IMPACTOS DA UTILIZAÇÃO DE CRIPTOGRAFIA BLOWFISH EM SISTEMAS DE COMUNICAÇÃO BASEADOS EM OFDM[1]

Carlos Felipe Celestino Leonel – cfleonel@gmail.com

Prof. Dr. Jefferson Jesus Hengles Almeida (Orientador) – jefferson.almeida@mackenzie.br

RESUMO

A crescente evolução tecnológica e o aumento na interação do homem com serviços digitais ocasionaram o incremento da necessidade de segurança para transações que envolvem dados e informações pessoais. Isto porque, a garantia de integridade e privacidade das informações passou a ser discutida com maior frequência, inclusive se tornando objeto de legislações. Tal fato vem de encontro com o constante surgimento de técnicas de invasão e que aumentam os riscos de acessos indevidos a dados de usuários. Deste modo, medidas de segurança tais como a implementação de algoritmos de criptografia são essenciais. Este trabalho visa o desenvolvimento do estudo teórico e prático sobre a possibilidade de implementação do algoritmo Blowfish como estágio de criptografia para um modelo baseado em *Orthogonal Frequency Division* Multiplexing (OFDM), de forma a verificar sua capacidade de proteger os dados transmitidos, seu impacto sobre a demanda computacional do sistema. Para tanto, o método criptográfico de interesse foi implementado em linguagem C para operação junto a blocos de processamento relacionados ao OFDM no ambiente de simulação GNU Radio. O projeto tem como foco o estudo de caso relacionado a transmissão sem fio criptografada, que possa ser aplicado, por exemplo, ao sistema de televisão digital para proteger dados do canal de retorno, que pode ser utilizado para ações de *Healthcare.* O desenvolvimento foi realizado a partir do estudo dos aspectos essenciais para assegurar a privacidade das informações transmitidas, utilizando técnicas e métodos matemáticos que garantem a proteção dos dados.

Palavras-chave: *Blowfish. OFDM. GRC.*

[1] Artigo do Trabalho de Conclusão de Curso, Graduação em Engenharia Elétrica, EE, UPM, São Paulo, 2021.

IMPACT OF USING BLOWFISH ENCRYPTION IN COMMUNICATION SYSTEMS BASED ON OFDM

ABSTRACT

The growing technological evolution and the increase in human interaction with digital services led to the increased need for security for transactions involving personal data and information. This is because the guarantee of integrity and privacy of information began to be discussed more frequently, even becoming the object of legislation. This fact is in line with the constant emergence of hacking techniques that increase the risk of unauthorized access to user data. Therefore, security measures such as the implementation of encryption algorithms are essential. This work aims to develop a theoretical and practical study on the possibility of implementing the Blowfish algorithm as a cryptography stage for a model based on Orthogonal Frequency Division Multiplexing, in order to verify its ability to protect transmitted data, its impact on computational demand. of the system. Therefore, the cryptographic method of interest was implemented in C language for operation with processing blocks related to OFDM in the GNU Radio simulation environment. The project focuses on the case study related to encrypted wireless transmission, which can be applied, for example, to the digital television system to protect data from the return channel, which can be used for Healthcare actions. The development was carried out based on the study of essential aspects to ensure the privacy of the information transmitted, using mathematical techniques and methods that guarantee data protection.

Key-words: *Blowfish. OFDM. GRC.*

1 INTRODUÇÃO

A evolução tecnológica tem mudado constantemente a forma como as pessoas vivem e interagem em sociedades. Isto porque diversos serviços passaram a ser disponibilizados de forma digital e a ter maior difusão, principalmente no tocante às formas de comunicação. Contudo, neste cenário, a preocupação com a segurança e privacidade dos dados e informações dos usuários tornou-se indispensável. Assim, a necessidade de aplicação de técnicas de criptografia nos sistemas ganhou destaque como ferramenta fundamental para minimizar o risco de exposições indesejadas. (REIS, 1989).

No passado, a criptografia era um recurso que tinha aplicações restritas ao ambiente militar. (REIS, 1989). Porém, o incremento da capacidade de processamento dos sistemas computacionais propiciou a implementação de algoritmos cada vez mais complexos e que garantiam maior grau de proteção das informações. Não obstante, por exemplo, o volume de tráfego criptografado da internet se mantém acima de 90% no mundo. (GOOGLE, 2021). Ainda assim, há espaço para utilização de técnicas de encriptação em comunicações *wireless*.

Os sistemas de comunicação digital sem fio são utilizados para diversas aplicações, contemplando a entrega de variados tipos de conteúdo que vão desde textos até vídeos, por exemplo. Diante desde fato, o estudo de técnicas de criptografia aplicadas a esta forma de transmissão de dados passa a ser fundamental, uma vez que há o surgimento constante de métodos mal-intencionados que visam o extravio e roubo de informações. (SARADAH, 2018). Neste sentido, pesquisas voltadas para implementações de criptografia aplicadas a camada física desses sistemas ganham destaques.

Atualmente, a técnica de modulação mais utilizada no mundo para camada física dos sistemas de telecomunicações é o *Orthogonal Frequency Division Multiplexing* (OFDM). Este modelo matemático consiste na divisão de espectro de frequências em diversas subportadoras que são ortogonais entre si. (ZHANG, 2016). Além disso, há a aplicação do conceito de "prefixo cíclico" que minimiza os efeitos do canal de comunicação sobre os sinais transmitidos. (SARADAH, 2018). Desta forma, há a possibilidade de estudos sobre os impactos de aplicação de um estágio de criptografia atrelado ao OFDM.

O desenvolvimento de trabalhos referentes a utilização de encriptação em sistemas baseados na multiplexação por divisão de frequência tem explorado diversas vertentes, tais como a utilização de técnicas caóticas e de determinados algoritmos de criptografia. (LIU et al., 2017). Neste projeto, o foco é o desenvolvimento de um estágio de criptografia Blowfish que possa operar em conjunto com blocos de processamento do ambiente de simulação GNU-RADIO COMPANION (GRC) e aplicá-lo a modelo computacional do OFDM, visando identificar os impactos causados tanto do ponto de vista de segurança, quanto em relação ao incremento de complexidade do sistema.

2 REVISÃO DA LITERATURA

Nas seções seguintes que estão apresentados os conceitos teóricos sobre os tópicos de interesse para o desenvolvimento deste trabalho, exemplificando como estudos prévios abordaram o tema discutido.

2.1. SEGURANÇA E PRIVACIDADE

O modo como a tecnologia alcança as pessoas atualmente é muito diferente em relação há anos. Isto porque estão constantemente surgindo inovações que visam trazer maior conforto e segurança para os usuários em relação a serviços e comunicações. Devido a este fato, houve o crescimento da preocupação com a privacidade e proteção dos dados pessoais que são necessários para utilização desde recursos ou que são por eles tratados. Tal situação vem de encontro com pesquisas recentes que tem relatado o aumento no volume de vazamentos de dados sensíveis e privados, trazendo uma estatística que gera um alerta ainda maior sobre a necessidade de segurança das informações. (MACHADO et al., 2019).

O tema referente à segurança de dados está relacionado à garantia de fidelidade de seu conteúdo e à restrição de acesso, inclusive com a incidências de leis que criminalizam a utilização de meios maliciosos para roubo de informações. (REVISTA DE CONCORRÊNCIA E REGULAÇÃO, 2018). Os aspectos que fazem parte do critério de proteção de informações são: confiabilidade, integridade e disponibilidade. (MACHADO et al., 2019). O primeiro pilar está relacionado à utilização de mecanismos que garantem o acesso apenas a pessoas autorizadas, a partir da utilização de senhas, *tokens* e criptografia. O segundo item refere-se à necessidade de que os dados estejam em formato correto e sejam verdadeiros para quem for utilizá-los. Por fim, o último aspecto consiste em assegurar que as informações estejam disponíveis, mesmo diante da possibilidade de manutenções de hardware, atualizações de patch de software, dentre outros. (SERVICES, 2018).

Diante de um cenário crítico de ameaça à segurança das informações, governos tomaram medidas preventivas para garantir que as empresas aumentassem os investimentos no tocante à proteção de dados. A União Europeia (EU) criou em 2016, por exemplo, uma regulamentação para o controle, proteção e privacidade das informações chamado Regulamento Geral de Proteção de Dados, em ingles *General Data Protection Regulation* (GDPR). No Brasil, em 2018, seguindo o modelo europeu, foi criada a Lei Geral de Proteção de Dados (LGPD). (MACHADO, 2019). Para ambos os casos, o ponto chave é fornecer segurança e proteção aos dados tratados e ou armazenados e que pertençam a usuários, bem como estabelecer responsabilidades sobre estas ações.

2.2. CRIPTOGRAFIA

A modernização e difusão dos meios de comunicação atualmente possibilita o alcance de cada vez mais usuários. Porém, a necessidade de garantir segurança às informações que trafegam pelo canal tem crescido a cada instante. Neste cenário, a criptografia consiste em uma técnica que confere proteção à transmissão e recepção de dados. De forma simplificada, seu objetivo é converter informações para um formato não legível, a partir da utilização de um algoritmo e chave que sejam conhecidos apenas pelo emissor e pelo receptor. (BHARGAVA et al., 2017).

De acordo com BHARGAVA (2017), o principal objetivo da criptografia é manter a confiabilidade e a integridade dos dados, possibilitando a utilização de meio seguro de comunicação. A Figura 1 apresentam um modelo simplificado de um estágio de encriptação e descriptografia. Nele, por exemplo, uma mensagem que foi escrita em "Texto Simples" e pode ser claramente interpretada passa por um algoritmo de criptografia que a torna ilegível para qualquer terceiro, se tornando um "Texto cifrado". Então, ela é transmitida até o destino, de forma que seu conteúdo está associado a um determinado grau de segurança, minimizando as chances que as informações sejam captadas e analisadas antes de chegar ao receptor, que possui a chave necessária para sua interpretação.

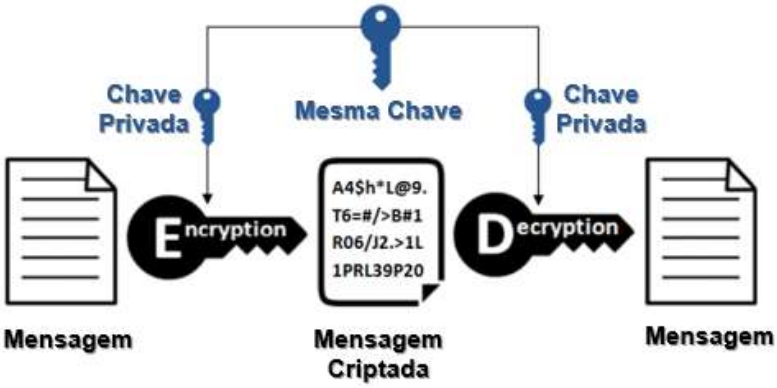
Figura 1: Construção básica em blocos para a criptografia.

Fonte: Autoria Própria.

2.2.1. Criptografia Simétrica

Criptografia simétrica é aquela em que o algoritmo possui uma chave privada que possibilita o acesso à mensagem oculta que fora enviada. Para esse método, a chave é representada por uma senha, que é utilizada tanto na origem quanto no destino, e privada pois apenas quem estiver a autorização da senha pode utilizar. Uma das principais vantagens é a sua simplicidade, dado que esta técnica apresenta algoritmos que facilitam o uso e a rapidez para executar os procedimentos criptográficos. (SILVA, 2019). A Figura 2 demonstra o fluxo de criptografia de uma mensagem com o uso de chave simétrica.

Figura 2: Procedimento para a Criptografia Simétrica.



Fonte: Silva, 2019.

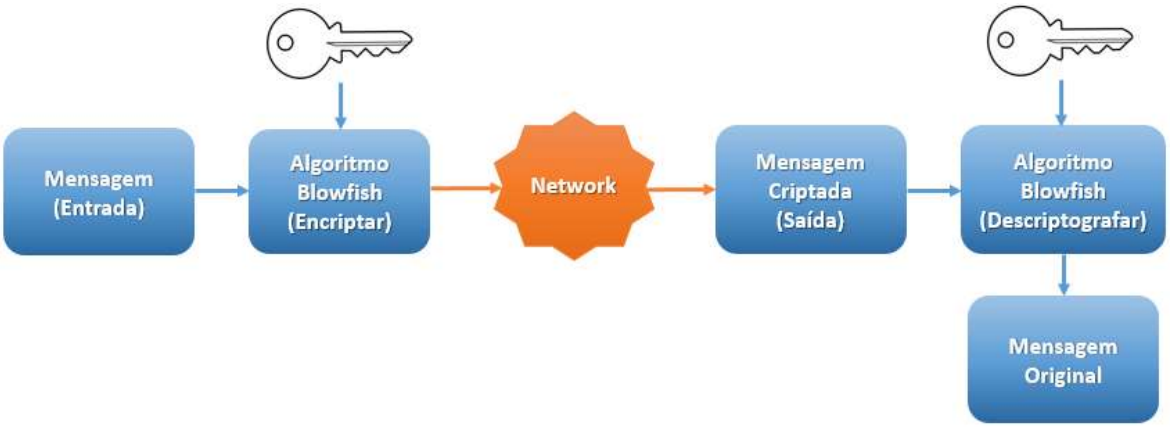
A técnica de criptografia simétrica contempla diversos modelos que são licenciados pelo Instituto Nacional de Padrões e Tecnologia, em inglês *National Institute of Standards and Technology* (NIST). Como exemplos, citam-se os padrões mais utilizados que são o *Data Encryption Standard* (DES) e o *Advanced Encryption Standard* (AES). O DES opera com blocos e chave de 64 bits sendo o algoritmo com tamanho inferior ao dos demais, tendo um comportamento com a menor estabilidade. (VOITECHEN, 2015). Contudo, o AES apresenta melhorias e aumenta a dificuldade de que a chave de criptografia seja obtida por meio de técnicas de força bruta (VOITECHEN, 2015). Ainda assim, há outros métodos que apresentam vantagens.

Em comparação com o AES, a técnica chamada de Blowfish confere maior segurança aos sistemas e mantem a eficiência, sendo mais rápido no processo de criptografar e descriptografar (MUIN. et al., 2018). Uma vez que o foco deste trabalho versará sobre a implementação do Blowfish, a próxima subseção apresenta os detalhes necessários.

2.2.1.1. Blowfish

O algoritmo Blowfish (AB) é uma técnica de criptografia de bloco simétrico e utilizando o conceito de “chave privada”. Este código criptográfico foi projetado em 1993 por Bruice Schneier e não foi atrelado a registro de patente, estando disponível gratuitamente para todos os usuários. A Figura 3 apresenta o diagrama de bloco básico para a configuração do sistema utilizando este método de encriptação.

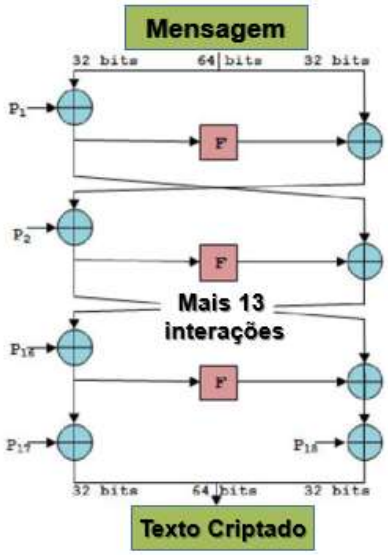
Figura 3: Diagrama do Sistema Blowfish



Fonte: Autoria Própria.

O processo de criptografia de dados ocorre por meio da interação da rede de Feistel com 16 estágios (Figura 4), onde cada um deles é composto por operações de permutação e de substituição, dependendo da chave variável que pode ter entre 32 e 448 bits (NALAWADE *et al*., 2017).

Figura 4: Algoritmo Blowfish Rede Feistel.



Fonte: NALAWADE, 2017, traduzido.

O conceito das redes de Feistel consiste em dividir um bloco de determinado comprimento em dois conjuntos (“E” e “D”), que representam as partes que caminharão pela esquerda e direita no diagrama da Figura 6. Em cada estágio são feitas as interações das Sub-chaves “K_i” e da mensagem por meio da função “F”. As equações 1 e 2 demonstram matematicamente a interação de cada bloco. (SCHNEIER, 1996).

$$E_{i+1} = D_i \quad (1)$$

$$D_{i+1} = E_i \otimes \mathrm{F}(D_i, K_i) \quad (2)$$

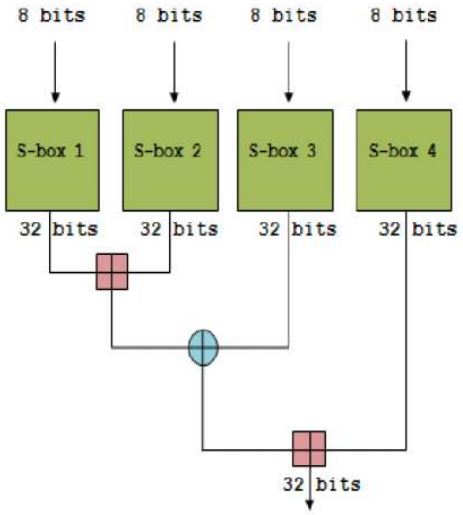
Para que seja possível iniciar o processo, o total de interações deverá ser par. Pode-se definir uma cifra de bloco em que a entrada do i-ésimo estágio é determinado a partir da saída da interação anterior. A utilização da Rede Feistel é importante por apresentar uma função garantidamente reversível, a partir da aplicação de uma operação XOR que combina as duas partes da função. A descriptografia dos dados aplica o mesmo sistema que a criptografia, porém para P1, P2 ... P18 as ordens são invertidas. (ALABAICHI *et al*., 2017), o cálculo matemático para decriptação pode ser observado nas equações 3 e 4.

$$D_i = E_{i+1} \tag{3}$$

$$E_i = D_{i+1} \otimes \mathrm{F}(E_{i+1}, K_i) \tag{4}$$

A função "F" é considerada a parte mais complexa, dado que se refere propriamente ao algoritmo de criptografia. Ela aceita um fluxo de dados de 32 bits e divide os dados em quatro partes iguais. (ALABAICHI *et al*., 2017). Os valores de cada partição de 8 bits são utilizados como entrada para cada uma das S-boxes e, em seguida, operações XOR e de adição são executadas, conforme apresentado na Figura 5. (NALAWADE *et al*., 2017).

Figura 5: Função F do Algoritmo *Blowfish*.



Fonte: NALAWADE, 2017.

2.3. SOFTWARES DE SIMULAÇÃO

O GRC (GNU-RADIO, 2019) é um software que opera nos sistemas operacionais Linux, MAC e Windows. Este ambiente permite a criação de blocos funcionais para simulações de sistemas de comunicações e possibilita a utilização de *front-ends* de rádio frequência. (ALMEIDA, 2016). Os modelos gráficos criados no GRC são chamados de fluxogramas, em inglês *FlowGraphs*. Em sua área de trabalho, é possível conectar blocos de sistemas de comunicação disponíveis em vários módulos. Além disso, é possível a manipulação de diversos tipos de variáveis que garante flexibilidade na implementação.

Já o Matlab (MATLAB, 2019) trata-se de um software de simulação baseado em programação matemática e cálculos numéricos de matrizes. A partir dele é possível realizar a implementação de algoritmos e funções que emulam sistemas e processos de diversas áreas do conhecimento. Suas ferramentas contemplam *toolboxes* aplicadas a processamento de sinais e de imagem, equações diferenciais e ordinárias, além de estatística e finanças.

3 METODOLOGIA

Para alcançar o objetivo proposto, inicialmente, o algoritmo de criptografia Blowfish foi adaptado e validado utilizando linguagem de programação C/C++. Os estágios de cifragem e decifragem foram avaliados e o modelo foi testado com as chaves disponíveis.

Em seguida, um modelo de simulação baseado no OFDM foi implementado no ambiente GRC visando possibilitar a realização de testes de transmissão de dados criptografados. Para tanto, os blocos de processamento necessários foram conectados, gerando *FlowGraph* que permitia as observações sobre os impactos do algoritmo a partir, por exemplo, de visualizadores de constelação e calculador de BER.

Em paralelo, um modelo também foi implementado no programa Matlab, com o objetivo de validar resultados intermediários de forma rápida e efetiva. Nesse sentido, funções de leitura de dados provenientes de arquivos gravados no GRC foram utilizadas e os tratamentos necessários foram realizados.

Ao final, a possibilidade de utilização do algoritmo Blowfish foi avaliada a partir dos resultados obtidos nas simulações e testes desenvolvidos. Então, as discussões e conclusões pertinentes foram apresentadas.

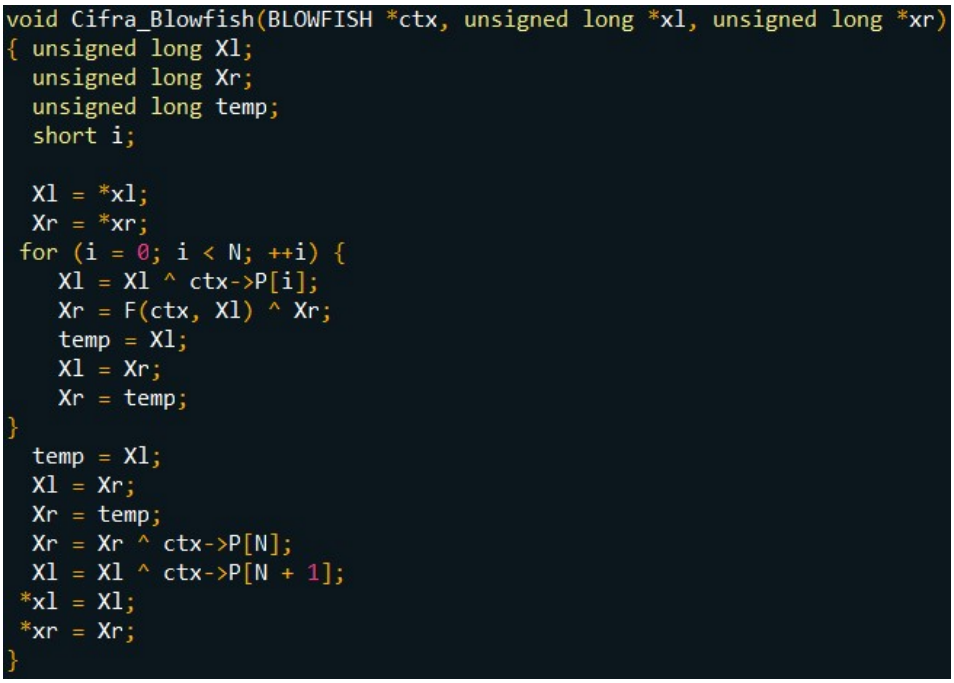
3.1. DESENVOLVIMENTO

A partir de estudos disponíveis na literatura (BUGATTI, 2005), o algoritmo Blowfish foi adaptado e validado utilizando a linguagem de programação C, tendo como base 4 funções principais:

- Inicia_Blowfish – Primeira etapa da programação onde há expansão de chave, que consiste na inicialização das variáveis do algoritmo e definição dos valores que são utilizados, tais como o vetor P das interações de comutação e as quatro caixas S;
- Cifra_Blowfish – Função que aplica a chave de criptografia e realiza a cifragem dos dados;
- Decifra_Blowfish – Função que opera a decifragem dos dados criptografados;
- F – Consiste na função que indexa os dados nas caixas S utilizando tamanho de 8 bits que são obtidos por meio da partição de blocos de 32 bits, tendo operações de adição e XOR nos dados obtidos.

O código da função que realiza a cifragem é apresentado na Figura 6.

Figura 6: Cifragem em linguagem C.



```c
void Cifra_Blowfish(BLOWFISH *ctx, unsigned long *xl, unsigned long *xr)
{ unsigned long Xl;
  unsigned long Xr;
  unsigned long temp;
  short i;

  Xl = *xl;
  Xr = *xr;
 for (i = 0; i < N; ++i) {
    Xl = Xl ^ ctx->P[i];
    Xr = F(ctx, Xl) ^ Xr;
    temp = Xl;
    Xl = Xr;
    Xr = temp;
}
  temp = Xl;
  Xl = Xr;
  Xr = temp;
  Xr = Xr ^ ctx->P[N];
  Xl = Xl ^ ctx->P[N + 1];
 *xl = Xl;
 *xr = Xr;
}
```

Fonte: Autoria própria.

O referido código realiza a operações XOR relacionadas ao operador “^” e a partir de parâmetros de ponteiros, possibilita o acesso ao vetor P e as S-boxes, tratando os blocos de 32 bits que passam pelo algoritmo (xl e xr). O processo de criptografia é feito por meio da realização de 16 interações baseadas na rede de Feistel.

Já para o processo de decifragem, é necessário executar a função de repetição “for” de forma inversa, mudando assim os índices utilizados na indexação do vetor P. A Figura 7 expressa a função de decifragem.

Figura 7: Decifragem em linguagem C.

```c
void Decifra_Blowfish(BLOWFISH *ctx, unsigned long *xl, unsigned long *xr)
{
 unsigned long Xl;
 unsigned long Xr;
 unsigned long temp;
 short  i;

 Xl = *xl;
 Xr =*xr;
 for (i = N + 1; i > 1; --i)
{
 Xl = Xl ^ ctx->P[i];
 Xr = F(ctx, Xl) ^ Xr;
 temp = Xl;
 Xl = Xr;
 Xr = temp;
}
 temp = Xl;
 Xl = Xr;
 Xr = temp;
 Xr = Xr ^ ctx->P[1];
 Xl = Xl ^ ctx->P[0];
 *xl = Xl;
 *xr = Xr;
}
```

Fonte: Autoria própria.

A rede de Feistel está implementada, conforme apresentado na Figura 8.

Figura 8: Função F em linguagem C.

```c
static unsigned long F(BLOWFISH *ctx, unsigned long x)
{ unsigned short a, b, c, d;
unsigned long y;
d = (unsigned short)(x & 0xFF); //mascara
x >>= 8;
c = (unsigned short)(x & 0xFF);
x >>= 8;
b = (unsigned short)(x & 0xFF);
x >>= 8;
a = (unsigned short)(x & 0xFF);

y = ctx->S[0][a] + ctx->S[1][b];
y = y ^ ctx->S[2][c];
y = y + ctx->S[3][d];

return y;
}
```

Fonte: Autoria própria.

Por último, a função que é responsável por inicializar o algoritmo Blowfish e a expansão de chave, pode ser vista na Figura 9.

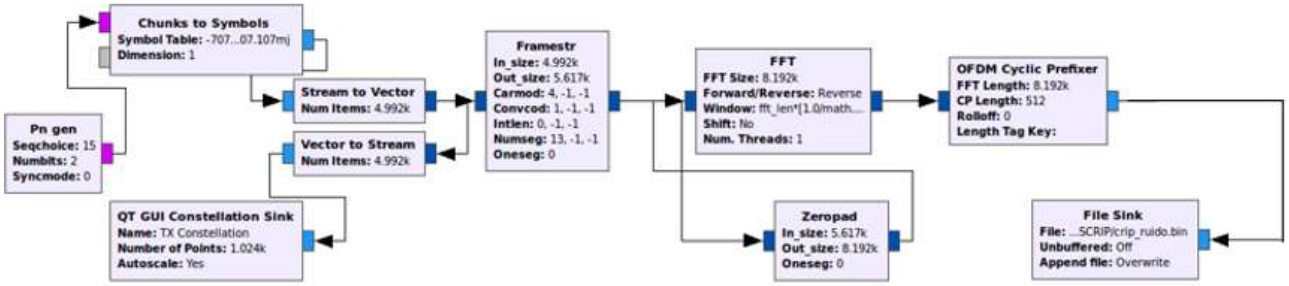
Figura 9: Inicialização do código em linguagem C.

```
void Inicia_Blowfish(BLOWFISH *ctx, unsigned char *key, int keyLen)
{
int i, j, k;
unsigned long data, datal, datar;

//inicializa S-Boxes
for (i = 0; i < 4; i++) {
  for (j = 0; j < 256; j++)
  ctx->S[i][j] = ORIG_S[i][j];
}
j = 0;

for (i = 0; i < N + 2; ++i)
{ data = 0x00000000;
//pego os 32 primeiros bits da chave
for (k = 0; k < 4; ++k) {
data = (data << 8) | key[j];
j = j + 1;
if (j >= keyLen)
j = 0;
}
ctx->P[i] = ORIG_P[i] ^ data; // faz um XOR com P[i] e com 32i bits da chave
}

datal = 0x00000000; //32 bits
datar = 0x00000000; //32 bits
for (i = 0; i < N + 2; i += 2)
{ Cifra_Blowfish(ctx, &datal, &datar);
ctx->P[i] = datal;
ctx->P[i + 1] = datar;
}

for (i = 0; i < 4; ++i)
{
for (j = 0; j < 256; j += 2)
{ Cifra_Blowfish(ctx, &datal, &datar);
ctx->S[i][j] = datal;
ctx->S[i][j + 1] = datar;
 }
}
}
```

Fonte: Autoria própria.

Após a adaptação e validação do algoritmo de criptografia, foi criado no software GRC um sistema genérico baseado na transmissão e recepção OFDM. A Figura 10 mostra o fluxograma do modulador implementado no ambiente de simulação. O bloco “Pn Gen” é a fonte que gera uma sequência binária pseudoaleatória. Em seguida, o bloco “Chunks to symbols” realiza o mapeamento dos bits de entrada para símbolos em fase e quadratura, do inglês InPhase/Quadrature (I/Q). O bloco “Stream to Vector” realiza a conversão de amostra para vetor com tamanho de 4992, que é o número de portadoras úteis no ISDB-TB. Em seguida o bloco “Framestr” realiza a inserção de pilotos resultando em um vetor com tamanho N=5617 portadoras. O bloco “Zeropad” insere nulos no centro do vetor resultando em um vetor com tamanho de 8192. Em seguida, a transformada inversa rápida de Fourier, em inglês Inverse Fast Fourier Transform (IFFT), é aplicada. A próxima etapa consiste na inserção do intervalo de guarda e o sinal então é gravado no formado numérico binário *complex* com o bloco “File Sink”. Este arquivo é então tratado externamente pelo algoritmo de criptografia implementado em linguagem C.

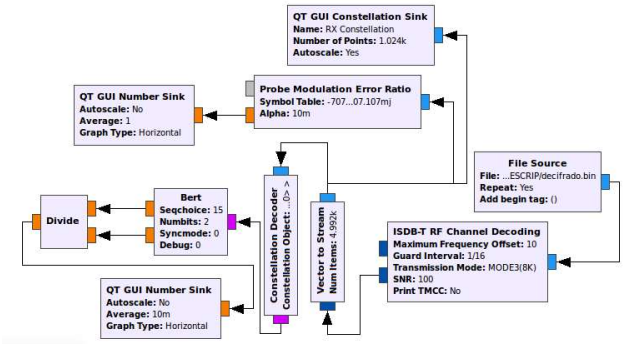
Figura 10: Modulador ISDB-TB no GRC.



Fonte: Autoria própria.

A Figura 11 mostra o fluxograma do demodulador usado nos testes. Após o processo de descriptografia do arquivo gerado na transmissão, ele é inserido novamente no sistema por meio do bloco “File Source”. Em seguida, o sinal OFDM é decodificado pelo bloco “ISDB-T RF Channel Decoding”. A Taxa de Erro de Modulação, em inglês Modulation Error Ratio (MER) é medida no bloco “Probe Modulation Error Rate”, enquanto o BER que é observado no bloco “Bert”, a partir dos cálculos descritos na Equação 5 e 6. Os blocos QT (Constellation, Frequency, Time Sink e Number Sink) são utilizados como monitores gráficos do sinal.

Figura 11: Demodulador ISDB-TB no GRC.



Fonte: Autoria própria.

$$\boldsymbol{MER(dB) = 10log_{10}\frac{\sum_{j=1}^{N}(I_j^2 + Q_j^2)}{\sum_{j=1}^{N}(\delta I_j^2 + \delta Q_j^2)}} \quad (5)$$

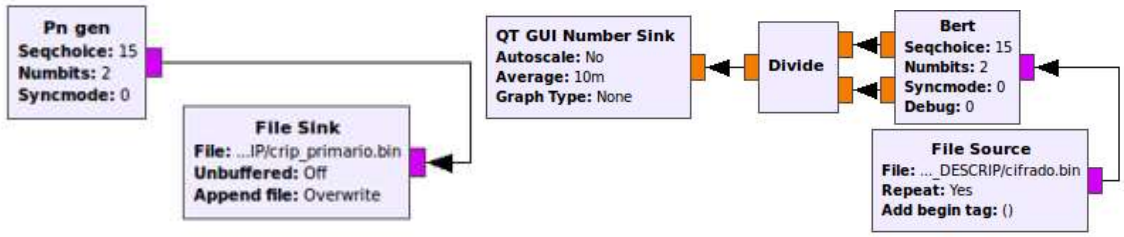
$$\boldsymbol{BER = \frac{\sum Erros}{\sum Bits}} \quad (6)$$

A equação do MER apresenta a variável *N* como o número de portadoras, e $\boldsymbol{\delta I}$ e $\boldsymbol{\delta Q}$ como o erro (distância entre o símbolo I/Q transmitidos e recebido).

4 RESULTADOS E DISCUSSÃO

A primeira validação de funcionamento do sistema consistiu em realizar a criptografia e descriptografia dos dados e realizar a medida de BER, conforme visto na Figura 12. Com o sinal extraído do bloco "Pn Gen", o arquivo em formato binário *complex* foi criptografado e descriptografado. Em seguida, o arquivo gerado foi inserido no bloco "File Source" o que permitiu realizar a medida BER que resultou em zero, demonstrando que não ocorreu erro no processo.

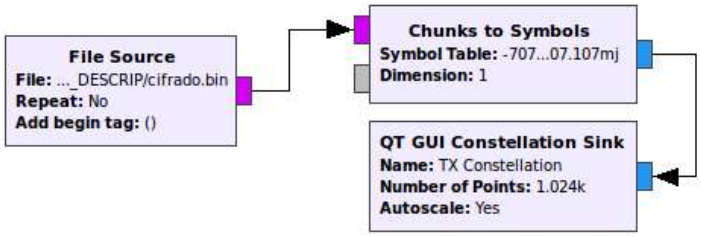
Figura 12: Medição do BER.



Fonte: Autoria própria.

No segundo teste, a influência da criptografia e descriptografia foi avaliada a partir das características observadas na constelação, utilizando o fluxograma da Figura 13.

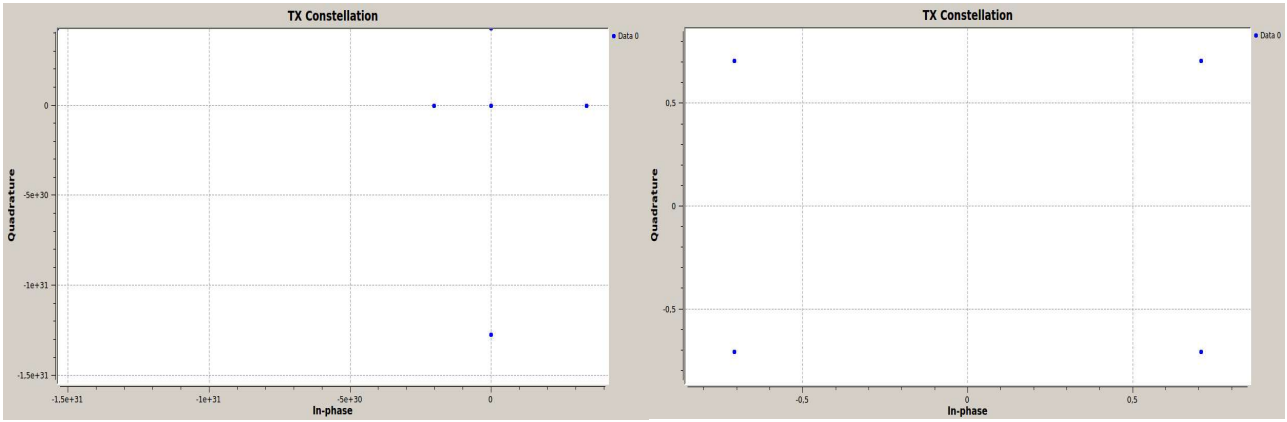
Figura 13: Gravação do bloco.



Fonte: Autoria própria.

A partir do arquivo criptografado, a constelação apresentou característica divergente da esperada, conforme Figura 14a. Já a Figura 14b apresenta a constelação do sinal descriptografado.

Figura 14: Constelação QPSK no processo de Criptografia (a), Constelação QPSK no processo de Descriptografia (b).



(a) (b)

Fonte: Autoria própria.

Visando uma análise mais precisa do comportamento dos arquivos de dados após o processo de criptografia e descriptografia, o Matlab foi utilizado para permitir a observação dos dados em formato binário e analisar a diferença. Nas seguintes Figuras, 15a comportamento do arquivo criptografado, 15b comportamento do arquivo descriptografado.

Figura 15: Comportamento da Criptografia (a), Comportamento da Descriptografia (b).

	1
1	209
2	246
3	54
4	223
5	180
6	125
7	1
8	89
9	36
10	92
11	116
12	179
13	221
14	108
15	122

(a)

	1
1	3
2	3
3	3
4	3
5	3
6	3
7	3
8	2
9	0
10	0
11	0
12	0
13	0
14	0
15	1

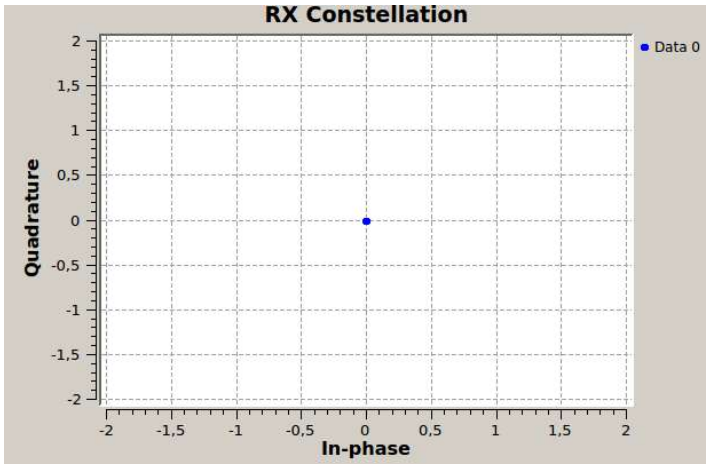
(b)

Fonte: Autoria própria.

Conforme Figura 15a, o algoritmo criptográfico alterou, devido ao seu modelo de processamento, o comportamento dos símbolos da modulação QPSK. Contudo, quando o processo de descriptografia é realizado, os dados voltam a apresentar valores entre 0 e 3, ou seja, correntes com a modulação utilizada.

Diante deste cenário, foi realizado ensaio no qual a saída do bloco “OFDM Cyclic Prefixer” foi gravada em arquivo, visando realizar a criptografia e inserir ruído pelo Matlab (simulando o canal de transmissão), e em seguida descriptografar o arquivo e encaminhar para o circuito de demodulação, para verificar a constelação. Na Figura 16 é possível ver o comportamento da constelação de recepção.

Figura 16: Constelação de Recepção.



Fonte: Autoria própria.

Diante do fato de que o processo de criptografia estava alterando o padrão das informações, foram realizados diversos testes para modificar as formas como o código do Blowfish exporta os dados criptografados. Foi utilizado funções diferentes que tivessem o mesmo tamanho da variável de escrita (32 bits), no entanto, foi concluído que a separação feita pela rede de Feistel impossibilita a implementação do algoritmo junto ao OFDM que opera com símbolos complexos.

5 CONSIDERAÇÕES FINAIS

A partir da implementação do algoritmo Blowfish em linguagem de programação C e de um sistema genérico baseado no OFDM pelo *software* GRC, foi possível avaliar a possibilidade de utilização do Blowfish como estado de criptografia.

O processo de criptografia do algoritmo Blowfish, em linguagem de programação C, foi desenvolvido e validado. No entanto, devido à forma de tratamento dos bits, não houve compatibilidade com o formato de dados que o OFDM necessita. Isto ocorreu devido às interações da rede de Feistel e a criação de sub chaves, de forma que as sequências não puderam ser lidas de

forma adequada pelos blocos utilizados no GRC.

Para trabalhos futuros, pretende-se propor uma modificação na rede de Feistel para operar de forma adequada para dados complexos, tal como demandado pelo sistema OFDM genérico implementado no GRC.

REFERÊNCIAS

ALABAICHI, Ashwak; AHMAD, Faudziah; MAHMOD, Ramlan. Security analysis of blowfish algorithm. **2013 Second International Conference On Informatics & Applications (Icia)**, p. 12-18, 14 set. 2013. Anual. IEEE. http://dx.doi.org/10.1109/icoia.2013.6650222.

ALMEIDA, Jefferson Jesus Hengles. Transmissão de Sinais do ISDB-TB em Modulação Avançada: um estudo de caso em fbmc. 2016. 105 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Presbiteriana Mackenzie, São Paulo, 2021. Cap. 2.

BUGATTI, Pedro Henrique. **IMPLEMENTAÇÃO E AVALIAÇÃO DO ALGORITMO BLOWFISH EM C, JAVA E MICROCONTROLADORES PIC**. 2021. 146 f. Monografia (Especialização) - Curso de Ciência da Computação, Centro Universitário Eurípides de Marília, Marília, 2005. Cap. 6.

BHARGAVA, Umang; SHARMA, Aparna; CHAWLA, Raghav; THAKRAL, Prateek. A new algorithm combining substitution & transposition cipher techniques for secure communication. **2017 International Conference On Trends In Electronics And Informatics (Icei)**, 619-624, 11 maio 2017. Anual. IEEE. http://dx.doi.org/10.1109/icoei.2017.8300777.

GOOGLE. **Criptografia HTTPS na Web**. 2021. Google. Disponível em: https://transparencyreport.google.com/https/overview. Acesso em: 08 maios 2021.

GNU-Radio Companion (GRC). Disponível em: Acesso em: 17 de maio de 2021.

MACHADO, R. et al. Vazamentos de Dados: Histórico, Impacto Socioeconômico e as Novas Leis de Proteção de Dados. In: 4o Workshop Regional de Segurança da Informação e de Sistemas Computacionais. Alegrete-RS, Brasil: [s.n.], 2019

MACHADO, Rodrigo Bisso. **SSPD-LGPD: uma Solução para Segurança e Privacidade de Dados no cenário da Lei Geral de Proteção de Dados**. 2019. 50 f. Tese (Doutorado) - Curso de Ciência da Computação, Universidade Federal do Pampa, Alegrete, 2021. Cap. 1.

MATLAB. Disponível em: <https://www.mathworks.com/products/matlab.html>. Acesso em: 30 nov. 2020.

MUIN, Muhammad Abdul; MUIN, Muhammad Abdul; SETYANTO, Arief; SUDARMAWAN; SANTOSO, Kartika Imam. Performance Comparison Between AES256-Blowfish and Blowfish-

AES256 Combinations. **2018 5Th International Conference On Information Technology, Computer, And Electrical Engineering (Icitacee)**, p. 137-141, 14 set. 2018. Anual. IEEE. http://dx.doi.org/10.1109/icitacee.2018.8576929.

REIS, Verônica Lagrange Moutinho dos. **Criptografia, Segurança de Dados e Privacidade - Até que ponto pode-se confiar na discrição dos computadores?** 1989. 139 f. Dissertação (Mestrado) - Curso de Ciências em Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2021. Cap. 15.

REVISTA DE CONCORRÊNCIA E REGULAÇÃO. Lisboa: Adc e Ideff, v. 35, n. 27, 13 nov. 2018. Anual. 15º Aniversário da Autoridade da Concorrência.

Saradah, N., Puja Astawa, I. G., & Sudarsono, A. (2018). Performance of OFDM Communication System with RSA Algorithm as Synchronization on SR Security Scheme Using USRP Devices. 2018 International Electronics Symposium on Engineering Technology and Applications (IES-ETA). doi:10.1109/elecsym.2018.8615566

NALAWADE, Shraphalya B.; GAWALI, Dhanashri H.. Design and implementation of blowfish algorithm using reconfigurable platform. **2017 International Conference On Recent Innovations In Signal Processing And Embedded Systems (Rise)**, p. 479-484, 27 out. 2017. IEEE. http://dx.doi.org/10.1109/rise.2017.8378204.

SCHNEIER, Bruce. **Applied Cryptography**: protocols, algorthms, and source code in c. 2. ed. New York: John Wiley & Sons, Inc., 1996. 666 p.

SILVA, Willian Wallace de Matteus. **A EVOLUÇÃO DA CRIPTOGRAFIA E SUAS TÉCNICAS AO LONGO DA HISTÓRIA**. 2019. 30 f. TCC (Graduação) - Curso de Sistemas de Informação, Instituto Federal Goiano, Goiânia, 2020. Cap. 2.

SERVICES, Getti. **Segurança de dados: confidencialidade, integridade e disponibilidade (CID)**. 2018. Blog GetTI. Disponível em: https://getti.net.br/2018/10/04/seguranca-de-dados-confidencialidade-integridade-e-disponibilidade-cid/. Acesso em: 22 maio 2021.

VOITECHEN, Dainara Aparecida. **ANÁLISE E COMPARAÇÃO DE ALGORITMOS PARA CRIPTOGRAFIA DE IMAGENS**. 2015. 158 f. Tese (Doutorado) - Curso de Informática, Departamento Acadêmico de Informática Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2021. Cap. 6.

LIU, Zhaofeng; ZHANG, Lin; RAO, Weiwei; WU, Zhiqiang. Demonstrating high security subcarrier shifting chaotic OFDM cognitive radio system using USRP. 2017 Ieee/Cic International Conference On Communications In China (Iccc), China, v. 6, n. 1, p. 1-27, 4 out. 2017. Anual. IEEE. http://dx.doi.org/10.1109/iccchina.2017.8330482.

Zhang, J., Marshall, A., Woods, R., & Duong, T. Q. (2017). Design of an OFDM Physical Layer Encryption Scheme. IEEE Transactions on Vehicular Technology, 66(3), 2114 - 2127. https://doi.org/10.1109/TVT.2016.2571264