatos, inclusive com número ímpar de símbolos por mensagem (ROGAWAY, 2010) e (BELLARE; ROGAWAY, 2010).

4.1.2.1 Estrutura da Rede Feistel

A estrutura da rede Feistel consiste de diversas iterações chamadas de rodadas, as quais executam as funções de confusão e difusão (DWORKIN, 2016) e (BELLARE; ROGAWAY, 2010). Essa rede permite cifragem e decifragem similares e funciona da seguinte forma:

1. Os dados iniciais são divididos em duas partes, procurando manter a simetria, ou seja, a mesma quantidade de bits em cada metade da mensagem dividida;
2. Aplicação de função de rodada (função pseudoaleatória), controlada por uma chave criptográfica, em uma das partes do texto claro;
3. Realização de permutação entre as partes que foram divididas inicialmente para compor a próxima rodada.

Figura 12 – Estrutura da rede Feistel.



Fonte: adaptado de (DWORKIN, 2016).

A Figura 12 exibe a estrutura desse método com um exemplo de 4 rodadas.

Como descrito por (PAAR; PELZL, 2010), as redes de Feistel permitem criar cifras muito fortes. Uma das grandes vantagens é que o processo de decifragem é muito semelhante ao

processo para cifrar o texto claro. Os dados de entrada são divididos em duas strings numéricas, nomeadas de U e V, sendo u o tamanho da string U e v o tamanho da string V. O número total de caracteres, é chamado de n, ou seja ($n = u + v$). Durante a rodada i, a função de rodada, chamada de F_k, é aplicada a uma das *strings* de entrada, chamada de B_i, com tamanho n. Os parâmetros *Tweak* T e número de rodada i são adicionados à entrada. O resultado é usado para modificar a outra *string*, chamada de A_i, via adição modular. A *string* que representa o número resultante é chamada de C_i em uma variável temporária. Os nomes das duas partes são trocadas para a próxima rodada, onde C_i será B_{i+1} e B_i será A_{i+1}, por exemplo.

Na Figura 12, os retângulos não possuem o mesmo tamanho com o objetivo de demonstrar que os dados podem ter tamanhos diferentes, ou seja, u será diferente de v se n for ímpar. Nesses casos, a função de rodada é construída de forma que o tamanho das entradas e saídas dependam se o índice da rodada é par ou ímpar.

Para a decifragem, a estrutura de Feistel é praticamente idêntica à cifragem, mas com pequenas diferenças, sendo elas:

1. As ordens dos índices são invertidas;
2. A regra da mudança dos dados na função de rodada é alterada de forma que na entrada da função estará sempre na parte de A_{i+1} e não em B_i. A saída será combinada com B_{i+1}, e não mais com A_i, justamente para produzir A_i e não B_{i+1};
3. A adição modular é substituída pela subtração modular.

4.1.2.2 Função de Rodada

Como se pode observar na Figura 12, cada rodada é composta por uma função F_k, a qual é baseada em alguma cifra de bloco, tal como no AES. No trabalho desenvolvido por (ROGAWAY, 2010), utilizou-se a cifra AES com chaves criptográficas de 128, 192 e 256 bits. Essa função de bloco pode receber 4 parâmetros distintos: bloco de informação (n), número da rodada, valor de *Tweak* (T) e chave criptográfica (K). No método FF1 a função pseudoaleatória PRF (*Pseudorandom Function*) efetua a chamada da cifra de bloco, nomeada como $CIPH_k$. A quantidade de chamadas da cifra de bloco está diretamente relacionada ao tamanho do bloco a ser cifrado, uma vez que o tamanho de cada bloco da cifra é 128 bits, qualquer bloco com tamanho superior a esse irá ensejar mais de uma chamada.

4.1.2.3 Especificações do FF1

As especificações para funcionamento do FF1, conforme descrito em (DWORKIN, 2016) são as seguintes:

- raiz $\in \{2, ..., 2^{16}\}$;
- raiz$^{\text{minlen}} \geqslant 1000000$;

- $2 \leqslant \text{minlen} \leqslant \text{maxlen} < 2^{32}$.

Nas especificações acima, *minlen* é o parâmetro relacionado ao tamanho mínimo da mensagem a ser codificada e *maxlen* é o tamanho máximo.

O algoritmo FF1 permite a cifragem de mensagem com base 2 até 2^{16}. No estudo em questão, será trabalhada a mensagem ADS-B como uma cadeia de Símbolos $\in \{0, 1\}$, ou seja, raiz = 2.

4.1.2.4 Cifragem no modo FF1

O Algoritmo 1 descreve o funcionamento da cifragem para o modo FF1. Esse algoritmo é definido da seguinte forma, conforme (DWORKIN, 2016): divisão da *string* em duas *substring*, A e B, realizada nos passos 2 e 3. Se n, que é o tamanho total da mensagem a ser cifrada, for par, o tamanho de A será igual a B, caso contrário $LEN(A) = LEN(B) - 1$. As variáveis b e d representam valores em bytes e são definidas nos passos 4 e 5. O bloco fixo P, utilizado como bloco inicial para a chamada da função pseudoaleatória PRF no passo 9, está definido no passo 6. Os passos de iteração para as 10 rodadas se inicia no passo 7 executando 9 subpassos como descrito a seguir: O *Tweak T*, *substring B* e o número de rodada *i*, são codificados como *string* binária *Q* no passo 8. A função PRF é aplicada na concatenação de *P* e *Q* para produzir um bloco *R*, que será ou truncado ou expandido para uma *string* de bytes *S* com o número de bytes definido por *d* no passo 11 (esse passo corresponde a saída de F_k na Figura 12). Nos passos de 12 a 19, *S* é combinado com a *substring A* para produzir uma string *C* com a mesma base e tamanho. No passo 12, *S* é convertido em um número *y*. No passo 13, o tamanho *m* de *A* é determinado. No passo 18, *y* é somado ao número representado pela *substring A* e o resultado é reduzido módulo raíz$^{\text{m}}$ resultando em um número *c*, posteriormente convertido em uma *string*. Nos passos 20 e 21 as ordens de *A* e *B* são trocadas para a próxima rodada. Após a décima rodada *A* e *B* são concatenados gerando a saída.

Algoritmo 1: FF1.ENCRYPT(K,T,X)

Entrada:

Texto Claro X, na base raiz de tamanho n, tal que $n \in [minlen..maxlen]$;

Tweak T, formado por t bytes, tal que $t \in [0..maxTlen]$.

Saída:

Mensagem cifrada Y, tal que LEN(Y) = n.

início

$u = \lfloor n/2 \rfloor; v = n - u$

$A = X[1..u]; B = X[u+1..n]$

$b = \lceil \lceil v * LOG(raiz) \rceil / 8 \rceil$

$d = 4\lceil b/4 \rceil + 4$

$P = [1]^1 \| [2]^1 \| [1]^1 \| [raiz]^3 \| [10]^1 \| [u \; mod \; 256]^1 \| [n]^4 \| [t]^4$

para $i \leftarrow 0 \; a \; 9$ **faça**

$Q = T \| [0]^{(-t-b-1) mod 16} \| [i]^1 \| [NUM_{radix}(B)]^b$

$R = PRF(P \| Q)$

$Seja \; S \; os \; primeiros \; d \; bytes \; do \; bloco \; de \; mensagem \; \lceil d/16 \rceil :$

$R \| CIPH_K(R \oplus [1]^{16}) \| CIPH_K(R \oplus [2]^{16}) \| ... CIPH_K(R \oplus [\lceil d/16 \rceil - 1]^{16})$

$y = NUM(S)$

if $i = par$ **then**

$m = u;$

else

$m = v;$

end

$c = (NUM_{radiz}(A) + y) \; mod \; raiz^m$

$C = STR^m_{raiz}(c)$

$A = B$

$B = C$

fim

fim

retorna $A \| B$

Como observado por (DWORKIN, 2016), os parâmetros *raiz*, *minlen* e *maxlen* afetam diretamente a segurança imposta pelo algoritmo de cifragem e por essa razão, foi modificada a especificação das quantidades possíveis de entrada de $raiz^{minlen} \geqslant 100$ para $raiz^{minlen} \geqslant 1000000$, com a finalidade de evitar um ataque *meet-in-the-middle*, o qual é quando um adversário consegue interceptar os dados trocados em um meio para poder decifrar e interferir.

4.1.2.5 Decifragem no modo FF1

O algoritmo de decifragem é similar ao algoritmo de cifragem, sendo que as diferenças consistem em: inverter a ordem dos índices, trocar as regras de A e B e substituir a adição

modular pela subtração modular. O Algoritmo 2 descreve o funcionamento da decifragem para o modo FF1.

Algoritmo 2: FF1.DECRYPT(K,T,X)

Entrada:

Texto Claro X, na base raiz de tamanho n, tal que $n \in [minlen..maxlen]$;

Tweak T, formado por t bytes, tal que $t \in [0..maxTlen]$.

Saída:

Mensagem cifrada Y, tal que LEN(Y) = n.

início

$u = \lfloor n/2 \rfloor; v = n - u$

$A = X[1..u]; B = X[u+1..n]$

$b = \lceil \lceil v * LOG(raiz) \rceil / 8 \rceil$

$d = 4\lceil b/4 \rceil + 4$

$P = [1]^1 \| [2]^1 \| [1]^1 \| [raiz]^3 \| [10]^1 \| [u \ mod \ 256]^1 \| [n]^4 \| [t]^4$

para $i \leftarrow 9 \ a \ 0$ **faça**

$Q = T \| [0]^{(-t-b-1) mod 16} \| [i]^1 \| [NUM_{radix}(A)]^b$

$R = PRF(P \| Q)$

$Seja \ S \ os \ primeiros \ d \ bytes \ do \ bloco \ de \ mensagem \ \lceil d/16 \rceil :$

$R \| CIPH_K(R \oplus [1]^{16}) \| CIPH_K(R \oplus [2]^{16}) \| ... CIPH_K(R \oplus [\lceil d/16 \rceil - 1]^{16})$

$y = NUM(S)$

if $i = par$ **then**

$m = u;$

else

$m = v;$

end

$c = (NUM_{radiz}(B) - y) \ mod \ raiz^m$

$C = STR^m_{raiz}(c)$

$B = A$

$A = C$

fim

fim

retorna $A \| B$

4.1.2.6 Função Pseudoaleatória

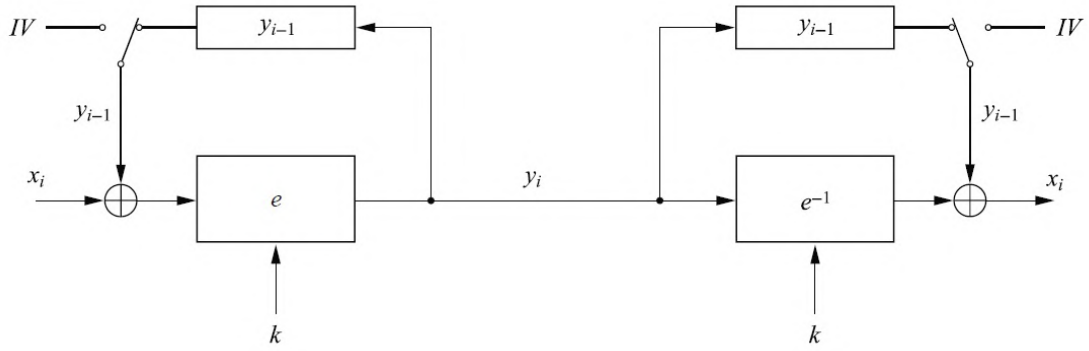
O FF1 utiliza uma função pseudoaleatória em cada rodada, chamada de *Pseudorandom Function*. O objetivo dessa função é criar uma forte permutação pseudoaleatória, ou seja, incapacidade de um observador externo diferenciar uma determinada sequência de uma outra gerada por um processo aleatório, com a finalidade de garantir a segurança, sendo, portanto, escolhida uma cifra de bloco para compor a função. Conforme (ROGAWAY, 2010), os resulta-

dos criptográficos contemporâneos e a experiência indicam que o FF1 atinge vários objetivos criptográficos. A segurança depende diretamente do número de rodadas usadas, da divisão do texto claro (balanceamento) e do acesso do adversário aos pares de texto claro/texto cifrado. A PRF do modo FF1 foi projetada para ter como saída um texto com 128 bits de comprimento, sendo sua entrada composta por múltiplos de 128 bits, além da utilização da chave criptográfica para gerar o texto de saída.

A cifra de bloco escolhida foi o AES no modo CBC-MAC (*Cipher-Block Chaining Message Authentication Code*). Segundo (ROGAWAY, 2010; DWORKIN, 2016), esse modo de operação permite uma excelente segurança criptográfica para o FF1. No entanto, outras cifras de blocos podem ser utilizadas para formar a função pseudoaletória.

Conforme descrito por (PAAR; PELZL, 2010), no modo CBC, a cifragem de todos os blocos é realizada em cadeia. Dessa forma, o texto cifrado y_i dependerá não somente de x_i e da chave, mas sim de todos os outros blocos antecessores como mostrado na Figura 13 e exposto na Definição 4.1. Esse modo de funcionamento também prevê a existência de uma variável de entrada chamada vetor de inicialização ou *IV* (*Initialization Vector*). Esse vetor serve para aumentar a aleatoriedade da cifra. No caso da função PRF do FF1, o vetor de inicialização para chamada é zero.

Figura 13 – Estrutura do modo CBC.



Fonte: (PAAR; PELZL, 2010)

A definição do modo CBC pode ser expressa da seguinte forma:

Definição 4.1. Modo CBC (PAAR; PELZL, 2010)

Seja e() uma cifra de bloco com bloco de tamanho b; sejam x_i e y_i mensagens formadas por bits com comprimento b; seja um IV de comprimento b e a realização de operação XOR.

Cifragem (primeiro bloco): $y_1 = e_k(x_1 \oplus IV)$

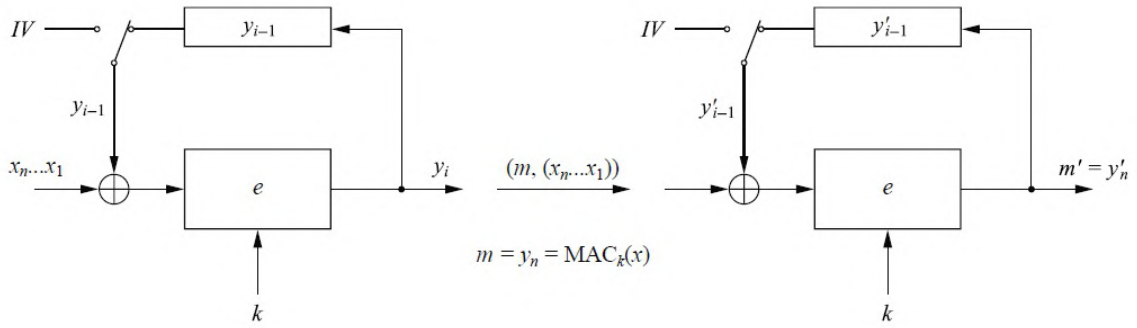
Cifragem (blocos posteriores): $y_i = e_k(x_i \oplus y_{i-1}), i \geq 2$

Decifragem (primeiro bloco): $x_1 = e_k^{-1}(y_1 \oplus IV)$

Decifragem (blocos posteriores): $x_i = e_k^{-1}(y_i \oplus y_{i-1}), i \geq 2$

O modo CBC-MAC, utilizado na função pseudoaleatória do FF1, é um método alternativo para criação de código de autenticação de mensagem ou MAC (*Message Authentication Codes*) com o uso de cifras de bloco, muito similar a assinaturas digitais, com a diferença que utiliza chaves criptográficas simétricas tanto para gerar a autenticação, quanto para realizar a verificação (PAAR; PELZL, 2010). A Figura 14 exibe o modo CBC-MAC onde m é transmitido com a mensagem, sendo o último valor calculado pelo modo CBC aplicado à mensagem.

Figura 14 – Estrutura do modo CBC-MAC.



Fonte: (PAAR; PELZL, 2010)

4.2 TIPOS DE CIFRAS DE BLOCO

Atualmente, há diversas cifras de blocos disponíveis para estudo e utilização que foram desenvolvidas para atender a crescente demanda de integração entre dispositivos. Essas cifras procuram manter a segurança das informações entre dispositivos mas levando em consideração a velocidade de processamento, quantidade de memória e hardware mínimo necessário.

4.2.1 AES

O AES, sigla para *Advanced Encryption Standard*, é uma cifra de bloco derivada do algoritmo Rijndael desenvolvido por Joan Daemen e Vincent Rijmen e submetida ao NIST em 1997, selecionada como padrão pelo governo americano em 26 de novembro de 2001. Atualmente, o AES é amplamente utilizado em diversos sistemas, tais como o padrão de segurança da Internet IPsec, TLS, Wi-Fi IEEE 802.11i e o protocolo de rede shell seguro SSH (*Secure Shell*) (PAAR; PELZL, 2010).

O NIST padronizou o AES através do FIPS PUB 197 (DWORKIN et al., 2001). O AES utiliza blocos de 128 bits e a chave criptográfica pode ser de 128, 192 ou 256 bits, as quais podem ser nomeadas como AES-128, AES-192 ou AES-256. O algoritmo é baseado em uma rede de substituição-permutação que utiliza as seguintes funções como principais:

- **MixColumns():** transformação na cifra que gera difusão dos valores da matriz de Estado. Matriz de Estado é a matriz de bytes que será manipulada entre as diversas rodadas. A

Matriz de Estado é composta de 4 linhas e Nb colunas, onde Nb é o número de bits do bloco dividido por 32;

- **ShiftRows():** transformação na cifra que processa a matriz de Estado deslocando ciclicamente as três últimas linhas por diferentes valores;
- **SubBytes():** transformação na cifra que processa a matriz de Estado usando uma tabela de substituição de bytes não linear (*S-box*) que opera em cada um dos bytes de forma independente.
- **SubWord():** função utilizada que pega uma palavra de entrada de quatro bytes e aplica uma *S-box* a cada um dos quatro bytes para produzir uma palavra de saída. Utilizada na rotina de expansão de chave.

As informações no AES são apresentadas como a concatenação de bits individuais, sendo que os valores em bytes são tratados como um polinômio em corpo finito com a seguinte representação (DWORKIN et al., 2001):

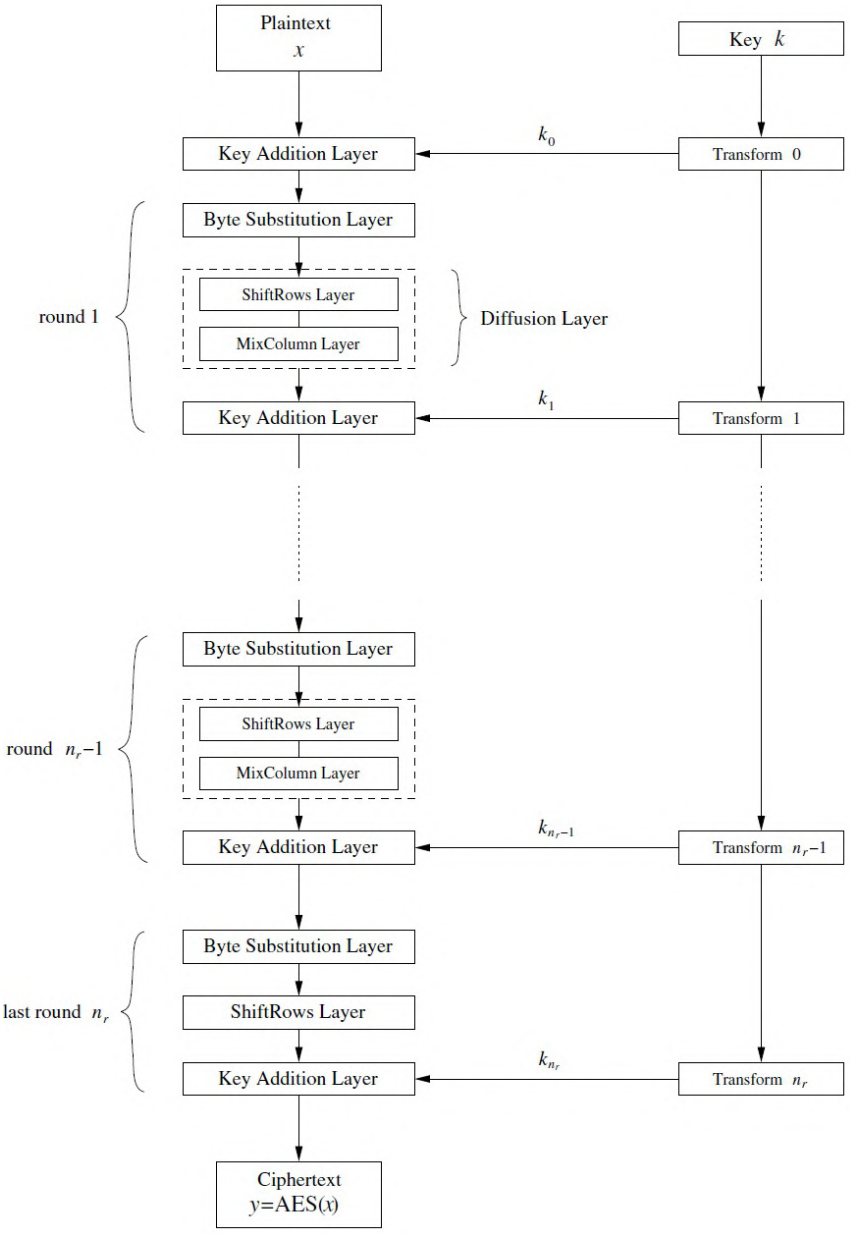
$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = \sum_{i=0}^{7} b_ix^i$$

No algoritmo AES, o número de rodadas depende do tamanho da chave, representado por N_k, sendo $N_k = 10$ quando a chave for de 128 bits, $N_k = 12$ para chave de 192 bits e $N_k = 14$ para chave criptográfica de 256 bits. Na Figura 15 é possível observar as funções que realizam as operações de substituição e difusão, além da função relativa a operação da chave, conforme (PAAR; PELZL, 2010). O texto claro sofre uma transformação inicial com a chave expandida da primeira rodada. Após essa transformação, a matriz de estado passa as funções de rodada subsequente. A primeira é a substituição de bytes não linear. Posteriormente, a matriz de estados passa por uma rede de difusão composta por *ShiftRows()*, onde as três últimas linhas são deslocadas ciclicamente, e também por *MixColumns()* onde os valores são misturados e formadas novas colunas. No fim da rodada, há adição da chave de rodada. Ao final de todas as rodadas, sendo que o número de rodadas depende do tamanho da chave criptográfica, será obtido o texto cifrado.

4.2.2 LEA

O LEA (*Lightweight Encryption Algorithm*) foi desenvolvido como uma solução de segurança para dispositivos embarcados que possuem recursos limitados como memória e velocidade de processamento. A maior parte das aplicações estão voltadas para microcontroladores operando em soluções IoT (HONG et al., 2014). O LEA utiliza valores de estado formados por palavras de 32 bits. As funções para cifrar e decifrar utilizam operações chamadas de ARX (*Addition, Rotation, XOR*), ou seja são operações que envolvem soma modular em 32 bits, rotação de bits e OU Exclusivo. Esse algoritmo foi desenvolvido com foco em processadores de 32 ou 64 bits, mas pode operar em microcontroladores de 8 ou 16 bits. O tamanho de bloco definido para esse

Figura 15 – Estrutura do AES.



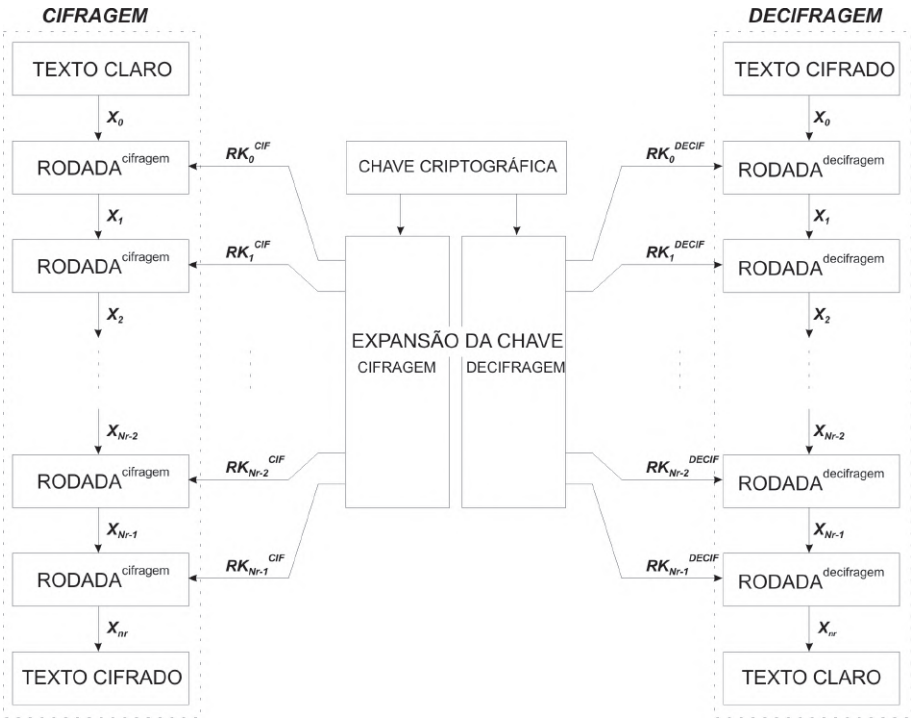
Fonte: (PAAR; PELZL, 2010)

algoritmo é de 128 bits e o tamanho de chave pode ser de 128, 192 ou 256 bits. O tamanho de chave a ser utilizado define o nome da cifra, ou seja, LEA-128, LEA-192 ou LEA-256.

Na Figura 16, é exibida a estrutura para cifragem e decifragem do LEA. A função responsável pela expansão da chave gera as chaves de rodada (RK_i^{COD}) para a cifragem e (RK_i^{DECOD}) para a decifragem, com $0 \leq i \leq N_{r-1}$, sendo N_r o número de rodadas selecionado. A cifragem efetua a cifragem de um texto claro de 128 bits em um texto cifrado também de 128 bits, após X_{Nr} operações. A decifragem segue o caminho inverso da cifragem (SUNG; BAE; SHIN, 2016).

As funções de rodada do LEA podem ser melhor entendidas observando a Figura 17.

Figura 16 – Estrutura do LEA.



Fonte: Adaptado de (SUNG; BAE; SHIN, 2016)

Nesta, observa-se que o texto claro é dividido em 4 palavras de 32 bits nomeados de $X[0...3]$, sendo posteriormente aplicadas as transformações ARX para $0 \leq i \leq N_{r-1}$. O valor de estado $X_{i+1}[0]$ corresponde a operação XOR entre o valor de estado $X_i[1]$ e a chave expandida para a rodada i e seu resultado é somado módulo em 32 bits com valor de estado $X_i[0]$ e deslocado 9 bits para a esquerda. Essas operações são executadas de forma semelhante para os demais valores de estado com alteração no deslocamento que será por 5 bits a direita no caso de $X_{i+1}[1]$ e por 3 a direita para $X_{i+1}[2]$. O estado $X_{i+1}[3]$ terá o mesmo valor do estado $X_i[0]$, ou seja, sem qualquer tipo de operação (SUNG; BAE; SHIN, 2016).

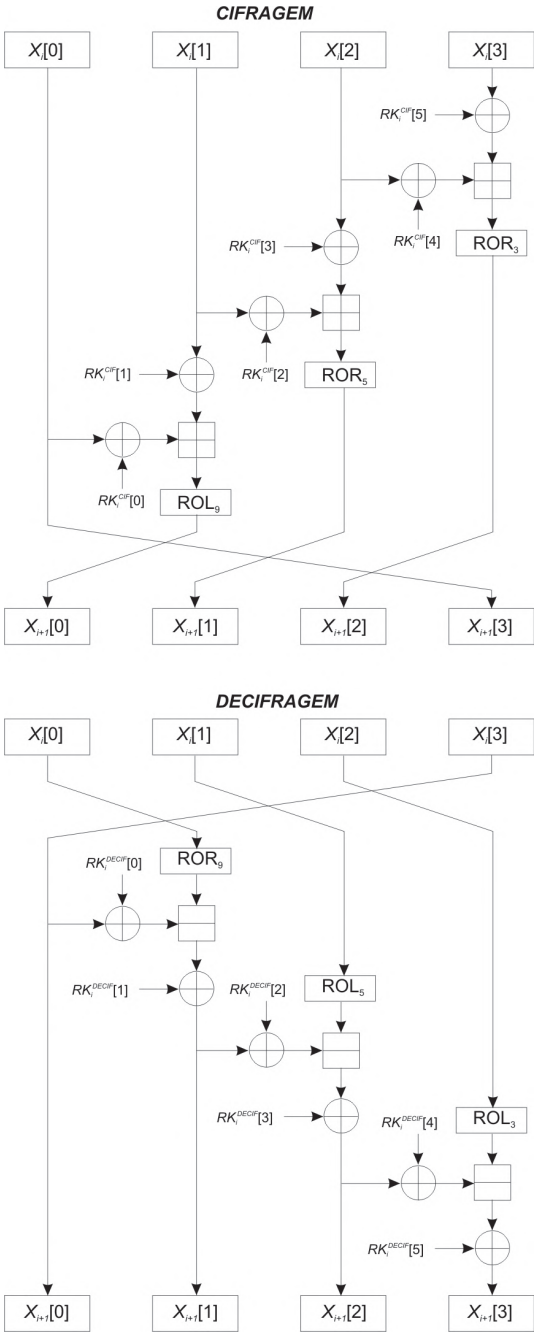
A quantidade de rodadas depende diretamente do tamanho da chave criptográfica, podendo ser de 24, 28 ou 32 rodadas caso a chave seja de 128, 192 ou 256 bits, respectivamente. O processo de cifragem tem como entrada o texto claro definido por $X = (X[0], X[1], X[2], X[3])$ e saída que será o texto cifrado com o mesmo tamanho da entrada $C = (C[0], C[1], C[2], C[3])$, ou seja, formado por 128 bits em 4 palavras de 32 bits. Pode-se definir o processo em 3 etapas: inicialização, rodadas de cifragem e finalização.

• *Inicialização*: é gerado um valor intermediário chamado X_0 com o texto claro e executada a expansão de chave para gerar r chaves de rodada, sendo r o valor de rodadas dependente do tamanho da chave.

• *Rodadas de cifragem*: a saída dos valores intermediários em cada rodada é obtida com base nas seguintes funções:

$$X_{i+1}[0] \leftarrow ROL_9(X_i[0] \oplus RK_i[0]) \boxplus (X_i[1] \oplus RK_i[1]),$$
$$X_{i+1}[1] \leftarrow ROR_5(X_i[1] \oplus RK_i[2]) \boxplus (X_i[2] \oplus RK_i[3]),$$

Figura 17 – Funções de rodada do LEA.



Fonte: Adaptado de (SUNG; BAE; SHIN, 2016)

$X_{i+1}[2] \leftarrow ROR_3(X_i[2] \oplus RK_i[4]) \boxplus (X_i[3] \oplus RK_i[5])$,

$X_{i+1}[3] \leftarrow X_i[0]$,

• *Finalização*: o texto cifrado é gerado com o valor obtido ao final da última rodada.

$C[0] \leftarrow X_r[0], C[1] \leftarrow X_r[1], C[2] \leftarrow X_r[2], C[3] \leftarrow X_r[3]$

A geração da expansão da chave criptográfica gera uma chave de 192 bits por rodada, chamada de RK_i. Ela é realizada com a utilização das seguintes constantes (HONG et al., 2014):

$\delta[0] = 0xc3efe9db$;

$\delta[1] = 0x44626b02$;

$\delta[2] = 0x79e27c8a$;
$\delta[3] = 0x78df30ec$;
$\delta[4] = 0x715ea49e$;
$\delta[5] = 0xc785da0a$;
$\delta[6] = 0xe04ef22a$;
$\delta[7] = 0xe5c40957$;

Para a expansão da chave criptográfica de 128 bits chamada K, tal que $K = (K[0], K[1], K[2], K[3])$, seleciona-se $T[i] = K[i]$ para $0 \leq i < 4$. O resultado será a chave de rodada (chave expandida) $RK_i = (RK_i[0], RK_i[1], RK_i[2], RK_i[3], RK_i[4], RK_i[5])$ para $0 \leq i < 24$, onde a operação $\boxplus$ se refere a soma modular, produzido da seguinte forma:

$T[0] \leftarrow ROL_1(T[0] \boxplus ROL_i(\delta[i \mod 4]))$,
$T[1] \leftarrow ROL_3(T[1] \boxplus ROL_{i+1}(\delta[i \mod 4]))$,
$T[2] \leftarrow ROL_6(T[2] \boxplus ROL_{i+2}(\delta[i \mod 4]))$,
$T[3] \leftarrow ROL_11(T[3] \boxplus ROL_{i+3}(\delta[i \mod 4]))$,
$RK_i \leftarrow (T[0], T[1], T[2], T[1], T[3], T[1])$,

Para a expansão da chave criptográfica de 192 bits, tal que $K = (K[0], K[1], K[2], K[3], K[4], K[5])$, seleciona-se $T[i] = K[i]$ para $0 \leq i < 6$. A chave de rodada será $RK_i = (RK_i[0], RK_i[1], RK_i[2], RK_i[3], RK_i[4], RK_i[5])$ para $0 \leq i < 28$, da seguinte forma:

$T[0] \leftarrow ROL_1(T[0] \boxplus ROL_i(\delta[i \mod 6]))$,
$T[1] \leftarrow ROL_3(T[1] \boxplus ROL_{i+1}(\delta[i \mod 6]))$,
$T[2] \leftarrow ROL_6(T[2] \boxplus ROL_{i+2}(\delta[i \mod 6]))$,
$T[3] \leftarrow ROL_{11}(T[3] \boxplus ROL_{i+3}(\delta[i \mod 6]))$,
$T[4] \leftarrow ROL_{13}(T[4] \boxplus ROL_{i+4}(\delta[i \mod 6]))$,
$T[5] \leftarrow ROL_{17}(T[5] \boxplus ROL_{i+5}(\delta[i \mod 6]))$,
$RK_i \leftarrow (T[0], T[1], T[2], T[3], T[4], T[5])$,

Quando a chave criptográfica for de 256 bits, tal que $K = (K[0], K[1], K[2], K[3], K[4], K[5], K[6], K[7])$, seleciona-se $T[i] = K[i]$ para $0 \leq i < 8$. A chave de rodada será $RK_i = (RK_i[0], RK_i[1], RK_i[2], RK_i[3], RK_i[4], RK_i[5])$ para $0 \leq i < 32$, da seguinte forma:

$T[6i \mod 8] \leftarrow ROL_1(T[6i \mod 8] \boxplus ROL_i(\delta[i \mod 8]))$,
$T[6i+1 \mod 8] \leftarrow ROL_3(T[6i+1 \mod 8] \boxplus ROL_{i+1}(\delta[i \mod 8]))$,
$T[6i+2 \mod 8] \leftarrow ROL_6(T[6i+2 \mod 8] \boxplus ROL_{i+2}(\delta[i \mod 8]))$,
$T[6i+3 \mod 8] \leftarrow ROL_{11}(T[6i+3 \mod 8] \boxplus ROL_{i+3}(\delta[i \mod 8]))$,
$T[6i+4 \mod 8] \leftarrow ROL_{13}(T[6i+4 \mod 8] \boxplus ROL_{i+4}(\delta[i \mod 8]))$,
$T[6i+5 \mod 8] \leftarrow ROL_{17}(T[6i+5 \mod 8] \boxplus ROL_{i+5}(\delta[i \mod 8]))$,
$RK_i \leftarrow (T[6i \mod 8], T[6i+1 \mod 8], T[6i+2 \mod 8], T[6i+3 \mod 8], T[6i+4 \mod 8], T[6i+5 \mod 8])$,

Para a decifragem o processo será o inverso com alteração nas adições modulares que serão substituídas por subtrações modular em 32 bits (SUNG; BAE; SHIN, 2016), conforme Figura 17.

4.2.3 ASCON

O ASCON foi desenvolvido em 2014 por pesquisadores da Universidade de Tecnologia de Graz na Austria, Infineon Technologies, Lamarr Security Research e Universidade Radboud da Holanda. O ASCON é uma suíte de cifras que provê cifragem autenticada com dados associados (AEAD - *authenticated Encryption with associated Data*) com funcionalidade *hash*. Ele foi a primeira escolha na competição CAESAR (*Competition for Authenticated Encryption: Security, Applicability, and Robustness*) em 2019 para o Caso 1, referente a aplicações *lightweight* dimensionados para dispositivos de baixo desempenho e em 2023 o NIST publicou o (TURAN et al., 2023) com a descrição dos critérios de avaliação e o processo para selecionar criptografia autenticada e esquemas de *hash* adequados para aplicações em ambientes restritos, onde o ASCON foi escolhido. A suite do ASCON é composta das seguintes aplicações possíveis (DOBRAUNIG et al., 2019):

- Cifragem autenticada: ASCON-128 e ASCON-128a
- Hash: ASCON-HASH e ASCON-XOF
- Função pseudoaleatória: ASCON-PRF

A estrutura do ASCON foi desenvolvida com base na construção esponja. As funções esponja são baseadas em uma quantidade fixa de permutações e regra de preenchimento. A abordagem que levou à criação do método esponja está relacionada à necessidade de efetuar uma função *hash* com tamanho variável de entrada e saída com tamanho determinado (PEETERS; BERTONI; DAEMEN, 2011).

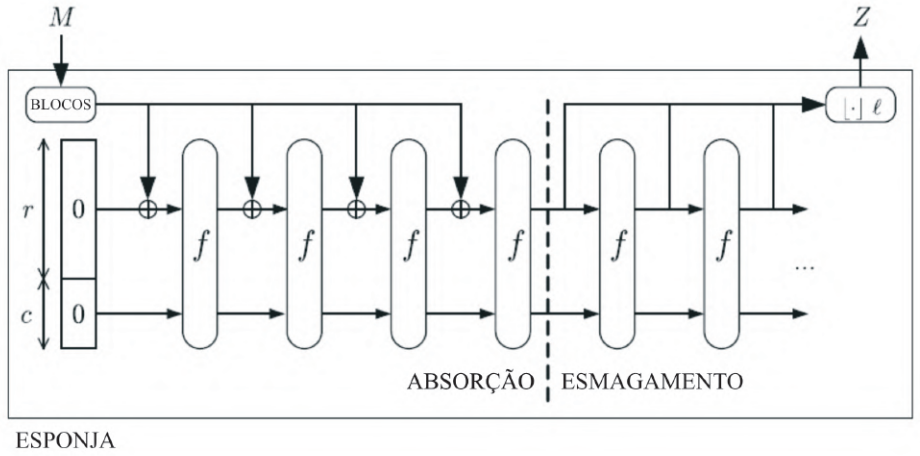
4.2.3.1 Construção da Esponja

A Figura 18 ilustra a formação da esponja com a mensagem de entrada M e a função que retorna a saída Z. A construção da esponja é realizada em estados de $b = r + c$ bits, em que b está relacionada como o número fixo de bits da permutação e também é chamado de *largura*. O valor de r é chamado de *razão* e o valor de c, *capacidade*. Durante a inicialização, todos os bits dos estados são inicializados como zero. A mensagem de entrada é preenchida e dividida em blocos de r bits. Após a inicialização, a construção da esponja é dividida em duas fases (PEETERS; BERTONI; DAEMEN, 2011):

- Fase de absorção: os r bits dos blocos de entrada passam por uma operação XOR com os primeiros r bits do estado, intercalando com a aplicação da função f. Quando todos os blocos da mensagem forem processados, será iniciada a próxima fase chamada de *esmagamento*.
- Fase de esmagamento: nessa fase os primeiros r bits do estado retornam como blocos de saída, intercalando com aplicação da função f, até que seja obtida a capacidade c definida pelo projetista da aplicação.

Os últimos c bits de estado nunca serão afetados diretamente pelos blocos de entrada e nunca serão saída durante a fase de esmagamento (PEETERS; BERTONI; DAEMEN, 2011).

Figura 18 – Construção esponja com $Z = esponja[f, blocos, r](M, l)$.



Fonte: Adaptado de (PEETERS; BERTONI; DAEMEN, 2011)

4.2.3.2 Cifragem Autenticada

O modelo de cifragem autenticada do ASCON é parametrizado para operar com chave criptográfica $k \leq 160$ bits, tamanho de bloco r e número de rodadas das funções de permutação definidas como a e b, tal que o algoritmo de cifragem será $\varepsilon_{k,r,a,b}$ e o algoritmo de decifragem $\mathcal{D}_{k,r,a,b}$. O processo de entrada recebe como entrada a chave K com k bits, um *nonce* (valor numérico utilizado na cifragem que pode ser público) N com 128 bits, dados associados A de tamanho arbitrário e um texto claro P também de valor arbitrário. A saída do algoritmo será uma cifra C com o mesmo tamanho do texto claro P mais um *tag*, T, com 128 bits resultando em (DOBRAUNIG et al., 2019):

$$\varepsilon_{k,r,a,b}(K, N, A, P) = (C, T).$$

O processo de verificação e decifragem $\mathcal{D}_{k,r,a,b}$, recebe como entrada a chave K, *nonce* N, dados associados A, texto cifrado C e *tag* T. A saída será o texto claro decifrado, caso a verificação do *tag* seja verdadeira, ou um erro $\perp$, se a verificação for negativa:

$$\mathcal{D}_{k,r,a,b}(K, N, A, C, T) \in \{P, \perp\}.$$

Os parâmetros recomendados conforme (DOBRAUNIG et al., 2019) estão listados na Tabela 4.

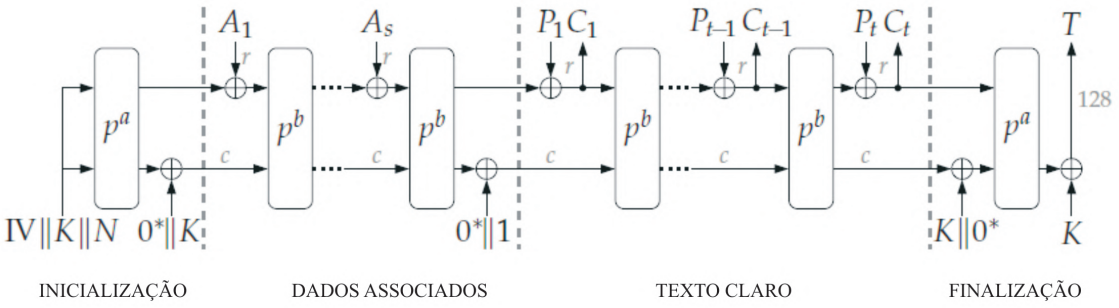
O modo de operação do ASCON está demonstrado na Figura 19 e sua explicação consta no detalhamento subsequente.

Tabela 4 – Parâmetros recomendados para os esquemas de cifragem autenticada.

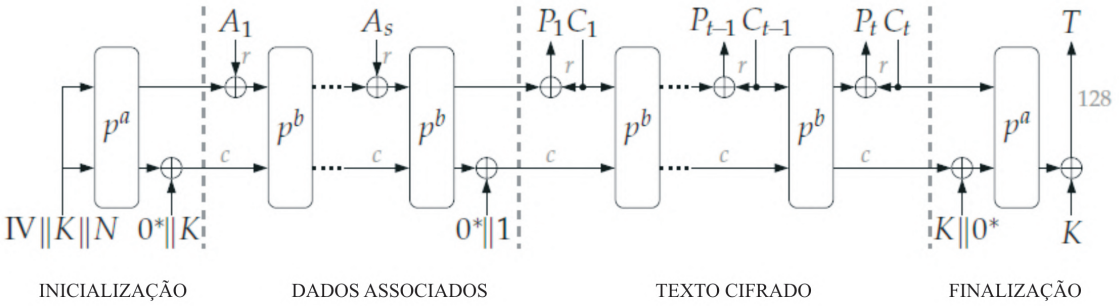
Nome	Algoritmo	Tamanho em bits				Rodadas	
		Chave	Nonce	Tag	Bloco	p^a	p^b
ASCON-128	$\varepsilon, \mathcal{D}_{128,64,12,6}$	128	128	128	64	12	6
ASCON-128a	$\varepsilon, \mathcal{D}_{128,128,12,8}$	128	128	128	128	12	8

Fonte: Adaptado de (DOBRAUNIG et al., 2019).

Figura 19 – Modo de operação do ASCON



a) Cifragem



b) Decifragem

Fonte: Adaptado de (DOBRAUNIG et al., 2019)

4.2.3.3 Valores de Estado

O ASCON trabalha com estados S de 320 bits que são atualizados com permutações p^a e p^b, sendo a e b valores para a quantidade de rodadas de permutação. Esse estado é dividido em uma fração externa nomeada S_r de r bits e outra interna, S_c, com c bits, onde a razão r e capacidade $c = 320 - r$ dependem de qual algoritmo ASCON será utilizado. Para o entendimento das rodadas de transformação, o estado de 320 bits S será considerado como uma junção de 5 registradores de 64 bits cada:

$$S = S_r \| S_c = x_0 \| x_1 \| x_2 \| x_3 \| x_4, \text{ onde } \| \text{ significa concatenação.}$$

Deve-se interpretar S como uma matriz de bytes, sendo iniciado pelo byte mais significativo de x_0 como byte 0 e finalizando com o byte menos significativo de x_4 como byte 39 (DOBRAUNIG et al., 2019).

4.2.3.4 Inicialização

O estado inicial do ASCON é formado pela chave secreta K de k bits e pelo *nonce* N de 128 bits, bem como por um vetor de inicialização especificado pelo algoritmo que se pretende utilizar.

$IV_{k,r,a,b} \leftarrow k\|r\|a\|b\|0^{160-k} = 80400c0600000000$ para ASCON-128

$IV_{k,r,a,b} \leftarrow k\|r\|a\|b\|0^{160-k} = 80800c0800000000$ para ASCON-128a

$S \leftarrow IV_{k,r,a,b}\|K\|N$

Na inicialização são efetuadas a rodadas de permutação p aplicadas ao estado inicial, seguido por uma operação XOR com a chave criptográfica K(DOBRAUNIG et al., 2019):

$S \leftarrow p^a(S) \oplus (0^{320-k}\|K)$

4.2.3.5 Dados Associados

O ASCON processa os dados associados A em blocos de r bits e acrescenta um bit 1 e uma pequena quantidade de zeros aos dados de A para obter múltiplo de r bits e juntar eles em s blocos de r bits, $A_1\|...\|A_s$. Caso A seja vazio, não haverá acréscimo de bits e será aplicado $s = 0$ (DOBRAUNIG et al., 2019).

4.2.3.6 Processamento da Cifragem e Decifragem

Para a cifragem, o texto claro P é dividido em blocos de r bits e, em cada iteração, é realizada uma operação XOR entre um bloco do texto claro e os primeiros r bits S_r do estado interno S, seguido da extração de um bloco cifrado C_i. O estado interno S é transformado por permutações p^b usando b rodadas, com exceção do último bloco. O último bloco de texto cifrado é então truncado de forma que o texto cifrado possua o mesmo tamanho do texto claro (DOBRAUNIG et al., 2019).

4.2.3.7 Finalização

No processo de finalização, a chave criptográfica K passa por uma operação XOR com o valor do estado interno, sendo esse estado transformado por permutações p^a usando a rodadas. O valor da *tag* T será obtido dos últimos 128 bits do valor de estado em operação XOR com os últimos 128 bits da chave K como observado na Figura 19 (DOBRAUNIG et al., 2019).

$S \leftarrow p^a(S \oplus (0^r\|K\|0^{c-k}))$

$T \leftarrow \lceil S \rceil^{128} \oplus \lceil K \rceil^{128}$

O NIST selecionou em 2023 o ASCON como padrão de cifragem *lightweight* para ser utilizado em diversas tipo de aplicações, principalmente para IoT.

4.3 CONSIDERAÇÕES

A cifragem simétrica, mais precisamente a cifragem com preservação de formato, mostra-se como uma possível solução por manter o mesmo formato de dados do protocolo e possuir forte resistência a ataques, mesmo com número pequeno de símbolos. Além disso, a utilização de novas cifras de blocos desenvolvidas para diversas aplicações *lightweight*, servem como excelentes funções pseudoaleatórias, garantindo forte segurança com excelente desempenho.

5 DESENVOLVIMENTO

Este capítulo aborda o desenvolvimento e análise de resultados de um sistema proposto para capturar os sinais ADS-B OUT transmitidos pelas aeronaves e que realiza a cifragem FPE utilizando quatro variações de funções pseudoaleatórias, baseadas nas cifras de bloco AES (DWORKIN et al., 2001), LEA (HONG et al., 2014), ASCON e ASCON-PRF (DOBRAUNIG et al., 2019).

5.1 SISTEMA DESENVOLVIDO

O motivo para utilizar dispositivos embarcados está relacionado à adaptação da solução de criptografia em *transponder* já instalados em aeronaves, sem que seja necessário substituir o equipamento por inteiro. O algoritmo FPE/FF1 e as cifras de bloco utilizadas como funções pseudoaleatórias citadas nessa pesquisa podem ser aplicadas em soluções desenvolvidas em FPGA (*field-programmable gate array*) ou em microcontroladores (AGBEYIBOR et al., 2014), devido a sua estrutura. Nesse trabalho de pesquisa, o autor desenvolveu o hardware, firmware e software para avaliação das cifras de bloco em um ambiente de preservação de formato. Optou-se por trabalhar com a família de microntroladores de 32 bits LPC17xx, mais precisamente o LPC1768 (ARM CORTEX M3) (NXP SEMICONDUCTORS, 2011), sendo esse dispositivo disponível para testes no laboratório de pesquisa e que apresentava as características de baixo custo e baixo tempo de desenvolvimento pretendidas, ainda mais com a utilização da IDE CooCox CoIDE e compilador GCC (*GNU Compiler*). A arquitetura de 32 bits pode ser aplicada a todas as cifras de blocos presentes nesse trabalho, principalmente com LEA (SEO et al., 2016).

5.1.1 Modelo Proposto

A Figura 20 exibe o modelo para recepção, tratamento e cifragem dos pacotes de dados ADS-B. O sistema é composto de um dispositivo RTL-SDR (*Realtek software-defined radio*), responsável por receber os dados ADS-B transmitidos pelas aeronaves, modulado em 1090 MHz, e em conjunto com o software RTL1090 essas informações são extraídas em palavras no formato hexadecimal. Com a recepção desses dados, o hardware desenvolvido realiza a cifragem e retransmite a informação cifrada para análise através do software analisador.

5.1.1.1 Radio Definido por Software

Um sistema de rádio em que as funções de camada física e todo o processamento de sinal é realizado por definições via software, chama-se de rádio definido por software ou SDR (*software defined radio*). A origem dos rádios definidos por software está no ambiente militar, sendo migrado para o uso no ambiente civil (JAHNAVI et al., 2016). A flexibilidade e o baixo custo desses dispositivos os tornaram muito utilizados atualmente.

Figura 20 – Plataforma do modelo proposto.

Fonte: O Autor (2023).

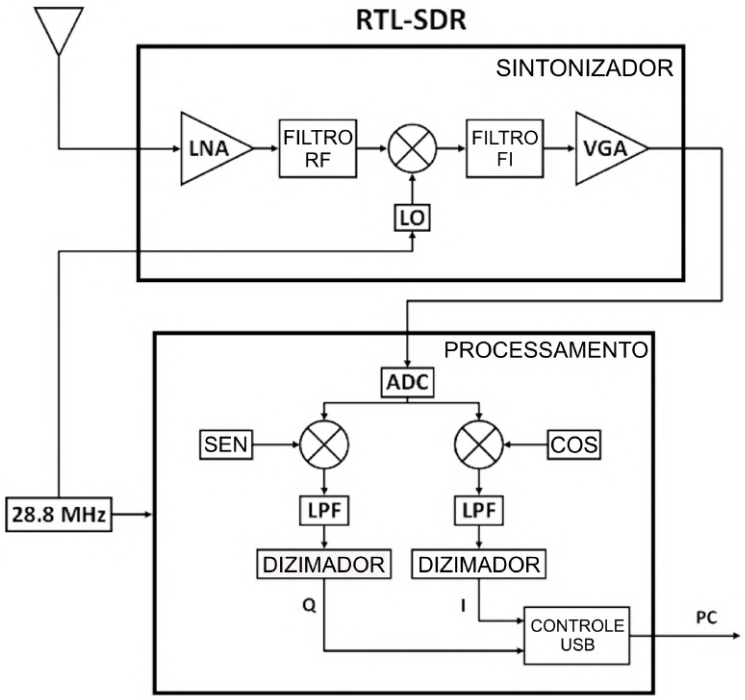
5.1.1.1.1 RTL-SDR

O dispositivo RTL-SDR é um rádio definido por software via USB muito utilizado atualmente, DVB-T (*Digital Video Broadcast-Terrestrial*) (JAHNAVI et al., 2016). O dispositivo utiliza o circuito integrado RTL2832U, (REALTEK SEMICONDUCTOR CORP., 2010), o qual possui as seguintes especificações técnicas (MOHAMED et al., 2016):

- Largura de banda: 2,4 MHz;
- Faixa de Frequência: 500kHz - 1766 MHz;
- Impedância de entrada: 50 Ohms.

Pode-se observar pela Figura 21, que o RTL-SDR é composto por um sintonizador RF que contém um amplificador de baixo ruído ou LNA, (do inglês *low noise amplifier*), seguido por um filtro de rejeição de imagem, além do misturador que converte o sinal RF em FI (frequência intermediária). O dispositivo utiliza um sintonizador RT820 e um RTL2832U. Para uma determinada frequência central, o sintonizador produz uma frequência intermediária entre 4 e 6 MHz. Esse sinal de IF é enviado para um conversor analógico para digital do RTL2832U onde o mesmo é amostrado em 28 MHz. O sinal resultante é digitalmente misturado com um sinal de 5 MHz gerado localmente e misturado em fase e quadratura. Os sinais misturados passam por um filtro passa-baixa com frequência de corte de 1,2 MHz e reamostrados para uma taxa configurável até 3,2 MS/s. O sinal resultante é digitalizado e enviado via USB para o computador (BNILAM et al., 2019).

Figura 21 – Diagrama em bloco do RTL-SDR.



Fonte: Adaptado de (BNILAM et al., 2019)

5.1.1.2 Software RTL1090

A aplicação RTL1090, desenvolvida pela empresa alemã Jetvision, roda em plataformas X86 e sistema operacional Windows e é capaz de processar dados ADS-B e modo S, quando em conjunto com RTL-SDR que opere em 1090MHz (JETVISION, 2023). A Figura 22 ilustra a tela de trabalho do programa.

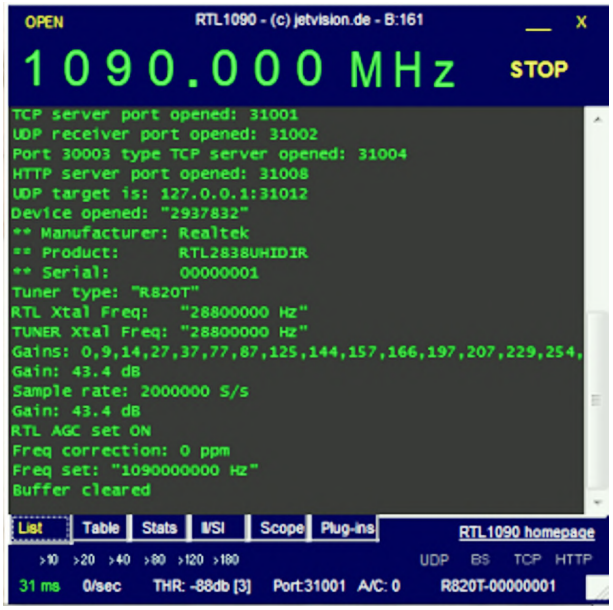
Esse programa recebe os dados oriundos do RTL-SDR e realiza a análise do preâmbulo das mensagens ADS-B para posteriormente realizar a extração dos pacotes de informação (KUMAR et al., 2021). Ele permite transmitir dados por meio de *socket* TCP/IP no formato ASCII (*American Standard Code for Information Interchange*), funcionando como servidor em uma configuração cliente-servidor.

5.1.1.3 Cifrador Desenvolvido

O cifrador foi concebido para realizar a cifragem dos dados ADS-B captados pelo RTL-SDR e extraídos pelo RTL1090. O formato dessas informações está descrito na Seção 2.6.1. Os dados são cifrados com FPE/FF1, utilizando 4 opções de cifras de blocos como função pseudoaletória:

- AES-128
- LEA-128

Figura 22 – Tela de trabalho do RTL1090.



Fonte: O Autor (2023).

- ASCON-128
- ASCON-PRF

Os dados cifrados são enviados para o computador, onde o programa Analisador FPE realiza o processamento e armazena os dados para posterior análise de entropia e desempenho. Foi realizada análise estatística das funções pseudoaleatórias com base na suíte de testes do NIST.

5.1.1.3.1 Suíte de Testes do NIST

O NIST desenvolveu uma suíte, ou seja, um pacote composto por 15 testes estatísticos com o objetivo testar a aleatoriedade de geradores de números aleatórios e geradores de números pseudoaleatórios (BASSHAM et al., 2010). Os 15 testes são:

- **Teste de Frequência (Monobit):** O objetivo deste teste é determinar se o número de "zeros" e "uns" em uma sequência é aproximadamente o mesmo que seria esperado para uma sequência verdadeiramente aleatória.
- **Teste de Frequência dentro de um Bloco:** O objetivo deste teste é determinar se a frequência de números "uns" em um bloco de M bits é aproximadamente M/2, como seria esperado sob uma suposição de aleatoriedade.
- **Teste de Execução:** O objetivo do teste de execução é determinar se o número de execuções de "uns" e "zeros" de vários comprimentos é o esperado para uma sequência aleatória,

ou seja, o número total de execuções na sequência, onde uma execução é uma sequência ininterrupta de bits idênticos.

- **Teste de Execução de mais longa de 1 em um Bloco:** O objetivo deste teste é determinar se a duração da sequência mais longa de números "uns" em uma sequência é consistente com a duração da sequência mais longa de "uns" que seria esperada em uma sequência aleatória. A irregularidade no tamanho do comprimento esperado da série mais longa de números "uns" implica que também existe uma irregularidade no comprimento esperado da série mais longa de "zeros".

- **Teste de Rank de matriz binária:** O objetivo deste teste é verificar se há dependência linear entre *substrings* de comprimento fixo em relação a sequência original.

- **Teste da Transformada Discreta de Fourier (Espectro):** O objetivo deste teste é detectar características periódicas na sequência testada. Isso indicaria que há um desvio da suposição de aleatoriedade.

- **Teste de Não-sobreposição de Padrão:** O objetivo deste teste é verificar se há ocorrências de um determinado padrão não periódico nas sequências geradas. Para este teste, uma janela de m bits é usada para procurar um padrão específico de m bits. Se o padrão não for encontrado, a janela desliza uma posição de bit. Se o padrão for encontrado, a janela será redefinida para o bit após o padrão encontrado e a busca será retomada.

- **Teste de Sobreposição de Padrão:** Esse teste funciona de forma muito semelhante ao Teste de Não-sobreposição de Padrão. A diferença consiste no fato de que quando o padrão é encontrado, a janela desliza apenas um bit antes de retomar a busca.

- **Estatística Universal de Maurer:** O objetivo do teste é detectar se a sequência pode ou não ser significativamente comprimida sem perda de informação, uma vez que a sequência significativamente compressível é considerada não-aleatória.

- **Teste de Complexidade Linear:** O objetivo deste teste é determinar se a sequência é complexa o suficiente para ser considerada aleatória, visto que sequências aleatórias são caracterizadas por LFSR longos e LFSR muito curto implica em não aleatoriedade.

- **Serial:** O objetivo deste teste é determinar se o número de ocorrências dos padrões de sobreposição de bits é aproximadamente o mesmo que seria esperado para uma sequência aleatória.

- **Teste de Entropia Aproximada:** O objetivo desse teste é comparar a frequência de blocos sobrepostos de dois comprimentos consecutivos com o resultado esperado para uma sequência aleatória.

- **Teste de Somas Cumulativas:** O propósito do teste é determinar se a soma cumulativa das sequências parciais que ocorrem na sequência testada é muito grande ou muito pequena em relação ao comportamento esperado dessa soma cumulativa para sequências aleatórias. Para certos tipos de sequências não-aleatórias, as excursões deste passeio aleatório a partir de zero serão grandes.

- **Teste de Excursões Aleatórias:** Esse teste consiste em uma sequência de passos de comprimento unitário dados aleatoriamente que começam e retornam à origem. O objetivo deste teste é determinar se o número de visitas a um determinado estado dentro de um ciclo se desvia do que seria esperado para uma sequência aleatória. Este teste é na verdade uma série de oito testes.

- **Teste de Variante para Excursões Aleatórias:** O objetivo deste teste é detectar desvios do número esperado de visitas a vários estados no passeio aleatório. Este teste é na verdade uma série de dezoito testes.

Conforme (BASSHAM et al., 2010) para que seja possível executar todos os testes do pacote é necessária uma massa de dados acima de 10^6 bits para cada *bitstreaming*. Dessa forma, foram aplicados dados ADS-B aleatórios nas cifras de blocos especificadas para posterior avaliação das funções pseudoaleatórias na suíte do NIST. A massa de informação extraída de cada cifra foi composta de 10 pacotes de 1.000.000 bits cada, totalizando 10^7 bits por cifra. A suíte é disponibilizada na página do NIST na internet por meio do endereço eletrônico <https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software>, acessado em 01 de julho de 2023. A tabela 5 exibe os resultados obtidos para cada um dos testes da suíte do NIST e compara esses resultados entre as cifras utilizadas como funções pseudoaleatórias.

Tabela 5 – Resultado obtido com a suíte do NIST para cada cifra.

RESULTADO DA UNIFORMIDADE DOS VALORES-P E A PROPORÇÃO DAS SEQUÊNCIAS PROCESSADAS												
TESTE ESTATÍSTICO	AES			LEA			ASCON			ASCON-PRF		
	VALOR-P	*PROPORÇÃO*	*RESULTADO*	*VALOR-P*	*PROPORÇÃO*	*RESULTADO*	*VALOR-P*	*PROPORÇÃO*	*RESULTADO*	*VALOR-P*	*PROPORÇÃO*	*RESULTADO*
Frequency	0,066882	9/10	Aleatório	0,991468	10/10	Aleatório	0,350485	10/10	Aleatório	0,122325	10/10	Aleatório
BlockFrequency	0,739918	10/10	Aleatório	0,213309	9/10	Aleatório	0,122325	10/10	Aleatório	0,002043	8/10	Aleatório
CumulativeSums	0,637032	9/10	Aleatório	0,7227795	10/10	Aleatório	0,2086835	10/10	Aleatório	0,534146	10/10	Aleatório
Runs	0,739918	10/10	Aleatório	0,911413	10/10	Aleatório	0,739918	10/10	Aleatório	0,739918	10/10	Aleatório
LongestRun	0,534146	10/10	Aleatório	0,350485	10/10	Aleatório	0,350485	10/10	Aleatório	0,350485	10/10	Aleatório
Rank	0,350485	10/10	Aleatório	0,122325	10/10	Aleatório	0,534146	10/10	Aleatório	0,350485	10/10	Aleatório
FFT	0,911413	10/10	Aleatório	0,739918	10/10	Aleatório	0,122325	10/10	Aleatório	0,350485	10/10	Aleatório
NonOverlappingTemplate	0,486213	10/10	Aleatório	0,476765	9/10	Aleatório	0,493612	10/10	Aleatório	0,516559	10/10	Aleatório
OverlappingTemplate	0,213309	10/10	Aleatório	0,213309	10/10	Aleatório	0,534146	10/10	Aleatório	0,739918	10/10	Aleatório
Universal	0,739918	10/10	Aleatório	0,991468	9/10	Aleatório	0,911413	10/10	Aleatório	0,739918	10/10	Aleatório
ApproximateEntropy	0,534146	10/10	Aleatório	0,213309	10/10	Aleatório	0,534146	10/10	Aleatório	0,122325	9/10	Aleatório
AleatórioExcursions	—	5/5	Aleatório	—	7/7	Aleatório	—	6/6	Aleatório	—	5/5	Aleatório
AleatórioExcursionsVariant	—	5/5	Aleatório	—	6/7	Aleatório	—	6/6	Aleatório	—	5/5	Aleatório
Serial	0,4766135	10/10	Aleatório	0,4034	10/10	Aleatório	0,4034	10/10	Aleatório	0,000319	5/10	Não-aleatório
LinearComplexity	0,739918	10/10	Aleatório	0,739918	9/10	Aleatório	0,035174	10/10	Aleatório	0,350485	10/10	Aleatório

Fonte: O Autor (2023).

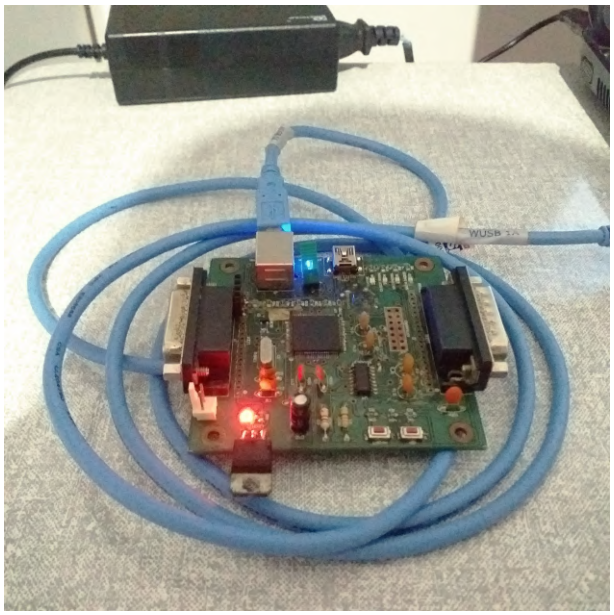
As funções pseudoaleatórias geradas pelas cifras de bloco apresentadas passsaram em todos os testes do NIST, com exceção apenas do ASCON-PRF que não passou em um único teste, correlação serial. Dessa forma, considera-se satisfatório o desempenho para aplicação em um ambiente de preservação de formato para dados ADS-B.

5.1.1.3.2 Hardware

A implementação do cifrador foi realizada com base no microcontrolador ARM Cortex M3 modelo LPC1768 da fabricante NXP semiconductors. Esse componente opera com frequência de *clock* de até 100 MHz, possui 512 kB de memória *flash* e 64 kB de RAM. Além disso, há diversos periféricos, tais como: controlador DMA de 8 canais, 4 UART, 2 interfaces CAN, 2 controladores SSP, conversor ADC de 12 bits e 8 canais, DAC de 10 bits, dentre outros (NXP SEMICONDUCTORS, 2011). Os dados extraídos são enviados para o cifrador via comunicação USB CDC a uma taxa de 115200 bps (*bits per second*) e a alimentação é provida pela própria porta USB.

O circuito desenvolvido possui duas portas USB sendo uma porta para programação e outra para comunicação com a aplicação e o esquema elétrico da placa contendo o microcontrolador está no Apêndice A. O LPC1768 possui internamente um *bootloader*, espaço de memória que possui as diretivas para realização da programação na memória *flash*. O programa Flash Magic desenvolvido pela fabricante NXP Semiconductors, comunica-se com o *bootloader* do LPC1768 através da interface ISP (*in-system programming*) utilizando as portas P0.2 e P0.3 configuradas como TX e RX, respectivamente, da UART0 (*universal asynchronous receiver/transmitter*). Para a realização da conversão USB para UART do LPC1768, foi utilizado o circuito integrado CP2102 da fabricante Silicon Labs (SILICON LABS, 2017). A Figura 23 exibe o circuito em funcionamento.

Figura 23 – Cifrador desenvolvido.



Fonte: O Autor (2023).

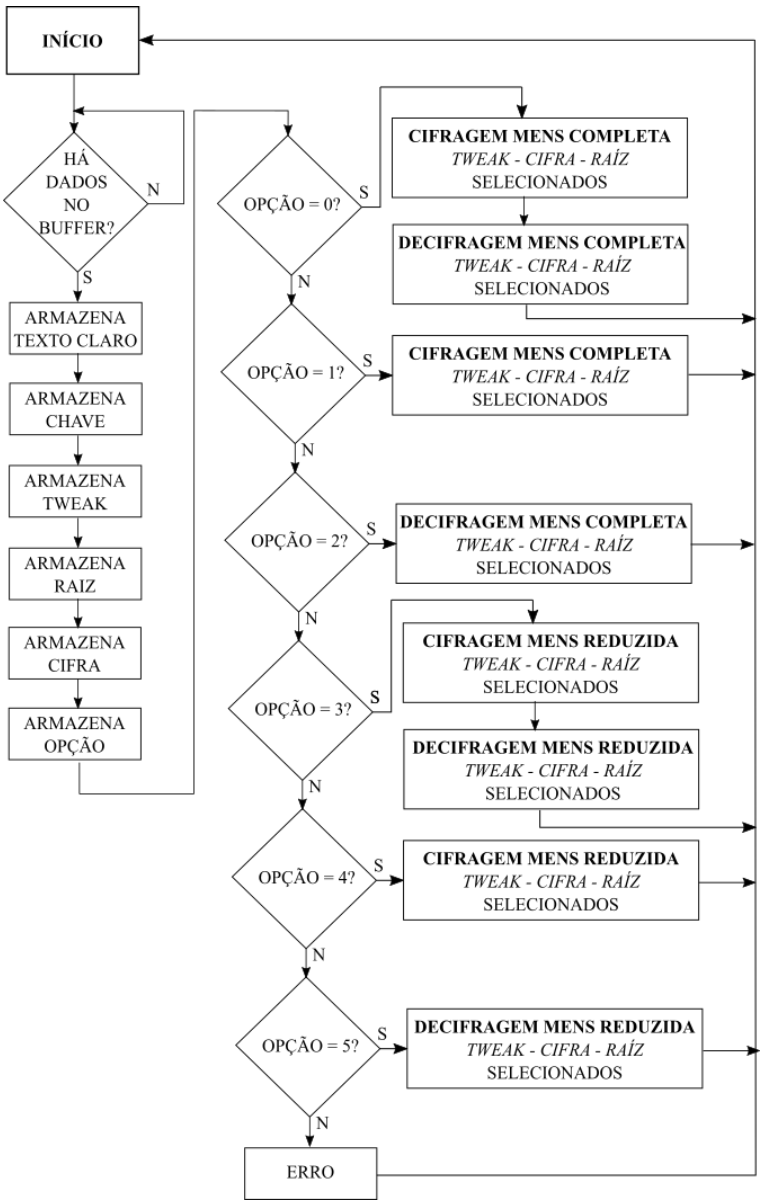
O LPC1768 possui interface física integrada para comunicação USB através das portas P0.29 e P0.30. Essa interface é utilizada para realizar a comunicação entre o cifrador e aplicação que está no computador. A classe implementada para essa comunicação foi a CDC (*commu-*

nication device class) que simula uma comunicação serial entre os dispositivos. No circuito desenvolvido, foi inserido um LED para informação de estado da USB acionado pela porta P2.9.

5.1.1.3.3 Firmware

O código foi desenvolvido em linguagem C e sua função fundamental é realizar a cifragem dos pacotes de dados ADS-B. A utilização de FPGA (*field-programmable gate array*) foi levada em consideração inicialmente, mas por conta do tempo de desenvolvimento, tempo para aquisição de periféricos e custo, optou-se pela utilização de microcontroladores. A Figura 24 exibe o fluxograma o firmware desenvolvido.

Figura 24 – Fluxograma do firmware desenvolvido.



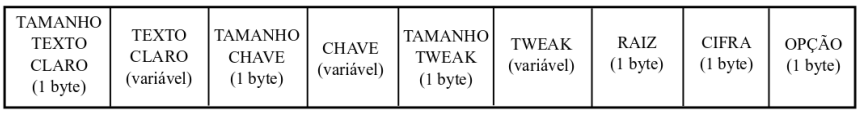
Fonte: O Autor (2023).

Observa-se pelo fluxograma do algoritmo que o sistema aguarda a chegada do pacote de dados. O formato desse pacote está ilustrado na Figura 25. Após o recebimento do pacote, o firmware armazena os valores dos campos relativos aos dados a serem cifrados e dos dados de configuração. Posteriormente são processados os dados de configuração que determinam qual a cifra e a raíz a ser utilizada, bem como a opção do modo de funcionamento:

- Cifrar: efetua a cifragem FPE/FF1 com a cifra de bloco selecionada como função pseudoaleatória;
- Decifrar: efetua a decifragem FPE/FF1 com a cifra de bloco selecionada;
- Cifrar e Decifrar: efetua a cifragem e, logo após, efetua a decifragem FPE/FF1 com a cifra de bloco selecionada, com a finalidade de verificar se os algoritmos retornam o texto claro original sem erros.

Para cada uma das opções acima, é possível selecionar antecipadamente se o cabeçalho e o campo CRC (*cyclic redundancy check*) do pacote ADS-B fazem parte da mensagem a ser cifrada ou decifrada. O código desenvolvido conta com as quatro cifras de blocos tratadas nessa pesquisa, sendo selecionada, através do Software Analisador FPE, qual cifra deve ser utilizada para cada mensagem transmitida.

Figura 25 – Formato do pacote de dados.



Fonte: O Autor (2023).

As principais funções elaboradas são:

- *int FF1encrypt2 (char *T, int LENT, char *X, int LENX, char Modo, char Resultado[])*
- *int FF1decrypt2 (char *T, int LENT, char *X, int LENX, char Modo, char Resultado[])*
- *void prfAES (char *X, int LENX)*
- *void prfLEA (char *X, int LENX)*
- *void prfASCON (char *X, int LENX)*
- *void prfASCONPRF (char *X, int LENX)*

A função *FF1encrypt2* é responsável por efetuar a cifragem e a função *FF1decrypt2* é responsável por efetuar a decifragem FPE utilizando o parâmetro **Modo** para determinar qual será a cifra de bloco a ser utilizada como função pseudoaleatória. O parâmetro **T** é o ponteiro

para o vetor *TWEAK*, **LENT** é o valor correspondente ao tamanho em bytes do *TWEAK*, *X* é o ponteiro para a mensagem a ser cifrada ou decifrada, **LENX** é o tamanho da mensagem em bytes e o vetor **Resultado** contém o resultado da operação.

O processamento da função pseudoaleatória é realizada por meio das funções que chamam as cifras de bloco. Para o sistema desenvolvido há quatro possibilidades: AES, LEA, ASCON e ASCON-PRF, ou seja, *void prfAES (char *X, int LENX)*, *void prfLEA (char *X, int LENX)*, *void prfASCON (char *X, int LENX)* e *void prfASCONPRF (char *X, int LENX)* respectivamente. Para todas essas funções os parâmetros **X** e **LENX** relaciona-se ao ponteiro para o texto claro/cifrado e tamanho em bytes do texto claro/cifrado.

Após o processamento das informações o resultado é transmitido via USB para a aplicação ANALISADOR FPE.

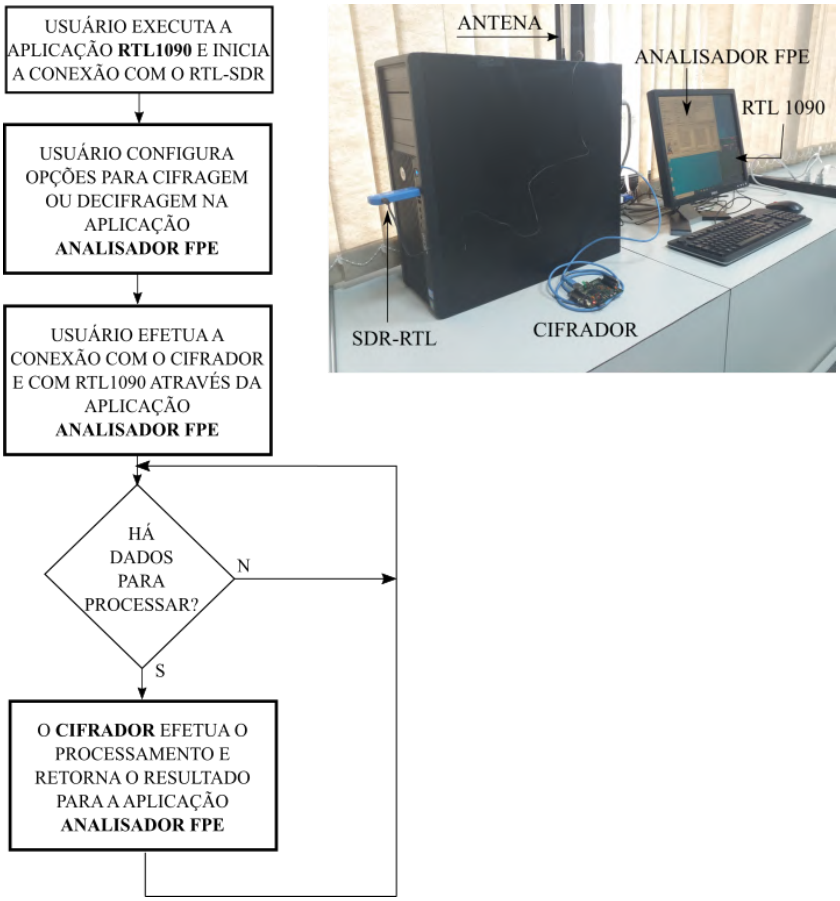
5.1.1.4 Software Analisador FPE

Foi desenvolvida uma aplicação para realizar a configuração do cifrador e para servir de interface com o usuário com a finalidade de obter os dados de texto claro e os dados cifrados. Esse programa foi desenvolvido em C# e se chama *Software Analisador FPE*. A comunicação com o cifrador é realizada por uma classe de porta serial emulando uma serial sobre USB. A conexão com o programa RTL1090 é realizada por um socket TCP, utilizando a configuração cliente-servidor, onde o RTL1090 é o servidor. A Figura 26 demonstra como é o funcionamento do sistema com a obtenção dos dados pelo SDR-RTL e as interligações com o software Analisador FPE desenvolvido, cifrador e RTL1090.

A Figura 27 exibe a tela de trabalho do programa. No campo superior esquerdo estão os campos correspondentes a configuração da comunicação com o cifrador e com o RTL1090. No lado superior direito, há um setor nomeado como “Estático”. Essa função realiza a cifragem ou decifragem da informação contida no *textbox* correspondente. No campo central estão as opções que permitem ao usuário configurar o cifrador para realizar cifragem, decifragem ou ambos, além de possibilitar a escolha da cifra de bloco desejada como função pseudoaleatória do FPE, a opção de raiz (símbolos), se haverá cifragem dos 112 bits da mensagem ADS-B ou apenas 80 bits (descarte do cabeçalho e CRC), utilização do TWEAK com o código ICAO da aeronave e a opção de salvar os dados cifrados em arquivo binário. O campo “Tempo Real” processa as informações recebidas da comunicação TCP/IP e exibe a mensagem de entrada, mensagem cifrada e tempo entre a recepção da mensagem pelo *socket* e a recepção da mensagem cifrada pela serial quando transmitida pelo cifrador. O campo “Debug” permite ao usuário saber a quantidade de bytes recebidos no último pacote ADS-B, o código ICAO de 24 bits da aeronave detectada e permite saber a entropia da informação de entrada e da mensagem cifrada.

Vale ressaltar que a opção por não cifrar o HEADER e o CRC faz com que o firmware insira o cabeçalho original do texto claro na mensagem cifrada e insere um novo CRC recalculado. Em relação ao TWEAK, o programa insere automaticamente o valor do código ICAO de 24 bits da aeronave como valor para *TWEAK* no momento da cifragem.

Figura 26 – Fluxograma do funcionamento do sistema.



Fonte: O Autor (2023).

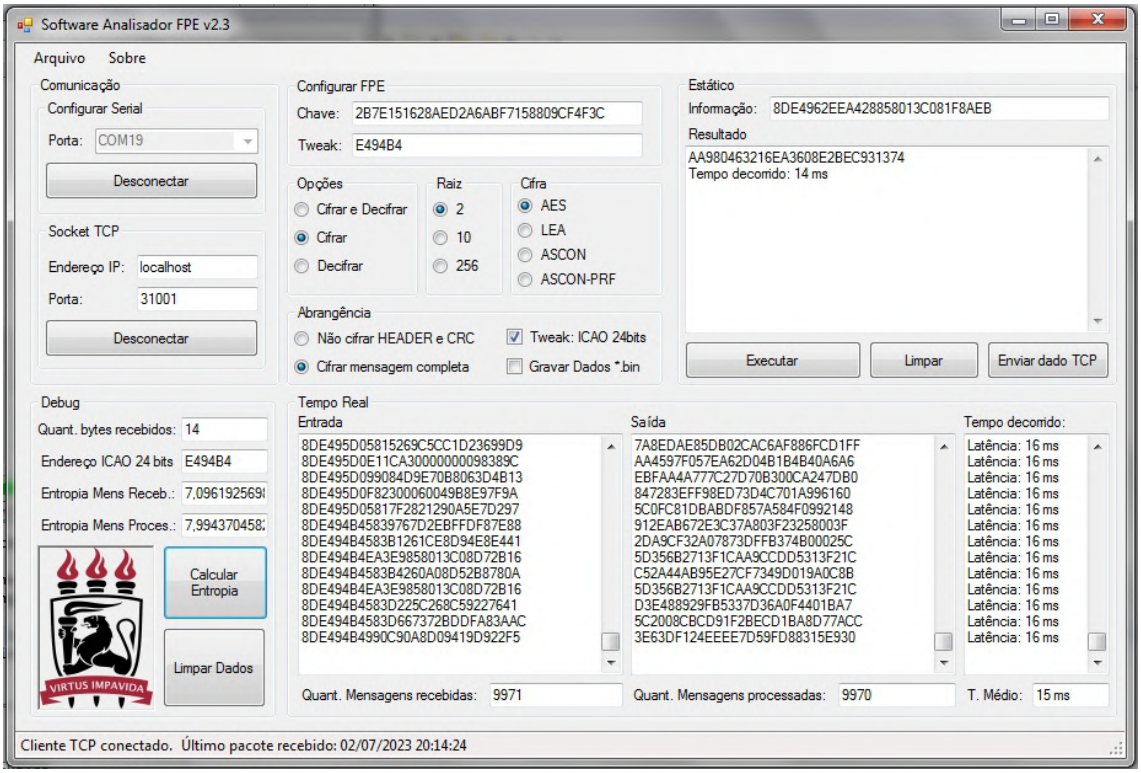
5.1.1.4.1 Opção Estático

Na configuração "Estático"o usuário insere a informação a ser cifrada, a chave criptográfica e, poderá também, inserir um *TWEAK*. A escolha da raiz deve estar condicionada ao alfabeto da informação e da chave. No *TextBox* correspondente, será exibida a cifragem, decifragem ou ambas, conforme selecionado, além do tempo de processamento que o microcontrolador levou para realizar a operação.

5.1.1.4.2 Opção Tempo Real

O programa recebe em tempo real todos os dados enviados pelo RTL1090 por meio do socket TCP, retransmite para o cifrador e recebe os dados cifrados. O formato dos dados recebidos e a dinâmica de cifragem dependerão das opções selecionadas pelo usuário na configuração do FPE. O programa calcula o tempo que o microcontrolador levou para realizar a operação selecionada e exibe ao lado do resultado da operação. Os dados recebidos e os dados processados são salvos em arquivos binários no diretório do executável da aplicação, quando a opção “Gravar Dados *.bin” é selecionada. O programa cria um arquivo para cada identificador ICAO de 24

Figura 27 – Tela de trabalho do Software Analisador FPE proposto.



Fonte: O Autor (2023).

bits para cada dia de operação e um arquivo com todas as aeronaves por dia, seguindo o seguinte formato como nomenclatura:

- Arquivo de **texto claro** (total por dia): "Texto Claro"+ "Nome da Cifra selecionada"+ "Data da geração".bin;

  Exemplo: Texto_Claro_AES_27052023.bin

- Arquivo de **texto claro** (total por aeronave e por dia): "Texto Claro"+ "Nome da Cifra selecionada"+ "ICAO 24 bits"+ "Data da geração".bin;

  Exemplo: Texto_Claro_AES_E4962E_27052023.bin

- Arquivo de **texto claro** com *TWEAK* (total por dia): "Texto Claro"+ "Nome da Cifra selecionada"+ "TWEAK"+ "Data da geração".bin;

  Exemplo: Texto_Claro_AES_TWEAK_27052023.bin

- Arquivo de **texto claro** com *TWEAK* (total por aeronave e por dia): "Texto Claro"+ "Nome da Cifra selecionada"+ "ICAO 24 bits"+ "TWEAK"+ "Data da geração".bin;

  Exemplo: Texto_Claro_AES_E4962E_TWEAK_27052023.bin

- Arquivo de **texto cifrado** (total por dia): "Cifrado"+ "Nome da Cifra selecionada"+ "Data da geração".bin;

Exemplo: Cifrado_AES_27052023.bin

- Arquivo de **texto cifrado** (total por aeronave e por dia): "Cifrado"+ "Nome da Cifra selecionada"+ "ICAO 24 bits"+ "Data da geração".bin;

  Exemplo: Cifrado_AES_E4962E_27052023.bin

- Arquivo de **texto cifrado** com *TWEAK* (total por dia): "Cifrado"+ "Nome da Cifra selecionada"+ "TWEAK"+ "Data da geração".bin;

  Exemplo: Cifrado_AES_TWEAK_27052023.bin

- Arquivo de **texto cifrado** com *TWEAK* (total por aeronave e por dia): "Cifrado"+ "Nome da Cifra selecionada"+ "ICAO 24 bits"+ "TWEAK"+ "Data da geração".bin;

  Exemplo: Cifrado_AES_E4962E_TWEAK_27052023.bin

O objetivo da criação desses arquivos está na posterior análise dos dados, tanto para os dados de texto claro, quanto para os dados cifrados.

5.1.1.5 Software Servidor ADS-B

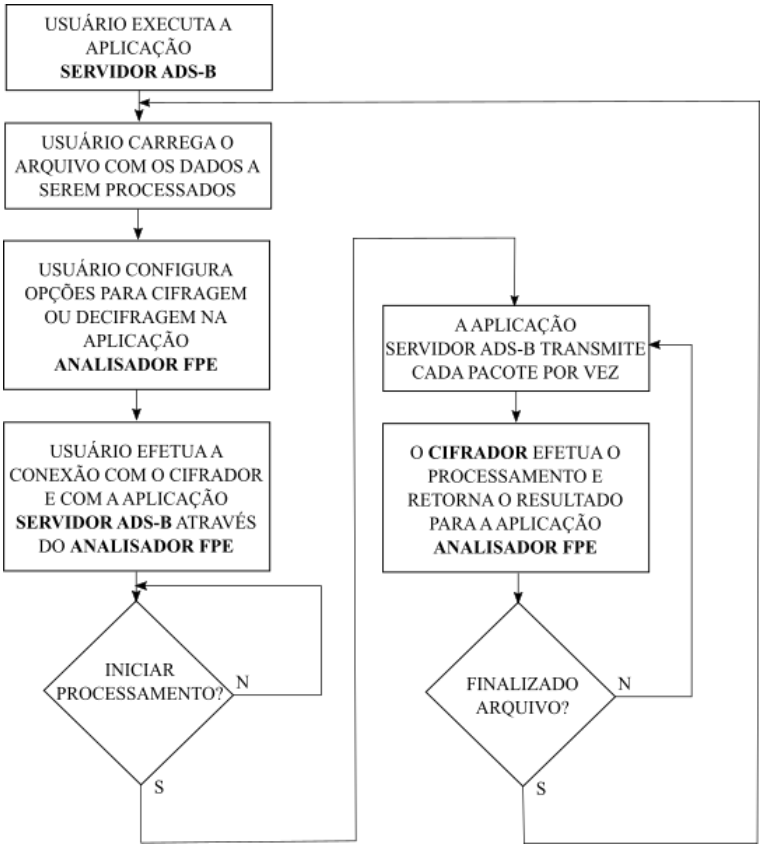
Durante o desenvolvimento da pesquisa, tornou-se necessário realizar o processamento dos dados ADS-B adquiridos das aeronaves por meio do RTL1090 e armazenados no computador. Dessa forma, como medida de disponibilidade e com base nos dados salvos pela aplicação anterior, foi desenvolvido um programa servidor em C# que possibilita transferir mensagem por mensagem ADS-B ao Software Analisador FPE. A Figura 28 exibe o modo de funcionamento com a utilização da aplicação Servidor ADS-B em detrimento ao RTL1090.

Como observado na Figura 29, através desse programa, o usuário seleciona o IP e a porta de comunicação e abre o *socket* para o cliente se conectar. Na opção “Abrir Arquivo”, ele seleciona o arquivo contendo as mensagens ADS-B capturadas pelo RTL1090 em conjunto com o Software Analisador FPE. No botão “Transmitir Arquivo” o programa inicia a transferência das mensagens mantendo o tempo entre mensagens de acordo com o selecionado em “Tempo envio”. Cada mensagem transferida é exibida no *textbox* principal. O usuário pode também enviar mensagens individuais através do botão “Enviar dado”. Por último, o programa pode criar outros arquivos, no diretório do executável, para cada código ICAO, ou seja, para um arquivo com diversas mensagens ADS-B de aeronaves distintas, o programa pode separar por aeronave as mensagens ADS-B que constam no arquivo principal. Essa funcionalidade foi criada para possibilitar análise de entropia em mensagens com pouca variação de informação.

5.2 RESULTADOS

Para análise de comportamento dos algoritmos de cifragem, foram utilizados dados reais transmitidos pelos *transponders* ADS-B de aeronaves que estavam em fase de pouso

Figura 28 – Fluxograma do funcionamento do sistema com o uso da aplicação Servidor ADS-B.



Fonte: O Autor (2023).

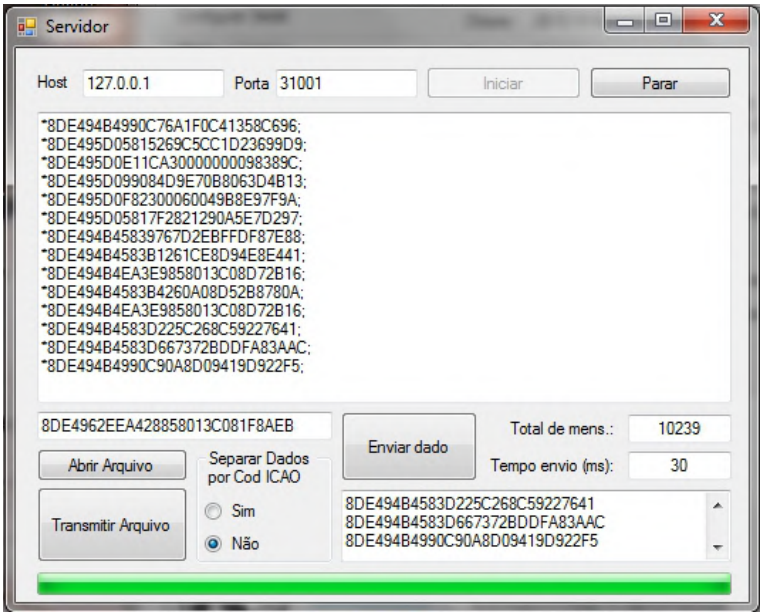
ou decolagem no aeroporto internacional do Recife/Guararapes - Gilberto Freyre, sediado em Recife/PE. Conforme (CIVIL, 2010) e (CIVIL, 2023), as fases de pouso e decolagem são fases críticas do voo, motivo pelo qual essa pesquisa focou na obtenção dos dados na área terminal do aeroporto. A plataforma de aquisição e cifragem dos dados foi instalada no Terceiro Centro Integrado de Defesa Aérea e Controle do Espaço Aéreo (CINDACTA III), conforme Figura 30, que fica distante aproximadamente 1 km do aeroporto, com visada direta.

Os dados adquiridos passaram por duas análises: entropia e desempenho da cifragem. A primeira análise tem por objetivo verificar o nível de incerteza imposta à informação, enquanto a segunda está relacionada ao tempo de processamento da cifragem/decifragem.

5.2.1 Análise de Entropia

O conceito de entropia, aplicado à Teoria da Informação, foi desenvolvido por Claude Elwood Shannon, que é a incerteza associado a uma variável aleatória. Ele adotou essa palavra por causa da sua semelhança formal com a definição de entropia de Boltzmann, utilizada na mecânica estatística. Além disso, ele também via a linguagem como um processo estocástico, ou seja, como um sistema governado por probabilidades que produz uma sequência de símbolos (HOFFSTEIN; PIPHER; SILVERMAN, 2014). Conforme Shannon, a entropia $H(X)$ de uma variável X, pode ser calculada da seguinte forma:

Figura 29 – Tela de trabalho do Software Servidor ADS-B.



Fonte: O Autor (2023).

Figura 30 – Local de instalação da plataforma.

Fonte: O Autor (2023).

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{5.1}$$

Nesta pesquisa, é levada em consideração a quantidade de informação que poderia ser obtida dos dados cifrados e, para efetuar os cálculos de entropia, utilizou-se o próprio programa Analisador FPE e também o programa ENT disponível em <https://www.fourmilab.ch/random/> (FINKE et al., 2013), acessado em 28 de maio de 2023. Essa ferramenta exibe a ocorrência de

cada byte do arquivo e calcula a entropia expressando o resultado em bits por byte. A metodologia para cômputo da entropia foi construída com base na cifragem sem qualquer valor como *TWEAK* e também com o uso do código ICAO de 24 bits nesse campo. Isso possibilitou analisar se a inserção de informação como *TWEAK* iria alterar significantemente o valor da entropia. Foram descritos 4 cenários para obtenção dos dados:

- Cenário 1: cifragem do pacote ADS-B completo, sem utilização de TWEAK na cifragem;
- Cenário 2: cifragem do pacote ADS-B completo, com utilização de TWEAK na cifragem;
- Cenário 3: cifragem do pacote ADS-B sem inserir o cabeçalho e recalculando o CRC, sem utilização de TWEAK na cifragem;
- Cenário 4: cifragem do pacote ADS-B sem inserir o cabeçalho e recalculando o CRC, com utilização de TWEAK na cifragem;

Para cada cenário, foram utilizadas as seguintes cifras de blocos como função pseudoaleatória no FPE: AES, LEA, ASCON e ASCON-PRF. A principal diferença entre esses cenários está na quantidade de dados cifrados em cada mensagem. Quando a mensagem completa é cifrada, utiliza-se 112 bits, ou seja, 14 bytes de informação. No entanto, quando o cabeçalho e o CRC não são cifrados, esse universo diminui para 80 bits ou 8 bytes. Em todos os testes, foram adotados os seguintes parâmetros como chave criptográfica e TWEAK, sendo que o valor adotado para chave criptográfica foi retirado do vetor de testes fornecido pelo NIST (NIST, 2023):

- Chave: 2B7E151628AED2A6ABF7158809CF4F3C;
- TWEAK: Valor da identificação da aeronave no formato ICAO 24 bits;

Foram adquiridas 10.208 mensagens ADS-B transmitidas por 55 aeronaves diferentes. Para a pesquisa, foram considerados os dados de todas as aeronaves em uma primeira análise e posteriormente selecionados os dados das 10 aeronaves que haviam transmitido mais informações. Essa distinção entre as análises se embasa na necessidade de avaliação da entropia quando há pequena variação de informação entre transmissões ADS-B. Dessa forma, com a separação dos dados por código ICAO, pode-se analisar informações que mantiveram os campos DF, CA e ICAO inalterados, ou seja, 32 bits entre os 112 ficaram fixos. As informações das aeronaves constam na Tabela 6. Os dados relativos a companhia aérea, modelo e registro foram obtidos por meio do site <https://www.planespotters.net/> com o uso da identificação das aeronaves no campo ICAO 24 bits.

Com base nos dados totais coletados, foram realizados cálculos de entropia para cada cifra de bloco por cenário. Na Tabela 8, constam os valores de entropia para cada cenário levando em consideração o quantitativo total de mensagens observadas. A Figura 31 exibe o gráfico com

Tabela 6 – As 10 Aeronaves que tiveram maior quantidade de dados ADS-B coletados.

COD ICAO	COMPANHIA	MODELO	REGISTRO	QUANT. MENS.
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

Fonte: O Autor (2023).

Tabela 7 – Entropia para cada cenário em relação a quantidade total de mensagens observadas.

CIFRA	CENÁRIO 1	CENÁRIO 2	CENÁRIO 3	CENÁRIO 4	QUANT. MENS.
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
Texto Claro	7,09332	7,09332	7,09332	7,09332	10208

Fonte: O Autor (2023).

a distribuição desses valores e a Figura 32 exibe de forma mais detalhada a diferença de entropia entre as cifras para cada cenário.

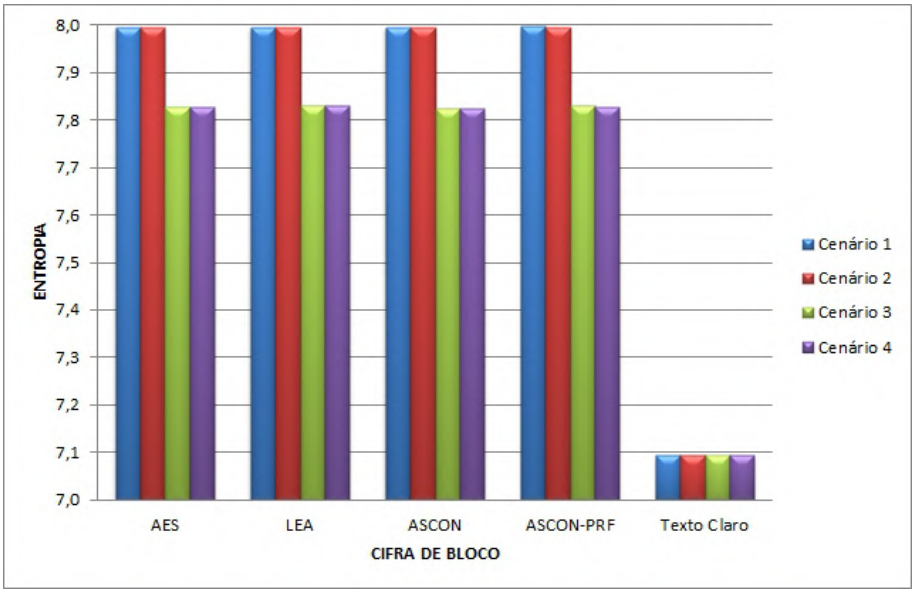
Os valores mais altos de entropia nos cenários 1 e 2 estão relacionados ao fato que há cifragem completa dos 112 bits de cada pacote de mensagem ADS-B, aumentando o número de variáveis pseudoaleatórias na composição da mensagem cifrada. Nos cenários 3 e 4, a quantidade de informação a ser cifrada reduz para 80 bits, pois o cabeçalho não é cifrado e o CRC é substituído por novo valor recalculado. Como o objetivo do dispositivo desenvolvido é servir como elemento de segurança criptográfica em *transponders* existentes, há a necessidade de avaliação do nível de entropia em mensagens que mantêm o código de identificação da aeronave (código ICAO de 24 bits) inalterado. As Tabelas 8, 9, 10 e 11 exibem os resultados obtidos para cada cenário das 10 aeronaves observadas.

Tabela 8 – Entropia de cada cifra para o cenário 1.

CENÁRIO 1 - Cifragem mensagem completa sem TWEAK												
CIFRA	IDENTIFICADOR ICAO 24 BITS										MÉDIA	DESV. PADRÃO
	E49A47	E498E8	E495B2	E49390	E4910B	E492CF	E495D0	E49165	E4953A	E4917D		
AES	**7,87399**	7,89808	7,91507	**7,86705**	7,94237	7,86830	7,84878	7,91820	7,88165	7,93189	7,89454	0,02967
LEA	7,86841	7,90093	**7,92048**	7,86356	**7,94728**	7,85114	7,86109	7,92359	**7,88632**	7,92804	7,89508	0,03189
ASCON	7,86178	7,90193	7,91007	7,85441	7,94030	7,85837	7,85650	**7,92364**	7,88536	**7,93770**	7,89301	0,03256
ASCON-PRF	7,86178	**7,90465**	7,91414	7,85528	7,94189	**7,87119**	**7,88791**	7,92211	7,88170	7,92772	**7,89684**	0,02812
Texto Claro	6,52363	6,58931	6,62835	6,45793	6,51030	6,48920	6,50881	6,53171	6,56181	6,52749	6,53285	0,04676

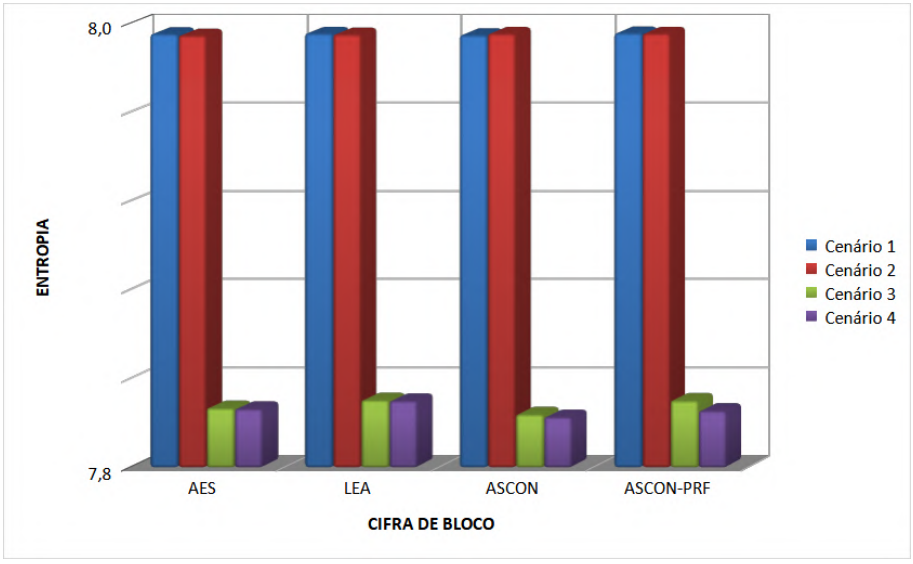
Fonte: O Autor (2023).

Figura 31 – Entropia de cada cifra por cenário.



Fonte: O Autor (2023).

Figura 32 – Detalhamento dos valores de entropia entre as cifras para cada cenário.



Fonte: O Autor (2023).

A Figura 33 demonstra as médias obtidas em cada cifra para os cenários correspondentes. Observa-se que os comportamentos em relação ao nível de incerteza causado às mensagens de entrada, são muito semelhantes. No entanto, percebe-se que para a cifragem com mensagens constituídas por 112 bits, as cifras LEA e ASCON-PRF apresentaram os melhores resultados, enquanto que o AES foi a melhor quando a mensagem possui 80 bits para a cifragem.

A Figura 34 exibe o gráfico da média de entropia dos dados cifrados de cada aeronave de forma ampliada.

Tabela 9 – Entropia de cada cifra para o cenário 2.

CENÁRIO 2 - Cifragem mensagem completa com TWEAK												
CIFRA	IDENTIFICADOR ICAO 24 BITS										MÉDIA	DESV. PADRÃO
	E49A47	E498E8	E495B2	E49390	E4910B	E492CF	E495D0	E49165	E4953A	E4917D		
AES	7,84875	**7,90421**	7,91352	7,83931	7,93586	7,85844	7,86873	7,92439	7,88602	7,92457	7,89038	0,03312
LEA	7,86491	7,89626	**7,92865**	7,86015	**7,94167**	**7,87571**	**7,87063**	7,92689	**7,88653**	7,92551	**7,89769**	0,02889
ASCON	7,87030	7,89122	7,92085	7,85807	7,93601	7,87483	7,86227	**7,93323**	7,88063	**7,92995**	7,89574	0,02950
ASCON-PRF	**7,87151**	7,88848	7,91583	**7,87961**	7,93614	7,87090	7,86770	7,92041	7,88360	7,92589	7,89601	0,02450
Texto Claro	6,52363	6,58931	6,62835	6,45793	6,51030	6,48920	6,50881	6,53171	6,56181	6,52749	6,53285	0,04676

Fonte: O Autor (2023).

Tabela 10 – Entropia de cada cifra para o cenário 3.

CENÁRIO 3 - Mensagem sem cifrar o cabeçalho e sem uso do TWEAK												
CIFRA	IDENTIFICADOR ICAO 24 BITS										MÉDIA	DESV. PADRÃO
	E49A47	E498E8	E495B2	E49390	E4910B	E492CF	E495D0	E49165	E4953A	E4917D		
AES	**7,66817**	**7,71160**	7,71572	7,63225	7,74287	7,64782	7,66770	**7,73152**	7,69861	**7,72261**	**7,69389**	0,03567
LEA	7,65325	7,69371	**7,73382**	**7,66018**	7,74916	7,63881	7,64187	7,72084	**7,71169**	7,71368	7,69170	0,03816
ASCON	7,63496	7,66505	7,71036	7,63600	7,74784	**7,65429**	7,64490	7,73121	7,70562	7,71930	7,68495	0,04023
ASCON-PRF	7,64432	7,69339	7,72965	7,62242	**7,75250**	7,63964	**7,67081**	7,73086	7,69824	7,71279	7,68946	0,04160
Texto Claro	6,52363	6,58931	6,62835	6,45793	6,51030	6,48920	6,50881	6,53171	6,56181	6,52749	6,53285	0,04676

Fonte: O Autor (2023).

Tabela 11 – Entropia de cada cifra para o cenário 4.

CENÁRIO 4 - Mensagem sem cifrar o cabeçalho e com uso do TWEAK												
CIFRA	IDENTIFICADOR ICAO 24 BITS										MÉDIA	DESV. PADRÃO
	E49A47	E498E8	E495B2	E49390	E4910B	E492CF	E495D0	E49165	E4953A	E4917D		
AES	**7,64807**	7,67614	7,71355	7,62964	**7,75312**	**7,65534**	7,66511	7,72089	7,69273	**7,73040**	**7,68850**	0,03811
LEA	7,63809	**7,70510**	**7,72734**	7,60822	7,74624	7,64152	7,65104	**7,73815**	**7,70266**	7,71726	7,68756	0,04604
ASCON	7,63194	7,69511	7,71134	**7,65320**	7,74637	7,63397	**7,66832**	7,72031	7,68471	7,72084	7,68661	0,03712
ASCON-PRF	7,61927	7,68669	7,71340	7,64848	7,74425	7,64092	7,65680	7,72356	7,67444	7,71785	7,68256	0,03916
Texto Claro	6,52363	6,58931	6,62835	6,45793	6,51030	6,48920	6,50881	6,53171	6,56181	6,52749	6,53285	0,04676

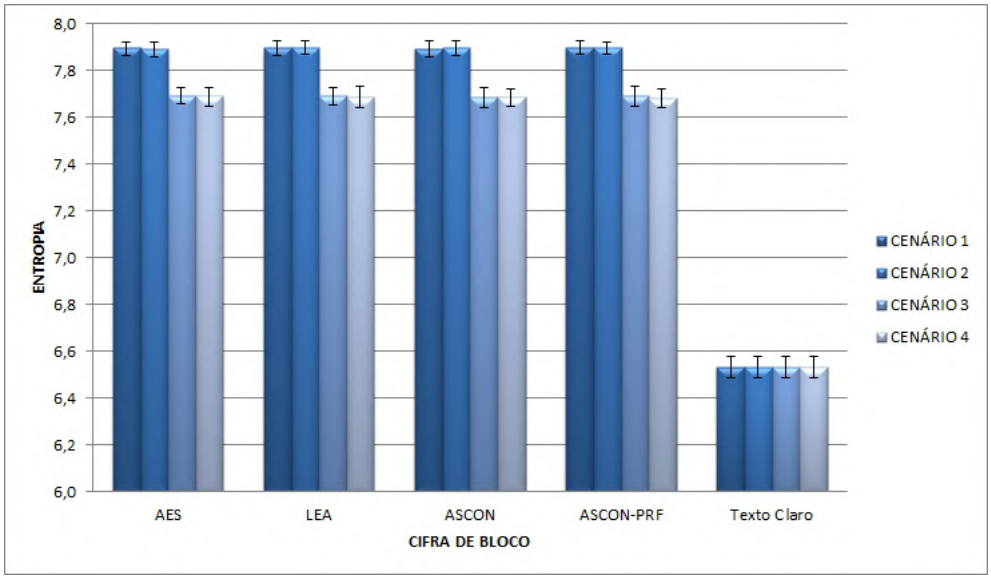
Fonte: O Autor (2023).

5.2.2 Análise de Desempenho do Tempo de Processamento

Nota-se que a cifragem FPE no modo FF1 com a cifra de bloco AES de 128 bits possui a maior latência dentre todas as cifras. Pode-se atribuir esse aspecto ao fato do AES utilizar várias etapas que compreendem a expansão da chave, substituição, permutação e embaralhamento para cada rodada, além de consumir mais memória com o uso de S-Box. A cifra LEA, desenvolvida na Coréia do Sul, possui uma estrutura muito mais simples, desenvolvida para ser utilizada em dispositivos *lightweight*, sendo a cifra que apresentou o melhor desempenho. O compilador utilizado na implementação foi o GCC (*GNU Compiler Collection*) com o modo de otimização -O2 habilitado.

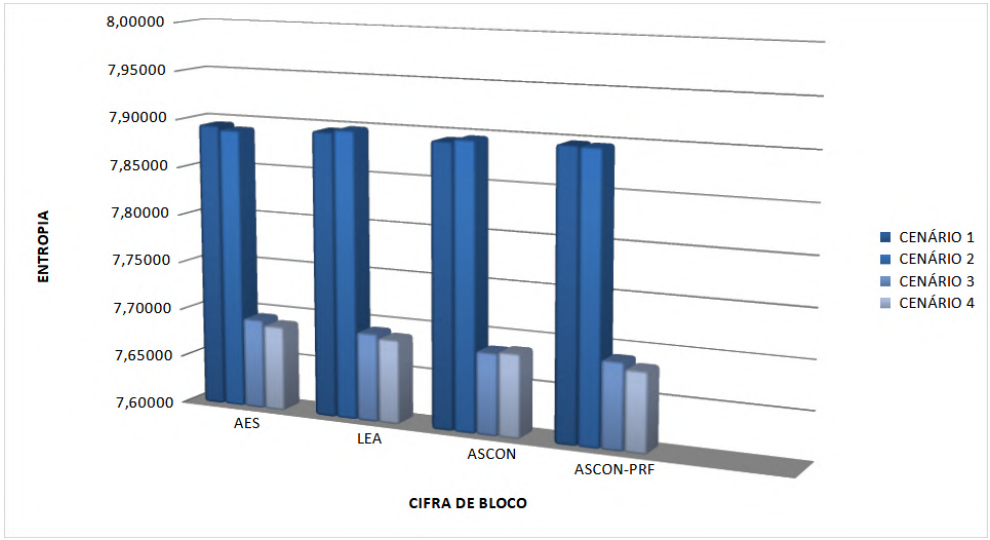
A Tabela 12 contém o resultado do tempo de processamento para cada cifra de bloco utilizada como PRF no modo FF1 realizados para todos os cenários observados.

Figura 33 – Média de Entropia dos dados cifrados de cada aeronave com desvio padrão.



Fonte: O Autor (2023).

Figura 34 – Média de Entropia dos dados cifrados de cada aeronave ampliado.



Fonte: O Autor (2023).

5.2.3 Comparação com Métodos Existentes

Nessa Seção, é efetuada a comparação com outros métodos existentes de cifragem para dados ADS-B com a utilização do FPE. Os modos de operação para cifragem por preservação de formato foram descritos por (BELLARE; ROGAWAY, 2010), que detalham os parâmetros para cifragem com alfabeto binário ou decimal. Esse trabalho serve como referência para as soluções propostas posteriormente na aplicação do FPE para a comunicação ADS-B.

O trabalho desenvolvido por (FINKE et al., 2013) foi o primeiro relacionado à aplicação da FPE em mensagens ADS-B. Os autores realizaram simulações com dados aleatórios e também simularam informações com bytes fixos. Atribuíram a análise de entropia da informação para

Tabela 12 – Latência média de cifragem para cada cifra.

LATÊNCIA (ms)				
CIFRA	CENÁRIO 1	CENÁRIO 2	CENÁRIO 3	CENÁRIO 4
AES	16	16	16	16
LEA	1	1	1	1
ASCON	5	5	5	5
ASCON-PRF	3	3	3	3

Fonte: O Autor (2023).

definir a segurança do método.

A pesquisa de (MARKANI et al., 2023), baseou-se no estudo de (FINKE et al., 2013), obtendo resultados semelhantes. Nesse caso, os autores utilizaram informações ADS-B reais coletadas de uma aeronave CESSNA e sugeriram uma forma de transmissão de chave por *blockchain*.

Os trabalhos apresentados aplicaram a FPE no protocolo de comunicação ADS-B e demonstraram o ganho relacionado ao ofuscamento das informações e o baixo impacto no padrão atual do protocolo, permitindo manter o formato das mensagens. No entanto, a cifra de bloco da função pseudoaleatória dessas implementações foi o AES-128. Não há outro estudo que aplique outras funções no algoritmo FPE para a comunicação ADS-B. Com a finalidade de observar os ganhos obtidos, executou-se a comparação das informações de texto claro e cifragem dos trabalhos anteriores em relação aos dados dessa pesquisa.

Tabela 13 – Valor de entropia dos testes com bytes fixos na mensagem de entrada, utilizando-se FPE com AES-128 como cifra de bloco na função pseudoaleatória.

QUANT. BYTES CIFRADOS	**FINKE et al., (2013)**		**MARKANI et al., (2023)**		**O AUTOR (2023)**	
	Entrada	Cifrado	Entrada	Cifrado	Entrada	Cifrado
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

Fonte: O autor

.

Observa-se pela Tabela 13 que os valores encontrados nessa pesquisa estão na mesma faixa dos estudos anteriores, levando-se em consideração condições semelhantes, como a utilização do AES-128 para obtenção desses dados.

Essa pesquisa não teve foco no tratamento da chave criptográfica. Pode-se efetuar uma nova pesquisa sobre o tratamento da chave e também sobre a aplicação periódica de valores no campo TWEAK em complemento a chave. Nesse tocante, esse autor sugere a utilização da identificação da aeronave obtida no campo 7 do plano de voo (DECEA, 2020). Esse campo deve ser preenchido pelos pilotos ou companhias aéreas. Os planos de voo são tramitados digitalmente através dos sistemas de tratamento e visualização de dados do Controle de Tráfego Aéreo. Con-

forme a página do DECEA na internet <https://www.decea.mil.br/?i=midia-e-informacao&p=pg_noticia&materia=sigma-um-jeito-inovador-de-enviar-planos-de-voo-pela-internet>, acessado em 03 de julho de 2023, o sistema SIGMA pode receber qualquer plano de voo pela internet e irá tramitar pelos sistemas informatizados do DECEA, desde a origem do voo até o destino. Na solução proposta, o aviônico da aeronave irá utilizar a identificação da aeronave no campo TWEAK para cifrar os dados ADS-B. Os sistemas de solo receberão o registro pelo plano de voo e poderão realizar a decifragem em conjunto com a chave criptográfica.

5.3 CONSIDERAÇÕES

O sistema desenvolvido para captura e cifragem de dados ADS-B instalado nas imediações do aeroporto internacional de Recife, permitiu armazenar e avaliar o desempenho do algoritmo FPE no modo FF1 com as cifras de bloco AES, LEA, ASCON e ASCON-PRF como funções pseudoaleatórias. Durante os testes, foram avaliados 4 cenários distintos e observado que todas as cifras de bloco permitiram aumento da entropia da informação ADS-B transmitida pelas aeronaves.

6 CONCLUSÃO E TRABALHOS FUTUROS

Na presente pesquisa, foi abordada a questão da segurança da informação relacionada à comunicação ADS-B na aviação, avaliando seus impactos perante a importância desse protocolo para aviação atual e propondo uma solução criptográfica com a utilização das mais recentes cifras simétricas desenvolvidas para ambiente *lightweight*. Foi realizada uma avaliação da cifragem com preservação de formato e da sua aplicabilidade nos meios atuais, bem como conduzida uma análise comparativa entre quatro cifras de bloco quando utilizadas como funções pseudoaleatórias na rede de Feistel.

Com esse propósito, realizou-se uma revisão bibliográfica sobre os importantes aspectos de vulnerabilidades da comunicação ADS-B e trabalhos desenvolvidos que visam mitigar esse problema. Além disso, observou-se as novas técnicas de cifras de blocos simétricas desenvolvidas para uso em dispositivos embarcados visando a utilização em ambiente IoT. Os dados coletados foram comparados a técnicas similares desenvolvidas por outros pesquisadores.

Para isso, foi desenvolvido um sistema para aquisição das informações transmitidas pelos *transporders* das aeronaves na área próxima ao terminal do aeroporto internacional do Recife/Guararapes - Gilberto Freyre, sediado em Recife/PE. Esse sistema baseado em microcontrolador ARM CORTEX M3 (NXP SEMICONDUCTORS, 2011) foi programado para realizar a cifragem e / ou decifragem com base em cifragem com preservação de formato e com quatro opções de funções pseudoaleatórias demonstrou atender as expectativas de ofuscamento das mensagens ADS-B transmitidas pelas aeronaves. Em comparação do algoritmo FPE com cifra AES-128 com outros trabalhos relacionados, como (FINKE et al., 2013) e (MARKANI et al., 2023), observa-se que o nível de entropia foi muito próximo quando cifrado os 112 bits do pacote ADS-B.

O tempo de latência, ou seja, o tempo de processamento das informações do sistema embarcado para realizar a cifragem, atende ao requerido pela norma internacional (RTCA, 2009), a qual estipula como limite 100 ms.

O trabalho desenvolvido demonstra que a cifragem com preservação de formato pode ser aplicada ao controle de tráfego aéreo e que cifras de bloco simétricas atuais utilizadas em sistemas embarcados fornecem a segurança necessária para as mensagens ADS-B permitindo melhor desempenho computacional o que reflete diretamente em custos, uma vez que dispositivos mais simples podem ser utilizados.

6.1 TRABALHOS FUTUROS

Sugere-se novas pesquisas relacionadas a tramitação da chave privada, como:

- A comunicação por enlace de dados entre controlador e piloto (CPDLC), conforme sugerido por (FINKE et al., 2013), onde a chave criptográfica seria transmitida pelos órgãos ATC para as aeronaves encapsuladas em mensagens de *uplink*;

- Trâmite da chave por meio da comunicação DCL (*Departure Clearance*) antes da decolagem, entre a Torre de Controle e a aeronave;
- Transmissão da chave juntamente com o plano de voo através dos enlaces seguros do órgão ATS (*Air Traffic Service*);

Por fim, a utilização de lógica programável (FPGA) é um campo que deve ser explorado com a aplicação da FPE com cifras *lightweight*, uma vez que o desempenho relacionado a velocidade (latência) tende a ser substancialmente menor que microcontroladores.

REFERÊNCIAS

AGBEYIBOR, R. et al. Evaluation of format-preserving encryption algorithms for critical infrastructure protection. **IFIP Advances in Information and Communication Technology**, v. 441, 03 2014. Citado na página 59.

AMIN, S. et al. Design of a cyber security framework for ads-b based surveillance systems. p. 304–309, 2014. Citado na página 37.

ANAC. **Evolução anual da movimentação de passageiro**. [S.l.]: Secretaria Nacional de Aviação Civil, 2022. <https://horus.labtrans.ufsc.br/gerencial/?auth=s#Movimentacao/Evolucao>. Acesso em: 08 de abr. de 2022. Citado na página 27.

BASSHAM, L. E. et al. **SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications**. Gaithersburg, MD, USA, 2010. Citado 2 vezes nas páginas 62 e 64.

BELLARE, M. et al. Format-preserving encryption. In: JACOBSON, M. J.; RIJMEN, V.; SAFAVI-NAINI, R. (Ed.). **Selected Areas in Cryptography**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 295–312. ISBN 978-3-642-05445-7. Citado 2 vezes nas páginas 16 e 39.

BELLARE, M.; ROGAWAY, P. The FFX Mode of operation for format-preserving encryption. 01 2010. Citado 4 vezes nas páginas 17, 41, 42 e 78.

BLACK, J.; ROGAWAY, P. Ciphers with arbitrary finite domains. In: PRENEEL, B. (Ed.). **Topics in Cryptology — CT-RSA 2002**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 114–130. ISBN 978-3-540-45760-2. Citado 2 vezes nas páginas 39 e 40.

BNILAM, N. et al. Low Cost AoA Unit for IoT Applications. In: . [S.l.: s.n.], 2019. Citado 2 vezes nas páginas 60 e 61.

BUCHANAN, W. J.; LI, S.; ASIF, R. Lightweight cryptography methods. **Journal of Cyber Security Technology**, Taylor Francis, v. 1, n. 3-4, p. 187–201, 2017. Disponível em: <https://doi.org/10.1080/23742917.2017.1384917>. Citado na página 17.

CHAN-TIN, E. et al. The frog-boiling attack: Limitations of secure network coordinate systems. **ACM Trans. Inf. Syst. Secur.**, v. 14, p. 27, 11 2011. Citado na página 33.

CIVIL, A. N. D. A. **RBAC 135: requisitos operacionais: operações complementares e por demanda**. 2010. Citado na página 72.

CIVIL, A. N. D. A. **IS 91-001: aprovação de aeronave e operadores para condução de operações PBN. Aprovação de aeronaves e operadores para condução de operações PBN**. 2023. <https://www.anac.gov.br/assuntos/legislacao/legislacao-1/iac-e-is/is/is-91-001>. Acesso em: 03 de jul. 2023. Citado na página 72.

DECEA. **MCA 100-11 PREENCHIMENTO DOS FORMULÁRIOS DE PLANO DE VOO**. 2020. Citado na página 79.

DECEA. **Melhoria da Vigilância no Espaço Aéreo**. [S.l.]: DECEA, 2022. <https://sirius.decea.mil.br/empreendimentos/meteorologia-aeronautica/melhoria-da-vigilancia-no-espaco-aereo-2/>. Acesso em: 03 de abr. de 2022. Citado 2 vezes nas páginas 20 e 30.

DOBRAUNIG, C. et al. **Ascon v1.2**. 2019. Submission to Round 1 of the NIST Lightweight Cryptography project. Disponível em: <https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/ascon-spec.pdf>. Citado 5 vezes nas páginas 54, 55, 56, 57 e 59.

DURAK, F. B.; VAUDENAY, S. Breaking the FF3 Format-Preserving Encryption Standard over Small Domains. In: KATZ, J.; SHACHAM, H. (Ed.). **Advances in Cryptology – CRYPTO 2017**. Cham: Springer International Publishing, 2017. p. 679–707. ISBN 978-3-319-63715-0. Citado na página 41.

DWORKIN, M. **Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption**. [S.l.]: Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, 2016. Citado 6 vezes nas páginas 41, 42, 43, 44, 45 e 47.

DWORKIN, M. et al. **Advanced Encryption Standard (AES)**. [S.l.]: Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD, 2001. Citado 3 vezes nas páginas 48, 49 e 59.

DWORKIN, M.; PERLNER, R. **Analysis of VAES3 (FF2)**. 2015. Cryptology ePrint Archive, Paper 2015/306. <https://eprint.iacr.org/2015/306>. Disponível em: <https://eprint.iacr.org/2015/306>. Citado na página 41.

FAA. **Automatic Dependent Surveillance-Broadcast (ADS-B)**. [S.l.]: FAA, 2021. <https://www.faa.gov/nextgen/programs/adsb/>. Acesso em: 03 de abr. de 2022. Citado na página 20.

FINKE, C. et al. Enhancing the security of aircraft surveillance in the next generation air traffic control system. **International Journal of Critical Infrastructure Protection**, v. 6, p. 3–11, 03 2013. Citado 5 vezes nas páginas 17, 73, 78, 79 e 81.

FRANCISCONE, B. G.; LIMA, P. A. L. 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**, v. 01, n. 1, p. 6–32, 2021. Citado na página 15.

GILBERT, G. A. Historical development of the air traffic control system. **IEEE Transactions on Communications**, v. 21, n. 5, p. 364–375, May 1973. Citado 3 vezes nas páginas 15, 20 e 21.

GUIMARÃES, A. K. A. de O. et al. O transporte aéreo e suas áreas de atuação. **IV Colóquio Estadual de Pesquisa Multidisciplinar e II Congresso Nacional de Pesquisa Multidisciplinar**, 05 2019. Citado na página 15.

HOFFSTEIN, J.; PIPHER, J.; SILVERMAN, J. **An Introduction to Mathematical Cryptography**. 2. ed. [S.l.]: Springer Publishing Company, Incorporated, 2014. ISBN 978-1-4939-1710-5. Citado na página 72.

HONG, D. et al. Lea: A 128-bit block cipher for fast encryption on common processors. In: KIM, Y.; LEE, H.; PERRIG, A. (Ed.). **Information Security Applications**. Cham: Springer International Publishing, 2014. p. 3–27. ISBN 978-3-319-05149-9. Citado 3 vezes nas páginas 49, 52 e 59.

JAHNAVI et al. Implementation of wide band FM receiver on RTL-SDR. **International Journal of Engineering Research and**, V5, 05 2016. Citado 2 vezes nas páginas 59 e 60.

JETVISION. **RTL1090 SOFTWARE FOR ADS-B DONGLES**. [S.l.]: JETVISION, 2023. <https://rtl1090.com/>. Acesso em: 08 de maio de 2023. Citado na página 61.

KUMAR, H. V. et al. Tracking of Aircrafts Using Software Defined Radio (SDR) With An Antenna. **International Journal of Scientific Research in Science and Technology**, p. 660–665, 06 2021. Citado na página 61.

MARKANI, J. et al. Security establishment in ADS-B by format-preserving encryption and blockchain schemes. **Applied Sciences**, v. 13, p. 3105, 02 2023. Citado 2 vezes nas páginas 79 e 81.

MOHAMED, H. et al. Partial Discharge Detection Using Low Cost RTL-SDR Model for Wideband Spectrum Sensing. In: . [S.l.: s.n.], 2016. Citado na página 60.

NIST. **Guidelines for Implementing and Using the NBS Data Encryption Standard**. [S.l.]: National Bureau of Standards, 1981. <https://csrc.nist.gov/publications/detail/fips/74/archive/1981-04-01>. Citado na página 39.

NIST. **Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption**. [S.l.]: NIST, 2019. <https://csrc.nist.gov/publications/detail/sp/800-38g/rev-1/draft>. Acesso em: 01 de mar. de 2023. Citado na página 41.

NIST, C. S. R. C. **Examples with Intermediate Values**. 2023. <https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/example-values>. Acesso em: 08 de jun. 2023. Citado na página 74.

NXP SEMICONDUCTORS. **LPC1769/68/67/66/65/64/63 Product data sheet**. [S.l.], 2011. Rev.7. Citado 3 vezes nas páginas 59, 65 e 81.

ORLANDO, V. A. The mode s beacon radar system. **Lincoln Laboratory Journal**, v. 2, n. 3, p. 345–362, 1989. Citado 3 vezes nas páginas 15, 23 e 25.

PAAR, C.; PELZL, J. **Understanding Cryptography - A Textbook for Students and Practitioners.** [S.l.]: Springer, 2010. I-XVIII, 1-372 p. ISBN 978-3-642-04100-6. Citado 6 vezes nas páginas 40, 42, 47, 48, 49 e 50.

PEETERS, M.; BERTONI, G. V. A. G.; DAEMEN, J. **Cryptographic sponges functions**. 2011. Citado 2 vezes nas páginas 54 e 55.

PROAKIS, J. G.; SALEHI, M. **Communication Systems Engineering.** 2nd. ed. [S.l.]: Prentice Hall, Inc., 2002. ISBN 0-13-095007-6. Citado na página 25.

REALTEK SEMICONDUCTOR CORP. **RTL2832U data sheet**. [S.l.], 2010. Rev.1.4. Citado na página 60.

RICHARDS, M. A.; SCHEER, J. A.; HOLM, W. A. **Principles of Modern Radar.** [S.l.]: SciTech Publishing, Inc., 2010. I. ISBN 978-1-891121-52-4. Citado 3 vezes nas páginas 15, 20 e 22.

ROGAWAY, P. A synopsis of format-preserving encryption. March 2010. Citado 6 vezes nas páginas 39, 40, 41, 43, 46 e 47.

RTCA. **RTCA DO-260B, Minimum operational performance standards for 1090MHz Extended Squitter Automatic Dependent Surveillance- Broadcast (ADS-B) and Traffic Information Services – Broadcast (TIS-B)**. 2009. Citado na página 81.

SCHÄFER, M.; LENDERS, V.; MARTINOVIC, I. Experimental analysis of attacks on next generation air traffic communication. In: . [S.l.: s.n.], 2013. p. 253–271. ISBN 9783642389795. Citado na página 32.

SEO, H. et al. Compact implementations of LEA block cipher for low-end microprocessors. In: . [S.l.: s.n.], 2016. v. 9503, p. 28–40. ISBN 978-3-319-31874-5. Citado na página 59.

SESAR. **ADS-B surveillance of aircraft in flight and on the surface**. [S.l.]: SESAR, 2022. <https://www.sesarju.eu/sesar-solutions/ads-b-surveillance-aircraft-flight-and-surface>. Acesso em: 03 de abr. de 2022. Citado na página 20.

SHANNON, C. E. A mathematical theory of communication. **The Bell System Technical Journal**, v. 27, p. 379–423, 1948. Citado na página 40.

SILICON LABS. **CP2102 data sheet**. [S.l.], 2017. Rev.1.8. Citado na página 65.

STROHMEIER, M.; LENDERS, V.; MARTINOVIC, I. Security of ADS-B: State of the art and beyond. **IEEE Communications Surveys e Tutorials**, v. 17, 07 2013. Citado 2 vezes nas páginas 32 e 33.

STROHMEIER, M.; LENDERS, V.; MARTINOVIC, I. On the security of the automatic dependent surveillance-broadcast protocol. p. 22, Abril 2014. Citado na página 16.

SUN, J. **The 1090 Megahertz Riddle. A Guide to Decoding Mode S and ADS-B Signals.** [S.l.]: TU Delft OPEN Publishing, 2021. ISBN 978-94-6366-402-8. Citado 5 vezes nas páginas 15, 16, 25, 26 e 28.

SUNG, M.-J.; BAE, G.-C.; SHIN, K.-W. Implementation of lightweight encryption algorithm LEA. **IDEC Journal of Integrated Circuits and Systems**, v. 02, July 2016. Citado 4 vezes nas páginas 50, 51, 52 e 53.

TRIM, R. M. A brief history of the development of radar in great britain up to 1945. **Measurement + Control**, v. 35, p. 299–301, December 2002. Citado 5 vezes nas páginas 15, 20, 21, 22 e 23.

TURAN, M. S. et al. **Status Report on the Final Round of the NIST Lightweight Cryptography Standardization Process**. NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, 2023. Disponível em: <https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=936814>. Citado na página 54.

WESSON, K. D.; HUMPHREYS, T. E.; EVANS, B. L. Can cryptography secure next generation air traffic surveillance? In: . [S.l.: s.n.], 2014. Citado 5 vezes nas páginas 21, 24, 27, 28 e 29.

WU, Z. et al. An ADS-B message authentication method based on certificateless short signature. **IEEE Transactions on Aerospace and Electronic Systems**, v. 56, n. 3, p. 1742–1753, June 2020. Citado 2 vezes nas páginas 28 e 29.

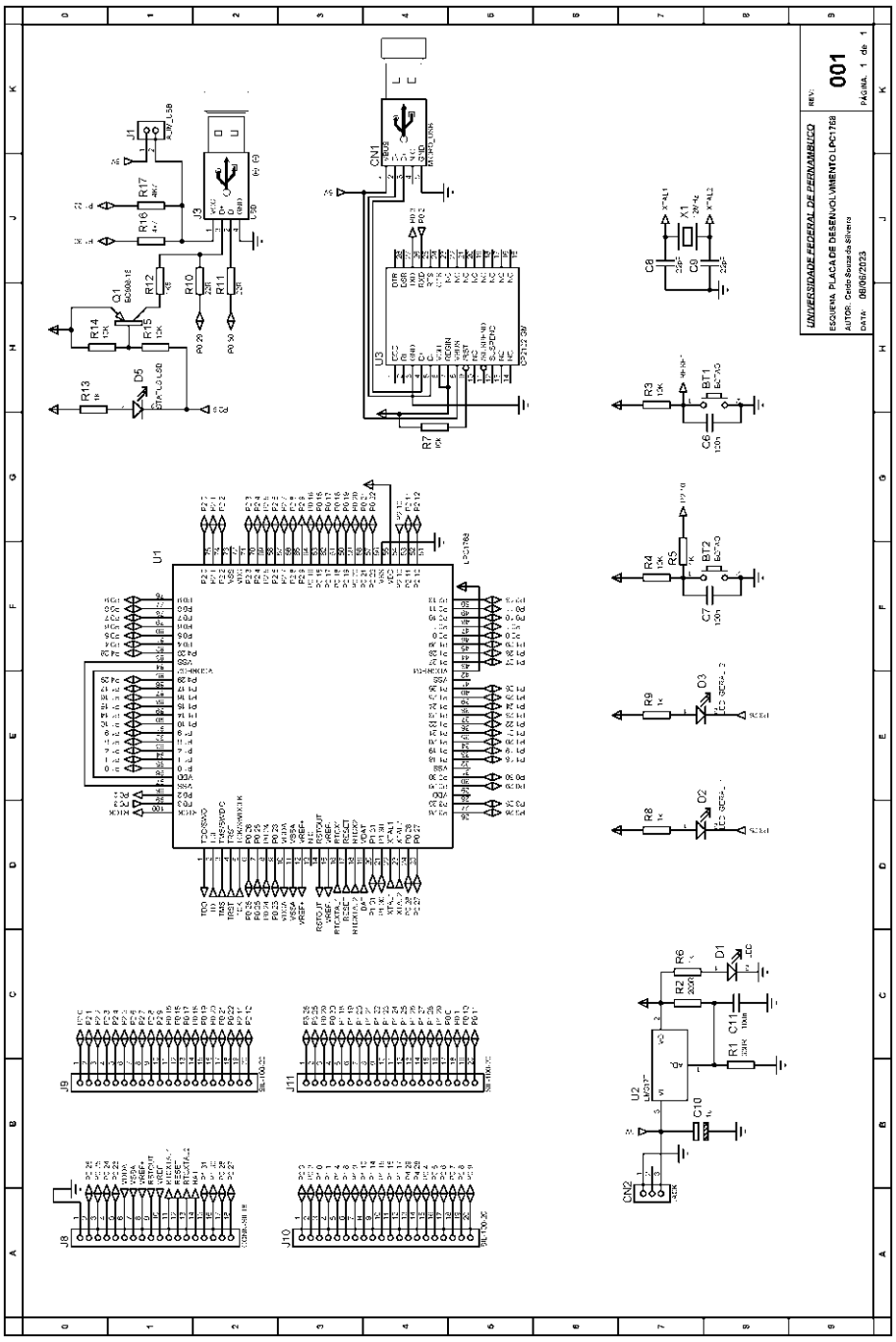
WU, Z.; SHANG, T.; GUO, A. Security issues in automatic dependent surveillance - broadcast (ADS-B): A survey. **IEEE Access**, v. 8, p. 122147–122167, July 2020. Citado 5 vezes nas páginas 16, 17, 31, 33 e 34.

APÊNDICE A – HARDWARE DESENVOLVIDO

PLACA DE DESENVOLVIMENTO LPC1768

A Figura 35 exibe o circuito desenvolvido para analisar o funcionamento do algoritmo FPE/FF1 com as cifras AES-128, LEA, ASCON e ASCON-PRF.

Figura 35 – Esquema elétrico da placa de desenvolvimento com microcontrolador LPC1768.



Fonte: O Autor (2023).