



Programa de Pós-Graduação em Engenharia Elétrica - PPGEL

Associação ampla UFSJ / CEFET-MG

Criptografia de Imagens Baseada em Extensões Intervalares Naturais e no Erro de Precisão Finita

Lucas Giovani Nardo

Orientador: Erivelton Geraldo Nepomuceno

Coorientador: Janier Arias Garcia

São João del-Rei, 9 de dezembro de 2019.





Programa de Pós-Graduação em Engenharia Elétrica - PPGEL

Associação ampla UFSJ / CEFET-MG

Criptografia de Imagens Baseada em Extensões Intervalares Naturais e no Erro de Precisão Finita

---

Lucas Giovani Nardo

Dissertação apresentada à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica, associação ampla entre a Universidade Federal de São João del-Rei e o Centro Federal de Educação Tecnológica de Minas Gerais, como parte dos requisitos necessários à obtenção do grau de Mestre em Engenharia Elétrica.

Área de Concentração: Modelagem e Controle de Sistemas

Linha de Pesquisa: Análise e Modelagem de Sistemas

Grupo de Pesquisa: GCOM - Grupo de Controle e Modelagem

Orientador: Erivelton Geraldo Nepomuceno

Coorientador: Janier Arias Garcia

São João del-Rei, 9 de dezembro de 2019.

Ficha catalográfica elaborada pela Divisão de Biblioteca (DIBIB)
e Núcleo de Tecnologia da Informação (NTINF) da UFSJ,
com os dados fornecidos pelo(a) autor(a)

Nardo, Lucas Giovani .
N224c Criptografia de Imagens Baseada em Extensões Intervalares Naturais e no Erro de Precisão Finita / Lucas Giovani Nardo ; orientador Erivelton Geraldo Nepomuceno; coorientador Janier Arias Garcia. -- São João del-Rei, 2019.
65 p.

Dissertação (Mestrado - Engenharia Elétrica) -- Universidade Federal de São João del-Rei, 2019.

1. Encriptação de imagem. 2. Extensão intervalar natural. 3. Limite inferior do erro. 4. Circuito de Chua. 5. Mapa logístico. I. Nepomuceno, Erivelton Geraldo, orient. II. Garcia, Janier Arias, co orient. III. Título.

Dedico este trabalho aos meus pais, Claudio e Ermelinda, à minha irmã, Rita, e à minha noiva, Vic, os quais sempre me ajudaram a buscar o meu ponto de equilíbrio.

AGRADECIMENTOS

A Deus por ter me guiado e ajudado a completar mais uma etapa da minha vida.

Agradecimentos especiais aos meus pais, Claudio e Ermelinda, por terem me dado os mais valiosos conselhos, ensinamentos, suporte que, no fim, moldaram meu caráter e fizeram o homem que sou hoje.

Agradeço à minha irmã, Rita, por ter me aturado, desde pequena, mas nunca deixou de me amar.

Aos meus avós, Maria Cecília, Mario, Maria Aparecida e Paulino, pelo apoio e amor incondicional.

À minha noiva, linda e mais que inteligente, Vic. Obrigado por ficar sempre ao meu lado.

Obrigado, Prof. Erivelton, pelo reconhecimento e orientação ao longo dos anos.

Agradeço ao Prof. Janier pela confiança e apoio dado durante o processo de elaboração deste trabalho.

Aos, companheiros de república, Vinícius e Wallace.

Aos colegas do GCOM, pelo suporte.

Gostaria de agradecer, também, à Universidade Federal de São João del-Rei e às agências financiadoras CNPq e CAPES.

"Uma máquina pode fazer o trabalho de cinquenta homens comuns. Nenhuma máquina pode fazer o trabalho de um homem extraordinário."

Elbert G. Hubbard

SUMÁRIO

Resumo **xiii**

Abstract **xv**

Lista de Figuras **xviii**

Lista de Tabelas **xix**

Lista de Símbolos **xxi**

Lista de Abreviações **xxv**

1 Introdução **1**

1.1 Objetivos 4

1.2 Contribuições da Dissertação 5

1.3 Organização do Trabalho 6

2 Conceitos Preliminares **9**

2.1 Sistemas Dinâmicos Não-Lineares 9

2.1.1 Sistemas Caóticos 9

2.2 Representações Numéricas e Padrão IEEE de Ponto Flutuante 12

2.2.1 Padrão IEEE 754 de Ponto Flutuante 14

2.3 Limite Inferior do Erro 16

2.4 Criptografia 17

3 Descrição das Cifras e Métodos **21**

3.1 Algoritmo I - Sistema Contínuo 21

3.1.1 Encriptação de Imagens 21

3.1.2 Decriptação de Imagens 23

3.1.3 Definição dos Parâmetros e Imagens Utilizados 25
3.2 Algoritmo II - Sistema Discreto 25
3.2.1 Encriptação de Imagens 26
3.2.2 Decriptação de Imagens 28
3.2.3 Definição dos Parâmetros e Imagens Utilizados 29
3.3 Tabela Comparativa e Testes Realizados 29

4 Análise de Desempenho **31**
4.1 NIST SP 800-22 32
4.2 Espaço Amostral da Chave 34
4.3 Análise da Sensibilidade a Pequenas Mudanças à Chave 34
4.4 Análise da Correlação dos *Pixels* Adjacentes na Imagem 37
4.5 Entropia da Informação 40
4.6 Análise de Histograma 42
4.7 Análise Diferencial 44
4.8 Tempo e Complexidade Computacional 45
4.9 Resistência à Perda de Informação 47
4.10 Resistência a Ataques Via Imagens Simples Conhecidas/Escolhidas 48
4.11 Resistência ao Ruído 49

5 Conclusões **53**
5.1 Considerações Finais 53
5.2 Perspectivas Futuras 54

Referências Bibliográficas **55**

A Rotinas Computacionais **65**

RESUMO

É inerente à natureza humana manter o controle sobre o acesso à informação. Por esse motivo, a busca por um processo de criptografia eficiente e seguro é de suma relevância. A criptografia de imagens é um tópico muito importante nessa área, uma vez que as imagens podem mostrar diversas informações, como os objetos em uma cena, a aparência de uma pessoa, o local, as especificações técnicas de projetos, dentre outras. Vários artigos usam sistemas caóticos para criptografar imagens, no entanto, recentemente, pesquisas têm mostrado que alguns algoritmos de criptografia de imagens são facilmente violados, e, até mesmo, não passam em testes estatísticos quando são esperados valores rigorosos. Uma justificativa para tal problema é a degradação de sistemas caóticos devido à precisão numérica finita dos computadores digitais. Com relação à essa limitação, esta dissertação formula dois processos de codificação de imagens, explorando a precisão numérica finita presente em computadores; bem como utilizando extensões intervalares naturais de sistemas caóticos. Tais extensões intervalares são matematicamente equivalentes, porém são distintas do ponto de vista computacional. Dessa maneira, esse aspecto é utilizado para produzir uma sequência criptográfica baseada no limite inferior do erro, o qual apresenta propriedades pseudo-aleatórias satisfatórias. Os autores adotaram, ora o sistema contínuo do circuito de Chua como sistema caótico, ora um sistema discreto com múltiplas perturbações no mapa logístico. Como resultado, as sequências criptográficas geradas passaram nos testes estatísticos, os quais elas foram submetidas. Por fim, várias imagens de referência foram efetivamente criptografadas, mostrando-se seguras contra vários ataques.

Palavras-chave: Encriptação de imagem; Extensão intervalar natural; Limite inferior do erro; Circuito de Chua; Mapa logístico; Caos; Computação Aritmética.

ABSTRACT

The control of information access is inherent to human nature. For this reason, the investigation of efficient and secure encryption process is relevant. Image encryption is an important topic in this area, since pictures might show a large amount of information, such as objects in a scene, a person's face, a place, technical specifications of projects, among others. Many articles have used chaotic systems to encrypt images. However, it has been shown that some algorithms of image encryption are easy to break, and even do not pass statistical tests when stringent values are expected. One reason for this issue is due to the degradation of chaotic systems overdue to the finite numeric precision of digital computers. Regarding this limitation, this dissertation formulates two processes for image encoding, exploiting the finite precision in computer, as well as utilizing natural interval extensions of chaotic systems. Such interval extensions are mathematically equivalent, but are computationally distinct. Hence, this aspect is used to produce a cryptographic sequence based on the lower bound error, which presents satisfactory pseudo-random properties. The authors have adopted the Chua's circuit as a continuous chaotic system, and the logistic map as discrete chaotic system with multiple perturbations. Results show that the generated cryptographic sequences passed the statistical tests, which they underwent. Finally, several benchmark images have been effectively encrypted and the algorithms have shown secure against several attacks.

Keywords: Image encryption; Natural interval extensions; Lower bound error; Chua's circuit; Logistic map; Chaos; Computer Arithmetic.

LISTA DE FIGURAS

1.1 Degradação do caos no mapa logístico. 3
1.2 Simulação de duas extensões intervalares naturais para quatro diferentes sistemas. 5

2.1 Regime caótico no mapa logístico. 11
2.2 Circuito de Chua e a curva linear por partes do diodo de Chua. 12
2.3 Resultados das simulações do circuito de Chua. 13
2.4 Formato *single* via padrão IEEE 754-2008. 14
2.5 Processo geral de criptografia. 17
2.6 Criptografia simétrica. 18
2.7 Criptografia assimétrica. 19

3.1 Processo de encriptação para o algoritmo I [1]. 23
3.2 Processo de decriptação para o algoritmo I [1]. 24
3.3 Circuito de Chua com os valores nominais dos componentes e as condições iniciais do sistema, como também os valores das inclinações e pontos de quebra da curva linear por partes. 25
3.4 Processo de encriptação usando o limite inferior do erro e múltiplas perturbações [2]. 28
3.5 Processo de decriptação usando o limite inferior do erro e múltiplas perturbações [2]. 29

4.1 Sequências obtidas através da operação do limite inferior do erro e séries temporais após o processo de normalização. 31
4.2 Representação de um processo criptográfico baseado na operação bit-XOR. . . 32
4.3 Resultados obtidos, com a imagem de Lena via decriptação, usando várias chaves secretas. 36
4.4 Distribuição de correlação de dois *pixels* adjacentes para as imagens originais, usando o criptossistema proposto em [1]. 38

4.5 Distribuição de correlação de dois *pixels* adjacentes para as imagens encriptadas, usando o criptossistema proposto em [1]. 39
4.6 Distribuição de correlação de dois *pixels* adjacentes para as imagens de Lena, Babuíno e Barbara, junto com as respectivas imagens encriptadas, quando usado o criptossistema [2]. 40
4.7 Histograma para sete imagens, antes e depois da realização da encriptação, usando a cifra [1]. 43
4.8 Distribuição de frequência de cores para as imagens de Lena, Babuíno e Barbara, antes e depois da realização da encriptação, usando a cifra [2]. 44
4.9 Perda de informação nas imagens cifradas e as respectivas imagens descriptografadas para o algoritmo I. 48
4.10 Perda de informação nas imagens cifradas e as respectivas imagens descriptografadas para o algoritmo II. 49
4.11 Imagens encriptadas/descriptografadas com ruído adicionado. 51

LISTA DE TABELAS

2.1 Diferentes padrões de representação numérica. 14

3.1 Fatores F_o obtidos a partir das imagens de referência. 26
3.2 Fatores c_i obtidos a partir das seguintes imagens e de acordo com a Eq. (3.12). . 29
3.3 Tabela comparativa dos algoritmos propostos nesta dissertação. 30

4.1 Resultados do $P - valor$ para quinze testes usando os algoritmos I e II. 33
4.2 Resultados da análise da sensibilidade no processo de encriptação e decriptação para diversas imagens de referência. 35
4.3 Resultados da análise de sensibilidade da chave para três diferentes imagens. . 36
4.4 Teste de coeficientes de correlação para as sete imagens de referência usadas na cifra [1]. 37
4.5 Valores de coeficiente de correlação para as imagens originais e as cifradas usando a cifra indicada em [2]. 38
4.6 Comparação dos coeficientes de correlação da imagem cifrada de Lena. 41
4.7 Entropia de informação para diferentes imagens. 41
4.8 Entropia de informação da imagem encriptada de Lena para diferentes abordagens. A entropia de informação ideal é 8. 42
4.9 Variância dos histogramas para as respectivas imagens e cifras. 44
4.10 Resultados dos valores de NPCR e UACI para o algoritmo proposto em [1]. . . 46
4.11 Resultados quantitativos para análise de resistência a ataques diferenciais, usando o algoritmo descrito em [2]. 46
4.12 Resultados obtidos com as métricas NPCR e UACI para a imagem de Lena e as suas comparações com outros valores encontrados na literatura. 46
4.13 Complexidade computacional. 47
4.14 Tempo de encriptação para diferentes tamanhos, usando a imagem de Lena. . . 47

LISTA DE SÍMBOLOS

G	Algoritmo;
$vec2mat$	Algoritmo de conversão de vetor para matriz;
$reshape$	Algoritmo para remodelar uma matriz;
$Uint8$	Algoritmo que converte um número real em um número inteiro positivo de 8 *bits*;
$Uint16$	Algoritmo que converte um número real em um número inteiro positivo de 16 *bits*;
$Uint32$	Algoritmo que converte um número real em um número inteiro positivo de 32 *bits*;
$sort(.)$	Algoritmo que reorganiza os elementos de uma matriz ou vetor em ordem crescente;
C_i	Capacitância do capacitor i;
Pu_K	Chave pública;
Pv_K	Chave privada;
K	Chave secreta;
ρ	Coeficiente de correlação dos *pixels* adjacentes de uma imagem;
Lg	Comprimento de uma imagem;
m_b	Comprimento em bits de *substrings* testadas no NIST;
X_0	Condição inicial do mapa logístico para a extensão intervalar de variável X;
Y_0	Condição inicial do mapa logístico para a extensão intervalar de variável Y;
m_i	Condição inicial i ou parâmetro de pertubação i para uma das cifras propostas;
$\mathbb{R}$	Conjunto dos números reais;
Q_f	Conjunto finito Q_f;
H	Coordenada do eixo horizontal de um pixel em determinada imagem;
V	Coordenada do eixo vertical de um pixel em determinada imagem;
i_L	Corrente que percorre o indutor L;
$i_R(V_{C1})$	Corrente que percorre o diodo de Chua;
$\oplus$	Criptograma bit-XOR;
$\dot{x}$	Derivada de uma função;
σ	Desvio padrão;

$Diff$	Diferença;
Nr	Diodo de Chua;
$/$	Divisão;
$H(\cdot)$	Entropia de Shannon;
S_f	Espaço finito S_f;
$E[\cdot]$	Esperança matemática;
x_n	Estado do sistema no instante n;
st	Estados específicos que são estudados no testes de ”Excursões Aleatórias"do NIST;
ϵ	Erro relativo;
Ex	Expoente;
F_o	Fator obtido da imagem a ser codificada;
Fq	Frequência de determinada intensidade de cor no histograma;
D	Função de decriptação;
E	Função de encriptação;
$f(\cdot)$	Função que representa a relação de um certo elemento;
$sign(\cdot)$	Função sinal;
$F_i(X)$	I-ésima extensão intervalar natural;
$=$	Igual a;
I_c	Imagem encriptada;
I_p	Imagem simples;
$G_{a,b}$	Inclinações da curva característica do diodo de Chua;
L	Indutância do indutor;
∞	Infinito;
$UACI(\%)$	Intensidade de mudança média unificada (*Unified Averaged Changed Intensity*);
N	Iterações;
W	Largura de uma imagem;
δ	Limite inferior do erro;
$\underline{X}$	Limite inferior do intervalo X;
$\overline{X}$	Limite superior do intervalo X;
log_{10}	Logaritmo na base 10;
log_2	Logaritmo na base 2;

$>$	Maior que;
$\geq$	Maior ou igual a;
$+$	Mais;
$\pm$	Mais ou menos;
Mn	Mantissa;
M	Matriz de condições iniciais e parâmetros para perturbar uma das cifras propostas;
μ	Média;
$\leq$	Menor ou igual a;
$\bigtriangledown\Psi_m^2$	Medida de acuracidade entre os valores observados e os esperados;
$<$	Menor que;
$-$	Menos;
m	Mensagem;
c	Mensagem ilegível;
$\times$	Multiplicação;
NaN	Não é um Número (*Not a Number*);
b_n	N-ésimo *bit* de uma determinada representação numérica no computador;
G_L	Nível de cinza;
α	Nível de significância;
$O(\dots)$	Notação O-grande;
$NPCR(\%)$	Número de alterações na taxa de *pixels* (*Number of Changing Pixel Rate*);
x	Número ordinário;
$\lvert\cdot\rvert$	Operador de valor absoluto;
mod	Operador de módulo;
$\otimes$	Operador de multiplicação realizado em um computador;
r	Parâmetro de controle do mapa logístico;
η	Parâmetro ordinário;
c_i	Parâmetro oriundo de uma imagem, a fim de perturbar uma das cifras propostas;
$\in$	Pertence a um conjunto;
B_p	Pontos de quebra da curva característica do diodo de Chua;
p	Precisão (comumente indicado em *bits*);
$round(\cdot)$	Procedimento de arredondamento realizado em um computador;

$P - valor$	Probabilidade de significância;
$P(sym_i)$	Probabilidade do símbolo sym;
$\hat{x}_{i,n}$	Pseudo-órbita i no n-ésimo estado do sistema;
I_X	Representação de um intervalo com variável X;
R	Resistência do resistor;
tr	Resposta transitória;
q	Saída;
s	Semente;
S	Sequência criptográfica;
Sn	Sequência criptográfica normalizada;
A	Série de *pixels* em uma determinada posição;
B	Série dos *pixels* adjacentes a A;
B_m	Série finita de uns e zeros definida em uma biblioteca de modelos de padrões não periódicos contidos nos testes estatíticos do NIST;
sym	Símbolo;
$\sum$	Somatório;
t	Tempo;
t_o	Tempo inicial;
V_{Ci}	Tensão do capacitor i;
V'_{C1}	Tensão do capacitor 1 somado com o fator de imagem F_o;
$\mathcal{N}^*_\alpha$	Valor crítico de NPCR para o nível de significância α;
$\mathcal{U}^{*-+}_\alpha$	Valor crítico inferior e/ou superior de UACI para o nível de significância α;
$\min$	Valor mínimo;
$\max$	Valor máximo;
$Var(h)$	Variância;
k	Variável igual a $1{,}2 \ldots 6$ que controla quando as perturbações ocorrem em uma das cifras propostas;
X	Variável ordinária ou estado de um determinado sistema;
Y	Variável ordinária ou estado de um determinado sistema;
h	Vetor dos valores do histograma;

LISTA DE ABREVIAÇÕES

3DES	Padrão Triplo de Criptografia de Dados (*Triple Data Encryption Standard*);
AES	Padrão Avançado de Criptografia (*Advanced Encryption Standard*);
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior;
CEFET-MG	Centro Federal de Educação Tecnológica de Minas Gerais;
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico;
DES	Padrão de Criptografia de Dados (*Data Encryption Standard*);
DFT	Transformada Discreta de Fourier (*Discrete Fourier Transform*);
FPGA	Arranjo de Portas Programáveis em Campo (*Field Programmable Gate Array*);
GCOM	Grupo de Controle e Modelagem;
IBM	*International Business Machines*;
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos (*Institute of Electrical and Electronics Engineers*);
NIST	Instituto Nacional de Padrões e Tecnologia (*National Institute of Standards and Technology*);
NPCR	Número de Alterações na Taxa de *Pixels* (*Number of Changing Pixel Rate*);
PPGEL	Programa de Pós-Graduação em Engenharia Elétrica;
PRNG	Gerador de Números Pseudo-aleatórios (*Pseudo-Random Number Generator*);
RNG	Gerador de Números Aleatórios (*Random Number Generator*);
RSA	Rivest-Shamir-Adleman;
UACI	Intensidade de Mudança Média Unificada (*Unified Averaged Changed Intensity*);
UFSJ	Universidade Federal de São João del-Rei;
VHDL	Linguagem de Descrição de Hardware VHSIC (*VHSIC Hardware Description Language*).

CAPÍTULO 1

INTRODUÇÃO

Tem sido parte integrante da natureza humana manter o controle sobre o acesso à informação. Todavia, ao longo das últimas décadas, houve um aumento exponencial no tráfico de dados, em que milhares de informações e arquivos são compartilhados diariamente, o que gerou um maior desafio no controle e segurança desse acesso à informação. Dessa forma, a criptografia recebeu demasiada atenção [3], tal que países como os Estados Unidos e China, investem em criptossistemas seguros [4].

No âmbito da segurança da informação, há a importância, também, da confidencialidade de imagens[5–8]. De acordo com Wu et al. [9], a exposição não planejada de fotos particulares e governamentais acentua a importância da segurança das imagens, as quais podem estar relacionadas a objetos, pessoas, especificações técnicas de projetos, entre outros [10].

Métodos criptográficos mais robustos e eficazes têm surgido das mais diversas áreas da ciência, como demonstrado pelo método congruente linear [11], pelo uso de números irracionais [12], pelos métodos tradicionais, como demonstrado pelo 3DES (Padrão Triplo de Criptografia de Dados, do inglês *Triple Data Encryption Standard*), RSA (*Rivest-Shamir-Adleman*), Twofish e AES (Padrão Avançado de Criptografia, do inglês *Advanced Encryption Standard*), e também, através dos sistemas caóticos [13]. Com exceção dos sistemas caóticos, tais algoritmos são capazes de criptografar informações textuais, embora não sejam apropriados em casos de imagens, devido às características presentes nas mesmas, como, por exemplo, a forte correlação entre os *pixels* adjacentes [7, 10, 14–20]. Por tal razão, algoritmos baseados em sistemas caóticos têm sido empregados nessas situações.

Os sistemas caóticos apresentam propriedades interessantes, como a sua transitividade, a alta densidade do conjunto dos pontos periódicos e sua sensibilidade às condições iniciais [21, 22]. Portanto, esses sistemas podem gerar sequências pseudo-aleatórias, que podem ser usadas na criptografia. De fato, Herring e Palmore [23] já disseram que geradores de números pseudo-aleatórios (PRNG, do inglês *Pseudo-Random Number Generators*) são exemplos de sistemas dinâmicos caóticos determinísticos.

Na literatura, Matthews [24] foi o pioneiro a apresentar um algoritmo/cifra de encriptação baseado em caos. Depois disso, muitos trabalhos [7, 14–20, 25, 26] propuseram outros métodos criptográficos a partir de diferentes sistemas caóticos. Por exemplo, Li et al. [26] projetaram um algoritmo de criptografia de imagens usando o mapa tenda. Da mesma forma, Wang et al. [14] introduziram um novo esquema usando o mapa logístico, como o sistema caótico e várias outras operações, tais como as aplicações de perturbação, o embaralhamento e substituição de *pixels*, a fim de garantir o desempenho criptográfico necessário. Existem, também, métodos que combinam sistemas caóticos, como é o caso do sistema logístico-tenda, logístico-seno e tenda-seno propostos por Chai [7] e, ainda, aqueles que empregam, em sua parte, sistemas caóticos e hipercaóticos, como o caso do sistema de Lorenz e do sistema hipercaótico de Chen, usados em [27]. Em outra perspectiva, Bao e Zhou [28] criaram um algoritmo que, ao criptografar uma imagem, ao invés de convertê-la em imagens semelhantes a ruídos, gera imagens visualmente significativas. No entanto, acredita-se que várias imagens de referência são necessárias no banco de dados para que, de fato, os ataques a esse tipo de método sejam evitados.

Enquanto a maioria dos trabalhos sobre cifras, baseados no caos, têm sido bem sucedidos e bem recebidos na literatura, alguns estudos questionam a eficácia de tais métodos. Recentemente, Özkaynak [29] fez vários estudos de caso indicando que algumas metodologias são fáceis de serem violadas, e mostrou, também, uma série de passos que os métodos criptográficos devem seguir para serem considerados seguros. Wu et al. [30] relataram que vários artigos, bem aceitos na comunidade acadêmica, não passam nos testes estatísticos NPCR (Número de Alterações na Taxa de *Pixels*, do inglês *Number of Changing Pixel Rate*) e UACI (Intensidade de Mudança Média Unificada, do inglês *Unified Averaged Changed Intensity*) quando rigorosos valores esperados são aplicados; portanto, tais métodos são vulneráveis a ataques diferenciais.

Além disso, um grande desafio a ser enfrentado na aplicação de sistemas caóticos, é que certos sistemas mostram a degradação de suas propriedades caóticas, devido ao uso da precisão finita em computadores digitais, como relatado por Li et al. [31]. Como exemplo, a Fig. 1.1 exibe a série temporal do mapa logístico [32] para seis parâmetros r distintos. Como observado, usando a aritmética de ponto fixo com oito *bits* disponíveis para representar a parte fracionária, após o transitório, há a degradação do caos, originando, então, sequências periódicas. Ademais, há outras ocorrências em que é encontrada a degradação de caos com diferentes precisões numéricas, para além dos exemplos que foram mostrados na Fig. 1.1, como é o caso de [33].

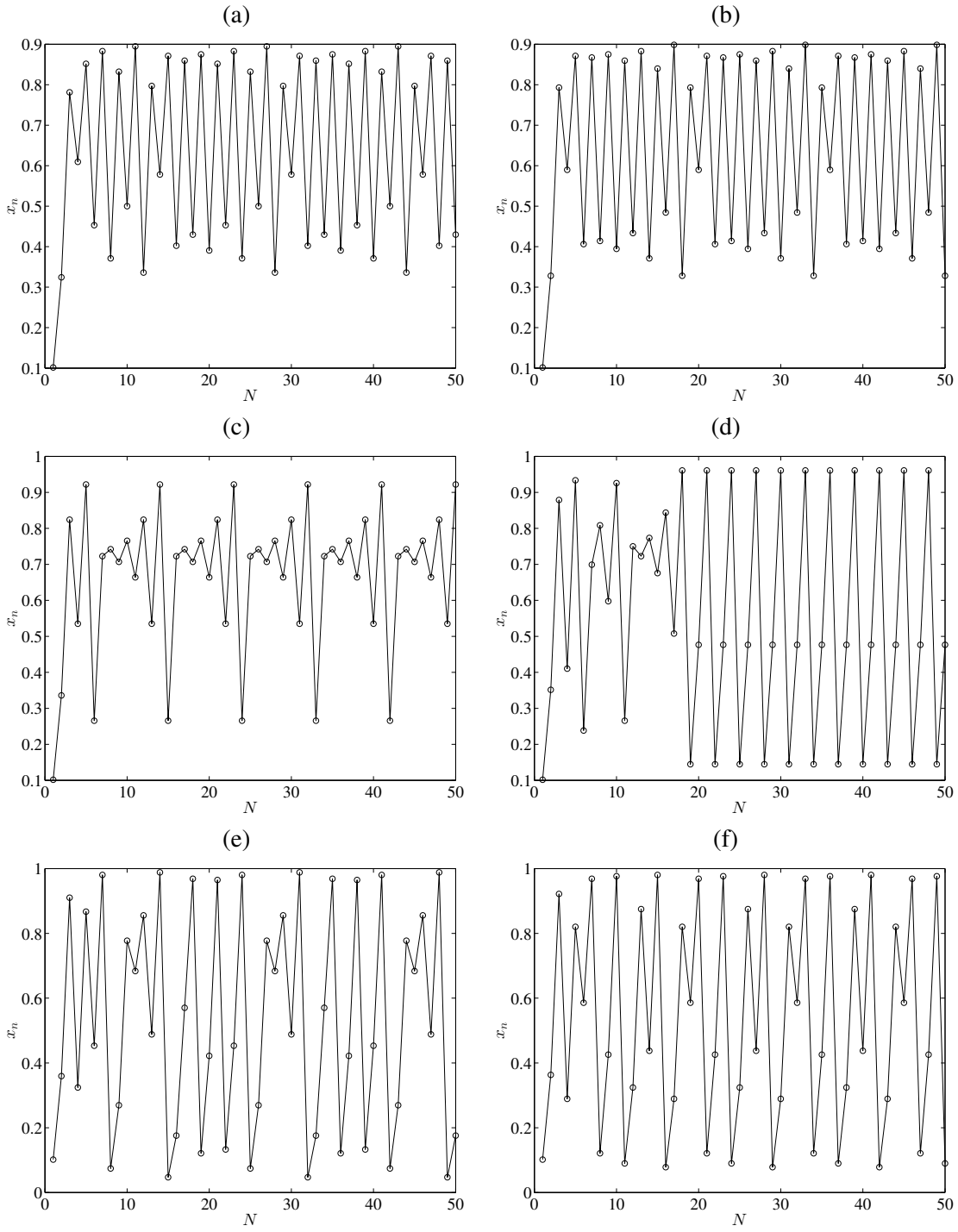


Figura 1.1: Degradação do caos no mapa logístico para os seguintes parâmetros r: (a) $r = 3{,}57$. (b) $r = 3{,}6$. (c) $r = 3{,}7$. (d) $r = 3{,}85$. (e) $r = 3{,}95$. (f) $r = 3{,}99$. O x_n indica o valor da trajetória do sistema no instante N. *Rotinas: Período no Mapa Logístico.*

Muitos pesquisadores foram bem sucedidos ao reduzir a degradação das propriedades caóticas de sistemas digitais [13, 34][1], como o uso de alta precisão finita [35], sistemas caóticos múltiplos em cascata [36], comutação de vários sistemas caóticos [37], acoplamento de dife-

[1] A leitura do trabalho de Cao et al. [13] se faz pertinente, caso o leitor anseie por mais informações.

rentes sistemas caóticos [38, 39] ou perturbar aleatoriamente o sistema caótico [40, 41].

Embora muitas maneiras são efetivas em reduzir esse problema, pouca atenção foi dada em explorar essa limitação para formular algoritmos criptográficos. Os autores em [42] consideraram a precisão finita, mas seu trabalho lida com fenômenos de curto período em sistema caótico usando abordagem binária, que pode ser vista como uma técnica padrão para lidar com a degradação do caos. Ao invés de olhar a precisão digital finita como um problema, esta dissertação propõe novos métodos de encriptação de imagens, usando o erro de precisão finita. O erro é obtido através da implementação de sistemas caóticos, usando extensões intervalares naturais. Ao resolver um sistema caótico, por meio de computação numérica, verifica-se que, com base em duas extensões intervalares naturais, partindo do mesmo conjunto de parâmetros e condições iniciais, após um certo número de iterações, os resultados de cada simulação do sistema divergem (ver Fig. 1.2). Tal evento não poderia ocorrer, devido aos conjuntos iniciais exatamente iguais de parâmetros e condições para cada simulação. Isso acontece em virtude das limitações construtivas dos computadores e ao padrão de ponto flutuante IEEE 754-2008 [43]. Como observado em [44, 45], a sequência de operações aritméticas é diferente em cada extensão intervalar natural. Esse é o elemento necessário para a produção de duas sequências diferentes, a fim de que se gere o limite inferior do erro, o qual apresentou propriedades pseudo-aleatórias satisfatórias.

1.1 Objetivos

Esta pesquisa, de maneira geral, tem como finalidade um estudo sobre criptografia, usando diferentes extensões intervalares naturais e o erro de precisão finita como elementos capazes de gerar uma sequência pseudo-aleatória. Os seguintes objetivos específicos podem ser definidos:

- Analisar como a sensibilidade de sistemas caóticos e as limitações construtivas do computador podem configurar métodos de criptografia.
- Estabelecer uma aproximação entre a Teoria da Segurança da Informação, enquanto codificação de imagens, e os estudos de Computação Aritmética.
- Identificar métodos criptográficos recentes e/ou largamente citados, a fim de analisar e comparar com o método proposto nessa dissertação.

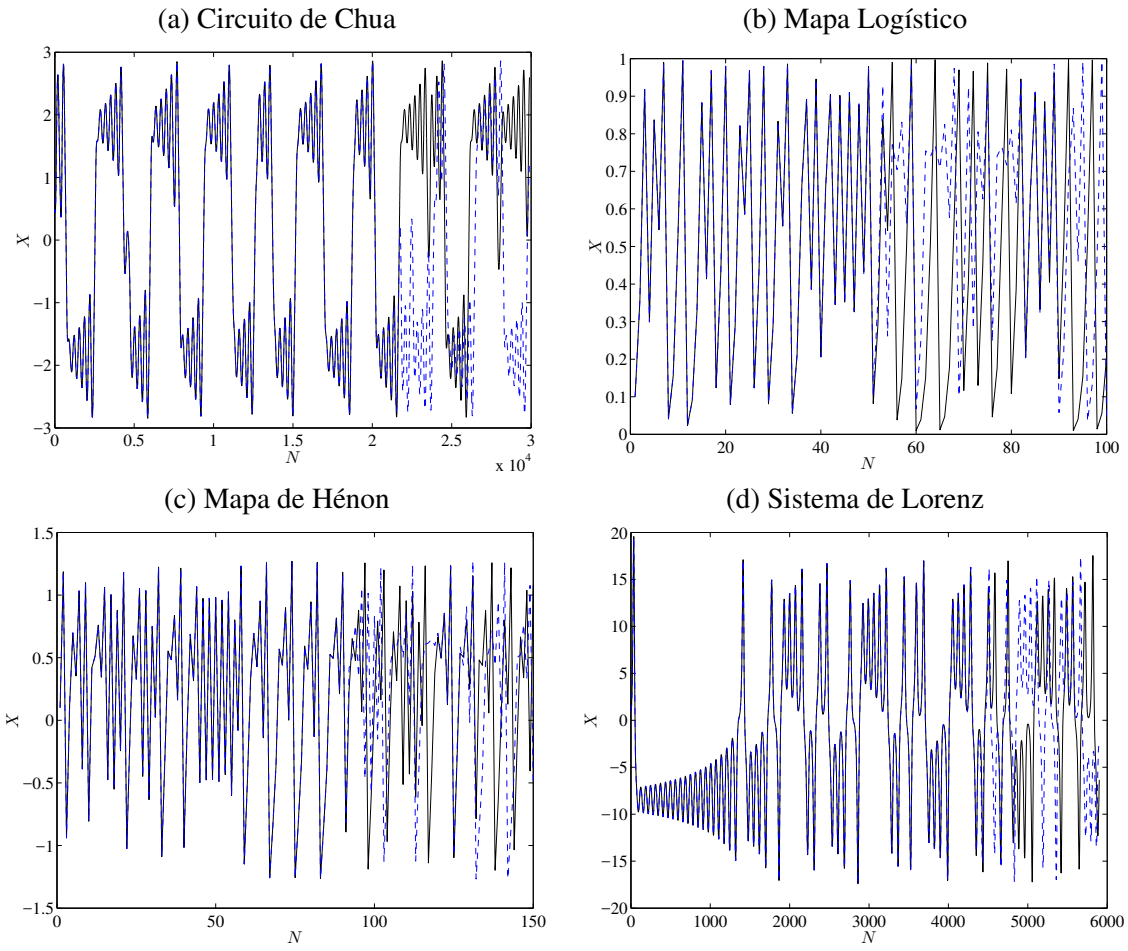


Figura 1.2: Simulação de duas extensões intervalares naturais para quatro diferentes sistemas. Note que, após um certo número de iterações, os resultados das simulações divergem, indicando uma limitação da aritmética de ponto-flutuante IEEE 754-2008. X indica o valor da trajetória do sistema no instante N. *Rotinas: Extensões Intervalares Naturais*.

- Comprovar a característica aleatória do erro.

1.2 Contribuições da Dissertação

Nesta seção, os trabalhos publicados são exibidos como resultados alcançados ao longo do curso de pós-graduação em engenharia elétrica e do desenvolvimento dessa dissertação.

- [46] - L. G. Nardo, A. M. Lima, E. G. Nepomuceno, e J. Arias-Garcia, "Image encryption algorithm using natural interval extensions," in *BTSym 2018 Proceedings*, vol. 1, 2018, pp. 1–5

- [1] - L. G. Nardo, E. G. Nepomuceno, J. Arias-Garcia, e D. N. Butusov, "Image encryption using finite-precision error," *Chaos, Solitons & Fractals*, vol. 123, pp. 69–78, 2019

- [2] - E. G. Nepomuceno, L. G. Nardo, J. Arias-Garcia, D. N. Butusov, e A. Tutueva, "Image encryption based on the pseudo-orbits from 1D chaotic map," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 6, p. 061101, 2019

O primeiro artigo contém as primeiras contribuições do criptossistema usando o circuito de Chua como gerador da sequência pseudo-aleatória. Nesse manuscrito, apenas três testes foram feitos: as análises do histograma, entropia e da correlação dos *pixels* adjacentes na imagem. Desse modo, com um estudo aprofundado sobre ataques cibernéticos, foi descoberto que o algoritmo é vulnerável a ataques diferenciais.

O segundo texto, publicado no jornal intitulado "Chaos, Solitons & Fractals", compõe a maior parte dos resultados apresentados nessa dissertação, além de ser a continuação e o aprimoramento do artigo anterior. Possui um maior número de testes executados e não é vulnerável, a priori, a ataques cibernéticos.

Por fim, o último artigo, publicado no jornal intitulado "Chaos: An Interdisciplinary Journal of Nonlinear Science", traz o desenvolvimento de outro algoritmo para criptografar imagens, usando o mapa logístico com múltiplas perturbações durante o tempo transitório. Os resultados publicados nesse jornal também estão compondo essa dissertação.

1.3 Organização do Trabalho

O trabalho está dividido em cinco capítulos, os quais serão descritos a seguir:

Capítulo 1

No primeiro capítulo, uma breve introdução foi apresentada, com algumas particularidades, como parte de uma revisão bibliográfica e o estado da arte. Dentre tais peculiaridades, foram ressaltadas as dificuldades que os algoritmos tradicionais apresentam ao criptografar imagens, bem como a degradação do caos em diversos sistemas, como é o caso do mapa logístico; além das limitações do padrão de ponto flutuante IEEE 754-2008 como motivações para criar os criptossistemas.

Capítulo 2

Uma breve imersão teórica, apresentando conceitos para a compreensão do restante do trabalho, é apresentada. Primeiramente, uma introdução aos sistemas dinâmicos não-lineares caóticos é descrita, com particular atenção para o circuito de Chua e para o mapa logístico, indi-

cando, também, suas propriedades e razões pelas quais foram escolhidas para integrar as cifras propostas. Em seguida, há a exposição sobre o sistema de ponto flutuante, em especial o padrão IEEE 754-2008, definindo que tal sistema tem suas limitações e estão sujeitos a erros. Assim, o trabalho de Nepomuceno e Martins [44] é detalhado, explicando o conceito de extensões intervalares e expondo o limite inferior do erro. Por fim, um panorama geral sobre criptografia é retratado.

Capítulo 3

Neste capítulo, o procedimento necessário para realizar a encriptação e decriptação, para ambos os algoritmos, são explicados. Ademais, a enumeração dos testes utilizados para validá-los é indicada.

Capítulo 4

Neste capítulo, cada teste estatístico e ataque simulado sob os algoritmos são descritos, exibindo, ainda, a análise de desempenho dos criptossistemas. Além disso, os resultados obtidos são comparados com outros encontrados na literatura.

Capítulo 5

Finalmente, o capítulo 5 contém a conclusão do trabalho, contribuições e implicações que a dissertação traz à literatura, além das perspectivas futuras de trabalho.

CAPÍTULO 2

CONCEITOS PRELIMINARES

2.1 Sistemas Dinâmicos Não-Lineares

Muitos dos elementos existentes no mundo, de certa forma, possuem caráter não-linear. Por esta razão, sistemas dinâmicos não-lineares são largamente estudados. Para elucidar o que está supracitado, seguem algumas não linearidades comuns: tempo de escape finito, múltiplos pontos de equilíbrio, ciclos limites, oscilações subarmônicas, harmônicas ou quase periódicas, bifurcações e caos [47, 48].

Nesta dissertação, o escopo da pesquisa se dedica em torno das aplicações para sistemas seguros, usando a teoria do caos. Além disso, os sistemas caóticos utilizados aqui, abrangem apenas as funções $\dot{x} = f(x)$ e $x_{n+1} = f(x_n)$, sendo $\dot{x}$ a derivada de uma função, x_{n+1} é o estado do sistema no instante $n + 1$, e $f(x)$ a função que representa a relação de um certo elemento.

2.1.1 Sistemas Caóticos

Como referenciado na seção acima, o caos ocorre em sistemas não-lineares [48]. Portanto, tais sistemas apresentam certas particularidades, como dependência das condições iniciais e dos parâmetros, ou seja, para variável de estado x, a solução $x(t,\eta)$, tem-se a condição inicial e o parâmetro $x(t_o,\eta)$, no qual t indica o tempo e η um parâmetro ordinário [47, 49]. Ademais, sistemas caóticos têm, também, a propriedade de existência, do ponto de vista de Lipschitz, e a unicidade de soluções, implicando, portanto, em determinismo.

Apesar dos sistemas caóticos serem determinísticos, não se pode confundir isto com o princípio da causalidade (ordem), pois como já supracitado no Capítulo 1, esses sistemas são sensíveis às condições iniciais. Dessa forma, uma insignificante mudança nas condições inicias, faz com que a resposta do sistema no futuro seja absolutamente diferente do esperado. Além disso, a imprecisão, ou o erro na medição dos parâmetros e/ou no cálculo do sistema, interfere no resultado final. Face a isso, o princípio da incerteza interfere na causalidade [50]. Assim sendo, muitos estudos relatam a interferência dos erros na dinâmica de inúmeros sistemas caóticos [44, 51, 52].

Ao longo dos anos, muitos sistemas, discretos e/ou contínuos, foram denominados caóticos. Desses, podem ser exemplificados o mapa logístico e o circuito de Chua.

2.1.1.1 Mapa Logístico

O mapa logístico, modelado por May [32], é uma simples equação de diferenças, as quais podem apresentar os mais complexos comportamentos, lineares e não-lineares, como trajetórias periódicas, duplicação de período, tempo de escape finito, além de apresentar órbitas caóticas.

A diversidade de comportamentos do mapa logístico pode ser obtida via Eq. (2.1):

$$x_{n+1} = r x_n (1 - x_n), \tag{2.1}$$

sendo que r é o parâmetro de controle, podendo variar de $1 \leq r \leq 4$, e x_n é o n-ésimo estado do sistema.

A Fig. 2.1 mostra o comportamento caótico do sistema para quatro distintos parâmetros r. Note que com o uso de alta precisão numérica, é possível, nesse caso, desfazer a degradação do caos, indicado na Fig. 1.1.

A partir de regimes caóticos, é possível criar criptossistemas seguros. Rostami et al. [53] aplicaram o mapa logístico com o intuito de criar uma cifra rápida e adequada para processamento paralelo. Similarmente, Ismail et al. [54] adicionaram dois parâmetros no mapa logístico fracionário, a fim de obter um grau a mais de liberdade, flexibilidade e controlabilidade no modelo; o que o torna mais atrativo para criar cifras mais seguras.

2.1.1.2 Circuito de Chua

O circuito autônomo de Chua [55] (ver Fig. 2.2(a)) é formado por componentes lineares: um resistor, um indutor e dois capacitores, combinado com um componente ativo e linear por partes (Fig. 2.2(b)), conhecido como diodo de Chua. O sistema formado por três equações diferenciais é representado pela Eq. (2.2). Ademais, a Eq. (2.3) representa a corrente que percorre o diodo de Chua ($i_R(V_{C1})$), sendo que G_a, G_b e B_p são as inclinações e os pontos de quebra do componente não-linear.

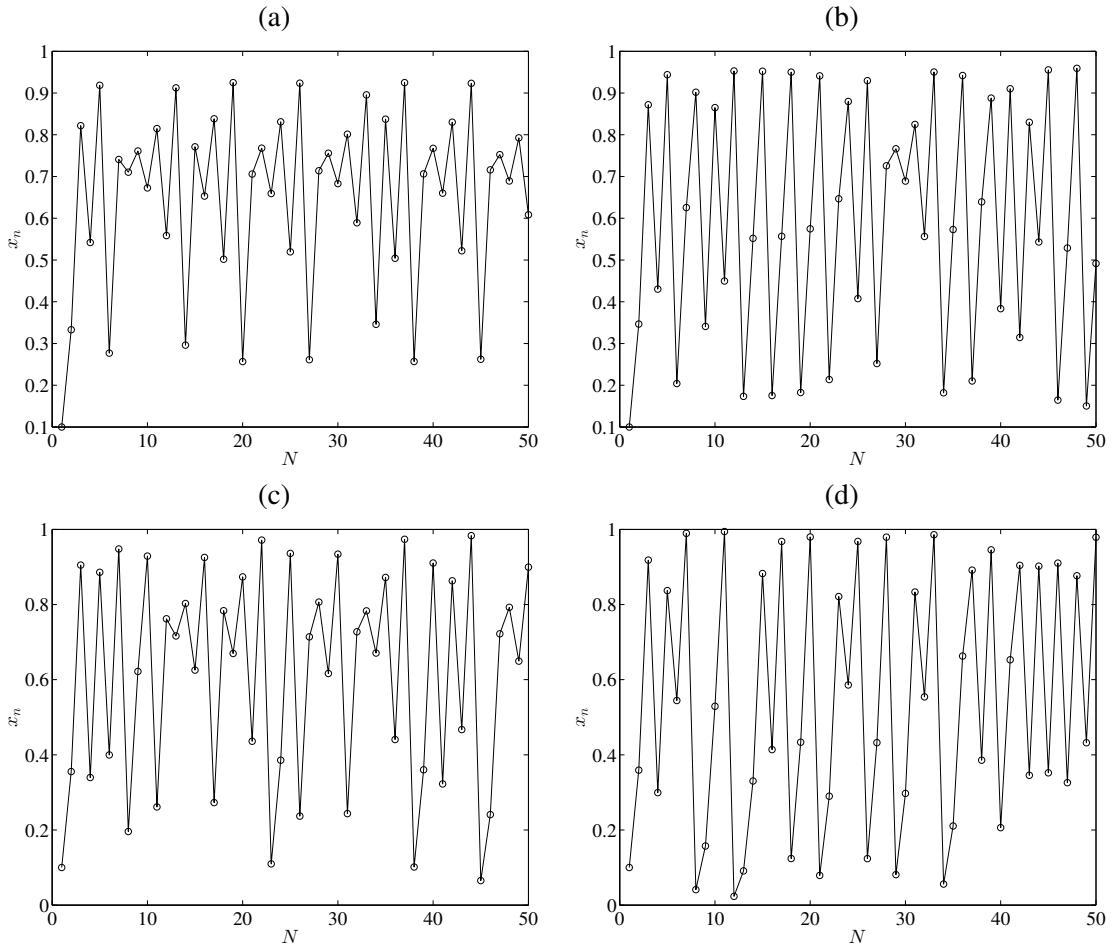


Figura 2.1: Regime caótico no mapa logístico para os seguintes parâmetros r: (a) $r = 3{,}7$. (b) $r = 3{,}85$. (c) $r = 3{,}95$. (d) $r = 3{,}99$. N é o número de iterações e x_n é o estado do sistema no instante N. *Rotinas: Mapa Logístico*.

$$\begin{cases} C_1 \dfrac{dV_{C1}}{dt} &= \dfrac{V_{C2} - V_{C1}}{R} - i_R(V_{C1}) \\ C_2 \dfrac{dV_{C2}}{dt} &= \dfrac{V_{C1} - V_{C2}}{R} + i_L \\ L \dfrac{di_L}{dt} &= -V_{C2} \end{cases} \tag{2.2}$$

$$i_R(V_{C1}) = \begin{cases} G_b V_{C1} + B_p(G_b - G_a), & \text{se} \quad V_{C1} < -B_p \\ G_a V_{C1}, & \text{se} \quad |V_{C1}| \leq B_p \\ G_b V_{C1} + B_p(G_a - G_b), & \text{se} \quad V_{C1} > B_p \end{cases} \tag{2.3}$$

Tal circuito é uma referência no estudo de sistemas dinâmicos. Como uma breve revisão bibliográfica, pode-se citar muitos trabalhos relevantes. Kennedy [56] e Matsumoto [57], de maneira independente, propuseram duas formas de construir o diodo de Chua. Tôrres e Aguirre

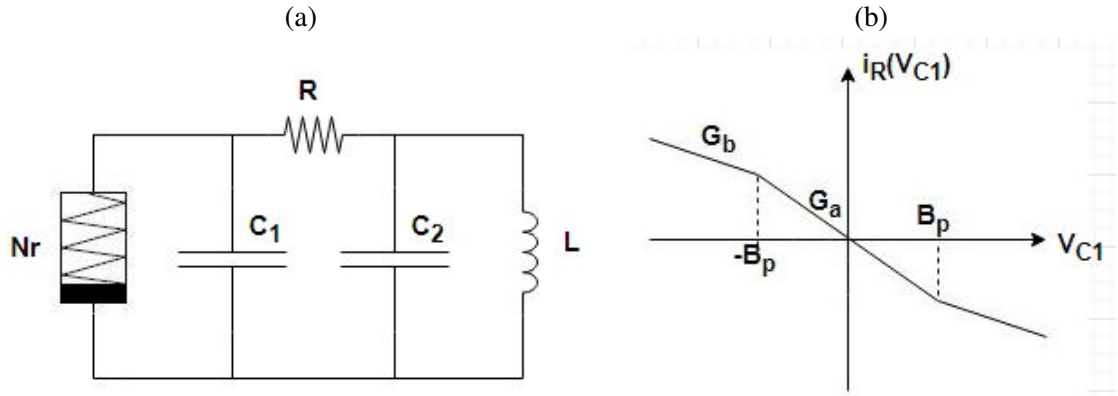


Figura 2.2: (a) Circuito de Chua, formado por componentes lineares, com exceção do diodo de Chua Nr. (b) Curva linear por partes do diodo de Chua, o maior responsável por gerar os mais diversos comportamentos não-lineares presentes neste circuito.

[58] criaram uma plataforma de baixo custo, com o intuito de estudar controle e sincronização, usando o circuito. Kiliç [59], também, fez uma extensa pesquisa com algumas implementações interessantes desse circuito.

De maneira similar ao mapa logístico, existem aplicações do circuito de Chua na criptografia de imagens. Em [60], os autores propuseram um algoritmo baseado em caos, juntamente com o DES (Padrão de Criptografia de Dados, do inglês *Data Encryption Standard*). Além disso, em [61] foi utilizado o memristor com o circuito modificado de Chua para criptografar imagens com eficiência.

Ao criar as cifras baseadas em sistemas caóticos, os cientistas buscam trazer as características caóticas para o criptossistema. Contudo, sob um primeiro olhar, esse comportamento pode apresentar caráter aleatório para um leigo. No caso do circuito de Chua, podem ser buscados dois tipos de atratores caóticos, o atrator dupla-volta e a espira. Note-se na Fig. 2.3, a aperiodicidade das trajetórias.

2.2 Representações Numéricas e Padrão IEEE de Ponto Flutuante

De acordo com Parhami [62], representações numéricas se configuram enquanto um tópico de suma importância, pois cada representação reflete em uma determinada precisão numérica, e tempo para realizar as operações aritméticas. Na literatura, existem diversas representações numéricas, como, por exemplo, representações baseadas em complemento de 1 ou de 2, sistema de número redundante, sistema de número de resíduos, além da aritmética de ponto fixo e de ponto flutuante [62].

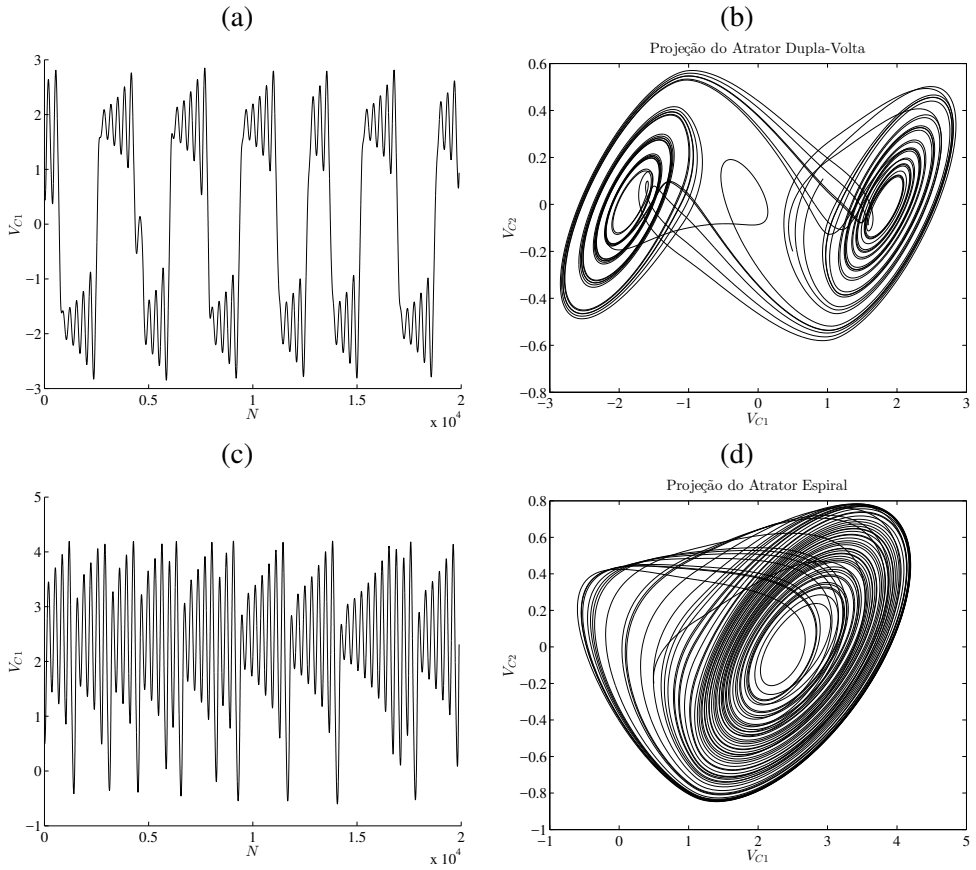


Figura 2.3: Resultados das simulações do circuito de Chua obtidos para $G_a = -0{,}68mS$, $G_b = -0{,}37mS$, $B_p = 1{,}1V$, $C_1 = 10nF$, $C_2 = 100nF$, $L = 19mH$ e a resistência R, ora $R = 1800\Omega$ (para (a) e (b)), ora $R = 1980\Omega$ (para (c) e (d)). V_{C1} e V_{C2} são as tensões respectivas do capacitor 1 e 2 do circuito, e N é o número de iterações, o qual pode ser facilmente convertido para o tempo t, multiplicando, apenas, com o passo de integração fixo. *Rotinas: Circuito de Chua.*

Baseada na notação exponencial, a representação de ponto flutuante é largamente utilizada atualmente, uma vez que apresenta inúmeras vantagens em relação às outras representações aritméticas. Como exemplo, o intervalo de representação numérica é superior às outras representações, justamente por usar um expoente na reprodução de um determinado número [63, 64].

Conforme descrito por Overton [63], a aritmética de ponto flutuante começou a ser usada, largamente, em meados na década de 1950. Sendo que cada empresa adotava o seu padrão, o que causava incompatibilidade numérica e, por conseguinte, dificuldade na reprodução de resultados. Anos mais tarde, em meados da década de 1970 e 1980, a colaboração entre acadêmicos e importantes indústrias do cenário tecnológico da época, resultou no padrão IEEE 754 de ponto flutuante [43], adotado majoritariamente em todos os computadores e máquinas usadas desde então. A Tabela 2.1 detalha, apresentando algumas máquinas, os diferentes padrões de representação de ponto flutuante.

Tabela 2.1: Diferentes padrões de representação numérica. As diversas características distintas entre os padrões traziam uma série de adversidades, como por exemplo, a dificuldade na reprodução de resultados. Na tabela a precisão indica o número de *bits* disponível na mantissa, e os valores de expoente mínimo e máximo apontam o intervalo que o expoente pode ser declarado usando a aritmética de ponto flutuante. Retirado e adaptado de [65].

Máquina	Base	Precisão	$Expoente_{min}$	$Expoente_{max}$
VAX	Binária	24	-128	127
VAX	Binária	56	-128	127
CRAY-I	Binária	48	-16384	16383
IBM 3081	Hexadecimal	6	-64	63
IBM 3081	Hexadecimal	14	-64	63
IEEE Single	Binária	24	-125	128
IEEE Double	Binária	53	-1021	1024

2.2.1 Padrão IEEE 754 de Ponto Flutuante

A partir da representação de ponto flutuante, um determinado número x pode ser representado por

$$x = \pm Mn \times 2^{Ex}, \tag{2.4}$$

sendo que Ex é o expoente, Mn é a mantissa, compreendido como $1 \leq Mn < 2$, e 2 é a base binária, muito trabalhada em sistemas digitais [63].

Na representação de ponto flutuante, padronizada pelo IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos, do inglês *Institute of Electrical and Electronics Engineers*), há dois tipos de formatos mais usuais, variando a precisão e o número de *bits*. O formato *single*, simples, utiliza 32 *bits*, dos quais 1 *bit* é para o sinal (sendo o valor 0 para números positivos e 1 para negativos), 8 *bits* designado para representar o expoente e 23 *bits* para a mantissa (ver Fig. 2.4). Por outro lado, o formato *double*, dupla, dispõe de 64 *bits*, sendo 1 *bit* para representar o sinal, 11 e 52 *bits* disponíveis para representar o expoente e a mantissa, respectivamente [66]. De modo que o formato *double* apresenta um maior número de *bits* e, por consequência, uma melhor precisão e uma maior faixa de representação numérica, este é o formato cujo uso é aconselhável.

b_0	$b_1b_2b_3b_4b_5b_6b_7b_8$	$b_9b_{10}b_{11}b_{12}b_{13}b_{14}b_{15}b_{16}b_{17}b_{18}b_{19}b_{20}b_{21}b_{22}b_{23}b_{24}b_{25}b_{26}b_{27}b_{28}b_{29}b_{30}b_{31}$
Sinal	Expoente	Mantissa

Figura 2.4: Formato *single* via padrão IEEE 754-2008. Retirado e Adaptado de [66].

A norma IEEE 754[1] traz várias vantagens e pontos que devem ser discutidos. Além da adoção do padrão, na maioria das máquinas usadas atualmente, gerando consistência na representação dos números e reprodução de resultados. Conforme comentado na seção anterior, o padrão também formula como devem ser abordados certos casos excepcionais, e prescreve como é feito o correto arrendondamento de um número [63].

No que tange os casos excepcionais descritos na norma IEEE 754-2008 [43], deve-se, primeiramente, declarar que a mesma traz representações especiais para $+0$, -0, $+\infty$ e $-\infty$. Dito isso, o padrão estabelece casos em que o zero é obtido via divisão por infinito, e vice-versa. Sendo que antes era apenas dito que ocorria um erro na operação [63].

Há casos, também, em que o resultado da operação não é um número complexo ou real, ou seja, o resultado da operação é inválida. Dessa forma, a norma estabelece que o resultado é NaN (Não é um Número, do inglês *Not a Number*). Exemplos que originam tal resultado são as operações $0 \times \infty$, $0/0$ e $\infty - \infty$ [63].

No tocante aos arredondamentos, a norma prevê que caso um número ou resultado de uma operação aritmética não seja um ponto flutuante, este deve ser arredondado. Existem quatro tipos de modos de arredondamento [63]:

- **Arredondamento para o mais próximo**: um determinado número x é arredondado para o ponto flutuante mais próximo a ele. Esse arredondamento é o padrão que a IEEE exige.
- **Arredondamento em direção para** $+\infty$: um número x é arredondado para o próximo ponto flutuante acima dele.
- **Arredondamento em direção para** $-\infty$: x é arredondado para o subsequente ponto flutuante menor que ele, ou maior em valor absoluto.
- **Arredondamento em direção para o zero**: Caso $x < 0$, o número x é arredondado para o próximo ponto flutuante acima dele. Caso $x > 0$, o número x é arredondado para o seguinte ponto flutuante menor que ele.

Overton [63] ainda menciona que as operações aritméticas implementadas em um computador, sujeitas à operação de arredondamento, podem gerar um erro associado ϵ, ou seja:

[1]Recentemente, em 2019, foi aprovada uma nova versão do padrão IEEE de ponto flutuante, alterando pequenos detalhes. Tais alterações podem ser consultadas no site: `http://grouper.ieee.org/groups/msc/ANSI_IEEE-Std-754-2019.html`

$$x \otimes y = round(x \times y) = (x \times y)(1 + \epsilon), \tag{2.5}$$

sendo que $\otimes$ é a operação de multiplicação, feita em um ambiente digital, $round$ é o procedimento de arredondamento, e ϵ é o erro relativo.

A partir do que foi supracitado, é possível perceber a importância de descobrir, ou, ao menos estimar o erro, investigando se ele afeta significativamente os resultados obtidos. Um modo de analisar o erro é calcular o limite inferior do erro, sendo que os autores de [45] já estabeleceram uma analogia entre essa técnica e os erros da aritmética de ponto flutuante.

2.3 Limite Inferior do Erro

Nepomuceno e Martins [44] desenvolveram uma técnica para estimar a propagação do erro em simulações numéricas. Para entender os mecanismos dessa técnica, algumas definições serão dadas a seguir:

Definição 1: Um mapa, ou um sistema, originam uma sequência de valores que configuram uma órbita ou trajetória, representada por $\{x_n\} = [x_0, x_1, x_2, x_3, \ldots, x_n]$.

Definição 2: Uma pseudo-órbita é a aproximação de uma órbita verdadeira, representada por $\{\hat{x}_{i,n}\} = [\hat{x}_{i,0}, \hat{x}_{i,1}, \ldots, \hat{x}_{i,n}]$. Uma pseudo-órbita é originada devido à precisão finita do computador [44].

Intervalo e a extensão intervalar natural foram definidos por Moore et al. [67] conforme apresentado a seguir:

Definição 3: Um intervalo é um conjunto fechado de números reais $x \in \mathbb{R}$ tal que $I_X = [\underline{X},\overline{X}] = x : \underline{X} \leq x \leq \overline{X}$.

Definição 4: Uma extensão intervalar natural de uma função f é uma função com valores dados em intervalo F de uma variável intervalar X, com a propriedade $F(x) = f(x)$ [44, 67].

Aqui, são apresentados exemplos de extensões intervalares naturais, sendo que para o circuito de Chua, tais extensões intervalares são dadas pelas Eq. (2.6) e Eq. (2.7), e pelo mapa logístico a partir das Eq. (2.8) e Eq. (2.9):

$$C_1 \frac{dV_{C1}}{dt} = \frac{V_{C2} - V_{C1}}{R} - i_R(V_{C1}), \tag{2.6}$$

$$C_1 \frac{dV_{C1}}{dt} = \frac{V_{C2}}{R} - \frac{V_{C1}}{R} - i_R(V_{C1}), \tag{2.7}$$

$$x_{n+1} = r x_n (1 - x_n), \tag{2.8}$$

$$x_{n+1} = r x_n - r x_n x_n. \tag{2.9}$$

Finalmente, o limite inferior do erro pode ser estabelecido da seguinte forma:

Definição 5: Sejam duas pseudo-órbitas $\hat{x}_{a,n}$ e $\hat{x}_{b,n}$, surgindo a partir de duas extensões intervalares naturais diferentes da função $f(x)$, o limite inferior do erro δ é dado por [44, 45]:

$$\delta = \frac{|\hat{x}_{a,n} - \hat{x}_{b,n}|}{2}, \tag{2.10}$$

sendo que δ tem a mesma unidade de medida das pseudo-órbitas $\hat{x}_{a,n}$ e $\hat{x}_{b,n}$.

2.4 Criptografia

Criptografia é a ciência que busca criar técnicas para tornar dados ilegíveis, possibilitando manter as informações seguras e transmiti-las de maneira inviolável por meio de redes não seguras [68–70].

A modo de exemplo, a Fig. 2.5 exibe o processo geral de criptografia, descrito por Boneh e Shoup [70], no qual, a partir de uma mensagem m e uma chave secreta K, por meio de uma função de encriptação E é possível gerar uma mensagem ilegível $c = E(K,m)$. Para conseguir obter a mensagem original novamente, é preciso ter o conhecimento da chave secreta usada e a função de decriptação D, assim $m = D(K,c)$.

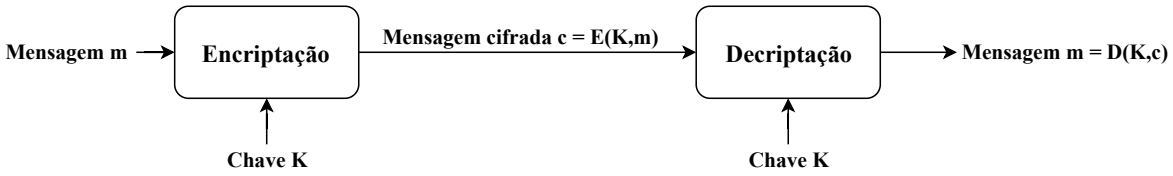


Figura 2.5: Processo geral de criptografia. Por meio de uma chave K, e uma função de encriptação E, é possível criptografar uma mensagem m. Ao chegar ao destinatário, a mensagem original pode ser obtida novamente via o conhecimento da chave secreta e da função de decriptação D. Retirado e adaptado de [70].

Dissecando a Fig. 2.5, as funções de encriptação e decriptação podem ser desempenhadas a partir de geradores de números aleatórios (RNG, do inglês *Random Number Generator*) e/ou

pseudo-aleatórios (PNRG). Segundo Burnett e Paine [69], os RNGs são geradores que fornecem números aleatórios a partir de várias entradas imprevisíveis, como o decaimento radioativo de um certo elemento e mudanças das condições atmosféricas. Entretanto, por se tratar de entradas imprevisíveis, a saída raramente será repetida, dificultando no processo de criptografia. Portanto, os RNGs não são praticáveis na realidade.

Uma alternativa para tal problema, seria o uso de PRNGs, os quais são algoritmos determinísticos que possuem característica aleatória e que ao contrário dos RNGs, a saída pode ser repetida, no caso do uso da mesma condição inicial [69]. Ademais, Boneh e Shoup [70] estabelecem uma definição mais formal no que tangem os PRNGs.

Definição 6: Um gerador de número pseudo-aleatório, PRNG, é um algoritmo G que, a partir de uma determinada condição inicial ou como é comumente chamado, uma semente s (definido em um espaço finito S_f), gera uma saída q, pertencente a um conjunto finito Q_f. Logo, G é definido a partir de (S_f,Q_f).

Continuando a analisar a Fig. 2.5, uma importante ferramenta, a qual é bastante usada, é o criptograma bit-XOR. Esta operação de controle de *bits* é útil, pois a probabilidade do resultado da operação ser 0, ou 1, é de 50%; ao contrário de outros manipuladores lógicos [69]. Outro motivo, é o fato de que o operador lógico XOR ($\oplus$) tem uma propriedade interessante, a qual é $(m \oplus K) \oplus K = m$ [71]. Em outras palavras, ao utilizar, juntamente com a mensagem m, uma chave secreta K duas vezes, o resultado final é a mensagem m; representando, portanto, todo o processo criptográfico indicado na Fig. 2.5.

Basicamente, há dois tipos gerais de criptossistemas, a criptografia simétrica (ver Fig. 2.6) e a criptografia assimétrica (ver Fig. 2.7). No primeiro caso, a chave para encriptar e descriptografar uma mensagem é a mesma [72]. No segundo caso, há duas chaves diferentes para a encriptação e decriptação, a primeira, conhecida como chave pública, é utilizada para realizar a encriptação, enquanto a chave privada, descriptografa a mensagem [69, 72].

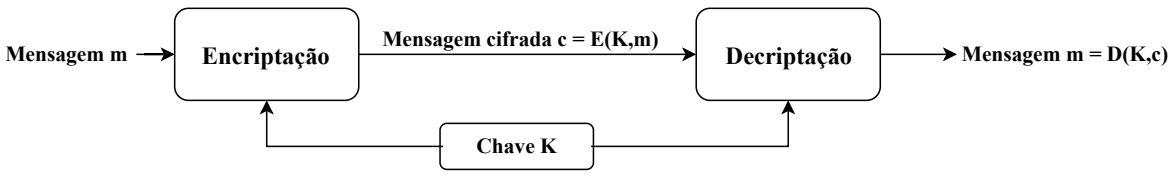


Figura 2.6: Criptografia simétrica. A partir da mesma chave K, é possível criptografar e descriptografar uma determinada mensagem. Retirado e adaptado de [70].

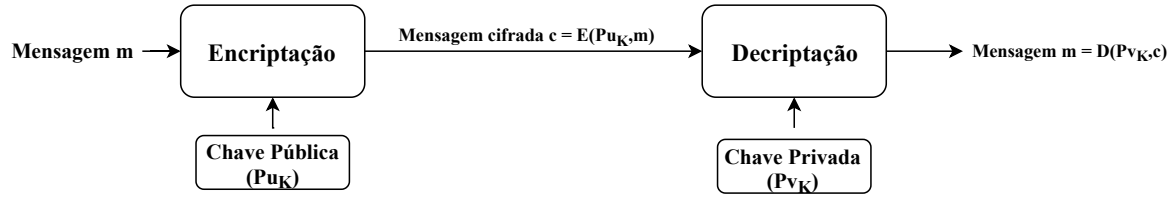


Figura 2.7: Criptografia assimétrica. Com uma chave pública, e outra secreta, são realizados os seguintes processos, encriptação e decriptação, respectivamente. O uso trocado das chaves, ou a utilização de uma mesma chave nos dois processos, faz com que ocorram erros no método. Retirado e adaptado de [70].

Por fim, Alvarez e Li [72] descrevem algumas propriedades que caracterizam as cifras, a fim de que elas sejam consideradas seguras e eficientes, tais como confusão, difusão, determinismo e alta complexidade. Dessa maneira, os autores também citam que os sistemas dinâmicos caóticos têm propriedades que se equivalem às requeridas pelas cifras, são elas: a ergodicidade, a sensibilidade às condições iniciais, a dinâmica determinista, a estrutura, e as trajetórias complexas. Concluindo, tais similaridades sustentam a justificativa para o estudo e para a criação de cifras baseadas em sistemas caóticos.

CAPÍTULO 3

DESCRIÇÃO DAS CIFRAS E MÉTODOS

Neste capítulo, o processo para criptografar e descriptografar imagens, para os dois algoritmos propostos, são descritos paulatinamente. Ambas as cifras foram construídas baseando-se nos conceitos de extensões intervalares naturais e no erro de precisão finita, presentes nos computadores. Por fim, ainda nesse capítulo, uma tabela comparativa de ambas as cifras é mostrada, enumerando, também, os testes realizados para validá-las.

Vale salientar que as cifras serão testadas em imagens com tons de cinza e que uma figura pode ser representada, matematicamente, por uma matriz de tamanho equivalente ao número de *pixels* na largura e no comprimento da imagem em questão. Cada elemento da matriz representa um *pixel* da figura, apresentando um número de 8 *bits*, o qual compreende, em base decimal, um valor entre 0 e 255, e que indica o tom de cinza do determinado *pixel*. Dessa maneira, o objetivo dos algoritmos, descritos a seguir, é criar uma chave secreta de mesma ordem que a imagem que, após o processo de encriptação, essa se torne ilegível.

3.1 Algoritmo I - Sistema Contínuo

Nas duas subsequentes seções serão descritos, gradualmente, os algoritmos para encriptação e decriptação de imagens propostas no artigo [1], intitulado “Image encryption using finite-precision error”. Por fim, uma outra seção é adicionada, a fim de detalhar os parâmetros e imagens utilizados para validar o criptossistema.

3.1.1 Encriptação de Imagens

A obtenção da chave secreta, a qual é uma sequência pseudo-aleatória, e o processo de criptografia podem ser resumidos pelos seguintes passos, adaptados de [13]:

Passo 1: Como forma de obter chaves secretas diferentes para imagens distintas, assegurando as propriedades de difusão e confusão, a condição inicial V_{C1} do circuito de Chua é

atualizada por um fator mostrado pela Eq. (3.1):

$$V'_{C1} = V_{C1} + F_o, \tag{3.1}$$

sendo que F_o é o fator obtido da imagem a ser codificada, dado por:

$$F_o = \frac{1}{Lg \times W} \sum_{i=1}^{Lg} \sum_{j=1}^{W} I_p(i,j) \times 10^{-5}, \tag{3.2}$$

em que Lg e W são o comprimento e a largura, nessa ordem, da imagem simples I_p e i e j são os respectivos valores de coordenadas (i,j). Dessa maneira, a imagem a ser encriptada e a chave secreta estão intrinsecamente ligadas, proporcionando, portanto, as características supracitadas.

Passo 2: A partir de uma imagem com $Lg \times W$ *pixels*, usando o método Runge-Kutta de 4^a ordem, um passo de integração igual a 10^{-6}, execute duas vezes a simulação do circuito de Chua com $tr + Lg \times W - 1$ iterações, empregando duas extensões intervalares naturais com as mesmas condições iniciais impostas, a fim de gerar duas sequências S_1 e S_2.

Devido ao fato de que o erro entre as pseudo-órbitas é pequeno no início da simulação, os tr pontos são descartados, pois, se deixados, a chave traria informações ao *hacker*, facilitando a quebra do algoritmo. Os tr pontos são determinados pela Eq. (3.3) [44]:

$$tr = \frac{\hat{x}_{a,n} - \hat{x}_{b,n}}{\hat{x}_{a,n} + \hat{x}_{b,n}}, \tag{3.3}$$

sendo que tr calcula o erro relativo entre as pseudo-órbitas e $\hat{x}_{i,n}$ é a pseudo-órbita i no n-ésimo estado do sistema.

Passo 3: Depois que as duas sequências S_1 e S_2 são geradas, o logaritmo do limite inferior do erro é calculado, gerando uma única sequência $S \in \mathbb{R}^{Lg \times W - 1}$:

$$S = log_{10} \frac{|S_1 - S_2|}{2}. \tag{3.4}$$

Como as imagens testadas são acinzentadas, posto que a matriz de *pixels* varia os valores entre 0 (tom preto) e 255 (tom branco), é necessário normalizar a sequência S.

Passo 4: O processo de normalização da sequência S é feito da seguinte maneira:

$$S_n = uint8(mod(S \times 10^{15}, 256)), \tag{3.5}$$

sendo que S_n é a sequência normalizada. $Uint8$ é um algoritmo disponível desde a versão do *software Matlab R2006a*, o qual converte um número real em um número inteiro positivo de 8 *bits*, e mod representa a operação do módulo: o resto da divisão $S \times 10^{15}$ por 256.

Passo 5: Para transformar a sequência S_n em uma matriz com o formato equivalente ao da imagem original, o seguinte processo é feito:

$$S_n = vec2mat(S_n,W), \tag{3.6}$$

$vec2mat$ é o processo de conversão de vetor para matriz, e W é o número de colunas que a matriz deve ter.

Passo 6: Para criptografar a imagem (I_p) em uma imagem cifrada (I_c), a operação bit-XOR, *bit* a *bit*, é executada com a sequência normalizada e a imagem, ou seja,

$$I_c(i,j) = S_n(i,j) \oplus I_p(i,j). \tag{3.7}$$

Esses passos são ilustrados pelo sistema criptográfico exibido na Fig. 3.1. É válido ressaltar que esse sistema respeita o princípio de Kerckhoffs [73]; em outras palavras, o único parâmetro secreto é a chave.

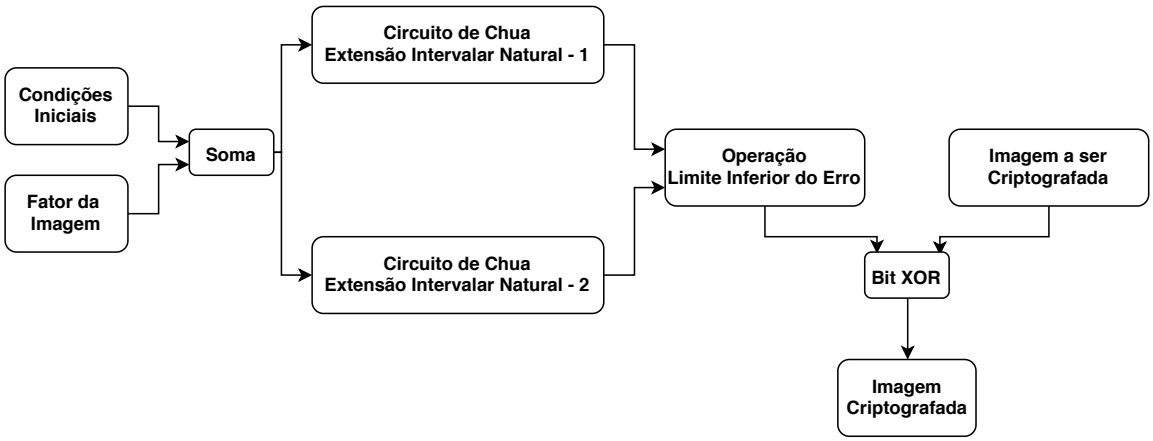


Figura 3.1: Processo de encriptação para o algoritmo I [1]. O esquema mostra os principais passos da técnica proposta. A novidade proposta, aqui, é baseada no limite inferior do erro [44, 45], como também em extensões intervalares naturais e no erro de precisão finita presente nos computadores.

3.1.2 Decriptação de Imagens

A cifra a ser utilizada é de chave simétrica, ou seja, a chave para encriptação e decriptação da imagem é a mesma. Dessa forma, o processo para converter a imagem cifrada na imagem

original se faz da seguinte maneira:

Passo 1: Com o conhecimento prévio do fator obtido via Eq. (3.2), atualize a condição inicial V_{C1}.

Passo 2: Para uma imagem cifrada, com $Lg \times W$ *pixels*, simule duas vezes, a partir das duas extensões intervalares escolhidas previamente, e com os mesmos parâmetros escolhidos no processo de encriptação, o circuito de Chua com $tr + Lg \times W - 1$ iterações, via método Runge-Kutta de 4^a ordem e um passo de integração igual a 10^{-6}, a fim de gerar as sequências S_1 e S_2.

Passo 3: Determine os tr pontos, vide Eq. (3.3), e exclua-os das sequências geradas anteriormente.

Passo 4: Gere uma única sequência S, vide Eq. (3.4), a partir das sequências S_1 e S_2.

Passo 5: Normalize a sequência S a partir da Eq. (3.5).

Passo 6: Transforme a sequência S_n em uma matriz com o formato equivalente ao da imagem cifrada, vide Eq. (3.6).

Passo 7: Para descriptografar a imagem (I_c) e obter novamente a imagem original (I_p), a operação bit-XOR, *bit* a *bit*, é executada com a sequência normalizada e a imagem cifrada, ou seja,

$$I_p(i,j) = S_n(i,j) \oplus I_c(i,j). \tag{3.8}$$

Os passos do algoritmo para decriptação são exibidos na Fig. 3.2.

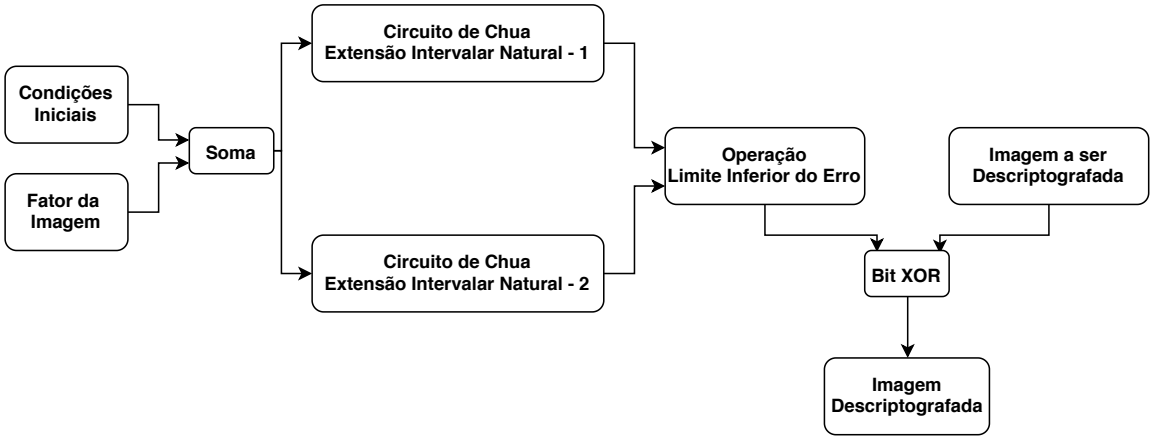


Figura 3.2: Processo de decriptação para o algoritmo I [1].

3.1.3 Definição dos Parâmetros e Imagens Utilizados

Os seguintes parâmetros, mostrados na Fig. 3.3, foram usados para gerar a chave secreta para esta cifra, baseados em um circuito montado por Aguirre [74]. Ademais, o índice tr seguiu a igualdade $log_{10}tr = -10$, e as extensões intervalares naturais escolhidas estão exibidas nas Eq. (3.9) e Eq. (3.10):

$$C_1 \frac{dV_{C1}}{dt} = \frac{V_{C2} - V_{C1}}{R} - i_R(V_{C1}) \tag{3.9}$$

$$C_1 \frac{dV_{C1}}{dt} = \frac{V_{C2}}{R} - \frac{V_{C1}}{R} - i_R(V_{C1}) \tag{3.10}$$

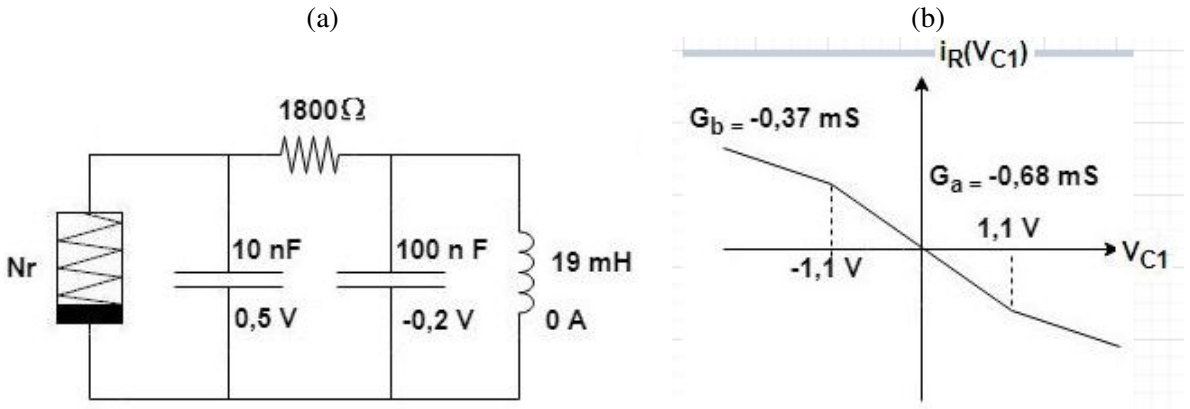


Figura 3.3: (a) Circuito de Chua com os valores nominais dos componentes e as condições iniciais do sistema. (b) Valores das inclinações e pontos de quebra da curva linear por partes.

Para demonstrar a eficiência e a segurança do método, foram utilizadas sete imagens, largamente empregadas em trabalhos similares: Lena (256×256), Barco (256×256), Casa (256×256), Pimenta (256×256) e Cameraman (256×256), além das imagens totalmente Preta (512×512) e Branca (512×512). Os fatores F_o das figuras estão mostrados na Tabela 3.1.

3.2 Algoritmo II - Sistema Discreto

A descrição dos algoritmos do artigo [2], intitulado "Image encryption based on the pseudo-orbits from 1D chaotic map", bem como a definição dos parâmetros e imagens utilizados são feitos a seguir.

Tabela 3.1: Fatores F_o obtidos a partir das seguintes imagens. Tais fatores foram calculados de acordo com a Eq. (3.2), visando aumentar as propriedades de difusão e confusão das cifras propostas. A representação hexadecimal de cada fator da imagem foi apresentada na terceira coluna, para garantir a reprodutibilidade dos resultados, aqui, apresentados.

Imagem	**Fator F_o**	**Representação Hexadecimal**
Lena	$98{,}67648315429686 \times 10^{-5}$	`3f502acaab8a5ce5`
Barco	$136{,}0362091064453 \times 10^{-5}$	`3f5649c5ac471b47`
Casa	$137{,}9846038818359 \times 10^{-5}$	`3f569b7e670e2c12`
Pimenta	$123{,}1443634033203 \times 10^{-5}$	`3f542d0c88a47ecf`
Cameraman	$118{,}7226562500000 \times 10^{-5}$	`3f537396d0917d6b`
Preto	$97{,}65624999999999 \times 10^{-9}$	`3e7a36e2eb1c432c`
Branco	$0{,}002550000000000$	`3f64e3bcd35a8587`

3.2.1 Encriptação de Imagens

O algoritmo para criptografar imagens é descrito paulatinamente [13, 46]:

Passo 1: Escolha as condições iniciais e os valores que serão usados para perturbar o sistema, vide Eq. (3.11):

$$M = [m_1 \quad m_2 \quad m_3 \quad m_4 \quad m_5 \quad c_i], \tag{3.11}$$

sendo que $m_1 \in [0;1]$ é a condição inicial do sistema caótico. Os parâmetros $m_i \in [0;1], i = 2 \ldots 5$ são os valores que perturbarão o sistema. Aqui, foram escolhidos quatro valores para a pertubação, porém tais números podem ser facilmente aumentados, o que levará a um espaço amostral maior para a chave secreta. O parâmetro c_i também é um valor que perturbará o sistema, entretanto, esse depende apenas da imagem que será encriptada. O valor c_i é definido como:

$$c_i = \frac{1}{(Lg \times W)^2} \sum_{i=1}^{Lg} \sum_{j=1}^{W} I_p(i,j), \tag{3.12}$$

em que Lg e W são o comprimento e a largura, respectivamente, e i e j são as coordenadas referentes ao *pixel* da imagem simples I_p.

Passo 2: Atualize a primeira condição inicial ($X_0 = Y_0 = M(1,1)$) e o parâmetro r do mapa logístico. Tal parâmetro de controle deve estar compreendido entre $3{,}57 \leq r < 4$ para apresentar comportamento caótico [32].

Passo 3: A partir de duas extensões intervalares naturais, simule duas vezes o mapa logístico com $tr + Lg \times W - 1$ iterações. Os tr pontos são respostas transitórias descartadas. Após $k \times 100$

iterações, $k = 1,2 \ldots 6$, as duas extensões intervalares são perturbadas com o vetor M,

$$X_{n+1} = \text{mod } (X_{n+1} + M(k,1),1), \tag{3.13}$$

$$Y_{n+1} = \text{mod } (Y_{n+1} + M(k,1),1), \tag{3.14}$$

no qual mod é a operação do módulo. Cem iterações são suficientes para garantir a perda de todos os dígitos significativos em um formato de ponto flutuante de precisão dupla [75, 76].

Passo 4: Gere a sequência criptográfica S, calculando o limite inferior do erro, dado através de:

$$S = \frac{|X - Y|}{2}. \tag{3.15}$$

Passo 5: O processo de normalização da sequência S é feito da seguinte forma:

$$S_n = uint8(mod((S/min(S)) \times 256,256)), \tag{3.16}$$

sendo $uint8$ o algoritmo para converter os números com precisão dupla em inteiros, sem sinal, de 8 *bits*, e $\min$ a operação que encontra o menor valor de uma determinada sequência.

Passo 6: Transforme o vetor S_n em uma matriz, usando o algoritmo $reshape$, disponível no *Matlab*:

$$S_n = reshape(S_n,[Lg, W]). \tag{3.17}$$

Passo 7: Obtenha a imagem cifrada I_c aplicando a operação bit-XOR entre I_p e S_n,

$$I_c(i,j) = I_p(i,j) \oplus S_n(i,j). \tag{3.18}$$

A Fig. 3.4 ilustra o criptossistema. O uso do mapa 1D com múltiplos valores de perturbação e o erro do limite inferior são as novas características do algoritmo proposto. É importante notar que o esquema aqui proposto pode ser facilmente adaptado à imagem colorida. Por exemplo, a geração da chave secreta pode ser feita seguindo o método projetado em [77], o qual propõe que a chave secreta, uma vez gerada, pode ser separada nos três componentes R (vermelho, do inglês *red*), G (verde, do inglês *green*) e B (azul, do inglês *blue*), que compõe o sistema de imagens coloridas RGB.

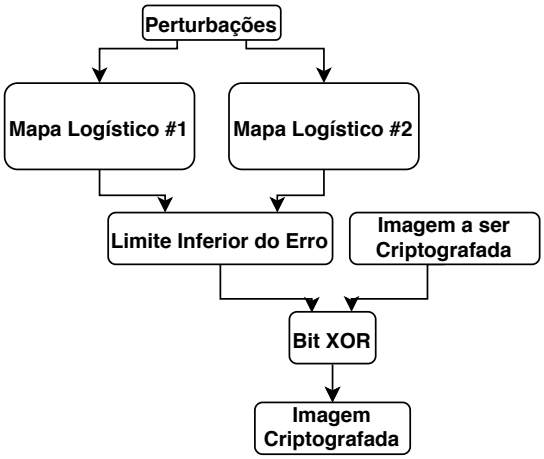


Figura 3.4: Processo de encriptação usando o limite inferior do erro e múltiplas perturbações [2].

3.2.2 Decriptação de Imagens

Assim como o primeiro algoritmo, esta cifra também é de chave simétrica. Portanto, o processo para converter a imagem cifrada na imagem original se faz da seguinte maneira:

Passo 1: A partir das condições iniciais, os valores para perturbar o sistema e o parâmetro c_i conhecidos a priori, atualize o vetor M, vide Eq. (3.11). Acrescente, também, o valor utilizado do parâmetro r do mapa logístico.

Passo 2: Considerando as duas extensões intervalares naturais do mapa logístico utilizadas previamente, além do vetor M, simule duas vezes o mapa 1D com $tr + Lg \times W - 1$ iterações. Posteriormente, descarte os tr pontos.

Passo 3: Gere uma única sequência S (Eq. (3.15)) a partir das trajetórias X e Y obtidas no passo anterior.

Passo 4: Normalize a sequência S a partir da Eq. (3.16).

Passo 5: A partir do algoritmo $reshape$, transforme a sequência S_n em uma matriz, com as mesmas dimensões da imagem encriptada:

$$S_n = reshape(S_n,[Lg, W]). \tag{3.19}$$

Passo 6: A imagem original (I_p) é novamente obtida, aplicando a operação bit-XOR entre a imagem encriptada (I_c) e a chave S_n:

$$I_p(i,j) = S_n(i,j) \oplus I_c(i,j). \tag{3.20}$$

A Fig. 3.5 ilustra como o processo de decriptação é realizado.

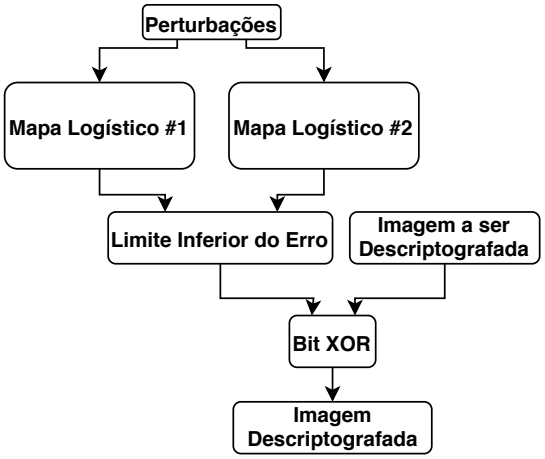


Figura 3.5: Processo de decriptação usando o limite inferior do erro e múltiplas perturbações [2].

3.2.3 Definição dos Parâmetros e Imagens Utilizados

Para gerar a chave secreta da cifra, foram usados os parâmetros aqui definidos: $r = 3.99$, $M = [0.1\ 0.2\ 0.3\ 0.4\ 0.5\ c_i]$, $tr = 700$. Outrossim, as duas extensões intervalares naturais do mapa logístico empregues são definidas pelas Eqs. (3.21) e (3.22):

$$F_1(X_n) = rX_n(1 - X_n) \tag{3.21}$$

$$F_2(Y_n) = rY_n - rY_nY_n. \tag{3.22}$$

As imagens empregadas para validar esta cifra são: Lena (512×512), Babuíno (512×512) e Barbara (512×512). Os fatores c_i das seguintes figuras estão mostrados na Tabela 3.2.

Tabela 3.2: Fatores c_i obtidos a partir das seguintes imagens e de acordo com a Eq. (3.12).

Imagem	**Fator c_i**	**Representação Hexadecimal**
Lena	$4{,}731966182589531 \times 10^{-4}$	`3f3f02ec00000000`
Babuíno	$4{,}997396608814597 \times 10^{-4}$	`3f40601ec0000000`
Barbara	$4{,}478178161662072 \times 10^{-4}$	`3f3d5922e0000000`

3.3 Tabela Comparativa e Testes Realizados

Nesta seção, uma breve tabela (ver Tabela 3.3) compara e resume as especificações técnicas de ambas as cifras. Note que, apesar das duas cifras usarem a limitação numérica presente em computadores, ambas têm suas particularidades. Outrossim, a Tabela 3.3 indica quais os testes

foram feitos para validar os criptossistemas propostos[1]. Vale ressaltar, inclusive, que foi empregado o *software* Matlab para realizar o processo de encriptação e validação dos algoritmos.

Tabela 3.3: Tabela comparativa dos algoritmos propostos nesta dissertação.

Especificações	**Algoritmo I [1]**	**Algoritmo II [2]**
Sistema discreto	–	✓
Sistema contínuo	✓	–
Possui matriz de Perturbação	–	✓
Usa a operação de limite inferior do erro	✓	✓
Tem propriedade de difusão	✓	✓
Tem propriedade de confusão	✓	✓
Imagem inerentemente ligada à cifra	✓	✓
Testes		
NIST SP 800-22 (4.1)	✓	✓
Espaço amostral da chave (4.2)	✓	✓
Análise da sensibilidade a pequenas mudanças à chave (4.3)	✓	✓
Análise da correlação dos *pixels* adjacentes na imagem (4.4)	✓	✓
Entropia da informação (4.5)	✓	✓
Análise de histograma (4.6)	✓	✓
Análise diferencial (4.7)	✓	✓
Tempo e complexidade computacional (4.8)	✓	✓
Resistência à perda de informação (4.9)	✓	✓
Resistência a ataques via imagens simples conhecidas/escolhidas (4.10)	✓	–
Resistência ao ruído (4.11)	✓	–

[1]O número de testes para validar o algoritmo II [2] é menor em relação ao algoritmo proposto em [1]. Isso se deve porque o jornal no qual foi publicado tal criptossistema havia um número limitado de páginas, além de ser um periódico *fast track*.

CAPÍTULO 4

ANÁLISE DE DESEMPENHO

Ao simular duas vezes, ora usando o circuito de Chua, ora usando o mapa logístico com as devidas perturbações e condições iniciais, aplicando a operação do limite inferior do erro em ambas as sequências obtidas, os resultados são mostrados na Fig. 4.1(a). Após o processo de normalização da sequência, vide Eq. (3.5) e Eq. (3.16), as duas séries de valores obtidas, continuam sendo caóticas, conforme mostrado na Fig. 4.1(b). À medida que todos os passos da metodologia são cumpridos, o processo de encriptação e de decriptação de uma imagem é feito. A Fig. 4.2 demonstra os resultados obtidos para os dois criptossistemas usando as imagens de Lena e Barbara.

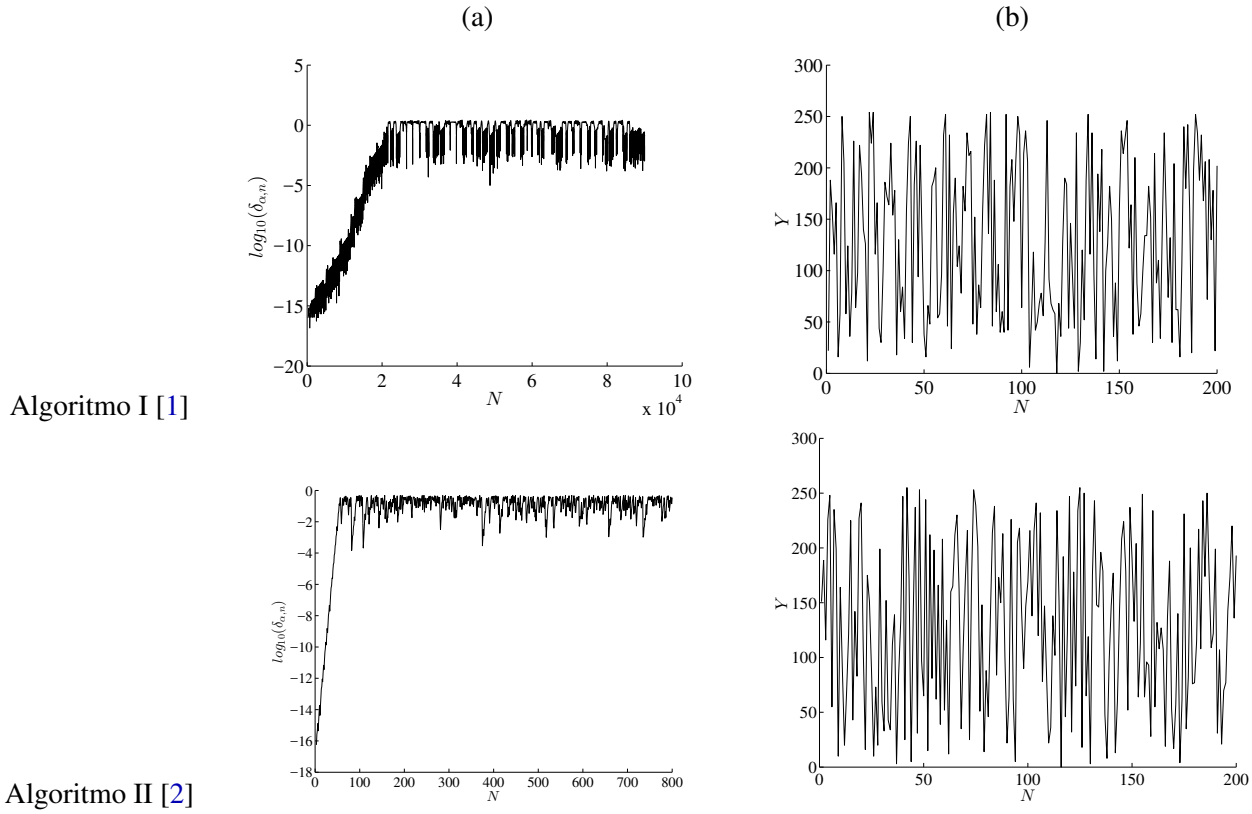


Figura 4.1: (a) Sequências obtidas através da operação do limite inferior do erro. (b) Séries temporais após o processo de normalização. Os resultados indicam que as sequências continuam sendo pseudo-aleatórias ao decorrer de todo o processo criptográfico. N é o número de iterações. *Rotinas: PRNG 1/Entropia e Coeficiente de Correlação* e *PRNG 2/Principal* .

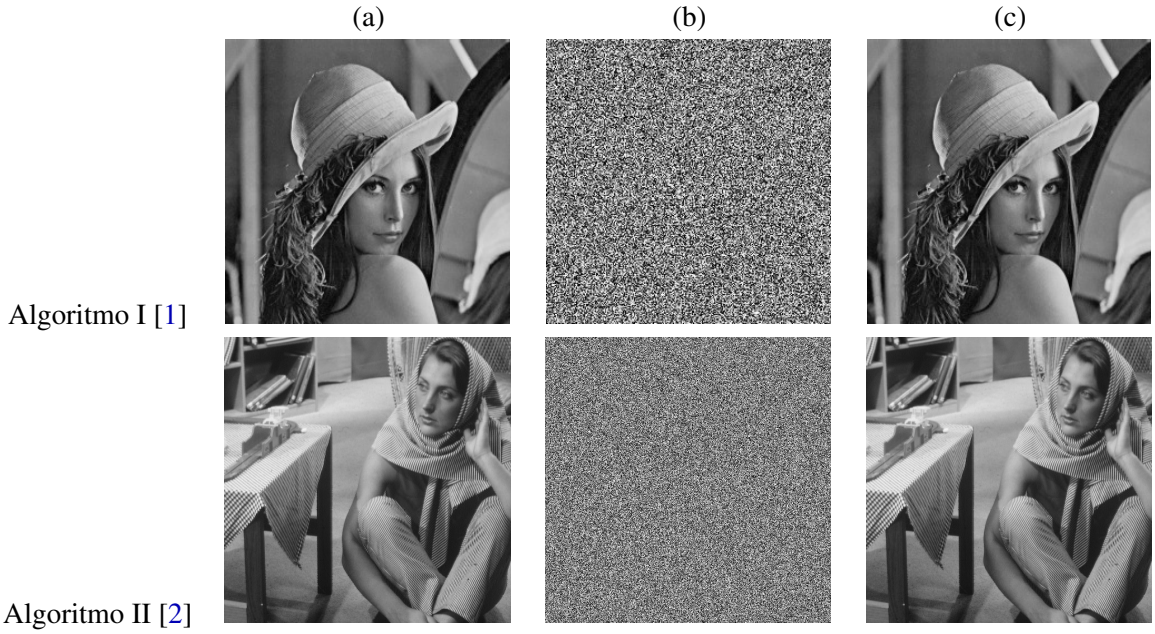


Figura 4.2: Representação de um processo criptográfico baseado na operação bit-XOR. (a) e (c): são imagens originais. (b): imagem cifrada. O processo de encriptação ocorre da imagem (a) para a (b). A decriptação é executada da imagem (b) para (c). *Rotinas: PRNG 1/Entropia e Coeficiente de Correlação* e *PRNG 2/Principal*.

4.1 NIST SP 800-22

O NIST SP 800-22 é um conjunto de 15 testes estatísticos para RNG (Geradores de Números Aleatórios, do inglês *Random Number Generators*) e PRNG (Geradores de Números Pseudo-Aleatórios, do inglês *Pseudo-Random Number Generators*). Tais testes verificam as propriedades aleatórias e/ou pseudo-aleatórias de uma determinada sequência, como a proporção de zeros e uns, uma sucessão contínua de *bits* idênticos, a detecção de periodicidade e/ou padrões, além de dependências lineares [78]. A partir de cada $P - valor$, produzido ao final de cada teste, é possível determinar se a sequência é, ou não, aleatória. O nível de significância α auxilia nessa decisão, se o $P - valor \geq \alpha$, então a sequência passa no teste proposto [13, 78].

A fim de gerar uma sequência e provar a viabilidade do uso do limite inferior do erro, foi feita uma pequena modificação em cada cifra proposta. Para o primeiro algoritmo, a Eq. (3.5) foi modificada por $S_n = uint32(mod(S \times 10^{15}, 2^{32}))$. Por outro lado, para o segundo algoritmo, a Eq. (3.16) foi alterada para $S_n = uint16(mod((S/min(S)) \times 2^{16}, 2^{16}))$. As mudanças foram realizadas, particularmente para estes testes, pois é necessário que haja uma longa sequência de *bits* para que os resultados sejam confiáveis, e como a sequência gerada, após o processo de normalização usado nas cifras, constitui de apenas 256 valores, nitidamente os testes dariam

uma falsa conclusão, por exemplo, de detecção de padrões.

A partir de um comprimento de fluxo de *bits* igual a 1000000 e $\alpha = 0{,}01$, o $P - valor$ de cada teste, em ambos os criptossistemas, é exibido na Tabela 4.1. Note que em todos os 15 testes realizados pelo NIST, o $P - valor \geq \alpha$. Dessa maneira, conclui-se que as sequências geradas são pseudo-aleatórias e que o uso do erro de precisão finita é viável em cifras.

Tabela 4.1: Resultados do $P - valor$ para quinze testes. A viabilidade dos algoritmos I e II são comprovadas, desde que $P - valor \geq \alpha$.

Algoritmo I		
Testes Estatísticos	$P - valor$	Conclusão
Frequência / *Frequency*	0,883171	✓
Frequência dentro de um bloco / *Block Frequency* ($m_b = 128$)	0,236810	✓
Soma cumulativa para frente / *Cusum-Forward*	0,437274	✓
Soma cumulativa para trás / *Cusum-Reverse*	0,437274	✓
Teste de sequência ininterrupta de *bits* idênticos / *Runs*	0,759756	✓
Teste de maior comprimento de uma sequência ininterrupta de *bits* em um bloco / *Long Runs of Ones*	0,759756	✓
Classificação / *Rank*	0,145326	✓
Transformada Discreta de Fourier / *Spectral DFT (Discrete Fourier Transform)*	0,719747	✓
Correspondência de modelos não sobrepostos / *NonOverlapping Templates* ($m_b = 9$, $B_m = 000000001$)	0,554420	✓
Correspondência de modelos sobrepostos / *Overlapping Templates* ($m_b = 9$)	0,595549	✓
Teste Universal Estatístico de Maurer / *Universal*	0,304126	✓
Entropia aproximada / *Approximate Entropy* ($m_b = 10$)	0,867692	✓
Excursões aleatórias / *Random Excursions* ($st = +1$)	0.494392	✓
Excursões aleatórias (variante) / *Random Excursions Variant* ($st = -1$)	0,236810	✓
Complexidade Linear / *Linear Complexity* ($m_b = 500$)	0,534146	✓
Serial / *Serial* ($m_b = 16$, $\bigtriangledown\Psi_m^2$)	0,554420	✓
Algoritmo II		
Testes Estatísticos	$P - valor$	Conclusão
Frequência / *Frequency*	0,935716	✓
Frequência dentro de um bloco / *Block Frequency* ($m_b = 128$)	0,798139	✓
Soma cumulativa para frente / *Cusum-Forward*	0,122325	✓
Soma cumulativa para trás / *Cusum-Reverse*	0,554420	✓
Teste de sequência ininterrupta de *bits* idênticos / *Runs*	0,181557	✓
Teste de maior comprimento de uma sequência ininterrupta de *bits* em um bloco / *Long Runs of Ones*	0,171867	✓
Classificação / *Rank*	0,816537	✓
Transformada Discreta de Fourier / *Spectral DFT (Discrete Fourier Transform)*	0,075719	✓
Correspondência de modelos não sobrepostos / *NonOverlapping Templates* ($m_b = 9$, $B_m = 000000001$)	0,262249	✓
Correspondência de modelos sobrepostos / *Overlapping Templates* ($m_b = 9$)	0,383827	✓
Teste Universal Estatístico de Maurer / *Universal*	0,080519	✓
Entropia aproximada / *Approximate Entropy* ($m_b = 10$)	0,964295	✓
Excursões aleatórias / *Random Excursions* ($st = +1$)	0.155209	✓
Excursões aleatórias (variante) / *Random Excursions Variant* ($st = -1$)	0,970538	✓
Complexidade Linear / *Linear Complexity* ($m_b = 500$)	0,262249	✓
Serial / *Serial* ($m_b = 16$, $\bigtriangledown\Psi_m^2$)	0,616305	✓

4.2 Espaço Amostral da Chave

No criptossistema proposto na subseção 3.1 [1], quatro parâmetros secretos são usados para compor a chave, são eles: as três condições iniciais do circuito de Chua e o fator obtido da imagem, de acordo com a Eq. (3.2). As três condições iniciais são representadas usando a aritmética de ponto flutuante [43], com precisão de $p = 53$ *bits*, o que leva a um espaço amostral de 2^{53^3}. O fator da imagem tem um espaço de $256 \times 256 = 2^{16}$. Portanto, o espaço amostral da chave é $2^{53^3} \times 2^{16} = 2^{175}$, que é maior que 2^{100}, o mínimo sugerido na literatura [16, 27] para garantir robustez contra ataques de força bruta.

De maneira similar, a cifra descrita na subseção 3.2 [2] tem uma condição inicial e quatro valores que perturbam o sistema, o qual dará um espaço amostral de 2^{53^5}, usando aritmética de ponto flutuante. Além disso, considerando um fator, que é baseado na média dos valores dos *pixels* da imagem a ser encriptada, o qual representa um acréscimo ao espaço amostral de $512 \times 512 = 2^{18}$, acarretará em espaço amostral total de 2^{283}. Tal valor é maior que o mínimo requerido na literatura, além de ser maior que o espaço amostral do primeiro algoritmo proposto nessa dissertação.

4.3 Análise da Sensibilidade a Pequenas Mudanças à Chave

Um sistema criptográfico deve ter alta sensibilidade a pequenas mudanças na chave secreta. Tal propriedade é extremamente importante, porque faz com que o usuário, que deseja descriptografar uma imagem, tenha exatamente a mesma chave usada para criptografar tal imagem; para, então, executar com êxito o processo de decriptação.

A análise foi feita de acordo com que é descrito por Yong [20]. Para o primeiro algoritmo [1], cada condição inicial V'_{C1}, V_{C2} e I_L foi perturbada, separadamente, por um fator igual a 10^{-14}. A diferença das imagens cifradas, com e sem perturbação, é quantificada por [20]:

$$\text{Diff}_1(\%) = \frac{1}{Lg \times W} \sum_{i=1}^{Lg} \sum_{j=1}^{W} |sign(I_{c1}(i,j) - I_{c2}(i,j))| \times 100, \tag{4.1}$$

sendo que Lg e W são o comprimento e a largura, respectivamente, das imagens cifradas I_{c1} (sem perturbação) e I_{c2} (com perturbação); $sign(.)$ é a função sinal.

O processo de decriptação foi analisado de maneira similar. A diferença entre duas imagens descriptografadas I_{p1} (sem perturbação) e I_{p2} (com perturbação) é determinada por [20]:

$$\mathrm{Diff}_2(\%) = \frac{1}{Lg \times W} \sum_{i=1}^{Lg} \sum_{j=1}^{W} |sign(I_{p1}(i,j) - I_{p2}(i,j))| \times 100. \tag{4.2}$$

A Tabela 4.2 exibe os resultados da sensibilidade da chave na encriptação e decriptação. O processo é altamente sensível às mudanças na chave secreta; as diferenças, para todas as imagens testadas são próximas de 100%. Inclusive, de maneira ilustrativa, a Fig. 4.3 mostra as imagens obtidas via decriptação, usando chaves corretas e incorretas.

Tabela 4.2: Resultados da análise da sensibilidade no processo de encriptação e decriptação para diversas imagens de referência. Uma pequena perturbação foi adicionada em cada estado do circuito de Chua. A terceira e quarta colunas apresentam os valores calculados de acordo com as Eq. (4.1) e (4.2) [20]. Valores próximos de 100%, indicam imagens obtidas completamente diferentes. *Rotinas: PRNG 1/Análise da Sensibilidade*.

Imagem	Chave Secreta	$\mathrm{Diff}_1(\%)$	$\mathrm{Diff}_2(\%)$
Lena	$V'_{C_1} + 10^{-14}$	99,64	99,64
	$V_{C_2} + 10^{-14}$	99,62	99,62
	$I_L + 10^{-14}$	99,59	99,59
Barco	$V'_{C_1} + 10^{-14}$	99,63	99,63
	$V_{C_2} + 10^{-14}$	99,61	99,61
	$I_L + 10^{-14}$	99,59	99,59
Casa	$V'_{C_1} + 10^{-14}$	99,62	99,62
	$V_{C_2} + 10^{-14}$	99,54	99,54
	$I_L + 10^{-14}$	99,59	99,59
Pimenta	$V'_{C_1} + 10^{-14}$	99,55	99,55
	$V_{C_2} + 10^{-14}$	99,60	99,60
	$I_L + 10^{-14}$	99,58	99,58
Cameraman	$V'_{C_1} + 10^{-14}$	99,59	99,59
	$V_{C_2} + 10^{-14}$	99,61	99,61
	$I_L + 10^{-14}$	99,61	99,61
Preto	$V'_{C_1} + 10^{-14}$	99,61	99,61
	$V_{C_2} + 10^{-14}$	99,60	99,60
	$I_L + 10^{-14}$	99,60	99,60
Branco	$V'_{C_1} + 10^{-14}$	99,62	99,62
	$V_{C_2} + 10^{-14}$	99,60	99,60
	$I_L + 10^{-14}$	99,60	99,60

Do mesmo modo, para o segundo algoritmo [2], um fator igual a 10^{-14} foi adicionado em cada elemento do vetor M (Eq. (3.11)). A partir das equações Eq. (4.1) e Eq. (4.2), foi computada a diferença das imagens e os resultados estão mostrados na Tabela 4.3. Como esperado, devido à mudança da chave, o resultado é completamente diferente, com valores de $\mathrm{Diff}_1(\%)$ e $\mathrm{Diff}_2(\%)$ próximos de 100%.

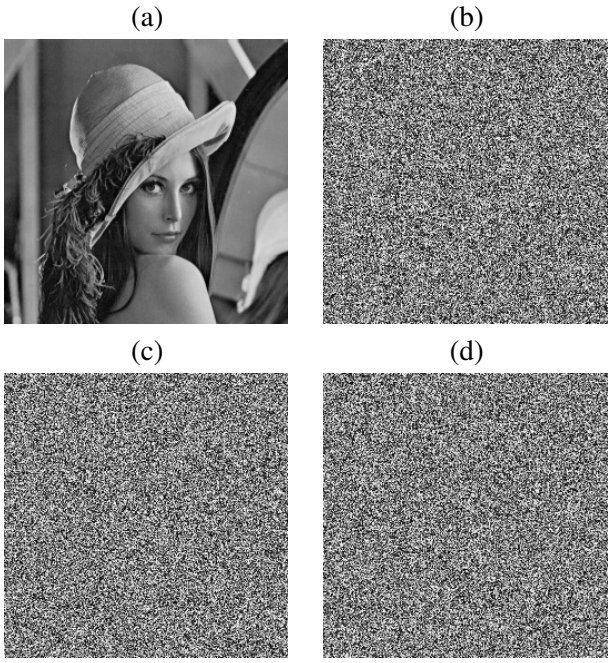


Figura 4.3: Resultados obtidos, com a imagem de Lena via decriptação, usando as seguintes chaves secretas: (a) Correta; (b) $V'_{C_1} + 10^{-14}$; (c) $V_{C_2} + 10^{-14}$; (d) $I_L + 10^{-14}$. *Rotinas: PRNG 1/Análise da Sensibilidade*.

Tabela 4.3: Resultados da análise de sensibilidade da chave para três diferentes imagens. Valores de $\mathrm{Diff}_1(\%)$ e $\mathrm{Diff}_2(\%)$, próximos a 100%, indicam imagens completamente diferentes. *Rotinas: PRNG 2/Análise da Sensibilidade*.

Imagem	Chave Secreta	$\mathrm{Diff}_1(\%)$	$\mathrm{Diff}_2(\%)$
Lena	$0{,}1 + 10^{-14}$	99,59	99,59
	$0{,}2 + 10^{-14}$	99,62	99,62
	$0{,}3 + 10^{-14}$	99,62	99,62
	$0{,}4 + 10^{-14}$	99,61	99,61
	$0{,}5 + 10^{-14}$	99,59	99,59
	$c_i + 10^{-14}$	99,61	99,61
Babuíno	$0{,}1 + 10^{-14}$	99,62	99,62
	$0{,}2 + 10^{-14}$	99,63	99,63
	$0{,}3 + 10^{-14}$	99,60	99,60
	$0{,}4 + 10^{-14}$	99,63	99,63
	$0{,}5 + 10^{-14}$	99,62	99,62
	$c_i + 10^{-14}$	99,59	99,59
Barbara	$0{,}1 + 10^{-14}$	99,61	99,61
	$0{,}2 + 10^{-14}$	99,61	99,61
	$0{,}3 + 10^{-14}$	99,62	99,62
	$0{,}4 + 10^{-14}$	99,62	99,62
	$0{,}5 + 10^{-14}$	99,60	99,60
	$c_i + 10^{-14}$	99,60	99,60

4.4 Análise da Correlação dos *Pixels* Adjacentes na Imagem

Os *pixels* adjacentes são fortemente correlacionados entre si em imagens. Portanto, *hackers* frequentemente buscam quebrar um criptossistema analisando as informações de correlação [27, 79]. Em uma imagem cifrada, é esperado que o coeficiente de correlação seja próximo de zero nas direções horizontal, vertical e diagonal, para evitar tais ataques. O coeficiente de correlação dos *pixels* adjacentes é medido pela Eq. (4.3) [79]:

$$\rho(A,B) = \frac{E[(A-\mu_A)(B-\mu_B)]}{\sigma_A \sigma_B}, \tag{4.3}$$

sendo que A representa uma série de *pixels* em uma determinada posição, B é a série dos *pixels* adjacentes, μ e σ são a média e o desvio padrão, e $E[.]$ é a esperança matemática.

Analisando o algoritmo I [1], a Tabela 4.4 mostra os coeficientes de correlação para diferentes figuras. Note que as imagens originais têm um alto coeficiente, indicando que os *pixels* estão fortemente correlacionados, enquanto as imagens cifradas, não. O que foi supracitado pode ser reafirmado ao observar as Fig. 4.4 e 4.5.

Tabela 4.4: Teste de coeficientes de correlação para as sete imagens de referência usadas na cifra [1]. É mostrada a correlação para cada imagem, a original e a criptografada. As imagens criptografadas exibem valores muito próximos de zero, o que é esperado para algoritmos criptográficos robustos. *Rotinas: PRNG 1/Entropia e Coeficiente de Correlação.*

Imagem		Coeficiente de Correlação		
		Horizontal	Vertical	Diagonal
Lena	Original	0,93998	0,96934	0,91793
	Cifrada	0,00405	0,00302	0,00113
Barco	Original	0,92066	0,93714	0,88052
	Cifrada	0,00842	0,00187	0,00136
Casa	Original	0,97807	0,96528	0,94835
	Cifrada	−0,00426	0,00561	−0,00259
Pimenta	Original	0,95223	0,95303	0,90949
	Cifrada	0,00130	−0,00159	0,00354
Cameraman	Original	0,93321	0,95928	0,90764
	Cifrada	−0,00089	−0,00096	−0,00084
Preto	Original	–	–	–
	Cifrada	−0,00091	−0,00177	−0,00043
Branco	Original	–	–	–
	Cifrada	−0,00161	0,00040	−0,00066

Similarmente, foram computados os coeficientes de correlação para as imagens de Lena, Babuíno e Barbara usando a cifra proposta na subseção 3.2 [2]. Da mesma forma, quando o

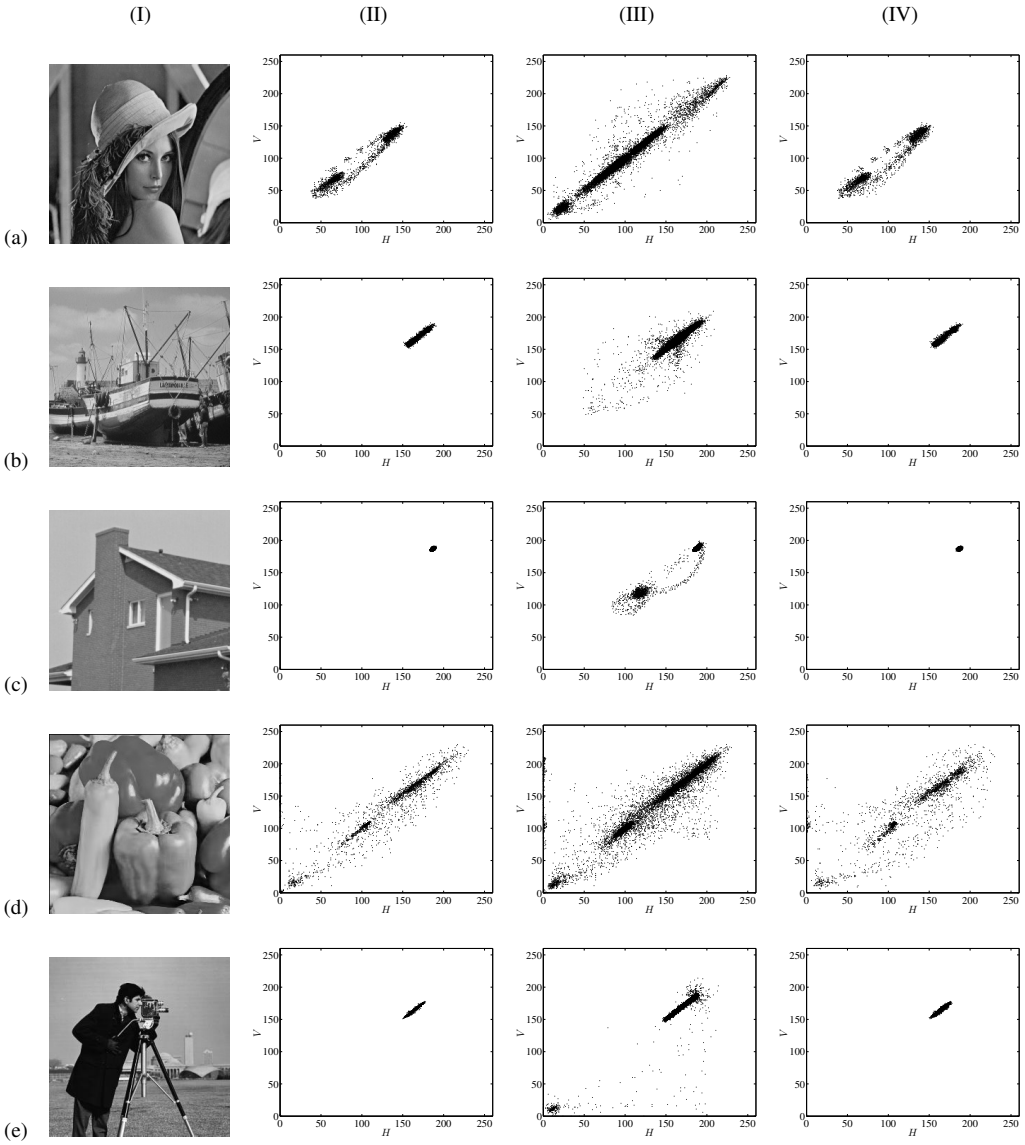


Figura 4.4: Distribuição de correlação de dois *pixels* adjacentes para o criptossistema proposto em [1]. A primeira coluna é para as imagens originais. As outras três colunas restantes são para os três tipos de correlação: *pixels* horizontalmente, verticalmente e diagonalmente adjacentes, respectivamente. O eixo representa a localização do valor do *pixel* acinzentado. *Rotinas: PRNG 1/Entropia e Coeficiente de Correlação*.

Tabela 4.5: Valores de coeficiente de correlação para as imagens originais e as cifradas usando a cifra indicada em [2]. Após o processo de encriptação os *pixels* das imagens deixam de ser altamente correlacionadas, e apresentam valores próximos de zero. *Rotinas: PRNG 2/Principal*.

Imagem		Coeficiente de Correlação		
		Horizontal	Vertical	Diagonal
Lena	Original	0,97323	0,98648	0,96031
	Cifrada	−0,00251	−0,00292	−0,00156
Babuíno	Original	0,86998	0,82920	0,80018
	Cifrada	−0,00180	0,00066	−0,00128
Barbara	Original	0,89539	0,95887	0,88304
	Cifrada	0,00011	0,00346	0,00184

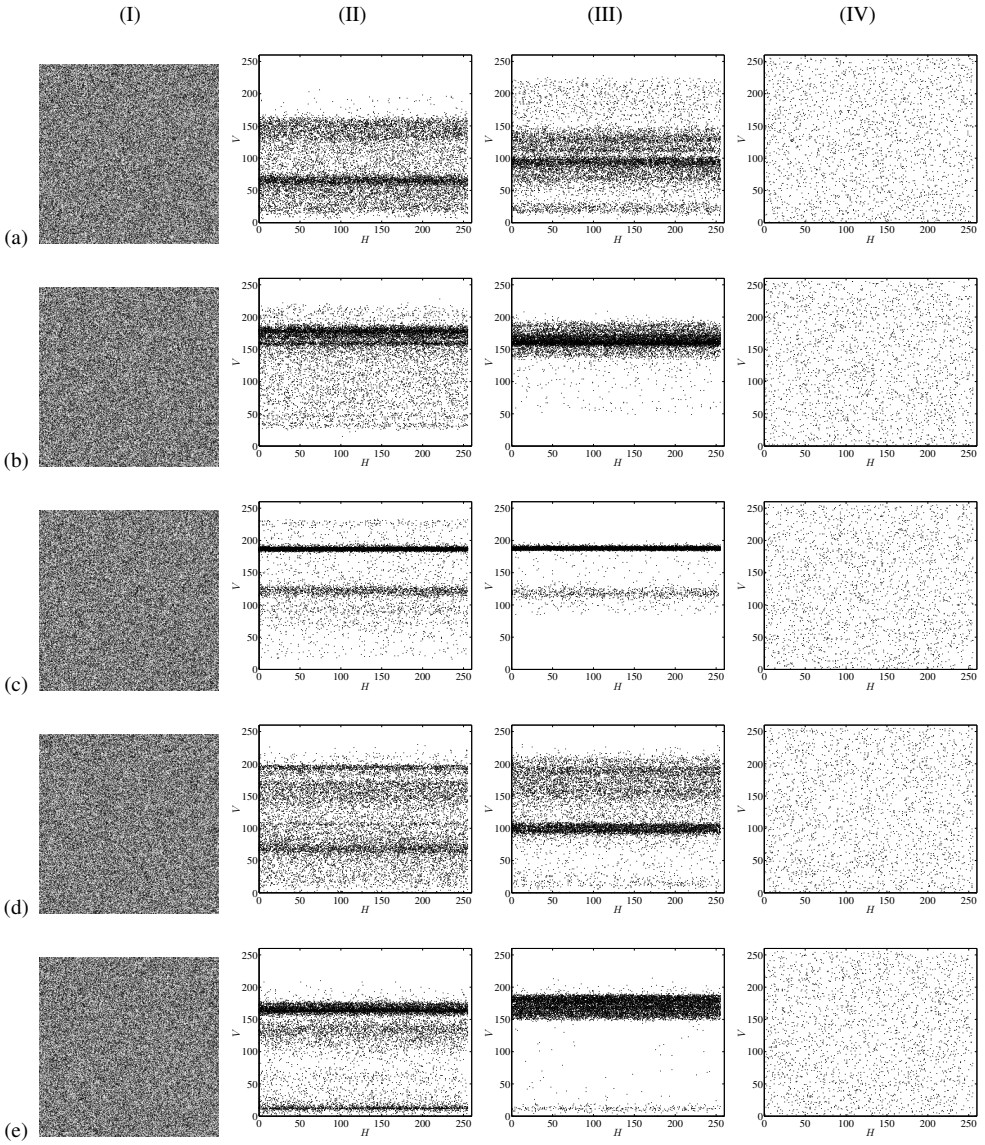


Figura 4.5: Distribuição de correlação de dois *pixels* adjacentes para o criptossistema proposto em [1]. A primeira coluna é para as imagens encriptadas, sendo elas: Lena, Barco, Casa, Pimenta e Cameraman, respectivamente. As outras três colunas restantes são para os três tipos de correlação: II): *pixels* horizontalmente adjacentes com $H = (1:50, 1:49)$ e $V = (1:50, 2:50)$. III) *pixels* verticalmente adjacentes com $H = (1:49, 1:50)$ e $V = (2:50, 1:50)$. IV) *pixels* diagonalmente adjacentes com $H = (1:49, 1:49)$ e $V = (2:50, 2:50)$. O eixo representa a localização do valor do *pixel* acinzentado. Note-se a baixa correlação dos *pixels* adjacentes nas imagens cifradas. Veja mais detalhes em [80]. *Rotinas: PRNG 1/Entropia e Coeficiente de Correlação*.

processo de encriptação ocorre, os *pixels* da imagem cifrada não estão altamente correlacionamos. A Tabela 4.5 e a Fig. 4.6 reafirmam o que foi referido acima.

Ademais, a Tabela 4.6 compara os coeficientes obtidos com a imagem de Lena para ambos os algoritmos propostos, além de diferentes trabalhos encontrados na literatura. Apesar dos coeficientes de correlação das cifras não serem os mais baixos, os valores calculados são próximos de zero.

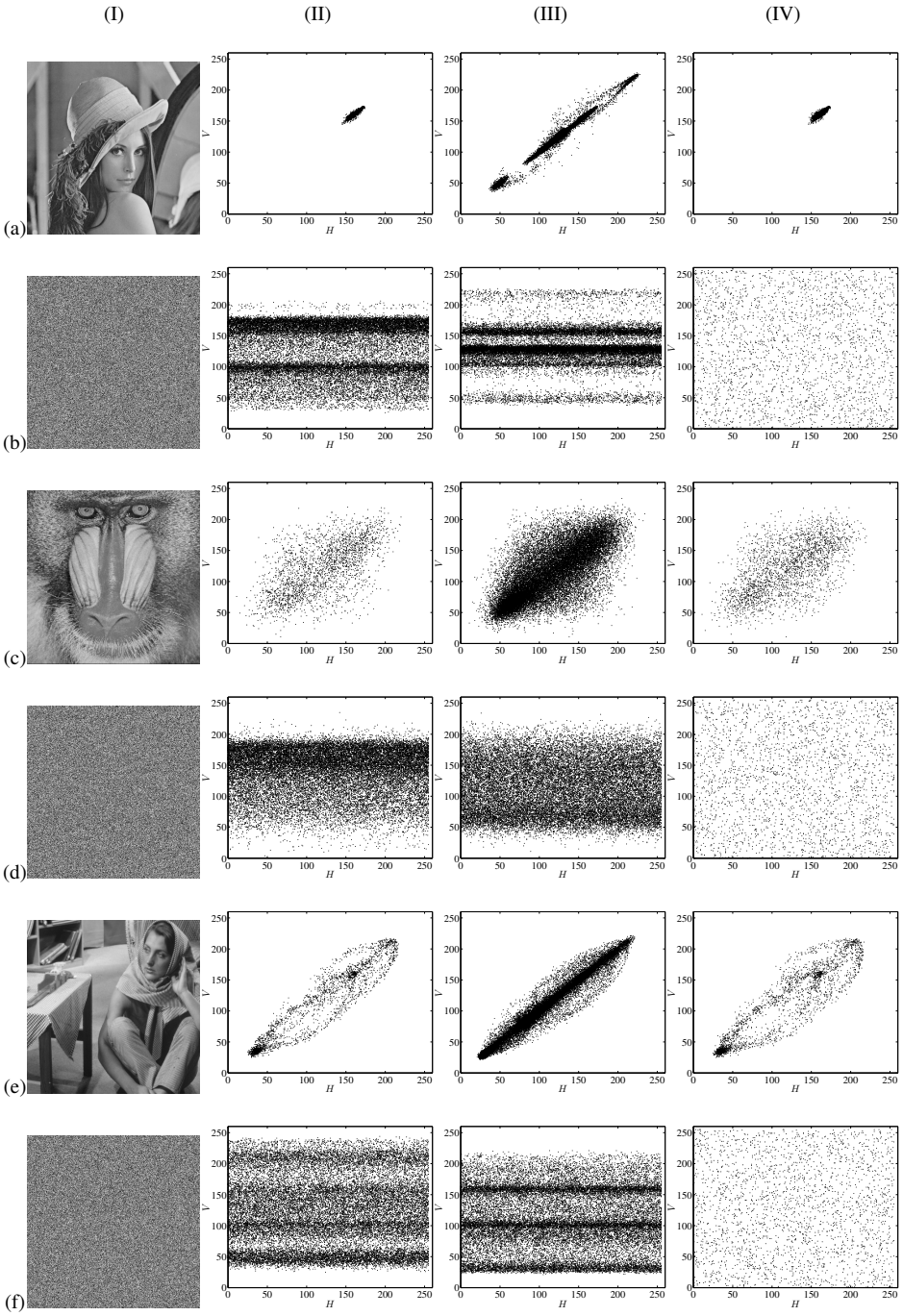


Figura 4.6: Distribuição de correlação de dois *pixels* adjacentes para as imagens de Lena, Babuíno e Barbara, junto com as respectivas imagens encriptadas, quando usado o criptossistema [2]. A primeira coluna é para as imagens originais, juntamente com as respectivas imagens cifradas. As outras três colunas restantes são para os três tipos de correlação: *pixels* horizontalmente, verticalmente e diagonalmente adjacentes, respectivamente. O eixo representa a localização do valor do *pixel* acinzentado. *Rotinas: PRNG 2/Principal*.

4.5 Entropia da Informação

A entropia de Shannon [86] é uma métrica para medir a aleatoriedade de um determinado sistema de comunicação, definido pela Eq. (4.4) [17]:

Tabela 4.6: Comparação dos coeficientes de correlação da imagem cifrada de Lena. Apesar dos coeficientes de correlação da cifra proposta não serem os mais baixos, os valores calculados são próximos de zero.

Algoritmo I [1]				Algoritmo II [2]			
Coeficiente de Correlação			Lena (256×256)	Coeficiente de Correlação			Lena (512×512)
Horizontal	Vertical	Diagonal		Horizontal	Vertical	Diagonal	
0,00405	0,00302	0,00113	Proposto	−0,00251	−0,00292	−0,00156	Proposto
0,00083	0,00223	0,00650	[17]	−0,00210	−0,01619	0,01781	[81]
0,00352	0,00649	0,00356	[82]	0,03211	0,02719	0,03839	[83]
0,00160	0,00250	0,00030	[26]	−0,00450	−0,00016	0,00530	[84]
−0,00170	0,00130	−0,00050	[27]	0,00171	−0,00217	−0,00087	[85]
0,93998	0,96934	0,91793	Original	0,97323	0,98648	0,96031	Original

$$H(sym) = \sum_{i=0}^{G_L-1} P(sym_i) log_2 \frac{1}{P(sym_i)}, \tag{4.4}$$

sendo $H(sym)$ o valor da entropia, dado em bits, sym é um símbolo, $P(sym_i)$ é o valor da probabilidade do símbolo sym e $G_L = 256$ é o nível de cinza.

O valor teórico para uma imagem encriptada é $H = log_2(256) = 8$ [87]. Como as imagens cifradas são geradas a partir de PRNG, estas não são verdadeiramente aleatórias. Dessa forma, o valor esperado de entropia, para que o sistema seja considerado seguro, é $H(X) \approx 8$. A Tabela 4.7 exibe os valores de entropia para todas as imagens testadas, usando ambas as cifras propostas. Outrossim, a partir das duas cifras, a Tabela 4.8 compara o valor de entropia obtido através da imagem encriptada de Lena com outros valores encontrados na literatura. Tendo em vista que os valores de entropia das cifras são próximos de 8 e, também, similares aos resultados de outros trabalhos, pode-se concluir que os métodos são seguros.

Tabela 4.7: Entropia de informação para diferentes imagens. *Rotinas: PRNG 1/Entropia e Coeficiente de Correlação* e *PRNG 2/Principal*.

Algoritmo	Imagem Encriptada	Entropia
Algoritmo I [1]	Lena (256×256)	7,9968
	Barco (256×256)	7,9971
	Casa (256×256)	7,9969
	Pimenta (256×256)	7,9975
	Cameraman (256×256)	7,9969
	Preto (512×512)	7,9975
	Branco (512×512)	7,9977
Algoritmo II [2]	Lena (512×512)	7,9993
	Babuíno (512×512)	7,9993
	Barbara (512×512)	7,9993

Tabela 4.8: Entropia de informação da imagem encriptada de Lena para diferentes abordagens. A entropia de informação ideal é 8.

Algoritmo I [1]		Algoritmo II [2]	
Entropia	Referências	Entropia	Referências
7,9968	Proposto	7,9993	Proposto
7,9826	[17]	–	[81]
7,9980	[82]	7,9983	[83]
7,9998	[26]	7,9993	[84]
7,9975	[27]	7,9993	[85]

4.6 Análise de Histograma

O histograma é uma importante ferramenta para comprovar a eficácia de um criptossistema. A distribuição de frequências ilustra como os *pixels* presentes nas imagens são distribuídos. Nesse sentido, quando é realizada a encriptação, o histograma da imagem cifrada deve ser uniforme. Em outras palavras, as frequências dos *pixels* devem ser aproximadamente iguais para todas as intensidades de cores. Dessa forma, a imagem encriptada não traz nenhuma informação relevante, e o *hacker* não consegue obter nenhum dado da imagem [10, 16, 18, 20, 27, 88].

Uma análise quantitativa do histograma pode ser realizada por [10, 20]:

$$Var(h) = \frac{1}{G_L^2} \sum_{i=0}^{G_L-1} \sum_{j=0}^{G_L-1} \frac{1}{2}(h_i - h_j)^2, \tag{4.5}$$

sendo $Var(h)$ o valor de variância, $G_L = 256$ é o nível de cinza e h é o vetor dos valores do histograma.

As Fig. 4.7 e Fig. 4.8 mostram as imagens originais e as cifradas, juntas com seus histogramas correspondentes, usando o algoritmo I [1] e II [2], respectivamente. Embora as imagens, antes de passarem pelo processo de encriptação, exibam uma alta presença de determinados valores de tons cinza, está claro que o esquema proposto consegue converter qualquer imagem em uma semelhante a ruído, com distribuição uniforme de *pixels*. A Tabela 4.9 revela que a variância da imagem encriptada é significantemente reduzida. Em todas as imagens, os algoritmos propostos conseguiram reduzir as variâncias das imagens originais em mais de 99%. Comparando, Namasudra e Deka [89] reduziram a variância, entre a imagem original e a cifrada, usando a imagem de Lena, em $99{,}089\%$; enquanto que os propostos nessa dissertação obtiveram valores próximos de $99{,}055\%$ e $99{,}84\%$ para o primeiro e segundo algoritmo, respectivamente.

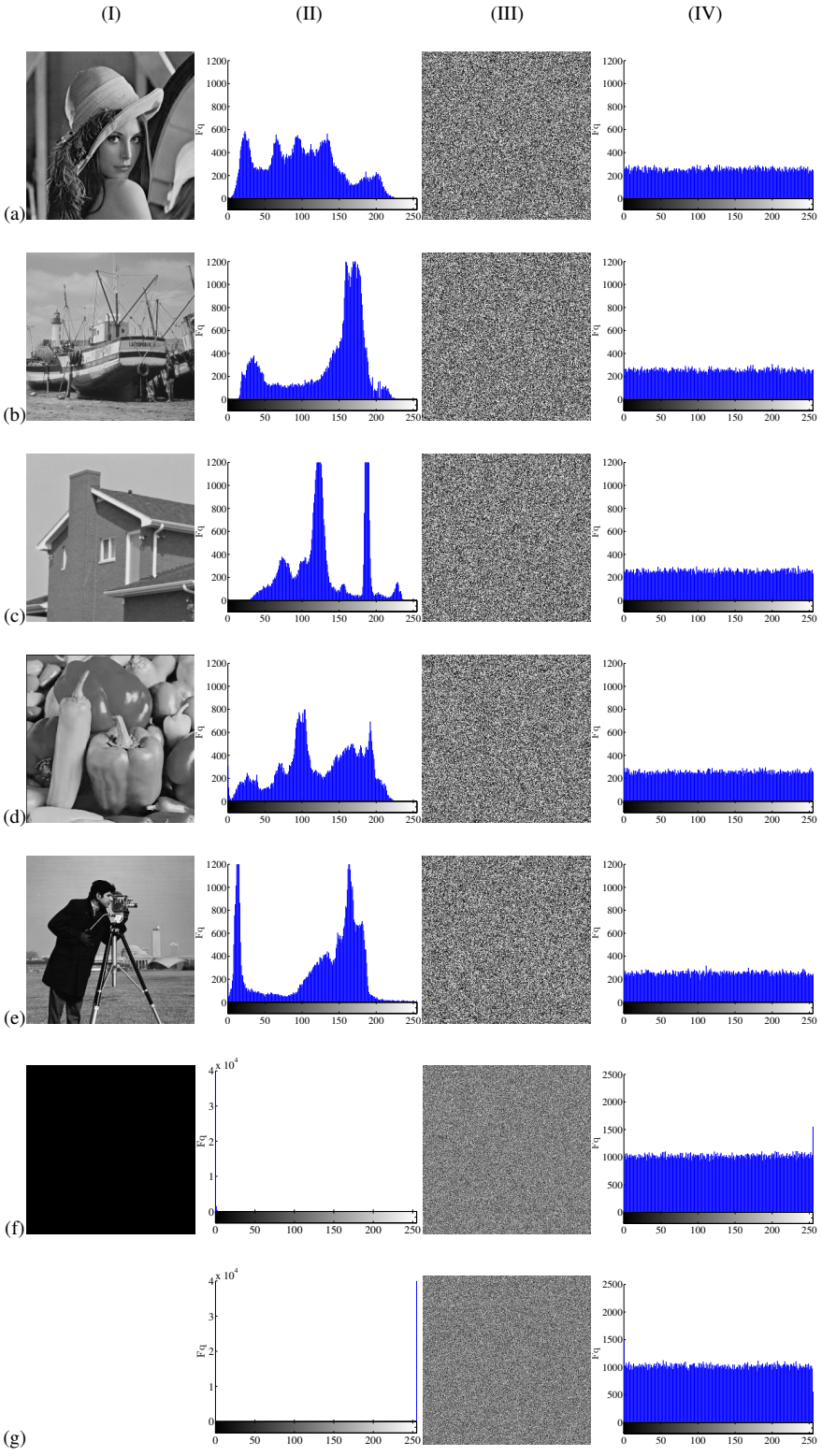


Figura 4.7: Histograma para sete imagens, antes e depois da realização da encriptação, usando a cifra [1]. Esta figura está organizada em quatro colunas, descritas a seguir: (I) imagem original; (II) histograma da respectiva imagem; (III) imagem encriptada; (IV) histograma da imagem cifrada. Cada linha representa uma imagem diferente, como: (a) Lena; (b) Barco; (c) Casa; (d) Pimenta; (e) Cameraman; (f) Preto; (g) Branco. Para as imagens totalmente pretas e brancas, no histograma a um pico da frequência (Fq) ora em 0, ora em 255, para as respectivas imagens. *Rotinas: PRNG 1/Histograma.*

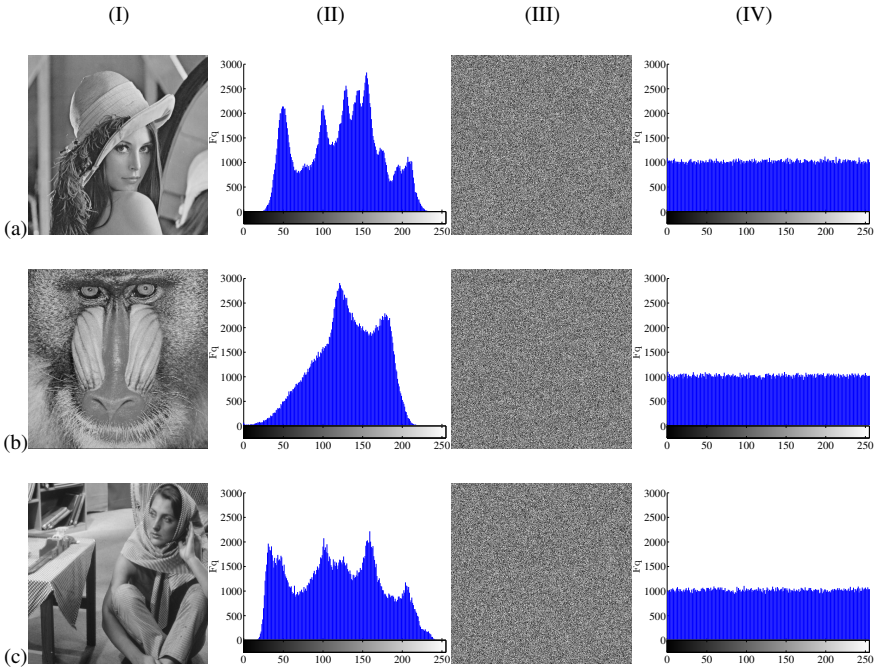


Figura 4.8: Distribuição de frequência de cores para as imagens de (a) Lena, (b) Babuíno e (c) Barbara, antes e depois da realização da encriptação, usando a cifra [2]. A figura está organizada em quatro colunas, descritas a seguir: (I) imagem a ser criptografada; (II) histograma da respectiva imagem; (III) imagem encriptada; (IV) histograma da imagem cifrada. Fq representa a frequência que determinada intensidade de cor aparece na imagem. *Rotinas: PRNG 2/Principal*.

Tabela 4.9: Variância dos histogramas para as respectivas imagens e cifras. A terceira coluna exibe a porcentagem de redução da variância entre a imagem encriptada e a imagem original. Como forma de comparação, Namasudra e Deka [89] reduziram a variância do histograma, para a imagem de Lena, em 99,089%. *Rotinas: PRNG 1/Histograma* e *PRNG 2/Principal* .

Algoritmo	Imagem	Variância		
		Original	Cifrada	Redução (%)
Algoritmo I [1]	Lena (256×256)	30697,616	290,157	99,055
	Barco (256×256)	99221,152	262,596	99,735
	Casa (256×256)	300964,870	276,760	99,908
	Pimenta (256×256)	37241,419	227,012	99,390
	Cameraman (256×256)	99630,761	283,522	99,715
	Preto (256×256)	$2{,}64 \times 10^8$	$3{,}56 \times 10^3$	99,999
	Branco (256×256)	$2{,}68 \times 10^8$	$3{,}24 \times 10^3$	99,999
Algoritmo II [2]	Lena (512×512)	643270.56	1000,18	99,845
	Babuíno (512×512)	847461,35	990,11	99,883
	Barbara (512×512)	383694,31	963,07	99,749

4.7 Análise Diferencial

Hackers buscam encontrar padrões em imagens encriptadas como forma de quebrar o algoritmo e identificar a imagem original [10, 16, 27]. O número de alterações na taxa de *pixels* (NPCR, do inglês *Number of Changing Pixel Rate*) e a intensidade de mudança média unificada

(UACI, do inglês *Unified Averaged Changed Intensity*) são as duas métricas mais usadas para avaliar a resistência da cifra contra ataques diferenciais [15, 19]. Os resultados de NPCR e UACI podem ser obtidos pelas seguintes equações [20]:

$$NPCR(\%) = \frac{1}{Lg \times W} \sum_{i=1}^{Lg} \sum_{j=1}^{W} |sign(I_{c1}(i,j) - I_{c2}(i,j))| \times 100, \tag{4.6}$$

$$UACI(\%) = \frac{1}{Lg \times W} \sum_{i=1}^{Lg} \sum_{j=1}^{W} \frac{|I_{c1}(i,j) - I_{c2}(i,j)|}{255} \times 100, \tag{4.7}$$

sendo que Lg e W são o comprimento e a largura, respectivamente, das imagens cifradas I_{ci}, e $sign()$ é a função sinal.

Wu et al. [30] estabeleceram pontuações como variáveis aleatórias e derivaram suas médias e variações, fornecendo o valor crítico para o índice NPCR e os valores intervalares aceitos para a métrica UACI, para uma variedade de tamanhos de imagem. Para realizar o teste [10, 16, 18], primeiramente é encriptada uma imagem qualquer. Em seguida, um *pixel* da imagem é aleatoriamente escolhido e o seu valor é modificado. A partir da imagem modificada, o processo de encriptação é realizado novamente. Com a obtenção de duas imagens cifradas, as métricas NPCR e UACI podem ser obtidas via Eqs. (4.6) e (4.7). Os resultados alcançados estão mostrados na Tabela 4.10 e na Tabela 4.11 para as respectivas cifras propostas. Pode ser observado que tais valores respeitam os limites descritos por Wu et al. [30]. Portanto, ambas as cifras resistem a ataques diferenciais. A Tabela 4.12 compara os valores de NPCR e UACI com outros encontradas na literatura. Note, também, que duas referências usadas, até então para comparar com os resultados obtidos nessa dissertação [17] e [26] não apresentam valores para essas duas métricas, indicando que as cifras propostas por esses dois trabalhos podem ser vulneráveis a ataques diferenciais.

4.8 Tempo e Complexidade Computacional

Os algoritmos devem encriptar e decriptar uma imagem eficientemente [16]. A fim de analisar a complexidade computacional dos criptossistemas, deve-se contar as operações matemáticas realizadas ao executar o processo de encriptação, assim como feito em [27] e [16]. A Tabela 4.13 mostra em detalhes quantas e quais operações ocorreram em todo o processo das cifras. Para uma imagem, de tamanho $Lg \times W$, o número total de operações é $78(Lg \times W) + 75tr + 32$

Tabela 4.10: Resultados dos valores de NPCR e UACI para o algoritmo proposto em [1]. *Rotinas: PRNG 1/NPCR-UACI.*

Imagem	Valor do NPCR (%)	Valores críticos de NPCR (%)		
		$\mathcal{N}^{*}_{0,05} = 99{,}57\%$	$\mathcal{N}^{*}_{0,01} = 99{,}55\%$	$\mathcal{N}^{*}_{0,001} = 99{,}53\%$
Lena	99,57%	✓	✓	✓
Barco	99,59%	✓	✓	✓
Casa	99,57%	✓	✓	✓
Pimenta	99,60%	✓	✓	✓
Cameraman	99,58%	✓	✓	✓
Preto	99,59%	✓	✓	✓
Branco	99,62%	✓	✓	✓
	Valor do UACI (%)	Valores críticos do UACI (%)		
		$\mathcal{U}^{*-}_{0.05} = 33{,}28\%$	$\mathcal{U}^{*-}_{0.01} = 33{,}23\%$	$\mathcal{U}^{*-}_{0.001} = 33{,}16\%$
		$\mathcal{U}^{*+}_{0.05} = 33{,}64\%$	$\mathcal{U}^{*+}_{0.01} = 33{,}70\%$	$\mathcal{U}^{*+}_{0.001} = 33{,}77\%$
Lena	33,41%	✓	✓	✓
Barco	33,44%	✓	✓	✓
Casa	33,55%	✓	✓	✓
Pimenta	33,40%	✓	✓	✓
Cameraman	33,30%	✓	✓	✓
Preto	33,52%	✓	✓	✓
Branco	33,44%	✓	✓	✓

Tabela 4.11: Resultados quantitativos para análise de resistência a ataques diferenciais, usando o algoritmo descrito em [2]. *Rotinas: PRNG 2/NPCR-UACI.*

Imagem	Valor do NPCR (%)	Valores críticos de NPCR (%)		
		$\mathcal{N}^{*}_{0,05} = 99{,}59\%$	$\mathcal{N}^{*}_{0,01} = 99{,}58\%$	$\mathcal{N}^{*}_{0,001} = 99{,}57\%$
Lena	99,61%	✓	✓	✓
Babuíno	99,61%	✓	✓	✓
Barbara	99,60%	✓	✓	✓
	Valor do UACI (%)	Valores críticos do UACI		
		$\mathcal{U}^{*-}_{0,05} = 33{,}37\%$	$\mathcal{U}^{*-}_{0,01} = 33{,}34\%$	$\mathcal{U}^{*-}_{0,001} = 33{,}31\%$
		$\mathcal{U}^{*+}_{0,05} = 33{,}55\%$	$\mathcal{U}^{*+}_{0,01} = 33{,}58\%$	$\mathcal{U}^{*+}_{0,001} = 33{,}62\%$
Lena	33,46%	✓	✓	✓
Babuíno	33,48%	✓	✓	✓
Barbara	33,43%	✓	✓	✓

Tabela 4.12: Resultados obtidos com as métricas NPCR e UACI para a imagem de Lena e as suas comparações com outros valores encontrados na literatura.

Algoritmo I [1]			Algoritmo II [2]		
Valor do NPCR	Valor do UACI	Referências	Valor do NPCR	Valor do UACI	Referências
99,57%	33,41%	Proposto	99,61%	33,46%	Proposto
———	———	[17]	99,62%	33,48%	[81]
99.57%	33.48%	[82]	99,62%	33,49%	[83]
———	———	[26]	99,59%	33,41%	[84]
99.61%	33.46%	[27]	99.61%	33.46%	[85]

para o algoritmo I [1] e $16(Lg \times W) + 7tr + 29$ para o algoritmo II [2], os quais fornecem uma complexidade computacional de $O(Lg \times W)$.

Tabela 4.13: Complexidade computacional. O Lg é o comprimento e W é a largura da imagem, e tr é o transiente do sistema caótico. As cifras propostas apresentam complexidade computacional de $O(Lg \times W)$, o mesmo apresentado, por exemplo, em [16].

Algoritmo I [1]		Algoritmo II [2]	
Operações	Processo de encriptação	Operações	Processo de encriptação
Soma/Subtração	$32(Lg \times W) + 30tr + 3$	Soma/Subtração	$4(Lg \times W) + 2tr + 13$
Multiplicação/Divisão	$37(Lg \times W) + 37tr + 14$	Multiplicação/Divisão	$8(Lg \times W) + 5tr + 3$
Potência	15	Potência	1
Absoluto	$5(Lg \times W) + 5tr$	Absoluto	$Lg \times W$
Logaritmo	$Lg \times W + tr$	Logaritmo	$----$
Módulo	$Lg \times W + tr$	Módulo	$Lg \times W + 12$
Uint8	$Lg \times W + tr$	Uint8	$Lg \times W$
bit-XOR	$Lg \times W$	bit-XOR	$Lg \times W$
Total de Operações	$78(Lg \times W) + 75tr + 32$	Total de Operações	$16(Lg \times W) + 7tr + 29$

Para analisar o tempo de encriptação frente a diferentes tamanhos de imagem é usada a ferramenta *Matlab Profile*. O resultado é apresentado na Tabela 4.14. Como pode ser observado, o criptossistema II [2] apresenta um tempo de resposta relativamente menor, se comparado ao criptossistema I[1], devido ao número reduzido de operações que o algoritmo II tem em relação ao algoritmo I. Entretanto, ambas as cifras propostas são superiores aos métodos tradicionais, como o AES, uma vez que esses são lentos ao criptografar imagens e vídeos [25].

Tabela 4.14: Tempo de encriptação para diferentes tamanhos, usando a imagem de Lena. *Rotinas: Tempo de Encriptação*.

Lena ($Lg \times W$)	Algoritmo I [1]	Algoritmo II [2]
	Tempo de Encriptação (s)	Tempo de Encriptação (s)
128×128	4,396	0,602
256×256	8,996	0,743
512×512	34,688	1,101
1024×1024	214,113	2,811

4.9 Resistência à Perda de Informação

Um criptossistema, para ser dito eficiente, deve considerar a perda de informação [5, 10]. Com base no criptossistema I [1], a Fig. 4.9 mostra blocos removidos da imagem criptografada de tamanhos 16×16, 32×32 e 64×64. Após o processo de decriptação, as imagens descriptografas continuam a ser visualmente significativas. Portanto, o método é robusto contra esse tipo de ataque.

Contudo, como modo de aumentar a robustez contra à perda de informação, foi acrescentado, para o criptossistema II [2], mais um passo na decriptação da imagem quando ocorre o tal

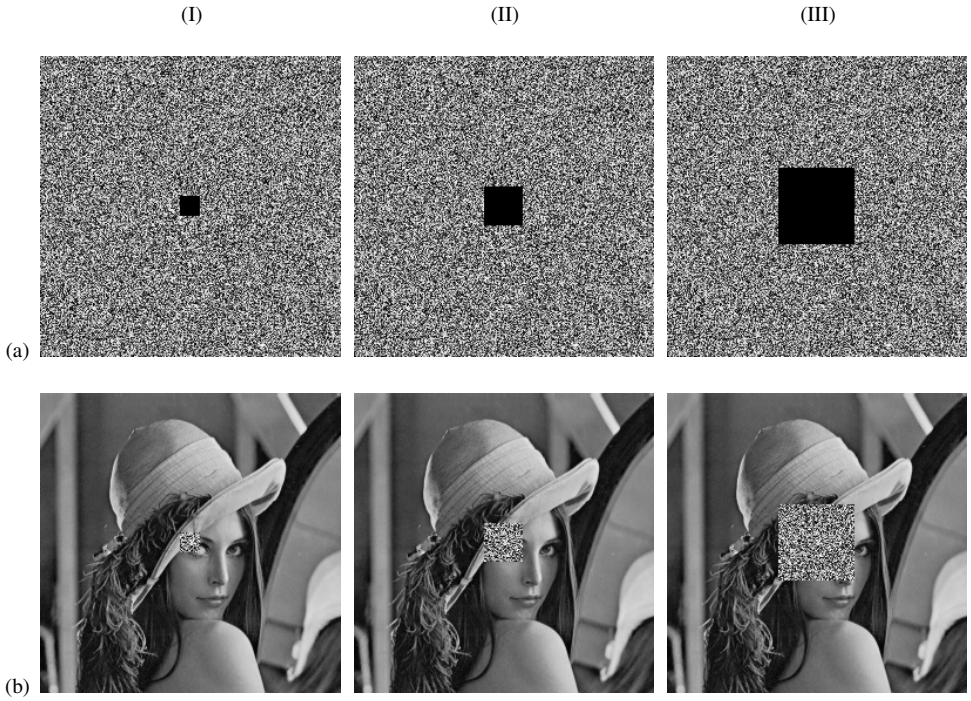


Figura 4.9: Perda de informação nas imagens cifradas ((a-I), (a-II) e (a-III)) e as respectivas imagens descriptografadas ((b-I), (b-II) e (b-III)) para o algoritmo I. As imagens descriptografadas mostram que a abordagem é resistente à perda de informação. *Rotinas: PRNG 1/Perda de Informação*

.

ataque. O procedimento acontece da seguinte forma:

Passo A: O vetor S na Eq. (3.15) é reorganizado em ordem crescente usando a função disponível no *Matlab* `[a, b] = sort(S)`. Antes de ocorrer a operação bit-XOR, a posição dos *pixels* é embaralhada usando o índice *b*, obtido da função.

Para analisar a robustez de tal criptossistema contra este tipo de ataque, foram removidos blocos de 512×256 *pixels* das imagens encriptadas. A Fig. 4.10 exibe as imagens cifradas com perda de informação e as respectivas imagens descriptografadas. Mesmo com perda significativa de informação, a imagem descriptografada de Lena ainda é reconhecível.

4.10 Resistência a Ataques Via Imagens Simples Conhecidas/Escolhidas

Ataques via imagens simples e/ou conhecidas são típicos ataques em criptossistemas [18]. Ataque via imagem conhecida consiste no conhecimento, por parte do *hacker*, de um fragmento da sequência da imagem cifrada e da imagem original [17]. Por outro lado, no ataque via imagem escolhida, o *hacker* consegue obter a imagem cifrada a partir da encriptação da imagem escolhida, buscando informações que comprometam o sistema criptográfico. Nas cifras pro-

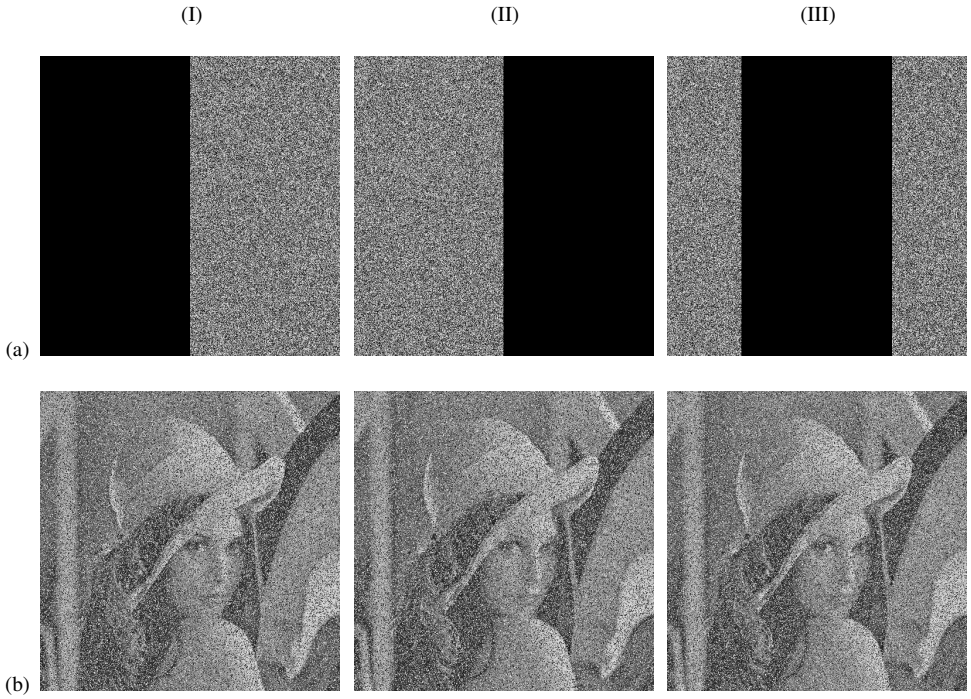


Figura 4.10: Perda de informação no algoritmo II. Na linha (a) é exibida as imagens encriptadas com perda de informação. As respectivas imagens descriptografadas são exibidas na linha (b). *Rotinas: PRNG 2/Principal*

.

postas nessa dissertação, a chave secreta está relacionada à imagem a ser encriptada, tornando o sistema difícil de ser quebrado. Além disso, com particular atenção para a cifra [1], quando imagens totalmente preta ou branca (comumente usadas em tais ataques [10, 15]) são criptografadas, os resultados mostrados, até então, indicam que as imagens cifradas não trazem nenhuma informação, passando em todos os testes submetidos.

4.11 Resistência ao Ruído

O criptossitema deve ser robusto o suficiente para resistir a pertubações [5, 10, 18]. A fim de verificar se o algoritmo I é resistente, foram realizados dois testes, ora adicionando ruído gaussiano na imagem, ora introduzindo ruído *salt and pepper*.

Seguindo a estratégia de [10, 53], um ruído gaussiano branco, com média igual a zero e variância entre 0,00001 e 0,1, foi acrescentado na imagem de Lena. Por outro lado, foi adicionado, também, à imagem de Lena ruído *salt and pepper* com densidades variando de 0,01 até 0,6 [5, 18, 27].

A Fig. 4.11, organizada em colunas, exibe os resultados obtidos para este teste. As colu-

nas (I) e (III) exibem as imagens encriptadas com o ruído, ora gaussiano, ora *salt and pepper* adicionado. Por fim, as colunas (II) e (IV) mostram as respectivas imagens descriptografadas obtidas. Tendo em vista tais informações, pode-se concluir que a maioria das informações da imagem original pode ser restaurada, e a cifra é resistente a esses tipos de perturbações.

A título de comparação, os autores de [10, 90] usaram ruído gaussiano com variância igual a $0{,}0001$ e $0{,}0025$, respectivamente, enquanto os resultados desta dissertação mostraram imagens descriptografadas de boa qualidade com uma variação significativamente maior, igual a $0{,}1$. Outrossim, os autores de [5, 18, 27] usaram ruídos *salt and pepper* com uma densidade igual a $0{,}01$, $0{,}01$ e $0{,}5$, nessa ordem, enquanto esse método consegue resistir a esse tipo de ruído com uma densidade igual a $0{,}6$.

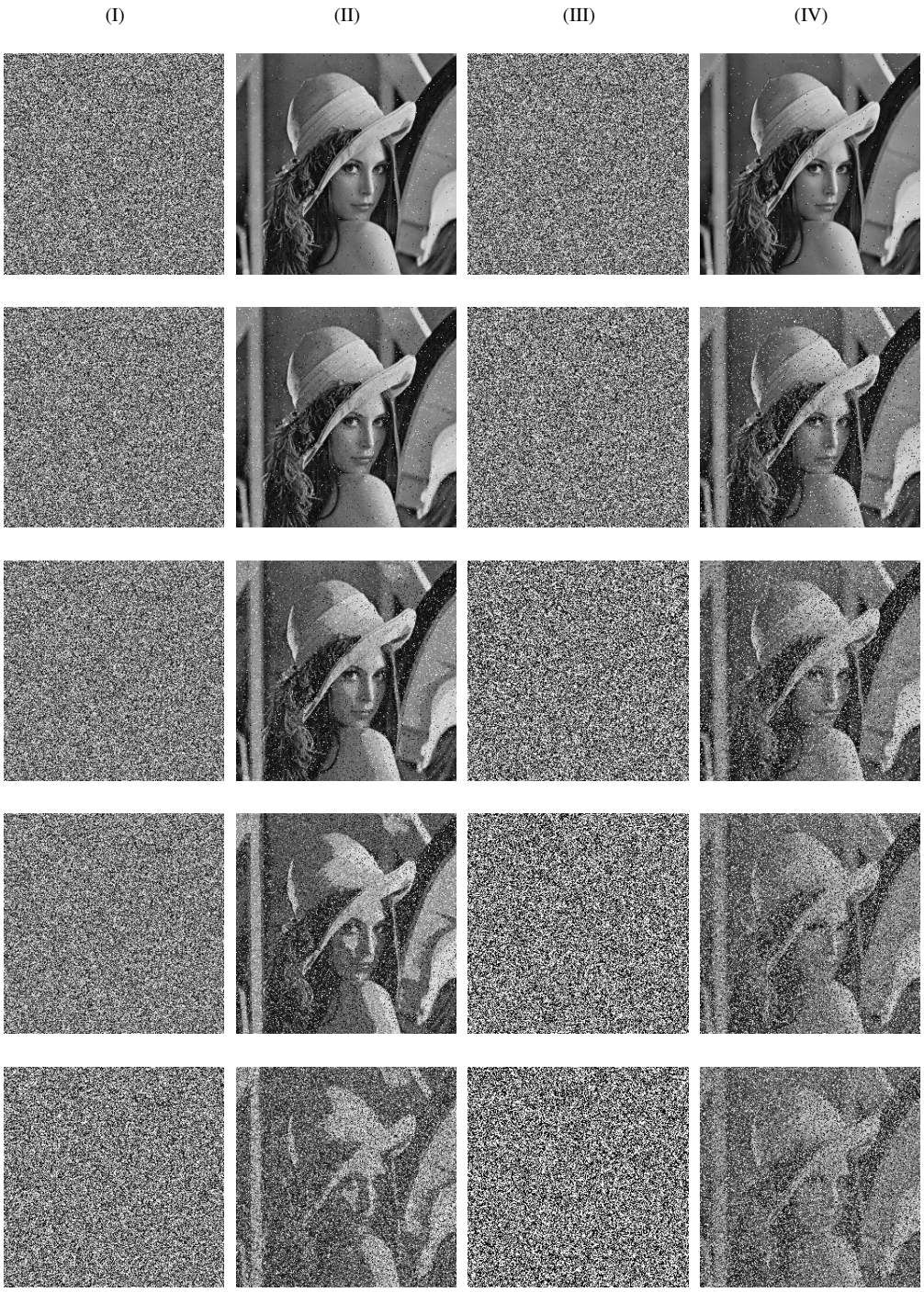


Figura 4.11: Imagens encriptadas/descriptografadas com ruído adicionado. A primeira coluna exibe as imagens cifradas com ruído gaussiano adicionado, com média zero e as respectivas variâncias iguais a 0,00001, 0,0001, 0,001, 0,01 e 0,1; enquanto a segunda coluna exibe as relativas imagens descriptografadas. Do mesmo modo, a terceira coluna exibe as imagens cifradas com ruído *salt and pepper* adicionado, com as respectivas densidades iguais a 0,01, 0,1, 0,3, 0,5 e 0,6; ao passo que a quarta coluna exibe as relativas imagens descriptografadas. É possível notar que, mesmo com a inserção de ruído, a maioria das informações ainda é recuperada. *Rotinas: Resistência ao ruído*.

CAPÍTULO 5

CONCLUSÕES

5.1 Considerações Finais

Nesta dissertação, foram apresentados novos sistemas de criptografia relacionados à imagem que se deseja criptografar. As abordagens são, essencialmente, baseadas nos conceitos de extensões naturais intervalares [44, 45, 52, 91, 92], na precisão finita nos computadores e nas limitações do padrão de ponto flutuante IEEE 754-2008 [43] na representação numérica, evitando a degradação de sistemas caóticos.

Apesar de ambas as cifras apresentarem particularidades, os métodos podem ser descritos, basicamente, como as simulações de duas extensões intervalares de um sistema caótico; sendo que a imagem a ser encriptada e as condições iniciais do sistema estão intrinsecamente ligadas. Após a simulação, o cálculo do limite inferior do erro é usado para gerar a chave secreta. Por fim, a operação bit-XOR entre a imagem e a chave secreta é feita, obtendo-se, assim, a imagem encriptada.

Os algoritmos propostos neste texto provaram ser muito eficientes, produzindo sequências pseudo-aleatórias com excelentes propriedades criptográficas, tornando possível a criptografia de todas as imagens impostas. Ademais, os criptossistemas realçaram a característica aleatória do erro.

Adicionalmente, os resultados das simulações mostraram que os criptossistemas são tão eficientes quanto outros métodos encontrados na literatura. Além de passar em todos os testes estatísticos, como as análises de entropia de informação e da correlação dos *pixels* adjacentes na imagem, eles resistem, também, a ciberataques muito comuns e/ou conhecidos, como a criptoanálise diferencial.

As abordagens apresentadas são eficientes, também, sob o ponto de vista de tempo computacional e memória utilizada, com particular atenção ao método que emprega mapas caóticos 1D. Ao analisar a complexidade dos algoritmos, os mesmos apresentaram uma complexidade de O($Lg \times W$); reafirmando, portanto, a competitividade que os criptossistemas mostrados na

dissertação têm frente a outros algoritmos.

5.2 Perspectivas Futuras

A fim de dar prosseguimento ao trabalho originado desta dissertação, os seguintes trabalhos são propostos:

- Analisar a influência dos mapas e sistemas caóticos no sistema criptográfico proposto, observando os dados quantitativos e qualitativos dos testes estatísticos e ataques cibernéticos estudados, juntamente com a complexidade e tempo de encriptação/decriptação do algoritmo, buscando encontrar o sistema “ótimo".

- Desenvolver uma metodologia de codificação em Sistemas Digitais Reconfiguráveis tais como os FPGAs (Arranjo de Portas Programáveis em Campo, do inglês *Field Programmable Gate Array*), utilizando a linguagem de descrição de *hardware* VHDL (Linguagem de Descrição de Hardware VHSIC, do inglês *VHSIC Hardware Description Language*), tendo um maior controle nos parâmetros de projeto de sistemas digitais em si, como por exemplo, a quantidade de recursos lógicos utilizados (área em silício), desempenho e a potência consumida.

- Criar uma interface gráfica para os algoritmos testados nesta dissertação, com o intuito de divulgar, simultaneamente, a pesquisa e o método, seguindo a mesma ideia do artigo de Silva et al. [93].

- Elaborar um método híbrido, fundamentado nos sistemas caóticos e nos sistemas de criptografia tradicionais, os quais sejam capazes de criptografar os mais diversos formatos de mídia (textos, imagens, vídeos).

REFERÊNCIAS BIBLIOGRÁFICAS

[1] L. G. Nardo, E. G. Nepomuceno, J. Arias-Garcia, e D. N. Butusov, "Image encryption using finite-precision error," *Chaos, Solitons & Fractals*, vol. 123, pp. 69–78, 2019.

[2] E. G. Nepomuceno, L. G. Nardo, J. Arias-Garcia, D. N. Butusov, e A. Tutueva, "Image encryption based on the pseudo-orbits from 1D chaotic map," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 29, no. 6, p. 061101, 2019.

[3] D. S. Terzi, R. Terzi, e S. Sagiroglu, "A survey on security and privacy issues in big data," in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, no. December. IEEE, 2015, pp. 202–207.

[4] P. P. Swire e K. Ahmad, "Encryption and Globalization," *SSRN Electronic Journal*, vol. 23, pp. 416–481, 2011.

[5] A. Ullah, S. S. Jamal, e T. Shah, "A novel scheme for image encryption using substitution box and chaotic system," *Nonlinear Dynamics*, vol. 91, no. 1, pp. 359–370, 2018.

[6] Ü. Çavuşoğlu, S. Kaçar, I. Pehlivan, e A. Zengin, "Secure image encryption algorithm design using a novel chaos based S-Box," *Chaos, Solitons & Fractals*, vol. 95, pp. 92–101, 2017.

[7] X. Chai, "An image encryption algorithm based on bit level Brownian motion and new chaotic systems," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 1159–1175, 2017.

[8] L. Liu, J. Lin, S. Miao, e B. Liu, "A Double Perturbation Method for Reducing Dynamical Degradation of the Digital Baker Map," *International Journal of Bifurcation and Chaos*, vol. 27, p. 1750103, 2017.

[9] Y. Wu, G. Yang, H. Jin, e J. P. Noonan, "Image encryption using the two-dimensional logistic chaotic map," *Journal of Electronic Imaging*, vol. 21, no. 1, p. 013014, 2012.

[10] Z.-h. Gan, X.-l. Chai, D.-j. Han, e Y.-r. Chen, "A chaotic image encryption algorithm based on 3-D bit-plane permutation," *Neural Computing and Applications*, vol. 31, no. 11, pp. 7111–7130, 2019.

[11] R. Jain, *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons, 1990.

[12] L. Milinković, M. Antić, e Z. Čiča, "Pseudo-random number generator based on irrational numbers," in *2011 10th International Conference on Telecommunication in Modern Satellite Cable and Broadcasting Services (TELSIKS)*, vol. 2, no. 1. IEEE, 2011, pp. 719–722.

[13] L.-C. Cao, Y.-L. Luo, S.-H. Qiu, e J.-X. Liu, "A perturbation method to the tent map based on Lyapunov exponent and its application," *Chinese Physics B*, vol. 24, no. 10, p. 100501, 2015.

[14] X. Wang, J. Zhao, e H. Liu, "A new image encryption algorithm based on chaos," *Optics Communications*, vol. 285, no. 5, pp. 562–566, 2012.

[15] H. Zhu, X. Zhang, H. Yu, C. Zhao, e Z. Zhu, "An image encryption algorithm based on compound homogeneous hyper-chaotic system," *Nonlinear Dynamics*, vol. 89, no. 1, pp. 61–79, 2017.

[16] B. Norouzi e S. Mirzakuchaki, "A fast color image encryption algorithm based on hyper-chaotic systems," *Nonlinear Dynamics*, vol. 78, no. 2, pp. 995–1015, 2014.

[17] Y. Luo, M. Du, e J. Liu, "A symmetrical image encryption scheme in wavelet and time domain," *Communications in Nonlinear Science and Numerical Simulation*, vol. 20, no. 2, pp. 447–460, 2015.

[18] X. Wu, K. Wang, X. Wang, e H. Kan, "Lossless chaotic color image cryptosystem based on DNA encryption and entropy," *Nonlinear Dynamics*, vol. 90, no. 2, pp. 855–875, 2017.

[19] W. Xing-Yuan e W. Qian, "A fast image encryption algorithm based on only blocks in cipher text," *Chinese Physics B*, vol. 23, no. 3, p. 030503, 2014.

[20] Z. Yong, "The image encryption algorithm with plaintext-related shuffling," *IETE Technical Review*, vol. 33, no. 3, pp. 310–322, 2016.

[21] L. Zhang, X. Liao, e X. Wang, "An image encryption approach based on chaotic maps," *Chaos, Solitons & Fractals*, vol. 24, no. 3, pp. 759–765, 2005.

[22] J. Banks, J. Brooks, G. Cairns, G. Davis, e P. Stacey, "On Devaney's Definition of Chaos," *The American Mathematical Monthly*, vol. 99, no. 4, p. 332, 1992.

[23] C. Herring e J. I. Palmore, "Random number generators are chaotic," *ACM SIGPLAN Notices*, vol. 24, no. 11, pp. 76–79, 1989.

[24] R. Matthews, "On the derivation of a "chaotic" encryption algorithm," *Cryptologia*, vol. 13, no. 1, pp. 29–42, 1989.

[25] Z. Su, G. Zhang, e J. Jiang, "Multimedia Security: A Survey of Chaos-Based Encryption Technology," in *Multimedia - A Multidisciplinary Approach to Complex Issues*. InTech, 2012, pp. 99–124.

[26] C. Li, G. Luo, K. Qin, e C. Li, "An image encryption scheme based on chaotic tent map," *Nonlinear Dynamics*, vol. 87, no. 1, pp. 127–133, 2017.

[27] T. Hu, Y. Liu, L.-H. Gong, e C.-J. Ouyang, "An image encryption scheme combining chaos with cycle operation for DNA sequences," *Nonlinear Dynamics*, vol. 87, no. 1, pp. 51–66, 2017.

[28] L. Bao e Y. Zhou, "Image encryption: Generating visually meaningful encrypted images," *Information Sciences*, vol. 324, pp. 197–207, 2015.

[29] F. Özkaynak, "Brief review on application of nonlinear dynamics in image encryption," *Nonlinear Dynamics*, vol. 92, no. 2, pp. 305–313, 2018.

[30] Y. Wu, J. P. Noonan, e S. Agaian, "NPCR and UACI Randomness Tests for Image Encryption," *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT)*, no. 2011, pp. 31–38, 2011.

[31] S. Li, G. Chen, e X. Mou, "On the Dynamical Degradation of Digital Piecewise Linear Chaotic Maps," *International Journal of Bifurcation and Chaos*, vol. 15, no. 10, pp. 3119–3151, 2005.

[32] R. M. May, "Simple mathematical models with very complicated dynamics." *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.

[33] L. Liu e S. Miao, "A universal method for improving the dynamical degradation of a digital chaotic system," *Physica Scripta*, vol. 90, no. 8, p. 085205, 2015.

[34] Y. Liu, Y. Luo, S. Song, L. Cao, J. Liu, e J. Harkin, "Counteracting Dynamical Degradation of Digital Chaotic Chebyshev Map via Perturbation," *International Journal of Bifurcation and Chaos*, vol. 27, no. 03, p. 1750033, 2017.

[35] D. D. Wheeler e R. A. J. Matthews, "Supercomputer investigations of a chaotic encryption algorithm," *Cryptologia*, vol. 15, no. 2, pp. 140–152, 1991.

[36] H. S. Kwok e W. K. S. Tang, "A fast image encryption system based on chaotic maps with finite precision representation," *Chaos, Solitons & Fractals*, vol. 32, no. 4, pp. 1518–1529, 2007.

[37] Y. Zhou, L. Bao, e C. L. P. Chen, "Image encryption using a new parametric switching chaotic system," *Signal Processing*, vol. 93, no. 11, pp. 3039–3052, 2013.

[38] L. Min, X. Yang, G. Chen, e D. Wang, "Some Polynomial Chaotic Maps Without Equilibria and an Application to Image Encryption with Avalanche Effects," *International Journal of Bifurcation and Chaos*, vol. 25, no. 09, p. 1550124, 2015.

[39] L. Shujun, M. Xuanqin, e C. Yuanlong, "Pseudo-random Bit Generator Based on Couple Chaotic Systems and Its Applications in Stream-Cipher Cryptography," in *International Conference on Cryptology in India*, 2001, pp. 316–329.

[40] L. Liu, H. Hu, e Y. Deng, "An analogue–digital mixed method for solving the dynamical degradation of digital chaotic systems," *IMA Journal of Mathematical Control and Information*, vol. 32, no. 4, p. 13, 2014.

[41] B. Bakhache, J. M. Ghazal, e S. E. Assad, "Improvement of the Security of ZigBee by a New Chaotic Algorithm," *IEEE Systems Journal*, vol. 8, no. 4, pp. 1024–1033, 2014.

[42] Z. Yan-Bin e D. Qun, "A New Digital Chaotic Sequence Generator Based on Logistic Map," in *2011 Second International Conference on Innovations in Bio-inspired Computing and Applications*. IEEE, 2011, pp. 175–178.

[43] Institute of Electrical and Electronics Engineers (IEEE), *754-2008 – IEEE standard for floating-point arithmetic*. IEEE, 2008.

[44] E. G. Nepomuceno e S. A. M. Martins, "A lower bound error for free-run simulation of the polynomial NARMAX," *Systems Science & Control Engineering*, vol. 4, no. 1, pp. 50–58, 2016.

[45] E. Nepomuceno, S. Martins, G. Amaral, e R. Riveret, "On the lower bound error for discrete maps using associative property," *Systems Science & Control Engineering*, vol. 5, no. 1, pp. 462–473, 2017.

[46] L. G. Nardo, A. M. Lima, E. G. Nepomuceno, e J. Arias-Garcia, "Image encryption algorithm using natural interval extensions," in *BTSym 2018 Proceedings*, vol. 1, 2018, pp. 1–5.

[47] H. K. Khalil, *Nonlinear Systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.

[48] J.-J. E. Slotine e W. Li, *Applied Nonlinear Control*. Prentice hall Englewood Cliffs, NJ, 1991, vol. 1.

[49] S. Wiggins, *Introduction to Applied Nonlinear Dynamical Systems and Chaos*. Springer Science & Business Media, 2003, vol. 2.

[50] H.-O. Peitgen, H. Jürgens, e D. Saupe, *Chaos and Fractals: New frontiers of science*. Springer, 2004.

[51] E. G. Nepomuceno, "Convergence of recursive functions on computers," *The Journal of Engineering*, vol. 2014, no. 10, pp. 560–562, 2014.

[52] E. G. Nepomuceno e E. M. A. M. Mendes, "On the analysis of pseudo-orbits of continuous chaotic nonlinear systems simulated using discretization schemes in a digital computer," *Chaos, Solitons & Fractals*, vol. 95, pp. 21–32, 2017.

[53] M. J. Rostami, A. Shahba, S. Saryazdi, e H. Nezamabadi-pour, "A novel parallel image encryption with chaotic windows based on logistic map," *Computers & Electrical Engineering*, vol. 62, pp. 384–400, 2017.

[54] S. M. Ismail, L. A. Said, A. G. Radwan, A. H. Madian, M. F. Abu-ElYazeed, e A. M. Soliman, "Generalized fractional logistic map suitable for data encryption," in *2015 International Conference on Science and Technology (TICST)*. IEEE, 2015, pp. 336–341.

[55] L. O. Chua, C. W. Wu, A. Huang, e G.-Q. Zhong, "A universal circuit for studying and generating chaos. I. Routes to chaos," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, no. 10, pp. 732–744, 1993.

[56] M. P. Kennedy, "Robust OP Amp Realization of Chua's Circuit," *Frequenz*, vol. 46, no. 3-4, pp. 66–80, 1992.

[57] T. Matsumoto, "Chaos in electronic circuits," *Proceedings of the IEEE*, vol. 75, no. 8, pp. 1033–1057, 1987.

[58] L. A. B. Tôrres e L. A. Aguirre, "PCCHUA — A laboratory setup for real-time control and synchronization of chaotic oscillations," *International Journal of Bifurcation and Chaos*, vol. 15, no. 08, pp. 2349–2360, 2005.

[59] R. Kiliç, *A practical guide for studying Chua's circuits*. World Scientific, 2010, vol. 71.

[60] Qian Gong-bin, Jiang Qing-feng, e Qiu Shui-sheng, "A new image encryption scheme based on DES algorithm and Chua's circuit," in *2009 IEEE International Workshop on Imaging Systems and Techniques*. IEEE, 2009, pp. 168–172.

[61] G. Peng e F. Min, "Multistability analysis, circuit implementations and application in image encryption of a novel memristive chaotic circuit," *Nonlinear Dynamics*, vol. 90, no. 3, pp. 1607–1625, 2017.

[62] B. Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, 2nd ed. Oxford University Press, 2010.

[63] M. L. Overton, *Numerical Computing with IEEE Floating Point Arithmetic*. Society for Industrial and Applied Mathematics, 2001.

[64] D. Morgan, *Numerical Methods: Real-Time and Embedded Systems Programming*. M & T Books, 1992.

[65] L. F. Shampine, R. C. Allen Jr., e S. Pruess, *Fundamentals of numerical computing*. John Wiley & Sons, Inc, 1997, vol. 1.

[66] V. Rajaraman, "IEEE standard for floating point numbers," *Resonance*, vol. 21, no. 1, pp. 11–30, 2016.

[67] R. E. Moore, R. B. Kearfott, e M. J. Cloud, *Introduction to interval analysis*, 1st ed. Society for Industrial and Applied Mathematics, 2009.

[68] R. A. Mollin, *An introduction to cryptography*, 2nd ed., ser. Discrete mathematics and its applications. Chapman & Hall/CRC, 2007.

[69] S. Burnett e S. Paine, *RSA Security's official guide to cryptography*, 1st ed. McGraw-Hill Osborne Media, 2001.

[70] D. Boneh e V. Shoup, "A Graduate Course in Applied Cryptography," *version 0.4*, 2017.

[71] R. J. Tocci, N. S. Widmer, e G. L. Moss, *Digital Systems Principles and Applications*, 10th ed. Prentice Hall Inc, 2007.

[72] G. Alvarez e S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International Journal of Bifurcation and Chaos*, vol. 16, no. 08, pp. 2129–2151, 2006.

[73] W. W. Fung, M. J. Golin, e J. W. Gray, "Protection of keys against modification attack," *Quality and Reliability Engineering International*, vol. 18, no. 3, pp. 217–230, 2002.

[74] L. A. Aguirre, *Introdução à identificação de sistemas -Técnicas lineares e não-lineares aplicadas a sistemas: Teoria e Aplicação*, 4th ed. UFMG, 2015.

[75] E. G. Nepomuceno, S. A. M. Martins, B. C. Silva, G. F. V. Amaral, e M. Perc, "Detecting unreliable computer simulations of recursive functions with interval extensions," *Applied Mathematics and Computation*, vol. 329, pp. 408–419, 2018.

[76] C. Li, B. Feng, S. Li, J. Kurths, e G. Chen, "Dynamic Analysis of Digital Chaotic Maps via State-Mapping Networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 6, pp. 2322–2335, 2019.

[77] L. Hongjun e W. Xingyuan, "Color image encryption based on one-time keys and robust chaotic maps," *Computers & Mathematics with Applications*, vol. 59, no. 10, pp. 3320–3327, 2010.

[78] L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. F. Dray, e S. Vo, "A

statistical test suite for random and pseudorandom number generators for cryptographic applications," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. April, 2010.

[79] A.-V. Diaconu e A. C. Dascalescu, "Correlation distribution of adjacent pixels randomness test for image encryption," *Proceedings of the Romanian Academy Series A - Mathematics Physics Technical Sciences Information Science*, vol. 18, pp. 351–359, 2017.

[80] R. K. Jangid, N. Mohmmad, A. Didel, e S. Taterh, "Hybrid approach of image encryption using DNA cryptography and TF Hill Cipher Algorithm," in *2014 International Conference on Communication and Signal Processing*. IEEE, 2014, pp. 934–938.

[81] H. Yang, K.-W. Wong, X. Liao, W. Zhang, e P. Wei, "A fast image encryption and authentication scheme based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 11, pp. 3507–3517, 2010.

[82] A.-V. Diaconu, "Circular inter–intra pixels bit-level permutation and chaos-based image encryption," *Information Sciences*, vol. 355-356, pp. 314–327, 2016.

[83] X. Wang e C. Jin, "Image encryption using Game of Life permutation and PWLCM chaotic system," *Optics Communications*, vol. 285, no. 4, pp. 412–417, 2012.

[84] X. Chai, Y. Chen, e L. Broyde, "A novel chaos-based image encryption algorithm using DNA sequence operations," *Optics and Lasers in Engineering*, vol. 88, pp. 197–213, 2017.

[85] X. Zhang, Z. Zhao, e J. Wang, "Chaotic image encryption based on circular substitution box and key stream buffer," *Signal Processing: Image Communication*, vol. 29, no. 8, pp. 902–913, 2014.

[86] C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[87] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, e P. Natarajan, "Local Shannon entropy measure with statistical tests for image randomness," *Information Sciences*, vol. 222, pp. 323–342, 2013.

[88] W. Xing-Yuan e W. Qian, "A fast image encryption algorithm based on only blocks in cipher text," *Chinese Physics B*, vol. 23, no. 3, p. 030503, 2014.

[89] S. Namasudra e G. C. Deka, *Advances of DNA computing in cryptography*. Chapman and Hall/CRC, 2019.

[90] H. Li e Y. Wang, "Double-image encryption based on iterative gyrator transform," *Optics Communications*, vol. 281, no. 23, pp. 5745–5749, 2008.

[91] E. M. A. M. Mendes e E. G. Nepomuceno, "A Very Simple Method to Calculate the (Positive) Largest Lyapunov Exponent Using Interval Extensions," *International Journal of Bifurcation and Chaos*, vol. 26, no. 13, p. 1650226, 2016.

[92] E. G. Nepomuceno, S. A. M. Martins, M. J. Lacerda, e E. M. A. M. Mendes, "On the Use of Interval Extensions to Estimate the Largest Lyapunov Exponent from Chaotic Data," *Mathematical Problems in Engineering*, vol. 2018, pp. 1–8, 2018.

[93] P. H. O. Silva, L. G. Nardo, S. A. M. Martins, E. G. Nepomuceno, e M. Perc, "Graphical interface as a teaching aid for nonlinear dynamical systems," *European Journal of Physics*, vol. 39, no. 6, p. 065105, 2018.

APÊNDICE A

ROTINAS COMPUTACIONAIS

Os algoritmos que foram elaborados ao longo deste trabalho foram desenvolvidos via Matlab. Tais programas podem ser encontrados no site: www.ufsj.edu.br/gcom/nardo2019.php.