UNIVERSIDADE FEDERAL DE PELOTAS
Instituto de Física e Matemática
Programa de Pós-Graduação em Modelagem Matemática

Dissertação

Um novo algoritmo de criptografia de alto desempenho baseado em modelos caóticos

João Inácio Moreira Bezerra

Pelotas, 2022

João Inácio Moreira Bezerra

Um novo algoritmo de criptografia de alto desempenho baseado em modelos caóticos

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Matemática da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Mestre em Modelagem Matemática

Orientador: Prof. Dr. Alexandre Molter
Coorientador: Prof. Dr. Vinícius Valduga de Almeida Camargo

Pelotas, 2022

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

B574n Bezerra, João Inácio Moreira

Um novo algoritmo de criptografia de alto desempenho baseado em modelos caóticos / João Inácio Moreira Bezerra ; Alexandre Molter, orientador ; Vinícius Valduga de Almeida Camargo, coorientador. — Pelotas, 2022.

108 f. : il.

Dissertação (Mestrado) — Programa de Pós-Graduação em Modelagem Matemática, Instituto de Física e Matemática, Universidade Federal de Pelotas, 2022.

1. Criptografia. 2. Caos. 3. Desempenho. 4. Segurança. I. Molter, Alexandre, orient. II. Camargo, Vinícius Valduga de Almeida, coorient. III. Título.

CDD : 511.8

Elaborada por Ubirajara Buddin Cruz CRB: 10/901

UM NOVO ALGORITMO DE CRIPTOGRAFIA DE ALTO DESEMPENHO BASEADO EM MODELOS CAÓTICOS

por

João Inácio Moreira Bezerra

Dissertação submetida ao Programa de Pós-Graduação em Modelagem Matemática, PPGMMat, do Instituto de Física e Matemática da Universidade Federal de Pelotas, como parte dos requisitos necessários para a obtenção do Título de

Mestre em Modelagem Matemática

Banca Examinadora:

Prof. Dr. A – Alexandre Molter (UFPel)

Prof. Dr. B – Vinícius Valduga de Almeida Camargo (Universidade onde trabalha)

Prof. Dr. - Leslie Darien Pérez Fernandes (UFPel)

Prof. Dr. - Rafael Iankowski Soares (UFPel)

Profa. Dra. - Everton Granemann Souza (UCPel)

Pelotas, 24 de novembro de 2021

Dedico este trabalho a minha família que sempre me deu todo o apoio do mundo, em especial ao meu pai José Amilton, a minha mãe Roseane, a minha irmã Elisa, a minha avó Rosenda e a Deus.

AGRADECIMENTOS

Agradeço aos meus pais, José Amilton e Roseane, por sempre terem me fornecido todo o carinho do mundo a apoio psicológico durante esta e todas as caminhadas da minha vida. Agradeço também a minha maninha Elisa, por ser a minha melhor amiga e companheira de todas as horas. Um agradecimento infinito vai a estes três porque, embora tristes pela situação do mundo, por vermos tantas pessoas morrendo, passamos estes quase dois anos de pandemia juntos em casa sempre se apoiando e tornando a situação a mais fácil possível.

Agradeço a minha avó Rosenda por todo o carinho e cuidado de sempre, assim como agradeço as minhas irmãs Camila e Juliana, e ao meu cunhado Eliézer, que foi meu grande incentivador, por todo o carinho e pela torcida de sempre.

Agradeço a minha grande amiga Laura por todo o carinho e apoio, em todos os momentos, a começar da nossa grande amizade da graduação, os envios de mimos no começo da pandemia, quando ainda não estávamos vacinados, pelas conversas e apoio quando necessitei, até as visitas recentes, sou incrivelmente abençoado pela nossa amizade. Agradeço também aos amigos que mantive da graduação, Cândido, Domarys, Nicholas, Ítalo e Camila, pelo apoio durante o mestrado e a pandemia.

Agradeço as amizades que fiz no mestrado, Tatiane e Daiane, tem sido um prazer muito grade trabalhar e ter sido colega de vocês.

Agradeço ao meu orientador Alexandre e ao meu coorientador Vinícius por toda a ajuda durante esta pesquisa que vem desde a graduação, sem a condução de vocês nunca teríamos alcançado os resultados que alcançamos.

Agradeço ao professor Marat e a professora Rejane por todo o auxílio na minha caminhada na pesquisa. Agradeço ao professor Leslie pelo auxílio na pesquisa e também pelas excelentes contribuições para o desenvolvimento deste trabalho. Pelas mesmas contribuições, agradeço aos professores Everton e Rafael.

Por fim, agradeço a UFPEL por toda a formação pessoal e profissional que me forneceu desde a graduação, e pelo ensino PÚBLICO nota 10 que fornece a todos.

A questão nunca é se você vai sobreviver. É as coisas bonitas que lhe aguardam quando você sobrevive. — CHANEL MILLER

RESUMO

MOREIRA BEZERRA, João Inácio. **Um novo algoritmo de criptografia de alto desempenho baseado em modelos caóticos**. 2022. 111 f. Dissertação (Mestrado em Modelagem Matemática) – Programa de Pós-Graduação em Modelagem Matemática, Instituto de Física e Matemática, Universidade Federal de Pelotas, Pelotas, 2022.

Atualmente, a internet é o principal canal para comunicação entre duas partes. Como a internet é um canal público, a importância da criptografia para proteger a comunicação é crescente. Os arquivos mais usados nessa comunicação são as imagens, para as quais os cifradores tradicionais não são apropriados, por causa do tamanho dos arquivos e do alto custo computacional. Neste contexto, a teoria do caos surge como uma alternativa promissora para a cifragem de imagens, devido a sensibilidade às condições iniciais, levando ao desenvolvimento de vários cifradores caóticos para a cifragem de imagens. Inicialmente, estes cifradores eram inseguros, sendo quebrados por ataques de informação escolhida, em virtude da não dependência da imagem original nas sequências caóticas, mas nos últimos anos, com esta dependência, a segurança melhorou consideravelmente. No entanto, do ponto de vista de desempenho, os cifradores caóticos ainda podem ser melhorados com uma escolha melhor de sistemas caóticos junto com pequenas mudanças na arquitetura dos cifradores. Neste trabalho, primeiro é feita uma comparação de modelos caóticos, de forma a escolher o modelo mais condutível ao alto desempenho do cifrador, e então, três versões do cifrador são propostas, a primeira com permutação-difusão separada, outra simultânea, e por fim, uma versão para aplicações na telemedicina. Testes extensivos foram realizados e mostraram que o cifrador é seguro contra várias formas de ataque, como estatístico, diferencial, ruído, oclusão e informação escolhida, além de apresentar desempenho superior em relação a outros trabalhos propostos na literatura.

Palavras-chave: Criptografia, Caos, Desempenho, Segurança.

ABSTRACT

MOREIRA BEZERRA, João Inácio. **A new high performance encryption algorithm based on chaotic models**. 2022. 111 f. Dissertação (Mestrado em Modelagem Matemática) – Programa de Pós-Graduação em Modelagem Matemática, Instituto de Física e Matemática, Universidade Federal de Pelotas, Pelotas, 2022.

Nowadays, the internet is the main channel for communication between two parties. Since the internet is a public channel, the importance of cryptography to protect the communication is increased. The most commonly shared archives are images, for which traditional ciphers are not suitable due to large file sizes and high computational cost. In this context, chaos theory surges as a promising alternative for image encryption due to its sensibility to initial conditions, leading to the development of many chaotic ciphers for image encryption. Initially, chaotic ciphers were unsafe, being broken by chosen plain text attacks, as there was no relation between the chaotic sequences generated and the original plain image, but in recent years, this relation has been established and thus the security has improved markedly. However, from the performance standpoint, chaotic ciphers can be improved with a better choice of chaotic systems coupled with small architecture adjustments. This work starts first with a comparison of chaotic systems made to choose the one that conduces to the highest performance of the cipher, and then, three versions of the new cipher are proposed one with separate permutation-diffusion, other with simultaneous, and the final version for telemedical applications. Extensive tests made show that the cipher is secure against various forms of attacks, such as statistical, differential, noise, occlusion and chosen-plaintext, while also having superior performance to other works in literature.

Keywords: Chaos, Cryptography, Performance, Security.

LISTA DE FIGURAS

1 Razão entre o número de dispositivos conectados à internet e a população mundial. Fonte: SAFAEI et al. (2017) 16
2 Formato genérico de um algoritmo criptográfico. Fonte: OLIVEIRA (2011). 17
3 Estrutura de uma rodada do AES. Fonte: ZHANG; TANG (2018). . . 24
4 Representação da operação de adição de chaves. Fonte: DOOLEY (2018). 25
5 Representação da operação de substituição de *bytes*. Fonte: DOOLEY (2018). 25
6 Representação da operação de deslocamento de linhas. Fonte: DOOLEY (2018). 26
7 Representação da operação de mistura de colunas. Fonte: DOOLEY (2018). 26

8 Comportamento de um sistema em que as iterações tendem a um único valor final. 35
9 Comportamentos do mapa logístico: (a) $r = 2,5$, (b) $r = 3,1$, (c) $r = 3,5$, (d) $r = 3,99$. 36
10 Diagrama de bifurcações para o mapa logístico. 36
11 Representação da separação exponencial de duas trajetórias inicialmente separadas por δ_0. Adaptado de QIAO (2021). 37
12 Expoente de Lyapunov para o mapa logístico. 39
13 Diferença entre as iterações de x e de y (a) $r = 2,5$ e (b) $r = 3,99$. . 39
14 Mapas de retorno. (a) $f^{(1)}(x)$. (b) $f^{(2)}(x)$. (c) $f^{(3)}(x)$. (d) $f^{(4)}(x)$. . . . 41
15 (a) Diagrama de bifurcações para a equação (13), e (b) Expoentes de Lyapunov. 41
16 Expoente de Lyapunov, de uma sequência gerada pela equação (13), com a aplicação da função de quantização dada pela equação (14). 43
17 (a) Número de vezes que cada região da equação (13) é visitada. (b) Número de vezes que cada região é visitada, após a aplicação da função de discretização dada por (14). 45

18 Fluxograma para o processo de cifragem do algoritmo proposto. . . 54
19 (a) Baboon. (b) Lena. (c) Iris. (d) Preta. (e) Branca. 58

20 (a) Black. (b) Black cifrada. (c) Black recuperada. (d) Histograma da imagem black original. (e) Histograma da imagem black cifrada. (f) Histograma da imagem black recuperada. (g) Lena. (h) Lena cifrada. (i) Lena recuperada. (j) Histograma da imagem Lena original. (k) Histograma da imagem Lena cifrada. (l) Histograma da imagem Lena recuperada. 60
21 (a) Imagem Baboon. (b) Correlação horizontal da imagem original. (c) Correlação vertical da imagem original. (d) Correlação diagonal da imagem original. (e) Imagem Baboon cifrada. (f) Correlação horizontal da imagem cifrada. (g) Correlação vertical da imagem cifrada. (h) Correlação diagonal da imagem cifrada. 65
22 Imagem Baboon com decifragem insatisfatória ao usar as sequências derivadas da cifragem de uma imagem diferente. 68
23 (a) Ruído *salt and pepper* com densidade de 0,005. (b) Imagem recuperada de a (a). (c) Ruído *speckle* com densidade de 0,000002. (d) Imagem recuperada de (c). (e) Ruído *gaussian* com densidade de 0,000001. (f) Imagem recuperada de (e). (g) Imagem 25% oculta (h) Imagem recuperada de (g). 69

24 Fluxograma para o algoritmo do cifrador de permutação-difusão simultânea. 77
25 (a) Baboon. (b) Lena. (c) Pepinos. (d) Casa. (e) Black. 78

26 Fluxograma do algoritmo proposto para cifragem de imagens médicas. 90
27 a) Cérebro. b) Ligamento cruzado anterior. c) Pescoço. d) Pedra nos rins. 92

LISTA DE TABELAS

1 Relação entre as propriedades da Teoria do Caos e da Criptografia. Fonte:ALVAREZ; LI (2006) 27

2 Comparação do tempo para iterar os mapas (13) e (14) com o tempo necessário para resolver o sistema de Lorenz. 46

3 Comparação do tempo necessário para resolver 4 conjuntos de iterações das equações (13) e (14) com o tempo necessário para resolver o sistema hiper-caótico de Lorenz. 46

4 Testes do NIST para as sequências caóticas gerada pelas equações (13),(14) e (18). 48

5 Testes numéricos para medir a uniformidade do histograma das imagens cifradas. 59

6 Testes do NIST para as imagens cifradas. 61

7 Entropia de Shannon para a imagem cifrada. 62

8 Comparações do espaço de chaves do cifrador proposto com outros trabalhos da literatura 62

9 Comparação de coeficientes de correlação para pixels adjacentes. . 64

10 Média dos Coeficientes de Correlação. 64

11 NPCR para as imagens cifradas, medindo sensibilidade à mudança na imagem (IS) e à chave (KS). 66

12 UACI para as imagens cifradas, medindo sensibilidade à mudança na imagem (IS) e à chave (KS). 66

13 Valores de MSE para as imagens cifradas 67

14 Comparação dos valores de MSE. 67

15 Valores de PSNR para ruídos SP (Salt and Pepper), S (Speckle), G (Gaussian) e O (Occlusion). 69

16 Valores de SSIM para ruídos SP (Salt and Pepper), S (Speckle), G (Gaussian) e O (Occlusion). 70

17 Tempo necessário para cifrar uma imagem, em milissegundos. . . . 71

18 Comparação de *throughput*. 71

19 Testes estatísticos para as imagens cifradas. 79

20 Testes do NIST para as imagens cifradas. 79

21 Coeficientes de correlação das imagens cifradas. 80

22 Testes para medir a resistência do cifrador a ataques diferenciais, com I representando sensibilidade à mudanças na imagem e K representando sensibilidade à mudanças na chave. 80

23 SSIM usado para comparar a imagem original com a imagem recuperada utilizando o ataque da informação escolhida. 81
24 PSNR para as imagens recuperadas quando a respectiva imagem cifrada é poluída com ruído. 81
25 SSIM para as imagens recuperadas quando a respectiva imagem cifrada é poluída com ruído. 82
26 Comparação do tempo necessário para cifrar uma imagem em preto e branco de dimensões 256×256, no MATLAB. 82
27 Tempo necessário para a cifragem e a decifragem de imagens em preto e branco de diferentes dimensões. 83

28 Métricas estatísticas para a semelhança entre a imagem original e a recuperada. 93
29 Coeficientes de correlação para a imagem cifrada. 94
30 Valores dos coeficientes de Pearson, NPCR e UACI para a imagem cifrada. 94
31 Tempo de execução e *throughput* para o cifrador proposto. 95

LISTA DE ABREVIATURAS E SIGLAS

AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Exchange
CBC	Cipher Block Chaining
DES	Data Encryption Standard
ECC	Eliptic Curve Cryptography
FFT	Fast Fourier Transform
GPU	Graphisc Processing Unit
IDEA	International Data Encryption Algorithm
IEEE	Institute of Electrical and Electronics Engineers
IOT	Internet of Things
LSB	Least Significant Bits
MB	Mega-Bytes
MSE	Mean Square Error
NaN	Not a Number
NIST	National Institute of Security and Technology
NPCR	Number of Pixels Rate Change
UACI	Unified Average Changing Intensity
PSNR	Peak-Signal-to-Noise Ratio
QR	Quick Response
RAM	Random Access Memory
RSA	Rivest-Shamir-Adleman
SHA	Secure Hash Algorithm
SSIM	Structural Similarity Index Measure
XOR	Exclusive or

SUMÁRIO

1 INTRODUÇÃO 16
1.1 Criptografia 17
1.1.1 Hashing 19
1.2 Criptoanálise 21
1.3 Contextos histórico e atual da criptografia 22
1.3.1 Advanced Encryption Standard (AES) 23
1.3.2 Contexto atual: cifragem de Imagens 26
1.4 Objetivos e organização do trabalho 29

2 SISTEMAS DE DINÂMICA DISCRETA E TEORIA DO CAOS 31
2.1 Conceitos de sistemas de dinâmica discreta 31
2.1.1 Análise de estabilidade de sistema de tempo discreto 32
2.2 Teoria do Caos 33
2.3 Diagrama de bifurcações 34
2.4 Expoentes de Lyapunov 36
2.5 Mapa da tenda 39
2.5.1 Robustez 42
2.5.2 Densidade uniforme invariante 43
2.5.3 Eficiência computacional 44
2.5.4 Degradação dinâmica 46

3 ALGORITMO PARA CIFRAGEM DE IMAGENS 49
3.1 Revisão bibliográfica 49
3.2 Descrição do cifrador 51
3.2.1 Cifragem 51
3.2.2 Decifragem 53
3.3 Análise de resultados 58
3.3.1 Análise de histograma 59
3.3.2 Testes do NIST 60
3.3.3 Entropia de Shannon 61
3.3.4 Espaço de chaves 61
3.3.5 Coeficientes de correlação 63
3.3.6 Ataque diferencial 64
3.3.7 Efeito avalanche 67
3.3.8 Ataque de texto original conhecido e escolhido 67
3.3.9 Resistência à ruído 68
3.3.10 Desempenho e complexidade 70

4 CIFRADOR COM PERMUTAÇÃO-DIFUSÃO SIMULTÂNEA 72
4.1 Motivação 72
4.2 Algoritmo 74
4.2.1 Cifragem 74
4.2.2 Decifragem 76
4.3 Análise de segurança 77
4.3.1 Ataques estatísticos 78
4.3.2 Ataque diferencial 79
4.3.3 Ataques da informação escolhida e força bruta 80
4.3.4 Ataques de ruído 81
4.4 Análise de desempenho 82
4.5 Comparação com o AES 83

5 ALGORITMO PARA APLICAÇÕES NA TELEMEDICINA 85
5.1 Revisão bibliográfica 85
5.2 Descrição 88
5.2.1 Inserção dos dados textuais 90
5.2.2 Cifragem 91
5.2.3 Decifragem 91
5.2.4 Remoção dos dados textuais 92
5.3 Resultados 92
5.3.1 Similaridade estrutural 92
5.3.2 Análise de segurança 93
5.3.3 Análise de performance 94

6 CONCLUSÕES 96

REFERÊNCIAS 97

7 ANEXO - PRODUÇÃO CIENTÍFICA 109

1 INTRODUÇÃO

Nas últimas décadas, em virtude dos avanços tecnológicos, a maioria das comunicações ocorre em redes públicas, como é o caso da internet (MURILLO-ESCOBAR et al., 2015). Em 2003, a razão entre o número de dispositivos conectados à internet e a população mundial era de $0,08$ (EVANS, 2011), enquanto em 2010, em virtude da popularização do uso de computadores pessoais, essa razão passou a ser de $1,83$ (SAFAEI et al., 2017). No começo da década passada, também ouve a popularização do uso de *smart-phones* e o desenvolvimento da Internet das Coisas (IoT), que é a integração de pequenos dispositivos equipados com microcontroladores e transceptores na internet (ATZORI; IERA; MORABITO, 2010; NAUMAN et al., 2020). Dessa forma, a expectativa é que esta taxa de dispositivos conectados à internet cresça de maneira exponencial, como mostra a Figura 1.

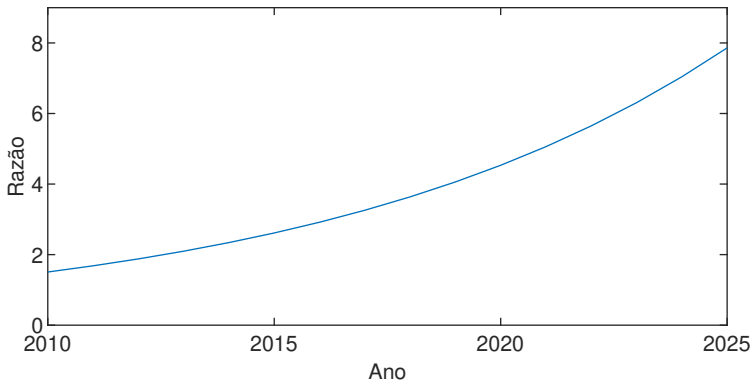


Figura 1: Razão entre o número de dispositivos conectados à internet e a população mundial. Fonte: SAFAEI et al. (2017)

Dados transmitidos em redes públicas como a internet estão sujeitos a interceptação por partes não autorizadas, o que faz com que a criptografia seja necessária para proteger estes dados. Neste capítulo, é feita uma revisão de conceitos relacionados a criptografia, incluindo seu contexto histórico e atual.

1.1 Criptografia

A área científica que estuda técnicas para codificar mensagens, que sejam transmitidas de maneira segura é denominada criptografia (PAAR; PELZL, 2010). Um algoritmo criptográfico normalmente é dividido nos dois passos que seguem:

- **Cifragem:** processo pelo qual uma mensagem é codificada, de forma a ficar inelegível para um atacante.
- **Decifragem:** processo pelo qual a mensagem codificada é decodificada pelo receptor autorizado.

Durante a cifragem, a informação original é submetida a uma combinação de método de cifragem e chave criptográfica, enquanto que durante a decifragem a informação cifrada é combinada com a chave, aplicando a operação inversa da cifragem, para recuperar a informação original, como mostra a Figura 2.

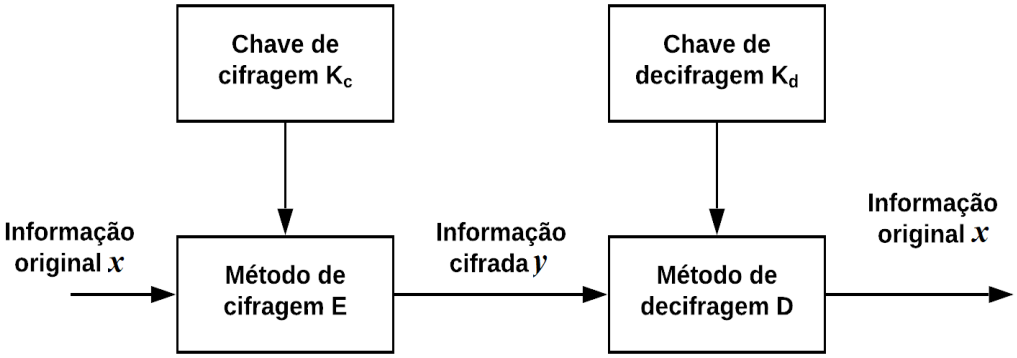


Figura 2: Formato genérico de um algoritmo criptográfico. Fonte: OLIVEIRA (2011).

De acordo com o formato das chaves, o cifrador possui duas classificações (PAAR; PELZL, 2010), que seguem:

- **Chave simétrica:** em um cifrador de chave simétrica, a chave utilizada é a mesma na cifragem e na decifragem, sendo também chamados de cifradores de chave privada.
- **Chave assimétrica:** em um cifrador de chave assimétrica, um par de chaves é utilizado, uma delas sendo pública e podendo ser utilizada por qualquer pessoa que queira enviar uma mensagem. Por isso, estes cifradores também são chamados de cifradores de chave pública. A chave privada, por sua vez, é de posse apenas de partes autorizadas, sendo utilizada para decifrar uma mensagem. Qualquer mensagem cifrada utilizando uma chave pública pode ser decifrada apenas utilizando a chave privada equivalente, com a aplicação do mesmo algoritmo (AL-JABALI, 2016).

Tradicionalmente, grande parte dos algoritmos são de chave privada. O problema destes algoritmos é quanto à segurança na troca de chaves, existindo o risco de que uma parte não autorizada se aproprie da chave. Dessa forma, no trabalho de DIFFIE; HELLMAN (1976) foi proposto o primeiro algoritmo de chave pública, o *Rivest-Shamir-Adleman* (RSA). Em virtude de serem consideravelmente mais lentos que os algoritmos de chave privada (PEREIRA, 2008), os algoritmos de chave pública não costumam ser utilizados para cifrar arquivos, mas sim para fornecer autenticidade de um documento, por meio da assinatura digital, e ainda para a troca de chaves simétricas, assim garantindo que elas não sejam tomadas por uma parte não autorizada.

Para um algoritmo criptográfico ser seguro, dois requisitos são essenciais, que são a confusão e a difusão (SHANNON, 1949):

- **Confusão:** a saída deve ser uniforme, ou seja, cada caractere, no caso de um texto, ou cada pixel, no caso de uma imagem, deve aparecer o mesmo número de vezes na informação cifrada.
- **Difusão:** uma pequena mudança na informação original deve gerar uma mudança significativa na informação cifrada.

Além da definição dada por SHANNON (1949), confusão e difusão são definidas de outras formas, mas gerando um entendimento similar. A seguir, estas abordagens são apresentadas:

- Em SCHNEIER (2007) e em (PAAR; PELZL, 2010), confusão é definido como o processo que esconde relações entre o arquivo original e o arquivo cifrado, enquanto difusão é o processo que dissipa a redundância do arquivo original por todo o arquivo cifrado.
- Em SWENSON (2008), confusão é definida como a dificuldade de entender como foi gerada a informação cifrada, enquanto difusão é definida como a influência que a informação original tem na informação cifrada.
- Em AUMASSON (2017), confusão está relacionada com o fato de o arquivo cifrado ser gerado a partir de transformações complexas na informação original em conjunto com as chaves, enquanto a difusão estabelece que essas transformações dependem de todas as partes da informação original e da chave igualmente.
- Em DOOLEY (2018), o conceito de difusão está ligado ao efeito avalanche, com uma mudança de um *bit* no arquivo original ou na chave afetando todos os outros *bits* da informação cifrada, enquanto confusão é o processo de tornar a relação entre a informação original e a cifrada a mais complexa possível.

1.1.1 Hashing

Um conceito bastante importante para a criptografia é o conceito de *hashing*. Para garantir a proteção de um cifrador caótico contra ataques, há a necessidade de fazer com que as sequências caóticas geradas sejam dependentes do arquivo original, e para isso, usam-se funções matemáticas *hash*. Estas funções geram uma sequência de *bits*, ou seja, uma sequência composta por zeros (0) e uns (1), como por exemplo, 100001110101, que é usada para modificar as condições inicias de um mapa caótico, por exemplo, considerando que uma leve mudança na imagem original gera uma mudança significativa na *hash*, assim gerando sequências diferentes para imagens diferentes. Ademais, em virtude dessa sensibilidade da *hash* em relação à imagem original, essas funções também podem ser usadas para garantir a autenticidade da informação recuperada, como assinatura digital, garantindo ser igual à informação original, com a aplicação da *hash* na imagem recuperada, resultando na mesma sequência recebida para perturbar as chaves das sequências caóticas. A probabilidade de que estas funções sejam invertidas é consideravelmente baixa (AL-JABALI, 2016).

O algoritmo de *hashing* mais usado é o *Secure Hash Algorithm* (SHA) (EASTLAKE; JONES, 2001; PADHI; CHAUDHARI, 2017; DOOLEY, 2018), utilizado como padrão pelo Instituto Nacional de Segurança dos Estados Unidos (NIST). A seguir, é realizada a descrição do algoritmo (ABDOUN, 2019), que gera oito valores de oito caracteres hexadecimais $H_{1:8}$, que formam a *hash* de 256 *bits*, a partir de um arquivo M, denominado mensagem.

Inicialmente, os valores de H são inicializados de acordo com a parte fracionária da raiz quadrada dos oito primeiros números naturais primos, da seguinte forma:

$$H_1(0) = 6a09e667 \qquad H_2(0) = bb67ae85$$
$$H_3(0) = 3c6ef372 \qquad H_4(0) = a54ff53a$$
$$H_5(0) = 510e527f \qquad H_6(0) = 9b05688c$$
$$H_7(0) = 1f83d9ab \qquad H_8(0) = 5be0cd19$$

Após a inicialização, a mensagem é preparada, tendo seu tamanho em *bits* estendido para um valor que esteja a 64 *bits* de ser um múltiplo de 512. Nesta situação, desconta-se 64 *bits* pois estes serão inseridos no final da mensagem, contendo o tamanho da mensagem. Como exemplo, supõe-se que o tamanho de M seja dado por l, então um *bit* 1 é inserido no final de M, junto com mais v *bits* zero, sendo que v é o menor número natural que seja solução da seguinte equação:

$$l + 1 + v \equiv 448 \mod 512$$

Por exemplo, a mensagem abc tem tamanho de 3 caracteres, que correspondem a

24 *bits*. Então, um *bit* 1 e mais $v = 448 - (24 + 1) = 423$ *bits* zero seriam inseridos na mensagem. Após isto, o tamanho da mensagem é um múltiplo de 512, e esta mensagem é dividida em N blocos de 512 *bits*, denominados $M(1), M(2), ..., M(N)$.

Depois que a mensagem é estendida, realizam-se as operações para gerar a *hash*. A dependência da *hash* em relação a mensagem original se dá a partir da variável $W_{1:15}$, que durante o processo, é usada para modificar as variáveis de a até h, que são continuamente acrescidas à $H_{1:8}$ até gerar a *hash*. Abaixo, o processo é mostrado de acordo com o algoritmo que segue, com operações realizadas com valores de 32 *bits*, que tem a saída H como o resultado da *hash*, que é dada pela concatenação de oito valores de 32 *bits*, assim apresentando 256 *bits*.

for $i \leftarrow 1 : N$ **do**
  $a \leftarrow H_1(i-1)$
  $b \leftarrow H_2(i-1)$
  $\cdots$
  $h \leftarrow H_8(i-1)$
  for $j \leftarrow 0 : 15$ **do**
    $W_j \leftarrow M_j(i)$
  end for
  for $j \leftarrow 16 : 73$ **do**
    $W_j \leftarrow \sigma 1(W_{j-2}) + W_{j-7}\sigma 0(W_{j-15}) + W_{j-16}$
  end for
  for $j \leftarrow 0 : 63$ **do**
    Compute $C_h(e,f,g), Maj(a,b,c), \sum 0(a), \sum 1(e), W_j$
    $T_1 \leftarrow h + \sum 1(e) + C_h(e,f,g) + K_j + W_j$
    $T_2 \leftarrow \sum 0(a) + Maj(a,b,c)$
    $h \leftarrow g$
    $g \leftarrow f$
    $f \leftarrow e$
    $e \leftarrow d + T_1$
    $d \leftarrow c$
    $c \leftarrow b$
    $b \leftarrow a$
    $a \leftarrow T_1 + T_2$
  end for
  $H_1(i) \leftarrow a + H_1(i-1)$
  $H_2(i) \leftarrow b + H_2(i-1)$
  $\cdots$
  $H_8(i) \leftarrow h + H_8(i-1)$
end for

$H = H_1(N)H_2(N)H_3(N)H_4(N)H_5(N)H_6(N)H_7(N)H_8(N)$

Neste algoritmo, tem-se que K_j é dado pelos primeiros 32 *bits* da parte fracionária da raiz cúbica dos primeiros 64 números primos. As outras funções utilizadas para o algoritmo são definidas a seguir.

$$
\begin{aligned}
Ch(x, y, z) &= (x \wedge y) \oplus (\neg x \wedge z) \\
Maj(x, y, z) &= (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \\
\sum 0(x) &= S^2(x) \oplus S^{13}(x) \oplus S^{22}(x) \\
\sum 1(x) &= S^6(x) \oplus S^{11}(x) \oplus S^{25}(x) \\
\sigma 0(x) &= S^7(x) \oplus S^{18}(x) \oplus R^3(x) \\
\sigma 1(x) &= S^{17}(x) \oplus S^{19}(x) \oplus R^{10}(x)
\end{aligned}
$$

Em que $\oplus$ representa a operação lógica XOR(*Exclusive-or*), que tem como resultado 0 se os *bits* comparados são iguais e 1 caso sejam diferentes, $\wedge$ representa a operação lógica de conjugação, que entre dois *bits* tem resultado igual à 1 apenas caso ambos os *bits* sejam 1, e $\neg$ representa a operação lógica de negação, que na prática significa que os valores dos *bits* são invertidos, ou seja, a negação de 0 é 1 e a negação de 1 é 0. Além disso, a operação $S^n(x)$ representa a rotação a direita em n *bits* da sequência de *bits* e $R^n(x)$ representa o deslocamento a direita em n *bits*.

1.2 Criptoanálise

A criptoanálise, segundo PAAR; PELZL (2010), é ciência de recuperar uma informação *x* a partir de uma informação cifrada *y*, por intermédio das mais variadas técnicas de ataque. As técnicas são as que seguem:

- **Ataque à informação cifrada:** forma de ataque mais básica, em que ocorre a interceptação da informação cifrada, e com isso, busca-se recuperar a informação original, e se possível, a chave.
- **Ataque da informação original:** busca-se decodificar a informação original usando a informação cifrada, junto com amostras anteriores da informação original em conjunto com a respectiva informação cifrada. Neste caso, o atacante sabe qual a informação correspondente a algumas cifras, de forma que possa ser possível deduzir a chave, de acordo com essas informações.
- **Ataque da informação original provável:** forma de ataque que se baseia na frequência dos *bits* da informação cifrada. Por exemplo, se a informação original estiver codificada de acordo com o *American Standard Code for Information Exchange* (ASCII), certos *bits* irão aparecer com mais frequência. Caso um *e-mail*

seja interceptado, os primeiros caracteres podem ser relativamente fáceis de se descobrir, tendendo a representar o remetente, data ou título, ou ainda quando se sabe que uma informação está em um certo idioma, certos caracteres, como as vogais, tendem a aparecer com mais frequência, assim podendo gerar uma associação entre uma vogal e o respectivo caractere cifrado.

- **Ataque da informação original escolhida:** uma das formas de ataque menos realista, mas mais forte. Funciona de maneira semelhante ao ataque da informação original. Contudo, neste caso, o atacante pode escolher a informação a ser cifrada. Neste ataque, busca-se cifrar informações com certas propriedades, com o intuito de alterar a informação cifrada, para tentar descobrir a chave.
- **Ataque da informação cifrada escolhida:** há a tentativa de decifrar um arquivo inelegível com certas chaves. É a forma de ataque menos realista, funcionando de forma bastante semelhante ao ataque da informação original escolhida.

Os ataques podem também ser classificados quanto ao tipo, como segue.

- **Força Bruta:** tipo de ataque padrão, no caso do atacante ter acesso à informação, seja o arquivo original ou cifrado. Ocorre o teste com todas as combinações possíveis para formar a chave, até a informação de interesse ser obtida. Sempre irá obter o resultado desejado, embora possa, muitas vezes, levar décadas.
- **Criptoanálise Linear:** neste ataque, tem-se conhecimento da informação original que é cifrada, e o objetivo é relacionar a informação original com a informação cifrada, com o objetivo de formar um sistema de equações lineares que forneça informações sobre a chave.
- **Criptoanálise Diferencial:** neste ataque, também tem-se conhecimento da informação original, e o objetivo é encontrar uma relação nas mudanças que ocorrem nos *bits* da saída quando a informação de entrada é modificada.

Os conceitos de confusão e difusão estão altamente relacionados com a resistência de um cifrador a estes ataques. Quanto à confusão, a saída uniforme impede ataques estatísticos como os ataques da informação original, cifrada, e original provável. Quanto à difusão, uma pequena mudança no arquivo original ou na chave gera uma mudança considerável no arquivo cifrado, assim aumentando a dificuldade em estabelecer relações entre o arquivo original e o cifrado, impedindo os ataques de informação escolhida, além da criptoanálise linear e diferencial.

1.3 Contextos histórico e atual da criptografia

A ciência da proteção de dados, que é a criptografia, é utilizada desde a antiguidade. De acordo com DOOLEY (2018), os primeiros cifradores, utilizados por gregos

e romanos, usavam substituição monoalfabética, como por exemplo a Cifra de César, em que um caractere era substituído por outro caractere do mesmo alfabeto, como mostra o exemplo que segue.

Texto original: abcdefghijklmnopqrstuvwxyz

Texto cifrado: defghijklmnopqrstuvwxyzabc

Depois deste cifrador, vieram os poli-alfabéticos, como o de Viginére, usando também aritmética modular, associando um número a cada caractere do alfabeto, e usando operações modulares entre o caractere original e o caractere que se comportava como chave. Este cifrador foi quebrado apenas após 350 anos de uso. O advento da criptografia moderna se deu na Primeira Guerra Mundial, com o surgimento da comunicação à rádio, e no período entre guerras houve o advento de algoritmos de máquina, como o Enigma, o Typex e o SIGABA.

Após a Segunda Guerra Mundial, com o desenvolvimento computacional, o Instituto Nacional de Padrões e Tecnologias (NIST), dos Estados Unidos, teve o desejo de único cifrador para todas as aplicações no país, e o escolhido para isso foi o *Data Encryption Standard* (DES), desenvolvido pela IBM (PAAR; PELZL, 2010). O DES é um cifrador simétrico que cifra blocos de 64 *bits* com uma chave de 56 *bits*, utilizando operações complexas em conjunto com modos de operação (PAAR; PELZL, 2010). Para gerar difusão, é realizada uma operação *Exclusive-or* (XOR) entre a informação cifrada no passo anterior e a informação cifrada no passo atual.

Por causa do tamanho de chave do DES e do desenvolvimento de computadores modernos de alta capacidade de processamento, o algoritmo passou a ser facilmente quebrado por ataques de força bruta, e em virtude disso, o *Advanced Encryption Standard* (AES) surgiu como substituto do DES. Como chaves de 128, 192 e 256 *bits*, o AES é seguro contra ataques de força bruta, sendo atualmente o algoritmo recomendado pelo NIST. A difusão se dá mediante o uso de caixas de substituição, que atribuem um elemento equivalente a cada elemento da informação a ser cifrada, e para a cifragem são necessárias entre 10 e 14 rodadas.

1.3.1 Advanced Encryption Standard (AES)

O AES é um cifrador simétrico de bloco de 128 *bits*, ou seja, a entrada do cifrador é um bloco composto por uma sequência de 128 *bits*, 16 números inteiros entre 0 e 255, e o usuário tem opção de três tamanhos de chave, de 128, 192 e 256 *bits* (DOOLEY, 2018). O número de rodadas varia de acordo com o tamanho da chave, da seguinte forma:

- Com chave de 128 *bits*, o AES apresenta 10 rodadas.
- Com chave de 192 *bits*, o AES apresenta 12 rodadas.
- Com chave de 256 *bits*, o AES apresenta 14 rodadas.

Durante a cifragem, o arquivo é dividido em múltiplas matrizes de dimensão 4×4, e o conjunto destas matrizes é denominado Estado. Na Figura 3, a estrutura de uma rodada de operação do AES é mostrada, com as operações sendo aplicadas sobre cada Estado gerado a partir do arquivo, em que i corresponde a rodada do algoritmo. A seguir, cada uma das operações, adição de chaves, que é representada pelo sinal $\bigoplus$ na Figura 3, substituição de *bytes*, deslocamento de linhas e mistura de colunas, são explicadas. Na última rodada, não há a operação de mistura de colunas. Na Figura 3, e nas demais figuras que envolvem as ilustrações das operações, as variáveis são apenas ilustrativas, pois o AES não é implementado neste trabalho. Ademais, na Figura 3, tem-se que *plaintext* corresponde à informação original, *secret key* corresponde à chave e *cipher text* correponde à informação cifrada.

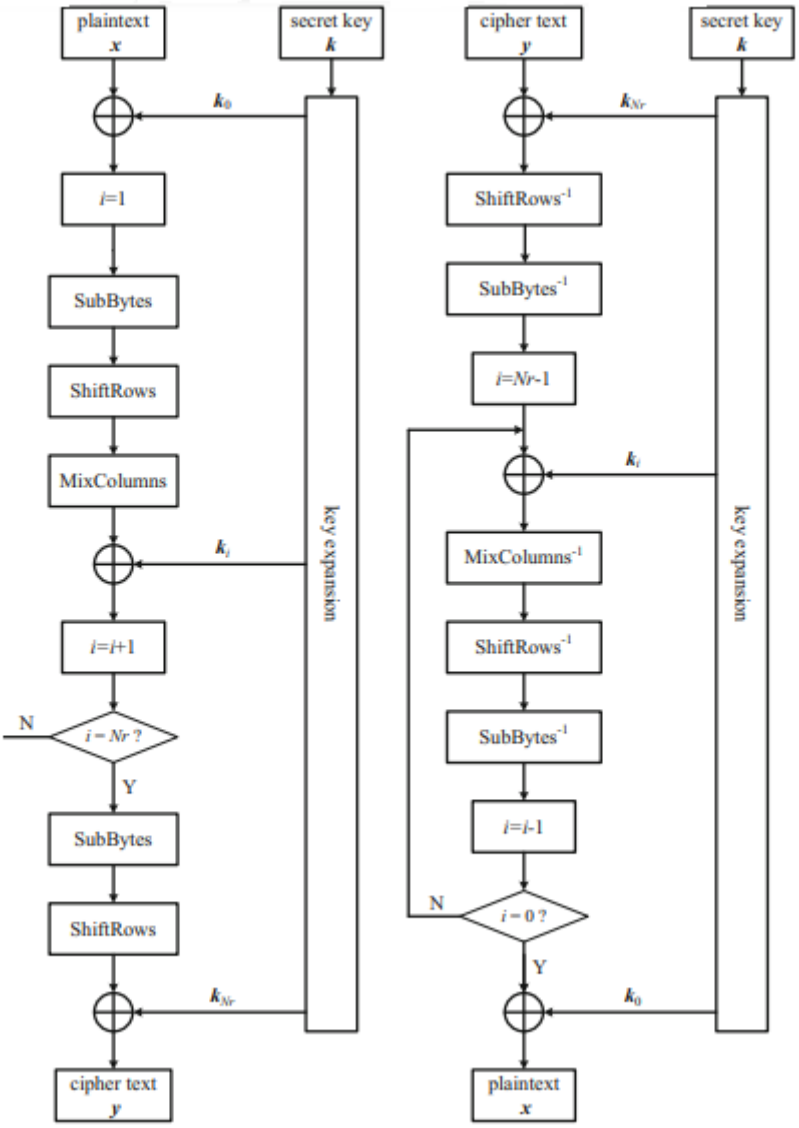


Figura 3: Estrutura de uma rodada do AES. Fonte: ZHANG; TANG (2018).

1.3.1.1 ***Adição de chaves(Key expansion)***

Nesta operação, cada coluna de cada matriz que compõe o Estado é somada com a coluna correspondente da chave expandida, que é gerada para a rodada, de acordo com a chave trocada durante o protocolo de troca de chaves. Na Figura 4, esta operação é ilustrada.

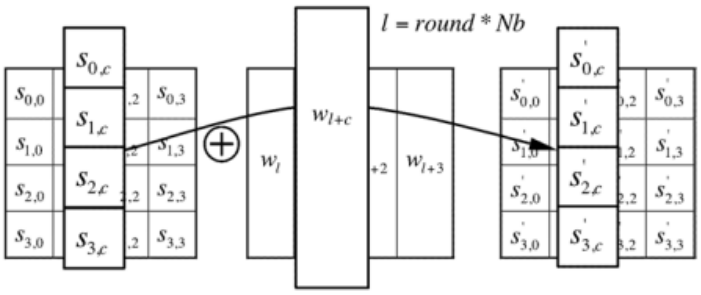


Figura 4: Representação da operação de adição de chaves. Fonte: DOOLEY (2018).

1.3.1.2 ***Substituição de bytes(SubBytes)***

Um *byte* é formado por oito *bits*, assim correspondendo a um número inteiro entre 0 e 255. Nesta operação, cada *byte* possui um *byte* equivalente em uma tabela, denominada caixa de substituição(S-BOX), e de acordo com esta tabela, o valor do *byte* é trocado, como mostra a Figura 5.

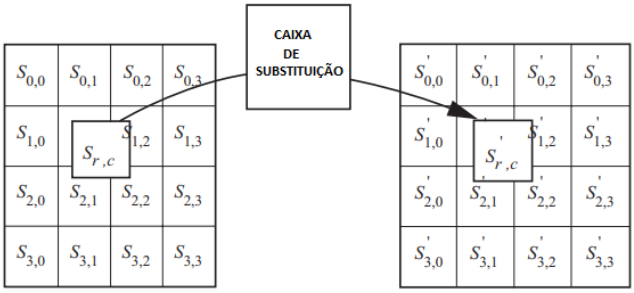


Figura 5: Representação da operação de substituição de *bytes*. Fonte: DOOLEY (2018).

1.3.1.3 ***Deslocamento de linhas(ShiftRows)***

Nesta operação, as linhas do arquivo são trocadas, de acordo com uma constante fixa de deslocamento, com a primeira coluna de cada matriz do Estado sendo deslocada 0 *bytes*, a segunda coluna 1 *byte*, a terceira 2 *bytes* e a quarta 3 *bytes*. Os *bytes* são rotacionados, de forma que eles ainda estão no Estado com os mesmos valores, só que em posições diferentes. Na Figura 6, é mostrado um resumo do funcionamento desta operação.

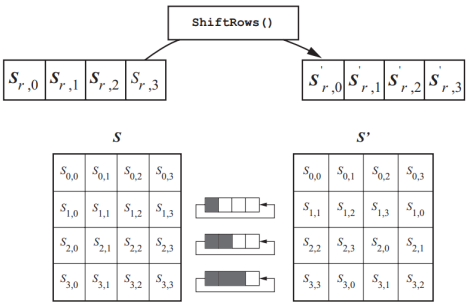


Figura 6: Representação da operação de deslocamento de linhas. Fonte: DOOLEY (2018).

1.3.1.4 ***Mistura de colunas(MixColumns)***

Nesta operação, em cada coluna de cada matriz do Estado, é aplicada uma transformação linear, que é equivalente à multiplicação de cada elemento por uma função, e com o resultado desta função, é calculado o resto inteiro em relação à 256. Na Figura 7, é ilustrado o funcionamento desta operação.

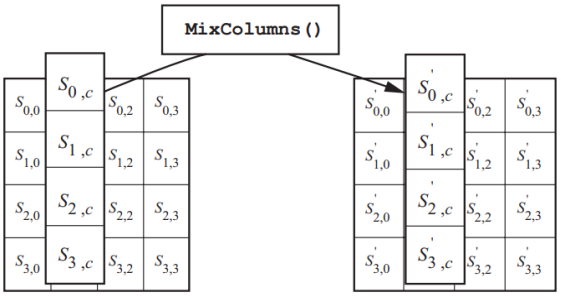


Figura 7: Representação da operação de mistura de colunas. Fonte: DOOLEY (2018).

1.3.2 Contexto atual: cifragem de Imagens

Atualmente, tem-se que imagens são os arquivos mais utilizados para a comunicação, sendo que essas imagens podem ser tanto em preto e branco, ou coloridas, e assim o algoritmo de cifragem utilizado precisa ser eficiente para a cifragem de imagens (LIAO; CHEN; YIN, 2016; FAN et al., 2018). Um arquivo de imagem tem o formato de uma matriz, em que cada elemento é denominado pixel. Uma imagem pode ser preta e branca, com apenas um canal de cor, ou ainda colorida, com os canais de cor vermelho, verde e azul.

Algoritmos tradicionais, de chave privada, como o DES(*Data Encryption Standard*), AES(*Advanced Encryption Standard*) e IDEA (*International Data Encryption Algorithm*) foram desenvolvidos focados na cifragem de textos (ZHOU; BAO; CHEN, 2014; TOUGHI; FATHI; SEKHAVAT, 2017; CHAI et al., 2019), sendo inapropriados para a cifragem de imagens devido aos fatores que seguem:

- Tamanho dos arquivos;

- Alta correlação entre pixels adjacentes;
- Alto consumo de potência (AHMED; ABD; HAMIDA, 2012; AHMAD; HWANG, 2015).

Modos de operação ajudam a lidar com a alta correlação entre pixels adjacentes, mas a estrutura do algoritmo ainda é bastante complexa, mantendo o alto consumo de potência e de tempo (ÇAVUŞOĞLU et al., 2018). Por outro lado, algoritmos de chave pública, tais como o Rivest-Shamir-Adleman (RSA) e o Eliptic Curve Cryptography (ECC) possuem complexidadade computacional e temporal ainda mais elevada que os algoritmos de chave privada (HAMZA et al., 2020), sendo usados apenas para a troca de chaves, em que a chave privada é enviada do remetente ao destinatário usando um algoritmo de chave pública. Nesse contexto, em virtude de características semelhantes à criptografia, como mostra a Tabela 1, a teoria do caos surge como uma alternativa promissora para a criptografia, atraindo considerável interesse na literatura (FARAH; FARAH; FARAH, 2019; ZHOU; TAN; YU, 2019; PENG; HE; SUN, 2021; LI et al., 2021; BEZERRA; ALMEIDA CAMARGO; MOLTER, 2021).

Propriedade Caótica	Propriedade Criptográfica	Descrição
Ergodicidade	Confusão	*A saída é uniforme independentemente da entrada.*
Sensibilidade às Condições Iniciais	Difusão	*Uma pequena mudança na entrada gera uma mudança significativa na saída.*
Dinâmicas determinísticas	Pseudo-aleatoriedade randômica	*Um processo determinístico pode gerar comportamento pseudo-aleatório.*
Estruturas complexas	Complexidade de algoritmos	*Um processo simples computacionalmente possui alta complexidade teórica.*

Tabela 1: Relação entre as propriedades da Teoria do Caos e da Criptografia. Fonte:ALVAREZ; LI (2006)

.

O interesse de pesquisadores em aplicar a teoria do caos na criptografia surgiu com a publicação do trabalho de (CUOMO; OPPENHEIM, 1993), em que foi mostrado que as órbitas de dois sistemas caóticos podem ser sincronizadas, assim sendo usadas para transmitir informações de forma privada. Em processos que requerem sincronização, denominados analógicos, há um *trade-off* entre a robustez do processo e a segurança do sistema, pois no caso de um sistema muito sensível às condições iniciais, podem ocorrer erros que impeçam a sincronização de ser alcançada, enquanto em um sistema sem esta sensibilidade, a segurança será baixa. Em virtude disso, atualmente a criptografia com caos é mais usada para dados digitais, como texto e

imagens. Um dos primeiros trabalhos de grande destaque nesta área foi o de BAPTISTA (1998), em que um mapa caótico é utilizado, sendo dividido em 128 partes, cada uma correspondente a um caractere da Tabela ASCII. Após isso, para cada caractere, a correspondência no arquivo cifrado é dada pelo número de iterações do mapa necessária para atingir a região correspondente ao caractere. Na decifragem, o número de iterações lido pelo receptor é o número de vezes que ele precisa iterar o sistema, e o caractere recuperado é o correspondente à região do resultado da última iteração do sistema. Este trabalho despertou grande interesse de pesquisadores, e algumas melhoras foram feitas à ele, como em WONG; LEE (2001), em que um número aleatório de iterações do mapa é resolvida antes da cifragem de cada caractere, para deixar o histograma uniforme, e no trabalho de WONG (2002), em que uma *look-up table* foi inserida à fim de diminuir o consumo de tempo.

Estes trabalhos foram quebrados por ALVAREZ et al. (2003); ALVAREZ; LI (2006), por causa de uma falha considerável notada no começo do estudo da criptografia caótica, que era o uso de combinações de chaves que não geravam órbitas caóticas para o mapa. Trabalhos seguintes, já envolvendo cifragem de imagens, como (PATIDAR et al., 2010; USAMA et al., 2010; WANG et al., 2010) trataram a problemática da seleção de chaves, mas foram quebrados em virtude da não dependência entre a imagem original e as sequências caóticas geradas (XIE et al., 2017; FAN et al., 2018). Em trabalhos recentes, como em (FARAH; FARAH; FARAH, 2019), esta problemática também foi solucionada, de forma que pode-se dizer que o grande objetivo atual de pesquisadores é tornar os cifradores mais rápidos, a partir da escolha do sistema caótico e na arquitetura do cifrador.

Além da cifragem pura de imagens, a relação entre teoria do caos e criptografia também vem atraindo considerável atenção na telemedicina, em que protocolos seguros são necessários para o compartilhamento de arquivos. Com a telemedicina, é possível reduzir o custo de serviços de saúde, além de ser mais conveniente aos pacientes. Em virtude da pandemia de COVID-19, com a necessidade de distanciamento social, a importância desses serviços aumenta, e isto deve continuar após a pandemia, em virtude da praticidade para os pacientes (JALALI; LANDMAN; GORDON, 2021). Em aplicações de telemedicina, normalmente, uma imagem do diagnóstico médico é compartilhada em conjunto com informações textuais do paciente, do médico responsável ou da clínica/hospital.

No contexto destes trabalhos, aproveitando a relação da teoria do caos com a criptografia, foram propostas duas versões de um cifrador caótico: uma para a cifragem pura de imagens, e outro para aplicações na telemedicina.

1.4 Objetivos e organização do trabalho

Neste trabalho, o objetivo geral é aproveitar os paralelos entre a Teoria do Caos e Criptografia, e com isso, montar dois cifradores, realizando testes para validá-lo do ponto de vista de segurança e de desempenho, assim mostrando sua viabilidade para aplicação para a cifragem pura de imagens e para a telemedicina.

Os objetivos específicos do trabalho são:

- Analisar os algoritmos de criptografia baseados em sistemas caóticos na literatura do ponto de vista de segurança e de custo computacional.
- Ilustrar a influência que a escolha do sistema caótico apresenta na redução do custo computacional do cifrador.
- Desenvolver um cifrador para cifragem pura de imagens e outro para aplicações na telemedicina, em que há a inclusão da esteganografia.
- Simular o cifrador a nível de software para validá-lo do ponto de vista de segurança e de desempenho.

Estrutura da dissertação

O trabalho está organizado em cinco capítulos, conclusões, referências bibliográficas e anexos. Além da introdução, que já foi apresentada, os demais quatro capítulos são descritos no que segue.

- No segundo capítulo, a Teoria do Caos é apresentada. Primeiro, apresentam-se as ferramentas de análise de caos, como o diagrama de bifurcações e os expoentes de Lyapunov. Depois, é apresentado o mapa caótico tenda, dado por uma função definida por partes, que será utilizado neste trabalho, com o seu caminho para o caos, além da análise do seu desempenho.
- No terceiro capítulo, trabalha-se com a cifragem pura de imagens. Primeiro, é feita a revisão bibliográfica desta área, citando a arquitetura de permutação-difusão, a necessidade da dependência entre as sequências caóticas geradas e a imagem a ser cifrada, e os problemas de desempenho de muitos cifradores propostos recentemente na literatura. Após isso, apresenta-se o cifrador proposto, além de testes relativos à sua resistência aos mais variados ataques e ao seu desempenho.
- No quarto capítulo, trabalha-se com a versão de permutação-difusão simultânea do cifrador de imagens. Primeiro, é realizada a revisão bibliográfica justificando o porque da escolha desta arquitetura, e depois, o algoritmo é apresentado. Por fim, são realizados testes para mostrar a segurança e o alto desempenho do cifrador.

- No quinto capítulo, é realizado um desenvolvimento similar ao do terceiro capítulo, agora com foco em aplicações na telemedicina. Primeiro é feita a revisão bibliográfica, de trabalhos que juntam criptografia com telemedicina, e depois, uma versão do cifrador é proposta, com a junção de informações textuais na cifragem. Testes são realizados para mostrar a similaridade estrutural entre a imagem original e a recuperada, e também a segurança do processo de cifragem e sua eficiência.
- Finalmente, são apresentadas conclusões deste trabalho, junto com a apresentação de propostas para trabalhos futuros.

2 SISTEMAS DE DINÂMICA DISCRETA E TEORIA DO CAOS

Este capítulo compreende o estudo de sistemas com dinâmica discreta e a teoria do caos para sistemas discretos. Primeiro, conceitos relativos à sistemas dinâmicos de tempo discreto são apresentados, e posteriormente a teoria do caos é apresentada. Por fim, é realizado o estudo do mapa da tenda, que será usado para gerar as sequências caóticas neste trabalho.

2.1 Conceitos de sistemas de dinâmica discreta

Sistemas dinâmicos se dividem em sistemas de tempo contínuo e sistemas de tempo discreto.

- **Sistemas de tempo contínuo:** são descritos por uma equação diferencial no formato da equação (1), em que x é um vetor de dimensão de n que representa o espaço de estados, $\dot{x}$ são as derivadas de x, F são as funções que representam cada derivada de x, p o conjunto de parâmetros e t é o vetor de tempo, com a condição inicial sendo dada por $x(0) = x(t_0)$.

$$\dot{x} = F(x, t, p), t \in [t_0, \infty). \tag{1}$$

- **Sistemas de tempo discreto:** também denominado mapa, é um sistema dinâmico que evolui no tempo de forma discreta, ou seja, sua entrada são valores que pertencem ao conjunto dos números naturais, e é descrito pela equação (2) (QIAO, 2021).

$$x_{n+1} = F(x_n, n, p), n = 0, 1, 2..., \tag{2}$$

  em que n é o índice de tempo, começando em $t = 0$, x_n é a variável de estado e p é o conjunto de parâmetros para a função F.

A seguir, alguns conceitos importantes quanto a dinâmica de sistemas de tempo discreto são enunciados (QIAO, 2021).

- **Espaço topológico:** Um espaço topológico é um conjunto X com subconjuntos

T que apresentem as seguintes propriedades (TRIEBEL, 2010):

  - O conjunto vazio pertence à T.
  - O conjunto X pertence à T.
  - A intersecção de um número arbitrário finito de subconjuntos de T também pertence à T.
  - A união de um número arbitrário finito de subconjuntos de T também pertence à T.

  Em um mapa de definição $f : a \rightarrow b$, o espaço topológico é dado por a e b, que são os intervalos que os valores de entrada e saída do mapa assumem, respectivamente, (ELMANFALOTY; ABOU-BAKR, 2020).

- **Órbitas:** o conjunto que contém todos os estados do sistema $(x_0, x_1, ..., x_n)$ é denominado de órbitas (FIEDLER-FERRARA; PRADO, 1994). O estado x_n é dado pela aplicação da função F do estado anterior x_{n-1}, que é a aplicação da função F em x_{n-2}, de forma que $F(x_n)$ é dado por n composições a partir da condição inicial x_0, ou seja:

$$\begin{aligned} x_n &= F(x_{n-1}) = F^{(2)}(x_{n-2}) = ... = F^{(n)}(x_0) \\ &= F \circ ... \circ F(x_0) \end{aligned}$$

- **Ponto fixo:** diz-se que x^* é um ponto fixo de (2) se $x^* = F(x^*)$. Isso significa que no gráfico de x_n por x_{n+1}, o ponto fixo está na reta $x_{n+1} = x_n$.

Um ponto fixo está ligado a estabilidade do sistema, e para estudar a estabilidade de x^*, deve-se considerar o que ocorre com as sucessivas iterações de um mapa a partir de um ponto x_i próximo de x^* (FIEDLER-FERRARA; PRADO, 1994), e na próxima subseção, esta relação é abordada.

2.1.1 Análise de estabilidade de sistema de tempo discreto

O ponto fixo x^* pode ser estável, assintoticamente estável ou instável, dependendo da sequência de pontos $x_{i+1}, x_{i+2}, x_{i+3}, ..., x_n$. Caso essa sequência se aproxime de x^*, o ponto fixo será assintoticamente estável, e caso se afaste, será instável.

Teorema 2.1.1 (Estabilidade de um ponto fixo)**.** *Seja um ponto x_i próximo de x^*, tal que $x_i = x^* + \varepsilon_i$, sendo ε_i uma pequena distância, e $\dfrac{dF(x^*)}{dx} = C$. Sendo assim, x^* é assintoticamente estável se $|C| < 1$ e instável se $|C| > 1$ (FIEDLER-FERRARA; PRADO, 1994).*

Prova: A prova do Teorema 2.1.1 consiste em analisar a estabilidade de x^*, identificando o que ocorre com x^* na expressão (3),

$$x_{i+1} = F(x_i) = F(x^* + \varepsilon_i) = x^* + \varepsilon_{i+1}, \tag{3}$$

em que x_{i+1} se afasta ou se aproxima de x^*, fazendo com que ε_{i+1} seja maior ou menor que ε_i. Para tanto, faz-se a expansão da equação (3) em série de Taylor em torno de x^*, e tem-se:

$$x^* + \varepsilon_{i+1} = F(x^* + \varepsilon_i) = F(x^*) + \frac{dF(x^*)}{dx}\varepsilon_i + \frac{1}{2}\frac{d^2F(x^*)}{dx^2}\varepsilon_i^2 + ...$$

Levando em conta que $F(x^*) = x^*$ e desconsiderando os termos após a primeira ordem, obtém-se $\varepsilon_{i+1} = C\varepsilon_i$, onde $C = \dfrac{dF(x^*)}{dx}$. A partir deste cálculo, podem aparecer dois casos:

Caso 1: $|C| < 1$, $\varepsilon_{i+1} < \varepsilon_i$ e o ponto fixo é assintoticamente estável. Se $0 < C < 1$, a distância entre as sucessivas iterações($x_{i+1}, x_{i+2}, x_{i+3}, ...$) e x^* diminui assintoticamente. Caso $-1 < C < 0$, essa distância também diminui, mas de maneira oscilante, com $\varepsilon_i, \varepsilon_{i+1}, \varepsilon_{i+2}$ tendo sinais alternados.

Caso 2: Caso $|C| > 1$, a distância entre os valores em sucessivas iterações irão se afastando cada vez mais de x^*, e o ponto fixo é instável. □

Quando o ponto fixo é estável, o sistema não será caótico pois as iterações tenderão todas à x^*. Por outro lado, caso seja instável, é possível que o sistema seja caótico, o que é verificado a partir da análise de características do sistema, que serão vistas na próxima seção. Dessa forma, se diz que um ponto fixo instável pode fornecer uma rota para o caos.

2.2 Teoria do Caos

De acordo com DEVANEY (2018), um sistema dinâmico é considerado caótico se apresentar as características que seguem.

- Transitividade topológica.
- Órbitas periódicas densas.
- Sensibilidade às condições iniciais.

A seguir, estas características são definidas, de acordo com MONTEIRO (2002); ELMANFALOTY; ABOU-BAKR (2020).

- **Transitividade topológica:** Seja X um espaço topológico, um mapa contínuo $f : X \to X$ é topologicamente transitivo em X se para dois conjuntos não vazios

$U_1, U_2 \subset X$, existe um valor de $n > 0$ tal que $f^n(U_1) \cup U_2 \neq \emptyset$, ou seja, que a partir de qualquer entrada, possa-se obter uma saída não-nula, implicando que exista órbita que passe tão perto quanto se queira de qualquer ponto do domínio X.

- **Órbitas periódicas densas:** Um ponto $x \in X$ é um ponto periódico de f, com período dado por $n \in \mathbb{N} > 0$, se $f^{(n)}(x) = x$, o que implica a partir de qualquer entrada, a saída em uma iteração n futura tem um valor aproximado ao da entrada.

- **Sensibilidade às condições iniciais:** O mapa $f : X \to X$ é sensível às condições iniciais se para qualquer $x \in X$ e qualquer vizinhança N de x existe $y \in N$ e $n \geq 0$ tal que $|f^{(x)}(x) - f^{(x)}(y)| > \delta$, com $\delta > 0$, ou seja, com duas condições iniciais próximas, os resultados das iterações se distanciam a medida que o número de iterações tende ao infinito.

Em um sistema dinâmico, a transitividade e a densidade das órbitas periódicas são verificadas por meio de mapas de retorno, que plotam $F^{(n)}(x)$ em função de x, enquanto à sensibilidade às condições iniciais é verificada por duas medidas, uma qualitativa e uma quantitativa: o diagrama de bifurcações e os expoentes de Lyapunov. Nas próximas duas subseções, estes métodos são apresentados, e após, apresenta-se o mapa caótico usado neste trabalho, com foco na robustez do mapa, no seu consumo de tempo computacional e na degradação dinâmica.

2.3 Diagrama de bifurcações

Bifurcação é classificada como uma mudança no comportamento do sistema quando um parâmetro passa por um valor crítico (ALLIGOOD; SAUER; YORKE, 1996). Neste contexto, o diagrama de bifurcações computa a mudança qualitativa das soluções de um sistema em relação a um parâmetro deste sistema, de forma que sejam classificadas em periódicas e não periódicas, e no caso de ser periódica, verificar qual o período.

Para a construção do diagrama, primeiro é escolhido o parâmetro do sistema a ser variado, além do intervalo e o passo dado pela norma de partição do intervalo de variação do parâmetro. Para cada valor do parâmetro, o sistema é iterado até um critério de parada estipulado. Os resultados das primeiras iterações, denominadas transientes, são descartadas. A quantidade de valores transientes depende de cada sistema, e fica a critério de quem está realizando as simulações. Após as iterações iniciais do sistema, o resultado de cada iteração é marcado no diagrama, em função do valor do parâmetro.

Os valores transientes são descartados porque são necessárias algumas(normalmente) iterações para que um padrão no comportamento do sistema seja

estabelecido. Isto é mostrado na Figura 8, em que o sistema tende a um único valor, mas são necessárias algumas iterações até que a estabilidade do sistema seja observada. Em virtude disso, as iterações transientes são descartadas, e apenas as iterações posteriores as transientes são marcadas no diagrama.

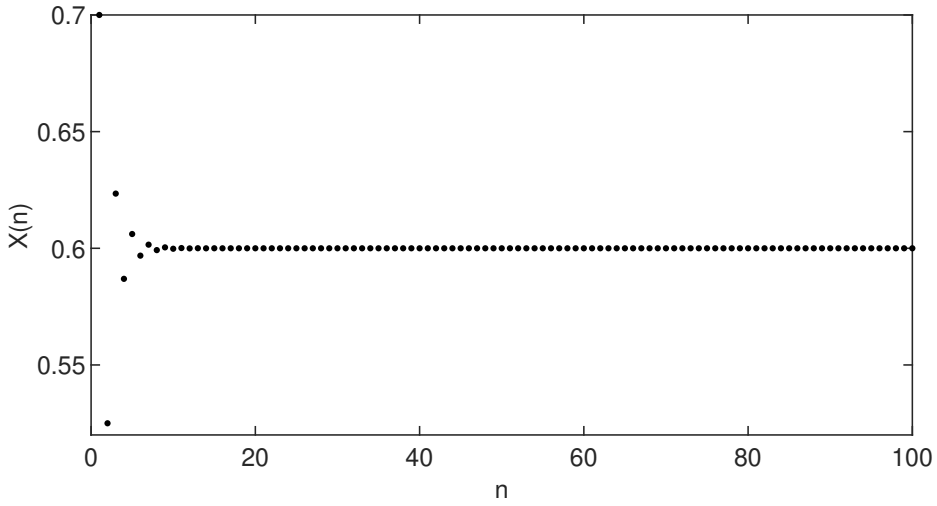


Figura 8: Comportamento de um sistema em que as iterações tendem a um único valor final.

O diagrama de bifurcações pode ser apresentado pelo mapa logístico, proposto por MAY (1976), e descrito pela equação (4):

$$x_{n+1} = r x_n (1 - x_n). \tag{4}$$

Este mapa é equivalente a equação logística de crescimento populacional, porém em tempo discreto. Neste sistema, $0 \leq x_n \leq 1$ é o valor adimensional da população na enésima geração e $0 < r \leq 4$ é a taxa de crescimento intrínseco, que é a diferença *per capita* entre as taxas de natalidade e mortalidade em uma determinada espécie (KOT, 2001). Este sistema representa um mapa unidimensional de simples compreensão, no entanto, em termos de comportamento, apresenta alta complexidade. Isso pode ser observado ao verificarmos o comportamento de x_n para diferentes valores de r, como mostrado pela Figura 9, na qual com $r = 2,5$, as iterações do sistema tendem a um único valor final, com $r = 3,1$, as iterações oscilam entre dois valores, com $r = 3,5$, oscilam entre quatro valores, e por fim, com $r = 3,99$, oscilam de forma aleatória, entre múltiplos valores.

Para observar o comportamento do sistema para múltiplos valores do parâmetro r, usa-se o diagrama de bifurcações, dado pela Figura 10, o qual mostra que os comportamentos do sistema são consistentes com os apresentados na Figura 9.

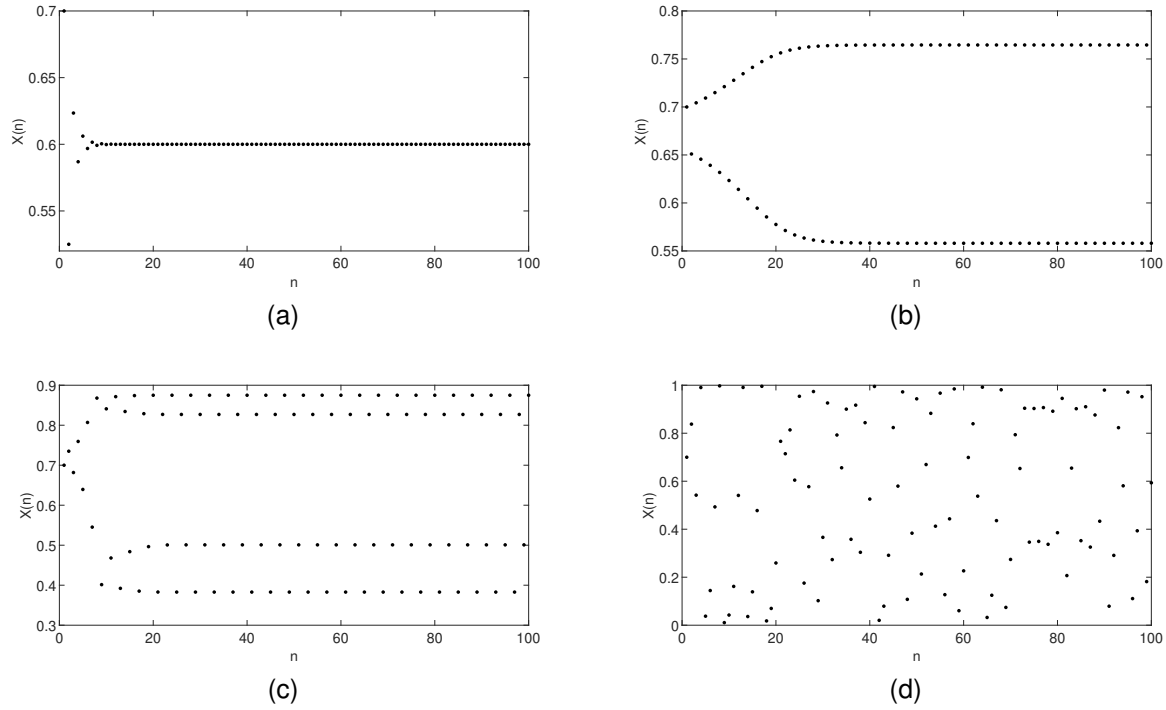


Figura 9: Comportamentos do mapa logístico: (a) $r = 2,5$, (b) $r = 3,1$, (c) $r = 3,5$, (d) $r = 3,99$.

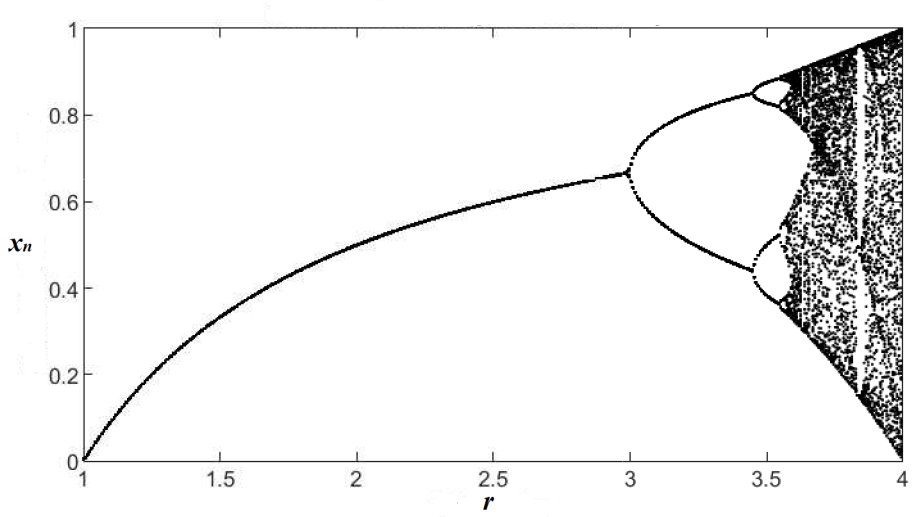


Figura 10: Diagrama de bifurcações para o mapa logístico.

2.4 Expoentes de Lyapunov

O expoente de Lyapunov é o valor que mede o afastamento exponencial de duas órbitas inicialmente próximas. No caso de ser positivo, denota o afastamento exponencial destas trajetórias, o que implica que há a sensibilidade às condições iniciais. Caso contrário, o sistema é periódico e/ou estável, não apresentando sensibilidade às condições iniciais. A seguir, é feita a dedução da fórmula de obtenção deste expoente

para sistemas de tempo discreto, baseado em FIEDLER-FERRARA; PRADO (1994).

Junto com o mapa dado pela equação (2), são consideradas duas condições iniciais, x_0 e y_0, e a distância entre elas, dada pela equação (5):

$$\delta_0 = y_0 - x_0, \tag{5}$$

Após uma iteração, a distância é dada pela equação (6), considerando $y_1 = F(y_0)$ e $x_1 = F(x_0)$.

$$\delta_1 = y_1 - x_1. \tag{6}$$

De forma que:

$$\delta_1 = e^L \delta_0, \tag{7}$$

A distância é calculada ao longo de n iterações, e o afastamento é ilustrado na Figura 11.

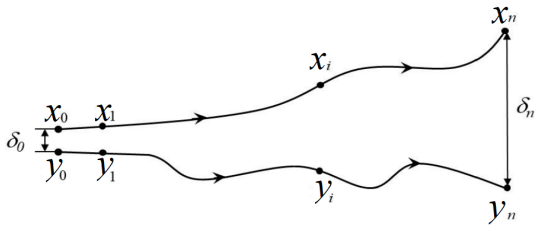


Figura 11: Representação da separação exponencial de duas trajetórias inicialmente separadas por δ_0. Adaptado de QIAO (2021).

Com L medindo a taxa exponencial de expansão da distância δ até a distância δ' como resultado de uma única iteração. A equação (6) pode ser escrita como

$$\begin{aligned} \delta' &= F(y_0) - F(x_0) \\ &= F(x_0 + \delta) - F(x_0), \end{aligned} \tag{8}$$

onde na última passagem utilizou-se a equação (6). Usando as equações (7) e (8), o resultado é $|F(x_0 + \delta) - F(x_0)| = |\delta| e^L$. Iterando o mapa N vezes, tem-se que:

$$|F^N(x_0 + \delta) - F^N(x_0)| = |\delta| e^{NL}, \tag{9}$$

com $F^N(x)$ representando o $N-$ésimo valor de x. Da equação (9), tem-se que:

$$L = \frac{1}{N} \ln \left| \frac{F^N(x_0 + \delta) - F^N(x_0)}{\delta} \right|.$$

Mas L depende de N, e então consideramos uma distância inicial infinitesimal $(\delta \to 0)$ seja considerada depois de um número infinito de iterações $(N \to \infty)$. Isso

leva a

$$\lambda(x_0) \equiv L(x_0) = \lim_{N\to\infty}\lim_{\delta\to 0}\frac{1}{N}\ln\left|\frac{F^N(x_0+\delta)-F^N(x_0)}{\delta}\right| = \lim_{N\to\infty}\frac{1}{N}\ln\left|\frac{dF^N(x_0)}{dx_0}\right|, \tag{10}$$

que é, por definição, o expoente característico de Lyapunov do mapa e constitui uma medida de divergência exponencial das órbitas $(\lambda > 0)$ ou de convergência exponencial $(\lambda < 0)$. Observa-se que o expoente de Lyapunov depende do ponto inicial, e a equação (10) pode ser expandida da seguinte forma:

$$\frac{d}{dx_0}F^N(x_0) = \frac{d}{dx_0}F(x_{N-1})\frac{d}{dx_0}F(x_{N-2})\cdots\frac{d}{dx_0}F(x_0), \tag{11}$$

onde $x_i = F^i(x_0)$. Substituindo a equação (11) na equação (10), obtém se que o expoente de Lyapunov para um mapa como o dado pela equação (2) é dado por:

$$\lambda = \lim_{N\to\infty}\frac{1}{N}\ln\left|\prod_{i=0}^{\infty}\frac{d}{dx_0}F(x_i)\right| = \lim_{N\to\infty}\frac{1}{N}\sum_{i=0}^{N-1}\ln|F'(x_i)|$$

No caso do mapa logístico, dado pela equação (4), tem-se que:

$$F(x_n) = rx_n(1-x_n),$$
$$F'(x_n) = r(1-2x_n).$$

Levando em conta os possíveis valores de r para este sistema, o expoente de Lyapunov é mostrado na Figura 12, apresentando resultados bastante consistentes com o diagrama de bifurcações, na Figura 10. Para os valores de r nos quais o sistema apresenta poucas soluções, ou seja, $r \in [1, 3, 5]$ o expoente de Lyapunov é negativo e o sistema não é caótico. Por outro lado, para os valores de r nos intervalos $[3, 5, 3, 8]$ e $[3, 9, 4]$, que o sistema apresenta um número maior de soluções, o expoente de Lyapunov é positivo e o sistema é caótico.

O conceito de caos está relacionado à sensibilidade às condições iniciais de um sistema. Na Figura 13, considerou-se dois conjuntos de iterações da equação (4), x e y, inicialmente próximas, com $x_0 = 0,7$ e $y_0 = x_0 + 10^{-15}$, ou seja, $\delta_0 = 10^{-15}$. Na Figura 13 (a) , com $r = 2,5$ o sistema não é caótico e dessa forma os resultados dessas iterações seguem próximos. Já na Figura 13 (b), com $r = 3,99$, o sistema é caótico, e assim, os valores das iterações divergem.

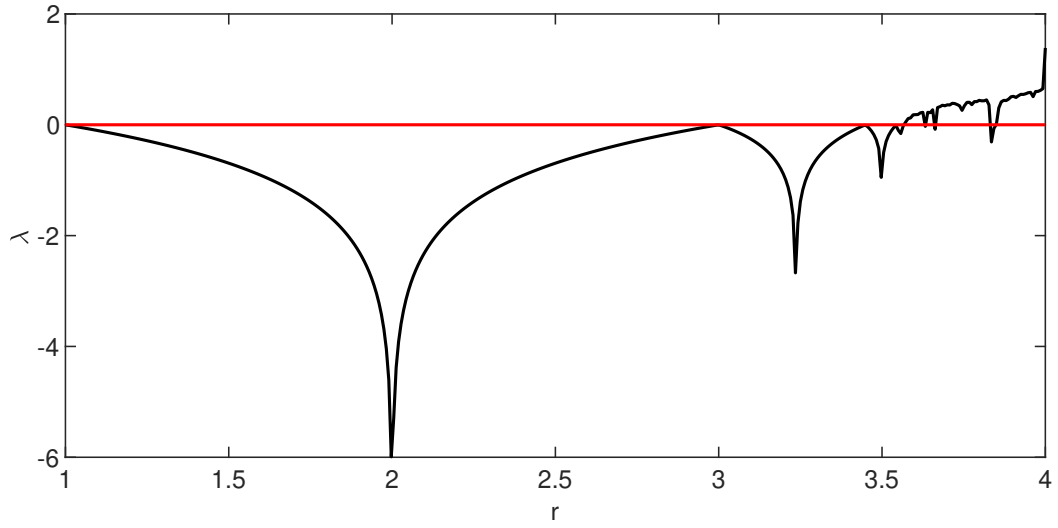
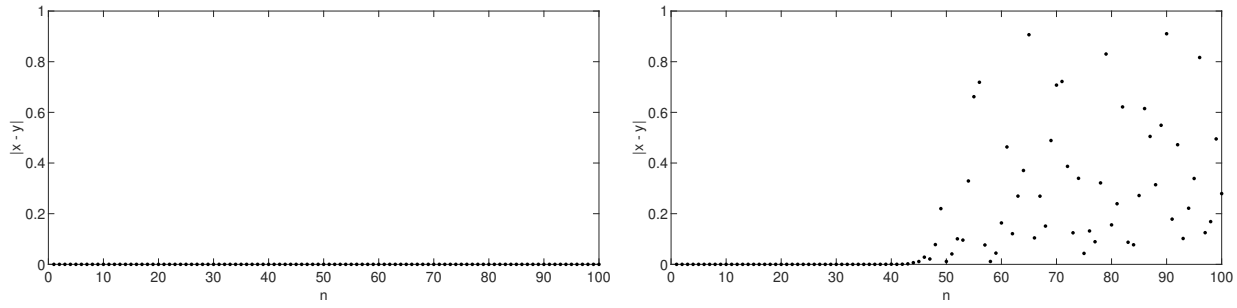


Figura 12: Expoente de Lyapunov para o mapa logístico.



(a) (b)

Figura 13: Diferença entre as iterações de x e de y (a) $r = 2,5$ e (b) $r = 3,99$.

2.5 Mapa da tenda

Neste trabalho, o foco se dá em torno de uma família que mapeiam um intervalo unitário para si próprio, com dois parâmetros, $p \in (0,1)$ e $b \in [0,1]$, denominado mapa da tenda e dado pela equação (12). Como o mapa da tenda é definido em duas partes, em cada uma com uma função de primeiro grau, também é chamado de *Piecewise Linear Chaotic Map*(PWLCM), no espaço $F : [0,1] \to [0,1]$.

$$F(x_n) = x_{n+1} = \begin{cases} b + \frac{1-b}{p} x_n, & x_n \in [0,p), \\ \frac{1-x_n}{1-p} & x_n \in (p,1). \end{cases} \tag{12}$$

Neste trabalho, considera-se um caso especial da equação (12), com $b = 0$, que é o mapa da tenda assimétrico (ALVAREZ; LI, 2006), dado pela equação (13).

$$F(x) = x_{n+1} = \begin{cases} \frac{x_n}{p}, & x_n \in (0,p), \\ \frac{1-x_n}{1-p} & x_n \in (p,1). \end{cases} \tag{13}$$

Embora a definição do mapa seja simples, ele dá origem a uma gama considerável de comportamentos (BASSEIN, 1998). O primeiro passo no estudo desse mapa é a análise da estabilidade de seus pontos fixos, de acordo com o Teorema 2.1.1, e considerando a primeira definição da função, para $x_n \in [0, p)$, tem-se que:

$$F(x^*) = x^* \Rightarrow \frac{x^*}{p} = x^* \Rightarrow x^* - \frac{x^*}{p} = 0 \Rightarrow x^* \left(1 - \frac{1}{p}\right) = 0 \Rightarrow x^* = 0.$$

Como o interesse é no ponto fixo não trivial, leva-se em conta a segunda parte da equação (13), então:

$$\begin{aligned} x^* &= \frac{1 - x^*}{1 - p}, \\ x^*(1 - p) &= 1 - x^*, \\ x^* - x^*p &= 1 - x^*, \\ 2x^* - x^*p &= 1, \\ x^*(2 - p) &= 1, \\ x^* &= \frac{1}{2 - p}. \end{aligned}$$

Considerando que o ponto fixo $x^* = \frac{1}{2-p}$ é válido no intervalo $[p, 1)$, tem-se que:

$$\begin{aligned} F'(x^*) &= -\frac{1}{(1 - p)^2}, \\ C &= -\frac{1}{(1 - p)^2}. \end{aligned}$$

Como $0 < p < 1$, tem-se a garantia que $|C| > 1$, e dessa forma, o ponto fixo $x^* = \frac{1}{2-p}$ é instável, o que implica que o resultado das iterações serão repelidos do ponto fixo, podendo fornecer um caminho para o caos.

Sabendo-se que o mapa da equação (13) apresenta ponto fixo instável, então busca-se verificar se apresenta comportamento caótico. Na Figura 14, tem-se os valores de $f^{(1)}(x), f^{(2)}(x), f^{(3)}(x)$ e $f^{(4)}(x)$ plotados em função de x, mostrando a densidade de soluções periódicas e a transitividade (GUO, 2014).

- **Densidade de soluções periódicas:** A função F^n mapeia cada intervalo $\left[\frac{k}{2^n}, \frac{k+1}{2^n}\right]$ para $[0, 1]$ para $k = 0, 1, ..., 2^{n-1}$. Então, como é mostrado na Figura (14), F^n intersecciona a reta $y = x$ uma vez para cada intervalo, garantindo que em cada intervalo exista um valor de n tal que $F^n(x) = x$, o que implica que soluções periódicas são densas em $[0, 1]$.

- **Transitividade:** Considerando U_1 e U_2 dois intervalos em $[0, 1]$, assim, de acordo com a Figura 14, para n suficientemente grande e para algum valor de k, U_1

contém um intervalo de forma $\left[\frac{k}{2^n}, \frac{k+1}{2^n}\right]$, assim mapeando U_1 para um intervalo de $[0, 1]$ que contém U_2, garantindo a transitividade.

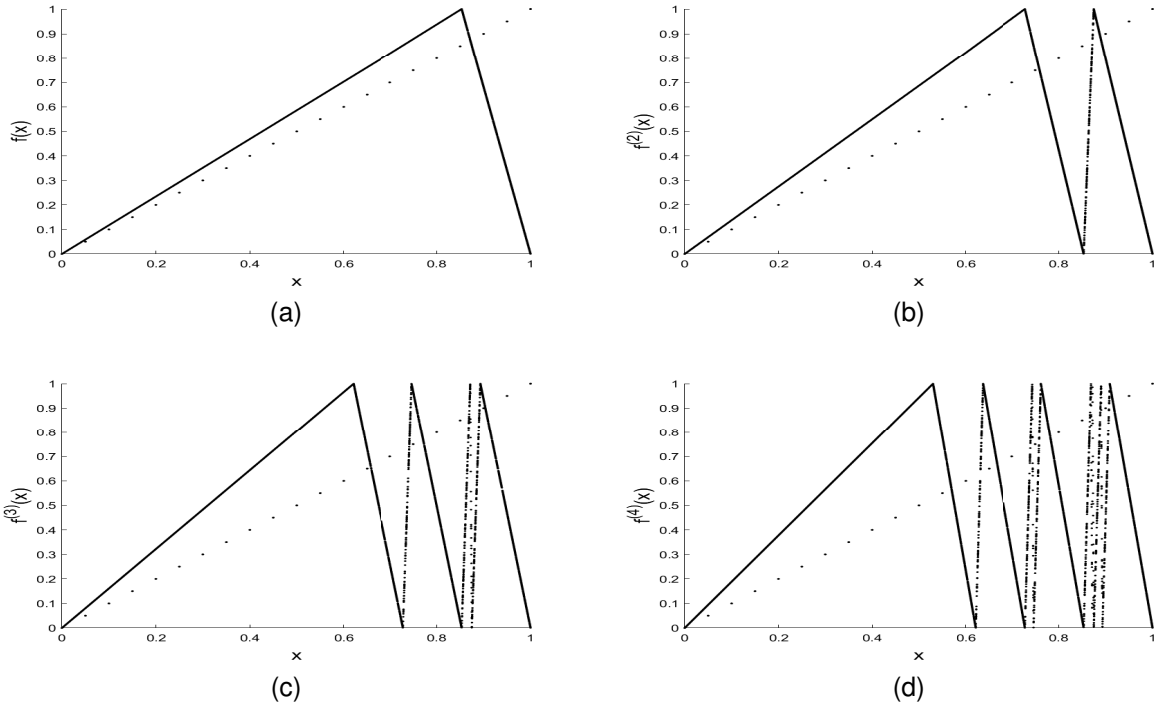


Figura 14: Mapas de retorno. (a) $f^{(1)}(x)$. (b) $f^{(2)}(x)$. (c) $f^{(3)}(x)$. (d) $f^{(4)}(x)$.

A sensibilidade às condições iniciais também é verificada visualmente, a partir do diagrama de bifurcações e do expoente de Lyapunov. No item Figura 15 (a), tem-se que o diagrama é bastante denso para todos os valores do parâmetro p, dando a indicação de que o mapa não apresenta janelas periódicas, o que é confirmado pelos expoentes de Lyapunov positivos para todos os valores de p, na Figura 15 (b).

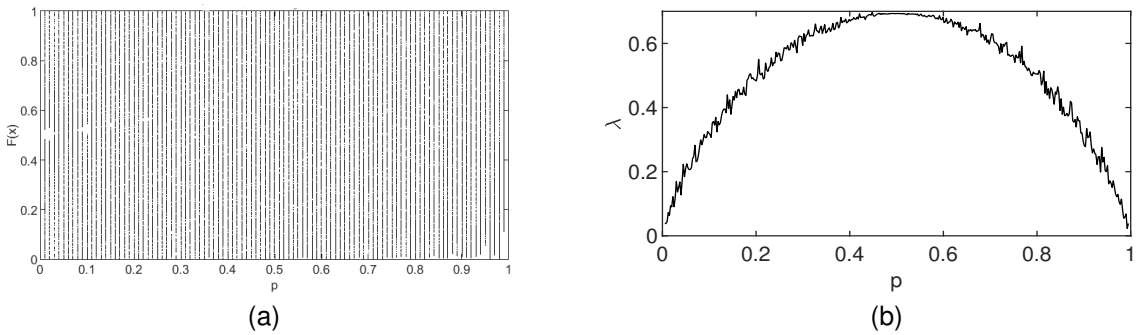


Figura 15: (a) Diagrama de bifurcações para a equação (13), e (b) Expoentes de Lyapunov.

Na criptografia, os símbolos dos arquivos, sejam texto, imagem, ou áudio, são tratados como valores inteiros. Dessa forma, a variável de estado x_n é traduzida para

linguagem de máquina, representada por um número inteiro de N *bits*, de acordo com a equação (14), chamada de função de quantização e proposta por GUARDEÑO (2009), que tem comportamento caótico para $\alpha = 33$.

$$h(x_n) = [2^{\alpha} \cdot x_n] \mod 2^N. \tag{14}$$

Por exemplo, considerando $N = 8$ e $x_n = 0,5$, tem-se que:

$$\begin{aligned} h(0,5) &= \left[2^{33} \cdot 0,5\right] \mod 256 \\ &= 4.294.967.296 \mod 256 = 0 \end{aligned}$$

Observa-se que a operação $\mod 256$ é o resto da divisão inteira por 256, tomando valores inteiros entre 0 e 255. Com valores escolhidos aleatoriamente, apenas para ilustrar, seguem alguns exemplos:

$$\begin{aligned} 1 \mod 256 &= 1, \\ 2 \mod 256 &= 2, \\ 100 \mod 256 &= 100, \\ 257 \mod 256 &= 1, \\ 300 \mod 256 &= 44, \\ 511 \mod 256 &= 255, \\ 512 \mod 256 &= 0, \end{aligned}$$

A Figura 16 mostra que mesmo após os resultados do mapa caótico serem convertidos em valores inteiros, as sequências geradas apresentação sensibilidade às condiçõe iniciais, em virtude dos expoentes de Lyapunov positivos.

2.5.1 Robustez

A Figura 16 destacou um conceito importante, que junta a teoria do caos com a criptografia, que é a robustez. Um mapa caótico é classificado como robusto se apresenta comportamento caótico para todos os valores do parâmetro do sistema. Mapas tradicionais, como o mapa logístico, dado pela equação (4), não são robustos, como mostra a Figura 12, e assim apresentam um espaço de chaves menor, ou seja, há menos opções possíveis para os valores do parâmetro do sistema, o que além de ser custoso computacionalmente por haver a necessidade de controle para manter os valores do parâmetro em um intervalo caótico faz com que cifradores baseados nestes mapas tenham uma menor resistência à ataques de força bruta (ALVAREZ; LI, 2006; MURILLO-ESCOBAR et al., 2015). Neste trabalho, empregam-se as sequências geradas pelas equações (13) e (14), que são robustas, gerando assim um cifrador que

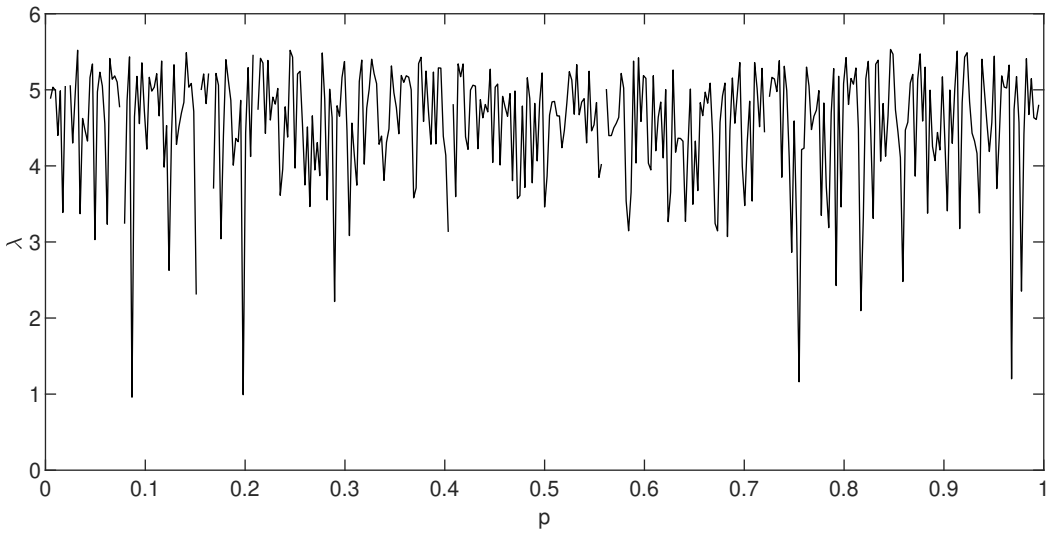


Figura 16: Expoente de Lyapunov, de uma sequência gerada pela equação (13), com a aplicação da função de quantização dada pela equação (14).

será mais resistente à ataques de força bruta.

Além da robustez, tanto o mapa caótico, dado pela equação (13), como a função de quantização dada pela equação (14), foram escolhidas em função de dois outros fatores, que serão abordados nas próximas subseções.

- densidade uniforme invariante;
- eficiência computacional.

Ademais, deve-se considerar que quando um modelo matemático é implementado em uma máquina com aritmética de precisão finita, a degradação digital é um problema, que será abordado no fim deste capítulo.

2.5.2 Densidade uniforme invariante

Se dividirmos todos os valores possíveis da variável de estado $x_n \in (0, 1)$ em um conjunto de M subintervalos iguais e calcularmos o número de vezes que a trajetória visita cada sub-intervalo i, $(1 \leq i \leq M)$, como m_i, a probabilidade de cada intervalo ser visitado é dada pela equação (15) (PATIDAR et al., 2010), considerando N o número de iterações.

$$pr_i = \frac{m_i}{N}. \tag{15}$$

A densidade de probabilidade ser uniforme invariante implica que todos os intervalos tem a mesma probabilidade de ser visitado. Assim, do ponto da vista de segurança, impede-se que informações da sequência caótica sejam descobertas. Por exemplo, para um mapa que não apresentar distribuição uniforme, a cada iteração, certos resultados serão mais prováveis que outros, ou ainda, certas condições iniciais ou valores do parâmetro podem gerar densidades diferentes, assim fornecendo a um atacante

informações sobre a sequência caótica ou ainda sobre a chave do cifrador (LI, 2003), com um exemplo disso sendo o trabalho de PAREEK; PATIDAR; SUD (2003), quebrado por ALVAREZ et al. (2003) em virtude de não apresentar densidade uniforme invariante, assim fornecendo ao atacante informações sobre o valor de parâmetro usado para gerar as sequências caóticas.

Em BILLINGS; BOLLT (2001), é provado que mapas da família (12), como por exemplo (13), apresentam densidade uniforme invariante. Visualmente, a densidade uniforme invariante é representada por histogramas uniformes, como na Figura 17, em que o número de vezes que cada região é visitada, em 1.000,000 de iterações, tanto da equação (13) como após a aplicação da função de discretização (14), é similar para cada região.

Para obter o gráfico (a) da Figura 17, considerou-se que o mapa representado pela equação (13) apresenta valor de x_n entre 0 e 1. Então, dividindo entre 256 regiões iguais, com passo de $\frac{1}{256}$, a primeira região será composta por valores de x_n entre 0 e $\frac{1}{256}$, a segunda região por valores entre $\frac{1}{256}$ e $\frac{2}{256}$, até a última região que foi composta por valores entre $\frac{255}{256}$ e 1. Por exemplo, considerando-se $p = 0,5$ e $x_0 = 0,8147$.

- $x_1 = 0,8576$, e assim, a região $\frac{0,8576}{\frac{1}{256}} = 219$ foi visitada pela primeira vez, logo o contador $m(i)$ em $i = 219$ é incrementado em 1.

- $x_2 = 0,9027$, e assim, a região $\frac{0,9027}{\frac{1}{256}} = 231$ foi visitada pela primeira vez, logo o contador $m(i)$ em $i = 231$ é incrementado em 1.

- O padrão mostrado pelos dois casos acima segue-se até encontrar $x_{1.000,000} = 0,5755$, em que a região $\frac{0,5755}{\frac{1}{256}} = 147$ foi visitada pela vez número 3826, assim o contador $m(i)$ na posição $i = 147$ assumiu o valor de 3826.

Para obter o gráfico (b) da Figura 17, ao invés de calcular a região visitada na enésima iteração aplicando a divisão de x_n por $\frac{1}{256}$, aplicou-se a função de discretização dada pela equação (14).

2.5.3 Eficiência computacional

Neste trabalho, usa-se um mapa caótico unidimensional, pois em um mapa caótico de tempo discreto e de uma dimensão, as iterações ocorrem mais rapidamente do que em sistemas de dimensão superior, ou de tempo contínuo, em virtude de não haver a utilização de métodos numéricos para obter as soluções (ZHU; SUN, 2018). Cifradores caóticos, considerando imagens de dimensão $M \times N \times C$, em que M é a altura, N a largura e C o número de canais de cores, normalmente apresentam complexidade de ordem $O(M \times N \times C)$, pois o número esperado de iterações do mapa caótico é ao menos $M \times N \times C$, ou ainda $2 \times M \times N \times C$, no caso da arquitetura permutação-difusão. Para o cifrador ser eficiente, há a necessidade de minimizar o

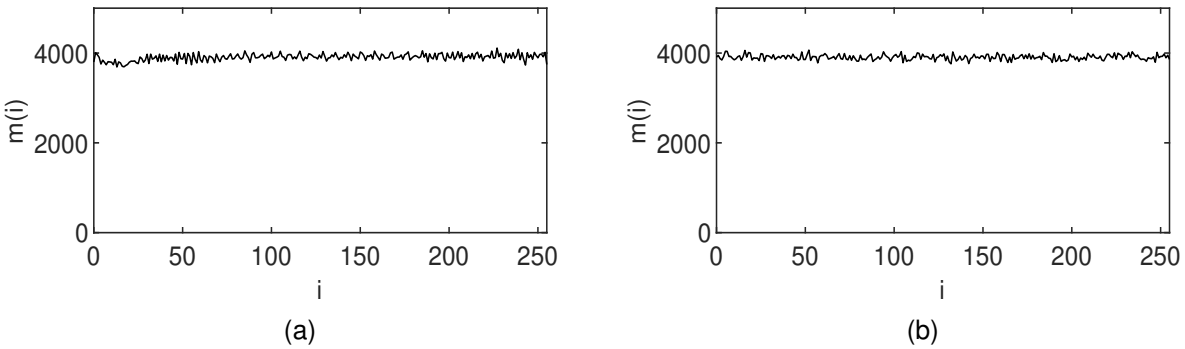


Figura 17: (a) Número de vezes que cada região da equação (13) é visitada. (b) Número de vezes que cada região é visitada, após a aplicação da função de discretização dada por (14).

consumo de tempo, com a escolha do sistema caótico. Na Tabela 2, mostra-se que iterar um mapa caótico unidimensional é consideravelmente mais rápido que obter solução numérica do mesmo número de iterações do sistema de Lorenz, de dimensão $3D$, usado no trabalho de KHAN et al. (2017), dado pela equação (16), que neste trabalho foi simulado com $\sigma = 10, \beta = \frac{8}{3}, r = 48$.

$$\begin{aligned} \dot{x} &= \sigma(y - x), \\ \dot{y} &= rx - y - xz, \\ \dot{z} &= xy - \beta z. \end{aligned} \tag{16}$$

No trabalho de CHEN et al. (2019), defende-se o uso do sistema hiper-caótico de Lorenz, de dimensão $4D$, ou seja, com 4 variáveis, dado pela equação (17) e simulado considerando $a = 10, b = \frac{8}{3}, c = 28, r = -1$, com o argumento de que cada uma das variáveis pode servir como uma sequência caótica, e que resolver numericamente um número de iterações do sistema hiper-caótico de Lorenz é mais rápido que quatro conjuntos de iterações de um mapa unidimensional. Contudo, na Tabela 3 mostra-se que o tempo computacional necessário para resolver estes conjuntos de iterações do mapa dado pela equação (13) é consideravelmente menor que o tempo usado para resolver numericamente o sistema hiper-caótico de Lorenz por intermédio do método de Runge-Kutta de quarta ordem.

$$\begin{aligned} \dot{x} &= a(y - x) + w, \\ \dot{y} &= cx - y - xz, \\ \dot{z} &= xy - bz, \\ \dot{w} &= -yz + rw. \end{aligned} \tag{17}$$

Muitos trabalhos, como por exemplo os de HUA et al. (2015, 2018); WU; LIAO; YANG (2018); CAO; SUN; LIU (2018); CHEN et al. (2019) também defendem o uso dos mapas caóticos de dimensão superior, além do uso de sistemas de tempo contínuo (KHAN et al., 2017; KAUR; KUMAR, 2018; CHENG; WANG; XU, 2020; XIAN; WANG, 2021; ZENG; WANG, 2021), argumentando que as dinâmicas mais complexas destes sistemas levam a um comportamento caótico mais forte, e assim os cifradores ficam mais seguros. Ao longo deste trabalho, uma abordagem diferente será usada, utilizando mapas caóticos unidimensionais, em virtude de serem mais rápido para resolver as iterações, sem causar prejuízos a segurança do cifrador. Isto ocorre pois a segurança do mapa caótico é medida pelo expoente de Lyapunov, não relacionado à dimensão do sistema (GUARDEÑO, 2009). Por exemplo, as sequências caóticas geradas da combinação das equações (13) e (14) têm expoente de Lyapunov maior que o sistema de Lorenz, de dimensão $3D$ (BARRIO; SERRANO, 2007). Além disso, a arquitetura do cifrador também possui um papel importante quanto à segurança (ALVAREZ; LI, 2006).

Nº de iterações	Equações (13) e (14)	Lorenz (KHAN et al., 2017)
256×256	0,0058 s	0,88 s
$2 \times 256 \times 256$	0,0089 s	1,40 s
512×512	0,01 s	2,67 s
$2 \times 512 \times 512$	0,02 s	5,61 s

Tabela 2: Comparação do tempo para iterar os mapas (13) e (14) com o tempo necessário para resolver o sistema de Lorenz.

Nº de Iterações	4 conjuntos de iterações de (13)	Lorenz Hiper-caótico
256×256	0,018 s	0,82 s
$2 \times 256 \times 256$	0,03 s	1,55 s
512×512	0,04 s	2,88 s
$2 \times 512 \times 512$	0,07 s	6,24 s

Tabela 3: Comparação do tempo necessário para resolver 4 conjuntos de iterações das equações (13) e (14) com o tempo necessário para resolver o sistema hipercaótico de Lorenz.

2.5.4 Degradação dinâmica

Em um esquema de cifragem, os mapas caóticos são implementados utilizando arquiteturas de precisão finita, o que implica que o comportamento caótico mostrado em uma arquitetura de precisão infinita não será mais confiável, o que prejudica a segurança do cifrador (LI et al., 2018a). Isto, porque as iterações caóticas tendem à um mesmo valor, ou a uma sequência padrão, assim sendo possível que partes não

autorizadas ataquem o cifrador. Esta questão é chamada de degradação dinâmica, e neste trabalho, isto é tratado perturbando o parâmetro p após cada 100 iterações da equação (13), com o novo valor dependendo do valor do sistema x_i, usando a equação (18):

$$p = \text{mod}(p + x_i, 1). \tag{18}$$

Um exemplo de utilização da equação (18) é dado a seguir, considerando $x_0 = 0,65, p = 0,4$.

- Na iteração 100, tem-se que $x_{100} = 0,641494$. Assim, $x_{100} + p = 0,641494 + 0,4 = 1,041494$, e $\mod(1,041494, 1) = 0,041494$. Portanto, as próximas 100 iterações da equação (13) são resolvidas com $p = 0,041494$.
- Na iteração 200, tem-se que $x_{200} = 0,917389$. Assim, $x_{200} + p = 0,917389 + 0,041494 = 0,958883$, e $\mod(0,958883, 1) = 0,958883$. Portanto, as próximas 100 iterações da equação (13) são resolvidas com $p = 0,958883$. E a cada 100 iterações, o mesmo processo vai sempre se repetindo.

Para provar que a sequência gerada pelas equações (13) e (14), utilizando a perturbação dada pela equação (18), apresenta sensibilidade às condições iniciais, que somada a densidade das órbitas periódicas e a transitividade faz com que a sequência seja caótica, aplicam-se os testes do NIST em uma sequência de *bits* geradas por essas equações, considerando a solução em uma arquitetura de precisão finita de 64 *bits*. Estes testes foram implementados em linguagem C, com o ambiente fornecido por (COMPUTER SECURITY DIVISION, 2010). Para cada um destes testes, o *p-value*, que é a medida estatística que mede a probabilidade de se encontrar a de *bits* obtida, está acima do nível de significância de $0,01$, como mostra a Tabela 4, e dessa forma, as sequências caóticas usadas neste trabalho serão resistentes à degradação dinâmica, podendo ser usadas sem apresentar riscos à segurança do cifrador caótico.

Teste	*p-value*	Resultado
Monobit	0,259355	Passou
Block Frequency	0,736456	Passou
Runs	0,280899	Passou
Longest Runs	0,818819	Passou
Rank	0,403751	Passou
Discrete Fourier Transform	0,816329	Passou
Non-overlapping Template Matching	0,955318	Passou
Overlapping Template Matching	0,435723	Passou
Universal	0,480172	Passou
Linear Complexity	0,678820	Passou
Serial 1	0,269286	Passou
Serial 2	0,213062	Passou
Approximate Entropy	0,134445	Passou
Cumulative Sums Forward	0,142899	Passou
Cumulative Sums Backward	0,082721	Passou
Random Excursions	0,967304	Passou
Random Excursions Variant	0,965824	Passou

Tabela 4: Testes do NIST para as sequências caóticas gerada pelas equações (13),(14) e (18).

3 ALGORITMO PARA CIFRAGEM DE IMAGENS

Neste capítulo, o algoritmo para cifragem de imagens é descrito. Primeiramente, é realizada a revisão bibliográfica acerca de cifradores caóticos de imagens. Após, descreve-se o algoritmo proposto neste trabalho, e testes são realizados, os quais mostram que o cifrador é resistente às mais variadas formas de ataque, como estatísticos, diferencial, ruído e de informação escolhida, enquanto possui desempenho superior a outros trabalhos recentemente publicados na literatura.

3.1 Revisão bibliográfica

Cifradores de imagens normalmente possuem a arquitetura de permutação-difusão. Durante a permutação, pixels ou *bits* da imagem original são embaralhados para gerar uma imagem não identificável, enquanto na difusão, uma função é aplicada para substituir os valores dos *pixels*. Em um algoritmo baseado na teoria do caos, as sequências usadas nas operações de permutação e difusão são geradas a partir de mapas caóticos. Esta arquitetura foi usada pela primeira vez no trabalho de FRIDRICH (1998), e vários outros trabalhos seguiram esta mesma linha (GAO et al., 2006; PAREEK; PATIDAR; SUD, 2006; SUN et al., 2008; BEHNIA et al., 2008; PATIDAR et al., 2010; USAMA et al., 2010; WANG et al., 2010). Contudo, estes cifradores acabaram sendo mostrados inseguros (SOLAK et al., 2010; RHOUMA; SOLAK; BELGHITH, 2010; ZHANG et al., 2012; XIE et al., 2017; LI; ZHANG; XIE, 2019), por problemas como escolha de chaves que geravam sequências não-caóticas, alta correlação entre os pixels cifrados, e acima de tudo, a não dependência das sequências caóticas geradas em relação às imagens, ou seja, a sequência caótica usada era sempre a mesma, independente da imagem a ser cifrada.

Neste cenário, uma inovação considerável para a aplicação da teoria do caos na cifragem de imagens foi o trabalho de ZHANG; WANG (2014b), em que informações relacionadas às imagens foram utilizadas para alterar levemente as chaves dos mapas caóticos, procedimento este que se tornou bastante utilizado na literatura (MURILLO-ESCOBAR et al., 2015; AHMAD et al., 2018; KHAN et al., 2017; WU; LIAO; YANG,

2018; YE et al., 2018; FARAH; FARAH; FARAH, 2019; CHENG; WANG; XU, 2020; WANG; GAO, 2020a; DENG et al., 2021; ZENG; WANG, 2021). A estratégia mais utilizada para isso são funções *hash*, em que uma sequência de *bits* é gerada a partir da imagem original, e uma pequena mudança na imagem gera uma mudança significativa na sequência de *bits*.

Do ponto de vista de segurança, a relação entre as sequências caóticas e a imagem a ser cifrada foi um avanço significativo, diminuindo consideravelmente o número de ataques sofridos por cifradores caóticos. Contudo, do ponto de vista de desempenho, a maioria dos cifradores caóticos propostos ainda podem ser melhorados, em virtude das razões que seguem.

- Custo computacional elevado, em virtude de má escolha de sistemas caóticos, como em KHAN et al. (2017); ÇAVUŞOĞLU et al. (2017); KAUR; KUMAR (2018); CHEN et al. (2019); CHENG; WANG; XU (2020); ZENG; WANG (2021); XIAN; WANG (2021), em que sistemas caóticos de tempo contínuo são utilizados. No capítulo 2, nas Tabelas 2 e 3, mostra-se que iterar um sistema unidimensional de tempo discreto é consideravelmente mais rápido que resolver numericamente um sistema de tempo contínuo.

- Permutação realizada sobre *bits* ao invés de pixels, tal como nos trabalhos de LIU; WANG (2011); ZHANG et al. (2013); ZHANG; WANG (2015); ZHOU et al. (2015); ZHANG et al. (2016); CHAI (2017); WANG; LI; JIN (2020). Como um pixel é composto por oito *bits*, a permutação realizada sobre *bits* exige um número de operações oito vezes maior (HUANG et al., 2018; ZHU; SUN, 2018).

- Necessidade de múltiplas rodadas de permutação-difusão (MOLLAEEFAR; SHARIF; NAZARI, 2017; WU et al., 2017; YE; HUANG, 2017; ZHANG; TANG, 2018; WANG; GAO, 2020b).

- Uso de técnicas de DNA (LIU; WANG, 2012; CHAI et al., 2019), que possuem alta complexidade computacional.

Levando em conta o estado da arte, neste trabalho um novo cifrador é proposto, utilizando arquitetura de permutação-difusão que é ao mesmo tempo rápida e segura. A velocidade se dá em virtude do mapa caótico utilizado e da arquitetura do algoritmo, com o mapa sendo uma simples função de dimensão $1D$, gerada a partir das equações (13) e (14), fazendo com que a geração das sequências caóticas seja praticamente instantânea, como é mostrado nas Tabelas 2 e 3. Além disso, o uso de permutação de pixels, e apenas uma rodada de permutação-difusão também aumentam a eficiência computacional do cifrador. Quanto à segurança, o comportamento caótico do mapa existe independente do valor do parâmetro, ou seja, o mapa é robusto, e com a aplicação de uma função *hash* SHA-256, que gera uma sequência de

256 *bits* diferente para cada imagem, tornando as sequências caóticas dependentes da imagem original. Testes e comparações com outros trabalhos mostraram que os dois objetivos, de segurança e de desempenho, são atingidos.

3.2 Descrição do cifrador

Nesta seção, é feita a descrição da arquitetura do algoritmo que será usado para a cifragem de imagens. O cifrador apresenta as seguintes chaves:

- x_{R_0} e p_r, relacionadas ao mapa caótico x_R, com valores no intervalo $(0, 0, 95)$;
- x_{C_0} e p_c, relacionadas ao mapa caótico x_C, com valores no intervalo $(0, 0, 95)$;
- $x_{D_{a_0}}$ e p_{D_a}, relacionadas ao mapa caótico x_{D_a}, com valores no intervalo $(0, 0, 95)$;
- $x_{D_{b_0}}$ e p_{D_b}, relacionadas ao mapa caótico x_{D_b}, com valores no intervalo $(0, 0, 95)$;
- C_{in}, usada para a operação *xor*, de 8 *bits*;
- *Hash* de 256 *bits*.

O cifrador precisa ser seguro contra os ataques típicos, e essa resistência é dependente da arquitetura do cifrador. O ataque típico mais poderoso é o do texto original escolhido, ou seja, *chosen-plaintext*, e se um cifrador é seguro contra este ataque, também será contra outros ataques típicos (CHAI et al., 2019). A resistência de um cifrador contra este ataque está relacionada à dependência das sequências caóticas em relação à imagem original. Neste trabalho, usa-se uma *hash* SHA-256 para perturbar as chaves originais, assim garantindo a resistência do cifrador ao ataque do texto original escolhido, de forma que ataques, tais como os apresentados nos trabalhos de RHOUMA; SOLAK; BELGHITH (2010); LIU et al. (2016); WEN (2016); NOROUZI; MIRZAKUCHAKI (2017); LAIPHRAKPAM; KHUMANTHEM (2017); ZHU; SUN (2018); WANG et al. (2018); HUANG et al. (2020) não funcionem neste trabalho. Em outros trabalhos, como MURILLO-ESCOBAR et al. (2015); LI et al. (2018b), usou-se a soma dos valores dos pixels, ou ainda entropia, para modificar as chaves, porém essas estratégias se provaram inseguras (DOU et al., 2017; FAN et al., 2018; LIU; ZHANG; CHEN, 2019), indicando que o uso de um algoritmo de *hashing* é o método mais eficiente para garantir a dependência das sequências caóticas em relação a imagem original.

3.2.1 Cifragem

A cifragem se dá de acordo com os seguintes passos, resumidos no fluxograma da Figura 18.

- **Passo 1:** a imagem original I é carregada, de dimensão $M \times N$.
- **Passo 2:** aplica-se o algoritmo SHA-256 na imagem original, retornando a sequência de 256 *bits* $S = S(1)S(2)S(3)\cdots S(256)$. Esta sequência é transferida ao receptor durante o protocolo de troca de chaves, junto com as chaves para os mapas caóticos, usando um algoritmo de chave pública como o RSA, assim garantindo a segurança da troca de chaves.
- **Passo 3:** ocorre a perturbação nas chaves do cifrador. Primeiramente, as perturbações λ são geradas de acordo com a sequência de *bits* resultante da *hash* no passo anterior, da seguinte forma:

$$\begin{aligned}
&\lambda_1 = S(1:32), && \lambda_2 = S(33:64),\\
&\lambda_3 = S(65:96), && \lambda_4 = S(97:128),\\
&\lambda_5 = S(129:160), && \lambda_6 = S(161:192),\\
&\lambda_7 = S(193:224), && \lambda_8 = S(225:256),\\
&\lambda_9 = S(1)S(33)S(65)S(97)S(129)S(161)S(191)S(225).
\end{aligned}$$

  Então, esses valores, com exceção de λ_9 são convertidos para decimal, e multiplicados por 10^{-11}, para que assumam valores no intervalo $(0,0,05)$. Então, tem-se que:

$$\lambda_i = \text{conversãoDecimal}(\lambda_i), \qquad \lambda_i = 10^{-11} \times \lambda_i.$$

  Por fim, se dá a perturbação nas chaves, da seguinte forma:

$$\begin{aligned}
x_{R_0} &= x_{R_0} + \lambda_1, & p_r &= p_r + \lambda_2,\\
x_{C_0} &= x_{C_0} + \lambda_3, & p_C &= p_C + \lambda_4,\\
x_{D_{a_0}} &= x_{D_{a_0}} + \lambda_5, & p_{D_{a_0}} &= p_{D_{a_0}} + \lambda_6,\\
x_{D_{b_0}} &= x_{D_{b_0}} + \lambda_7, & p_{D_{b_0}} &= p_{D_{b_0}} + \lambda_8,\\
C_{in} &= C_{in} \oplus \lambda_9.
\end{aligned}$$

- **Passo 4:** resolvem-se 200 iterações transientes de x_R, x_C, x_{D_a} e x_{D_b}, que são descartadas.
- **Passo 5:** as sequências caóticas são geradas. Inicialmente, $M \times N$ iterações de

$x_R, x_C, x_{D_a}, x_{D_b}$ são resolvidas, criando as seguintes sequências:

$$\begin{aligned} x_R &= (x_R(0), x_R(1), \cdots, x_R(MN-1)), \\ x_C &= (x_C(0), x_C(1), \cdots, x_C(MN-1)), \\ x_{D_a} &= (x_{D_a}(0), x_{D_a}(1), \cdots, x_{D_a}(MN-1)), \\ x_{D_b} &= (x_{D_b}(0), x_{D_b}(1), \cdots, x_{D_b}(MN-1)). \end{aligned} \tag{19}$$

A função de quantização, dada pela equação (14), é aplicada a cada elemento das sequências da equação (19), gerando as seguintes sequências:

$$\begin{aligned} h_R &= (h_R(0), h_R(1), \cdots, h_R((M-1)(N-1)), \\ h_C &= (h_C(0), h_C(1), \cdots, h_C((M-1)(N-1))), \\ h_{D_a} &= (h_{D_a}(0), h_{D_a}(1), \cdots, h_{D_a}((M-1)(N-1))), \\ h_{D_b} &= (h_{D_b}(0), h_{D_b}(1), \cdots, h_{D_b}((M-1)(N-1))). \end{aligned} \tag{20}$$

- **Passo 6:** processo de permutação, utilizando as sequências h_R e h_C da equação (20), de acordo com as equação (21), para $i \in [0, M-1], j \in [0, N-1]$.

$$I(i,j) \Leftrightarrow I(h_R(i \times N + j), h_C(i \times N + j)). \tag{21}$$

- **Passo 7:** processo de difusão, usando as sequências h_{D_a} e h_{D_b} da equação (20), de acordo com a equação (22), com $C(-1) = C_{in}$. Após estes passos, tem-se que a imagem cifrada é dada por C.

$$C(i,j) = I(i,j) \oplus h_{D_a}(i \times N + j) \oplus h_{D_b}(i \times N + j) \oplus C((i \times N + j) - 1). \tag{22}$$

3.2.2 Decifragem

O processo de decifragem ocorre em ordem inversa à cifragem, e os passos são os que seguem.

- **Passo 1:** leitura da imagem cifrada, que é dada por C.
- **Passo 2:** perturbação das chaves, similar ao passo 3 da cifragem, com a *hash* recebida durante o protocolo de troca de chaves.
- **Passo 3:** resolver as iterações transientes dos mapas caóticos, assim como no passo 4 da cifragem.
- **Passo 4:** gerar as sequências caóticas, assim como no passo 5 da cifragem.

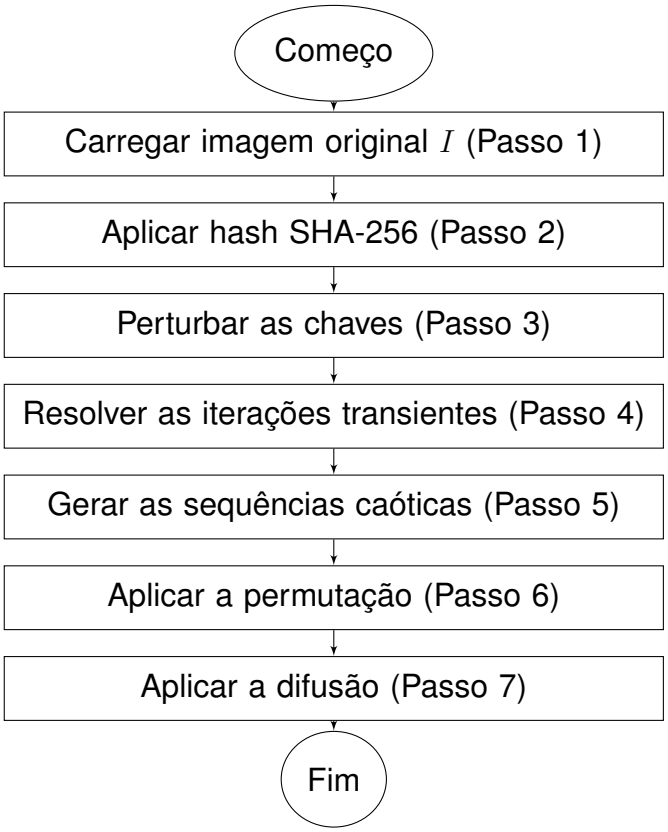


Figura 18: Fluxograma para o processo de cifragem do algoritmo proposto.

- **Passo 5:** difusão reversa, em que a imagem permutada é recuperada, de acordo com a equação (23), em que $C_p(i) = C(i-1), C_p(0) = C_{in}$.

$$I = C \oplus h_{D_a} \oplus h_{D_b} \oplus C_p. \tag{23}$$

- **Passo 6**: permutação reversa, aplicada a imagem permutada recuperada I, encontrada no passo anterior. Na decifragem, trabalha-se com os índices i e j em ordem descendente, de M e N, respectivamente, até 0, A imagem recuperada é dada pela equação (24), enquanto a Figura 20 mostra que esta arquitetura é reversível.

$$I(i,j) \Leftrightarrow I(h_R(i \times N + j), h_C(i \times N + j)). \tag{24}$$

Um exemplo de cifragem e decifragem é dado a seguir, considerando uma imagem

lida no passo 1 I de dimensões 256×256, e utilizando as seguintes chaves.

$$\begin{aligned} x_{R_0} &= 0,456781, & p_r &= 0,100713, \\ x_{C_0} &= 0,172020, & p_c &= 0,234568, \\ x_{D_{a0}} &= 0,646504, & p_{Da} &= 0,423221, \\ x_{D_{b0}} &= 0,246557, & p_{Db} &= 0,499542, \\ C_{in} &= 27. \end{aligned}$$

- Supõe-se que a sequência binária gerada pela *hash*, gerada no passo 2 e utilizada no passo 3 é dada por:

$$\begin{aligned} S(1:32) &= 11101001101110101001001100010100, \\ S(33:64) &= 11000010011111110111010000110101, \\ S(65:96) &= 10010000100110101010110000001100, \\ S(97:128) &= 11111011111011111110101100100011, \\ S(129:160) &= 11100100111101100101000101111100, \\ S(161:192) &= 00100011001000010010110110010100, \\ S(193:224) &= 01101111010111110110111000111110, \\ S(225:256) &= 11110000100101000111010010101111. \end{aligned}$$

- Com isso, também tem-se que: $\lambda_9 = 11111001$.

- Então, é realizada a conversão de binário para decimal de cada um valores, $\lambda_1 = S(1:32), \lambda_2 = S(33:64)$, até $\lambda_9 = 11111001$. Após, o resultado das conversões são multiplicados por 10^{-11}, e o resultado de cada perturbação é dado a seguir.

$$\begin{aligned} \lambda_1 &= 0,006843, & \lambda_2 &= 0,028888, \\ \lambda_3 &= 0,008088, & \lambda_4 &= 0,033025, \\ \lambda_5 &= 0,010493, & \lambda_6 &= 0,006997, \\ \lambda_7 &= 0,020882, & \lambda_8 &= 0,041134, \\ \lambda_9 &= 249. \end{aligned}$$

- Com isso, as chaves, com as perturbações, serão as seguintes, para as 100

primeiras iterações:

$$
\begin{aligned}
x_{R_0} &= x_{R_0} + \lambda_1 = 0,456781 + 0,006843 = 0,463624,\\
p_r &= p_r + \lambda_2 = 0,100713 + 0,028888 = 0,129601,\\
x_{C_0} &= x_{C_0} + \lambda_3 = 0,172020 + 0,008088 = 0,180108,\\
p_c &= p_c + \lambda_4 = 0,234568 + 0,033025 = 0,267593,\\
x_{D_{a_0}} &= x_{D_{a_0}} + \lambda_5 = 0,646504 + 0,010493 = 0,659997,\\
p_{D_a} &= p_{D_a} + \lambda_6 = 0,423221 + 0,006997 = 0,430218,\\
x_{D_{b_0}} &= x_{D_{b_0}} + \lambda_7 = 0,246557 + 0,020882 = 0,267439,\\
p_{D_b} &= p_{D_b} + \lambda_8 = 0,499542 + 0,041134 = 0,540676,\\
C_{in} &= C_{in} \oplus \lambda_9 = 27 \oplus 249 = 226.
\end{aligned}
$$

De 100 em 100 iterações, os valores de p_r, p_c, p_{D_a} e p_{D_b} são modificados de acordo com a equação (18).

- No passo 4, aplicam-se as 200 iterações transientes de cada sequência caótica.
- No passo 5, são calculadas as $M \times N$ iterações de cada uma das 4 sequências caóticas, gerando as sequências:

$$
\begin{aligned}
x_R &= (0,820995, 0,205658, \cdots, 0,341039),\\
x_C &= (0,628688, 0,506975, \cdots, 0,132666),\\
x_{D_a} &= (0,838923, 0,282700, \cdots, 0,978832),\\
x_{D_b} &= (0,881023, 0,259026, \cdots, 0,739062).
\end{aligned}
$$

Sobre estas sequências, a função de quantização (14) é aplicada. Para a sequência x_R, para gerar h_R a operação de resto $\mod$ é aplicada em relação ao número de linhas M, neste caso 256. Para x_C, para gerar h_C a operação é aplicada em relação ao número de colunas N, que neste caso, também é 256. Quanto as sequências x_{D_a} e x_{D_b}, que irão gerar h_{D_a} e h_{D_b}, a operação sempre será aplicada em relação à 256, pois os pixels sempre irão assumir valores entre 0 e 255. Com isso, as sequências quantizadas são dadas por:

$$
\begin{aligned}
h_R &= (74, 227 \cdots, 56),\\
h_C &= (12, 228, \cdots, 97),\\
h_{D_a} &= (71, 170, \cdots, 115),\\
h_{D_b} &= (48, 27, \cdots, 104).
\end{aligned}
$$

- Depois da quantização das sequências, vem a permutação. De acordo com a equação (21), e as sequências encontradas no passo anterior, tem-se que:

$$I(0,0) \leftrightarrow I(74,12), I(0,1) \leftrightarrow I(227,228) \cdots I(255,255) \leftrightarrow I(56,97)$$

  Ou seja, o pixel na posição $(0,0)$ é trocado com o pixel na posição $(74,12)$, depois o pixel na posição $(0,1)$ é trocado com o pixel na posição $(227,228)$, com as trocadas sendo realizadas de acordo com as sequências h_R e h_c até o pixel na posição $(255,255)$ ser trocado com o pixel na posição $(56,97)$.

- Por fim, após a permutação, ocorre a difusão. Na difusão, considera-se que uma matriz, no caso a imagem, também pode ser representada como um vetor. Em uma imagem de dimensões $M \times N$, ou seja, N colunas, os primeiros N elementos do vetor são dados pela primeira linha da imagem, os elementos de N até $2N$ são a segunda linha, e assim por diante. Dessa forma, tem-se que:

$$I(i,j) = I(i \times N + j)$$

  Considerando que após a permutação, $I(0,0) = 5, I(0,1) = 6, \cdots, I(255,255) = 7, C((255 \times 255 + 255) - 1) = 210, C(-1) = 226$, tem-se que:

$$\begin{aligned}
C(0,0) &= I(0,0) \oplus h_{D_a}(0) \oplus h_{D_b}(0) \oplus C(-1) = 5 \oplus 71 \oplus 48 \oplus 226 = 144 \\
C(0,1) &= I(0,1) \oplus h_{D_a}(1) \oplus h_{D_b}(1) \oplus C(0,0) = 6 \oplus 170 \oplus 27 \oplus 144 = 39 \\
&\vdots \\
C(255,255) &= I(255,255) \oplus h_{D_a}(255 \times 256 + 255) \oplus h_{D_b}(255 \times 256 + 255) \oplus C(255,254) \\
&= 7 \oplus 115 \oplus 104 \oplus 210 = 206
\end{aligned}$$

- Depois destes passos, a imagem cifrada é dada por C. Na decifragem, a partir da imagem cifrada C, o objetivo é recuperar a imagem original I. As chaves e a *hash* utilizada para a perturbação são recebidas durante o protocolo de troca de chaves. Depois, a resolução das iterações transientes, e a geração das sequências caóticas se dá da mesma forma que na cifragem. Após, se dá a difusão

reversa, da seguinte forma:

$$
\begin{aligned}
I(0,0) &= C(0,0) \oplus h_{D_a}(0) \oplus h_{D_b}(0) \oplus C(-1) = 144 \oplus 71 \oplus 48 \oplus 226 = 5 \\
I(0,1) &= C(0,1) \oplus h_{D_a}(1) \oplus h_{D_b}(1) \oplus C(0,0) = 39 \oplus 170 \oplus 27 \oplus 144 = 6 \\
&\vdots \\
I(255,255) &= C(255,255) \oplus h_{D_a}(255 \times 256 + 255) \oplus h_{D_b}(255 \times 256 + 255) \oplus C(255,254) \\
&= 206 \oplus 115 \oplus 104 \oplus 210 = 7
\end{aligned}
$$

- Por fim, ocorre a permutação, em ordem reversa, ou seja: Primeiro, o pixel na posição $(255, 255)$ é trocado com o pixel na posição $(56, 97)$, com todo o processo se realizando, até que no fim o pixel na posição $(0, 1)$ é trocado com o pixel na posição $(227, 228)$ e o pixel na posição $(0, 0)$ é trocado com o pixel na posição $(74, 12)$. Após a permutação, a imagem original I é recuperada.

3.3 Análise de resultados

Nesta seção, é realizada a análise dos resultados do cifrador proposto, do ponto de vista de segurança contra:

- Ataques estatísticos, a partir da análise de histograma, testes do NIST, entropia de Shannon e coeficientes de correlação;
- Ataques diferenciais, pelos coeficientes de mudança nos pixels e efeito avalanche;
- Ataques tradicionais, como o ataque de texto original conhecido e escolhido;
- Ruídos, aos quais uma imagem pode sofrer quando é enviada em uma rede pública como a internet.

As imagens utilizadas para as análises são dadas pela Figura 19.

(a) (b) (c) (d) (e)

Figura 19: (a) Baboon. (b) Lena. (c) Iris. (d) Preta. (e) Branca.

3.3.1 Análise de histograma

O histograma de uma imagem descreve como os pixels estão distribuídos nesta imagem. A segurança de um cifrador contra ataques estatísticos depende da uniformidade do histograma. Neste trabalho, na Figura 20, é mostrado o histograma das imagens cifradas Black e Lena, mostrando que o histograma da imagem cifrada é uniforme.

Também é possível medir a uniformidade do histograma de forma quantitativa, usando o teste de Pearson (PATRO; ACHARYA, 2019; ZHANG et al., 2020). A medida quantitativa do teste é dada pela equação (25), para uma imagem de dimensões 256×256, considerando que $z(i)$ é o valor do histograma para o pixel i da imagem, ou seja, o número de pixel da imagem que apresentam como valor i. O parâmetro segue aproximadamente uma distribuição χ^2, com 255 graus de liberdade. Para testar a uniformidade do histograma, a um nível de significância de $0,05$, o valor crítico é $\chi^2_C = 293,248$, então se $\chi^2 \leq 293,248$, considera-se o histograma uniforme.

$$\chi^2 = \sum_{k=0}^{255} \frac{(z(i) - 256)^2}{256}. \tag{25}$$

Outra medida quantitativa para a uniformidade do histograma é sua variância (ZHANG; WANG, 2014), dada pela equação (26). Neste caso, z_i e z_j são os valores do histograma de duas imagens cifradas, da mesma imagem original, com chaves levemente modificadas.

$$\text{var}(Z) = \frac{1}{n^2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \frac{1}{2}(z_i - z_j)^2. \tag{26}$$

Na Tabela 5, calculam-se o coeficiente de Pearson e a variância para as imagens usadas neste trabalho. Todas as imagens cifradas apresentam $\chi^2 \leq 293,248$, enquanto a variância é em torno de 270, valor consideravelmente menor que no trabalho de ZHANG; WANG (2014), indicando que a mudança média para cada valor do histograma é em torno de 16,5. Dessa forma, com os resultados destes dois testes, o cifrador é resistente a ataques estatísticos baseados no histograma.

Coeficiente	Baboon	Iris	Lena	Black	White
χ^2 original	48.274	42.411	38.685	16.711.680	16.711.680
χ^2 cifrada	285	272	263	230	258
Variância	288	260	264	234	263

Tabela 5: Testes numéricos para medir a uniformidade do histograma das imagens cifradas.

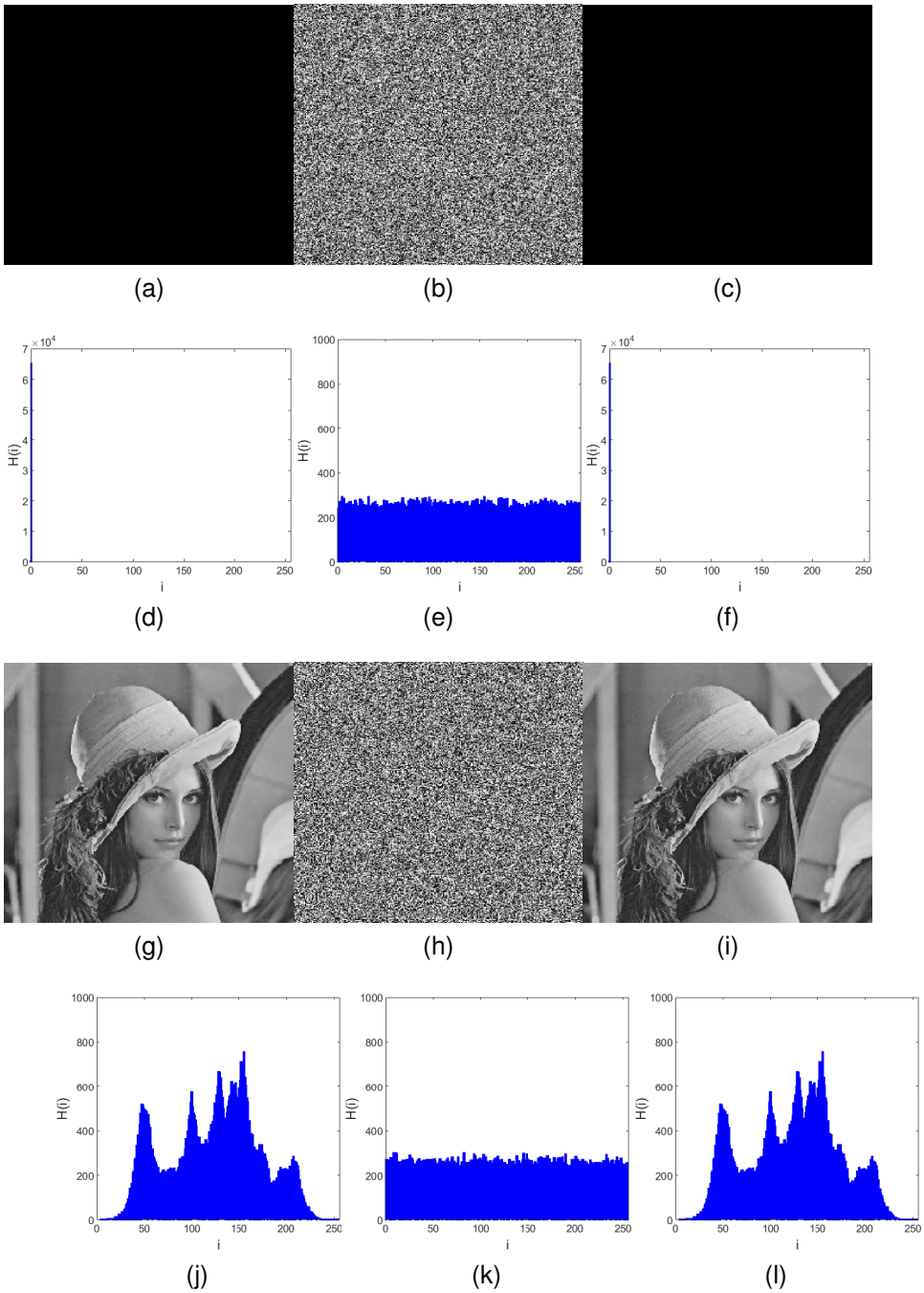


Figura 20: (a) Black. (b) Black cifrada. (c) Black recuperada. (d) Histograma da imagem black original. (e) Histograma da imagem black cifrada. (f) Histograma da imagem black recuperada. (g) Lena. (h) Lena cifrada. (i) Lena recuperada. (j) Histograma da imagem Lena original. (k) Histograma da imagem Lena cifrada. (l) Histograma da imagem Lena recuperada.

3.3.2 Testes do NIST

Outra métrica para a segurança das imagens cifradas é dada pelos testes do NIST, em que cada pixel da imagem cifrada é convertido em uma sequência de 8 *bits*, e os testes são aplicados sobre esta sequência. Na Tabela 6, tem-se que todos os valores

de *p-value* calculados estão acima de 0,01, o que significa que as imagens passam os testes de aleatoriedade, e que o algoritmo criptográfico é seguro.

NIST test	Baboon	Lena	Iris	Black	White
Monobit	0,364216	0,921888	0,721604	0,845604	0,497706
Block Frequency	0,455933	0,957896	0,895687	0,319655	0,585579
Runs	0,343015	0,612263	0,709166	0,73303	0,69105
Longest Runs	0,282445	0,487567	0,251354	0,579378	0,505901
Rank	0,246148	0,99966	0,312726	0,999999	0,999941
Discrete Fourier Transform	0,223251	0,733165	0,955844	0,823493	0,251924
Non-overlapping Template Matching	0,992696	0,980969	0,949014	0,857015	0,934283
Overlapping Template Matching	0,905256	0,885590	0,317224	0,669721	0,228337
Universal	0,140181	0,195249	0,756837	0,989693	0,965422
Linear Complexity	0,765865	0,197672	0,236203	0,921871	0,300006
Serial 1	0,719141	0,843704	0,473674	0,089817	0,362828
Serial 2	0,771570	0,589935	0,285193	0,383431	0,405505
Approximate Entropy	0,554277	0,075517	0,785908	0,404118	0,176535
Cumulative Sums Forward	0,582070	0,924485	0,974102	0,880344	0,503874
Cumulative Sums Backward	0,137653	0,974102	0,991813	0,765258	0,642635
Random Excursions	0,989089	0,780361	0,681695	0,930439	0,978539
Random Excursions Variant	0,949858	0,777297	0,9806	0,890834	0,9875

Tabela 6: Testes do NIST para as imagens cifradas.

3.3.3 Entropia de Shannon

A entropia é usada para medir a aleatoriedade dos pixels em uma imagem (PATRO; ACHARYA, 2019). Para uma imagem em que os pixels assumem valores de N *bits*, a entropia é calculada pela equação (27), proposta em SHANNON (1949), em que p_i representa a frequência do pixel i na imagem cifrada, e o valor ideal para o cifrador ser seguro é $S \approx N$, o que é obtido neste trabalho, como mostra a Tabela 7. Em implementações do AES, como no trabalho de ÇAVUŞOĞLU et al. (2018), os valores de entropia são de aproximadamente 7,96, um pouco abaixo do valor ideal de 8, mostrando assim que o cifrador proposto é mais resistente a ataques de entropia e lida melhor com a alta correlação entre pixels adjacentes das imagens.

$$S = -\sum_{k=0}^{2^N-1} p_i \log_2(p_i). \tag{27}$$

3.3.4 Espaço de chaves

O cifrador apresenta as seguintes chaves:

- 4 condições iniciais para a equação (13);

Imagem	Baboon	Iris	Lena	Black	White
256×256	7,9971	7,9973	7,9972	7,9976	7,9974
512×512	7,9992	7,9993	7,9992	7,9993	7,9994

Tabela 7: Entropia de Shannon para a imagem cifrada.

- 4 valores do parâmetro de controle para iterar a equação (13);
- um número inteiro de 8 bits;
- uma *hash* de 256 bits.

O padrão da IEEE (Institute of Electrical and Electronics Engineers) para a representação de números em ponto flutuante é de 10^{-15} (STEVENSON, 1985), ou seja, a precisão decimal é de 15 dígitos, o que implica que cada uma das chaves para a equação (13) apresenta espaço de chaves de 10^{15}. O valor inteiro de 8 *bits* possui espaço de chaves de 2^8, pois pode assumir valores entre 00000000 e 11111111 enquanto o espaço de chaves da *hash* é de 2^{256}. Dessa forma, a equação (28) fornece o espaço de chaves para o cifrador proposto.

$$\begin{aligned} K_s &= (10^{15})^8 \times 2^8 \times 2^{256}, \\ &= 10^{120} \times 2^{264}, \\ &= 2^{\log_2(10^{120})} \times 2^{264}, \\ &= 2^{399} \times 2^{264}, \\ &= 2^{663}. \end{aligned} \tag{28}$$

Na Tabela 8, é apresentada a comparação do espaço de chaves do cifrador proposto neste trabalho com outros trabalhos da literatura, e é mostrado que o espaço de chaves do nosso cifrador é maior. Isto é uma consequência do uso de quatro sequências caóticas neste trabalho, que é mais do que o número de sequências usadas nos trabalhos da Tabela 8.

Trabalho	Espaço de Chaves
Proposto	2^{663}
BELAZI; EL-LATIF; BELGHITH (2016)	2^{624}
ZHANG et al. (2020)	2^{512}
LUO et al. (2018)	2^{400}
WANG; LIU; ZHANG (2015)	2^{319}
FARAH; FARAH; FARAH (2019)	2^{186}

Tabela 8: Comparações do espaço de chaves do cifrador proposto com outros trabalhos da literatura

No trabalho de ALVAREZ; LI (2006), tem-se que um cifrador é resistente a um ataque de força bruta se possuir um espaço de chaves superior a 2^{100}, e considerando o desenvolvimento da computação quântica em algumas décadas, o espaço de chaves deve ser superior a 2^{256} (PAAR; PELZL, 2010). A Tabela (8) mostra que o espaço de chaves do cifrador proposto neste trabalho é consideravelmente superior à estes valores, por uma ordem de 2^{553}, indicando que o cifrador é seguro contra ataques de força bruta.

3.3.5 Coeficientes de correlação

Os coeficientes de correlação medem a correlação entre pixels adjacentes de uma imagem. É um recurso simples para notar as diferenças entre a imagem original, em que pixels próximos estão altamente correlacionados, com a imagem cifrada, em que os pixels próximos devem ser bastante diferentes. A medida quantitativa é dada pela equação (29), para uma imagem de dimensões $M \times N$, considerando que x e y são dois pixels adjacentes. Valores próximos a 1 indicam que existe alta correlação entre os pixels, enquanto valores próximos a 0 indicam que não há correlação e os pixels são randômicos. Na teoria, um cifrador seguro possui coeficientes de correlação bastante próximos de 0,

$$
\begin{aligned}
C(x,y) &= \frac{\mathsf{Cov}(x,y)}{\sqrt{\mathsf{Var}(x)}\sqrt{\mathsf{Var}(y)}} \qquad (29)\\
\mathsf{Cov}(x,y) &= \frac{1}{M \times N} \sum_{i=1}^{M \times N} (x_i - E(x))(y_i - E(y)),\\
E(x) &= \frac{1}{M \times N} \sum_{i=1}^{M \times N} x_i,\\
Var(x) &= \frac{1}{M \times N} \sum_{i=1}^{M \times N} (x_i - E(x))^2.
\end{aligned}
$$

Para o cifrador proposto, os valores dos coeficientes de correlação são dados pela Tabela 9, com as imagens cifradas apresentando valores próximos de 0, indicando que não há correlação entre os pixels adjacentes, ao contrário das imagens originais, que apresentam coeficientes de correlação próximos de 1, ou ainda infinitos, definidos como *Not-a-Number (NaN)*, para as imagens que apresentam todos os pixels com o mesmo valor, em que no cálculo do coeficiente há uma divisão por zero. O entendimento destes coeficientes também pode ser obtido visualmente, na Figura 21, construída com 3000 pixels selecionados aleatoriamente da imagem cifrada e da imagem original. Ademais, a média dos coeficientes de correlação deste trabalho é comparada com outros trabalhos similares da literatura, conforme mostrado na Tabela 10. Estes trabalhos são considerados semelhantes em virtude de usarem sistemas

caóticos de tempo discreto e arquitetura permutação-difusão. A Tabela mostra que o algoritmo proposto neste trabalho possui coeficientes de correlação abaixo destes trabalhos, sendo assim mais robusto contra ataques estatísticos.

Imagem	Direção	Imagem original	Imagem cifrada
Baboon	Horizontal	0,9445	-0,0018
	Vertical	0,9293	0,0025
	Diagonal	0,9040	-0,0015
Iris	Horizontal	0,9832	0,0023
	Vertical	0,9843	-0,00005
	Diagonal	0,9779	-0,0080
Lena	Horizontal	0,9052	0,0068
	Vertical	0,9452	0,0031
	Diagonal	0,8844	-0,0037
Black	Horizontal	NaN	-0,0015
	Vertical	NaN	-0,0015
	Diagonal	NaN	0,0002
White	Horizontal	NaN	0,0030
	Vertical	NaN	-0,0007
	Diagonal	NaN	0,0019
Average	Horizontal	-	0,0018
	Vertical	-	0,0007
	Diagonal	-	-0,0022

Tabela 9: Comparação de coeficientes de correlação para pixels adjacentes.

Trabalho	Horizontal	Vertical	Diagonal
Proposto	0,0018	0,0007	-0,0022
FARAH; FARAH; FARAH (2019)	0,0029	-0,0032	-0,0023
BELAZI; EL-LATIF; BELGHITH (2016)	-0,0113	-0,0147	0,0099
WANG; TENG; QIN (2012)	-0,0086	-0,0090	-0,0067

Tabela 10: Média dos Coeficientes de Correlação.

3.3.6 Ataque diferencial

A resistência de um cifrador à ataque diferencial está relacionada com sua sensibilidade a pequenas mudanças na imagem original ou na chave do cifrador. Esta sensibilidade é medida utilizando duas métricas, que são:

- Number of Pixels Rate Change (NPCR), que é a porcentagem de pixels da imagem cifrada que muda quando ocorre uma pequena mudança na imagem original ou na chave.

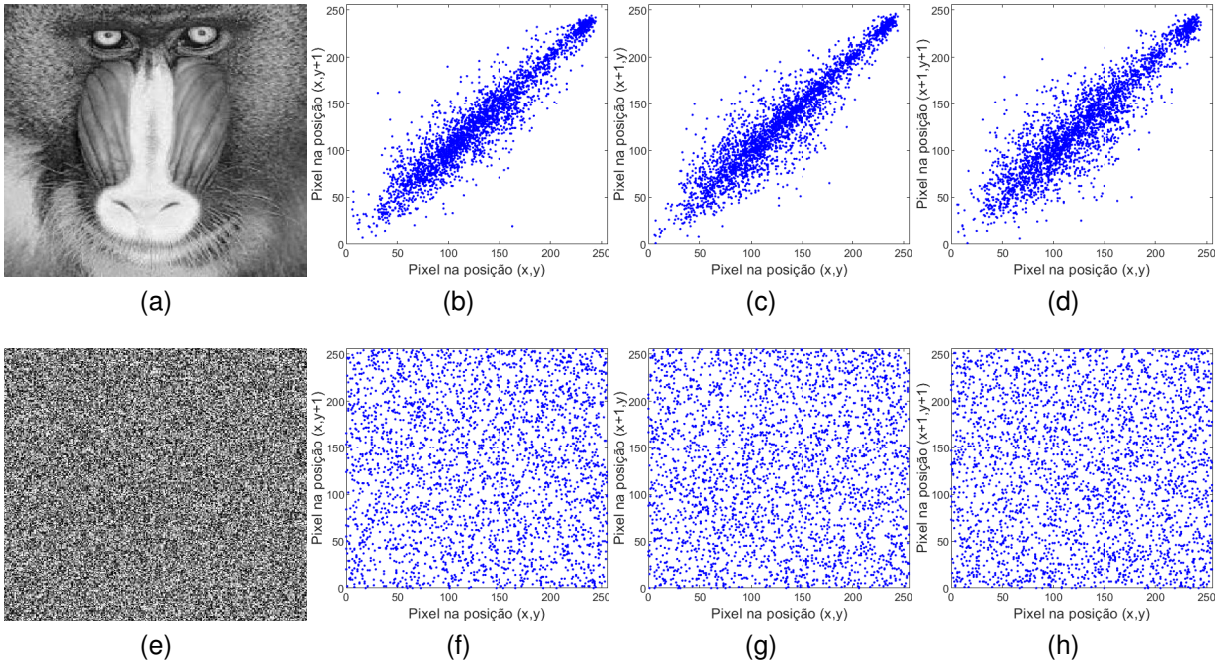


Figura 21: (a) Imagem Baboon. (b) Correlação horizontal da imagem original. (c) Correlação vertical da imagem original. (d) Correlação diagonal da imagem original. (e) Imagem Baboon cifrada. (f) Correlação horizontal da imagem cifrada. (g) Correlação vertical da imagem cifrada. (h) Correlação diagonal da imagem cifrada.

- Unified Average Changing Intensity (UACI), que é a média de valor de mudança destes pixels.

Esta sensibilidade também é dividida em relação à imagem e às chaves. Em relação à chave, uma imagem P é cifrada duas vezes, primeiramente com as chaves padrão gerando a imagem C_1, e por fim com cada chave modificada por 10^{-15}, gerando C_2. Para a sensibilidade em relação à imagem, a cifragem também ocorre duas vezes, primeiro com a imagem original intacta, gerando C_1, e depois com um pixel aleatoriamente selecionado da imagem tendo seu *bit* menos significativo modificado, gerando C_2. As equações para o NPCR e UACI são dadas pelas Equações (30) e (31), respectivamente.

$$NPCR = \frac{\sum_{i=1}^{M\times N} D(i)}{M \times N} \times 100\%. \tag{30}$$

Em que

$$D(i) = \begin{cases} 1 & C_1(i) \neq C_2(i), \\ 0 & C_1(i) = C_2(i). \end{cases}$$

$$UACI = \frac{1}{M \times N} \frac{\sum_{i=1}^{M\times N} |C_1(i) - C_2(i)|}{255} \times 100\%. \tag{31}$$

Realizando um teste estatístico, os valores críticos para o NPCR são dados pela

equação (32), em que $0,05, 0,01$ e $0,001$ são os níveis de significância, que são os valores mínimos que o NPCR deve assumir. Na Tabela 11, mostra-se que para ambos os testes, relativos à chave e à imagem, o cifrador apresenta valores de NPCR acima do limiar.

$$\nu^*_{0,05} = 99,5693\% \qquad \nu^*_{0,01} = 99,5527\% \qquad \nu^*_{0,001} = 99,5341\%. \tag{32}$$

Imagem	NPCR-IS	NPCR-KS	$\nu^*_{0,05}$	$\nu^*_{0,01}$	$\nu^*_{0,001}$
Baboon	99,61%	99,65%	Passou/Passou	Passou/Passou	Passou/Passou
Iris	99,63%	99,62%	Passou/Passou	Passou/Passou	Passou/Passou
Lena	99,63%	99,62%	Passou/Passou	Passou/Passou	Passou/Passou
White	99,59%	99,61%	Passou/Passou	Passou/Passou	Passou/Passou
Black	99,64%	99,58%	Passou/Passou	Passou/Passou	Passou/Passou

Tabela 11: NPCR para as imagens cifradas, medindo sensibilidade à mudança na imagem (IS) e à chave (KS).

Para o UACI, os valores devem estar nos intervalos dados pela equação (33), enquanto a Tabela 12 mostra que os valores desejados foram obtidos para ambos os casos.

$$\begin{aligned}
\mu^*_{0,05} &= (\mu^{*-}_{0,05}, \mu^{*+}_{0,05}) = (33,2824\%, 33,6447\%),\\
\mu^*_{0,01} &= (\mu^{*-}_{0,01}, \mu^{*+}_{0,01}) = (33,2255\%, 33,7016\%),\\
\mu^*_{0,001} &= (\mu^{*-}_{0,001}, \mu^{*+}_{0,001}) = (33,1594\%, 33,7677\%).
\end{aligned} \tag{33}$$

Imagem	UACI-IS	UACI-KS	$\mu^*_{0,05}$	$\mu^*_{0,01}$	$\mu^*_{0,001}$
Baboon	33,40%	33,34%	Passou/Passou	Passou/Passou	Passou/Passou
Iris	33,57%	33,45%	Passou/Passou	Passou/Passou	Passou/Passou
Lena	33,43%	33,52%	Passou/Passou	Passou/Passou	Passou/Passou
White	33,49%	33,54%	Passou/Passou	Passou/Passou	Passou/Passou
Black	33,43%	33,52%	Passou/Passou	Passou/Passou	Passou/Passou

Tabela 12: UACI para as imagens cifradas, medindo sensibilidade à mudança na imagem (IS) e à chave (KS).

Como mostrado nas Tabelas 11 e 12, o cifrador proposto neste trabalho apresenta valores de NPCR e UACI dentro dos intervalos desejados, e portanto, é seguro contra ataques diferenciais. No AES (ÇAVUŞOĞLU et al., 2018), os valores de UACI estão entre 30 e 31, dependendo da implementação, de forma que neste trabalho, a alta correlação entre pixels adjacentes das imagens originais não afeta sua segurança em relação à ataques diferenciais.

Imagem	Baboon	Iris	Lena	White	Black
MSE-IS	10.871	10.966	10.929	10.946	10.917
MSE-KS	10.773	10.908	10.966	10.959	10.936

Tabela 13: Valores de MSE para as imagens cifradas

Trabalho	MSE
Proposto	10.925,8/10.908,4
DIAB (2018)	10.109,7
AHMAD; HWANG (2015)	10.437,5
WANG; LIU; ZHANG (2015)	10.065,1
CHEN et al. (2015)	10.028,8

Tabela 14: Comparação dos valores de MSE.

3.3.7 Efeito avalanche

Uma propriedade desejada em algoritmos de criptografia é o efeito avalanche, que define que uma pequena mudança na imagem original ou na chave do cifrador deve ocasionar uma mudança significativa na imagem cifrada (AHMAD et al., 2018). O grau desta mudança é quantificado usando o erro médio quadrático, ou *Mean Square Error* (MSE), dado pela equação (34) para uma imagem de tamanho $M \times N$, em que C_1 e C_2 são duas imagens cifradas correspondendo a pequenas mudanças na chave ou na imagem. Um valor ideal de MSE é acima de 10.000 (DIAB, 2018), ou seja, que a média da diferença dos quadrados entre os pixels de duas imagens cifradas, uma com a chave/imagem padrão e outra com uma leve mudança, seja acima de 10.000,

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M \times N} (C_1(i) - C_2(i))^2. \tag{34}$$

Na Tabela 13, tem-se os valores de MSE para as imagens cifradas utilizando o cifrador proposto neste trabalho, mostrando que para todas as imagens, os valores estão acima de 10.000, tanto para mudanças relativas à chave como mudanças relativas à imagem. Na Tabela 14 os valores são comparados com outros trabalhos na literatura, mostrando que o efeito avalanche no nosso trabalho é mais forte do que nos outros citados.

3.3.8 Ataque de texto original conhecido e escolhido

A resistência de um cifrador aos ataques de texto original conhecido e escolhido está relacionada à dependência entre as sequências caóticas usadas para os processos de permutação e difusão e a imagem original (CHENG; WANG; XU, 2020; PATRO; ACHARYA, 2019; DENG et al., 2021). Neste trabalho, usa-se uma *hash* SHA-256, ge-

rada a partir da imagem original, que é usada para modificar as condições iniciais e os parâmetros dos mapas caóticos utilizados, assim estabelecendo uma dependência do cifrador em relação à imagem original, e garantindo a resistência contra estes dois tipos de ataque.

A resistência do cifrador proposto a estes ataques pode ser mostrada visualmente. Considera-se uma imagem com todos os pixels iguais, por exemplo, todos em 0, que gera uma imagem toda preta. Esta imagem é cifrada, e então, durante a decifragem, os valores caóticos usados para a permutação e para a difusão são armazenados. Em um cifrador não resistente aos ataques de texto original conhecido e escolhido, a aplicação dessas sequências em uma imagem cifrada retornaria a imagem original. Contudo, a aplicação destas sequências para recuperar a imagem Baboon retorna uma imagem ilegível, como mostra a Figura 22, e assim o cifrador é resistente aos ataques de texto original conhecido e escolhido.

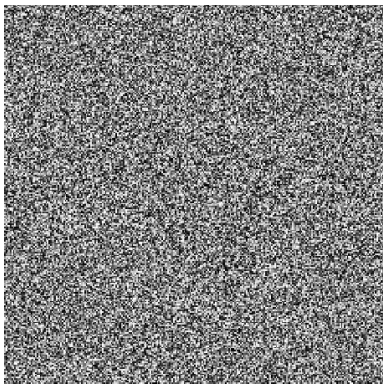


Figura 22: Imagem Baboon com decifragem insatisfatória ao usar as sequências derivadas da cifragem de uma imagem diferente.

3.3.9 Resistência à ruído

A comunicação ocorre em redes públicas, como a internet, e assim a comunicação pode apresentar problemas quanto ao ruído, de forma que seja necessário que o cifrador resista a estes ataques. Esta resistência é obtida medindo a qualidade de uma imagem recuperada após um ataque de ruído, dada pelo *Peak-Signal-to-Noise Ratio* (PSNR), de acordo com a equação (35), para uma imagem de tamanho $M \times N$, considerando que C_1 é a imagem original e C_2 é a imagem recuperada após um ataque de ruído, e MSE é dado pela equação (34).

$$PSNR = 10 \times \log_{10} \left(\frac{M \times N}{MSE} \right). \tag{35}$$

Neste trabalho, os testes foram realizados considerando três tipos de ruído: *Salt and Pepper*, de densidade 0,005, *Speckle*, de densidade 0,000002 e *Gaussian*, de densidade 0,000001, em adição a um ataque em que 25% da imagem é ocultada.

O ruído foi incluído utilizando o comando *imnoise* do MATLAB, nas imagens cifradas. Na Figura 23, mostra-se que ainda é possível reconhecer a imagem recuperada Lena para os quatro tipos de ataque, enquanto na Tabela 15, os valores de PSNR, após a aplicação destes ataques, são aplicados às cinco imagens deste trabalho, apresentando valores que são consistentes com outros trabalhos (HUANG et al., 2019; CHAI et al., 2019), o que indica que o cifrador é resistente a ataques de ruído.

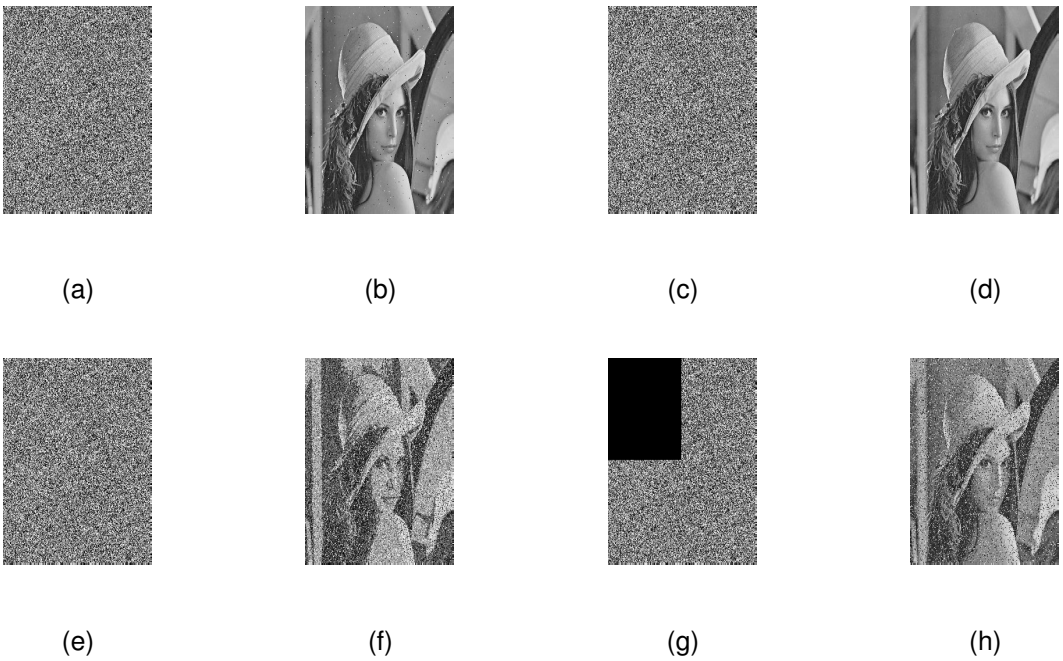


Figura 23: (a) Ruído *salt and pepper* com densidade de 0,005. (b) Imagem recuperada de a (a). (c) Ruído *speckle* com densidade de 0,000002. (d) Imagem recuperada de (c). (e) Ruído *gaussian* com densidade de 0,000001. (f) Imagem recuperada de (e). (g) Imagem 25% oculta (h) Imagem recuperada de (g).

Imagem	SP	S	G	O
Baboon	29,0566	46,4649	12,7690	14,9966
Lena	29,2686	45,0997	13,3594	15,1191
Iris	29,2923	45,2662	13,0821	15,2436
White	24,7885	41,3024	8,1349	10,7255
Black	24,9176	40,8231	8,1761	10,7270

Tabela 15: Valores de PSNR para ruídos SP (Salt and Pepper), S (Speckle), G (Gaussian) e O (Occlusion).

Outra métrica usada para medir a resistência do cifrador à ataques de ruído é a similaridade estrutural, que é medida pelo índice SSIM. O desejável é que este índice assuma valores próximos de 1 para que as imagens sejam semelhantes. A equação (36) é usada para fornecer esta métrica, entre duas imagens C_1 e C_2, em que C_1 é

a imagem original e C_2 é a imagem recuperada. Ademais, μ_{C_1} e μ_{C_2} são os valores médios de C_1 e C_2, com $\sigma^2_{C_1}$ e $\sigma^2_{C_2}$ representando as variâncias, $\sigma_{C_1C_2}$ é a covariância de C_1 e C_2, e por padrão, usam-se as constantes $k_1 = 0,01, k_2 = 0,03$ para estabilizar o denominador.

$$SSIM = \frac{(2\mu_{C_1}\mu_{C_2} + k_1)(2\sigma_{C_1C_2} + k_2)}{(\mu^2_{C_1} + \mu^2_{C_2} + k_1)^2(\sigma^2_{C_1} + \sigma^2_{C_2} + k_2)}. \tag{36}$$

Na Tabela 16, são mostrados os valores do SSIM calculados entre a imagem original e a recuperada, após a imagem cifrada ser poluída com ruido.

Imagem	SP	S	G	O
Baboon	0,9106	0,9961	0,1570	0,2582
Lena	0,8818	0,9933	0,1367	0,2046
Iris	0,8676	0,9897	0,0623	0,1135
White	0,7116	0,9534	0,0537	0,0843
Black	0,6819	0,9531	0,0219	0,0475

Tabela 16: Valores de SSIM para ruídos SP (Salt and Pepper), S (Speckle), G (Gaussian) e O (Occlusion).

3.3.10 Desempenho e complexidade

Nesta seção, o desempenho e a complexidade do algoritmo proposto é analisada. O cifrador foi implementado em linguagem C, em um computador Acer Aspire 3, com processador Intel Core i5 240-250 GHz, 4GB de RAM e sistema operacional Windows 10,

O cifrador apresenta arquitetura permutação-difusão, em que a imagem é percorrida por duas vezes, uma para cada processo. Com isto, a complexidade do algoritmo é de ordem $O(M \times N)$, que é a mesma que os outros trabalhos citados na Tabela 18, com exceção de FARAH; FARAH; FARAH (2019), que apresenta complexidade de ordem $O(M \times N \log MN)$.

Na Tabela 17, é mostrado o tempo necessário para cifrar uma imagem, computado a partir da média de tempo de 50 cifragens de cada uma das 5 imagens utilizadas. O baixo tempo utilizado evidencia a eficiência do cifrador para aplicações de tempo real, e na Tabela 18, mostra-se que o cifrador consegue cifrar mais informações, medidas em *mega-bytes* (MB), por segundo que outros cifradores da literatura (CHAI et al., 2019; FARAH; FARAH; FARAH, 2019; PATRO; ACHARYA, 2019; ZHANG et al., 2020; WANG; LIU; ZHANG, 2015) que foram implementados em máquinas similares, assim apresentando melhor desempenho. A equação de *throughput*, que é a quantidade de *mega-bytes* que é cifrada por segundo, é dada pela equação (37), em que s é o

tamanho da imagem em *mega-bytes* e t é o tempo de cifragem, em segundos.

$$T = \frac{s}{t}. \tag{37}$$

Comparando com o cifrador simétrico mais tradicional, o AES, também mostra-se a viabilidade do uso do cifrador proposto para a cifragem de imagens. No trabalho de ARAB; ROSTAMI; GHAVAMI (2019), a simulação computacional da implementação em linguagem C da versão tradicional do AES leva em torno de 500 s, enquanto com melhoras no código foi possível reduzir esse tempo para 2 s. No trabalho de ÇAVUŞOĞLU et al. (2018), a simulação computacional da implementação padrão leva em torno de 60 s, enquanto a versão otimizada leva aproximadamente 10 s. Em ambos os casos, as otimizações reduziram o consumo de tempo do AES, mas o consumo de tempo computacional ainda é consideravelmente superior ao deste trabalho.

Tamanho	256×256	512×512	1024×1024
Tempo (ms)	25	105	385

Tabela 17: Tempo necessário para cifrar uma imagem, em milissegundos.

Trabalho	*Throughput* (MB/s)
Proposto	2,62
CHAI et al. (2019)	1,28
FARAH; FARAH; FARAH (2019)	0,03
PATRO; ACHARYA (2019)	2,38
ZHANG et al. (2020)	1,43
WANG; LIU; ZHANG (2015)	1,5

Tabela 18: Comparação de *throughput*.

4 CIFRADOR COM PERMUTAÇÃO-DIFUSÃO SIMULTÂNEA

Neste capítulo, o cifrador com arquitetura permutação-difusão simultânea é apresentado. Primeiro, apresenta-se a motivação do uso desta arquitetura, depois o algoritmo em si é apresentado, e depois é feita a análise dos resultados, do ponto de vista de segurança e de desempenho. Por fim, é realizada a comparação com o AES.

4.1 Motivação

Em geral, em cifradores caóticos, as operações de permutação e difusão são realizadas separadamente, como é o caso no cifrador proposto no capítulo anterior. Nestes cifradores, o desempenho é adequado para troca de arquivos de tamanho pequeno em tempo real, mas não há uma melhora significativa se comparado ao AES implementado por ZHANG (2018). Dessa forma, levando em conta que o AES não é otimizado para a cifragem de imagens, existe a necessidade de melhorar o desempenho dos cifradores caóticos para que sejam aptos para a comunicação em tempo real de imagens de alta definição. O caminho mais nítido para obter esta melhora é a diminuição no número de vezes que a imagem é escaneada durante a cifragem. Na maioria dos trabalhos até agora, como o cifrador apresentado no Capítulo 3 e publicado em BEZERRA; ALMEIDA CAMARGO; MOLTER (2021), e outros como PATRO; ACHARYA (2019); CHAI et al. (2019); ZHANG et al. (2020), em que a permutação-difusão é separada, é necessário escanear a imagem pelo menos três vezes. Isto ocorre pois é necessária uma varredura da imagem para gerar a *hash*, outra para a permutação e outra para a difusão. Tornando o processo de permutação-difusão simultâneo, é possível economizar uma varredura, assim evitando acessos a memória e ao disco, que são custosos computacionalmente.

Existem alguns trabalhos na literatura que utilizam permutação-difusão simultânea, e estes trabalhos são apresentados e discutidos a seguir.

- No cifrador de DIAB (2018), a permutação-difusão simultânea é utilizada em conjunto com outra operação que mistura os pixels na direção vertical e também na direção horizontal. Contudo, como destacado em (HUANG et al., 2019), as ope-

rações de mistura de pixels exigem uma varredura da imagem, assim, contando com a *hash* e a permutação-difusão simultânea, ainda há três varreduras.

- O cifrador de HUANG et al. (2019) também apresenta problemas de desempenho, pois além da permutação-difusão simultânea, há uma varredura para redimensionar a imagem original antes da cifragem, e outra para obter a soma dos pixels, que além de custosa computacionalmente, é menos segura do que a *hash* contra ataques de informação escolhida.

- No cifrador de LIU; LEI; WANG (2020), a difusão é separada em difusão por linha e por coluna, cada uma exigindo uma varredura. A dependência entre a imagem original e as sequências caóticas é gerada a partir da soma dos valores dos pixels, que exige outra varredura. Além disso, a soma é escondida em um pixel da imagem cifrada, ao invés de ser trocada durante o protocolo de troca de chaves, assim, a imagem original não é 100% recuperada na decifragem, e o cifrador também não é resistente a ruídos.

- O trabalho de TALHAOUI; WANG (2021) é similar ao de LIU; LEI; WANG (2020), utilizando permutação-difusão simultânea tanto nas linhas como nas colunas, e cada uma requer uma varredura. A terceira varredura se dá na geração da dependência entre as sequências caóticas e a imagem original.

- No trabalho de ENAYATIFAR et al. (2017), operações de DNA são usadas para a permutação-difusão simultânea. Como as operações de DNA são custosas (CHAI et al., 2019), o desempenho do cifrador é limitado.

- Em WANG; YANG; GUAN (2021), antes da permutação-difusão simultânea, os pixels são substituídos por outros valores, de acordo com uma caixa de substituição, que atribui a um pixel um valor equivalente, assim exigindo uma varredura a mais, além da permutação-difusão simultânea e da geração da dependência entre as sequências caóticas e a imagem original.

A revisão dos trabalhos recentes de criptoanálise indica que o ponto fraco de segurança dos cifradores que foram quebrados é a dependência fraca das sequências caóticas em relação à imagem original. A dependência necessita ser mais robusta do que a soma ou a média dos pixels da imagem original, pois a soma ou a média pode ser igual para diferentes imagens, e assim, o cifrador ainda é sensível à ataques da informação escolhida (FAN et al., 2018; CHEN; CHEN; ZHOU, 2020; MA; LI; OU, 2020; LIU; ZHANG; CHEN, 2019; ALANAZI et al., 2021; HUANG et al., 2020; CHEN et al., 2018). Outros problemas são a degradação dinâmica e o uso mapas caóticos de distribuição não-uniforme (LIU et al., 2020), dois tópicos que são abordados na seção

2 deste trabalho, com o mapa da tenda apresentação distribuição uniforme, e a perturbação a cada 100 iterações do valor do parâmetro sendo usada para evitar degradação dinâmica. Estratégias citadas na revisão bibliográfica de permutação-difusão simultânea, como operações de DNA, caixas de substituição e mistura de pixels são utilizadas para adicionar uma nova camada de segurança, mas da análise dos trabalhos recentes de criptoanálise tem-se que permutação-difusão com dependência forte entre a imagem original e as sequências caóticas, por meio de funções *hash*, é mais que suficiente para garantir a segurança do cifrador. Isto se dá pois nenhum dos trabalhos citados acima quebrou um cifrador que usa *hash* para gerar a dependência entre a imagem original e as sequências caóticas.

Neste contexto, é proposto neste capítulo um cifrador com arquitetura de permutação-difusão simultânea, com o cifrador sendo seguro e mais rápido, pois apenas duas varreduras da imagem são necessárias. A aplicação de testes mostra que o cifrador é resistente a ataques estatísticos, diferenciais, da informação escolhida e de ruído, além da apresentar desempenho superior à outros trabalhos da literatura e ao AES.

4.2 Algoritmo

Nesta seção, apresenta-se o algoritmo, primeiro o processo de cifragem, e depois o processo de decifragem.

4.2.1 Cifragem

Assim como na versão do cifrador proposta no capítulo 3, as sequências caóticas são geradas a partir do mapa da tenda, dado pela equação (13), com a função de quantização dada pela equação (14).

A cifragem é realizadas tanto para imagens em preto e branco como para imagens coloridas. Em imagens coloridas, cada canal de cor, pode ser tratado como uma imagem em preto e branco separada, ou ainda como a mesma imagem, com o número de colunas triplicado. Neste trabalho trabalha-se com o número de colunas triplicado, pois tratar cada canal como uma imagem separada faria com que fossem necessárias duas varreduras extras da imagem.

Os mapas caóticos utilizados são os seguintes:

- x_P, que é usado para determinar qual será o pixel permutado com o pixel atual durante a varredura da imagem. Deste mapa, há duas chaves secretas, x_{P_0}, que é a condição inicial, e p_P, que é o valor do parâmetro.

- x_D, para o processo de difusão, com chaves x_{D_0} e p_D, que são a condição inicial e o valor do parâmetro, respectivamente.

Além disso, assim como no cifrador proposto no Capítulo 3, a *hash* $SHA-256$ faz parte da chave secreta, garantindo a segurança do cifrador contra ataques da informação escolhida, além do valor de 8 *bits* C_{in}, que é usado para as operações *xor* entre o pixel cifrado atual e o anterior.

Os passos de cifragem são os seguintes:

Passo 1: A imagem original I, de dimensões $M \times N \times C$, é carregada.

Passo 2: A *hash* $SHA-256$ é aplicada, retornando uma sequência de 256 bits $S = S(1)S(2)S(3)\cdots S(256)$.

Passo 3: As chaves secretas são perturbadas. Primeiro, as perturbações λ são geradas a partir da *hash* obtida no passo anterior, da seguinte maneira:

$$\begin{aligned} &\lambda_1 = S(1:64), && \lambda_2 = S(65:128),\\ &\lambda_3 = S(129:192), && \lambda_4 = S(192:256),\\ &\lambda_5 = S(249:256). \end{aligned}$$

Para as perturbações relacionadas aos mapas caóticos, que são $\lambda_{1:4}$, são desejados valores no intervalo $(0,0,05)$, pois as chaves secretas relacionadas à estes mapas, x_{P_0}, x_{D_0}, p_P e p_D, assumem valores no intervalo $(0,0,95)$. Para λ_5, que irá realizar uma operação *xor* com C_{in}, o objetivo é um valor entre 0 e 255. Para isso, estes valores são convertidos para decimal, e depois multiplicados por 2.7174×10^{-21}. Então:

$$\lambda_{1:5} \rightarrow \text{conversaoDecimal}(\lambda_{1:5}), \qquad \lambda_{1:4} \rightarrow 2.7174 \times 10^{-21}\lambda_{1:4}.$$

Após encontrar as perturbações λ, as chaves são modificadas da seguinte forma:

$$\begin{aligned} &x_{P_0} = x_{P_0} + \lambda_1, && p_P = p_P + \lambda_2,\\ &x_{D_0} = x_{D_0} + \lambda_3, && p_D = p_D + \lambda_4,\\ &C_{in} = C_{in} \oplus \lambda_5. \end{aligned}$$

Passo 4: Para evitar o efeito transiente, 50 iterações de cada mapa caótico são resolvidas e descartadas.

Passo 5: As sequências caóticas são geradas, de acordo com a equação (38).

$$\begin{aligned} x_P &= (x_P(0), x_P(1), \cdots, x_P([M-1][NC-1])),\\ x_D &= (x_D(0), x_D(1), \cdots, x_D([M-1][NC-1]))). \end{aligned} \tag{38}$$

Após isso, usando a equação (14), estes valores são convertidos para inteiro, ge-

rando as sequências caóticas dadas pela equação (39).

$$h_P = (h_P(0), h_P(1), \cdots, h_P([M-1][NC-1])), \\ h_D = (h_D(0), h_D(1), \cdots, h_D([M-1][NC-1]))). \tag{39}$$

Para a operação de permutação-difusão simultânea, o pixel na posição (i, j) é trocado com o pixel na posição escolhido de acordo com a sequência caótica h_P, e então a difusão é aplicada com o valor dado pela sequência caótica h_D, e após isto, o valor não é mais modificado. Isto significa que após cada iteração, o número de pixels disponíveis para a permutação é subtraído por 1, então, para o valor de $h_P(i)$, a operação de resto inteiro na equação (14) é aplicada relativa à $M \times N \times C - i$, ou seja:

$$h_P(i) = [2^{33} \times x_P(i)] \mod MNC - i.$$

Para a difusão, a operação envolve valores inteiros de 8 *bits*, ou seja, o resto é calculado em relação à $2^8 = 256$, e dessa forma:

$$h_D(i) = [2^{33} \times x_D(i)] \mod 256.$$

Passo 6: Aplica-se a operação de permutação-difusão simultânea. A imagem é tratada como um vetor unidimensional, de tamanho $M \times N \times C$, e a operação é dada pela equação (40), para $i \in [0, M-1]$ e $j \in [0, CN-1]$. Para imagens coloridas, considera-se $C = 3$, e para imagens em preto e branco, considera-se $C = 1$. Após este processo, a imagem cifrada é dada por E. Na difusão do primeiro pixel, a operação *xor* com o último pixel cifrado é realizada com C_{in}, o número inteiro de 8 *bits* que é parte da chave secreta.

$$I(i \times NC + j) \Leftrightarrow I(h_P(i \times NC + j)), \\ E(i \times NC + j) = I(i \times NC + j) \oplus h_D(i \times NC + j) \oplus E(i \times NC + j - 1). \tag{40}$$

Na Figura 24, o fluxograma do algoritmo é dado.

4.2.2 Decifragem

Fundamentalmente, a decifragem é o inverso da cifragem, apenas descartando a aplicação da *hash*, pois esta já é recebida no protocolo de troca de chaves. Os passos são dados a seguir.

- **Passo 1:** A imagem cifrada E é lida, com as mesmas dimensões da imagem original.
- **Passo 2:** Este passo é equivalente ao passo 3 da cifragem, em que as chaves

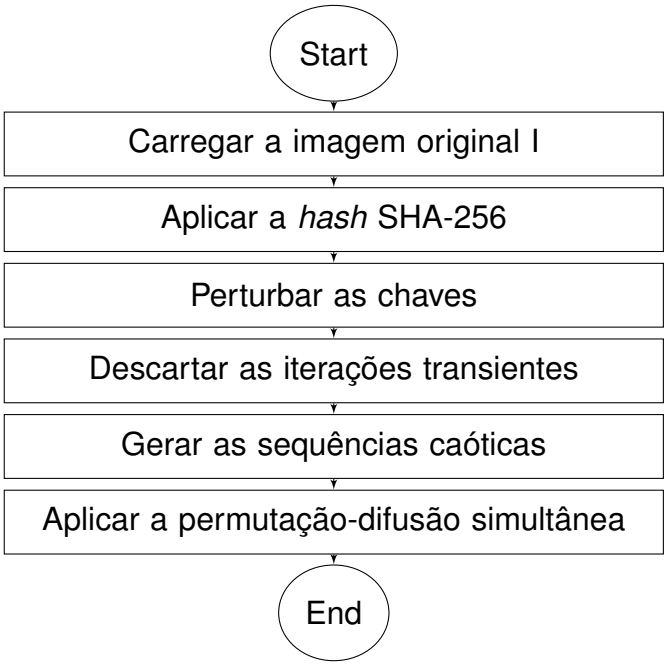


Figura 24: Fluxograma para o algoritmo do cifrador de permutação-difusão simultânea.

são perturbadas.

Passo 3: Este passo é equivalente ao passo 4 da cifragem, em que há o descarte das iterações transiente do mapa caótico.

Passo 4: Este passo é equivalente ao passo 5 da cifragem, em que há a geração das sequências caóticas.

Passo 5: Este passo é equivalente ao passo 6 da cifragem, em que a permutação-difusão simultânea é aplicada. Na decifragem, a ordem é inversa à cifragem, ou seja, começa no pixel inferior direito e vai até o pixel superior esquerdo.

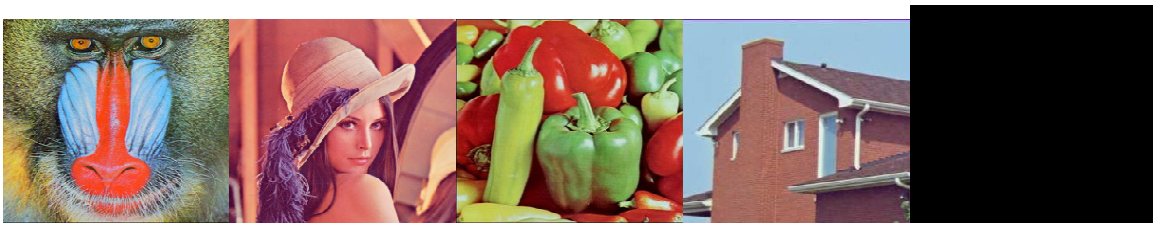
4.3 Análise de segurança

Nesta seção, é realizada a análise da segurança do cifrador, relativa aos seguintes ataques:

- Ataque estatístico, relacionado a aleatoriedade da imagem cifrada.
- Ataque diferencial, relacionado a sensibilidade do cifrador a mudanças na imagem original ou na chave.
- Ataque da informação escolhida, relacionado a dependência entre as sequências caóticas e a imagem original;
- Ataque de força bruta, relacionado ao espaço de chaves.

- Ataques de ruído, relacionado a aptidão do cifrador para comunicação em redes públicas como a internet.

Os testes são realizados em imagens coloridas de dimensão $256 \times 256 \times 3$, dadas pela Figura 25, e também são validos para imagem em preto e branco, pois uma imagem colorida é composta por três imagens em preto e branco.



(a) (b) (c) (d) (e)

Figura 25: (a) Baboon. (b) Lena. (c) Pepinos. (d) Casa. (e) Black.

4.3.1 Ataques estatísticos

A habilidade do cifrador de resistir ataques estatísticos é relacionada a uniformidade da imagem cifrada, e métricas apresentadas no capítulo anterior como coeficientes de Pearson, variância de histograma, entropia de Shannon, testes do NIST e coeficientes de correlação.

- O coeficiente de Pearson mede a uniformidade do histograma, e é dado pela equação (25). Os valores do coeficiente devem estar abaixo de $293,248$, e na Tabela 19, tem-se que isto ocorre neste cifrador, mostrando que por esta métrica o histograma é uniforme.

- A variância do histograma é dada pela equação (26), e na Tabela 19, tem-se que os resultados são consistentes com (PING et al., 2018), que cifra imagens de tamanho similar.

- A entropia de Shannon é dado pela equação (27), e deve assumir valores perto de 8, o que a Tabela 19 mostra que ocorre neste cifrador.

- Na Tabela 20, o *p-value* de todos os testes apresenta valor acima de $0,01$, de forma que a imagem cifrada passa nos testes de aleatoriedade.

- Os coeficientes de correlação são dados pela equação (29), e as imagens cifradas devem apresentar coeficientes próximos de 0. Na Tabela 21, tem-se que as imagens cifradas apresenta coeficientes de correlação próximos de 0, assim garantindo a segurança contra ataques estatísticos por esta métrica.

	Baboon	Black	Casa	Lena	Pepinos
Pearson	239	237	251	238	262
Variância do histograma	738	708	777	787	837
Entropia de Shannon	7,99913	7,99913	7,99908	7,99913	7,99891

Tabela 19: Testes estatísticos para as imagens cifradas.

	Baboon	Black	Casa	Lena	Pepinos
Monobit	0,1294	0,9390	0,4909	0,5194	0,8985
Block Frequency	0,1830	0,0790	0,1962	0,4741	0,9692
Runs	0,9316	0,2589	0,7092	0,4557	0,4603
Longest Runs	0,2329	0,5128	0,3732	0,9196	0,3203
Rank	0,2165	0,8254	0,7485	0,2450	0,8067
DFT	0,8929	0,7675	0,3512	0,3976	0,4934
Non-overlapping Template	0,8047	0,8812	0,7712	0,6222	0,4015
Overlapping Template	0,6827	0,7151	0,5757	0,3739	0,5773
Universal	0,7246	0,2511	0,7779	0,7444	0,7919
Linear Complexity	0,9669	0,6620	0,9268	0,2337	0,2727
Serial 1	0,4850	0,6333	0,6572	0,8470	0,7139
Serial 2	0,5412	0,5904	0,8261	0,9323	0,7271
Approximate Entropy	0,9534	0,4044	0,1732	0,7565	0,6505
Cusums Forward	0,2043	0,8291	0,2033	0,1980	0,9592
Cusums Backward	0,1418	0,7595	0,6699	0,6242	0,9749
Random Excursions	0,6817	0,5014	0,2353	0,4434	0,6322
Random Excursions Variant	0,7169	0,6804	0,5428	0,6878	0,6272

Tabela 20: Testes do NIST para as imagens cifradas.

Nesta subseção, testes foram realizados para verificar a segurança do cifrador contra ataques estatísticos. A uniformidade do histograma é mensurada utilizando o coeficiente de Pearson e a variância do histograma, na Tabela 19, que também mostra que os valores da entropia de Shannon são próximos do valor ideal de 8, o que além com os resultados dos testes do NIST na Tabela 20, indicam que as imagens cifradas são aleatórias. Além disso, coeficientes de correlação próximos de 0 na Tabela 21 mostram que os pixels da imagem cifrada não estão correlacionados, e assim, este conjunto de testes mostra que o cifrador é resistente à ataques estatísticos.

4.3.2 Ataque diferencial

A resistência do cifrador à ataques diferenciais é relacionada ao seu processo de difusão, ou seja, o quão forte é o efeito causado por uma pequena modificação na imagem original ou na chave. Esta resistência é medida por três métricas.

- NPCR, dado pela equação (30), que deve assumir valores acima de 99,50% (ALGHAFIS et al., 2020).

Imagem	Canal	Vertical	Horizontal	Diagonal
Baboon	R	-0,0016	0,0016	0,0020
	G	0,0024	0,0012	-0,0051
	B	0,0021	0,0036	-0,0065
Black	R	-0,0005	-0,0015	0,0007
	G	-0,0002	-0,0054	0,0027
	B	-0,0028	-0,0031	-0,0043
Casa	R	-0,0052	-0,0064	0,0021
	G	0,0064	0,0017	-0,0024
	B	0,0026	0,0004	-0,0027
Lena	R	-0,0049	-0,0006	-0,0009
	G	0,0015	0,0012	-0,0028
	B	0,0059	0,0070	-0,0021
Pepinos	R	0,0031	-0,0037	-0,0087
	G	-0,0095	-0,0045	0,0025
	B	-0,0052	-0,0022	0,0004

Tabela 21: Coeficientes de correlação das imagens cifradas.

- UACI, dado pela equação (31), que deve apresentar valores acima de 33,35% (ALGHAFIS et al., 2020).
- MSE, dado pela equação (34), que deve apresentar valores acima de 10,000 (DIAB, 2018).

Na Tabela 22, as métricas são calculadas, e para todas as imagens, apresentam valores acima do limiar, o que implica que o cifrador é resistente a ataque diferencial.

Image	NPCR-I	NPCR-K	UACI-I	UACI-K	MSE-I	MSE-K
Baboon	99,62%	99,60%	33,46%	33,45%	10929	10921
Black	99,61%	99,62%	33,44%	33,53%	10921	10965
Casa	99,61%	99,62%	33,44%	33,40%	10900	10868
Lena	99,60%	99,60%	33,42%	33,40%	10907	10905
Pepinos	99,63%	99,63%	33,46%	33,47%	10934	10931

Tabela 22: Testes para medir a resistência do cifrador a ataques diferenciais, com I representando sensibilidade à mudanças na imagem e K representando sensibilidade à mudanças na chave.

4.3.3 Ataques da informação escolhida e força bruta

A segurança contra ataques da informação escolhida é relacionada a dependência entre a imagem original e as sequências caóticas gerada. Neste trabalho, usa-se a *hash* $SHA-256$ para gerar esta dependência. Para quantificar a segurança do cifrador a este tipo de ataque, usa-se a métrica de similaridade estrutural SSIM, dada pela

equação (36), neste caso calculada entre a imagem original e a imagem recuperada após uma tentativa de ataque pela informação escolhida. Na Tabela 23, tem-se que esta métrica apresenta resultados próximos de 0, o que implica que não há similaridade entre a imagem original e a recuperada, de forma que o cifrador é resistente a ataques da informação escolhida.

Image	Baboon	Black	Casa	Lena	Pepinos
SSIM	0,0087	0,0004	0,0078	0,0076	0,0085

Tabela 23: SSIM usado para comparar a imagem original com a imagem recuperada utilizando o ataque da informação escolhida.

A resistência de um cifrador à ataques de força bruta é dependente do seu espaço de chaves, que deve ser superior à 2^{100} (ALVAREZ; LI, 2006) ou 2^{256} contando com o desenvolvimento da computação quântica em algumas décadas (PAAR; PELZL, 2010). Este cifrador apresenta 4 chaves relacionadas aos mapas caóticos, cada uma com espaço de chaves de 10^{15} (STEVENSON, 1985), adicionado 10^{60} ao espaço de chaves, além da *hash* com espaço de chaves de 2^{256} e da condição inicial C_{in}, com espaço de chaves 2^8. Dessa forma, o espaço de chaves do cifrador de permutação-difusão simultânea é de $10^{60} \times 2^{264}$, de forma que é resistente a ataques de força bruta.

4.3.4 Ataques de ruído

A métrica $PSNR$, dada pela equação (35), calculada entre a imagem original e a imagem recuperada quando a imagem original é poluída com ruído, é usada para quantificar a resistência do cifrador à estes ataques de ruído. Na Tabela (24), tem-se que os valores são consistentes com os encontrados na Tabela 15, e dessa forma, o cifrador com permutação-difusão simultânea é resistente a ataques de ruído.

Imagem	Salt and Pepper	Speckle	Gaussian	Occlusion
Baboon	28,9656	44,9703	12,9367	14,9595
Black	24,6812	41,3129	8,1102	10,7554
Casa	28,9112	45,6022	12,9720	14,8809
Lena	28,6983	45,5032	12,5166	14,6398
Pepinos	28,3590	45,0858	12,0731	14,1104

Tabela 24: PSNR para as imagens recuperadas quando a respectiva imagem cifrada é poluída com ruído.

Na Tabela 25, a métrica de similaridade estrutural, dada pela equação (36), é calculada, e os valores também são consistentes com a tabela 16, que é outra métrica que mostra que o cifrador é resistente a ataques de ruido.

Imagem	Salt and Pepper	Speckle	Gaussian	Occlusion
Baboon	0,9570	0,9985	0,3460	0,4624
Black	0,4654	0,9310	0,0002	0,0002
Casa	0,9404	0,9981	0,2561	0,3656
Lena	0,9713	0,9994	0,3888	0,5292
Pepinos	0,9710	0,9992	0,4092	0,5373

Tabela 25: SSIM para as imagens recuperadas quando a respectiva imagem cifrada é poluída com ruído.

4.4 Análise de desempenho

Nesta seção, o desempenho do cifrador é analisado, no mesmo ambiente que o cifrador implementado no Capítulo 3, ou seja, processador Intel Core i5 2.4-2.5 GHz, com 4 GB de memória e sistema operacional Windows 10. Para efeitos de comparação, as métricas são dadas para imagens em preto e branco, e a implementação do cifrador se deu tanto no MATLAB como em linguagem C.

Primeiro, é realizada a comparação com outros trabalhos, no MATLAB. Os cifradores comparados também apresentam arquitetura permutação-difusão simultânea, e na Tabela 26, mostra-se que o cifrador proposto neste trabalho apresenta consumo de tempo inferior à estes outros trabalhos, pois requer ao menos uma varredura a menos da imagem, sem sacrificar a segurança.

Cifrador	Tempo de Cifragem (s)
Proposto	0,0359
(WANG; YANG; GUAN, 2021)	0,1847
(HUANG et al., 2019)	0,3039
(LIU; LEI; WANG, 2020)	0,2768
(ENAYATIFAR et al., 2017)	0,56

Tabela 26: Comparação do tempo necessário para cifrar uma imagem em preto e branco de dimensões 256×256, no MATLAB.

Na Tabela 27, são dados os tempos de cifragem e defragem para imagens em preto e branco de diferentes dimensões. Esta tabela mostra que para imagens de dimensão até 1024×1024, o cifrador é propício para aplicações em tempo real, por gastar menos de 300 ms (KUDRLE et al., 2010). Além disso, para imagens de dimensão 256×256, o *throughput* é o triplo do encontrado no cifrador de permutação-difusão separada no Capítulo 3. Para imagens de dimensões maiores, há uma queda no *throughput*, por causa que a limitação da memória do computador faz com que sejam necessários acessos ao disco.

Dim.	Cif. (s)	Dec. (s)	Total (s)	Throughput (MB/s)
256×256	0,007	0,009	0,016	9,36
512×512	0,03	0,033	0,063	8,74
1024×1024	0,11	0,123	0,233	9,53
2048×2048	0,554	0,627	1,181	7,57
4096×4096	2,27	2,35	4,72	7,39
8192×8192	10,39	11	21,39	6,46

Tabela 27: Tempo necessário para a cifragem e a decifragem de imagens em preto e branco de diferentes dimensões.

4.5 Comparação com o AES

No trabalho de ZHANG (2018), o AES foi implementado, em conjunto com uma operação de *Cipher Block Chaining* (CBC) para garantir difusão, que envolve operações *exclusive-or* entre o bloco cifrado atual e o anterior, em conjunto com um vetor de inicialização aleatório (PAAR; PELZL, 2010).

A análise de segurança do AES para a cifragem de imagens é realizada levando em conta a resistência a ataques estatísticos, apresentando coeficientes de correlação próximos de 0 e entropia de Shannon próxima de 8, como também é o caso no cifrador proposto neste capítulo, assim sendo resistente a ataques estatísticos. Quanto a resistência a ataques diferenciais, as métricas NPCR e UACI são usadas, apresentando valores próximos de 99.60% e 33.45% respectivamente, assim sendo resistente a estes ataques. Por fim, é feita a análise de resistência contra ataques de força bruta, e o espaço de chaves de 2^{256} indica que o AES é resistente a ataques de força bruta. Deste parágrafo, tem-se que assim como o cifrador proposto neste capítulo, o AES é seguro para a cifragem de imagens.

Onde o cifrador proposto destoa do AES é no desempenho. Como este cifrador foi implementado em uma arquitetura diferente do AES em ZHANG (2018), o tempo de implementação da Transformada Rápida de Fourier (FFT) no MATLAB foi usada para realizar a comparação.

- No ambiente de simulação de Zhang, a FFT é realizada em 0,0186 s, enquanto a cifragem mais rápida para uma imagem em preto e branco de dimensões 256×256 leva em média 0,022 s, enquanto a decifragem leva em média 0,024 s, de forma que o processo total leva 0,046 s, que é 2,47 vezes o tempo necessário para a realização da FFT.

- No ambiente de simulação em que este trabalho é realizado, a FFT é realizada em 0,0262 s. A cifragem para uma imagem em preto e branco de dimensão 256×256 é realizada em 0,007 s, enquanto a decifragem é realizada em 0,009 s, com um tempo total de 0,016 s, que é apenas 0,61 do tempo necessário para a FFT,

de forma que o cifrador de permutação-difusão simultânea é aproximadamente quatro vezes mais rápido que o AES.

Os resultados apresentados acima evidenciam que o cifrador de permutação-difusão simultânea apresenta uma melhora de desempenho significativa comparado ao AES para a cifragem de imagens, sem pecar na segurança.

5 ALGORITMO PARA APLICAÇÕES NA TELEMEDICINA

Neste capítulo, o algoritmo para cifragem de imagens médicas, que junta a inserção de informações textuais com a cifragem de uma imagem, é descrito. Primeiro, é feita a revisão bibliográfica acerca de trabalhos que utilizam a criptografia na telemedicina, depois o cifrador é descrito. Por fim, testes mostram que a imagem original é recuperada com baixa distorção, que a cifragem é segura contra várias formas de ataque, e que o procedimento é mais rápido que os de outros trabalhos na literatura.

5.1 Revisão bibliográfica

Para a troca de arquivos médicos, é necessária a proteção destes arquivos, por intermédio de técnicas de criptografia, como por exemplo cifragem, já abordada nos capítulos prévios, e a marca d'água, que consiste na inserção de um arquivo dentro de outro arquivo, por exemplo, um texto dentro de uma imagem, com o objetivo de escondê-lo. Neste caso, bastante aplicado na telemedicina, após a imagem original ser recebida, de forma cifrada, ela precisa ser decifrada sem distorções, para evitar equívocos no diagnóstico, que podem por a vida do paciente em risco. A marca d'água é usada para garantir a integridade e a autenticidade de um documento, sendo que a integridade protege o documento de ser modificado por uma parte não autorizada, enquanto a autenticidade garante que o documento foi recebido de sua fonte original (SHIH et al., 2018). Em geral, informações do paciente, tais como identidade, nome, gênero, idade e endereço são utilizadas para ser colocadas dentro de uma imagem de diagnóstico (SANIVARAPU et al., 2020).

A marca d'água pode ser realizada tanto no domínio espacial como no domínio de frequência de uma imagem. No domínio espacial, a informação é inserida em pixels selecionados de forma aleatória, tendo como vantagem o baixo custo (SHIH et al., 2018). No domínio da frequência, a informação é inserida nos coeficientes de uma imagem, como por exemplo, os coeficientes de uma transformada de Fourier, aumentando a imperceptibilidade (ELHOSENY et al., 2018a). A marca d'água também pode ser classificado como reversível ou não-reversível, e como frágil ou robusto. No re-

versível, tanto a imagem original, quanto as informações embutidas na imagem, são 100% recuperáveis, enquanto no não reversível, apenas as informações embutidas são 100% recuperáveis. A marca d'água frágil, o qual cobre grande parte da literatura, é quando as informações embutidas não são mais legíveis se o documento for modificado por partes não-autorizadas, assim revelando que a integridade do documento foi prejudicada. No caso robusto, a informação ainda é legível mesmo se o documento foi modificado.

Exemplos recentes de marca d'água digital na literatura incluem:

- SANIVARAPU et al. (2020), no qual as informações do paciente foram inseridas em um sinal de eletrocardiograma, usando transformadas no domínio da frequência para decompor a imagem de duas dimensões, capturando informações sutis do sinal, e transformando as informações do paciente em um código *Quick Response* (QR) para ser embutida. Este esquema pode ser implementado em tempo-real, tendo como único ponto prejudicial o alto custo espacial do código QR.

- ARSALAN et al. (2017), em que uma técnica reversível é utilizada para proteger informações do paciente, usando uma transformada no domínio da frequência para embutir as informações, com processamento do histograma para evitar *underflow* ou *overflow*, e usando programação genética para selecionar os coeficientes.

- LIU et al. (2019), foram usadas transformadas no domínio da frequência da imagem para proteger a autenticidade da imagem, dada por um valor *hash* originado de uma imagem que contém a logomarca do hospital, enquanto os dados para a integridade são detecção, localização e recuperação de adulteração.

- SHEHAB et al. (2018), uma técnica de marca d'água frágil foi proposta, com a imagem original sendo decomposta em blocos de tamanho 4×4, e então aplicando a decomposição de valor simples nos *bits* menos significativos (LSB), que são os últimos bits da representação de um número binário, por exemplo, o número 5, representado por um binário de 4 *bits*, é dado por 0101, com o *bit* mais significativo sendo 0, tendo peso de 2^3, e o bit menos significativo sendo 1, tendo peso de 2^0. Os *bits* adulterados podem ser recuperados, mesmo após um alto índice de adulteração, usando uma transformada no domínio da frequência.

Em todos estes trabalhos, testes analíticos e experimentais mostraram que a qualidade dos sinais recuperados é boa, mesmo após a extração dos dados. O desempenho também foi satisfatório, fazendo com que estes cifradores sejam propícios para comunicação em tempo real, além de garantir autenticidade e integridade.

Outra técnica bastante usada em aplicações na medicina é a cifragem, em especial de imagens. Exemplos de algoritmos que usam cifragem para imagens médicas incluem:

- HAMZA et al. (2020), em que um cifrador probabilístico rápido foi proposto, para proteger molduras de imagem que foram extraídas de endoscopias. Estas molduras são extraídas utilizando técnicas de extração de vídeos, e então, um cifrador de chave simétrica é aplicado, com uma rodada de permutação-difusão.

- THAKUR et al. (2019) e THAKUR et al. (2020), nos quais utilizou-se marca d'água no domínio de frequência para inserir os dados do paciente, e depois a imagem com os dados foi cifrada, usando o mapa logístico 2D. Estes trabalhos buscam superar o compromisso entre velocidade e imperceptibilidade, quando a cifragem é usada em conjunto com a marca d'água.

- KANG et al. (2020), um esquema robusto de marca d'água foi proposto para imagens coloridas, em conjunto com a cifragem a partir do uso de um sistema hiper-caótico. Primeiramente, a imagem original é decomposta de acordo com a transformada *wavelet*, com os componentes de baixa frequência sendo decompostos em blocos, e das características dos blocos, uma matriz é extraída. Além disso, uma sequência é gerada a partir de um mapa hiper-caótico, e uma rodada de permutação-difusão é aplicada em uma logomarca desejada. Por fim, uma operação *xor* é aplicada entre a matriz com as características do bloco e a logomarca cifrada.

- ELHOSENY et al. (2018b), em que primeiro uma chave ótima é escolhida, e depois a imagem é cifrada usando o algoritmo de chave pública ECC. Do ponto de vista de performance, essa escolha é prejudicial, pois o tempo médio de cifragem de uma imagem é em torno de um minuto e meio.

- DAGADU; LI (2018), em que a marca d'água no domínio da frequência é combinado com cifragem caótica. Na primeira parte, os dados do paciente são embutidos na imagem, e depois os pixels da imagem embutida são cifrados com uma arquitetura permutação-difusão, usando uma sequência caótica. No lado do receptor, as informações do paciente são utilizadas para verificar autenticidade e integridade.

- SHANKAR et al. (2018), em que o mapa logístico e o mapa da tenda foram utilizados como subchaves para transmissão de imagens médicas, e essas subchaves são usadas para a cifragem e decifragem. Em CHEN et al. (2019), as imagens são cifradas utilizando difusão hierárquica e cifragem não-sequencial, usando o sistema hipercaótico de Lorenz para gerar as chaves. Como ponto

fraco destas duas propostas, tem-se o uso da cifragem pura, não permitindo garantir a integridade e autenticidade das imagens.

Também levando em conta o estado da arte, neste trabalho é proposto um cifrador para aplicações em imagens médicas, com duas fases: a primeira, em que ocorre a marca d'água, com a inserção dos dados do paciente na imagem, e a segunda, na qual há a cifragem da imagem marcada com as informações textuais, com o uso do cifrador de permutação-difusão simultânea.

Durante a marca d'água, a técnica espacial é usada, em virtude de seu menor consumo de tempo. Para isso, usa-se a estratégia do *bit* menos significativo, uma técnica bastante simples, que funciona substituindo o LSB de pixels selecionados aleatoriamente por um *bit* da mensagem a ser inserida. A simplicidade se correlaciona positivamente com o desempenho. Porém, como ponto negativo, tem-se que a quantidade de *bits* que pode ser inserida é pequena. Contudo, considerando que as imagens possuem dimensões grandes, enquanto o número de *bits* é de apenas alguns milhares, isto não é relevante para a aplicação deste trabalho.

Com isso, o problema da técnica de marca d'água espacial, comparada as técnicas no domínio da frequência, é a baixa imperceptibilidade, ou seja, pode ser fácil para uma parte não autorizada perceber que há informação escondida na imagem. Para compensar este problema, neste trabalho a imagem com os *bits* inseridos é cifrada aplicando o cifrador de permutação-difusão simultânea, em virtude de seu baixo consumo de tempo.

Na decifragem, o caminho inverso é aplicado, e então a mensagem de texto inserida é recuperada. Por fim, em virtude da alta correlação entre pixels adjacentes, a comparação da variância entre os pixels é usada para recuperar a imagem original, nas posições em que *bits* da mensagem de texto foram inseridos.

Testes mostram que a imagem original é recuperada com distorção baixa, o que significa que um profissional de tele-medicina vai chegar à mesma conclusão avaliando a imagem original e a recuperada. A autenticidade é garantida pela mensagem recuperada, pois contém informações sobre o profissional ou sobre a clínica envolvidos, e em virtude da marca d'água frágil, adulterações significam que a informação textual recuperada será diferente da original, assim sendo detectada e garantindo a integridade. O algoritmo proposto também é bastante eficiente, tendo *throughput* superior a outros trabalhos na literatura.

5.2 Descrição

O algoritmo para aplicações na medicina possui as seguintes características, considerando M o número de linhas, N o número de colunas e C o número de canais de cor.

- **Entrada:** imagem original, de tamanho $M \times N \times C$, combinada com a informação textual de tamanho S.
- **Saída:** imagem cifrada, contendo os dado textuais inseridos, de tamanho $M \times N \times C$.

O procedimento utiliza três sequências caóticas, sendo que h_t é usada para inserir as informações textuais, h_p para a permutação e h_d para a difusão. As chaves secretas são dadas por:

- x_{t_0} and p_t, relacionadas a sequência caótica h_t;
- x_{p_0} and p_p, relacionadas a sequência caótica h_p;
- x_{d_0} and p_d, relacionadas a sequência caótica h_d.

Os valores das chaves irão pertencer ao intervalo $(0, 0, 95)$. Para fazer com que o cifrador seja seguro contra os ataques do texto conhecido e do texto escolhido, de forma com que as sequências caóticas sejam dependentes da imagem original, a *hash* SHA-256 também é usada para esta versão do cifrador, sendo aplicada à imagem original, gerando a sequência $H = H(1)H(2)H(3)\cdots H(256)$. As perturbações são dadas por:

$$\lambda_1 = H(1:43), \qquad \lambda_2 = H(44:86),$$
$$\lambda_3 = H(87:129), \qquad \lambda_4 = H(130:172),$$
$$\lambda_5 = H(173:214), \qquad \lambda_6 = H(215:256).$$

Após realizar a perturbações, os valores são convertidos para decimal, e multiplicados por $2,8409 \times 10^{-15}$, de forma a pertencerem ao intervalo $(0, 0, 05)$. As equações são dadas por:

$$\lambda_i = \text{conversaoDecimal}(\lambda_i),$$
$$\lambda_i = 2,8409 \times 10^{-15} \times \lambda_i.$$

As chaves perturbadas são dadas por:

$$x_{t_0} = x_{t_0} + \lambda_1 \qquad p_t = p_t + \lambda_2,$$
$$x_{p_0} = x_{p_0} + \lambda_3 \qquad p_p = p_p + \lambda_4,$$
$$x_{d_0} = x_{d_0} + \lambda_5 \qquad p_d = p_d + \lambda_6.$$

As sequências caóticas são geradas usando a equação (13), que tem como saída x_n, e para cada valor de x_n, a equação (14) é aplicada, que tem como saída o valor h_n.

Durante a marca d'água, um texto que contém informações sobre o hospital, sobre os profissionais de saúde e dados do paciente é inserido na imagem original. Neste passo usa-se a sequência caótica h_t. Após, na cifragem, usa-se um algoritmo de permutação-difusão simultânea. Antes do começo da cifragem em si, ou seja, antes da marca d'água, são resolvidas 200 iterações de cada mapa caótico, e estas iterações são descartadas, para anular o efeito transiente. Na Figura 26, um fluxograma é apresentando, mostrando os passos do algoritmo.

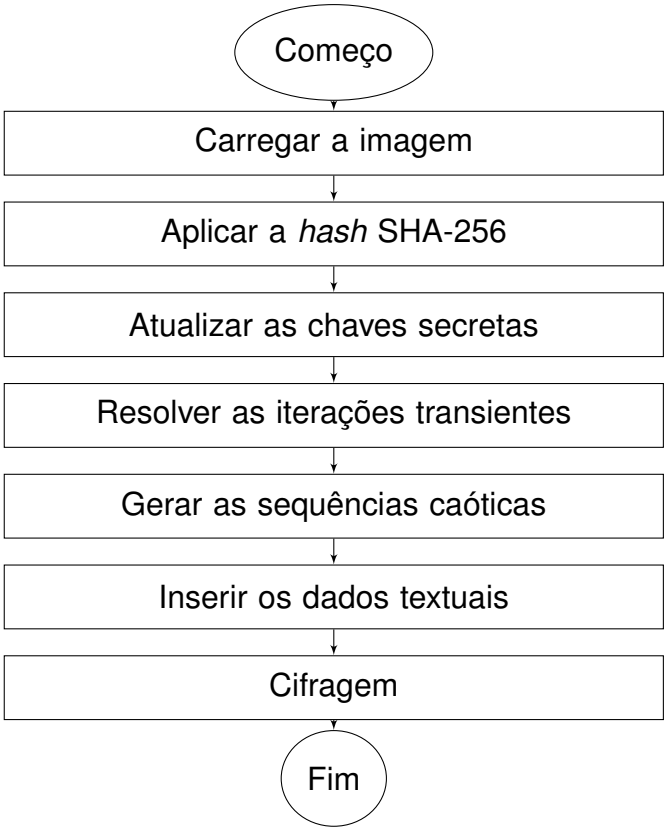


Figura 26: Fluxograma do algoritmo proposto para cifragem de imagens médicas.

5.2.1 Inserção dos dados textuais

A posição em que o *bit* i da mensagem de texto é colocado é dada pelo valor caótico h_t, usando o resultado x_t da equação (13), que é a entrada da função (14), para encontrar h_t. Caso nesta posição um *bit* da mensagem de texto já estiver inserido, repete-se este passo para encontrar outra posição. Para controlar as posições já marcadas, usa-se um vetor chamado $marked$, em que cada posição assume valores $True = 1$ ou $False = 0$. O procedimento de inserção dos dados textuais é dado pelo algoritmo abaixo, e com isso, a imagem marcada é gerada.

for $i \leftarrow 1, 8N$ **do**
  $x_t \leftarrow$ pwlcm(x_t, p_t)
  $h_t \leftarrow 2^{33} x_t \bmod (M \times N \times C)$
  if (marked$(h_t)) = 0$ **then**

marked(h_t) $\leftarrow 1$
image(h_t) $\leftarrow$ embedLSB(text(i), image(h_t))
else
while marked(h_t) $\neq 0$ **do**
$x_t \leftarrow$ pwlcm(x_t, p_t)
$h_t \leftarrow 2^{33} x_t \bmod (M \times N \times C)$
end while
marked(h_t) $\leftarrow 1$
image(h_t) $\leftarrow$ embedLSB(text(i), image(h_t))
end if
end for

Para verificar qual o *bit* menos significativo assumido por um pixel, divide-se seu valor por dois. Caso o resto seja 0, o *bit* menos significativo também é 0, assim como se o resto for 1, o *bit* menos significativo é 1. Na inserção, o *bit* a ser inserido apenas substitui, caso sejam diferentes, o *bit* menos significativo do pixel escolhido aleatoriamente.

5.2.2 Cifragem

Na cifragem, o pixel i da imagem marcada é trocado com outro pixel, dado pelo valor caótico h_p, usando o resultado x_p da equação (13). Então, o valor caótico h_d é encontrado a partir do mesmo processo de h_p, usando os resultados de x_d, para a difusão. Estes dois procedimentos podem ocorrer simultaneamente, ou seja, no mesmo passo dentro de um *loop for*, o que implica que os processos de permutação e difusão são simultâneos, diminuindo a complexidade computacional e aumentando a eficácia do cifrador. O processo de cifragem é dado pelo algoritmo abaixo, e após isso, tem-se a imagem marcada e cifrada.

len $\leftarrow M \times N \times C$
for $i \leftarrow 0, len - 1$ **do**
$x_p \leftarrow$ pwlcm(x_p, p_p)
$h_p \leftarrow 2^{33} x_p \bmod [(M \times N \times C) - i]$
image(i) $\leftrightarrow$ image(h_p) ▷ Permutação
$x_d \leftarrow$ pwlcm(x_d, p_d)
$h_d \leftarrow 2^{33} x_d \bmod (256)$
image(i) $\leftarrow$ image(i) $\oplus h_d$ ▷ Difusão
end for

5.2.3 Decifragem

Durante este passo, a imagem marcada e cifrada é decifrada, aplicando o processo inverso da cifragem. A sequências caóticas h_p e h_d são geradas antes do processo

(a) (b) (c) (d)

Figura 27: a) Cérebro. b) Ligamento cruzado anterior. c) Pescoço. d) Pedra nos rins.

inverso de permutação-difusão, e este processo ocorre na ordem contrária, do lado direito inferior até o lado esquerdo superior da imagem cifrada. Após este passo, a imagem marcada é recuperada.

5.2.4 Remoção dos dados textuais

Esta é a fase final do algoritmo, em que se verifica a autenticidade e a integridade das informações cifradas. A sequência caótica h_t, gerada na fase de inserção dos dados da cifragem, é gerada também pelo receptor, e o *bit* menos significativo de cada um destes pixels é removido, de forma com que o texto original é recuperado. Então, para recuperar o valor original do pixel, a variância local entre o pixel e seus vizinhos superior, inferior, esquerdo e direito é testado para dois casos: quando o *bit* menos significativo do pixel é 0, e quando é 1, e o caso em que a variância local for menor irá determinar o que o *bit* menos significativo do pixel irá assumir na imagem recuperada.

5.3 Resultados

Nesta seção, é realizada a análise dos resultados do cifrador, do ponto de vista de similaridade estrutural entre a imagem original e a recuperada, da segurança do processo de eficiência e da performance computacional. Foram usadas tomografias computadorizadas dadas pela Figura 27, que são do cérebro, do ligamento cruzado anterior, do pescoço e pedra nos rins, e que foram retiradas de MEDPIX (2021).

5.3.1 Similaridade estrutural

Para evitar erros médicos, é preciso que a imagem original e a recuperada sejam semelhantes, e essa semelhança é quantificada por três estatísticas: *Peak Signal-to-Noise Ratio* (PSNR), Erro Quadrático Médio (MSE) e o Indíce de Similaridade Estrutural (SSIM).

Para duas imagens, de tamanho $M \times N \times C$, o MSE é dado pela equação (41), em que C_1 é a imagem original e C_2 é a imagem recuperada, após a remoção das

Imagem	MSE	PSNR	SSIM
Cérebro	0,0272	76,4951dB	0,9999
Ligamento cruzado anterior	0,0569	69,2249dB	0,9997
Pescoço	0,0516	77,4791dB	0,9997
Pedras nos rins	0,0724	74,5636dB	0,9997

Tabela 28: Métricas estatísticas para a semelhança entre a imagem original e a recuperada.

informações textuais. O PSNR é dado pela equação (35), e são desejados baixos valores para o MSE e altos valores para o PSNR, o que indica baixa distorção na imagem recuperada, o que é necessário para garantir a qualidade visual da imagem recuperada.

$$MSE = \frac{1}{M \times N \times C} \sum_{i=1}^{M \times N \times C} (C_1(i) - C_2(i))^2, \tag{41}$$

A similaridade estrutural é medida pelo SSIM, e deseja-se que este valor esteja próximo de 1 para que as imagens sejam semelhantes. A equação (36) é usada para fornecer esta métrica, entre duas imagens C_1 e C_2, em que C_1 é a imagem original e C_2 é a imagem recuperada. Ademais, μ_{C_1} e μ_{C_2} são os valores médios de C_1 e C_2, com $\sigma^2_{C_1}$ e $\sigma^2_{C_2}$ representando as variâncias, $\sigma_{C_1C_2}$ é a covariância de C_1 e C_2, e por padrão, usam-se as constantes $k_1 = 0,01, k_2 = 0,03$ para estabilizar o denominador.

Neste trabalho, o texto inserido tem aproximadamente 4000 *bits*. Na Tabela 28, os três parâmetros são encontrados para as imagens utilizadas neste trabalho, obtendo baixo valores de MSE, altos valores de PSNR e de SSIM, indicando que existe um alto grau de semelhança entre a imagem original e a imagem recuperada, não existindo o risco de diagnósticos médicos diferentes entre as duas imagens, e que o cifrador proposto é seguro do ponto de vista médico.

5.3.2 Análise de segurança

O cifrador precisa ser resistente contra as mais variadas formas de ataque, que são ataques estatísticos, diferencial, de força-bruta e os tradicionais, e a segurança do cifrador é dependente da resistência a todos estes tipos de ataques. A resistência contra os ataques estatísticos é medida pelo coeficiente de Pearson, dado pela equação (25), e pelos coeficientes de correlação, dados pela equação (29). A resistência a ataques diferenciais é dada pelo NPCR e para o UACI, dados pelas Equações (30) e (31), tanto para pequenas mudanças na chave ou na imagem original, enquanto a resistência à ataques de força bruta é dada pelo espaço de chaves. Contra ataques tradicionais, a resistência é dada pela dependência entre as sequências caóticas geradas e a imagem original, e neste trabalho, a *hash* SHA-256 garante esta dependência.

Coeficientes de correlação perto de 0, na Tabela 29, junto com valores do coeficiente de Pearson abaixo do limiar de $293,248$ na Tabela 30, mostram que o cifrador proposto é resistente à ataques estatísticos. Além disso, na Tabela 30, os valores de NPCR e UACI estão nos intervalos desejados tanto quando a mudança é na chave como quando é na imagem, mostrando que o cifrador é resistente a ataques diferenciais.

O cifrador usa três mapas caóticos, cada um com uma condição inicial e um valor de parâmetro como chave, em adição à *hash* de 256 *bits*. Com isso, o espaço de chaves do cifrador é de $(10^{15})^6 = 10^{90}$ multiplicado por 2^{256}, ou seja, de $2^{346} \times 5^{90}$, consideravelmente acima do limiar de 2^{100} (ALVAREZ; LI, 2006), indicando que o cifrador é resistente a um ataque de força bruta.

Imagem	Canal	Vertical	Horizontal	Diagonal
Cérebro	R	-0,00045	-0,0017	-0,0017
	G	-0,00015	-0,0013	-0,00014
	B	-0,0015	-0,0033	-0,0027
Ligamento cruzado anterior	R	-0,00085	0,0019	-0,00042
	G	-0,00085	-0,0035	-0,00014
	B	0,0017	-0,00087	-0,00077
Pescoço	R	-0,00046	0,00058	0,00025
	G	-0,0016	-0,0003	-0,0006
	B	-0,00006	-0,0003	0,00032
Pedra nos rins	R	-0,00042	0,0016	0,0022
	G	-0,0016	-0,00036	-0,0016
	B	-0,00015	-0,0014	-0,00056

Tabela 29: Coeficientes de correlação para a imagem cifrada.

Imagem	Pearson	NPCR-IS/KS	UACI-IS/KS
Cérebro	238	99,59%/99,60%	33,44%/33,45%
Ligamento cruzado anterior	279	99,63%/99,63%	33,46%33,49%
Pescoço	260	99,61%/99,60%	33,47%/33,47%
Pedra nos rins	247	99,59%/99,61%	33,46%/33,49%

Tabela 30: Valores dos coeficientes de Pearson, NPCR e UACI para a imagem cifrada.

5.3.3 Análise de performance

As simulações foram realizadas na mesma máquina que o cifrador de imagens, ou seja, em um computador Acer Aspire 3, com processador Intel Core i5 240-250 GHz, 4GB de RAM e sistema operacional Windows 10, Foram realizadas 50 simulações para cada imagem, e o *throughput* é dado pela equação (18).

A Tabela 31 mostra as métricas para o cifrador proposto, com o uso da permutação-difusão simultânea aumentando consideravelmente o *throughput* do cifrador em relação ao cifrador de imagens de permutação-difusão separada, que já apresentava resultados destacados em relação a outros trabalhos na literatura. Dessa forma, o objetivo de obter desempenho eficiente sem sacrificar imperceptibilidade, com a junção da marca d'água espacial e cifragem, foi atingido.

Imagem	Dimensões	Tempo (ms)	*Throughput* (MB/s)
Cérebro	$630 \times 630 \times 3$	147	8,1
Ligamento Cruzado Anterior	$400 \times 400 \times 3$	50	9,6
Pescoço	$977 \times 986 \times 3$	285	10,14
Pedra nos rins	$700 \times 986 \times 3$	200	10,35

Tabela 31: Tempo de execução e *throughput* para o cifrador proposto.

6 CONCLUSÕES

Neste trabalho, estudou-se a aplicação de modelos matemáticos caóticos na criptografia. Inicialmente, no capítulo 1, foi feita uma apresentação de conceitos sobre a criptografia e a criptoanálise, além de uma revisão bibliográfica desde os primórdios até o contexto atual, em que a cifragem de imagens apresenta um papel de destaque. No capítulo 2, foi feita a abordagem da teoria do caos. Primeiro, conceitos de sistemas dinâmicos são apresentados, e depois, a teoria do caos em si. Da análise da teoria do caos, tem-se que o mapa da tenda é o modelo ideal a ser usado para a geração das sequências caóticas, em virtude de sua robustez e de ser rápido de simular computacionalmente. No capítulo 3, foi apresentada a versão do cifrador de imagens, os testes de segurança desta versão, que apresentou desempenho superior mediante outros trabalhos na literatura. No capítulo 4, o cifrador implementado no capítulo 3 foi melhorado, com a operação de permutação-difusão sendo realizada de forma simultânea. Os testes de segurança mostraram que este cifrador é seguro, e o desempenho foi três vezes superior ao do cifrador proposto no capítulo 3, e quatro vezes superior ao desempenho do AES. No capítulo 5, foi apresentada a versão do cifrador médico, em que textos são inseridos na imagem, antes dela ser cifrada. Estas informações são 100% recuperáveis, e a imagem é recuperada com distorções mínimas, de forma que não exista diferenças notáveis entre a imagem e a recuperada. Testes mostram que o procedimento é seguro, e apresenta desempenho superior em relação ao de outros trabalhos na literatura.

Para trabalhos futuros, o objetivo é melhorar ainda mais o desempenho deste cifrador com o uso de *Graphics Processing Unit* (GPU), e também implementar o cifrador em *hardware*.

REFERÊNCIAS

ABDOUN, N. **Design, implementation and analysis of keyed hash functions based on chaotic maps and neural networks**. 2019. Tese (Doutorado em Ciência da Computação) — Nantes.

AHMAD, J.; HWANG, S. O. Chaos-based diffusion for highly autocorrelated data in encryption algorithms. **Nonlinear Dynamics**, [S.l.], v.82, n.4, p.1839–1850, 2015.

AHMAD, J.; KHAN, M. A.; AHMED, F.; KHAN, J. S. A novel image encryption scheme based on orthogonal matrix, skew tent map, and XOR operation. **Neural Computing and Applications**, [S.l.], v.30, n.12, p.3847–3857, 2018.

AHMED, B.; ABD, S.; HAMIDA, A. A novel image encryption using an integration technique of blocks rotation based on the magic cube and the AES algorithm. **International Journal of Computer Science Issues**, [S.l.], v.9, n.4, p.41–47, 2012.

AL-JABALI, M. F. **Image Encryption System by Generating Chains from the Secret Key**. 2016. Tese (Doutorado em Ciência da Computação) — .

ALANAZI, A. S.; MUNIR, N.; KHAN, M.; ASIF, M.; HUSSAIN, I. Cryptanalysis of Novel Image Encryption Scheme Based on Multiple Chaotic Substitution Boxes. **IEEE Access**, [S.l.], v.9, p.93795–93802, 2021.

ALGHAFIS, A.; FIRDOUSI, F.; KHAN, M.; BATOOL, S. I.; AMIN, M. An efficient image encryption scheme based on chaotic and Deoxyribonucleic acid sequencing. **Mathematics and Computers in Simulation**, [S.l.], v.177, p.441–466, 2020.

ALLIGOOD, K. T.; SAUER, T. D.; YORKE, J. A. **Chaos**. [S.l.]: Springer, 1996.

ALVAREZ, G.; LI, S. Some basic cryptographic requirements for chaos-based cryptosystems. **International Journal of Bifurcation and Chaos**, [S.l.], v.16, n.08, p.2129–2151, 2006.

ALVAREZ, G.; MONTOYA, F.; ROMERA, M.; PASTOR, G. Cryptanalysis of a discrete chaotic cryptosystem using external key. **Physics Letters A**, [S.l.], v.319, n.3-4, p.334–339, 2003.

ARAB, A.; ROSTAMI, M. J.; GHAVAMI, B. An image encryption method based on chaos system and AES algorithm. **The Journal of Supercomputing**, [S.l.], v.75, n.10, p.6663–6682, 2019.

ARSALAN, M.; QURESHI, A. S.; KHAN, A.; RAJARAJAN, M. Protection of medical images and patient related information in healthcare: Using an intelligent and reversible watermarking technique. **Applied Soft Computing**, [S.l.], v.51, p.168–179, 2017.

ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. **Computer networks**, [S.l.], v.54, n.15, p.2787–2805, 2010.

AUMASSON, J.-P. **Serious cryptography**: a practical introduction to modern encryption. [S.l.]: No Starch Press, 2017.

BAPTISTA, M. d. Cryptography with chaos. **Physics Letters A**, [S.l.], v.240, p.50–54, 03 1998.

BARRIO, R.; SERRANO, S. A three-parametric study of the Lorenz model. **Physica D**, [S.l.], v.229, n.1, p.43–51, 2007.

BASSEIN, S. The dynamics of a family of one-dimensional maps. **The American Mathematical Monthly**, [S.l.], v.105, n.2, p.118–130, 1998.

BEHNIA, S.; AKHSHANI, A.; MAHMODI, H.; AKHAVAN, A. A novel algorithm for image encryption based on mixture of chaotic maps. **Chaos, Solitons & Fractals**, [S.l.], v.35, n.2, p.408–419, 2008.

BELAZI, A.; EL-LATIF, A. A. A.; BELGHITH, S. A novel image encryption scheme based on substitution-permutation network and chaos. **Signal Processing**, [S.l.], v.128, p.155–170, 2016.

BEZERRA, J. I. M.; ALMEIDA CAMARGO, V. V. de; MOLTER, A. A new efficient permutation-diffusion encryption algorithm based on a chaotic map. **Chaos, Solitons & Fractals**, [S.l.], v.151, p.111235, 2021.

BILLINGS, L.; BOLLT, E. Probability density functions of some skew tent maps. **Chaos, Solitons & Fractals**, [S.l.], v.12, n.2, p.365–376, 2001.

CAO, C.; SUN, K.; LIU, W. A novel bit-level image encryption algorithm based on 2D-LICM hyperchaotic map. **Signal Processing**, [S.l.], v.143, p.122–133, 2018.

ÇAVUŞOĞLU, Ü.; KAÇAR, S.; PEHLIVAN, I.; ZENGIN, A. Secure image encryption algorithm design using a novel chaos based S-Box. **Chaos, Solitons & Fractals**, [S.l.], v.95, p.92–101, 2017.

ÇAVUŞOĞLU, Ü.; KAÇAR, S.; ZENGIN, A.; PEHLIVAN, I. A novel hybrid encryption algorithm based on chaos and S-AES algorithm. **Nonlinear Dynamics**, [S.l.], v.92, n.4, p.1745–1759, 2018.

CHAI, X. An image encryption algorithm based on bit level Brownian motion and new chaotic systems. **Multimedia Tools and Applications**, [S.l.], v.76, n.1, p.1159–1175, 2017.

CHAI, X.; FU, X.; GAN, Z.; LU, Y.; CHEN, Y. A color image cryptosystem based on dynamic DNA encryption and chaos. **Signal Processing**, [S.l.], v.155, p.44–62, 2019.

CHEN, J.; CHEN, L.; ZHANG, L. Y.; ZHU, Z.-l. Medical image cipher using hierarchical diffusion and non-sequential encryption. **Nonlinear Dynamics**, [S.l.], v.96, n.1, p.301–322, 2019.

CHEN, J.; CHEN, L.; ZHOU, Y. Cryptanalysis of a DNA-based image encryption scheme. **Information Sciences**, [S.l.], v.520, p.130–141, 2020.

CHEN, J.; HAN, F.; QIAN, W.; YAO, Y.-D.; ZHU, Z.-l. Cryptanalysis and improvement in an image encryption scheme using combination of the 1D chaotic map. **Nonlinear Dynamics**, [S.l.], v.93, n.4, p.2399–2413, 2018.

CHEN, J.-x.; ZHU, Z.-l.; FU, C.; YU, H.; ZHANG, Y. Reusing the permutation matrix dynamically for efficient image cryptographic algorithm. **Signal Processing**, [S.l.], v.111, p.294–307, 2015.

CHENG, G.; WANG, C.; XU, C. A novel hyper-chaotic image encryption scheme based on quantum genetic algorithm and compressive sensing. **Multimedia Tools and Applications**, [S.l.], v.79, n.39, p.29243–29263, 2020.

COMPUTER SECURITY DIVISION, I. T. L. **NIST SP 800-22**: Documentation and Software - Random Bit Generation: CSRC.

CUOMO, K. M.; OPPENHEIM, A. V. Circuit implementation of synchronized chaos with applications to communications. **Physical review letters**, [S.l.], v.71, n.1, p.65, 1993.

DAGADU, J. C.; LI, J. Context-based watermarking cum chaotic encryption for medical images in telemedicine applications. **Multimedia Tools and Applications**, [S.l.], v.77, n.18, p.24289–24312, 2018.

DENG, J.; ZHOU, M.; WANG, C.; WANG, S.; XU, C. Image segmentation encryption algorithm with chaotic sequence generation participated by cipher and multi-feedback loops. **Multimedia Tools and Applications**, [S.l.], p.1–20, 2021.

DEVANEY, R. **An introduction to chaotic dynamical systems**. [S.l.]: CRC Press, 2018.

DIAB, H. An efficient chaotic image cryptosystem based on simultaneous permutation and diffusion operations. **IEEE Access**, [S.l.], v.6, p.42227–42244, 2018.

DIFFIE, W.; HELLMAN, M. New Directions in Cryptography. **Information Theory, IEEE Transactions on**, [S.l.], v.22, p.644 – 654, 12 1976.

DOOLEY, J. **History of Cryptography and Cryptanalysis**: Codes, Ciphers, and Their Algorithms. [S.l.]: Springer, 2018.

DOU, Y.; LIU, X.; FAN, H.; LI, M. Cryptanalysis of a DNA and chaos based image encryption algorithm. **Optik**, [S.l.], v.145, p.456–464, 2017.

EASTLAKE, D.; JONES, P. **US secure hash algorithm 1 (SHA1)**. [S.l.]: RFC 3174, September, 2001.

ELHOSENY, M.; RAMÍREZ-GONZÁLEZ, G.; ABU-ELNASR, O. M.; SHAWKAT, S. A.; ARUNKUMAR, N.; FAROUK, A. Secure medical data transmission model for IoT-based healthcare systems. **IEEE Access**, [S.l.], v.6, p.20596–20608, 2018a.

ELHOSENY, M.; SHANKAR, K.; LAKSHMANAPRABU, S.; MASELENO, A.; ARUNKUMAR, N. Hybrid optimization with cryptography encryption for medical image security in Internet of Things. **Neural computing and applications**, [S.l.], p.1–15, 2018b.

ELMANFALOTY, R. A.; ABOU-BAKR, E. An Image Encryption Scheme Using a 1D Chaotic Double Section Skew Tent Map. **Complexity**, [S.l.], v.2020, 2020.

ENAYATIFAR, R.; ABDULLAH, A. H.; ISNIN, I. F.; ALTAMEEM, A.; LEE, M. Image encryption using a synchronous permutation-diffusion technique. **Optics and Lasers in Engineering**, [S.l.], v.90, p.146–154, 2017.

EVANS, D. The internet of things: How the next evolution of the internet is changing everything. **CISCO white paper**, [S.l.], v.1, n.2011, p.1–11, 2011.

FAN, H.; LI, M.; LIU, D.; AN, K. Cryptanalysis of a plaintext-related chaotic RGB image encryption scheme using total plain image characteristics. **Multimedia Tools and Applications**, [S.l.], v.77, n.15, p.20103–20127, 2018.

FARAH, M. B.; FARAH, A.; FARAH, T. An image encryption scheme based on a new hybrid chaotic map and optimized substitution box. **Nonlinear Dynamics**, [S.l.], p.1–24, 2019.

FIEDLER-FERRARA, N.; PRADO, C. P. C. do. **Caos: uma introdução**. [S.l.]: Editora Blucher, 1994.

FRIDRICH, J. Symmetric ciphers based on two-dimensional chaotic maps. **International Journal of Bifurcation and Chaos**, [S.l.], v.8, n.06, p.1259–1284, 1998.

GAO, H.; ZHANG, Y.; LIANG, S.; LI, D. A new chaotic algorithm for image encryption. **Chaos, Solitons & Fractals**, [S.l.], v.29, n.2, p.393–399, 2006.

GUARDEÑO, D. A. **Framework for the analysis and design of encryption strategies based on discrete-time chaotic dynamical systems**. 2009. Tese (Doutorado em Ciência da Computação) — Universidad Politécnica de Madrid.

GUO, J. **Analysis of chaotic systems**. [S.l.]: Citeseer, 2014.

HAMZA, R.; YAN, Z.; MUHAMMAD, K.; BELLAVISTA, P.; TITOUNA, F. A privacy-preserving cryptosystem for IoT E-healthcare. **Information Sciences**, [S.l.], v.527, p.493–510, 2020.

HUA, Z.; JIN, F.; XU, B.; HUANG, H. 2D Logistic-Sine-coupling map for image encryption. **Signal Processing**, [S.l.], v.149, p.148–161, 2018.

HUA, Z.; ZHOU, Y.; PUN, C.-M.; CHEN, C. P. 2D Sine Logistic modulation map for image encryption. **Information Sciences**, [S.l.], v.297, p.80–94, 2015.

HUANG, L.; CAI, S.; XIAO, M.; XIONG, X. A simple chaotic map-based image encryption system using both plaintext related permutation and diffusion. **Entropy**, [S.l.], v.20, n.7, p.535, 2018.

HUANG, L.; CAI, S.; XIONG, X.; XIAO, M. On symmetric color image encryption system with permutation-diffusion simultaneous operation. **Opt Laser Eng**, [S.l.], v.115, p.7–20, 2019.

HUANG, R.; LIAO, X.; DONG, A.; SUN, S. Cryptanalysis and security enhancement for a chaos-based color image encryption algorithm. **Multimed. Tools Appl.**, [S.l.], p.1–27, 2020.

JALALI, M. S.; LANDMAN, A.; GORDON, W. J. Telemedicine, privacy, and information security in the age of COVID-19. **Journal of the American Medical Informatics Association**, [S.l.], v.28, n.3, p.671–672, 2021.

KANG, X.-b.; LIN, G.-f.; CHEN, Y.-j.; ZHAO, F.; ZHANG, E.-h.; JING, C.-n. Robust and secure zero-watermarking algorithm for color images based on majority voting pattern and hyper-chaotic encryption. **Multimedia Tools and Applications**, [S.l.], v.79, n.1, p.1169–1202, 2020.

KAUR, M.; KUMAR, V. Efficient image encryption method based on improved Lorenz chaotic system. **Electronics Letters**, [S.l.], v.54, n.9, p.562–564, 2018.

KHAN, F. A.; AHMED, J.; KHAN, J. S.; AHMAD, J.; KHAN, M. A. A novel image encryption based on Lorenz equation, Gingerbreadman chaotic map and S 8 permutation. **Journal of Intelligent and Fuzzy Systems**, [S.l.], v.33, n.6, p.3753–3765, 2017.

KOT, M. **Elements of Mathematical Ecology**. [S.l.: s.n.], 2001.

KUDRLE, S.; PROULX, M.; CARRIERES, P.; LOPEZ, M. Fingerprinting for solving A/V synchronization issues within broadcast environments. In: SMPTE ANNUAL TECH CONFERENCE & EXPO, 2010, 2010. **Anais...** [S.l.: s.n.], 2010. p.1–18.

LAIPHRAKPAM, D. S.; KHUMANTHEM, M. S. Cryptanalysis of symmetric key image encryption using chaotic Rossler system. **Optik**, [S.l.], v.135, p.200–209, 2017.

LI, C.; LIN, D.; LÜ, J.; HAO, F. Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography. **IEEE MultiMedia**, [S.l.], v.25, n.4, p.46–56, 2018a.

LI, C.; TAN, K.; FENG, B.; LU, J. The Graph Structure of the Generalized Discrete Arnold's Cat Map. **IEEE Transactions on Computers**, [S.l.], 2021.

LI, C.; ZHANG, Y.; XIE, E. Y. When an attacker meets a cipher-image in 2018: A year in review. **Journal of Information Security and Applications**, [S.l.], v.48, p.102361, 2019.

LI, S.-J. **Analyses and new designs of digital chaotic ciphers**. 2003. Tese (Doutorado em Ciência da Computação) — Xi'an Jiaotong University.

LI, Z.; PENG, C.; LI, L.; ZHU, X. A novel plaintext-related image encryption scheme using hyper-chaotic system. **Nonlinear Dynamics**, [S.l.], v.94, n.2, p.1319–1333, 2018b.

LIAO, X.; CHEN, G.; YIN, J. Content-adaptive steganalysis for color images. **Security and Communication Networks**, [S.l.], v.9, n.18, p.5756–5763, 2016.

LIU, H.; WANG, X. Color image encryption using spatial bit-level permutation and high-dimension chaotic system. **Optics Communications**, [S.l.], v.284, n.16-17, p.3895–3903, 2011.

LIU, H.; WANG, X. Image encryption using DNA complementary rule and chaotic maps. **Applied Soft Computing**, [S.l.], v.12, n.5, p.1457–1466, 2012.

LIU, L.; LEI, Y.; WANG, D. A fast chaotic image encryption scheme with simultaneous permutation-diffusion operation. **IEEE access**, [S.l.], v.8, p.27361–27374, 2020.

LIU, L.; ZHANG, Z.; CHEN, R. Cryptanalysis and improvement in a plaintext-related image encryption scheme based on hyper chaos. **IEEE Access**, [S.l.], v.7, p.126450–126463, 2019.

LIU, X.; LOU, J.; FANG, H.; CHEN, Y.; OUYANG, P.; WANG, Y.; ZOU, B.; WANG, L. A novel robust reversible watermarking scheme for protecting authenticity and integrity of medical images. **IEEE Access**, [S.l.], v.7, p.76580–76598, 2019.

LIU, Y.; QIN, Z.; LIAO, X.; WU, J. Cryptanalysis and enhancement of an image encryption scheme based on a 1-D coupled Sine map. **Nonlinear Dynamics**, [S.l.], v.100, n.3, p.2917–2931, 2020.

LIU, Y.; ZHANG, L. Y.; WANG, J.; ZHANG, Y.; WONG, K.-w. Chosen-plaintext attack of an image encryption scheme based on modified permutation–diffusion structure. **Nonlinear Dynamics**, [S.l.], v.84, n.4, p.2241–2250, 2016.

LUO, Y.; ZHOU, R.; LIU, J.; QIU, S.; CAO, Y. An efficient and self-adapting colour-image encryption algorithm based on chaos and interactions among multiple layers. **Multimedia Tools and Applications**, [S.l.], v.77, n.20, p.26191–26217, 2018.

MA, Y.; LI, C.; OU, B. Cryptanalysis of an image block encryption algorithm based on chaotic maps. **Journal of Information Security and Applications**, [S.l.], v.54, p.102566, 2020.

MAY, R. M. Simple mathematical models with very complicated dynamics. **Nature**, [S.l.], v.261, n.5560, p.459–467, 1976.

MEDPIX. **https**://medpix.nlm.nih.gov/home. (Accessed on 08/15/2021).

MOLLAEEFAR, M.; SHARIF, A.; NAZARI, M. A novel encryption scheme for colored image based on high level chaotic maps. **Multimedia Tools and Applications**, [S.l.], v.76, n.1, p.607–629, 2017.

MONTEIRO, L. H. A. **Sistemas dinâmicos**. [S.l.]: Editora Livraria da Física, 2002.

MURILLO-ESCOBAR, M.; CRUZ-HERNÁNDEZ, C.; ABUNDIZ-PÉREZ, F.; LÓPEZ-GUTIÉRREZ, R. M.; DEL CAMPO, O. A. A RGB image encryption algorithm based on total plain image characteristics and chaos. **Signal Processing**, [S.l.], v.109, p.119–131, 2015.

NAUMAN, A.; QADRI, Y. A.; AMJAD, M.; ZIKRIA, Y. B.; AFZAL, M. K.; KIM, S. W. Multimedia Internet of Things: A comprehensive survey. **IEEE Access**, [S.l.], v.8, p.8202–8250, 2020.

NOROUZI, B.; MIRZAKUCHAKI, S. Breaking a novel image encryption scheme based on an improper fractional order chaotic system. **Multimedia Tools and Applications**, [S.l.], v.76, n.2, p.1817–1826, 2017.

OLIVEIRA, C. H. de. **Criptografia com Caos aplicando o Sistema de Chua**. 2011. 93p. Dissertação de Mestrado — Universidade Estadual de Londrina. Centro de Tecnologia e Urbanismo. Programa de Pós-Graduação em Engenharia Elétrica., Londrina.

PAAR, C.; PELZL, J. **Understanding Cryptography**: A Textbook for Students and Practitioners. [S.l.]: Springer, 2010.

PADHI, M.; CHAUDHARI, R. An optimized pipelined architecture of SHA-256 hash function. In: INTERNATIONAL SYMPOSIUM ON EMBEDDED COMPUTING AND SYSTEM DESIGN (ISED), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.1–4.

PAREEK, N. K.; PATIDAR, V.; SUD, K. Discrete chaotic cryptography using external key. **Physics Letters A**, [S.l.], v.309, n.1-2, p.75–82, 2003.

PAREEK, N. K.; PATIDAR, V.; SUD, K. K. Image encryption using chaotic logistic map. **Image and Vision Computing**, [S.l.], v.24, n.9, p.926–934, 2006.

PATIDAR, V.; PAREEK, N.; PUROHIT, G.; SUD, K. Modified substitution–diffusion image cipher using chaotic standard and logistic maps. **Commun Nonlinear Sci**, [S.l.], v.15, n.10, p.2755–2765, 2010.

PATRO, K. A. K.; ACHARYA, B. An efficient colour image encryption scheme based on 1-D chaotic maps. **Journal of Information Security and Applications**, [S.l.], v.46, p.23–41, 2019.

PENG, Y.; HE, S.; SUN, K. A higher dimensional chaotic map with discrete memristor. **AEU-International Journal of Electronics and Communications**, [S.l.], v.129, p.153539, 2021.

PEREIRA, S. R. **O sistema criptográfico de chave pública RSA**. 2008. 220p. Dissertação de Mestrado — Universidade Católica de Santos (UNISANTOS), Santos.

PING, P.; WU, J.; MAO, Y.; XU, F.; FAN, J. Design of image cipher using life-like cellular automata and chaotic map. **Signal Processing**, [S.l.], v.150, p.233–247, 2018.

QIAO, Z. **Nonlinear dynamics, applications to chaos-based encryption**. 2021. Tese (Doutorado em Ciência da Computação) — Ecole Centrale de Nantes.

RHOUMA, R.; SOLAK, E.; BELGHITH, S. Cryptanalysis of a new substitution–diffusion based image cipher. **Commun Nonlinear Sci**, [S.l.], v.15, n.7, p.1887–1892, 2010.

SAFAEI, B.; MONAZZAH, A. M. H.; BAFROEI, M. B.; EJLALI, A. Reliability side-effects in Internet of Things application layer protocols. In: INTERNATIONAL CONFERENCE ON SYSTEM RELIABILITY AND SAFETY (ICSRS), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.207–212.

SANIVARAPU, P. V.; RAJESH, K. N.; REDDY, N. R.; REDDY, N. C. S. Patient data hiding into ECG signal using watermarking in transform domain. **Physical and Engineering Sciences in Medicine**, [S.l.], v.43, n.1, p.213–226, 2020.

SCHNEIER, B. **Applied cryptography**: protocols, algorithms, and source code in C. [S.l.]: john wiley & sons, 2007.

SHANKAR, K.; ELHOSENY, M.; CHELVI, E. D.; LAKSHMANAPRABU, S.; WU, W. An efficient optimal key based chaos function for medical image security. **IEEE Access**, [S.l.], v.6, p.77145–77154, 2018.

SHANNON, C. E. Communication theory of secrecy systems. **Bell System Technical Journal**, [S.l.], v.28, n.4, p.656–715, 1949.

SHEHAB, A.; ELHOSENY, M.; MUHAMMAD, K.; SANGAIAH, A. K.; YANG, P.; HUANG, H.; HOU, G. Secure and robust fragile watermarking scheme for medical images. **IEEE Access**, [S.l.], v.6, p.10269–10278, 2018.

SHIH, F. Y.; ZHONG, X.; CHANG, I.-C.; SATOH, S. An adjustable-purpose image watermarking technique by particle swarm optimization. **Multimedia Tools and Applications**, [S.l.], v.77, n.2, p.1623–1642, 2018.

SOLAK, E.; ÇOKAL, C.; YILDIZ, O. T.; BIYIKOĞLU, T. Cryptanalysis of Fridrich's chaotic image encryption. **International Journal of Bifurcation and Chaos**, [S.l.], v.20, n.05, p.1405–1413, 2010.

STEVENSON, D. **Microprocessor Standards Subcommittee. IEEE standard for binary floating point arithmetic (IEEE/ANSI 754-1985)**. [S.l.]: Technical report, IEEE, 1985.

SUN, F.; LIU, S.; LI, Z.; LÜ, Z. A novel image encryption scheme based on spatial chaos map. **Chaos, Solitons & Fractals**, [S.l.], v.38, n.3, p.631–640, 2008.

SWENSON, C. **Modern cryptanalysis**: techniques for advanced code breaking. [S.l.]: John Wiley & Sons, 2008.

TALHAOUI, M. Z.; WANG, X. A new fractional one dimensional chaotic map and its application in high-speed image encryption. **Information Sciences**, [S.l.], v.550, p.13–26, 2021.

THAKUR, S.; SINGH, A. K.; GHRERA, S. P.; ELHOSENY, M. Multi-layer security of medical data through watermarking and chaotic encryption for tele-health applications. **Multimedia tools and Applications**, [S.l.], v.78, n.3, p.3457–3470, 2019.

THAKUR, S.; SINGH, A. K.; GHRERA, S. P.; MOHAN, A. Chaotic based secure watermarking approach for medical images. **Multimedia Tools and Applications**, [S.l.], v.79, n.7, p.4263–4276, 2020.

TOUGHI, S.; FATHI, M. H.; SEKHAVAT, Y. A. An image encryption scheme based on elliptic curve pseudo random and advanced encryption system. **Signal Processing**, [S.l.], v.141, p.217–227, 2017.

TRIEBEL, H. **Theory of function spaces. Modern Birkhäuser Classics**. [S.l.]: Birkhäuser/Springer Basel AG, Basel, 2010.

USAMA, M.; KHAN, M. K.; ALGHATHBAR, K.; LEE, C. Chaos-based secure satellite imagery cryptosystem. **Computers and Mathematics with Applications**, [S.l.], v.60, n.2, p.326–337, 2010.

WANG, H.; XIAO, D.; CHEN, X.; HUANG, H. Cryptanalysis and enhancements of image encryption using combination of the 1D chaotic map. **Signal processing**, [S.l.], v.144, p.444–452, 2018.

WANG, X.; GAO, S. Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network. **Information Sciences**, [S.l.], v.539, p.195–214, 2020a.

WANG, X.; GAO, S. Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory. **Information Sciences**, [S.l.], v.507, p.16–36, 2020b.

WANG, X.; LI, Y.; JIN, J. A new one-dimensional chaotic system with applications in image encryption. **Chaos, Solitons & Fractals**, [S.l.], v.139, p.110102, 2020.

WANG, X.; LIU, L.; ZHANG, Y. A novel chaotic block image encryption algorithm based on dynamic random growth technique. **Optics and Lasers in Engineering**, [S.l.], v.66, p.10–18, 2015.

WANG, X.; TENG, L.; QIN, X. A novel colour image encryption algorithm based on chaos. **Signal Processing**, [S.l.], v.92, n.4, p.1101–1108, 2012.

WANG, X.-Y.; YANG, L.; LIU, R.; KADIR, A. A chaotic image encryption algorithm based on perceptron model. **Nonlinear Dynamics**, [S.l.], v.62, n.3, p.615–621, 2010.

WANG, X.; YANG, J.; GUAN, N. High-sensitivity image encryption algorithm with random cross diffusion based on dynamically random coupled map lattice model. **Chaos, Solitons & Fractals**, [S.l.], v.143, p.110582, 2021.

WEN, W. Security analysis of a color image encryption scheme based on skew tent map and hyper chaotic system of 6th-order CNN against chosen-plaintext attack. **Multimed. Tools Appl.**, [S.l.], v.75, n.6, p.3553–3560, 2016.

WONG, K.; LEE, L.-p. A Modified Chaotic Cryptographic Method. **Computer Physics Communications**, [S.l.], v.138, 08 2001.

WONG, K. W. A fast chaotic cryptographic scheme with dynamic look-up table. **Physics Letters A**, [S.l.], v.298, p.238–242, 06 2002.

WU, J.; LIAO, X.; YANG, B. Image encryption using 2D Hénon-Sine map and DNA approach. **Signal Processing**, [S.l.], v.153, p.11–23, 2018.

WU, X.; ZHU, B.; HU, Y.; RAN, Y. A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps. **IEEE Access**, [S.l.], v.5, p.6429–6436, 2017.

XIAN, Y.; WANG, X. Fractal sorting matrix and its application on chaotic image encryption. **Information Sciences**, [S.l.], v.547, p.1154–1169, 2021.

XIE, E. Y.; LI, C.; YU, S.; LÜ, J. On the cryptanalysis of Fridrich's chaotic image encryption scheme. **Signal processing**, [S.l.], v.132, p.150–154, 2017.

YE, G.; HUANG, X. An efficient symmetric image encryption algorithm based on an intertwining logistic map. **Neurocomputing**, [S.l.], v.251, p.45–53, 2017.

YE, G.; PAN, C.; HUANG, X.; MEI, Q. An efficient pixel-level chaotic image encryption algorithm. **Nonlinear Dynamics**, [S.l.], v.94, n.1, p.745–756, 2018.

ZENG, J.; WANG, C. A novel hyperchaotic image encryption system based on particle swarm optimization algorithm and cellular automata. **Security and Communication Networks**, [S.l.], v.2021, 2021.

ZHANG, W.; WONG, K.-w.; YU, H.; ZHU, Z.-l. A symmetric color image encryption algorithm using the intrinsic features of bit distributions. **Communications in Nonlinear Science**, [S.l.], v.18, n.3, p.584–600, 2013.

ZHANG, W.; YU, H.; ZHAO, Y.-l.; ZHU, Z.-l. Image encryption based on three-dimensional bit matrix permutation. **Signal Processing**, [S.l.], v.118, p.36–50, 2016.

ZHANG, Y. Test and verification of AES used for image encryption. **3D Research**, [S.l.], v.9, n.1, p.1–27, 2018.

ZHANG, Y.; CHEN, A.; TANG, Y.; DANG, J.; WANG, G. Plaintext-related image encryption algorithm based on perceptron-like network. **Inf. Sci.**, [S.l.], 2020.

ZHANG, Y.; LI, C.; LI, Q.; ZHANG, D.; SHU, S. Breaking a chaotic image encryption algorithm based on perceptron model. **Nonlinear Dynamics**, [S.l.], v.69, n.3, p.1091–1096, 2012.

ZHANG, Y.-Q.; WANG, X.-Y. A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice. **Information Sciences**, [S.l.], v.273, p.329–351, 2014.

ZHANG, Y.-Q.; WANG, X.-Y. A new image encryption algorithm based on non-adjacent coupled map lattices. **Appl. Soft Comput.**, [S.l.], v.26, p.10–20, 2015.

ZHANG, Y.; TANG, Y. A plaintext-related image encryption algorithm based on chaos. **Multimedia Tools and Applications**, [S.l.], v.77, n.6, p.6647–6669, 2018.

ZHANG, Y.; WANG, X. Plaintext related image encryption scheme using chaotic map. **Telkomnika**, [S.l.], v.12, n.1, p.635–643, 2014b.

ZHOU, G.; ZHANG, D.; LIU, Y.; YUAN, Y.; LIU, Q. A novel image encryption algorithm based on chaos and Line map. **Neurocomputing**, [S.l.], v.169, p.150–157, 2015.

ZHOU, L.; TAN, F.; YU, F. A robust synchronization-based chaotic secure communication scheme with double-layered and multiple hybrid networks. **IEEE Systems Journal**, [S.l.], v.14, n.2, p.2508–2519, 2019.

ZHOU, Y.; BAO, L.; CHEN, C. P. A new 1D chaotic system for image encryption. **Signal Processing**, [S.l.], v.97, p.172–182, 2014.

ZHU, C.; SUN, K. Cryptanalyzing and improving a novel color image encryption algorithm using RT-enhanced chaotic tent maps. **IEEE Access**, [S.l.], v.6, p.18759–18770, 2018.

7 ANEXO - PRODUÇÃO CIENTÍFICA

- Artigos publicados em periódicos:
  - BEZERRA, João Inácio Moreira; FRIGHETTO, Daiane Frighetto. MOLTER, Alexandre; RAFIKOV, Marat; Biological control of the chaotic sugarcane borer-parasitoid agroecosystem; **Ecological Modelling**, v. 450, p. 109564, 2021.
  - BEZERRA, João Inácio Moreira; DE ALMEIDA CAMARGO, Vinícius Valduga; MOLTER, Alexandre. A new efficient permutation-diffusion encryption algorithm based on a chaotic map. **Chaos, Solitons and Fractals**, v. 151, p. 111235, 2021.
- Artigos publicados em anais de eventos:
  - BEZERRA, João Inácio Moreira; MOLTER, Alexandre; Implementação de um programa computacional para o diagrama de bifurcações de sistemas caóticos. **Anais do ERMAC-SC 2019**.
  - BEZERRA, João Inácio Moreira; MOLTER, Alexandre; FERNANDEZ, Leslie Darien Perez; Implementação de um programa computacional para gerar o diagrama de bifurcações de sistemas caóticos. **Anais do ENMC 2020**.
  - FRIGHETTO, Daiane Frighetto; BEZERRA, João Inácio Moreira; MOLTER, Alexandre; RAFIKOV, Marat; Análise da dinâmica populacional entre a broca da cana-de-açúcar e seus parasitoides do estágio do ovo e larval com variação sazonal. Aceito no **CNMAC 2021**.
- Resumos expandidos publicados em eventos:
  - BEZERRA, João Inácio Moreira; CAMARGO, Vinícius Valduga de Almeida; MOLTER, Alexandre; USO DO MAPA CAÓTICO PWLCM PARA A CIFRAGEM DE IMAGENS. ENPOS UFPEL, 2021.

- Pôster:
  - BEZERRA, João Inácio Moreira; CAMARGO, Vinícius Valduga de Almeida; MOLTER, Alexandre; EFFICIENT AND SECURE IMAGE ENCRYPTION ALGORITHM BASED ON CHAOTIC MAPS. **DVPSA 2020**.
  - BEZERRA, João Inácio Moreira; CAMARGO, Vinícius Valduga de Almeida; MOLTER, Alexandre; HIGH PERFORMANCE MEDICAL IMAGE COMMUNICATION USING SPATIAL WATERMARKING AND SIMULTANEOUS PERMUTATION-DIFFUSION CHAOTIC ENCRYPTION. **DVPSA 2021**.
- Registro de programa:
  - BEZERRA, João Inácio Moreira; MOLTER, Alexandre; CAMARGO, Vinícius Valduga de Almeida; Cifrador com permutação-difusão baseados na Teoria do Caos. Processo: BR 51 2021 000133-8. Registro em: 04 fev. 2021.
  - BEZERRA, João Inácio Moreira; MOLTER, Alexandre; CAMARGO, Vinícius Valduga de Almeida; Cifrador caótico de imagens com permutação-difusão simultânea. Processo: BR 51 2021 002843-0. Registro em: 07 dez. 2021.

Um novo algoritmo de criptografia de alto desempenho baseado em modelos caóticos – João Inácio Moreira Bezerra

UNIVERSIDADE FEDERAL DE PELOTAS
Instituto de Física e Matemática
Programa de Pós-Graduação em Modelagem Matemática

Dissertação

Um novo algoritmo de criptografia de alto desempenho baseado em modelos caóticos

JOÃO INÁCIO MOREIRA BEZERRA

Pelotas, 2022