Um Estudo sobre Criptografia Simétrica baseada em Autômatos Celulares Híbridos

Everton Rocha Lira

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2025

Everton Rocha Lira

Um Estudo sobre Criptografia Simétrica baseada em Autômatos Celulares Híbridos

Tese de doutorado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Luiz Gustavo Almeida Martins

Uberlândia
2025

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

L768 Lira, Everton Rocha, 1989-
2025 Um estudo sobre criptografia simétrica baseada em autômatos celulares híbridos [recurso eletrônico] / Everton Rocha Lira. - 2025.

Orientador: Luiz Gustavo Almeida Martins.
Tese (Doutorado) - Universidade Federal de Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.te.2025.55
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Martins, Luiz Gustavo Almeida,1974-, (Orient.). II. Universidade Federal de Uberlândia. Pós-graduação em Ciência da Computação. III. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Computação
Av. João Naves de Ávila, 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Tese, 50/2025, PPGCO				
Data:	30 de janeiro de 2025	Hora de início:	8:34	Hora de encerramento:	12:47
Matrícula do Discente:	11613CCP008				
Nome do Discente:	Everton Rocha Lira				
Título do Trabalho:	Um Estudo sobre Criptografia Simétrica baseada em Autômatos Celulares Híbridos.				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Inteligência Artificial				
Projeto de Pesquisa de vinculação:	Computação Bio-inspirada e Aprendizagem de Máquina na Resolução de Problemas.				

Reuniu-se por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Rodrigo Sanches Miani - FACOM/UFU, Bruno Augusto Nassif Travençolo - FACOM/UFU, Bruno Bogaz Zarpelão - DC/UEL, Marco Aurelio Amaral Henriques - DCA-FEEC/UNICAMP e Luiz Gustavo Almeida Martins - FACOM/UFU, orientador do candidato.

Os examinadores participaram desde as seguintes localidades: Bruno Bogaz Zarpelão- Londrina/PR, Marco Aurelio Amaral Henriques - Campinas/SP . Os outros membros da banca e o aluno participaram da cidade de Uberlândia.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Luiz Gustavo Almeida Martins, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação da Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir ao candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Doutor.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação

interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Bruno Augusto Nassif Travençolo**, **Professor(a) do Magistério Superior**, em 30/01/2025, às 14:37, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Luiz Gustavo Almeida Martins**, **Professor(a) do Magistério Superior**, em 30/01/2025, às 15:45, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Marco Aurelio Amaral Henriques**, **Usuário Externo**, em 30/01/2025, às 16:39, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Bruno Bogaz Zarpelão**, **Usuário Externo**, em 30/01/2025, às 19:35, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rodrigo Sanches Miani**, **Professor(a) do Magistério Superior**, em 30/01/2025, às 23:44, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5955174** e o código CRC **24549C3B**.

Referência: Processo nº 23117.086369/2024-75 SEI nº 5955174

Este trabalho é dedicado a todos os grandes questionadores,
àqueles que mudaram o mundo após rejeitar "porque sim" como resposta.

Agradecimentos

Em primeiro lugar deixo agradecimento póstumo à minha prévia orientadora, Dra. Gina Maira, pelo incentivo desde a época da graduação e cujo legado positivo se perpetua. Também deixo agradecimento especial ao Dr. Luiz Gustavo que me orientou no restante do processo, bem como ao corpo docente do PPGCO-UFU e à CAPES pelo fomento que possibilitou este trabalho. Deixo aqui menção grata aos meus pais, Rosângela e Elizeu Lira, pelo apoio constante à minha educação e desenvolvimento. E registro também agradecimento a diversos amigos e colegas, que não poderia listar sem cometer injustiças, pela escuta e motivação fundamentais no processo deste Doutorado.

"O destino embaralha as cartas, mas somos nós que as jogamos"
(Arthur Schopenhauer)

Resumo

Devido ao crescente número de sistemas computacionais conectados à Internet, há uma demanda por formas mais rápidas e seguras de permitir a comunicação entre tais sistemas. É comum que haja troca de dados sigilosos entre dispositivos pessoais e/ou servidores, e tais informações devem ser protegidas. Métodos e técnicas criptográficas têm sido utilizados para prover sigilo de dados entre entidades, e tais mecanismos possibilitam a existência de sistemas de *internet banking*, operações na bolsa de valores, identificação pessoal segura, entre outros. Existem algoritmos criptográficos clássicos, tais como o AES (*Advanced Encryption Standard*) e o DES (*Data Encryption Standard*), que foram padrões da indústria por décadas. Mesmo o DES já sendo considerado obsoleto, o AES ainda é tido como um algoritmo seguro. Entretanto, a versão original do AES tem suas desvantagens, especialmente ao lidar com a criptografia de imagens e com relação a otimizações para operação *multithread*. Uma alternativa é desenvolver novos métodos de cifração baseados em mecanismos que permitam uma melhor exploração do paralelismo presente nas máquinas atuais e em hardware dedicado. Algumas das possibilidades investigadas atualmente envolvem Autômatos Celulares devido à sua natureza intrinsecamente paralela. Nesta tese, foi desenvolvido um novo algoritmo chamado VHCA (*Very Heterogeneous Cellular Automata*). No VHCA, regras de Autômato Celular extraídas a partir de uma chave secreta são repetidamente aplicadas a células de reticulados inicializados com trechos do texto em claro. A sucessiva aplicação dessas regras a cada reticulado resulta em configurações (estados) referentes aos trechos correspondentes do texto cifrado. Os resultados experimentais apresentaram indícios de que o VHCA é uma solução promissora no contexto de novos modelos criptográficos.

Palavras-chave: Autômatos Celulares. Criptografia. Criptografia Simétrica. Paralelismo. .

Abstract

Due to the increasing number of computational systems connected to the Internet, there is a higher demand for faster and more secure ways to allow communication between such systems. Confidential information is often exchanged between personal devices and/or servers, and it must be protected. Cryptographic methods and techniques are used to provide data secrecy between parties and allow the existence of internet banking systems, online stock market operations, secure online personal identification, and so on. There are classical cryptographic methods, such as AES (*Advanced Encryption Standard*) and DES (*Data Encryption Standard*), which have been the industry standards for many years. While DES has been mostly deprecated, AES is still considered a secure algorithm. However, AES has some shortcomings, mainly when dealing with image data encryption and regarding multithread optimization. An alternative is developing new cryptographic methods based on systems that display a greater potential for parallelism in current-age computers and specialized hardware. Some of these novel possibilities involve Cellular Automata (CA) due to it being an inherently parallel model. A novel algorithm called VHCA (*Very Heterogeneous Cellular Automata*) was developed in the scope of this thesis. In VHCA, CA rules extracted from a secret key are repeatedly applied to cells in lattices whose values were initialized with chunks of the plaintext. These successive CA evolutions will result in configurations (states) matching the corresponding chunks of the ciphertext. Initial experiments point to VHCA being a promising solution in the search for new cryptographic models.

Keywords: Cellular Automata. Cryptography. Symmetric Cryptography. Parallelism
.

Lista de ilustrações

Figura 1 – Criptografia Simétrica. [Fonte: autor] 32
Figura 2 – Criptografia Assimétrica. [Fonte: autor] 33
Figura 3 – Exemplo da configuração inicial de um AC. [Fonte: autor] 36
Figura 4 – Formalização da configuração inicial de um AC. [Fonte: autor] 36
Figura 5 – Exemplo de regra de evolução para AC. [Fonte: autor] 37
Figura 6 – Evolução parcial de um reticulado. [Fonte: autor] 39
Figura 7 – Aplicação da condição de limite na evolução do reticulado. [Fonte: autor] . 40
Figura 8 – Mecanismo de nomenclatura de Wolfram. [Fonte: (WOLFRAM, 2002a)] . . 40
Figura 9 – Exemplo de regras com sensitividade. [Fonte: autor] 42
Figura 10 – Exemplos das classes de Wolfram. [Fonte: autor] 44
Figura 11 – Evolução de um reticulado híbrido. [Fonte: autor] 45
Figura 12 – Cifra de Wolfram. [Fonte: autor] 47
Figura 13 – Exemplo de ambiguidade no cálculo de pré-imagem. [Fonte: autor] 49
Figura 14 – Cálculo de bit da pré-imagem na posição com sensitividade. [Fonte: autor] . 49
Figura 15 – Cálculo de pré-imagem por regra com sensitividade. [Fonte: autor] 51
Figura 16 – Cifração por cálculo de pré-imagem no método de Gutowitz usando a regra 149. [Fonte: autor] 52
Figura 17 – Vizinhança e exemplo de bit com sensitividade para $r = 4$. [Fonte: autor] . 54
Figura 18 – Comportamento das regras com sensitividade total. [Fonte: autor] 56
Figura 19 – Cálculo de uma pré-imagem conforme reversibilidade do HCA. [Fonte: autor] 58
Figura 20 – Cifração no método HCA. [Fonte: autor] 60
Figura 21 – Decifração no método HCA. [Fonte: autor] 61
Figura 22 – Aplicação da chave a um bloco do VHCA. [Fonte: autor] 74
Figura 23 – Cálculo de pré-imagem VHCA - sensitividade à esquerda. [Fonte: autor] . . 77
Figura 24 – Rotação borda em evolução de AC - sensitividade esquerda. [Fonte: autor] 81
Figura 25 – Rotação borda cálculo pré-imagem - sensitividade esquerda. [Fonte: autor] 81

Figura 26 – Paralelismo da decifração no VHCA - sensitividade esquerda. [Fonte: autor] 82
Figura 27 – Equivalência de letras na Cifra de César com deslocamento 2. [Fonte: autor] 110
Figura 28 – Cifra de Vigenère. [Fonte: autor] 111
Figura 29 – Exemplo de aplicação do *One-Time Pad* (OTP). [Fonte: autor] 113
Figura 30 – Diagrama geral do algoritmo DES. [Fonte: autor] 115
Figura 31 – Exemplo de S-Box do DES. [Fonte: autor] 116
Figura 32 – Diagrama de funcionamento do TDEA. [Fonte: autor] 117
Figura 33 – Diagrama de funcionamento do AES. [Fonte: autor] 118

Lista de tabelas

Tabela 1 – Regras elementares com sensitividade total. 55
Tabela 2 – Equivalência de regras com sensitividade total. 56
Tabela 3 – Sequência de uso das chaves no processo de cifração do HCA. 63
Tabela 4 – Comparativo dos métodos, com características do VHCA destacadas em verde. 67
Tabela 5 – Regras elementares que exibem sensitividade 71
Tabela 6 – Regras elementares com sensitividade e não-lineares 72
Tabela 7 – Alfabeto da chave no VHCA. 73
Tabela 8 – Resultados - Efeito Avalanche de Texto Claro 88
Tabela 9 – Resultados - Efeito Avalanche de Chave 88
Tabela 10 – Resultados da bateria de testes estatísticos NIST 91
Tabela 11 – Resultados dos testes estatísticos PractRand. 92
Tabela 12 – Custo computacional das operações em blocos de 128 bits 93
Tabela 13 – Aplicação da Cifra de Vigenère ao exemplo. 111

Lista de Algoritmos

1 VHCA - Cifração por Evolução Tradicional de AC 75
2 VHCA - Decifração por Evolução Reversa de AC 79

Lista de siglas

3DHCA *Three-Dimensional Hybrid Cellular Automata*

AC Autômato Celular

AES *Advanced Encryption Standard*

CBC *Cipher Block Chaining*

DES *Data Encryption Standard*

ECB *Electronic Code Book*

FPGA *Field Programmable Gate Array*

HCA *Hybrid Cellular Automata*

NIST *National Institute of Standards and Technology*

NBS *National Bureau of Standards*

OTP *One-Time Pad*

OFB *Output Feedback*

PRNG *PseudoRandom Number Generator*

RSA Rivest–Shamir–Adleman

TDEA *Triple Data Encryption Algorithm*

THCA *Two-Dimensional Hybrid Cellular Automata*

VHCA *Very Heterogeneous Cellular Automata*

Sumário

1 INTRODUÇÃO 25
1.1 Motivação 26
1.2 Objetivos 27
1.3 Hipótese e Questões de Pesquisa 28
1.4 Contribuições 28
1.5 Organização da Tese 28

2 FUNDAMENTAÇÃO TEÓRICA 31
2.1 Criptografia 31
2.1.1 Classificação de Métodos Simétricos 33
2.1.2 Análise de Robustez 34
2.2 Autômatos Celulares 36
2.2.1 Regras de Transição 37
2.2.2 Condição de Limite 38
2.2.3 Exemplo de Evolução 38
2.2.4 Notação de Wolfram 40
2.2.5 Pré-Imagens e Reversibilidade 41
2.2.6 Sensitividade de Regras 42
2.2.7 Classes de Comportamento 43
2.2.8 Autômatos Celulares Híbridos 45
2.3 Criptografia baseada em Autômatos Celulares 47
2.3.1 AC como Gerador de Chaves Secretas 47
2.3.2 AC como Método Criptográfico 48
2.3.3 *Hybrid Cellular Automata* (HCA) 53
2.3.4 Estado da Arte 63

3 VHCA 69
3.1 VHCA - Introdução 69

3.2 VHCA - Chave Criptográfica 70
3.3 VHCA - Cifração 74
3.4 VHCA - Decifração 76
3.5 VHCA - Movimentação da Borda 80
3.6 VHCA - Paralelismo ao Decifrar 82

4 EXPERIMENTOS E ANÁLISE DOS RESULTADOS 85
4.1 Mecanismo XOR 86
4.2 Efeito Avalanche 87
4.3 Bateria de Testes Estatísticos do NIST 89
4.4 Bateria de Testes Estatísticos PractRand 91
4.5 Análise de Desempenho 92
4.6 Ataques Convencionais da Literatura 93
4.7 Comentários Adicionais 94

5 CONCLUSÃO 97
5.1 Principais Contribuições 98
5.2 Produção Bibliográfica 98
5.3 Trabalhos Futuros 99

REFERÊNCIAS 101

ANEXOS 107

ANEXO A – HISTÓRIA DA CRIPTOGRAFIA 109
A.1 Criptografia na Antiguidade 109
A.2 Criptografia na Idade Média e Moderna 110
A.3 Criptografia na Idade Contemporânea 112
A.3.1 Cifra de Uso Único (OTP) 113
A.3.2 DES 114
A.3.3 Triple DES (TDEA) 116
A.3.4 AES 117

Capítulo 1

Introdução

A elaborada comunicação humana, em suas diversas formas e idiomas, permitiu que a sociedade evoluísse mediante a transmissão de conhecimento entre as nações e sucessivas gerações. Além de possibilitar o desenvolvimento tecnológico, a comunicação é fundamental na criação e manutenção das relações interpessoais, sendo estudada pelo prisma de diversas áreas da ciência (TOMASELLO, 2010).

Entretanto, é importante frisar que, embora a comunicação seja frequentemente usada para transmitir informações de acesso público, muitas vezes com fins cooperativos, há situações em que deseja-se manter um conjunto de informações restrito ao conhecimento de um grupo específico. Esse tipo de restrição da informação pode ser motivado por situações de perigo à segurança pessoal dos envolvidos e é de suma importância garantir a qualidade desse controle de acesso aos dados.

A criptografia é uma prática que foi criada para dificultar o acesso de terceiros a dados sigilosos, geralmente por meio de mecanismos de substituição e permutação de trechos da mensagem original (texto em claro). Durante o curso da história, vários métodos criptográficos foram utilizados, sendo que novos métodos são propostos como tentativas de superar limitações ou falhas de seus predecessores. Nesse contexto, pesquisas têm investigado o uso de autômatos celulares em criptografia (WOLFRAM, 1985; GUTOWITZ, 1993; MACÊDO, 2007; LIMA, 2012).

Autômatos Celulares (ACs) são sistemas discretos em que a progressão temporal baseia-se na interação de seus elementos formativos, células binárias, com suas respectivas vizinhas (TOFFOLI; MARGOLUS, 1987). Tais interações sucessivas geram os mais diversos comportamentos complexos, mesmo a partir de regras de interação consideradas simples (WOLFRAM, 2002b). Em um mesmo instante de tempo, as evoluções/progressões das células que compõem o espaço celular (reticulado) de um AC podem ser realizadas de forma simultânea e descentralizada, o que é extremamente atrativo sob a perspectiva de algoritmos distribuídos e paralelos. Essa computação emergente e distribuída a partir de operações simples e interação local são características desejáveis para métodos criptográficos baseados em blocos. Nesse contexto, os ACs podem ser utilizados como geradores de chaves secretas para métodos já existentes (WOLFRAM, 1985) ou como novos algoritmos de criptografia (GUTOWITZ, 1993; WUENSCHE, 2009; MACÊDO, 2007). Entretanto, essas abordagens possuem suas limitações, quer seja por aumentar o tamanho do texto cifrado ou por necessitar de regras com um raio de vizinhança relativamente grande para assegurar a segurança da chave secreta, o que aumenta o custo computacional do método.

O algoritmo *Very Heterogeneous Cellular Automata* (VHCA), objeto desta tese, surge como resposta a tais limitações. O VHCA é um método criptográfico de bloco, cujo procedimento de cifragem é a própria evolução de um AC, o que lhe confere o paralelismo inerente de tal sistema. Assim como o seu predecessor, o algoritmo *Hybrid Cellular Automata* (HCA), o VHCA não causa um aumento no tamanho do bloco durante seu funcionamento. Seu principal diferencial está na aplicação de arranjos de regras de transição elementares (regras binárias de raio 1) como chave privada. Por meio de uma abordagem híbrida, o VHCA se propõe a unir a simplicidade e a eficiência computacional da adoção de regras elementares com a segurança e robustez promovidas pela combinação de diferentes regras. Tais características foram avaliadas por meio de uma investigação da robustez e eficiência do algoritmo, a partir de análises comparativas com outros métodos da literatura.

1.1 Motivação

No momento de escrita desta tese, o método criptográfico AES (PUB, 2001), mediante o uso de chave com tamanho adequado, continua sendo avaliado como seguro nas mais recentes sugestões de métodos criptográficos do departamento de segurança dos EUA (STANDARDS et al., 2023). Apesar de ser considerado seguro para casos gerais e de ser classificado como um algoritmo com alto desempenho, ao estudar o AES clássico percebe-se que ele foi escrito de forma sequencialmente procedural, sendo um desafio paralelizar suas etapas.

Dessa forma, embora exista a possibilidade de aplicar eficientemente o paralelismo no cálculo inter-blocos, quando há a disponibilidade de múltiplos núcleos de processamento, tanto a decomposição dos procedimentos intra-blocos de cifração e decifração em partes que podem ser executadas independentemente, bem como também a distribuição eficaz de tais partes para execução em núcleos de processamento distintos, não são tarefas triviais considerando o AES (DAEMEN; RIJMEN, 2005; ZEGHID et al., 2007). Tais limitações, bem como a constante evolução da capacidade de processamento dos computadores e a disponibilização crescente de arquiteturas com paralelismo, são aspectos que motivam a investigação de novos métodos criptográficos alternativos aos padrões atuais da indústria, que favoreçam a exploração do paralelismo intra e inter-blocos.

1.2 Objetivos

Conforme mencionado nas seções anteriores, os ACs representam um mecanismo de paralelismo inerente que demonstra alto potencial de dispersão a quaisquer dados originalmente fornecidos ao sistema. Tais características são desejáveis neste contexto da investigação de novos métodos criptográficos e, assim sendo, o objetivo principal deste trabalho é a criação de um método de criptografia simétrica baseado em ACs que adote um conjunto heterogêneo de regras simples como chave secreta, a fim de reduzir o custo computacional das operações de cifração/decifração e explorar o potencial de paralelismo inter-blocos inerente a tais sistemas, mas sem comprometer a robustez e segurança do método. No cumprimento desse objetivo, é apresentado o método VHCA, com o detalhamento de suas etapas no processo de transformação criptográfica do texto claro para um texto cifrado (cifragem) e vice-versa (decifragem), bem como a disponibilização do código-fonte implementado.

Um dos maiores diferenciais do método VHCA, quando comparado a outros métodos criptográficos baseados em autômatos celulares, é o uso de regras elementares em seu processo evolutivo híbrido. Esta escolha é motivada pela simplicidade no cálculo de tais regras, quando comparadas a outras de maior raio. Embora o conjunto de regras elementares seja relativamente limitado, a expectativa da pesquisa foi assegurar a segurança da chave pela quantidade de combinações que podem ser geradas.

Obviamente, a análise de relevância do método VHCA precisa ser embasada em comparações devidas com os métodos atuais do estado da arte, bem como se faz necessário validar a robustez criptográfica do método proposto, comparando seu desempenho em diferentes tipos de análises com os resultados alcançados pelo seu predecessor, HCA, e pelo AES, que ainda é o principal método adotado para criptografia simétrica.

1.3 Hipótese e Questões de Pesquisa

A hipótese central do trabalho é que a combinação de diferentes regras elementares possibilita a construção de um método de criptografia de blocos baseado em autômatos celulares, capaz de alcançar níveis de segurança e robustez similares aos demais métodos da literatura e ainda possibilitar o paralelismo inter e intra-blocos. Essa hipótese está relacionada com as seguintes questões de pesquisa:

- A combinação de diferentes regras de AC elementares é suficiente para assegurar a segurança da chave criptográfica?
- Considerando a utilização de regras heterogêneas elementares, o VHCA apresenta segurança e robustez adequados para um método criptográfico baseado em blocos, principalmente quando comparado a outros métodos da literatura?
- Há ganho real de desempenho na adoção de regras heterogêneas elementares? Ou seja, o VHCA apresenta um desempenho computacional superior ao seu predecessor, o HCA, considerando implementações e condições de hardware equivalentes?

1.4 Contribuições

A contribuição principal deste trabalho foi a elaboração de um algoritmo de criptografia simétrica em bloco, baseado em regras elementares de ACs, e a realização de uma variedade de análises comparativas com outros métodos da literatura, a fim de validar a eficiência e robustez do método proposto. O critério inovativo da proposta deve-se ao fato de regras elementares serem historicamente consideradas impróprias para uso em métodos criptográficos por haver previsibilidade nos padrões gerados por estas (LEPORATI; MARIOT, 2013). Ao implementar o algoritmo de autômato celular altamente híbrido proposto neste trabalho, que explora uma relação combinatória entre tais regras elementares, os resultados encontrados reforçam a hipótese de que as limitações observadas no uso individual das mesmas foram superadas e o método provê segurança criptográfica.

1.5 Organização da Tese

O restante do texto está estruturado da seguinte forma:

- O Capítulo 2 descreve conceitos necessários para o entendimento do método proposto, bem como provê argumentos para sua relevância no estado da arte. Nele também é apresentada uma descrição do método HCA, predecessor do VHCA.
- A especificação do método VHCA é apresentada no Capítulo 3, com detalhamento das etapas para cifração e decifração, bem como dos mecanismos de paralelização.

- O Capítulo 4 lista os experimentos realizados para confirmar a viabilidade, desempenho, e qualidade criptográfica do VHCA, além de apresentar os resultados obtidos.
- Por fim, o Capítulo 5 detalha as contribuições do trabalho e apresenta sugestões para desenvolvimento futuro.

Capítulo 2

Fundamentação Teórica

Neste capítulo são listados os termos e trabalhos correlatos mais relevantes à temática desta tese. Inicialmente, são introduzidas algumas definições básicas de criptografia (Seção 2.1). Em seguida, é descrito o funcionamento dos autômatos celulares, incluindo conceitos relacionados ao seu uso em criptografia (Seção 2.2). Por fim, são apresentados trabalhos correlatos (Seção 2.3) ao novo método VHCA que será descrito no Capítulo 3.

2.1 Criptografia

A comunicação segura entre entidades humanas sempre foi um grande desafio, visto que manter a segurança de dados transmitidos é uma tarefa árdua, especialmente quando considerando a existência de meios físicos de transmissão que são compartilhados com várias outras entidades. Dada a relevância do tema, estudos levaram à definição das características fundamentais de um sistema que provê segurança da informação, formando a tríade CID (SAMONAS; COSS, 2014):

- **Confidencialidade (Sigilo)**: Dados só podem ser acessados por aqueles que tem permissão para tal.
- **Integridade**: É possível confiar que as informações recebidas não foram alteradas indevidamente.
- **Disponibilidade**: O acesso às informações necessárias pode ser feito em um prazo aceitável.

Nesse contexto, a criptografia surge como uma ferramenta para garantir segurança a dados sigilosos por meio da confidencialidade e integridade. Criptografia (palavra originada do grego "*kryptós*", que significa "escondido") é o estudo e aplicação de métodos que permitem converter uma mensagem legível (texto em claro) em uma mensagem ilegível (texto cifrado). A partir disso, a mensagem cifrada pode ser transmitida abertamente e, dependendo da forma de cifração adotada, uma entidade que detenha a chave criptográfica pode realizar o procedimento inverso e recuperar a mensagem original.

Existem dois tipos clássicos de criptografia: "simétrica" e "assimétrica". Na criptografia simétrica (de chave secreta), ilustrada na Figura 1, a chave usada para cifrar a mensagem precisa ser conhecida tanto pelo emissor original quanto pelo receptor desejado. Isso pode representar uma vulnerabilidade, caso a chave precise ser enviada por um canal inseguro de comunicação, visto que um terceiro participante poderia acessá-la e usá-la para forjar falsas mensagens. No cenário tradicional da criptografia simétrica, só há uma chave, a qual é usada tanto para cifrar quanto para decifrar a mensagem. Métodos dessa modalidade são o DES (Seção A.3.2) e o AES (Seção A.3.4).

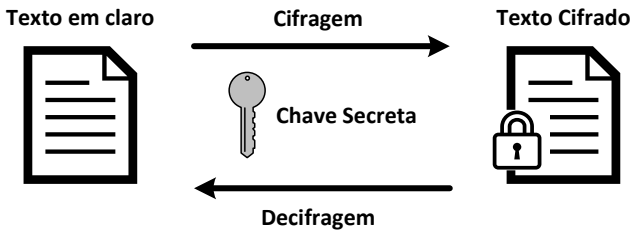


Figura 1 – Criptografia Simétrica. [Fonte: autor]

Na criptografia assimétrica (de chave pública), representada na Figura 2, há um par de chaves usado para realizar os procedimentos, uma pública e a outra privada. Considere um cenário onde temos um emissor que deseja enviar uma mensagem de cunho secreto a um dado receptor. Tal emissor deve cifrar o texto em claro utilizando a chave pública do receptor, chave esta que é divulgada abertamente. A chave privada, conhecida somente pelo receptor da mensagem, será usada por ele para decifrar tal mensagem. O método mais conhecido dessa modalidade é o algoritmo Rivest–Shamir–Adleman (RSA) (RIVEST; SHAMIR; ADLEMAN, 1978).

Uma breve descrição da história da criptografia é apresentada no Anexo A, na qual é introduzida uma linha do tempo com a evolução dos métodos utilizados para esse fim. Entretanto, dada a natureza simétrica do VHCA, a seguir é apresentada uma categorização dos algoritmos de chave secreta, bem como técnicas de criptoanálise empregadas para avaliar a robustez dos métodos criptográficos.

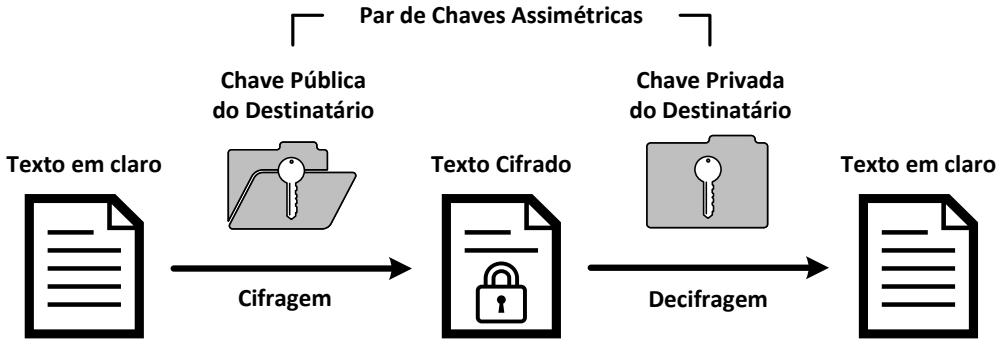


Figura 2 – Criptografia Assimétrica. [Fonte: autor]

2.1.1 Classificação de Métodos Simétricos

Na criptografia simétrica, costuma-se classificar os métodos de acordo com seu modo de funcionamento. Sob essa ótica, duas classes mais usuais emergem:

- **Cifra de Fluxo (*Stream Cipher*)**: Em algoritmos dessa classe, a cifração ocorre pela combinação, via operação XOR ou módulo (soma%N), de cada bit ou byte do texto em claro com seu equivalente em posição da "chave de fluxo" (*keystream*). O diferencial entre os algoritmos está no tipo de operações realizadas na criação da chave de fluxo a partir da chave criptográfica inicial. É desejável que a chave de fluxo apresente uma distribuição de bits o mais próxima possível da distribuição uniforme e, dado isso, é muito comum que tal chave seja criada por geradores de números pseudo-aleatórios (PRNG - *PseudoRandom Number Generator*). O OTP, descrito na Seção A.3.1, é um exemplo clássico de método de cifra de fluxo.

- **Cifra de Bloco (*Block Cipher*)**: O modo de execução para algoritmo de cifra em bloco baseia-se em repartir o texto em claro em blocos que tenham um mesmo tamanho pré-definido, e o método atua em cada bloco separadamente para cifrá-los utilizando a chave secreta. Há diversos "modos de operação", descritos na literatura, que são formas diferentes de recombinar e encadear os resultados da cifração feita separadamente em cada bloco, alguns deles são: *Electronic Code Book* (ECB), *Cipher Block Chaining* (CBC) , e *Output Feedback* (OFB). Tanto o DES, quanto o AES, são exemplos de cifração de bloco.

2.1.2 Análise de Robustez

A criptoanálise é um termo que abarca técnicas utilizadas na tentativa de quebrar, ou diminuir, a robustez de algoritmos criptográficos. Muitas dessas técnicas se baseiam na tentativa de detectar padrões entre os resultados de execuções distintas de tais algoritmos. Embora existam formas mais elaboradas de avaliação dos métodos criptográficos, como criptoanálise linear (MATSUI, 1993) e diferencial (BIHAM; SHAMIR, 1991), a seguir são apresentados os conceitos relacionados às análises de robustez realizadas neste trabalho.

2.1.2.1 Entropia

O conceito da entropia de dados foi definido por Claude E. Shannon, em (SHANNON, 1948). Esse princípio é utilizado para medir a diversidade inerente a um conjunto de dados, considerando a probabilidade de ocorrência P_i para um certo evento i, dada uma série de N eventos. No referido trabalho, foi proposta a seguinte equação para quantificar a entropia (H), dados tais parâmetros:

$$H = -\sum_{i=1}^{N} P_i \log_2 P_i \tag{1}$$

Uma versão normalizada da Equação (1) é dada por:

$$h = \frac{-\sum_{i=1}^{N} P_i \log_2 P_i}{\log_2 N} \tag{2}$$

Um valor h próximo a 1 indica uma alta diversidade nos dados testados, algo desejável no contexto criptográfico.

2.1.2.2 Avalanche

O efeito avalanche é desejável em modelos criptográficos. Ele indica que qualquer alteração no texto em claro ou na chave criptográfica, por menor que seja, terá um impacto significativo no texto cifrado resultante (FEISTEL, 1973). Esse mecanismo tem sido amplamente utilizado na avaliação de métodos criptográficos.

Considerando um algoritmo criptográfico f, o impacto da alteração é medido para um texto em claro T_{orig} e uma chave criptográfica K_{orig}. Considera-se também que a mudança no valor de um bit aleatório no texto em claro gera T_{alt}, assim como a troca no valor de um bit aleatório da chave gera K_{alt}. Dados esses parâmetros, tem-se as seguintes relações:

$$f(T_{orig}, K_{orig}) = C_{orig} \tag{3}$$

$$f(T_{alt}, K_{orig}) = C_{altTexto} \tag{4}$$

$$f(T_{orig}, K_{alt}) = C_{altChave} \tag{5}$$

A Equação 3 apresenta a execução do algoritmo com os parâmetros originais, T_{orig} e K_{orig}, resultando no texto cifrado C_{orig}. Na Equação 4, tem-se que $C_{altTexto}$ é o texto cifrado obtido após uma execução de f onde T_{orig} foi substituído por T_{alt}. Uma abordagem similar é exibida na Equação 5, onde K_{alt} foi utilizada em f ao invés de K_{orig}, resultando no texto cifrado $C_{altChave}$.

Para tal contexto, f apresenta efeito avalanche satisfatório se houver uma diferença de aproximadamente 50% no número de bits diferentes entre C_{orig} e $C_{altTexto}$, bem como também entre C_{orig} e $C_{altChave}$ (WEBSTER; TAVARES, 1985). Espera-se que a dispersão desses bits alterados seja o mais aleatória possível, a qual pode ser medida usando a entropia normalizada (h).

Um algoritmo que apresenta efeito avalanche satisfatório é menos vulnerável a tentativas mal-intencionadas de descobrir a chave criptográfica, mesmo em cenários onde o atacante é capaz de executar livremente o método de cifração para diversos textos claros de sua escolha e compará-los aos respectivos textos cifrados resultantes (tática conhecida na literatura como “ataque de texto em claro conhecido”) (MISHRA; GUPTA; PILLAI, 2011).

2.2 Autômatos Celulares

Autômatos Celulares (ACs) são sistemas dinâmicos em que um conjunto de elementos, chamado de espaço celular ou reticulado, representado por s, passa por evoluções discretas com relação à variável tempo (t) (ROZENBERG; BÄCK; KOK, 2012). Em sua versão mais simples (AC elementar), cada elemento (célula) tem um valor binário indicativo de seu estado em um determinado instante, que pode ser '0' (morto/inativo) ou '1' (vivo/ativo). A Figura 3 exemplifica o reticulado inicial ($t = 0$) de um AC unidimensional com 10 células ($N = 10$).

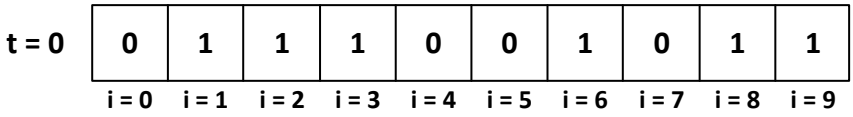


Figura 3 – Exemplo da configuração inicial de um AC. [Fonte: autor]

Sendo o AC da Figura 3 unidimensional, cada configuração do autômato, em um dado instante de tempo t, pode ser representada graficamente como um vetor de elementos s^t, onde cada elemento tem sua posição interna i ($0 \leq i < N$). Dadas estas variáveis, também é possível referenciar formalmente cada célula do AC unidimensional com a representação $s^t[i]$, como mostrado na Figura 4.

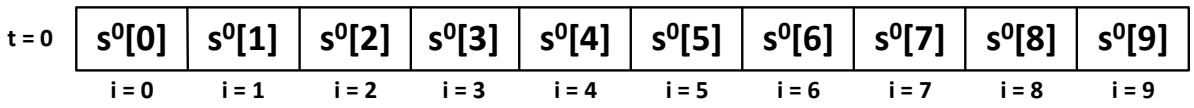


Figura 4 – Formalização da configuração inicial de um AC. [Fonte: autor]

Considerando a configuração do reticulado no instante inicial ($s^{t=0}$), é possível explorar as configurações subsequentes do AC. Para um instante t, onde $t > 0$, as células $s^t[i]$ podem assumir estados distintos do que tinham em $s^0[i]$. O estado que uma célula $s^t[i]$ assume em $s^{t+1}[i]$ é calculado usando uma regra de transição.

Constam na literatura diversos modelos de AC onde as regras de transição assumem caráter estocástico. Ou seja, em tais casos, o estado resultante de uma regra de transição é definido mediante uma probabilidade. Entretanto, para os fins desta tese, será considerada a definição mais simples de AC, onde suas regras de transição assumem caráter determinístico.

2.2.1 Regras de Transição

Conforme previamente mencionado, o cálculo evolutivo de ACs que gera uma configuração s^{t+1} a partir do reticulado da configuração s^t depende do uso de regras. No contexto desta tese, tais regras são funções determinísticas que mapeiam conjuntos de células adjacentes (vizinhanças) a um único valor novo para a célula na posição central da referida vizinhança, um conceito que será melhor detalhado a seguir.

Ao implementar um autômato celular, uma das primeiras definições necessárias é o valor de raio (r) para suas regras. O raio define, por consequência, o tamanho da vizinhança a ser considerado pela regra de transição, onde r é a distância máxima que uma vizinha pode estar da célula central. Por exemplo, considere arbitrariamente a célula $s^0[4]$ da Figura 4. Ao adotar uma vizinhança de raio 1, suas células vizinhas são $s^0[3]$ e $s^0[5]$. Analogamente, ao empregar uma vizinhança de raio 2, as células vizinhas de $s^0[4]$ são $s^0[2]$, $s^0[3]$, $s^0[5]$ e $s^0[6]$. E assim por diante. Em resumo, em um AC unidimensional, a vizinhança sempre é composta pela célula central e suas respectivas adjacentes, de ambos os lados, até atingir o limite determinado por r.

Definido o conceito de vizinhança, a Figura 5 exemplifica uma regra que pode ser utilizada na evolução de ACs. Na imagem, é possível observar o mapeamento (indicado por seta) de todas as combinações possíveis de vizinhança de raio 1 (indicadas por borda sólida) para os respectivos valores resultantes (estado da célula central no próximo instante de tempo).

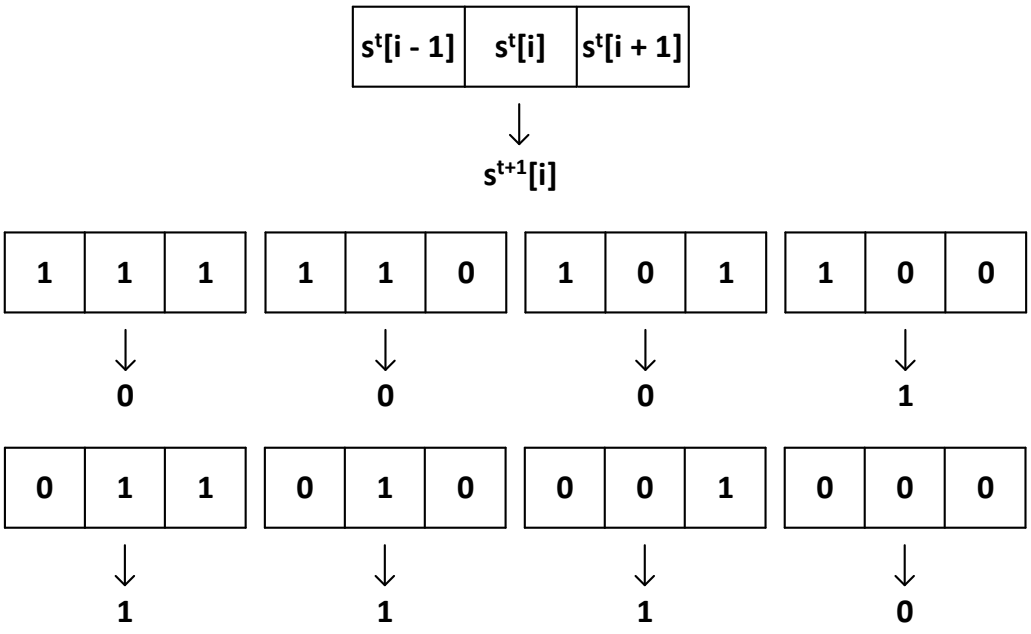


Figura 5 – Exemplo de regra de evolução para AC. [Fonte: autor]

A regra definida na Figura 5 é classificada como uma regra de raio 1 ($r = 1$), pois corresponde a uma função f tal que: $s^{t+1}[i] = f(s^t[i-1], s^t[i], s^t[i+1])$. É importante salientar a natureza determinística desse tipo de regra de transição, onde há uma única transição definida para cada combinação de valores que a vizinhança poderia assumir.

2.2.2 Condição de Limite

É importante destacar a necessidade de adaptar a fórmula genérica da vizinhança para as posições limítrofes (bordas) do reticulado. Por exemplo, pela fórmula descrita acima, o cálculo de $s^{t+1}[0]$ depende dos valores das células $s^t[-1]$, $s^t[0]$ e $s^t[1]$, mas $s^t[-1]$ não existe, pois o reticulado inicia pela célula $s^t[0]$ e não há outra à sua esquerda. Um cenário equivalente ocorre ao calcular o novo estado da última célula do reticulado, visto que não há outra célula à sua direita para compor a vizinhança.

Na literatura, o mais usual em tais situações é adotar uma "condição de limite", ou seja, uma estratégia que define os valores faltantes para o cálculo completo da evolução do reticulado. Exemplificando as definições com um reticulado de $N = 10$ posições ($s^t[0]$ ... $s^t[9]$), algumas das possíveis condições de limite são:

- **Limite de Valor Fixo**: Um valor arbitrário é definido para as células faltantes (ex.: considerar $s^t[-1] = 0$ e $s^t[10] = 0$).
- **Limite Reflexivo**: Ao alcançar um limite, os valores após o limite seriam uma cópia espelhada dos valores próximos ao mesmo (ex.: considerar $s^t[-1] = s^t[0]$ e $s^t[10] = s^t[9]$).
- **Limite Periódico**: Ao alcançar um limite, os valores após o limite seriam iguais aos valores do lado oposto do reticulado, como se o mesmo fosse circular/toroidal (ex.: considerar $s^t[-1] = s^t[9]$ e $s^t[10] = s^t[0]$).

A menos que explicitamente definido, em todos os algoritmos citados neste trabalho, deve-se considerar o uso do "Limite Periódico" como forma de tratamento padrão utilizada para selecionar as vizinhanças nos reticulados.

2.2.3 Exemplo de Evolução

Iniciando a evolução do reticulado da Figura 3, de s^0 para s^1, por meio da aplicação da regra expressa na Figura 5, têm-se os passos listados na Figura 6.

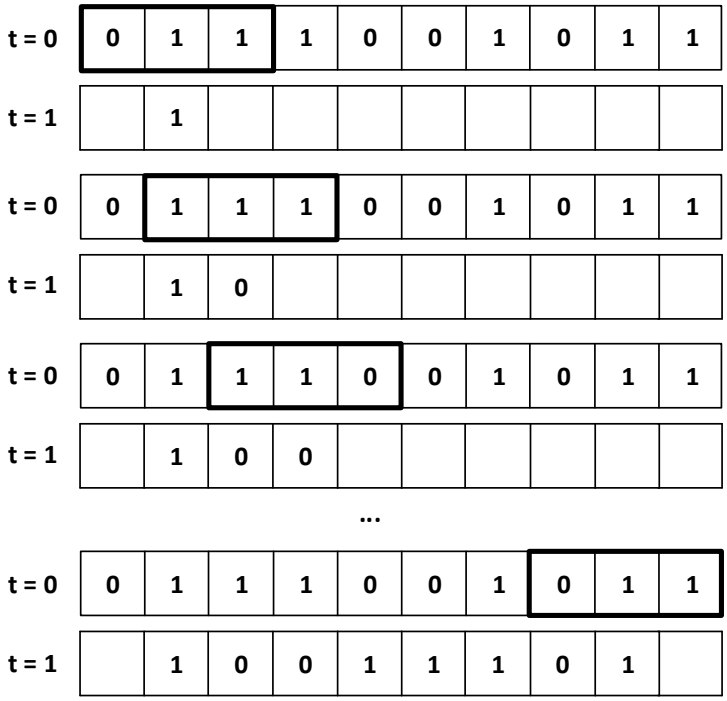


Figura 6 – Evolução parcial de um reticulado. [Fonte: autor]

Uma espessa borda preta é usada na Figura 6 para evidenciar a vizinhança em $t = 0$ que está sendo considerada para calcular o bit resultante preenchido em s^1. A separação do cálculo de s^1 em etapas demonstradas na referida figura tem mero fim didático, visto que, como todos os valores de s^0 são previamente conhecidos, tal cálculo poderia ser realizado em paralelo, sem necessidade de calcular sequencialmente cada um dos bits resultantes em s^1 para cada uma das vizinhanças de s^0.

Notamos que não consta o cálculo de valores para a primeira e última células do reticulado s^1, dado que este dependeria dos valores das células $s^0[-1]$ e $s^0[10]$, inexistentes no reticulado. Conforme mencionado na Seção 2.2.2, a condição de limite periódico será arbitrariamente definida como padrão para os cálculos nesta tese, por ser a mais usual na literatura e também ter sido utilizada no trabalho predecessor. Dessa forma, por meio dela, o restante da evolução do reticulado apresentado na Figura 6 ficaria como demonstrado na Figura 7.

Assim sendo, na Figura 7 fica evidenciado o fim da aplicação da regra apresentada na Figura 5, em todas as vizinhanças de s^0, resultando na configuração s^1.

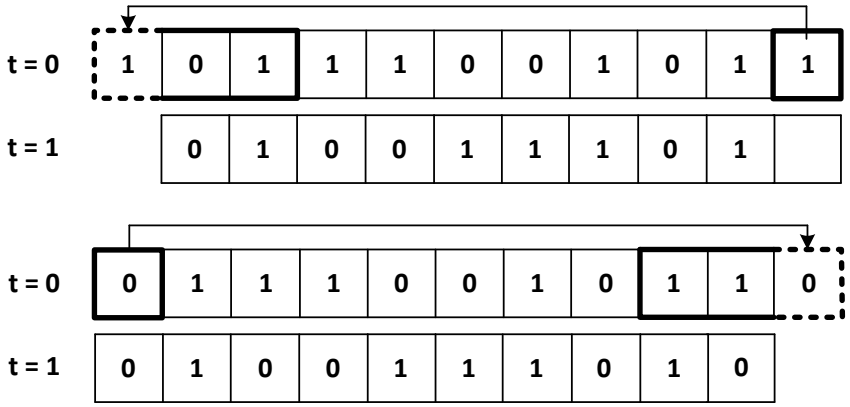


Figura 7 – Aplicação da condição de limite na evolução do reticulado. [Fonte: autor]

2.2.4 Notação de Wolfram

Visto que a vizinhança em um AC unidimensional, onde $r = 1$, é composta por três valores binários; e sabendo que há 8 (2^3) combinações possíveis de valores para tal arranjo, a regra de tal AC pode ser expressa pelos valores resultantes da aplicação de f a cada uma de tais vizinhanças. Wolfram (WOLFRAM, 2002a) utiliza-se deste fato para propor a nomenclatura das regras de ACs. A Figura 8 exibe como esse mecanismo é aplicado às 256 regras elementares (binárias e com vizinhança de raio 1). Na referida figura, Wolfram usa simbologia equivalente à descrita para a Figura 5, entretanto, omitindo setas de mapeamento e utilizando cor preta para indicar células '1' e cor branca para células '0'.

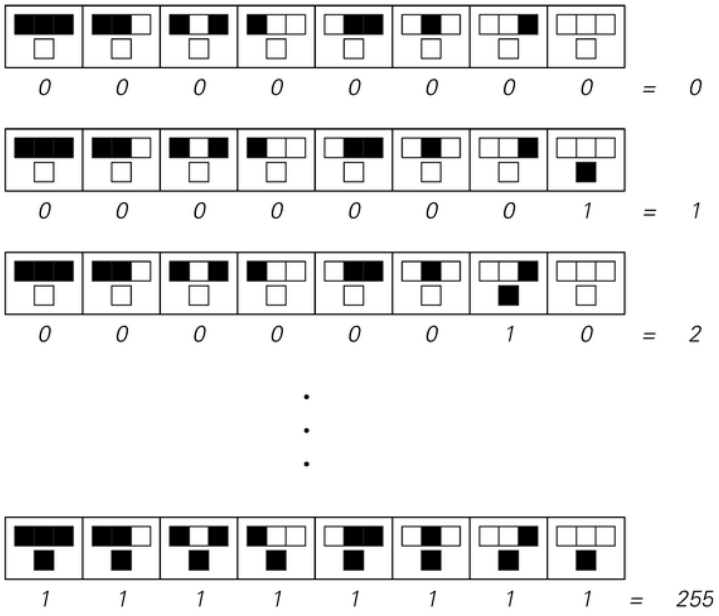


Figura 8 – Mecanismo de nomenclatura de Wolfram. [Fonte: (WOLFRAM, 2002a)]

Dado que em um AC unidimensional de raio r a vizinhança é composta por $(2 \times r) + 1$ células, e visto que cada célula assume valor binário, uma regra para tal AC é descrita por $2^{(2\times r)+1}$ transições de vizinhanças possíveis para cada bit resultante. Isso significa que, tomando uma ordenação padrão, é possível expressar regras por meio da sequência de bits resultantes de suas transições para as vizinhanças.

Considerando, por exemplo, a regra da Figura 5, e avaliando as combinações binárias em ordem decrescente de valor decimal correspondente ($(1,1,1) = 7_{10}, (1,1,0) = 6_{10}, \ldots, (0,0,1) = 1_{10}, (0,0,0) = 0_{10}$), a respectiva sequência de bits resultantes para tais vizinhanças é: $0,0,0,1,1,1,1,0$. Conhecidos os bits em tal sequência, eles podem ser concatenados e lidos como um número único, $00011110_2 = 30_{10}$. Assim, a função expressa na Figura 5 trata-se da regra 30 (R_{30}).

A notação Wolfram também permite expressar regras de $r > 1$ como sequências binárias e, consequentemente, valores numéricos. Assim, temos que regras com $r = 2$ podem ser expressas com sequências de $2^5 = 32$ bits, regras com $r = 3$ por sequências de $2^7 = 128$ bits, e regras com $r = 4$ por sequências de $2^9 = 512$ bits e assim por diante.

2.2.5 Pré-Imagens e Reversibilidade

Visto que a evolução do reticulado s^0 produz a configuração s^1 mediante a aplicação da regra 30, nesse contexto o reticulado s^0 é considerado uma pré-imagem de s^1. O cálculo de um reticulado s^{t+1} a partir de s^t utilizando uma regra R é sempre uma operação determinística; mas, dependendo de R, uma certa configuração s^{t+1} pode ser alcançável a partir de mais do que uma única pré-imagem.

Temos um exemplo extremo disso, por exemplo, na regra R_{255}. Nela, qualquer vizinhança converge para o bit resultante '1'. Sendo assim, qualquer que seja o reticulado inicial s^0, o seu sucessor s^1 obtido à partir da R_{255} será '111...'. E, visto que todas as configurações futuras desse AC serão iguais a s^1, nenhum outro estado de reticulado pode ser alcançado. No contexto de ACs a literatura refere-se às configurações tornadas inalcançáveis devido à escolha da regra como "Jardim do Éden" (MOORE, 1962).

Entretanto, existe também o conceito de "reversibilidade", uma propriedade que ocorre quando todas as configurações possíveis do reticulado são alcançáveis por meio da evolução do AC. Isso só ocorre quando a escolha da regra R garante a existência de uma, e somente uma, pré-imagem válida s^t para qualquer que seja a configuração s^{t+1}. Ou seja, quando a operação de cálculo de pré-imagem é uma operação determinística.

Tanto o método VHCA, objeto de proposição nesta tese, quanto seu predecessor HCA, que é apresentado na Seção 2.3.3, são fundamentados na escolha de regras e operações que garantem essa propriedade da reversibilidade.

2.2.6 Sensitividade de Regras

Entre as regras de autômatos celulares observa-se um comportamento peculiar a algumas delas. Mediante explicado na Seção 2.2.1, uma regra de $r = 1$ pode ser descrita como $s^{t+1}[i] = f(s^t[i-1], s^t[i], s^t[i+1])$. Ou seja, para cada vizinhança de três células consecutivas (esquerda, centro, direita) no instante t a regra é uma função que determina o estado da célula na posição central i da vizinhança na configuração seguinte $t+1$.

Algumas regras seguem um padrão onde, comparando as transições expressas pela regra, nota-se que o valor da célula de uma posição específica da vizinhança em t é determinante para o valor resultante $s^{t+1}[i]$. A Figura 9 traz exemplos de tais regras.

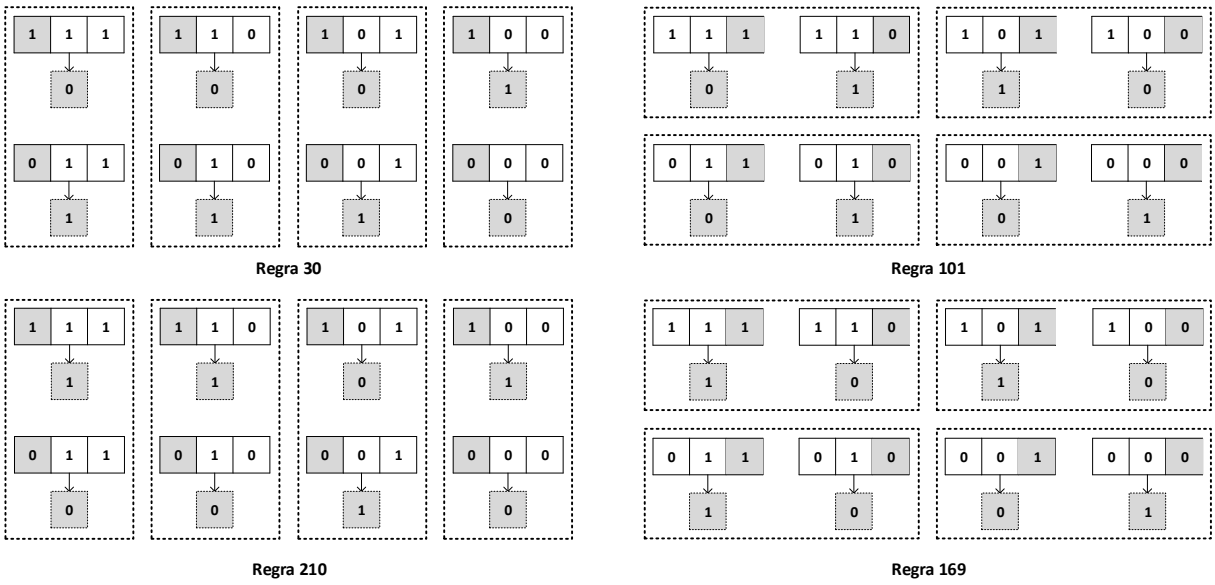


Figura 9 – Exemplo de regras com sensitividade. [Fonte: autor]

A Figura 9 apresenta as transições de quatro regras selecionadas arbitrariamente, uma em cada quadrante da figura. As regras 30 e 120 exibidas ao lado esquerdo da figura exemplificam o conceito de sensitividade à esquerda, enquanto as regras 101 e 169 apresentadas ao lado direito da figura demonstram a sensitividade à direita. Nas transições das referidas regras foi adicionado um destaque de fundo cinza tanto aos estados resultantes quanto também à célula de cada vizinhança que está ao lado dito sensitivo da regra (célula esquerda para regras sensíveis à esquerda e vice-versa). Nota-se também que cada uma das regras teve suas oito transições agrupadas, de duas a duas, por bordas tracejadas. Esse agrupamento facilita a comparação visual entre transições onde as vizinhanças se diferenciam apenas pelo valor da célula que está na posição da sensitividade.

O conceito de sensitividade é fundamental na descrição do método VHCA, objeto de proposta desta tese. E tal conceito pode ser expandido para regras com $r > 1$. No caso, sempre que o conceito de sensitividade à esquerda ou à direita for citado, fica subentendida a sensitividade à célula limítrofe da vizinhança na direção indicada. Ou seja, considere sensitividade ao valor da célula mais à esquerda da vizinhança ou ao valor da célula mais à direita da vizinhança, de acordo com a indicação. Para conhecimento, existem também regras com sensitividade simultânea a ambos os lados (sensitividade bilateral), que seguem um princípio correspondente ao dos exemplos apresentados acima.

Com relação à notação de Wolfram apresentada na Seção 2.2.4, é necessário salientar uma diferença com relação a regras sem sensitividade. Enquanto uma regra convencional é expressa com $2^{(2\times r)+1}$ bits, uma regra com sensitividade a um dos lados pode ser representada com metade disso, $2^{(2\times r)}$ bits.

Isso ocorre pois, em regras com sensitividade à direita, os quatro primeiros bits da regra serão os valores opostos aos quatro últimos. Na R_{210}, por exemplo, expressa como '11010010', uma representação possível é dada por '1101', e os bits restantes podem ser recuperados ao concatenar os valores opostos dos bits em questão. Visto que são valores binários, a oposição também pode ser expressa pela operação complemento ($^{-}$), onde temos: $1101\overline{1101} = 11010010$. Ao mesmo tempo, para regras com sensitividade à esquerda, bits sequenciais na representação de Wolfram serão sempre valores opostos um ao outro. Na R_{169}, por exemplo, expressa como '10101001', uma representação reduzida ocorre pela seleção do primeiro bit de cada par consecutivo na sequência, no caso '1110'. A representação original da regra pode ser recuperada ao inserir o valor oposto na frente de cada bit da representação reduzida: $1\overline{1}1\overline{1}1\overline{0}0\overline{} = 10101001$. Apesar de ter sido exemplificada em regras de $r = 1$, essa lógica é observada independente do tamanho do raio.

2.2.7 Classes de Comportamento

Mediante o uso de diferentes regras é possível obter os comportamentos mais diversos, mesmo considerando um modelo básico de autômato celular. De fato, tentativas foram feitas de classificar o comportamento que cada uma das 256 regras elementares impõe a ACs quando aplicadas a todas as células do reticulado de forma homogênea. A primeira tentativa conhecida de classificação foi divulgada em (WOLFRAM, 1984). Embora expansões e melhorias dessa classificação inicial tenham sido propostas por outros pesquisadores, vide (AGUIAR, 2014; BORRIELLO; WALKER, 2017; STRICKER, 2023; ALFARO; SANJUÁN, 2024), a proposta de Wolfram ainda se mantém relevante. A classificação de Wolfram separa o comportamento das regras dos ACs em quatro classes:

- **Classe 1 - Comportamento Fixo**:
  A evolução do AC tende rapidamente a um reticulado uniforme (todas as células com mesmo estado), independente da configuração inicial.

- **Classe 2 - Comportamento Cíclico ou Periódico**:
  Regras dessa classe geram dinâmicas de repetição, seja pelo surgimento de estruturas simples que começam a se repetir periodicamente, ou por uma configuração "final" que é alcançada e passa a se repetir.

- **Classe 3 - Comportamento Caótico**:
  Surge comportamentos de aparência caótica ao aplicar regras dessa classe. As dinâmicas complexas exibidas são altamente imprevisíveis e dependentes da configuração inicial do AC.

- **Classe 4 - Comportamento Complexo**:
  Regras dessa classe geram evoluções mistas entre ordem e caos. O comportamento gerado por regras dessa classe apresenta estruturas que se repetem, porém com tamanhos e posições variantes gerando dinâmicas complexas.

Tais dinâmicas podem ser observadas na Figura 10, que contém exemplos da aplicação de regras elementares atribuídas a cada uma das classes, mediante o estudo de Wolfram. É relevante notar o comportamento das regras de classes 3 e 4, que demonstra uma aparente aleatoriedade, característica esta que é desejável em aplicações criptográficas.

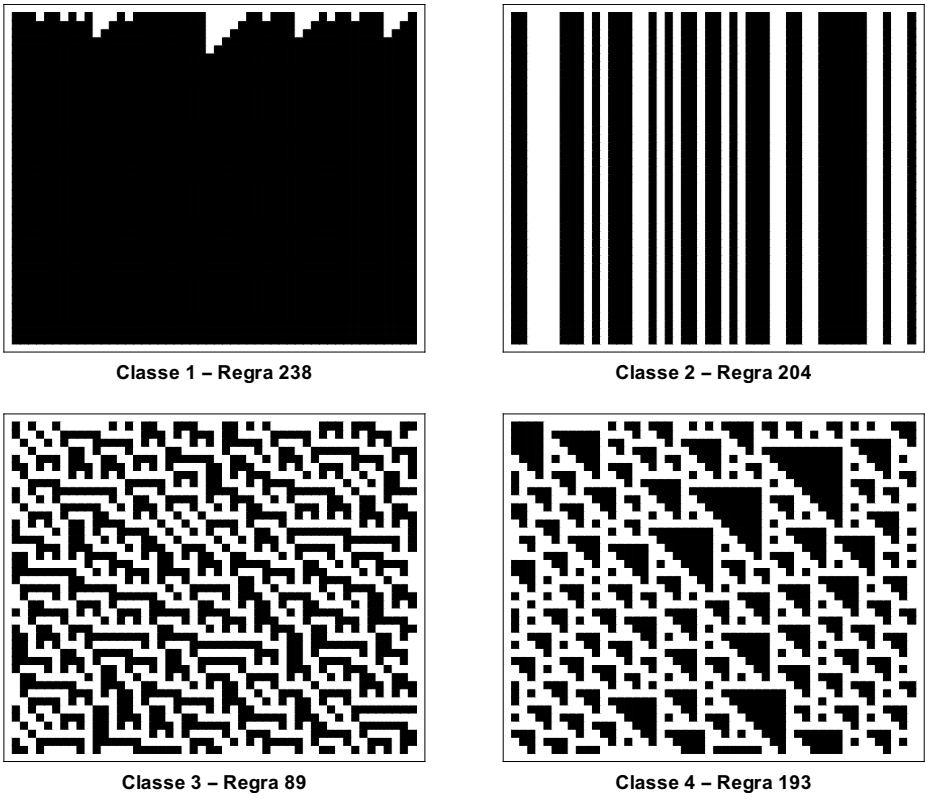


Figura 10 – Exemplos das classes de Wolfram. [Fonte: autor]

Embora as dinâmicas apresentadas na Figura 10 sejam de regras específicas sendo aplicadas de igual forma a todas as células do reticulado, característica essa que define um AC "homogêneo", também há ACs que não seguem essa tendência e são nomeados "heterogêneos" ou "híbridos".

2.2.8 Autômatos Celulares Híbridos

Em um AC heterogêneo, mais de uma regra é utilizada na evolução do reticulado, de forma que diferentes regras podem ser utilizadas para evoluir trechos distintos do reticulado (nunca havendo sobreposição), conforme exemplo na Figura 11.

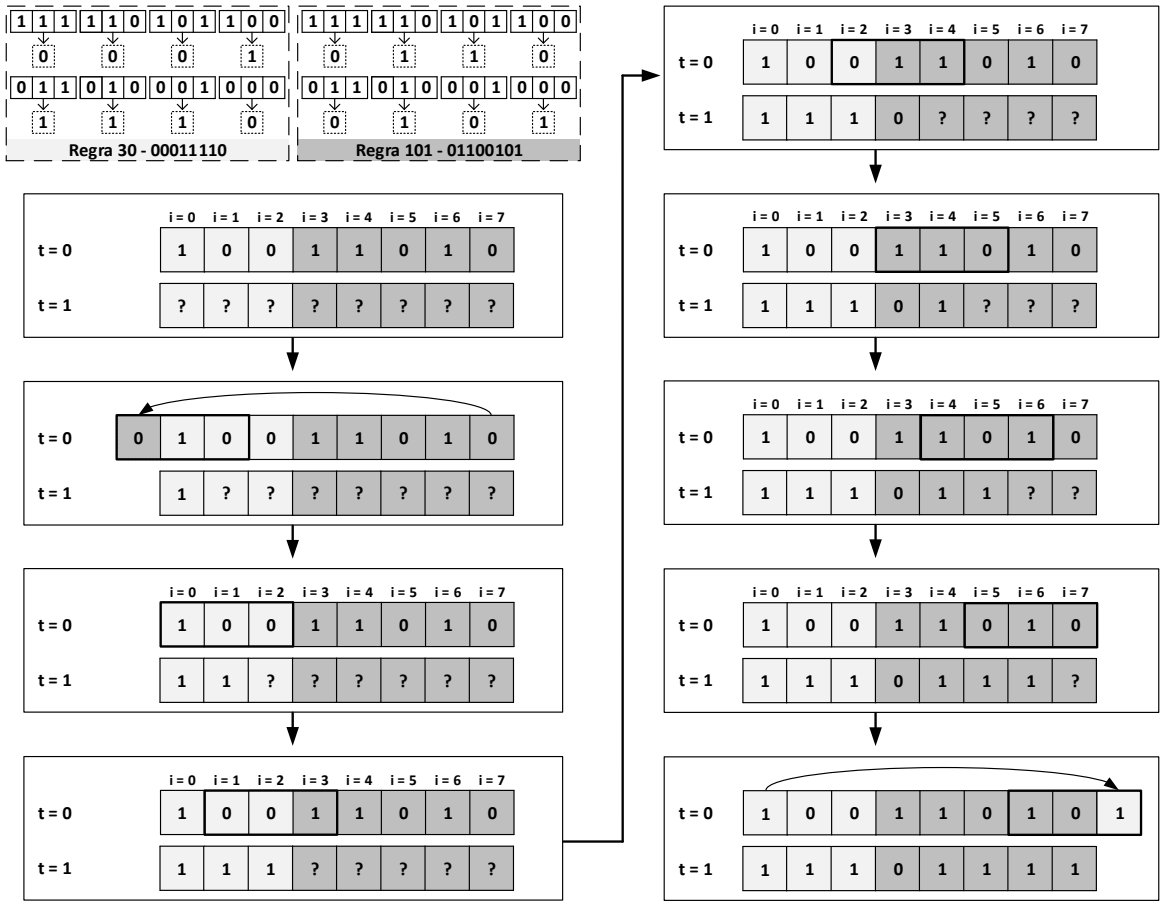


Figura 11 – Evolução de um reticulado híbrido. [Fonte: autor]

O AC da Figura 11 é considerado híbrido pois o reticulado é evoluído pela aplicação de mais de uma regra. Neste caso específico, é definido arbitrariamente que as três células das posições $i = 0$, $i = 1$ e $i = 2$ são evoluídas pela regra 30, enquanto a regra 101 será aplicada nas cinco células restantes das posições $i = 3$, $i = 4$, $i = 5$, $i = 6$ e $i = 7$. O tamanho reduzido do reticulado e o esquema de cores diferente entre as células evoluídas com regras diferentes são apenas para fins didáticos do exemplo.

A primeira configuração do reticulado é apresentada no canto superior esquerdo da figura, logo abaixo da definição das regras utilizadas. A evolução das células de $t = 0$ é apresentada sequencialmente, da esquerda para a direita, iniciando pela evolução da posição $i = 0$ até a posição $i = 7$. A vizinhança considerada para cada evolução é destacada por ênfase de borda, e cada seta indica a próxima etapa na sequência.

É importante destacar que, assim como feito na Figura 6, a quebra do procedimento evolutivo na Figura 11 também é meramente didática. Visto que os valores das células de $t = 0$ são conhecidos, não há uma ordem correta para o cálculo dos valores para $t = 1$ e, na verdade, todos os valores podem ser calculados simultaneamente, o que reflete um grande atrativo para utilizar autômatos celulares: o paralelismo inerente do sistema.

Além dos ACs unidimensionais há também outras classes de autômatos celulares com formas diversas de representação, entre elas constam os ACs bidimensionais (PACKARD; WOLFRAM, 1985) e extensões tridimensionais do modelo que têm sido utilizadas na literatura (BARCA et al., 1994; RAABE, 1999; ZHANG et al., 2012; CHEN; GUILLEMOT; GANDIN, 2016). Assim como para ACs unidimensionais, cada uma dessas adaptações do modelo tem sua definição própria de vizinhança e regras de evolução.

Também é relevante apontar que, conceitualmente, nada impede a existência de um AC evoluído por mais que duas regras, ou mesmo que as regras sejam trocadas após cada etapa de evolução, conquanto haja uma definição clara de qual regra será aplicada a cada célula de posição i para cada instante t. Este conceito de ACs híbridos é fundamental na definição do método VHCA, bem como no entendimento do método predecessor HCA.

2.3 Criptografia baseada em Autômatos Celulares

A possibilidade de utilizar ACs para gerar comportamentos complexos inspirou pesquisadores a descrever métodos criptográficos baseados nesse mecanismo, em suas diversas formas. As seções seguintes descrevem alguns dos métodos criptográficos baseados em ACs, organizando-os de acordo com a forma como o autômato celular é empregado no processo de cifragem/decifragem.

2.3.1 AC como Gerador de Chaves Secretas

O primeiro trabalho feito na área de criptografia usando ACs foi divulgado em 1985 (WOLFRAM, 1985) por Stephen Wolfram. Na versão inicial de seu método, Wolfram propõe que a regra 30 seja usada para, a partir de uma semente original, gerar uma sequência pseudo-aleatória por meio da evolução de um AC, e utilizar tal sequência como chave criptográfica em uma cifra OTP, conforme Figura 12.

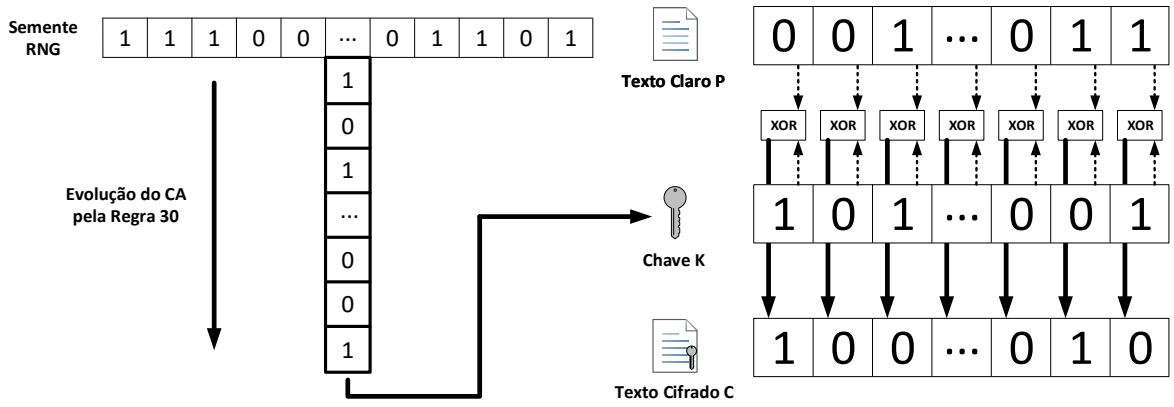


Figura 12 – Cifra de Wolfram. [Fonte: autor]

Na época em que a proposta foi feita, Wolfram ainda não havia encontrado evidências de que a Regra 30, por si só, não provê boa segurança criptográfica. Coube aos trabalhos posteriores (MEIER; STAFFELBACH, 1991a) e (KOC; APOHAN, 1997) demonstrar que havia previsibilidade no padrão de evolução de tal regra, uma fragilidade do método. Esse método de Wolfram é nitidamente classificado como uma cifra de fluxo e, desde então, outros trabalhos nessa linha de pesquisa foram divulgados, por exemplo:

- **(HORTENSIUS et al., 1989)**: São analisadas as propriedades criptográficas das regras 30, 45, 90 e 150. Após tal avaliação, é sugerido o uso de um AC híbrido utilizando as regras 90 e 150 para a geração de vetores pseudoaleatórios. Testes de ciclo das regras, bem como o estudo da correlação estatística entre bits vizinhos, foram utilizados para avaliar o método.

- **(TOMASSINI; PERRENOUD, 2001)**: Um algoritmo genético é utilizado para avaliar a qualidade criptográfica de todas as regras possíveis nos raios 1 e 2. As regras com melhor desempenho de cada raio são selecionadas para investigar duas instâncias do método, uma de $r = 1$ e outra de $r = 2$. Para $r = 1$, as melhores regras encontradas foram 90, 105, 150 e 165. Essas regras são então permutadas em uma atribuição pseudo-uniforme entre as posições de células do reticulado inicial para um AC híbrido, na intenção de que uma regra possivelmente diferente (entre as 4) possa ser atribuída a cada célula. Após a evolução do AC híbrido, tem-se um novo vetor pseudoaleatório. O mesmo comportamento é implementado para $r = 2$, porém com regras distintas.

A maioria dos algoritmos dessa linha de pesquisa foca na evolução convencional de ACs homogêneos ou híbridos como forma de obter o vetor pseudoaleatório utilizado em seus processos de cifragem e decifragem.

2.3.2 AC como Método Criptográfico

O método de Gutowitz (GUTOWITZ, 1993) é a primeira referência de criptografia simétrica onde são explorados os conceitos da reversibilidade de ACs e da sensitividade de regras. Essa abordagem utiliza diretamente o mecanismo de evolução dos ACs como forma de cifração do texto em claro, um princípio que também foi utilizado pelo HCA e pelo VHCA.

Dado um reticulado s em um instante de tempo $t+1$, um mecanismo de reversibilidade deve garantir que a pré-imagem s^t possa ser calculada sem ambiguidade para qualquer que seja a configuração s^{t+1}. A ambiguidade no cálculo de pré-imagem surge do fato de que, considerando um índice genérico i, para cada estado $s^{t+1}[i]$ geralmente não há meios de determinar qual foi a transição utilizada para gerá-lo, mesmo tendo ciência da regra que define tais transições. A Figura 13 exemplifica essa condição.

A regra 217, apresentada na parte superior da Figura 13, foi escolhida arbitrariamente, mas várias outras regras exemplificariam a situação apresentada. Na parte inferior da figura, ao centro, está exibida uma representação genérica da vizinhança em s^t para um certo índice central i: $s^t[i-1]$, $s^t[i]$ e $s^t[i+1]$. Aos lados, têm-se exemplos de etapas do cálculo de pré-imagem onde fica visível que, mesmo ciente da regra aplicada e do estado resultante $s^{t+1}[i]$, não é possível determinar sem ambiguidade os estados das células de vizinhança na pré-imagem. Isso ocorre pois, tanto para o estado resultante "0", quanto para "1", há mais de uma transição da regra que resultaria em tal estado. Para o estado resultante "0", por exemplo, tanto os valores "101" quanto "010" ou "001" na vizinhança da pré-imagem satisfariam esse resultado.

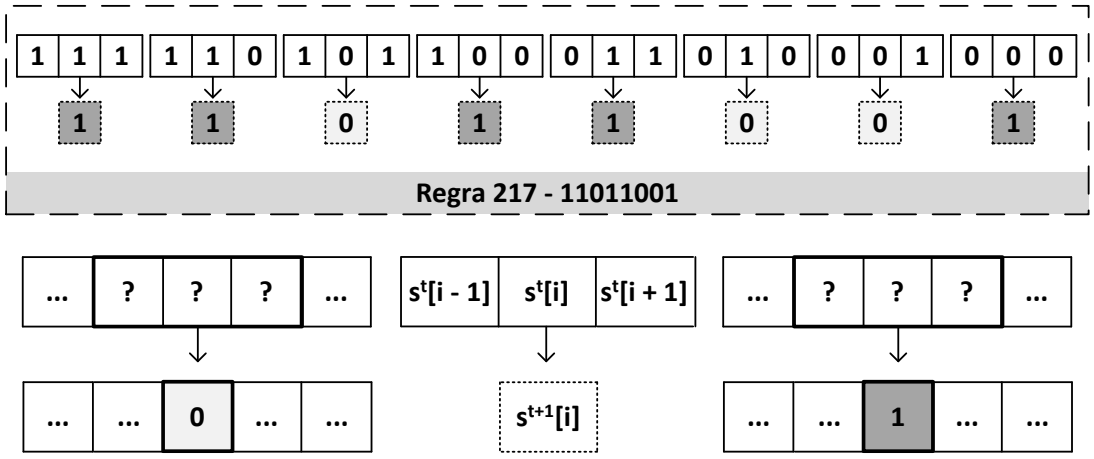


Figura 13 – Exemplo de ambiguidade no cálculo de pré-imagem. [Fonte: autor]

Nota-se então que não há um mecanismo simples que permita a reversibilidade para qualquer regra de AC. Entretanto, as regras com sensitividade são um subconjunto que segue uma lógica específica, por meio da qual o bit determinante (na posição de sensitividade da pré-imagem) sempre influencia diretamente o estado resultante na imagem. De forma que, tomando por exemplo a regra 30 (sensitividade à esquerda) e dado o conhecimento dos valores de bits não-determinantes da vizinhança (supondo "10"), tem-se a situação da Figura 14.

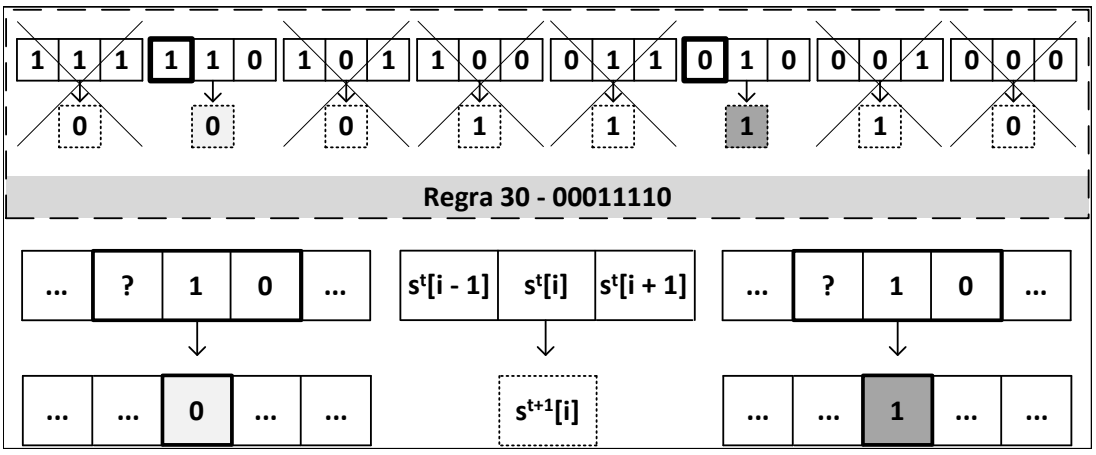


Figura 14 – Cálculo de bit da pré-imagem na posição com sensitividade. [Fonte: autor]

Na Figura 14, seis das oito transições da regra são desconsideradas (indicado pelo risco em xis) visto que seus bits de vizinhança da pré-imagem não correspondem aos bits conhecidos "10". Há duas transições que não foram descartadas (nas quais a vizinhança só difere pelo valor do bit determinante), e nota-se que o estado resultante é diferente entre elas; isso sempre acontecerá para regras com sensitividade, visto que nelas, por definição, as transições de vizinhanças diferindo apenas pelo bit determinante sempre resultam em estados distintos.

Neste caso, dado que o estado resultante é conhecido, a única transição que satisfaz a condição no reticulado ao lado esquerdo da figura é $f(1,1,0) = 0$; com esse artifício definiu-se, sem ambiguidade, que "1" é o estado da célula determinante na pré-imagem à esquerda, que antes era desconhecido. De forma análoga, tem-se que "0" é o estado da célula determinante para a pré-imagem da direita. Esse cálculo é adaptável tanto para regras com sensitividade à esquerda quanto à direita.

Pelo mecanismo apresentado, sempre é possível calcular o valor de um dado bit desconhecido da pré-imagem, caso:

- ❏ A regra de evolução dessa vizinhança seja conhecida e apresente sensitividade à esquerda ou à direita;
- ❏ Deseja-se descobrir o estado da célula determinante da vizinhança selecionada;
- ❏ Os estados das células restantes dessa vizinhança são conhecidos.
- ❏ O estado resultante da célula central da vizinhança na imagem é conhecido.

Dessa forma, se o reticulado inteiro foi evoluído por meio de regras conhecidas, com sensitividade a uma mesma direção, a partir do momento em que essa lógica foi usada para calcular o estado de uma dada célula da pré-imagem, o estado da célula "seguinte" poderá também ser calculado em sequência, pelo mesmo mecanismo. E esse procedimento pode ser repetido até terminar o cálculo da pré-imagem inteira, conforme a Figura 15.

A Figura 15 exemplifica o cálculo de uma pré-imagem pelo mecanismo acima detalhado considerando a regra 149, que exibe sensitividade à direita, e considerando que os primeiros dois bits "01" da pré-imagem são previamente conhecidos. Em **(a)**, o primeiro bit determinante a ser calculado está na vizinhança $(0,1,?)$ e o estado resultante da aplicação da regra a essa vizinhança é "0". Ao comparar essa condição às transições da regra 149, a única transição condizente seria $f(0,1,1) = 0$; logo, o bit desconhecido da vizinhança tem valor "1", conforme atribuído em **(b)**. Visto que os bits são calculados no sentido da sensitividade, à medida que as condições de cálculo são satisfeitas, o cálculo do bit determinante da nova vizinhança destacada em **(b)** pode ser efetuado. Por meio da transição $f(1,1,0) = 0$, que satisfaz $f(1,1,?) = 0$, o estado "0" do bit desconhecido é obtido. Esse mecanismo é repetido entre as etapas **(c)** e **(e)** até que a pré-imagem esteja completa em **(f)**.

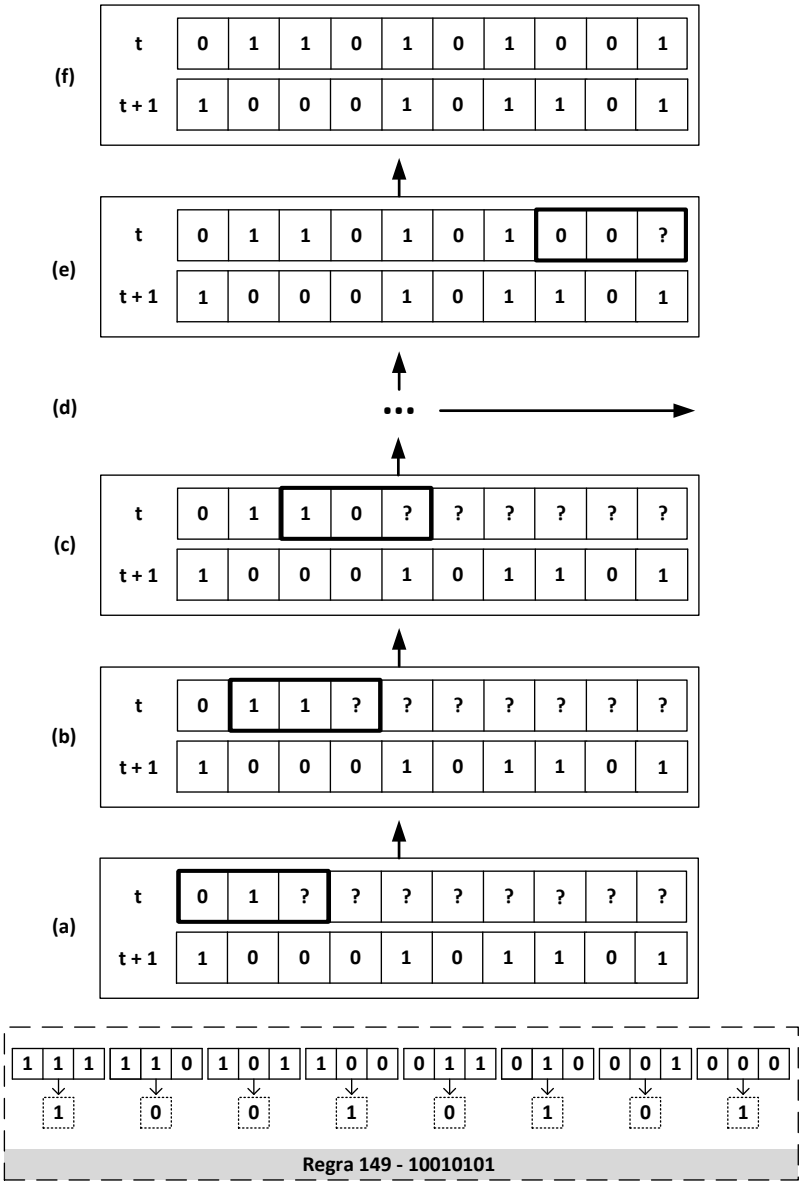


Figura 15 – Cálculo de pré-imagem por regra com sensitividade. [Fonte: autor]

Esse mecanismo de reversibilidade permite a Gutowitz calcular uma pré-imagem para qualquer reticulado inicial considerando a aplicação homogênea de uma regra com sensitividade à esquerda ou à direita. A garantia de viabilidade e determinismo desse cálculo de pré-imagem torna possível utilizar essa operação de evolução reversa para cifrar um reticulado contendo o texto em claro. Assim sendo, a decifração envolve aplicar evoluções convencionais do AC.

Entretanto, cabe uma ressalva fundamental referente ao cálculo de pré-imagem utilizado por Gutowitz. Na Figura 15 só foi possível começar o cálculo da pré-imagem pois dois valores iniciais da mesma foram supostos, "01" no caso. Porém, em situações reais, a menos que se defina uma solução específica, todos os valores da pré-imagem são desconhecidos ao iniciar cada iteração da evolução reversa. A proposta de Gutowitz para resolver essa questão é que os bits iniciais de cada pré-imagem sejam gerados aleatoriamente e adicionados no reticulado. Feito isso, considerando uma regra de raio r, há um acréscimo de $2 \times r$ bits no tamanho do reticulado a cada etapa da decifração, conforme ilustrado na Figura 16, onde observa-se a adição de uma nova célula em cada lado do reticulado a cada instante anterior. Essa estratégia visa garantir a obtenção de uma pré-imagem adequada, sem a necessidade de validação do critério de contorno (GUTOWITZ, 1993).

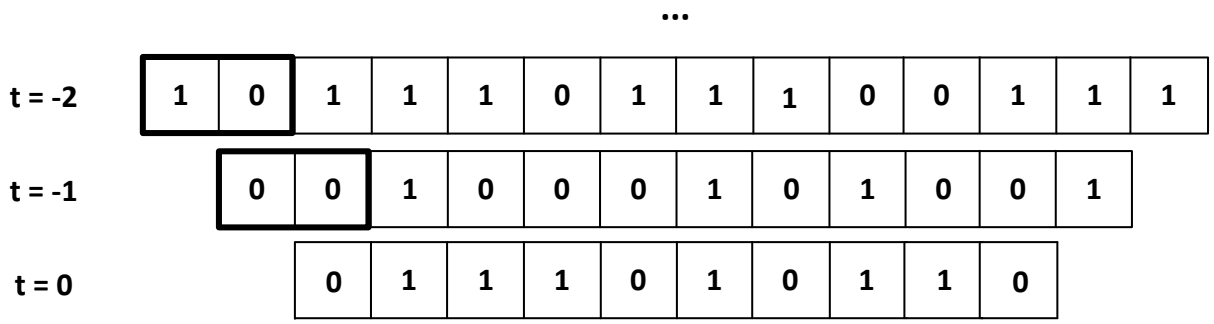


Figura 16 – Cifração por cálculo de pré-imagem no método de Gutowitz usando a regra 149. [Fonte: autor]

Na Figura 16 deve-se considerar que a configuração $t = 0$ contém o texto em claro, e as demais configurações ($t = -1$, $t = -2$, ...) são cálculos sucessivos de pré-imagens considerando uma evolução pela regra 149, que apresenta sensitividade à direita. Nesse contexto também, os valores das células em destaque foram preenchidos aleatoriamente para possibilitar o cálculo dos bits seguintes. Fica evidente o aumento gradual no tamanho do reticulado, que foi previamente citado.

O método de Gutowitz, apesar de seu critério inovador para a época, recebeu questionamentos devido ao aumento no tamanho do texto cifrado com relação ao texto em claro. Esse aumento poderia ser grande a ponto de inviabilizar o seu uso, dependendo da aplicação desejada. A fim de contornar tal limitação, (MACÊDO, 2007) propôs uma nova abordagem de criptografia, denominada *Hybrid Cellular Automata* (HCAs), que adota diferentes regras de transição do AC para calcular as pré-imagens sem o aumento do texto cifrado. Como o algoritmo proposto nesta tese é inspirado no HCAs, ele será melhor detalhado a seguir.

2.3.3 *Hybrid Cellular Automata* (HCA)

Esse método criptográfico, assim como o de Gutowitz, utiliza o mecanismo intrínseco de evolução dos ACs como base para os procedimentos de cifração e decifração, mediante a reversibilidade de reticulados. Ele também utiliza regras com sensitividade para garantir a existência de pré-imagem única para qualquer configuração possível. Por outro lados, os modelos se diferem em relação ao tipo de autômato celular e critério de contorno de reticulado adotados, bem como pela quantidade de possíveis pré-imagens que podem ser obtidas a partir de um mesmo reticulado. No modelo de Gutowitz, o AC é homogêneo, o contorno do reticulado não é periódico e existem diferentes pré-imagens possíveis para cada reticulado, dependendo dos valores escolhidos para os bits iniciais. No HCA é utilizado um AC híbrido, de contorno periódico, onde define-se que um certo conjunto de células vizinhas será evoluído por uma segunda regra com um tipo especial de sensitividade. Esse conjunto de células, diferente a cada iteração, é nomeado "região de borda"; e sua existência garante a manutenção do tamanho de texto original, visto que, para cada configuração do AC, ela garante a existência de uma única pré-imagem de mesmo tamanho.

O modelo do HCA foi expandido para a cifragem de reticulados bidimensionais em (JÚNIOR, 2010), resultando no algoritmo *Two-Dimensional Hybrid Cellular Automata* (THCA), e tridimensionais em (LIMA, 2012), originando o algoritmo *Three-Dimensional Hybrid Cellular Automata* (3DHCA). A motivação para esses modelos foi a aplicação em cifração de imagens em preto e branco (THCA) e coloridas (3DHCA). Posteriormente, o método HCA foi modificado para utilizar uma estrutura mais flexível de regras de transição baseada em redes complexas (MACÊDO; OLIVEIRA; RIBEIRO, 2014). As seções seguintes detalham o mecanismo de funcionamento do modelo original, que serviu de base para a nossa abordagem (VHCA).

2.3.3.1 HCA - Cálculo de Pré-Imagem

A reversibilidade de um AC é garantida quando, dado um reticulado (s) e um instante genérico de tempo t, qualquer configuração s^t sempre terá uma única pré-imagem s^{t-1} e será também a pré-imagem de outra configuração s^{t+1} (MOORE, 1962). Assim, todas as configurações são alcançáveis e o cálculo de pré-imagem é determinístico, ou seja, não há "Jardim do Éden".

Conforme apresentado na Seção 2.3.2, o método de Gutowitz propõe um modelo com reversibilidade onde é possível calcular uma pré-imagem para qualquer reticulado; porém, essa pré-imagem é acrescida em tamanho por $2 \times r$ bits. E esse incremento ocorre a cada cálculo sucessivo de pré-imagem.

Visto que os valores dos bits em s^t são previamente conhecidos, o acréscimo no tamanho do reticulado s^{t-1}, com relação ao citado, ocorre porque, para calcular o valor de uma dada célula de s^{t-1}, é necessário ter ciência dos valores dos outros $2 \times r$ bits da vizinhança em que essa célula é o bit determinante devido à sensitividade da regra. A Figura 17 exibe a vizinhança para uma evolução usando a regra R com $r = 4$, que é o tamanho de raio definido para regras do HCA.

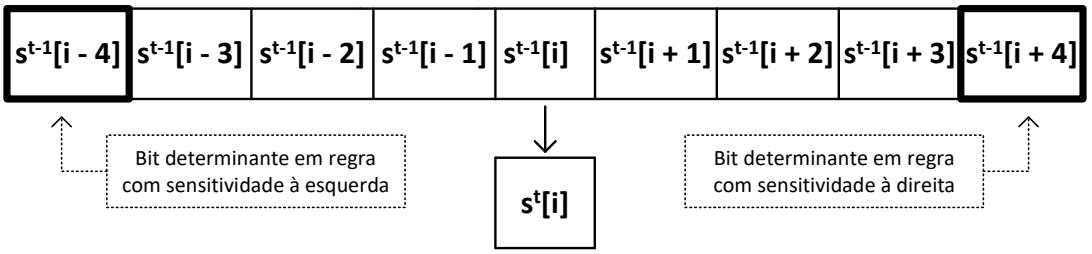


Figura 17 – Vizinhança e exemplo de bit com sensitividade para $r = 4$. [Fonte: autor]

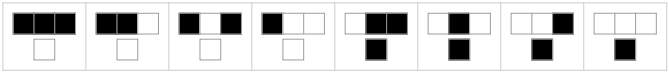
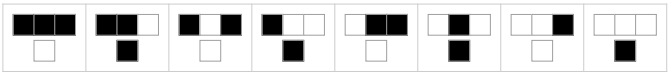
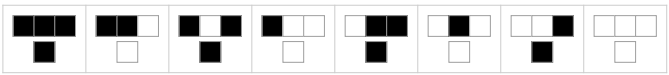
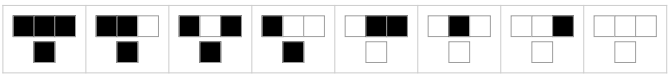
Mediante a Figura 17, o cálculo evolutivo tradicional seria aplicar os valores das nove células da vizinhança $s^{t-1}[i-4], s^{t-1}[i-3], \ldots, s^{t-1}[i+4]$ à regra R para calcular o valor desconhecido da célula $s^t[i]$. Entretanto, em um cenário de cálculo de pré-imagem, a mesma figura pode ser utilizada para pontuar elementos relevantes. A Seção 2.3.2 descreveu um mecanismo proposto por Gutowitz para realizar tal cálculo, de célula em célula, dadas algumas pré-condições.

Considerando que a regra R tem sensitividade à esquerda, pelo mecanismo de Gutowitz seria possível determinar, por meio de comparação com as transições da regra R, o valor desconhecido do bit da sensitividade, $s^{t-1}[i-4]$. Isso, todavia, dependeria também de um conhecimento prévio dos valores de $s^t[i]$ e das outras 8 células da vizinhança da pré-imagem $s^{t-1}[i-3], s^{t-1}[i-2], \ldots, s^{t-1}[i+4]$. É importante reiterar que o mesmo artifício também pode ser usado, com mínimas alterações, em caso de regras com sensitividade à direita (onde $s^{t-1}[i+4]$ seria o bit calculado).

Retomando a análise do cálculo de pré-imagem, todos os bits de s^t são conhecidos previamente, porém nenhum dos bits de s^{t-1} é conhecido no início do procedimento. Como a mesma regra é aplicada de forma homogênea no reticulado, essa dependência de conhecer os valores de $2 \times r$ bits da vizinhança é compartilhada por todos os bits da pré-imagem; logo, tem-se uma situação de impasse que impediria o cálculo dos primeiros bits da pré-imagem. Esse impasse motivou Gutowitz a acrescentar $2 \times r$ bits aleatórios em cada novo reticulado de pré-imagem para, em seguida, conseguir calcular os restantes.

A proposta de Macêdo (2007) para evitar esse aumento do tamanho do reticulado foi buscar um tipo especial de regras onde há uma relação de sensitividade ainda mais forte entre o bit de sensitividade da pré-imagem e o bit resultante. Em tais regras, conhecer o valor do bit resultante em s^t é suficiente para ser capaz de afirmar o valor do bit da sensitividade na vizinhança correspondente na pré-imagem, sem para isso depender dos valores do restante da vizinhança. Essa busca resultou por encontrar as quatro regras elementares listadas na Tabela 1.

Tabela 1 – Regras elementares com sensitividade total.

Regra	**Comportamento**	**Sensitividade**
15 - 00001111		Esquerda
85 - 01010101		Direita
170 - 10101010		Direita
240 - 11110000		Esquerda

As regras da Tabela 1 demonstram um comportamento único no qual o bit resultante em s^t é sempre uma cópia ou complemento do bit da sensitividade em s^{t-1}. As transições da regra 170 sempre copiam o bit à direita, enquanto as da regra 240 sempre copiam o bit à esquerda da vizinhança. Já as transições da regra 85 complementam o bit à direita, e as da regra 15 complementam o bit à esquerda da vizinhança. Dessa forma, uma aplicação homogênea dessas regras em um reticulado causaria um deslocamento de posição dos valores das células, demonstrado na Figura 18.

A aplicação homogênea de qualquer uma das quatro regras, conforme observado na Figura 18, garante uma reversibilidade do reticulado sem a necessidade de aumento no tamanho do mesmo, visto que não haveria ambiguidade no cálculo de qualquer célula da pré-imagem. Entretanto, fica visível que esse comportamento de deslocamento com cópia ou complemento é muito simplista para compor um mecanismo criptográfico em si, visto que são mantidas as características básicas do texto em claro original.

Ao enumerar essas regras também pode-se observar que as sequências binárias em suas representações (“00001111”, “11110000”, “01010101” e “10101010”) seguem padrões extremamente simples, que podem ser replicados para raio 2 ou além, conforme Tabela 2, sem qualquer mudança de comportamento.

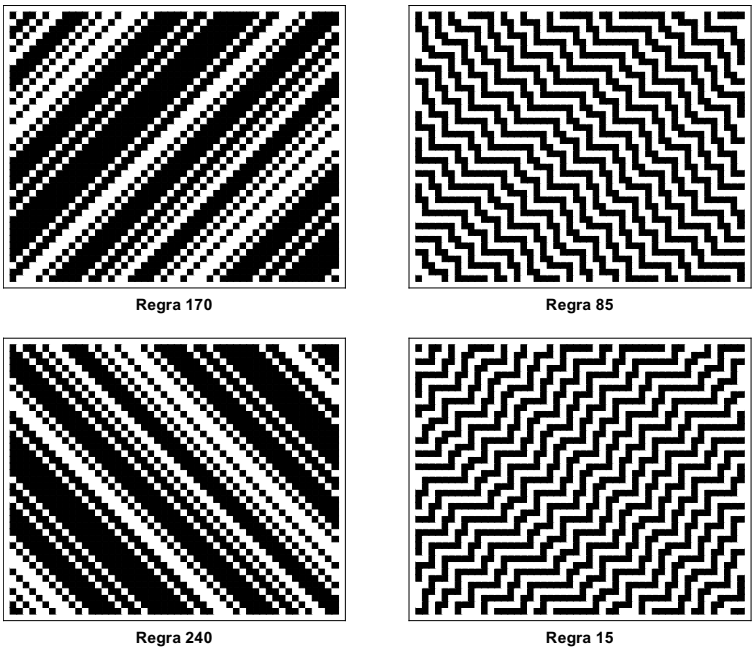


Figura 18 – Comportamento das regras com sensitividade total. [Fonte: autor]

Tabela 2 – Equivalência de regras com sensitividade total.

Raio 1	**Raio 2**	**Raio r - RegEx**
15 - 00001111	00000000000000001111111111111111	$[0000]\{4^r\}[1111]\{4^r\}$
85 - 01010101	01010101010101010101010101010101	$[0101]\{2 \times 4^r\}$
170 - 10101010	10101010101010101010101010101010	$[1010]\{2 \times 4^r\}$
240 - 11110000	11111111111111110000000000000000	$[1111]\{4^r\}[0000]\{4^r\}$

Então, na proposição do método HCA, foi descrito o uso de um AC híbrido no qual, para cada etapa de evolução ou cálculo de pré-imagem, duas regras R_p e R_b são utilizadas. Na nomenclatura adotada pelo método, R_p denota a "regra principal" que introduz comportamento caótico no reticulado visando cifração, enquanto R_b denota a "regra de borda" que garante reversibilidade do reticulado (LIRA et al., 2023). Como o HCA é um método onde são utilizadas regras de raio 4, para cada reticulado com 128 bits, R_b atua sobre a nomeada "borda" do reticulado que terá 8 bits ($2 \times r$), enquanto os 120 bits restantes são evoluídos por R_p, a regra principal. Os critérios da escolha de R_p serão detalhados na Seção 2.3.3.4. Já R_b é escolhida entre as quatro regras com sensitividade total. Essa escolha é determinística a cada passo, onde duas restrições são impostas:

- R_b precisa ter sensitividade à mesma direção que R_p.
- O primeiro bit de R_b precisa ser complementar ao primeiro bit de R_p.

A primeira restrição é necessária para o funcionamento do método, e a segunda restrição é puramente arbitrária, visando tornar a dinâmica do reticulado menos previsível, por meio da alternância das regras de borda. Embora o HCA utilize $r = 4$, a Figura 19 exemplifica o cálculo de pré-imagem proposto no HCA em regras $r = 1$ e com reticulado de $N = 10$ por fins didáticos.

Na Figura 19 estão indicadas duas regras com sensitividade à esquerda, uma delas sendo a regra principal (R_{120}) e a outra sendo a regra de borda (R_{240}). Esta figura indica um passo-a-passo do cálculo da pré-imagem $t = -1$, cujos valores de células são inicialmente desconhecidos, a partir da imagem $t = 0$, que já é conhecida. As setas direcionais, bem como as letras alfabéticas na lateral esquerda, indicam a progressão do cálculo iterativo de valores da pré-imagem. Visto que $r = 1$, tem-se que duas células do reticulado formam a região de borda (indicada na pré-imagem por uma coloração cinza) e o restante do reticulado é considerado a região principal.

O cálculo dos bits da pré-imagem é sempre realizado na direção da sensitividade, que no caso representado implica em um cálculo da direita para a esquerda. Além disso, o cálculo é iniciado pelas células que podem ser inferidas usando a definição de regra de borda. Sendo assim, em **(a)**, a vizinhança selecionada em $t = -1$ está centralizada na célula da borda que está mais à direita e um sublinhado foi aplicado ao bit resultante dessa vizinhança, de valor "0". Visto que a regra 240 estabelece uma relação de igualdade entre o bit determinante da vizinhança em $t = -1$ e o bit resultante em $t = 0$, tem-se que o bit à esquerda na vizinhança receberá o valor "0" (atribuição visível em **(b)**).

Em seguida, o instante **(b)** da figura destaca a vizinhança seguinte em $t = -1$, centrada na célula restante da região de borda, bem como exibe um destaque sublinhado no valor "1" do bit resultante em $t = 0$. Assim como no cálculo anterior, dado que a célula central da vizinhança faz parte da região de borda, R_{240} define uma relação de igualdade de valor entre o bit resultante em $t = 0$ e da célula determinante da vizinhança em $t = -1$. Logo, a célula à esquerda da vizinhança selecionada na pré-imagem recebe também o valor "1", cuja atribuição fica visível em **(c)**.

As etapas **(a)** e **(b)** demonstraram o cálculo de dois bits da pré-imagem, de forma determinística, utilizando a regra de borda. Agora, dispondo destes dois valores iniciais, será possível calcular sequencialmente todos os outros valores restantes da pré-imagem utilizando a regra principal, R_{120}. Logo, em **(c)**, observa-se a seguinte relação entre a vizinhança em $t = -1$ e o bit resultante em $t = 0$: $(?, 1, 0) \rightarrow 1$. Devido ao caráter determinante do bit desconhecido da vizinhança, dada a definição de regras com sensitividade à esquerda, tem-se que só existe uma transição de R_{120} que satisfaz os valores já conhecidos. Assim, dada a transição $f_{120}(1, 1, 0) = 1$, a célula à esquerda da vizinhança selecionada na pré-imagem recebe o valor "1", que consta preenchido em **(d)**.

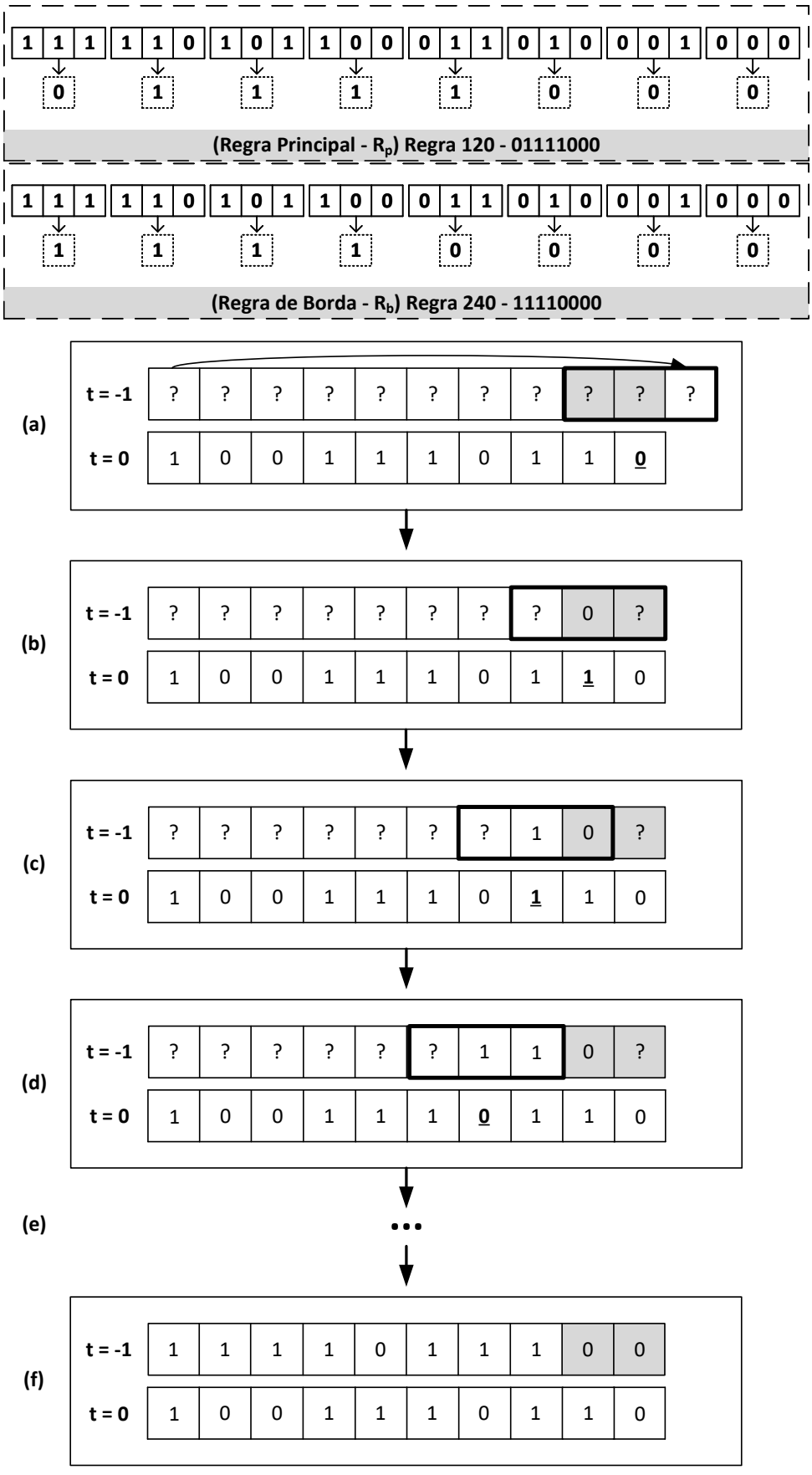


Figura 19 – Cálculo de uma pré-imagem conforme reversibilidade do HCA. [Fonte: autor]

Esta lógica aplicada em **(c)** é repetida para todos os bits restantes, tanto em **(d)** quanto em todas as etapas simbolicamente resumidas em **(e)**. A cada etapa, a vizinhança é deslocada na direção da sensitividade, conforme previamente mencionado. Finalmente, a pré-imagem completa é apresentada em **(f)** e tal processo pode ser repetido para obter, de forma determinística, tanto $t = -2$ quanto sucessivamente qualquer outra pre-imagem que se deseje. O arranjo híbrido proposto para o HCA garante a manutenção do tamanho do reticulado e teve sua reversibilidade provada em (ALT, 2013).

2.3.3.2 HCA - Cifração

Cada etapa do HCA envolve o uso de duas regras evolutivas, R_p (regra principal) e R_b (regra de borda). Assim sendo, ao contrário do método de Gutowitz onde o AC é homogêneo, o HCA faz uso de um AC híbrido para seus mecanismos criptográficos.

Na sua definição como um algoritmo de cifra em bloco, cada bloco do HCA contém 128 bits e corresponde a um reticulado que será evoluído separadamente. Esse tamanho foi escolhido arbitrariamente para estar em conformidade com padrões da literatura, mas o método suportaria outras especificações maiores com alterações mínimas.

Tem-se que a cifração do HCA corresponde a aplicar 128 iterações da evolução reversa (cálculo de pré-imagem) a cada bloco que é inicializado contendo parte do texto em claro, e a configuração resultante de tais operações sucessivas é a parte correspondente do texto cifrado. Também é relevante frisar que R_p e R_b são regras de raio 4 ($r = 4$) e, consequentemente, a vizinhança para aplicação dessas regras é composta por 9 bits.

Mediante apresentado na Seção 2.3.3.1, R_b é uma regra especial que permite calcular sem ambiguidade o valor de bits desconhecidos da pré-imagem, na quantidade de células que a região de borda tiver, dado o conhecimento de valores dos bits correspondentes da imagem resultante. O trecho do reticulado que é calculado utilizando R_b é nomeado "borda", e é composto por 8 bits. Assim sendo, R_p é aplicada no cálculo dos 120 bits restantes do reticulado. Tem-se que ambas as regras apresentam sensitividade a uma mesma direção, algo fundamental para aplicar esse cálculo da pré-imagem. Um resumo geral do procedimento de cifração do HCA é exibido na Figura 20.

Conforme a Figura 20, cada bloco contendo 128 bits do texto em claro é considerado a configuração inicial de um AC, o reticulado $t = 0$, em que são aplicadas sucessivas operações de cálculo de pré-imagem até alcançar a configuração $t = -128$. Em cada uma das etapas, é aplicado um par de regras R_p e R_b potencialmente diferentes, mecanismo esse que será melhor detalhado na Seção 2.3.3.4. Ao fim do procedimento em todos os blocos, os resultados dos reticulados são combinados para formar o texto cifrado final.

A etapa "Modo de Operação" exibida na mesma figura refere-se ao procedimento de cifras em bloco citado na Seção 2.1.1; o método HCA não teve um modo de operação definido em sua especificação, sendo ele compatível com todos os padrões descritos atualmente na literatura.

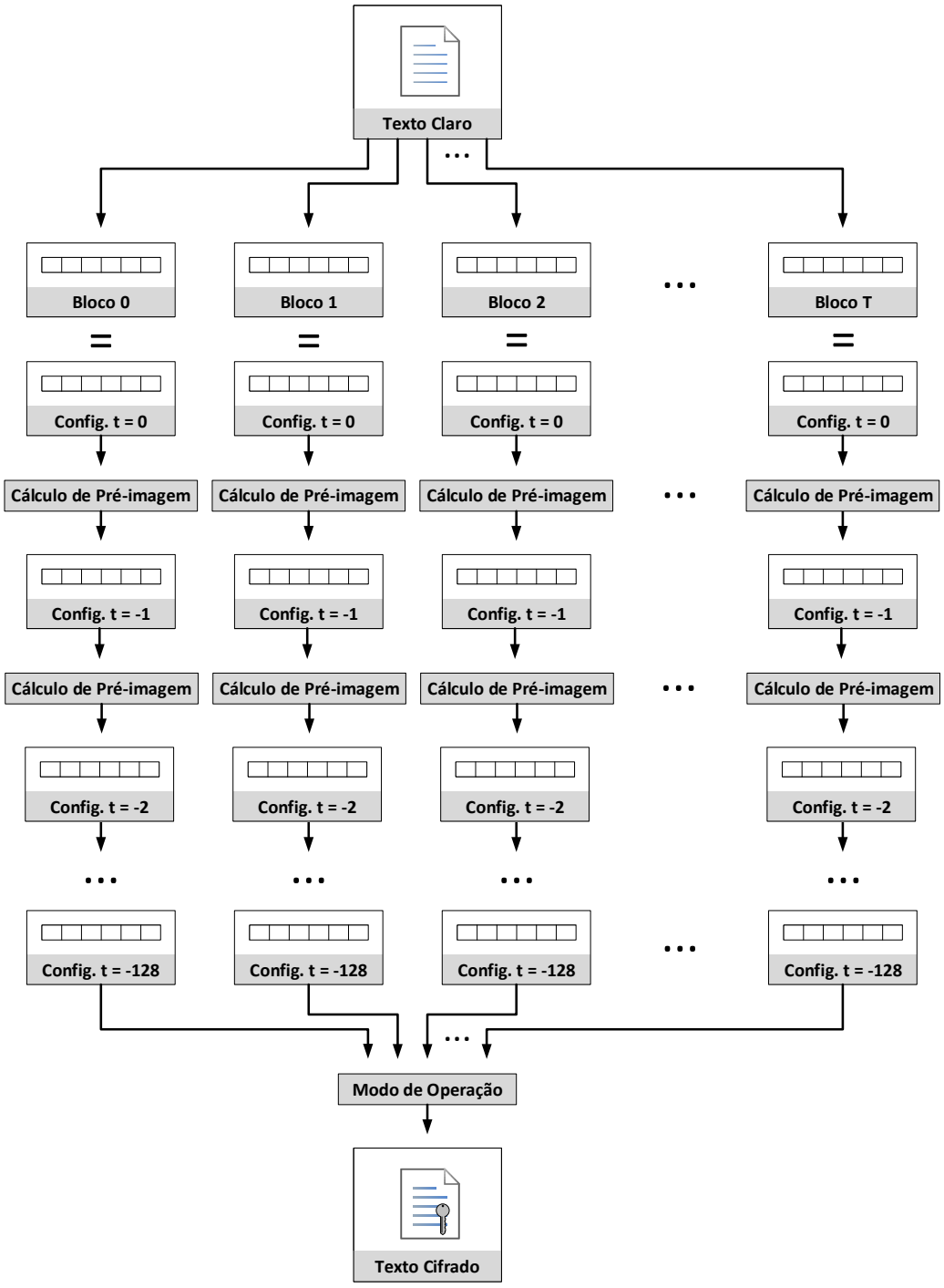


Figura 20 – Cifração no método HCA. [Fonte: autor]

2.3.3.3 HCA - Decifração

O processo de decifração do HCA é o procedimento oposto à cifração descrita na seção anterior. O passo inicial consiste na quebra do texto cifrado em blocos de 128 bits, onde cada bloco é um reticulado a ser decifrado separadamente por meio do processo de evolução intrínseco aos ACs.

Supondo que o procedimento de decifração está sendo aplicado a um dado texto que foi previamente cifrado pelo mesmo algoritmo (embora isso não seja estritamente necessário), didaticamente pode-se considerar que cada reticulado contém um trecho do texto que passou por 128 passos de evolução reversa (fato indicado por $t = -128$). Logo, para recuperar os trechos do texto em claro original ($t = 0$), será necessário aplicar 128 passos da evolução convencional do AC a cada reticulado, conforme Figura 21.

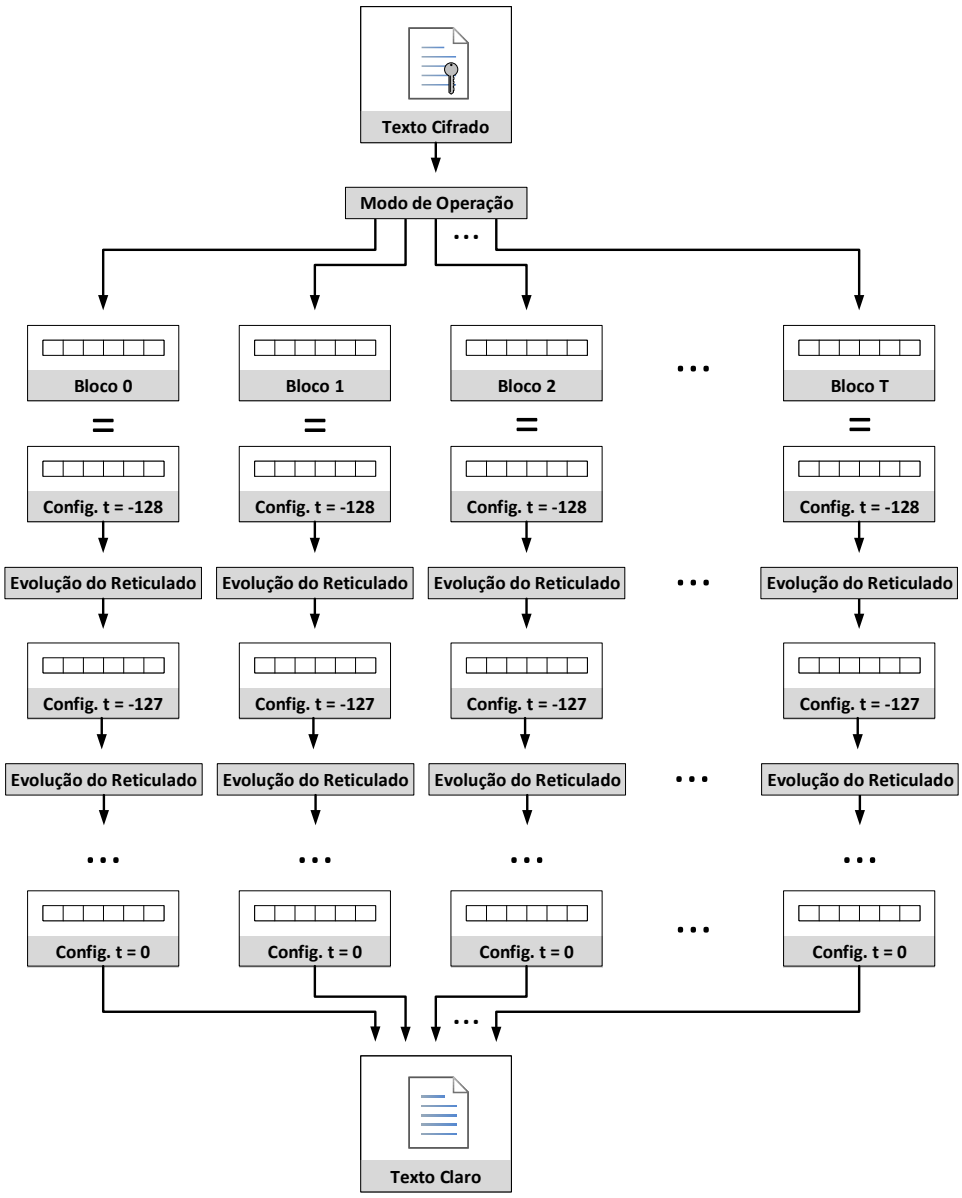


Figura 21 – Decifração no método HCA. [Fonte: autor]

Assim como no cálculo de pré-imagem da etapa de cifração, tem-se aqui um AC híbrido. Cada evolução dos reticulados utilizará uma regra principal (R_p) e uma regra de borda (R_b). E, mais que isso, é fundamental que haja uma paridade entre as regras utilizadas em etapas equivalentes da cifração e decifração (conforme será detalhado na Seção 2.3.3.4), bem como que as mesmas regras sejam aplicadas nas células correspondentes do reticulado, de acordo com tais etapas.

2.3.3.4 HCA - Chave Criptográfica

A chave criptográfica do HCA é uma sequência K de 257 bits, onde os primeiros 256 bits serão utilizados para gerar R_p, a regra principal do AC, e o último bit indica a direção da sensitividade para ambas as regras; se $K[256]$ for "0" a direção da sensitividade é esquerda, se $K[256]$ for "1" ocorre sensitividade à direita.

Conforme explicado na Seção 2.2.4, dado que o HCA é um algoritmo cujas regras são de $r = 4$, são necessários 512 bits para representar uma regra. Entretanto, mediante detalhamento visto na Seção 2.2.6, tem-se que regras com sensitividade a uma das direções podem ser representadas com metade dos bits que uma regra convencional. Assim, visto que R_p é uma regra com sensitividade à direção indicada por $K[256]$, ela pode ser expressa com 256 bits. Desta forma, os 256 bits iniciais da chave, $K[0] \dots K[255]$, serão expandidos aos 512 bits que denotam R_p de modo análogo ao detalhado na Seção 2.2.6. Em caso de sensitividade à esquerda, tem-se $R_p = K[0], K[1], \dots, K[255], \overline{K[0], K[1], \dots, K[255]}$. No caso de sensitividade à direita, ocorre $R_p = K[0], \overline{K[0]}, K[1], \overline{K[1]}, \dots, K[255], \overline{K[255]}$. Após conhecido R_p, obtém-se R_b conforme as restrições listadas na Seção 2.3.3.1.

Por meio de experimentação, averiguou-se que regras cuja representação binária apresenta maior entropia tendem a causar um comportamento mais caótico quando aplicadas na evolução de ACs. Visto isso, na tentativa de garantir uma boa cifração dos dados, ficou definido arbitrariamente que, para K ser uma chave válida do HCA, a sequência $K[0], \dots, K[255]$ precisa ter entropia de dados avaliada como maior ou igual a 0,75 por meio da Equação 2. Esse filtro, ao prevenir chaves de natureza homogênea, descarta aproximadamente 0.1×10^{-8} % do espaço total de chaves, um impacto considerado mínimo (LIRA et al., 2023).

Após a confirmação de que K é uma chave válida para o método, outro mecanismo do HCA é a rotação de chave para garantir a variabilidade das regras utilizadas no AC. A cada etapa da cifração, a sequência iniciada pelos primeiros 256 bits de K é rotacionada à esquerda, causando alterações nas regras R_p e R_b. A Tabela 3 apresenta esse comportamento.

Para o pleno funcionamento do algoritmo, é necessário que o processo de decifração utilize as chaves em sequência reversa à ordem de uso na cifração do texto em claro.

Tabela 3 – Sequência de uso das chaves no processo de cifração do HCA.

Passo	**Chave para gerar R_p e R_b**
1	$K_1 = K[0] \dots K[255]$
2	$K_2 = RotacaoEsquerda(K_1)$
3	$K_3 = RotacaoEsquerda(K_2)$
...	...
128	$K_{128} = RotacaoEsquerda(K_{127})$

2.3.4 Estado da Arte

A confiança de que a proposta é inovadora e relevante foi construída pelo estudo das soluções existentes de áreas correlatas, com foco em métodos criptográficos simétricos baseados em autômatos celulares. Dentre as soluções do referido perfil descritas na literatura, as mais relevantes são apresentadas abaixo, em ordem cronológica.

- **Trabalho Correlato:** (WOLFRAM, 1985)
  Tipo de Cifra: Cifra de fluxo
  Tipo do AC: Homogêneo
  Dimensões: Unidimensional
  Propriedades das Regras: Regra 30 - Caótica
  Raio das Regras: R1
  Chave: Configuração inicial do AC (de tamanho ímpar)
  Descrição: Algoritmo criptográfico baseado no uso da regra elementar 30, onde a chave define o reticulado inicial. Tal reticulado é evoluído pela regra 30 com contorno periódico, e o texto cifrado será resultante da operação XOR entre os bits do texto em claro e cada bit que é gerado pela evolução do reticulado, na posição central do mesmo.

- **Trabalho Correlato:** CA-1.0 (GUTOWITZ, 1993)
  Tipo de Cifra: Cifra de bloco com link
  Tipo do AC: Homogêneo
  Dimensões: Unidimensional
  Propriedades das Regras: Regras Irreversíveis, porém com Sensitividade
  Raio das Regras: R5
  Chave: 1088 bits (1024 + 64)
  Descrição: Método criptográfico baseado no uso de regras de raio 5. O texto em claro é separado em blocos de 384 bits, e links de 320 bits são gerados de forma pseudoaleatória. 1024 bits da chave original são usados para gerar duas regras, uma com sensitividade à esquerda e outra à direita. Tais regras são utilizadas na evolução reversa do reticulado, onde o link de 320 bits provê os bits iniciais necessários para executar cada etapa evolutiva conforme o mecanismo descrito na Seção 2.3.2. Tal mecanismo garante a possibilidade de recuperar o texto em claro a partir do texto cifrado que foi gerado.

- **Trabalho Correlato:** (NANDI; KAR; CHAUDHURI, 1994)
  Tipo de Cifra: Cifra de fluxo
  Tipo do AC: Heterogêneo
  Dimensões: Unidimensional
  Propriedades das Regras: Regras Aditivas
  Raio das Regras: R1
  Chave: Adaptável ao tamanho do texto em claro
  Descrição: Método criptográfico baseado em autômatos celulares híbridos e aditivos utilizando regras elementares. Uma sequência de regras aditivas é aplicada para obter a chave de fluxo; os elementos de tal sequência são as regras 51, 153 e 195. Para tais regras, existem outras complementares equivalentes, que são as regras 204, 102 e 60. De forma que, para uma cifração feita com o primeiro grupo de regras, existe uma transformação inversa por meio do segundo grupo. Fato este que garante a possibilidade de eventualmente recuperar o texto em claro a partir do texto cifrado.

- **Trabalho Correlato:** (TOMASSINI; PERRENOUD, 2000)
  Tipo de Cifra: Cifra de fluxo
  Tipo do AC: Heterogêneo
  Dimensões: Unidimensional ou Bidimensional
  Propriedades das Regras: Regras Caóticas
  Raio das Regras: R1 (em caso 2-D, vizinhança de von Neumann)
  Chave: Config. inicial do AC (de tamanho ímpar) + Vetor de regras
  Descrição: O algoritmo proposto utiliza um princípio derivado da solução proposta em (WOLFRAM, 1985). Entretanto, neste modelo é usado um AC híbrido cujo vetor de regras é descrito na chave. Uma permutação das regras elementares 90, 105, 150 e 165 é utilizada no processo. O reticulado inicial do AC é definido também pela chave original, e a chave de fluxo é obtida nos bits da coluna central do reticulado à medida que este é evoluído.

- **Trabalho Correlato:** (WUENSCHE, 2009)
  Tipo de Cifra: Cifra de bloco
  Tipo do AC: Homogêneo
  Dimensões: Unidimensional
  Propriedades das Regras: $Z_L = 1 \,\&\, 0.5 \leq Z_R < 1$ ou $Z_R = 1 \,\&\, 0.5 \leq Z_L < 1$
  Raio das Regras: Adaptável de acordo com o espaço de chaves desejado
  Chave: Regra de AC
  Descrição: Esta proposta se baseia no mecanismo para cálculo de pré-imagem para cifrar o texto em claro que é a configuração inicial do AC. O trabalho descreve um subconjunto de regras que atende aos critérios desejados para realizar o procedimento de cifração e permitir eventual decifração. A adequação de cada regra é analisada mediante o parâmetro Z, descrito pelo autor em trabalho prévio. Esse parâmetro Z é indicativo da quantidade de pré-imagens no ciclo de estados gerados por cada regra. Aplicações iterativas do cálculo de pré-imagem ao texto em claro produzem o texto cifrado. E a evolução tradicional do texto cifrado recupera o texto em claro.

- **Trabalho Correlato:** THCA (JÚNIOR, 2010)
  Tipo de Cifra: Cifra de bloco
  Tipo do AC: Heterogêneo
  Dimensões: Bidimensional
  Propriedades das Regras: Regras com Sensitividade
  Raio das Regras: R1 (vizinhança de von Neumann)
  Chave: Núcleo da regra principal
  Descrição: O THCA é um algoritmo de cifração simétrica implementado em autômatos celulares bidimensionais (2D) visando uso em imagens monocromáticas. Seu mecanismo de cifração e decifração é baseado em equivalentes unidimensionais propostos no método HCA (MACÊDO, 2007), de forma que a cada etapa evolutiva são utilizadas uma regra principal e uma regra de borda. No HCA, baseado em AC unidimensional, são consideradas regras com sensitividade à esquerda ou à direita; enquanto isso, no THCA, implementado em AC bidimensional, são consideradas 4 direções de sensitividade nomeadas pelas cardinalidades: norte, sul, leste, oeste. A variação da direção da sensitividade, a cada etapa evolutiva, favorece a qualidade da cifração.

- **Trabalho Correlato:** 3DHCA (LIMA, 2012)
  Tipo de Cifra: Cifra de bloco
  Tipo do AC: Heterogêneo
  Dimensões: Tridimensional
  Propriedades das Regras: Regras com Sensitividade
  Raio das Regras: R1 (vizinhança de von Neumann)
  Chave: Núcleo da regra principal
  Descrição: O 3DHCA é uma adaptação tridimensional (3D) do método HCA (MACÊDO, 2007). Essa abordagem é otimizada para cifrar imagens coloridas, aplicação que ainda era limitada tanto no HCA quanto no THCA (JÚNIOR, 2010). O modelo 3DHCA também implementa a cifração como uma sucessão de cálculos de pré-imagem e, para uma dada imagem colorida, decompõe cada pixel da mesma em suas componentes de cor RGB. Os valores de tais componentes de cor são, então, intercalados em um bloco único que é cifrado.

❑ **Trabalho Correlato:** HCA (LIRA et al., 2023)
Tipo de Cifra: Cifra de bloco
Tipo do AC: Heterogêneo
Dimensões: Unidimensional
Propriedades das Regras: Regras com Sensitividade
Raio das Regras: R4
Chave: Núcleo da regra principal
Descrição: O funcionamento do HCA, conforme detalhado na Seção 2.3.3, ocorre pela quebra do texto em claro em blocos cuja cifração é feita mediante sucessivos cálculos de pré-imagem. A operação de cálculo de pré-imagem utilizada não ocasiona incremento no tamanho do reticulado, e duas regras (R_p e R_b) são aplicadas simultaneamente no reticulado a cada etapa. A regra principal, R_p, é extraída diretamente da chave criptográfica e provê a força criptográfica do método. Após 128 cálculos de pré-imagem, tem-se o trecho cifrado equivalente ao reticulado inicial. A evolução tradicional do texto cifrado recupera o texto em claro.

A Tabela 4 resume atributos relevantes dos métodos de criptografia simétrica citados acima para facilitar a comparação com o algoritmo proposto na tese, o VHCA. A ordenação dos métodos na tabela segue a sequência crescente com relação à data de publicação.

Tabela 4 – Comparativo dos métodos, com características do VHCA destacadas em verde.

	Tipo / Modelo / AC
(WOLFRAM, 1985)	Fluxo / OTP / Hom - Caótica - 1D - R1
(GUTOWITZ, 1993)	Bloco+Link / Pré / Hom - Sensit - 1D - R5
(NANDI; KAR; CHAUDHURI, 1994)	Fluxo / Reversib / Het - Aditivas - 1D - R1
(TOMASSINI; PERRENOUD, 2000)	Fluxo / OTP / Het - Caót - 1D - R1
(WUENSCHE, 2009)	Bloco / Pré / Hom - Caót (Z) - 1D - *
(JÚNIOR, 2010)	Bloco / Pré / Het - Sensit - 2D - R1
(LIMA, 2012)	Bloco / Pré / Het - Sensit - 3D - R1
(LIRA et al., 2023)	Bloco / Pré / Het - Sensit - 1D - R4
VHCA	Bloco / Pré / Het - Sensit - 1D - R1

Na Tabela 4, as características presentes no VHCA estão destacadas na cor verde como forma de favorecer a comparação visual entre os métodos. A seção “Tipo” indica o tipo de algoritmo de criptografia simétrica, que é classificado em “cifra de bloco” ou “cifra de fluxo”. A seção “Modelo” exibe a qual categoria o método pertence, de acordo com a seguinte classificação: Métodos que utilizam da mecânica de ACs para a geração de um “OTP” são indicados com tal termo. Métodos onde a reversibilidade do AC é alcançada pelo uso exclusivo de regras lineares são indicados pelo termo “Reversib”. E métodos onde uma mecânica de cálculo de pré-imagem do AC inclui regras não-lineares são indicados pelo termo “Pré”. Com relação à seção “AC”, ela indica propriedades relevantes do autômato celular utilizado; no caso, são listados: Se o autômato é homogêneo ou híbrido; características das regras utilizadas (entre “caóticas”, “aditivas”, ou com “sensitividade”); em quantas dimensões o AC é expresso (se unidimensional “1-D”, bidimensional “2-D”, etc); e, finalmente, o raio “R” das regras de AC aplicadas. O próximo capítulo contém a descrição do método VHCA que traz um detalhamento maior do método e de como ele se posiciona frente aos aspectos mencionados acima.

Capítulo 3

Very Heterogeneous Cellular Automata (VHCA)

Este capítulo traz a descrição do algoritmo VHCA (*Very Heterogeneous Cellular Automata*), objeto desta tese, que foi desenvolvido a partir da sugestão "Método Criptográfico com Raio 1 baseado no modelo HSR" listada em (ALT, 2013) como um trabalho futuro. O conceito inicial proposto em (ALT, 2013) é adaptado do modelo HCA (apresentado na Seção 2.3.3) e define uma alteração significativa no formato e papel da chave criptográfica. Ao invés de atuar com regras de raio 4, avalia-se alternativamente o uso de regras de raio 1, cujo melhor desempenho e facilidade de implementação são desejáveis.

A implementação do VHCA foi iniciada após realizar um estudo prévio para definição do tamanho da chave, especificação de um alfabeto inicial de regras, bem como uma avaliação da relação entre os bits da chave criptográfica e a direção de sensitividade das regras, visando os padrões de funcionamento do algoritmo que serão detalhados neste capítulo. Os experimentos apresentados no capítulo seguinte demonstram resultados obtidos pela execução do referido algoritmo.

3.1 VHCA - Introdução

O VHCA, assim como o HCA, é um algoritmo de criptografia simétrica por cifra em bloco cujas operações de cifração e decifração correspondem à evolução tradicional e reversa (cálculo de pré-imagem) de um autômato celular híbrido. Assim como no HCA, a reversibilidade nas operações do VHCA é garantida ao restringir o conjunto de regras do AC àquelas que apresentam sensitividade a uma única direção. Também é mantido o uso da região de "borda" no reticulado do VHCA para garantir que este conserve seu tamanho original a cada cálculo de pré-imagem.

Quando comparado a algoritmos criptográficos tradicionais apresentados no Anexo A, o VHCA inova por ser uma alternativa baseada em ACs que traz um paralelismo inerente ao seu funcionamento. Esse paralelismo pode ser explorado em hardware dedicado e é extremamente desejável visando à escalabilidade da solução. Enquanto isso, quando comparado a outros métodos de criptografia de bloco baseados em AC descritos na literatura e apresentados na Seção 2.3, um dos destaques do VHCA é pela redução do raio das regras de AC utilizadas no processo evolutivo, resultando em um algoritmo de implementação relativamente simples que exige menos recursos de processamento. As próximas seções contêm a descrição do método, bem como uma listagem das medidas adotadas para atingir robustez criptográfica no algoritmo proposto.

3.2 VHCA - Chave Criptográfica

No VHCA, cada bloco obtido pela segmentação do texto em claro é composto por $N = 128$ bits, um tamanho que foi mantido em igualdade com o HCA e o *Advanced Encryption Standard* (AES). Esse bloco, que constitui o reticulado inicial do AC, é evoluído por um arranjo de regras a cada passo de tempo. Enquanto no algoritmo predecessor (HCA) duas regras, R_p e R_b, geradas a partir da chave, são aplicadas a cada passo evolutivo, no VHCA uma multitude de regras são aplicadas às diferentes células do reticulado. Sendo r o raio da vizinhança das regras do AC, uma sequência de $2 \times r$ células ainda é considerada a "borda" do reticulado, sendo as células de tal região evoluídas por regras de sensitividade total. Reiterando que, ao contrário do HCA, no VHCA as regiões ("principal" e "borda") de células deixam de ser evoluídas por uma única regra entre si e suas posições do reticulado passam a ser evoluídas por regras selecionadas individualmente de vetores de regras criados a partir da chave K.

Visto que a chave K é utilizada para gerar a sequência de regras aplicadas ao reticulado, é necessário garantir certo nível de heterogeneidade na sua distribuição de bits. Essa diversidade é medida por meio da "entropia espacial normalizada" (h), detalhada na Seção 2.1.2.1. Para que uma chave binária K, de comprimento $3 \times N$, possa ser considerada válida para uso no método VHCA, ela deve apresentar $h(K) > 0{,}75$. Tal restrição para garantir a diversidade dos bits da chave foi também utilizada no método predecessor HCA (LIRA et al., 2023).

Da chave criptográfica K é possível extrair vetores de regras de AC que indicam qual será utilizada para evoluir cada posição de célula do reticulado. Visto que, comparado ao HCA ($r = 4$), este algoritmo define um AC em que um número maior de regras é aplicado no reticulado a cada etapa, a diversidade de comportamentos alcançada pela permutação das regras inspirou a possibilidade de aplicar somente regras elementares ($r = 1$) no processo evolutivo do VHCA. Embora o conceito principal do método seja compatível com regras de raio maior, o tamanho do raio e eventual vizinhança das regras utilizadas em um AC têm impacto direto no poder computacional necessário para sua evolução; logo, têm-se uma vantagem na escolha de utilizar as regras mais simples possíveis. A Tabela 5 traz a listagem das regras de AC elementares ($r = 1$) que, entre todas as 256 regras possíveis, apresentam sensitividade a alguma direção da vizinhança.

Tabela 5 – Regras elementares que exibem sensitividade

Regra	**Sensitividade**	**Função**	**Tipo da Função**
15 - 00001111	Esquerda	$\neg E$	Complementar (240)
30 - 00011110	Esquerda	$E \oplus (C \vee D)$	Não-Linear
45 - 00101101	Esquerda	$E \oplus (C \vee (\neg D))$	Não-Linear
60 - 00111100	Esquerda	$E \oplus C$	Linear
75 - 01001011	Esquerda	$E \oplus ((\neg C) \vee D)$	Não-Linear
85 - 01010101	Direita	$\neg D$	Complementar (170)
86 - 01010110	Direita	$(E \vee C) \oplus D$	Não-Linear
89 - 01011001	Direita	$(E \vee (\neg C)) \oplus D$	Não-Linear
90 - 01011010	Bilateral	$E \oplus D$	Linear
101 - 01100101	Direita	$E \oplus (E \wedge C) \oplus (\neg D)$	Não-Linear
102 - 01100110	Direita	$C \oplus D$	Linear
105 - 01101001	Bilateral	$\neg(E \oplus C \oplus D)$	Complementar (150)
106 - 01101010	Direita	$(E \wedge C) \oplus D$	Não-Linear
120 - 01111000	Esquerda	$E \oplus (C \wedge D)$	Não-Linear
135 - 10000111	Esquerda	$(\neg E) \oplus (C \wedge D)$	Não-Linear
149 - 10010101	Direita	$(E \wedge C) \oplus (\neg D)$	Não-Linear
150 - 10010110	Bilateral	$E \oplus C \oplus D$	Linear
153 - 10011001	Direita	$\neg(C \oplus D)$	Complementar (102)
154 - 10011010	Direita	$E \oplus (E \wedge C) \oplus D$	Não-Linear
165 - 10100101	Bilateral	$\neg(E \oplus D)$	Complementar (90)
166 - 10100110	Direita	$(E \wedge C) \oplus C \oplus D$	Não-Linear
169 - 10101001	Direita	$(\neg(E \vee C)) \oplus D$	Não-Linear
170 - 10101010	Direita	D	Linear
180 - 10110100	Esquerda	$E \oplus C \oplus (C \wedge D)$	Não-Linear
195 - 11000011	Esquerda	$\neg(E \oplus C)$	Complementar (60)
210 - 11010010	Esquerda	$E \oplus (C \wedge D) \oplus D$	Não-Linear
225 - 11100001	Esquerda	$E \oplus (\neg(C \vee D))$	Não-Linear
240 - 11110000	Esquerda	E	Linear

Para cada uma das regras listadas na Tabela 5 consta também a função lógica com o valor resultante de sua aplicação ($s^{t+1}[i]$), em relação a uma vizinhança composta pelos bits: esquerdo (E: $s^t[i-1]$), central (C: $s^t[i]$), e direito (D: $s^t[i+1]$). É relevante notar que nessa listagem constam também as regras de sensitividade total usadas na borda (R_{15}, R_{85}, R_{170}, R_{240}), regras estas cujo comportamento simples precisa ter baixa influência no comportamento geral do AC para não comprometer a robustez criptográfica do algoritmo.

Nesse conjunto também há regras lineares (entre elas as bilaterais e seus complementos) que apresentam comportamento relativamente simples e seriam potencialmente vulneráveis a ataques de criptoanálise linear. Com a retirada dessas regras lineares, obteve-se o conjunto de regras apresentado na Tabela 6, o qual é usado pelo VHCA para a construção da chave criptográfica.

Tabela 6 – Regras elementares com sensitividade e não-lineares

Regra	**Representação**	**Sensitividade**	**Função**
30	00011110	Esquerda	$E \oplus (C \vee D)$
45	00101101	Esquerda	$E \oplus (C \vee (\neg D))$
75	01001011	Esquerda	$E \oplus ((\neg C) \vee D)$
86	01010110	Direita	$(E \vee C) \oplus D$
89	01011001	Direita	$(E \vee (\neg C)) \oplus D$
101	01100101	Direita	$E \oplus (E \wedge C) \oplus (\neg D)$
106	01101010	Direita	$(E \wedge C) \oplus D$
120	01111000	Esquerda	$E \oplus (C \wedge D)$
135	10000111	Esquerda	$(\neg E) \oplus (C \wedge D)$
149	10010101	Direita	$(E \wedge C) \oplus (\neg D)$
154	10011010	Direita	$E \oplus (E \wedge C) \oplus D$
166	10100110	Direita	$(E \wedge C) \oplus C \oplus D$
169	10101001	Direita	$(\neg(E \vee C)) \oplus D$
180	10110100	Esquerda	$E \oplus C \oplus (C \wedge D)$
210	11010010	Esquerda	$E \oplus (C \wedge D) \oplus D$
225	11100001	Esquerda	$E \oplus (\neg(C \vee D))$

Enquanto na Tabela 5 há 16 regras com sensitividade a cada direção, totalizando 28 regras (visto que as bilaterais são contabilizadas duas vezes), na Tabela 6 constam 8 regras com sensitividade para cada direção, totalizando 16 regras. Visto que $2^3 = 8$, para uma certa direção, cada regra individual pode ser representada com 3 bits. Essa representação dá origem ao alfabeto da chave no algoritmo VHCA, listado na Tabela 7.

Tabela 7 – Alfabeto da chave no VHCA.

	Sensitividade à Esquerda		**Sensitividade à Direita**	
Letra	**Regra**	**Tipo**	**Regra**	**Tipo**
000	R_{30} - 00011110	Principal	R_{86} - 01010110	Principal
001	R_{45} - 00101101	Principal	R_{89} - 01011001	Principal
010	R_{75} - 01001011	Principal	R_{101} - 01100101	Principal
011	R_{120} - 01111000	Principal	R_{106} - 01101010	Principal
100	R_{135} - 10000111	Principal	R_{149} - 10010101	Principal
101	R_{180} - 10110100	Principal	R_{154} - 10011010	Principal
110	R_{210} - 11010010	Principal	R_{166} - 10100110	Principal
111	R_{225} - 11100001	Principal	R_{169} - 10101001	Principal
0**	R_{15} - 00001111	Borda	R_{85} - 01010101	Borda
1**	R_{240} - 11110000	Borda	R_{170} - 10101010	Borda

Conforme a Tabela 7, cada trecho de 3 bits da chave corresponde a uma "letra" do alfabeto da chave do VHCA que, por sua vez, pode ser traduzida a uma regra do AC. O conjunto selecionado para o alfabeto, com exceção das regras de borda, contém apenas regras cujas funções são baseadas nos valores de todos os bits da vizinhança (E, C e D), algo desejável criptograficamente na tentativa de provocar maior interação entre as células do reticulado durante a evolução do AC. Visto que os blocos no VHCA são descritos com 128 células, optou-se por definir o tamanho da chave K do algoritmo como 384 bits (3×128), fixando assim também um espaço de chaves de 2^{384}. Esse comprimento de chave permite que uma regra do alfabeto seja atribuída separadamente a cada posição de célula no reticulado, conforme a Figura 22.

Como indicado na Figura 22, dois vetores de regras, K_M e K_B, são gerados a partir da chave criptográfica K. Cada sequência de 3 bits da chave define qual letra do alfabeto de regras será selecionada da Tabela 7 para compor os novos vetores. O vetor K_M será composto pelas regras correspondentes de tipo "Principal", enquanto o vetor K_B será composto pelas regras correspondentes de tipo "Borda". Observa-se, por exemplo, na Figura 22 que os três primeiros bits de K são "011". Considerando, arbitrariamente, que trata-se de uma execução do algoritmo com sensitividade à esquerda, isso resultará em $K_M[0] = R_{120}$ e $K_B[0] = R_{15}$. Da próxima sequência de três bits "101" será possível definir $K_M[1] = R_{180}$ e $K_B[1] = R_{240}$, e assim por diante. No momento de evoluir o reticulado, cada célula de índice i, onde $0 \leq i < 128$, terá sua vizinhança evoluída por $K_M[i]$, caso pertença à região principal, ou por $K_B[i]$, se pertencer à região de borda.

Para fins de padronização, ao iniciar a cifração ($t = 0$) as primeiras 126 células do reticulado ($0 \leq i < 126$) são consideradas principais, e as duas últimas ($126 \leq i < 128$) correspondem à borda. Entretanto, dada a simplicidade do comportamento imposto pelas regras da região de borda, sua posição é rotacionada dentro do reticulado a cada iteração de tempo t para minimizar seu efeito no comportamento geral do AC. Tal procedimento é melhor detalhado na Seção 3.5.

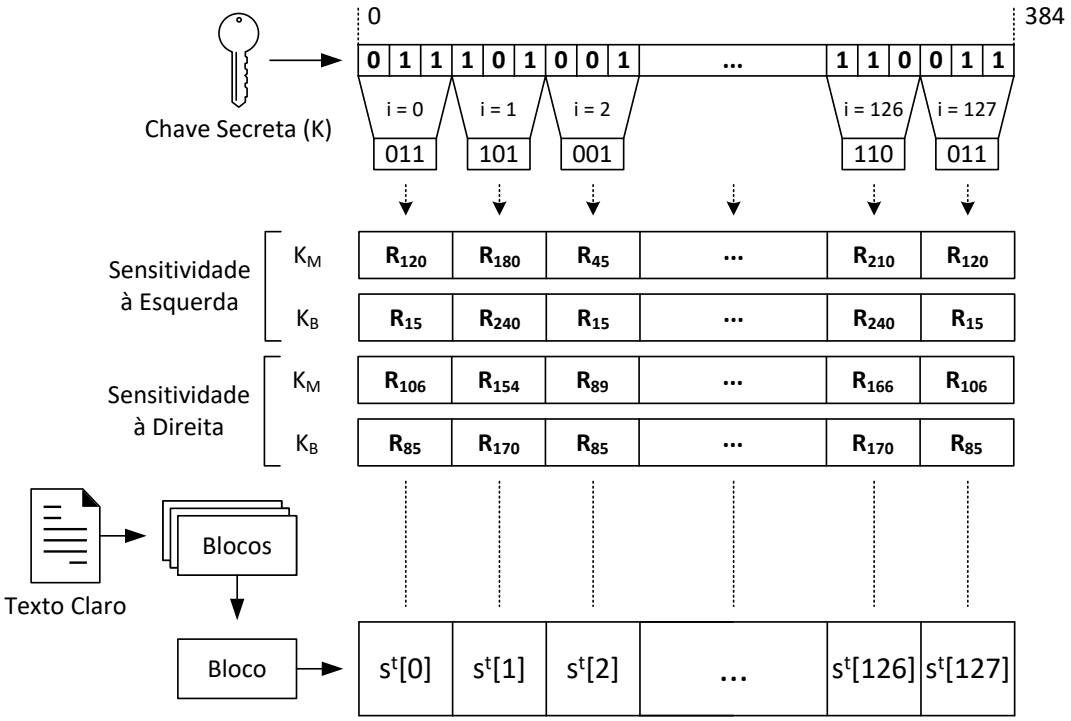


Figura 22 – Aplicação da chave a um bloco do VHCA. [Fonte: autor]

3.3 VHCA - Cifração

Na proposição do método HCA (LIRA et al., 2023), o procedimento de cifração corresponde à evolução reversa do reticulado (cálculo de pré-imagem) e a decifração ocorre pela evolução tradicional do AC. Em contraste, na proposição do VHCA optou-se por utilizar o padrão oposto. Ou seja, a cifração é realizada pela evolução do reticulado por 128 passos de tempo, e a decifração é implementada pelo cálculo sucessivo de 128 pré-imagens a partir do texto cifrado. Não foram encontradas evidências de que um padrão é criptograficamente melhor que o outro; entretanto, é sabido que o processo de evolução para frente é paralelizado de forma mais eficiente, pois a evolução de cada célula do reticulado pode ser feita de forma independente. O contexto necessário para descobrir a regra de evolução das 128 células do reticulado é saber a sensitividade das regras aplicadas e o discernimento de quais células serão evoluídas como principais e quais como borda.

Visto que as regras de borda não introduzem uma dinâmica caótica desejável no comportamento do AC, é desejável que a influência da região de borda seja baixa e dispersa; por isso, tal região de borda é movimentada conforme detalhamento na Seção 3.5, sem que isso altere o padrão de execução do método. Cada passo evolutivo t do AC, nessa etapa de cifração, equivale a aplicar 128 regras a 128 vizinhanças de um reticulado s^t para descobrir os valores das células no novo reticulado s^{t+1}. Dado que os valores de todas as células no instante t são previamente conhecidos, essas 128 aplicações de regras evolutivas podem ser executadas em paralelo usando hardware próprio para tal. O uso do paralelismo para ganho de desempenho é um fator extremamente atraente em soluções envolvendo ACs, como é o caso do VHCA. O pseudocódigo descrito em Algoritmo 1 resume tal processo.

Algoritmo 1: VHCA - Cifração por Evolução Tradicional de AC

Parâmetros: $\$key < Bit > [0...3 \times N]$, $\$plaintext < Bit > [0...N]$, $\$toggleDir < Bit >$
Saída: $\$ciphertext < Bit > [0...N]$

```
1  var $mainRules < Rule > [0...N] ← createMainRules($key, $toggleDdir);
2  var $borderRules < Rule > [0...N] ← createBorderRules($key, $toggleDir);
3  var $preimage < Bit > [0...N] ← $plaintext;
4  var $image < Bit > [0...N] ← new < Bit > [0...N];
5  var $positionBorder < Int >← N − 2;
6  for N times do
7  |  forall [PARALLEL] 0 ≤ $i < N do
8  |  |  if $i == $positionBorder OR $i == $positionBorder + 1 then
9  |  |  |  $image[$i] ← apply($borderRules[$i], $preimage[$i-1, $i, $i+1]);
10 |  |  end
11 |  |  else
12 |  |  |  $image[$i] ← apply($mainRules[$i], $preimage[$i-1, $i, $i+1]);
13 |  |  end
14 |  end
15 |  $positionBorder ← rotateBorder($positionBorder, $toggleDir);
16 |  swap($preimage, $image);
17 end
18 return $preimage;
```

No Algoritmo 1, a função $createMainRules$ da linha 1 é responsável por agrupar os bits de $\$key$ (K) para criar o vetor $\$mainRules$ (K_M) composto de N regras principais. De forma correspondente, a função $createBorderRules$ da linha 2 é responsável pela criação do vetor $\$borderRules$ (K_B) para regras de borda. Em seguida, a variável $\$preimage$ é inicializada com um segmento de N bits do texto em claro, que corresponde aos valores iniciais do reticulado; enquanto isso, a variável $\$image$ é inicializada com um vetor vazio, pois os bits da próxima configuração do reticulado ainda são desconhecidos e serão calculados durante a execução do algoritmo. Além disso, a variável $\$positionBorder$ é inicializada com o índice $N-2$, que é a posição inicial da borda, mediante explicação que será detalhada na Seção 3.5.

O laço de repetição da linha 6 é responsável por garantir a execução de N evoluções do reticulado indicado pela variável $\$preimage$ até alcançar o texto cifrado resultante para o bloco de N células em questão. Com relação ao procedimento evolutivo em si, o cálculo de cada uma das células de índice $\$i$ de $\$image$ é feito, em paralelo, dentro do laço de execução estabelecido na linha 7 do algoritmo. A cada etapa, as duas células de borda estarão definidas pelos índices $\$positionBorder$ e $\$positionBorder + 1$, e essa variável $\$positionBorder$ é alterada pelo uso da função $rotateBorder$. Caso uma dada célula de índice $\$i$ pertença à borda, sua vizinhança será evoluída pela função *apply* utilizando a regra de borda contida em $\$borderRules[\$i]$; caso contrário, será evoluída utilizando a regra principal contida em $\$mainRules[\$i]$.

Ao fim de cada etapa evolutiva, que ocorre por meio do cálculo dos valores para preencher a variável $\$image$, o conteúdo dessas duas variáveis ($\$preimage$ e $\$image$) é trocado entre elas, por meio da função *swap*, para que a próxima etapa evolutiva possa ser iniciada partindo novamente da variável $\$preimage$. Conforme previamente mencionado, o processo de cifração no VHCA consiste em sucessivas aplicações da evolução tradicional de ACs sob um contexto altamente híbrido, onde cada célula pode, independentemente, ter sua vizinhança evoluída por uma regra distinta do alfabeto do método. Esse procedimento inicia com um segmento do texto em claro e produz um trecho cifrado correspondente, de mesmo tamanho N, que passou por N etapas evolutivas.

3.4 VHCA - Decifração

No mecanismo de cifração do VHCA, apresentado na Seção 3.3, cada bloco contendo parte do texto em claro é, no instante $t = 0$, um reticulado inicial s^0; as sucessivas evoluções desse reticulado, por meio das regras em K_M e K_B (extraídas a partir da chave criptográfica K), resultam na configuração s^{128} que vai compor o texto cifrado. A decifração, que é executada pelo processo de evolução inversa do AC, ocorre por meio do cálculo de pré-imagens, mediante o modelo de reversibilidade adaptado do método HCA.

Enquanto o método HCA é baseado em regras de $r = 4$, o que resulta em uma região de borda composta por 8 células, no VHCA são utilizadas regras de $r = 1$ que reduzem a região de borda a 2 células. Outra diferença entre os métodos é que, a cada iteração do HCA, todas as células da região principal têm suas vizinhanças evoluídas por uma mesma regra e todas as vizinhanças de células da região de borda também são evoluídas por uma mesma regra, distinta da utilizada na região principal; enquanto isso, no VHCA, cada vizinhança de célula pode ser evoluída por uma regra selecionada independentemente a partir da chave criptográfica, porém ainda todas com sensitividade à mesma direção (assim como também ocorre no HCA). Apesar das distinções citadas acima, o mecanismo de cálculo de pré-imagem no VHCA é muito similar ao descrito para o HCA na Seção 2.3.3.1. A Figura 23 exibe a sequência de cálculo das células de uma pré-imagem considerando regras sensíveis à esquerda.

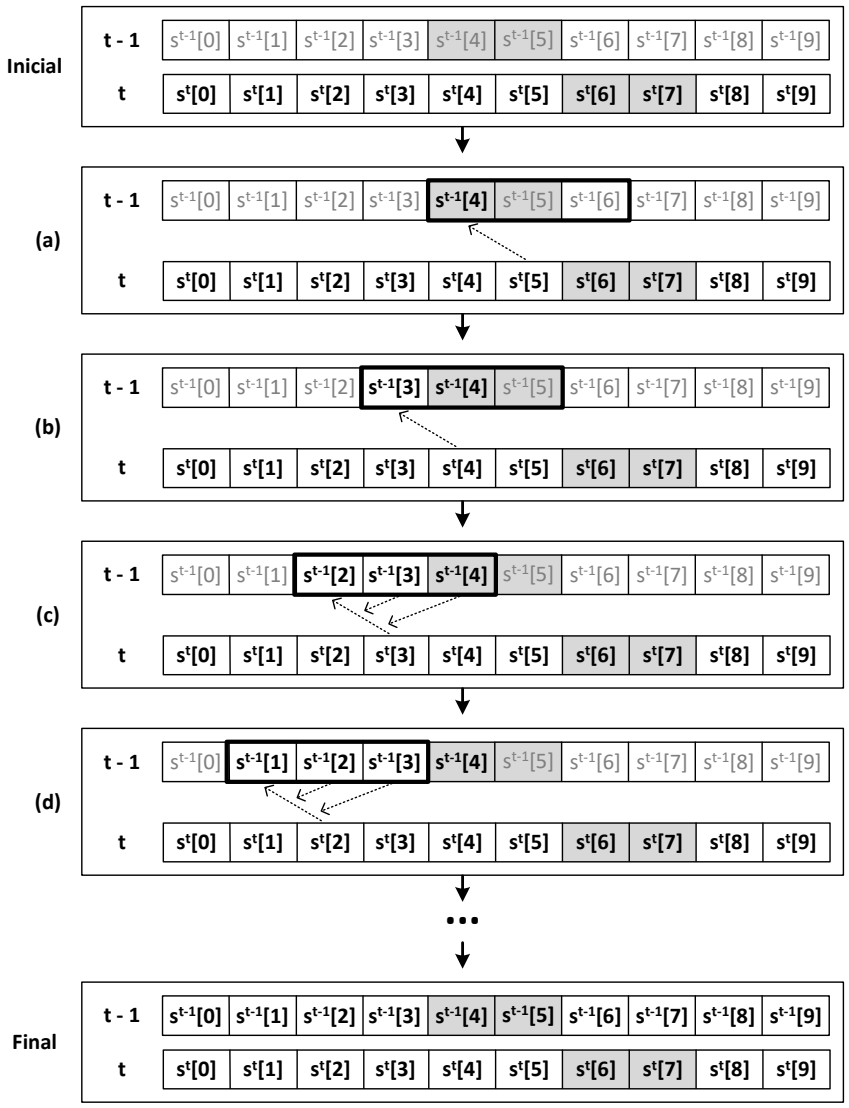


Figura 23 – Cálculo de pré-imagem VHCA - sensitividade à esquerda. [Fonte: autor]

Na Figura 23 considere o reticulado de uma configuração genérica s^t e também sua pré-imagem a ser calculada, s^{t-1}. Considere também que trata-se de uma execução do VHCA com regras de sensitividade à esquerda e a cor de fundo cinza nas células indica a região de borda em cada reticulado. Lembrando que a região de borda sofre uma rotação a cada passo iterativo conforme detalhamento na Seção 3.5. Também com relação à figura, as células da pré-imagem, cujo texto está acinzentado e com menos destaque, são aquelas cujo valor é desconhecido no instante retratado; porém, esses valores vão sendo calculados durante o procedimento, cujas iterações são indicadas verticalmente por letras do alfabeto.

A ordem de cálculo das células da pré-imagem é equivalente à adotada no método HCA; no caso, os valores são calculados na direção da sensitividade, iniciando com a região de borda, pela vizinhança da célula que esteja na extremidade em lado oposto à sensitividade. Nota-se que, para s^{t-1}, a região de borda compreende as células $s^{t-1}[4]$ e $s^{t-1}[5]$; entre elas, a célula mais à direita da borda é $s^{t-1}[5]$. Portanto, a partir de **(a)**, as vizinhanças serão selecionadas na seguinte ordem de células centrais: $s^{t-1}[5]$, $s^{t-1}[4]$, $s^{t-1}[3]$, ..., $s^{t-1}[6]$. Dada a condição de limite periódico, o reticulado tem comportamento toroidal; logo, em tempo, rotações são capazes de alcançar todas as células.

Começando em **(a)**, visto que $s^{t-1}[5]$ é uma célula da borda, considera-se que $s^t[5]$ é resultante da aplicação da regra $K_B[5]$ à vizinhança ($s^{t-1}[4]$, $s^{t-1}[5]$, $s^{t-1}[6]$). Entretanto, por $K_B[5]$ ser uma regra de borda e por apresentar sensitividade à esquerda, é possível inferir o valor de $s^{t-1}[4]$ a partir de $s^t[5]$, apesar de desconhecer os valores de $s^{t-1}[5]$ e $s^{t-1}[6]$. Dependendo de $K_B[5]$, tem-se que $s^{t-1}[4] = s^t[5]$ ou que $s^{t-1}[4] = \overline{s^t[5]}$. Passando à próxima vizinhança selecionada em **(b)**, tal vizinhança é centrada em $s^{t-1}[4]$ que também é uma célula de borda. Logo, de igual forma, é possível inferir o valor de $s^{t-1}[3]$, a partir do bit resultante $s^t[4]$, por meio da interpretação da regra de borda $K_B[4]$.

A partir da etapa **(c)** todas as vizinhanças selecionadas são de células da região principal; assim sendo, considera-se que tais vizinhanças estão sob influência de regras de K_M. A vizinhança selecionada em **(c)** é composta por ($s^{t-1}[2]$, $s^{t-1}[3]$, $s^{t-1}[4]$) e seu bit resultante da aplicação da regra $K_M[3]$ é $s^t[3]$. Logo, dado o conhecimento dos valores de $s^{t-1}[3]$ e $s^{t-1}[4]$, bem como desse bit resultante $s^t[3]$, é possível inferir o valor do bit determinante da sensitividade na pré-imagem, $s^{t-1}[2]$. Em seguida, em **(d)**, de igual forma, é possível inferir o valor de $s^{t-1}[1]$ a partir do conhecimento de $s^{t-1}[2]$, $s^{t-1}[3]$ e $s^t[2]$, por meio de análise da regra $K_M[2]$. Assim, consecutivamente, é possível calcular deterministicamente o valor de todos os bits restantes da pré-imagem s^{t-1}. E, por consequência, o mesmo mecanismo pode ser aplicado a outras pré-imagens a partir desta. A decifração no VHCA corresponde a aplicar N iterações de cálculo de pré-imagem ao segmento do texto cifrado; o pseudocódigo descrito em Algoritmo 2 exibe tal processo.

Algoritmo 2: VHCA - Decifração por Evolução Reversa de AC

Parâmetros: $\$key < Bit > [0...3 \times N]$, $\$ciphertext < Bit > [0...N]$, $\$toggleDir < Bit >$
Saída: $\$plaintext < Bit > [0...N]$

```
1  var $mainRules < Rule > [0...N] ← createMainRules($key, $toggleDir);
2  var $borderRules < Rule > [0...N] ← createBorderRules($key, $toggleDir);
3  var $preimage < Bit > [0...N] ← new < Bit > [0...N];
4  var $image < Bit > [0...N] ← $ciphertext;
5  var $positionBorder ← N − 2;
6  var $toggleShift ← getToggleShift($toggleDir);
7  for N times do
8  |  $positionBorder ← rotateBorderOpposite($positionBorder, $toggleDir);
9  |  forall 0 ≤ $i < N, sequentially from border cells in the toggle direction do
10 |  |  if $i == $positionBorder or $i == $positionBorder + 1 then
11 |  |  |  $preimage[$i + $toggleShift] ← toggleBit($borderRules[$i],
   |  |  |    $image[$i], $toggleDir);
12 |  |  end
13 |  |  else
14 |  |  |  $preimage[$i + $toggleShift] ← toggleBit($mainRules[$i],
   |  |  |    $image[$i], $preimage[$i-1, $i, $i+1], $toggleDir);
15 |  |  end
16 |  end
17 |  swap($preimage, $image);
18 end
19 return $image;
```

No Algoritmo 2, as funções $createMainRules$ e $createBorderRules$ têm a mesma responsabilidade já descrita para o Algoritmo 1 na Seção 3.3; no caso, criar respectivamente os vetores $\$mainRules$ e $\$borderRules$ à partir da chave $\$key$. Em seguida, a variável $\$preimage$ é inicializada com um vetor vazio, visto que os bits da pré-imagem ainda são desconhecidos e serão calculados durante a execução do algoritmo; enquanto isso, a variável $\$image$ é inicializada com um segmento de N bits do texto cifrado, que corresponde aos valores iniciais do reticulado. Com relação à variável $\$positionBorder$, ela é inicializada com o índice $N - 2$, que é a posição inicial da borda mediante explicação que será detalhada na Seção 3.5. Além disso, consta a inicialização da variável $\$toggleShift$ por meio da função $getToggleShift$ que indica a posição, relativa à célula central, do bit determinante da sensitividade na vizinhança. Em caso de sensitividade à esquerda, $\$toggleShift = -1$; em caso de sensitividade à direita, $\$toggleShift = 1$.

O laço de repetição da linha 7 é responsável por garantir a execução de N sucessivas iterações do cálculo de pré-imagem da configuração em $\$image$, até alcançar o texto cifrado resultante para o bloco em questão. Para fins didáticos, este pseudocódigo apresenta o processo de decifração como um processo de consecutivos cálculos de pré-imagem, sem paralelismo; entretanto, conforme será apresentado na Seção 3.6, é teoricamente possível aplicar um certo grau de paralelismo a esse mecanismo. Assim como na cifração, uma rotação da região de borda é realizada por meio da mudança no valor da variável $\$positionBorder$; entretanto, na decifração, isso é feito por meio da função $rotateBorderOpposite$, enquanto na cifração isso é feito pela função $rotateBorder$. Tal diferença evidencia o fato de que essas rotações ocorrem no sentido inverso uma da outra, fato melhor detalhado na Seção 3.5.

Com relação ao laço de execução definido na linha 9, ele define um índice i, que percorre todas as células do reticulado começando pela região de borda. Cada célula indicada por i tem sua vizinhança e, a cada valor que i assume, é feito um cálculo para descobrir o valor do bit determinante da sensitividade dessa vizinhança ($\$preimage[\$i + \$toggleShift]$). Quando a célula $\$preimage[\$i]$ pertence à borda, a regra de borda $\$borderRules[\$i]$ é utilizada para inferir o valor do bit determinante na função $toggleBit$; porém, em caso contrário, a regra principal $\$mainRules[\$i]$ será utilizada.

3.5 VHCA - Movimentação da Borda

A movimentação da região de borda ocorre a cada etapa do algoritmo VHCA para amenizar a influência das regras de borda com relação ao comportamento geral do AC, visto que a simplicidade de tais regras poderia representar uma vulnerabilidade criptográfica. Como os procedimentos de cifração e decifração representam inversos entre si, as movimentações da borda em cada um deles ocorrem em direções opostas para garantir que haja equivalência entre os passos intermediários; ou seja, a região de borda considerada no primeiro passo da decifração deve ser tal que permita desfazer o processo realizado na última etapa da cifração, e assim por diante.

Na cifração, a região de borda movimenta-se numa direção oposta à sensitividade adotada; ou seja, em caso de usar regras com sensitividade à esquerda, a região de borda movimenta-se para a direita a cada evolução temporal $s^t \rightarrow s^{t+1}$. De forma correspondente, fica visível uma movimentação da borda à esquerda quando regras com sensitividade à direita estão sendo utilizadas para a evolução do reticulado. Esse deslocamento, aplicado a cada passo, sempre ocorre por duas células de distância (o tamanho da região de borda), conforme Figura 24.

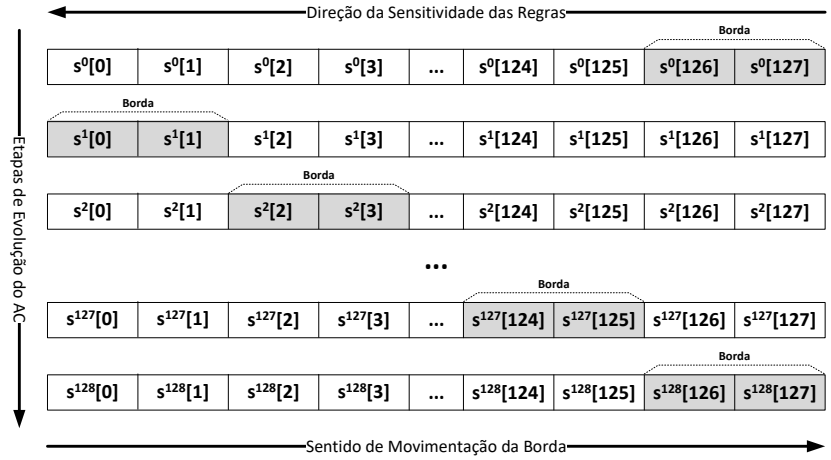


Figura 24 – Rotação borda em evolução de AC - sensitividade esquerda. [Fonte: autor]

Na Figura 24 são apresentados, de cima para baixo, uma sequência de reticulados que representam a sucessão de evoluções do AC feitas a partir da configuração inicial s^0 até a configuração final s^{128}. Apesar da omissão intencional de algumas configurações e das células centrais do reticulado, fica visível o deslocamento rotativo da borda, destacada pelo fundo cinza, em direção à direita (oposta à da sensitividade esquerda) por duas células de distância a cada etapa evolutiva.

Em relação à decifração, considerando um AC de regras também com sensitividade à esquerda, ocorre um deslocamento no mesmo sentido da sensitividade a cada cálculo de pré-imagem; ou seja, a região de borda movimenta-se para a esquerda, conforme a Figura 25. O mesmo raciocínio é aplicável no caso de regras com sensitividade à direita, dados os devidos ajustes, sem perda de generalidade.

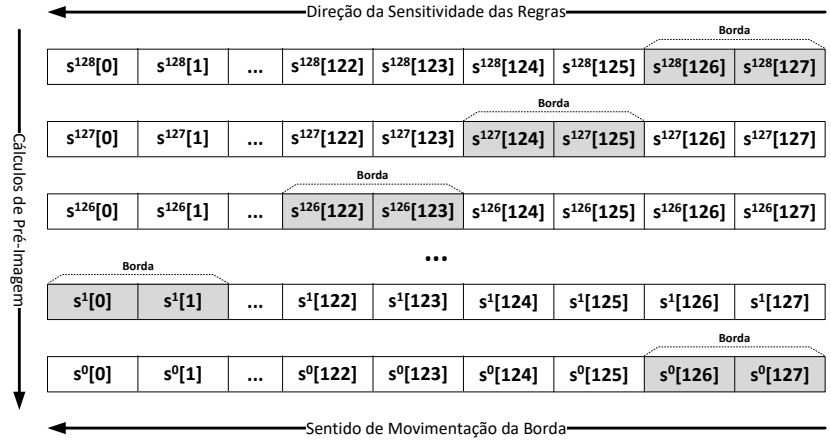


Figura 25 – Rotação borda cálculo pré-imagem - sensitividade esquerda. [Fonte: autor]

Com relação à Figura 25, ela demonstra verticalmente, de cima para baixo, a sequência de reticulados no cálculo de pré-imagens partindo da configuração s^{128}, que contém o trecho cifrado, até a configuração s^0, que contém o trecho equivalente do texto em claro original. Para esse procedimento de decifração, a rotação da borda ocorre na mesma direção que a sensitividade para que haja uma equivalência na posição da borda, durante ambos os processos de cifração e decifração, para qualquer configuração s^t intermediária.

Uma observação relevante é que, dada a quantidade de etapas do VHCA ser igual ao tamanho N do reticulado, visto que o deslocamento da borda é de duas células por etapa, e considerando que N é um número par, temos que a posição de borda das configurações s^0 e s^N será a mesma, independente da direção de sensitividade. Além disso, a definição de como essa rotatividade seria implementada levou em consideração também uma tentativa de otimização do paralelismo no cálculo de pré-imagem, fator descrito na Seção 3.6.

3.6 VHCA - Paralelismo ao Decifrar

Para fins de simplificação, o processo de decifração do VHCA apresentado na Seção 3.4 foi detalhado como uma sequência de cálculos consecutivos de pré-imagem, partindo da configuração s^{128} até alcançar s^0. Entretanto, um dos motivos para a aplicação do mecanismo de rotação de borda é adiantar a disponibilidade de valores de células que permitem o cálculo da pré-imagem seguinte. Ou seja, possibilitar que, dado um momento de tempo genérico t, o cálculo de células de s^{t-1} não dependa do conhecimento prévio de todos os valores de células de s^t. Nesse caso, os valores de células em instantes de tempo diferentes poderão ser calculados paralelamente em hardware especializado. A Figura 26 exemplifica a aplicação de tal paralelismo no cálculo de pré-imagens ao decifrar.

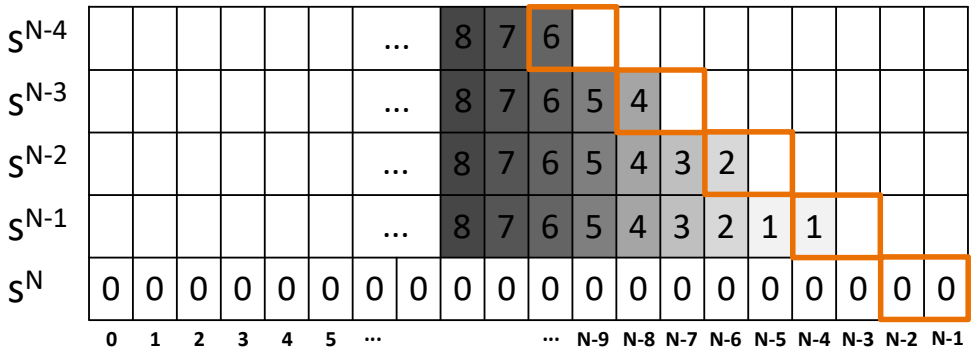


Figura 26 – Paralelismo da decifração no VHCA - sensitividade esquerda. [Fonte: autor]

Supondo que a operação de cálculo do valor de cada célula da pré-imagem é realizada em um ciclo de processamento, ou seja, é um cálculo de tempo constante $\mathcal{O}(1)$, têm-se que os números nas células da Figura 26 indicam o ciclo de processamento a partir do qual o valor da célula em questão estaria disponível para ser utilizado em cálculo de outras células. A borda laranja destaca a região de borda em cada etapa t, para as etapas indicadas por seus reticulados s^t representados nas linhas da tabela. Considerando uma execução com regras de sensitividade à esquerda, para qualquer célula de borda $s^{t-1}[i]$, $s^{t-1}[i-1]$ pode ter seu valor inferido assim que o valor de $s^t[i]$ estiver disponível. E, assim que duas células iniciais consecutivas de s^{t-1} forem inferidas usando regras de borda, as células restantes $s^{t-1}[i-1]$ podem ser inferidas, em sequência, usando regras principais quando cada $s^t[i]$ correspondente estiver disponível.

Uma implementação sequencial da decifração teria tempo de execução quadrático, $\mathcal{O}(N^2)$, visto que o cálculo de cada pré-imagem s^{t-2} somente seria iniciado após o cálculo de todas as N células de s^{t-1}. Entretanto, considerando a paralelização teórica de todas as células que constam com um mesmo número na Figura 26, seria possível completar a decifração em $3 \times (N-1)$ ciclos de processamento, ou seja, o tempo quadrático passa a ser otimizado para uma execução em tempo linear $\mathcal{O}(N)$.

Conforme previamente mencionado, até onde sabemos, o método VHCA é único por ter, simultaneamente, as características de: ser uma cifra de bloco baseada em AC unidimensional híbrido, cuja robustez criptográfica é alcançada pela utilização de um arranjo de regras elementares ($r = 1$) onde a maioria das regras tem dinâmica não-linear; ser um método onde o comprimento do texto cifrado resultante é igual ao do texto claro original; e ter a maleabilidade de ser facilmente adaptável para um espaço de chaves maior.

Capítulo 4

Experimentos e Análise dos Resultados

Conforme detalhado no capítulo anterior, a cifração no VHCA é implementada pela aplicação consecutiva da operação de evolução tradicional de AC em reticulados que foram inicializados com segmentos do texto em claro. Assim sendo, podemos inferir que a robustez criptográfica do método é consequência do próprio mecanismo de alteração de valores das células, ocasionado pela aplicação de regras de AC às vizinhanças das células.

Devido à repetição consecutiva da evolução de reticulados, desde a configuração inicial $s^{t=0}$ até alcançar a configuração final $s^{t=N}$, é possível observar um alto nível de interação entre os valores de células que foram computados a partir da aplicação de regras distintas extraídas da chave criptográfica K. Além disso, é importante salientar que, conforme apresentado na Seção 3.2, uma chave válida para uso no método VHCA deve apresentar um certo grau de diversidade em sua sequência de bits, visto que as regras do AC serão extraídas diretamente da chave K.

Os algoritmos criptográficos simétricos atuam sobre um texto em claro para, por meio de artifícios como a permutação e substituição de dados, modificá-lo a ponto de este tornar-se irreconhecível. Ou seja, a qualidade de tal algoritmo está claramente relacionada à diferença que o método impõe entre o texto em claro e o texto cifrado. Neste sentido, alguns dos critérios fundamentais para analisar essa alteração causada ao cifrar são:

- **Dispersão**: É desejável que as mudanças impostas estejam distribuídas por toda a extensão do texto resultante e não confinadas a um trecho deste.
- **Impacto**: Em um algoritmo de qualidade, alterações mínimas no texto em claro ou na chave criptográfica resultam em grande alteração no texto cifrado.
- **Aleatoriedade**: As alterações impostas ao texto em claro não devem seguir qualquer padrão estatístico causador de previsibilidade.

As próximas seções apresentam os resultados das análises de segurança aplicadas ao VHCA para validar sua robustez criptográfica, bem como os resultados de análises de desempenho. Em tais experimentos, para fins de comparação onde for cabível, serão exibidos também os resultados referentes a sequências produzidas pelo método predecessor HCA e outras produzidas pelo método AES. No caso do AES, foi utilizada a implementação nativa da *Microsoft* para a linguagem .NET *C#* (IEVANGELIST, 2022). No caso do HCA e do VHCA foram utilizadas implementações próprias do autor, disponibilizadas publicamente.

O VHCA pode ser aplicado na cifração de blocos de N bits, sendo que neste trabalho adotou-se $N = 128$ por similaridade com o AES, atual método de referência em criptografia de bloco. Apesar dessa convenção, o VHCA é naturalmente adaptável a outros tamanhos. De forma geral, por estar lidando com regras elementares, para um bloco de N bits, é necessária a evolução do reticulado por N passos de tempo para possibilitar que o efeito da alteração de um bit seja propagado a todo o bloco. Para fins comparativos, os experimentos desta seção referentes a análises de robustez criptográfica incluem também resultados para variantes do VHCA com $N = 64$, $N = 192$ e $N = 256$.

A menos que especificado de outra forma, os textos em claro, as chaves criptográficas e os vetores de inicialização dos experimentos serão inicializados com valores produzidos pelo gerador de números pseudo-aleatórios (*PseudoRandom Number Generator* (PRNG)) "Mersenne Twister" (MATSUMOTO; NISHIMURA, 1998). Todos os experimentos aqui listados podem ser reproduzidos por meio do código-fonte publicamente disponibilizado.

A Seção 4.1 descreve o mecanismo utilizado para geração de sequências, a partir de métodos criptográficos, nos experimentos realizados. A Seção 4.2 detalha o chamado "efeito avalanche" que avalia o critério "impacto", enquanto as Seções 4.3 e 4.4 exibem, respectivamente, os resultados obtidos pelo método VHCA quando avaliados nos critérios de "dispersão" e "aleatoriedade" por meio das baterias de testes estatísticos NIST e PractRand. A Seção 4.5 traz medições e comparativos de desempenho do método. A Seção 4.6 contém uma exposição de ataques criptográficos convencionais da literatura e uma análise de como o algoritmo resiste a eles. Por fim, a Seção 4.7 contém discussões relevantes ao método e comentários adicionais sobre os experimentos realizados.

4.1 Mecanismo XOR

O termo "XOR" denota o uso do operador lógico ou-exclusivo, onde duas sequências de bits são linearmente comparadas e uma nova sequência resultante é gerada. Em outras palavras, dadas as sequências de entrada A e B, uma nova sequência Z é calculada por:

$$Z = A \oplus B \tag{6}$$

Dessa forma, a sequência binária Z expressa em quais posições houve divergência nos valores dos bits entre A e B.

O mecanismo XOR pode ser aplicado de forma relevante no contexto da avaliação de métodos criptográficos, visto que esse texto da diferença, Z, quando calculado para um texto em claro A e o texto cifrado B (resultante da aplicação de um método criptográfico), expressa os bits alterados pela execução de tal algoritmo. A sequência Z é fruto da comparação entre A e B, minimizando a relevância da entropia interna de qualquer uma das duas sequências originais.

No caso, a sequência Z indica tanto a quantidade de bits alterados quanto a distribuição espacial de tais alterações; deste modo, qualquer padrão de dispersão, impacto e aleatoriedade pode ser extraído a partir da análise dessa sequência. A aplicação de ferramentas estatísticas para analisar Z é uma etapa tradicional ao medir a robustez de algoritmos de cifra (BOESGAARD et al., 2003; TURAN; DOGANAKSOY; CALIK, 2006; SIMION, 2015).

4.2 Efeito Avalanche

O teste de efeito avalanche foi aplicado ao VHCA, considerando o efeito da avalanche de texto em claro e de chave criptográfica, a fim de avaliar a robustez do método a esse tipo de ataque. Para ambos os casos, considerou-se como “mínima” a mudança de valor em um único bit de posição aleatória. Em outras palavras, no caso de avalanche do texto em claro, o novo texto em claro difere do original por um único bit e a chave criptográfica segue inalterada. Já no caso de avalanche da chave, o novo texto em claro é idêntico ao original, e a nova chave criptográfica difere da original por um único bit.

Conforme previamente descrito, para cada avaliação, os resultados foram calculados a partir de 100.000 conjuntos de execuções distintas da avaliação indicada. Todas as chaves dos experimentos foram geradas aleatoriamente, enquanto que metade dos textos claros foi gerada aleatoriamente e a outra metade foi formada por padrões simples (ex: “000...0”, “111...1”, “1010...10”, etc.), visando tornar mais justa a comparação entre os algoritmos. Como existem duas formas de avaliação do efeito avalanche, "texto claro"e "chave", os resultados também foram consolidados em tabelas separadas para facilitar a análise.

Para cada avaliação, o teste avalanche foi aplicado a cada um dos 100.000 dados de entrada iniciais, resultando em dois textos cifrados por execução. Então, o mecanismo XOR é aplicado sobre os pares de textos cifrados (A e B), resultando na sequência binária Z de N bits. Com base nessa sequência, é contabilizado o percentual de diferenças encontradas entre os bits dos textos (quantidade de 1’s em relação ao total de bits). As Tabelas 8 e 9 apresentam os resultados consolidados do efeito avalanche de texto claro e chave, respectivamente. Nelas são mostrados o menor e o maior percentual encontrado, bem como o valor médio e o desvio padrão (σ), considerando todas as execuções avaliadas.

Tabela 8 – Resultados - Efeito Avalanche de Texto Claro

Avaliação	**Mínima**	**Máxima**	**Média**	σ
AES	32,03%	69,53%	49,98%	4,41%
HCA	28,12%	68,75%	49,97%	4,41%
VHCA	26,56%	68,75%	49,49%	4,64%
VHCA-64	17,18%	75,00%	48,98%	6,77%
VHCA-192	31,77%	64,58%	49,70%	3,73%
VHCA-256	34,37%	64,45%	49,79%	3,21%

Todos os resultados referentes ao efeito avalanche de texto claro (Tabela 8) apresentam média de bits alterados próxima a 50%, conforme esperado de métodos criptográficos adequados. Entretanto, os resultados do experimento "VHCA-64" apresentam entropia mínima inferior e desvio padrão superior aos demais. Tais resultados indicam que a variante do VHCA para $N = 64$ não apresentaria robustez criptográfica satisfatória. O VHCA padrão, $N = 128$, teve comportamento satisfatório, e as variantes $N = 192$ e $N = 256$ demonstraram ótimos indicadores.

Tabela 9 – Resultados - Efeito Avalanche de Chave

Avaliação	**Mínima**	**Máxima**	**Média**	σ
AES	29,69%	73,44%	50,00%	4,41%
HCA	30,47%	67,97%	50,00%	4,43%
VHCA	30,47%	67,97%	50,00%	4,42%
VHCA-64	23,44%	76,56%	50,01%	6,26%
VHCA-192	35,42%	65,63%	50,00%	3,61%
VHCA-256	35,16%	64,06%	50,01%	3,13%

Os resultados da Tabela 9, referentes ao efeito avalanche de chave, apresentaram situação similar aos resultados analisados para avalanche de texto claro. Como pode ser observado, os resultados do experimento "VHCA-64" novamente apresentam entropia mínima inferior e desvio padrão superior aos demais. Por outro lado, o VHCA padrão e suas variantes $N = 192$ e $N = 256$ apresentaram resultados próximos ou superiores aos outros métodos.

O método VHCA apresenta um efeito avalanche satisfatório em ambos os cenários, com desvio padrão (σ) comparável ao obtido pelo algoritmo AES. Conforme descrito na Seção 2.1.2.2, a avaliação do efeito avalanche espera uma mudança de aproximadamente 50% no valor dos bits como resultado de qualquer alteração mínima, seja no texto simples ou na chave, supondo a utilização de um método criptográfico robusto. Após validar essa porcentagem média de alteração, as seções seguintes trazem experimentos que avaliam a qualidade da distribuição de tais alterações pela extensão dos textos cifrados resultantes.

4.3 Bateria de Testes Estatísticos do NIST

O *National Institute of Standards and Technology* (NIST), instituto americano de padrões e segurança fundado em 1901, divulgou em 2010 um conjunto de testes estatísticos que avaliam a qualidade de sequências pseudo-aleatórias (BASSHAM et al., 2010a). Essa ferramenta tem sido amplamente utilizada pela comunidade criptográfica para validação dos resultados obtidos com novos algoritmos (RAHIMOV; BABAEI; FARHADI, 2011; RADWAN; ABDELHALEEM; ABD-EL-HAFIZ, 2016; LAIPHRAKPAM; KHUMANTHEM, 2017). Os testes aplicados pelo NIST são:

- T01 - *Frequency (Monobits)*
  Verifica se há aproximadamente o mesmo número de 0s e 1s na sequência inteira.

- T02 - *Frequency within a Block*
  Verifica se há equilíbrio entre 0s e 1s dentro de blocos de tamanho fixo.

- T03 - *Runs*
  Avalia se a quantidade de sequências contínuas de 0s e 1s está dentro do esperado.

- T04 - *Longest Run of Ones in a Block*
  Verifica se o maior grupo contínuo de 1s dentro de blocos tem tamanho compatível com uma sequência aleatória.

- T05 - *Random Binary Matrix Rank*
  Avalia a dependência linear entre submatrizes da sequência.

- T06 - *Discrete Fourier Transform (Spectral)*
  Detecta padrões repetitivos (não aleatórios) na sequência por meio da transformada de Fourier.

- T07 - *Non-Overlapping (Aperiodic) Template Matching*
  Conta quantas vezes um padrão fixo ocorre sem sobreposição na sequência.

- T08 - *Overlapping (Periodic) Template Matching*
  Conta ocorrências de padrões fixos mesmo quando se sobrepõem, detectando padrões repetitivos.

- T09 - *Maurer's Universal Statistical*
  Avalia se a sequência pode ser comprimida, pois sequências facilmente comprimíveis não são aleatórias.

- T10 - *Linear Complexity*
  Mede a complexidade da sequência com base no tamanho do registrador de realimentação necessário para gerá-la.

- T11 - *Serial*
  Verifica se todos os padrões de bits sobrepostos ocorrem com frequência semelhante ao esperado.

- T12 - *Approximate Entropy*
  Compara a frequência de padrões sobrepostos de tamanho m e m+1 para medir aleatoriedade local.

- T13 - *Cumulative Sums (Cusum)*
  Avalia o quanto a soma cumulativa dos bits, em uma caminhada aleatória, diverge do valor esperado que é zero.

- T14 - *Random Excursions*
  Conta quantas vezes um estado específico é visitado em ciclos de uma caminhada aleatória.

- T15 - *Random Excursions Variant*
  Verifica quantas vezes certos estados ocorrem em uma caminhada aleatória, buscando desvios do esperado.

Cada um desses 15 testes verifica a existência de um tipo específico de padrão estatístico na distribuição de bits da sequência avaliada. Caso o padrão seja encontrado, o teste correspondente "falha" apresentando uma porcentagem de certeza estatística. Maiores detalhes sobre esses testes estatísticos estão disponíveis na documentação do NIST (BASSHAM et al., 2010b).

Neste experimento, foram avaliados 1.000 arquivos binários de 10 megabytes, os quais foram gerados pela concatenação de sequências resultantes da aplicação do mecanismo XOR, descrito na Seção 4.1. Na geração de cada arquivo, uma chave K e um texto binário inicial T_0 foram inicializados aleatoriamente. Então, o texto T_0 foi cifrado pelo método criptográfico em questão, aplicando a chave K, gerando o texto cifrado T_1. Em seguida, o segmento $T_0 \oplus T_1$ foi adicionado ao novo arquivo. Após isso, o texto T_1 foi cifrado pelo mesmo método e chave, para gerar T_2. De forma análoga, $T_1 \oplus T_2$ foi adicionado ao novo arquivo. Tal processo é repetido até que o arquivo alcance o tamanho esperado.

Conforme previamente mencionado, o uso da operação XOR entre o texto em claro original e o seu texto cifrado resultante gera uma sequência Z que expressa diretamente a alteração imposta pelo método criptográfico, minimizando a influência dos dados originais. Dessa forma, tal concatenação de sequências Z gera um arquivo que demonstra a forma de atuação do algoritmo em questão. Tais arquivos foram avaliados pela bateria de testes NIST e desses conjuntos foi extraída a porcentagem de aprovação dos métodos criptográficos avaliados (AES, HCA e VHCA), considerando cada um dos 15 testes, conforme mostrado na Tabela 10.

Tabela 10 – Resultados da bateria de testes estatísticos NIST

Teste	**AES**	**HCA**	**VHCA**	**VHCA-64**	**VHCA-192**	**VHCA-256**
T01	98,4%	99,6%	98,9%	98,8%	98,8%	99,0%
T02	98,9%	99,4%	99,1%	99,1%	99,0%	99,1%
T03	99,0%	99,1%	98,8%	98,8%	99,1%	98,6%
T04	98,1%	99,2%	99,0%	98,6%	97,5%	98,6%
T05	99,2%	99,2%	99,3%	99,3%	99,2%	99,0%
T06	99,1%	99,4%	99,0%	99,3%	99,4%	98,3%
T07	97,8%	97,1%	97,1%	97,9%	98,0%	98,2%
T08	99,0%	99,3%	99,1%	99,1%	99,1%	99,2%
T09	99,1%	99,3%	99,0%	98,9%	98,8%	98,9%
T10	98,3%	98,9%	98,9%	99,1%	99,2%	99,0%
T11	98,4%	98,4%	98,7%	98,0%	98,1%	98,6%
T12	98,8%	98,7%	98,8%	98,5%	99,0%	99,0%
T13	98,0%	99,2%	99,0%	98,3%	98,7%	98,2%
T14	92,1%	93,0%	93,4%	92,3%	93,7%	94,2%
T15	92,8%	93,9%	93,2%	91,1%	92,8%	93,2%

Como pode ser observado na Tabela 10, não há diferença significativa de desempenho entre os métodos avaliados, sendo que todos apresentaram altos índices de sucesso nos testes estatísticos investigados.

4.4 Bateria de Testes Estatísticos PractRand

Embora o conjunto de testes NIST seja considerado o padrão para testes estatísticos PRNG, outras alternativas estão sendo desenvolvidas, dentre as quais destaca-se o conjunto de testes PractRand. O nome PractRand vem do inglês "*Practically Random*" (praticamente aleatório). Esse conjunto de testes estatísticos teve sua versão inicial lançada em 2010 e recebeu sua atualização mais recente em 2018 (DOTY-HUMPHREY, 2018). O PractRand é proposto como uma melhoria para a clássica bateria de análise estatística Diehard/Dieharder (BROWN; EDDELBUETTEL; BAUER, 2006). E, apesar de não ter o renome do NIST, o PractRand está sendo cada vez mais utilizado e reconhecido como uma ferramenta promissora de testes estatísticos (ÁLVAREZ; ZAMORA, 2015; CIGLARIČ; ŠTRUMBELJ et al., 2019; SLEEM; COUTURIER, 2020).

O PractRand executou 140 análises para os mesmos 1.000 arquivos binários de 10 megabytes testados no NIST e os resultados foram agrupados pelo nome do teste em 06 grupos: G01 - BCFN (33 testes); G02 - BRank (9 testes); G03 - DC6-9x1Bytes-1 (4 testes); G04 - FPF-14+6/16 (53 testes); G05 - Gap-16 (8 testes); G06 - mod3n (33 testes). A Tabela 11 apresenta o resultado obtido pelos métodos criptográficos (AES, HCA e variações do VHCA) em cada um dos grupos.

Tabela 11 – Resultados dos testes estatísticos PractRand.

Testes	**AES**	**HCA**	**VHCA**	**VHCA-64**	**VHCA-192**	**VHCA-256**
G01	97,7%	97,4%	98,0%	98,3%	98,1%	97,2%
G02	100,0%	100,0%	99,9%	100,0%	100,0%	100,0%
G03	96,1%	97,4%	97,4%	97,7%	96,5%	95,1%
G04	99,2%	99,2%	99,4%	99,1%	99,2%	99,1%
G05	99,1%	98,3%	99,2%	99,0%	98,6%	98,8%
G06	99,8%	99,5%	99,9%	99,2%	99,8%	99,6%

Para cada grupo de testes listado na Tabela 11, uma entrada de arquivo só é contabilizada como aprovada se for validada por todos os testes individuais que compõem aquele grupo. Sendo assim, os valores correspondem à porcentagem de arquivos aprovados nos testes do referido grupo para cada um dos métodos criptográficos avaliados. Como pode ser observado, as versões do VHCA apresentam taxas de aprovação comparáveis ao AES e HCA, com o VHCA padrão alcançando os maiores valores absolutos em 5 dos 6 grupos avaliados. Tal desempenho é um indício da robustez criptográfica do método.

4.5 Análise de Desempenho

Os resultados apresentados nesta seção foram obtidos a partir de execuções sequenciais (sem paralelismo) dos algoritmos em um computador com arquitetura x86-64 tradicional. Tal ambiente não favorece o uso do paralelismo inerente aos ACs, o qual só é possível mediante o uso de *Field Programmable Gate Array* (FPGA) ou GPUs com memória compartilhada que permitam o armazenamento de todo o reticulado. Assim, é possível eliminar a necessidade de sincronização dos estados das células a cada passo de evolução do AC. Portanto, são considerados apenas os tempos alcançados a partir de execuções sequenciais dos algoritmos criptográficos. Além disso, quando comparado ao algoritmo AES, é necessário considerar a vantagem de otimizações a nível de processador que favorecem o desempenho de tal algoritmo, como, por exemplo, o conjunto de instruções AES-NI (HOFEMEIER; CHESEBROUGH, 2012). Experimentos feitos demonstram um ganho de desempenho de até $13,5x$ ao aplicar tais instruções na implementação do AES (ABDALLAH; KUANG; HUANG, 2020).

A Tabela 12 exibe os tempos de cifração e decifração para os algoritmos AES, HCA e VHCA, considerando a aplicação de tais operações em um único bloco de dados com 128 bits. Os tempos alcançados referem-se à execução em um único núcleo do processador Intel Core i5 12400F, a uma frequência aproximada de 4.40 GHz. Independentemente do processador em questão, o foco desejado é a noção comparativa de desempenho entre os métodos.

Tabela 12 – Custo computacional das operações em blocos de 128 bits

Operação	**Tempo de Execução**
AES - Cifração	1.006,6 ns
AES - Decifração	988,6 ns
HCA - Cifração	94.786,2 ns
HCA - Decifração	215.087,4 ns
VHCA - Cifração	70.196,1 ns
VHCA - Decifração	71.006,7 ns

Como pode ser observado na Tabela 12, o AES foi superior aos demais métodos, apresentando um menor tempo de execução tanto na cifração quanto na decifração. Também é possível notar um desempenho superior do método VHCA em relação ao HCA, demonstrando que o uso de arranjos de regras elementares como chave criptográfica é capaz de reduzir o custo computacional das operações. Também é importante notar que o novo método reduz consideravelmente a diferença entre tempos de cifração e decifração em relação ao seu predecessor, apresentando um comportamento mais próximo ao AES.

4.6 Ataques Convencionais da Literatura

Devido ao método VHCA apresentar alta conformidade com o efeito avalanche, mediante estabelecido na Seção 4.2, têm-se fortes indicativos de que ataques de texto em claro ou chave criptográfica escolhidos não seriam úteis para reduzir a força criptográfica do VHCA por extração de informações relevantes (DAWSON; GUSTAFSON; PETTITT, 1992; BIHAM; SHAMIR, 1991). Apesar disso, é altamente recomendável que a implementação do VHCA empregue um modo de operação considerado seguro para reduzir o risco de ataques diferenciais. Para tal finalidade, os modos de operação ECB e CBC não são mais considerados suficientemente seguros (VAUDENAY, 2002).

O algoritmo VHCA também é resistente à criptoanálise linear, visto que todas as regras de AC do conjunto principal do alfabeto da chave são representadas por expressões não-lineares. Somente as regras de borda são de natureza linear, entretanto, a influência delas é mínima. A cada etapa de tempo t do AC apenas duas células do reticulado são evoluídas por tais regras lineares de borda, enquanto as 126 células restantes são evoluídas usando regras principais não-lineares. Além disso, a posição da borda é rotacionada a cada passo de tempo, o que dispersa o efeito linear que a região de borda poderia produzir na dinâmica geral de evolução do AC. Portanto, a dinâmica resultante da evolução do AC (tanto na evolução tradicional quanto na reversa) não é linear.

Com relação aos ataques direcionados a algoritmos criptográficos de AC, o ataque Meier-Staffelbach diz respeito a um viés encontrado especificamente na regra 30 (MEIER; STAFFELBACH, 1991b). Ao usar um arranjo de 128 regras de raio-1 como sua chave criptográfica, o VHCA se torna mais robusto a vulnerabilidades específicas encontradas individualmente nessas regras, e os critérios de entropia usados para definir chaves válidas do VHCA garantem uma alta variabilidade na sequência binária que dá origem à chave K e, consequentemente, nos vetores de regras geradas por tal chave. Assim sendo, o efeito de nenhuma regra em particular domina o comportamento geral do AC no VHCA, reduzindo o risco de vulnerabilidade específica a alguma regra.

Embora possam ser feitas análises mais avançadas com relação à robustez do VHCA, os indicativos obtidos no presente trabalho corroboram a hipótese de que o VHCA apresenta qualidades criptográficas relevantes.

4.7 Comentários Adicionais

O VHCA pode ser aplicado na cifração de blocos de N bits, onde sugere-se $N \geq 128$ bits, sendo que neste trabalho os experimentos foram realizados com $N = 128$ por similaridade com o AES, método de criptografia simétrica mais difundido atualmente. Neste caso, a cifração é efetuada pela aplicação do arranjo de regras na atualização do reticulado, por $N = 128$ passos de tempo. Por outro lado, a decifração é obtida por meio de $N = 128$ aplicações sucessivas do cálculo de pré-imagens a partir do texto cifrado. Nada impede que sejam utilizados blocos de tamanho maior que 128 bits, pois o método é naturalmente adaptável a outros tamanhos. Entretanto, quanto maior o tamanho do bloco, maior o número de passos necessários na evolução temporal do AC para obter-se uma boa propagação de perturbações. De forma geral, para um bloco de N bits, é necessária a aplicação das regras por N passos de tempo.

A forma como o arranjo de regras sensíveis é construído garante que o comportamento emergente na cifração tenha uma alta entropia sem, entretanto, comprometer a reversibilidade do método que é necessária para a etapa de decifração. O arranjo de N regras sensíveis possui dois tipos de regras: não-lineares, chamadas aqui de "regras principais", e lineares, chamadas aqui de "regras de borda". A cada passo de tempo, $N-2$ células do reticulado são evoluídas utilizando regras principais, sendo que, devido à sua característica não-linear, resultam em um comportamento emergente caótico do reticulado como um todo, imprimindo a entropia que é importante a todo método criptográfico. Ao mesmo tempo, apenas 2 células do reticulado são atualizadas pelas regras de borda que são lineares. Essa região de borda composta por células evoluídas usando regras lineares garante a existência de um cálculo determinístico da pré-imagem de mesmo comprimento N para qualquer configuração.

O uso do arranjo híbrido com regras de borda lineares foi proposto no método predecessor HCA (LIRA et al., 2023), descrito na Seção 2.3.3, que utiliza regras de $r = 4$ com sensitividade. Entretanto, no HCA, o arranjo é composto apenas por uma única regra principal, aplicada em $N - 8$ células, e outra regra de borda, aplicada nas 8 células restantes do reticulado. Assim, para obter um espaço de chaves seguro no HCA, este foi concebido com regras de raio maior que 1, escolha esta que elevou a complexidade da implementação do método, além de aumentar o atraso entre os cálculos sucessivos de pré-imagem, mesmo considerando uma possível implementação paralela.

Embora o método de Gutowitz e o HCA tenham proposto o uso do cálculo de pré-imagens na etapa de cifração, o método VHCA, objeto desta tese, aplica esses mecanismos de forma inversa, ou seja, a evolução tradicional na cifração e o cálculo de pré-imagem na decifração. Dessa forma, no VHCA, a cifração é mais rápida que a decifração, enquanto o oposto ocorre nos métodos predecessores citados acima. Na verdade, como a evolução tradicional de ACs e o cálculo de pré-imagem são operações inversas, é possível trocar os mecanismos em qualquer um dos algoritmos citados da forma mais conveniente. Entretanto, embora a evolução para frente possa ser calculada com maior paralelismo e velocidade, sabemos por outros experimentos complementares que, em relação ao cálculo de pré-imagem, essa operação exige um número de passos de tempo maior para uma propagação de qualquer perturbação. Isso ocorre porque, para cada passo de tempo, qualquer perturbação mínima afeta no máximo uma célula vizinha e, por serem utilizadas regras de raio 1, é necessário evoluir por N passos iterativos para garantir a propagação desta perturbação por todo o reticulado. Por outro lado, devido à natureza sequencial do cálculo de pré-imagem, onde o valor da célula calculada é usado na computação de sua vizinha, a perturbação do valor de uma célula pode influenciar outra célula a uma distância maior que o raio, em um único passo de tempo.

Capítulo 5

Conclusão

O método de criptografia simétrica VHCA, apresentado e avaliado nesta tese, é um método baseado em autômato celular que se destaca pelo uso de arranjos híbridos de regras de transição elementares (binárias e de raio 1), com sensitividade a uma das direções extremas da vizinhança. Tal arranjo pode ser de regras exclusivamente com sensitividade à esquerda ou, alternativamente, de regras com sensitividade à direita. Essa característica da sensitividade comum a todas as regras do arranjo seria facilmente descoberta por meio de ataques com texto escolhido. Sendo assim, optou-se por deixar essa escolha como um parâmetro do usuário que não faz parte da chave criptográfica.

O método AES é o atual estado da arte da criptografia simétrica, sendo o padrão de mercado utilizado atualmente. Dito isso, o método proposto VHCA apresentou desempenho estatístico comparável ao AES e ao HCA, método predecessor do VHCA, considerando os experimentos realizados. Os resultados obtidos por meio das baterias de testes NIST e PractRand se mostram promissores no contexto de usos futuros do VHCA e de outras abordagens baseadas em ACs, no geral. O resultado "comparável" sob a perspectiva de robustez criptográfica pode ser visto positivamente, dado que o AES se baseia em estratégias que o caracterizam como um algoritmo sequencial, dificultando a sua paralelização. Enquanto isso, o paralelismo natural dos ACs é apontado como uma potencial vantagem para a sua utilização em métodos criptográficos. Além disso, a adaptação do AES para novos espaços de chaves e blocos, maiores do que os atualmente especificados, poderia se mostrar complexa, enquanto o VHCA pode ser diretamente parametrizado para utilizar blocos e reticulados maiores, aumentando seu espaço de chaves.

Ao comparar o método proposto a outros algoritmos criptográficos baseados em ACs, a principal desvantagem do VHCA é seu mecanismo de decifragem por meio do cálculo de pré-imagens que apresenta uma natureza inerentemente sequencial. Alguns dos métodos baseados em AC utilizam a cifragem tradicional de ACs em ambas as etapas e, por consequência, extraem o máximo de potencial do paralelismo inerente. Apesar disso, há vantagens na escolha do VHCA como, por exemplo, a simplicidade de sua implementação, dado o seu uso de regras elementares, seu uso majoritário de regras não-lineares que promove a robustez criptográfica e por manter o tamanho do reticulado inalterado durante todo o procedimento.

5.1 Principais Contribuições

Ao final do desenvolvimento deste trabalho, as contribuições não-bibliográficas foram:

- Especificação do método VHCA, com o detalhamento de suas etapas no processo de transformação criptográfica do texto em claro para um texto cifrado (cifração) e vice-versa (decifração);
- As análises de segurança criptográfica e desempenho do VHCA, embasadas nos resultados de análises formais e experimentos com bibliotecas estatísticas públicas e amplamente utilizadas;
- Implementação e disponibilização pública do código-fonte referente ao algoritmo VHCA, por meio do link: <https://github.com/evertonrlira/CACrypto>.

Com relação às hipóteses e questões de pesquisa, a descrição do novo método, bem como os resultados obtidos na seção de experimentos, indicam que o VHCA tem robustez criptográfica comparável aos algoritmos vigentes e também que seu desempenho supera o de seu predecessor, o HCA. Tais validações possibilitam novos estudos criptográficos utilizando regras elementares em arranjos híbridos; regras essas que, individualmente, não garantiriam a robustez criptográfica necessária.

5.2 Produção Bibliográfica

As contribuições bibliográficas resultantes do desenvolvimento deste trabalho foram:

- Everton R. Lira, Heverton B. de Macêdo, Danielli A. Lima, Leonardo Alt, Gina M. Oliveira. *A reversible system based on hybrid toggle radius-4 cellular automata and its application as a block cipher*. Natural Computing, 2023. Neste artigo é apresentada uma análise mais aprofundada da robustez do método HCA a partir da aplicação de testes estatísticos de criptografia disponíveis na literatura.

- ❑ Everton R. Lira, Bastien Chopard, Gina M. Oliveira. *A block cipher based on hybrid radius-1 cellular automata.* Journal of Cellular Automata, 2024. Este artigo apresenta o novo método criptográfico resultante desta tese, o VHCA, bem como as respectivas análises comparativas e de robustez do método.

5.3 Trabalhos Futuros

Conforme previamente mencionado no atual capítulo, caso a operação de cifração do VHCA fosse baseada no mecanismo de cálculo de pré-imagem, é possível que o número de etapas de tempo necessárias para cifrar pudesse ser reduzido a uma quantidade menor que N, mantendo ainda uma boa robustez criptográfica mediante análises estatísticas. Pretendemos explorar esta hipótese em trabalhos futuros. Além disso, outra investigação promissora para o futuro é explorar otimizações possíveis ao mecanismo de computação de pré-imagem atualmente usado.

Embora a escolha atual de usar a evolução tradicional de ACs para cifrar e o mecanismo de cálculo de pré-imagem para decifrar apresente um contraste com trabalhos correlatos que usam a abordagem oposta, esta é uma escolha que atualmente possibilitaria o uso de todo o potencial de paralelismo intrínseco a ACs em hardware especializado na operação de cifrar. Esta implementação do VHCA em hardware especializado ainda não foi possível devido a restrições de recursos e tempo. Entretanto, dada a computação naturalmente distribuída dos ACs, a implementação do VHCA em arquiteturas que favoreçam seu paralelismo (ex: FPGAs) se apresenta como uma evolução natural desta pesquisa. Assim, é possível investigar o impacto dessas implementações na criptografia de grandes volumes de dados, avaliando a eficiência dos algoritmos em explorar o paralelismo inter e intra-blocos.

Outro desenvolvimento planejado é o teste do VHCA em baterias de testes estatísticos utilizando arquivos de entrada maiores, bem como especificar e implementar uma versão em AC bidimensional equivalente do VHCA.

Referências

ABDALLAH, E. G.; KUANG, Y. R.; HUANG, C. Advanced encryption standard new instructions (aes-ni) analysis: Security, performance, and power consumption. In: **Proceedings of the 2020 12th International Conference on Computer and Automation Engineering**. [S.l.: s.n.], 2020. p. 167–172. (Citado na pg. 92)

AGUIAR, I. M. M. d. **Sobre a classificação dos autômatos celulares elementares finitos**. Dissertação (Mestrado) — Universidade do Minho, 2014. (Citado na pg. 43)

ALFARO, G.; SANJUÁN, M. A. Classification of cellular automata based on the hamming distance. **Chaos: An Interdisciplinary Journal of Nonlinear Science**, AIP Publishing, v. 34, n. 8, 2024. (Citado na pg. 43)

ALT, L. d. S. **Propriedades decidíveis de autômatos celulares finitos, híbridos, não-lineares, sensíveis e reversíveis**. Dissertação (Mestrado) — Universidade Federal de Uberlândia, 2013. (Citado nas pgs. 59 e 69)

ÁLVAREZ, R.; ZAMORA, A. An intermediate approach to spritz and rc4. In: SPRINGER. **Computational Intelligence in Security for Information Systems Conference**. [S.l.], 2015. p. 297–307. (Citado na pg. 91)

BARCA, D. et al. Cellular automata for simulating lava flows: a method and examples of the etnean eruptions. **Transport theory and statistical physics**, Taylor & Francis, v. 23, n. 1-3, p. 195–232, 1994. (Citado na pg. 46)

BARKER, E.; ROGINSKY, A. Nist special publication 800-131a - revision 2. transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths. Citeseer, 2019. (Citado na pg. 117)

BARKER, E. B. et al. **Sp 800-57. recommendation for key management, part 1: General (revised)**. [S.l.]: National Institute of Standards & Technology, 2007. (Citado na pg. 117)

BASSHAM, L. E. et al. **SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications**. Gaithersburg, MD, United States, 2010. (Citado na pg. 89)

_____. **A statistical test suite for random and pseudorandom number generators for cryptographic applications**. [S.l.]: National Institute of Standards & Technology, 2010. (Citado na pg. 90)

BATEMAN, G. M. The enigma cipher machine. **American Intelligence Journal**, JSTOR, v. 5, n. 2, p. 6–11, 1983. (Citado na pg. 112)

BEESLY, P. **Room 40: British naval intelligence, 1914-18**. [S.l.]: Hamish Hamilton, 1982. (Citado na pg. 112)

BELLASO, G. **La cifra del sig. Giovan Battista Belaso.** [s.n.], 1553. Disponível em: <https://books.google.com.br/books?id=zvActwAACAAJ>. (Citado na pg. 110)

BHARGAVAN, K.; LEURENT, G. On the practical (in-) security of 64-bit block ciphers: Collision attacks on http over tls and openvpn. In: **Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security**. [S.l.: s.n.], 2016. p. 456–467. (Citado na pg. 117)

BIHAM, E.; SHAMIR, A. Differential cryptanalysis of des-like cryptosystems. **Journal of CRYPTO.**, Springer, v. 4, n. 1, p. 3–72, 1991. (Citado nas pgs. 34 e 93)

BOESGAARD, M. et al. Rabbit: A new high-performance stream cipher. In: SPRINGER. **International Workshop on Fast Software Encryption**. [S.l.], 2003. p. 307–329. (Citado na pg. 87)

BORRIELLO, E.; WALKER, S. I. An information-based classification of elementary cellular automata. **Complexity**, Hindawi, v. 2017, 2017. (Citado na pg. 43)

BROWN, R. G.; EDDELBUETTEL, D.; BAUER, D. Dieharder. **Duke University Physics Department Durham, NC**, p. 27708–0305, 2006. (Citado na pg. 91)

BUNTEN-MINES, E. 48051 federal register/vol. 62, no. 177/friday, september 12, 1997/notices. 1997. (Citado na pg. 117)

CHEN, S.; GUILLEMOT, G.; GANDIN, C.-A. Three-dimensional cellular automaton-finite element modeling of solidification grain structures for arc-welding processes. **Acta materialia**, Elsevier, v. 115, p. 448–467, 2016. (Citado na pg. 46)

CIGLARIČ, T.; ŠTRUMBELJ, E. et al. An opencl library for parallel random number generators. **The Journal of Supercomputing**, Springer, v. 75, n. 7, p. 3866–3881, 2019. (Citado na pg. 91)

DAEMEN, J.; RIJMEN, V. Rijndael/aes. **Encyclopedia of Cryptography and Security**, Springer, p. 520–524, 2005. (Citado na pg. 27)

_____. **The design of Rijndael: AES-the advanced encryption standard**. [S.l.]: Springer Science & Business Media, 2013. (Citado na pg. 119)

DAS, D.; LANJEWAR, U.; SHARMA, S. The art of cryptology: From ancient number system to strange number system. **International Journal of Application or Innovation in Engineering & Management (IJAIEM)**, v. 2, n. 4, 2013. (Citado nas pgs. 109 e 110)

DAWSON, E.; GUSTAFSON, H.; PETTITT, A. N. Strict key avalanche criterion. **Australas. J Comb.**, Citeseer, v. 6, p. 147–154, 1992. (Citado na pg. 93)

DOTY-HUMPHREY, C. Practically random: C++ library of statistical tests for rngs. **URL: http://pracrand.sourceforge.net/**, 2018. (Citado na pg. 91)

FEISTEL, H. Cryptography and computer privacy. **Scientific american**, JSTOR, v. 228, n. 5, p. 15–23, 1973. (Citado na pg. 34)

FRASER, A. **Mary Queen of Scots**. [S.l.]: Hachette UK, 2010. (Citado na pg. 110)

GRABBE, J. O. **The DES algorithm illustrated**. 2010. (Citado na pg. 114)

GUTOWITZ, H. Cryptography with dynamical systems. **NATO ASI SERIES C MATHEMATICAL AND PHYSICAL SCIENCES**, Kluwer Academic Publishers, v. 396, p. 237–237, 1993. (Citado nas pgs. 25, 26, 48, 52, 64 e 67)

HOFEMEIER, G.; CHESEBROUGH, R. Introduction to intel aes-ni and intel secure key instructions. **Intel, White Paper**, v. 62, p. 6, 2012. (Citado na pg. 92)

HORTENSIUS, P. D. et al. Cellular automata-based pseudorandom number generators for built-in self-test. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, IEEE, v. 8, n. 8, p. 842–859, 1989. (Citado na pg. 47)

IEVANGELIST. **Encrypting data - .NET — learn.microsoft.com**. 2022. <https://learn.microsoft.com/en-us/dotnet/standard/security/encrypting-data>. [Accessed 03-05-2025]. (Citado na pg. 86)

JÚNIOR, T. A. d. M. **Método Criptográco Baseado em Autômatos Celulares Bidimensionais para Cifragem de Imagens**. Dissertação (Mestrado) — Universidade Federal de Uberlândia, 2010. (Citado nas pgs. 53, 66 e 67)

KARA, O. New security proofs and complexity records for advanced encryption standard. **IEEE Access**, v. 11, p. 131205–131220, 2023. (Citado na pg. 119)

KASISKI, F. W. **Die Geheimschriften und die Dechiffrir-Kunst: mit besonderer Berücksichtigung der deutschen und der französischen Sprache**. [S.l.]: ES Mittler und sohn, 1863. (Citado na pg. 112)

KOC, C.; APOHAN, A. Inversion of cellular automata iterations. **IEE Proceedings-Computers and Digital Techniques**, IET, v. 144, n. 5, p. 279–284, 1997. (Citado na pg. 47)

LAIPHRAKPAM, D. S.; KHUMANTHEM, M. S. Cryptanalysis of symmetric key image encryption using chaotic rossler system. **Optik**, Elsevier, v. 135, p. 200–209, 2017. (Citado na pg. 89)

LEPORATI, A.; MARIOT, L. 1-resiliency of bipermutive cellular automata rules. In: SPRINGER. **International Workshop on Cellular Automata and Discrete Complex Systems**. [S.l.], 2013. p. 110–123. (Citado na pg. 28)

LIMA, D. A. **Modelo criptográfico baseado em autômatos celulares tridimensionais híbridos**. Tese (Doutorado) — Universidade Federal de Uberlândia, 2012. (Citado nas pgs. 25, 53, 66 e 67)

LIRA, E. R. et al. A reversible system based on hybrid toggle radius-4 cellular automata and its application as a block cipher. **Natural Computing**, 2023. Disponível em: <https://doi.org/10.1007/s11047-023-09941-6>. (Citado nas pgs. 56, 62, 67, 70, 74 e 95)

MACÊDO, H. B. de. **Um novo método criptográfico baseado no cálculo de pré-imagens de autômatos celulares caóticos, não-homogêneos e não-aditivos**. Dissertação (Mestrado) — Universidade Federal de Uberlândia, 2007. (Citado nas pgs. 25, 26, 52, 55 e 66)

MACÊDO, H. B. de; OLIVEIRA, G. M. B. de; RIBEIRO, C. H. C. Dynamic behaviour of network cellular automata with non-chaotic standard rules. In: IEEE. **2014 Second World Conference on Complex Systems (WCCS)**. [S.l.], 2014. p. 451–456. (Citado na pg. 53)

MATSUI, M. Linear cryptanalysis method for des cipher. In: SPRINGER. **Workshop on the Theory and Application of of Cryptographic Techniques**. [S.l.], 1993. p. 386–397. (Citado na pg. 34)

MATSUMOTO, M.; NISHIMURA, T. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, ACM New York, NY, USA, v. 8, n. 1, p. 3–30, 1998. (Citado na pg. 86)

MEIER, W.; STAFFELBACH, O. Analysis of pseudo random sequences generated by cellular automata. In: SPRINGER. **Workshop on the Theory and Application of of Cryptographic Techniques**. [S.l.], 1991. p. 186–199. (Citado na pg. 47)

_____. Analysis of pseudo random sequences generated by cellular automata. In: SPRINGER. **Workshop Theory and Appl. of Crypto. Techniques**. [S.l.], 1991. p. 186–199. (Citado na pg. 94)

MILLER, F. **Telegraphic code to insure privacy and secrecy in the transmission of telegrams**. [S.l.]: CM Cornwell, 1882. (Citado na pg. 113)

MISHRA, P.; GUPTA, I.; PILLAI, N. R. Generalized avalanche test for stream cipher analysis. In: **Security Aspects in Information Technology**. [S.l.]: Springer, 2011. p. 168–180. (Citado na pg. 35)

MOORE, E. F. Machine models of self-reproduction. In: AMERICAN MATHEMATICAL SOCIETY NEW YORK. **Proceedings of symposia in applied mathematics**. [S.l.], 1962. p. 17–33. (Citado nas pgs. 41 e 53)

NANDI, S.; KAR, B. K.; CHAUDHURI, P. P. Theory and applications of cellular automata in cryptography. **IEEE Transactions on computers**, IEEE, v. 43, n. 12, p. 1346–1357, 1994. (Citado nas pgs. 64 e 67)

PACKARD, N. H.; WOLFRAM, S. Two-dimensional cellular automata. **Journal of Statistical physics**, Springer, v. 38, n. 5-6, p. 901–946, 1985. (Citado na pg. 46)

PUB, N. F. 197: Advanced encryption standard (aes). **Federal information processing standards publication**, v. 197, n. 441, p. 0311, 2001. (Citado nas pgs. 26 e 117)

RAABE, D. Introduction of a scalable three-dimensional cellular automaton with a probabilistic switching rule for the discrete mesoscale simulation of recrystallization phenomena. **Philosophical Magazine A**, Taylor & Francis, v. 79, n. 10, p. 2339–2358, 1999. (Citado na pg. 46)

RADWAN, A. G.; ABDELHALEEM, S. H.; ABD-EL-HAFIZ, S. K. Symmetric encryption algorithms using chaotic and non-chaotic generators: A review. **Journal of advanced research**, Elsevier, v. 7, n. 2, p. 193–208, 2016. (Citado na pg. 89)

RAHIMOV, H.; BABAEI, M.; FARHADI, M. Cryptographic prng based on combination of lfsr and chaotic logistic map. **Applied Mathematics**, Citeseer, v. 2, n. 12, p. 1531, 2011. (Citado na pg. 89)

RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. **Commun. ACM**, ACM, New York, NY, USA, v. 21, n. 2, p. 120–126, fev. 1978. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/359340.359342>. (Citado na pg. 32)

ROZENBERG, G.; BÄCK, T.; KOK, J. N. **Handbook of natural computing**. [S.l.]: Springer, 2012. (Citado na pg. 36)

SAMONAS, S.; COSS, D. The cia strikes back: Redefining confidentiality, integrity and availability in security. **Journal of Information System Security**, v. 10, n. 3, 2014. (Citado na pg. 31)

SEMERJIAN, H. 28907 federal register/vol. 96, no. 96/thursday, may 19, 2005/notices. 2005. (Citado na pg. 116)

SHANNON, C. E. A mathematical theory of communication. **Bell system technical journal**, Wiley Online Library, v. 27, n. 3, p. 379–423, 1948. (Citado na pg. 34)

_____. Communication theory of secrecy systems. **Bell system technical journal**, Wiley Online Library, v. 28, n. 4, p. 656–715, 1949. (Citado na pg. 113)

SIMION, E. The relevance of statistical tests in cryptography. **IEEE Security & Privacy**, IEEE, v. 13, n. 1, p. 66–70, 2015. (Citado na pg. 87)

SLEEM, L.; COUTURIER, R. Testu01 and practrand: Tools for a randomness evaluation for famous multimedia ciphers. **Multimedia Tools and Applications**, Springer, v. 79, n. 33, p. 24075–24088, 2020. (Citado na pg. 91)

STANDARD, D. E. Fips 46. **NBS (Jan. 77)**, 1977. (Citado na pg. 114)

STANDARDS, N. I. of et al. **Advanced Encryption Standard (AES)**. Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD, 2023. Disponível em: <https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=936594>. (Citado na pg. 26)

STRICKER, H.-P. Classification of elementary cellular automata based on their limit cycle lengths in z/k. **Complex Systems**, v. 32, n. 3, 2023. (Citado na pg. 43)

THORSTEINSON, P.; GANESH, A. . net security and cryptography. In: _____. [S.l.]: Prentice Hall Professional Technical Reference, 2003. p. 37. (Citado na pg. 112)

TOFFOLI, T.; MARGOLUS, N. **Cellular automata machines: a new environment for modeling**. [S.l.]: MIT press, 1987. (Citado na pg. 26)

TOMASELLO, M. **Origins of human communication**. [S.l.]: MIT press, 2010. (Citado na pg. 25)

TOMASSINI, M.; PERRENOUD, M. Stream cyphers with one-and two-dimensional cellular automata. In: SPRINGER. **International Conference on Parallel Problem Solving from Nature**. [S.l.], 2000. p. 722–731. (Citado nas pgs. 65 e 67)

_____. Cryptography with cellular automata. **Applied Soft Computing**, Elsevier, v. 1, n. 2, p. 151–160, 2001. (Citado na pg. 48)

TURAN, M. S.; DOGANAKSOY, A.; CALIK, C. Statistical analysis of synchronous stream ciphers. **SASC 2006: Stream Ciphers Revisited**, 2006. (Citado na pg. 87)

VAUDENAY, S. Security flaws induced by cbc padding—applications to ssl, ipsec, wtls... In: SPRINGER. **International Conference on the Theory and Applications of Cryptographic Techniques**. [S.l.], 2002. p. 534–545. (Citado na pg. 93)

VERNAM, G. S. Cipher printing telegraph systems: For secret wire and radio telegraphic communications. **Journal of the AIEE**, IEEE, v. 45, n. 2, p. 109–115, 1926. (Citado na pg. 113)

WEBSTER, A. F.; TAVARES, S. E. On the design of s-boxes. In: SPRINGER. **Conference on the theory and application of cryptographic techniques**. [S.l.], 1985. p. 523–534. (Citado na pg. 35)

WINTERBOTHAM, F. W. F. W. **The Ultra secret**. London, UK: Weidenfeld and Nicolson, 1974. xiii + 199 p. ISBN 0-297-76832-8. (Citado na pg. 112)

WOLFRAM, S. Cellular automata as models of complexity. **Nature**, Springer, v. 311, n. 5985, p. 419–424, 1984. (Citado na pg. 43)

_____. Cryptography with cellular automata. In: SPRINGER. **Conference on the Theory and Application of Cryptographic Techniques**. [S.l.], 1985. p. 429–432. (Citado nas pgs. 25, 26, 47, 63, 65 e 67)

_____. A new kind of science. In: _____. [S.l.]: Wolfram media Champaign, IL, 2002. v. 5, p. 53. (Citado nas pgs. 15 e 40)

_____. A new kind of science. In: _____. [S.l.]: Wolfram media Champaign, IL, 2002. v. 5, p. 23. (Citado na pg. 26)

WUENSCHE, A. Cellular automata encryption: the reverse algorithm, z-parameter and chain-rules. **Parallel Processing Letters**, World Scientific, v. 19, n. 02, p. 283–297, 2009. (Citado nas pgs. 26, 65 e 67)

ZEGHID, M. et al. A modified aes based algorithm for image encryption. **International Journal of Computer Science and Engineering**, v. 1, n. 1, p. 70–75, 2007. (Citado na pg. 27)

ZHANG, X. et al. A three-dimensional cellular automaton model for dendritic growth in multi-component alloys. **Acta Materialia**, Elsevier, v. 60, n. 5, p. 2249–2257, 2012. (Citado na pg. 46)

Anexos

ANEXO **A**

História da Criptografia

Este capítulo anexo traz uma linha do tempo com relação à evolução da criptografia, iniciando pelos primeiros registros históricos dessa prática até os métodos recentes da modernidade.

A.1 Criptografia na Antiguidade

As primeiras evidências históricas indicando tentativas de dificultar o entendimento de mensagens remontam ao Egito antigo, por volta de 1900 A.C., visto que na tumba de Khnumhotep, um nobre egípcio, foi encontrado um registro escrito onde a explícita substituição de hieróglifos por outros dificulta a interpretação da mensagem (DAS; LANJEWAR; SHARMA, 2013). Outras referências de uso da criptografia foram encontradas a partir de então, mas em Roma foi identificado o primeiro uso militar por meio da técnica "Cifra de César".

Nesse artifício criptográfico ocorre uma simples substituição de cada letra pela sua equivalente. Essa equivalência é estabelecida pela sobreposição de dois alfabetos em ordem tradicional, sendo que o primeiro representa a letra original e o segundo corresponde à letra equivalente. Considerando que cada letra do primeiro alfabeto tem uma posição X, a substituição se dá pela troca dessa letra pela equivalente na posição $X + N$ do segundo alfabeto, considerando um alfabeto circular. Tomando $N = 2$, teríamos a política de substituição proposta na Figura 27.

Na política de substituição da Cifra de César descrita na Figura 27, a letra "A" do texto original é substituída por "C" no texto resultante, "B" é substituída por "D", e assim por diante. De tal forma que o texto original:

O CRAVO BRIGOU COM A ROSA

Seria substituído por:

Q ETCXQ DTKIQW EQO C TQUC

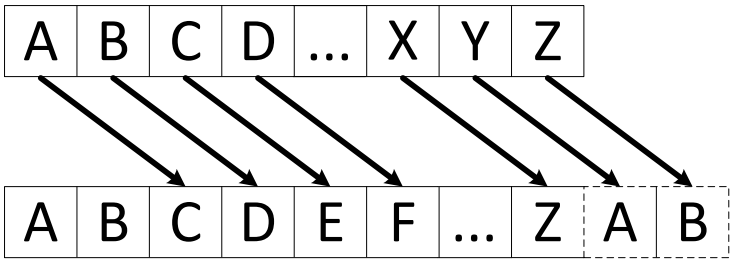


Figura 27 – Equivalência de letras na Cifra de César com deslocamento 2. [Fonte: autor]

Assim sendo, a Cifra de César, em suas diversas variações, é considerada um dos primeiros métodos criptográficos da história. Entretanto, também foram encontrados registros históricos de uso do artifício de substituição simples de letras nas antigas civilizações grega e hebraica (DAS; LANJEWAR; SHARMA, 2013).

A.2 Criptografia na Idade Média e Moderna

Na idade média também há registros da criptografia por substituição simples de letras, porém esse mecanismo passou a demonstrar sua principal vulnerabilidade, que é o fato de cada letra da mensagem original ser sempre convertida na mesma letra equivalente para a mensagem cifrada. Essa invariabilidade facilita a quebra do sigilo das mensagens, visto que na linguagem comum há letras e palavras usadas com maior frequência.

Um exemplo de situação em que essa vulnerabilidade acabou sendo explorada foi a condenação de Maria Stuart, que foi Rainha da Escócia em parte do século XVI. Segundo o relato histórico, ela tinha pretensão pelo trono da Inglaterra e conspirava assassinar Isabel I, a Rainha da Inglaterra, por meio de cartas cifradas trocadas com colaboradores. Tais cartas foram decifradas e seus conteúdos foram usados como provas no julgamento e eventual condenação de Maria (FRASER, 2010).

No século XVI, o estudioso Giovan Battista Bellaso divulgou em seu livro “La cifra del Sig” (BELLASO, 1553), um método de cifra polialfabética cuja tabela de transcrição pode ser vista na Figura 28.

Esse método é chamado “Cifra de Vigenère”, devido a uma atribuição errônea da sua autoria a Blaise de Vigenère, um criptógrafo francês do século XIX. O funcionamento da cifra dá-se pelo uso de uma chave secreta que é repetidamente concatenada a si mesma até alcançar o comprimento do texto em claro que será cifrado. Em seguida, sobrepondo o texto em claro e a chave concatenada, os pares de letras de cada posição dessas duas fontes serão usadas para descobrir o caractere da posição equivalente do texto cifrado usando a tabela da Figura 28.

Por exemplo, supondo que o texto em claro seja:

O CRAVO BRIGOU COM A ROSA

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figura 28 – Cifra de Vigenère. [Fonte: autor]

E considerando que a chave é:

SERENATA

A sobreposição entre eles e o texto cifrado resultante são apresentados na Tabela 13.

Tabela 13 – Aplicação da Cifra de Vigenère ao exemplo.

O	C	R	A	V	O	B	R	I	G	O	U	C	O	M	A	R	O	S	A
S	E	R	E	N	A	T	A	S	E	R	E	N	A	T	A	S	E	R	E
G	G	I	E	I	O	U	R	A	K	F	Y	P	O	F	A	J	S	J	E

Na primeira linha da Tabela 13 constam os caracteres do texto em claro. Na segunda linha, os caracteres da senha concatenados repetidamente até alcançar o comprimento do texto em claro. Na terceira linha, estão os caracteres do texto cifrado resultante. O caractere cifrado equivalente para cada coluna da Tabela 13 foi obtido a partir da tabela da Figura 28.

Por exemplo, considerando que a primeira letra do texto em claro é "O" e da chave concatenada é "S", a posição correspondente na tabela da Figura 28 (O × S) contém o caractere "G", o qual será o primeiro caractere do texto cifrado. Esse processo de substituição se repete para o restante do texto em claro, resultando no seguinte texto cifrado:

G GIEIO URAKFY POF A JSJE

A Cifra de Vigenère foi considerada inquebrável até o século XIX quando, separadamente, Charles Babbage (1854, em um trabalho não publicado (THORSTEINSON; GANESH, 2003)) e Friedrich Kasiski (KASISKI, 1863) descobriram vulnerabilidades baseadas na repetição de palavras que permitiam a quebra do sigilo de mensagens cifradas com esse método.

A.3 Criptografia na Idade Contemporânea

A quebra da Cifra de Vigènere foi um estopim para o surgimento de novos métodos criptográficos e, na verdade, esse fenômeno da inovação ser motivada pela necessidade têm se repetido na história da criptografia.

Na Primeira Guerra Mundial, o emprego de criptografia era necessário para a transmissão de mensagens, uma vez que os cabos submarinos de comunicação eram um meio compartilhado entre as nações e, portanto, totalmente aberto à escuta de terceiros. Nesse contexto, um dos fatos mais marcantes foi a interceptação e decifração, por parte de uma agência do Reino Unido, de um telegrama onde autoridades alemãs coordenavam um ataque aos Estados Unidos da América. Considera-se que tal interceptação, ao adiantar a entrada dos E.U.A. no conflito, mudou totalmente o rumo da guerra (BEESLY, 1982).

Ao final da Primeira Guerra Mundial, a máquina Enigma foi inventada na Alemanha pelo engenheiro Arthur Scherbius. Os modelos dessa máquina têm, no geral, a estética parecida com uma máquina de escrever, dispondo de um teclado e um painel iluminado. Apesar dessa semelhança inicial, a máquina Enigma funciona como um terminal criptográfico, dispondo de rotores que controlam as engrenagens internas responsáveis por uma avançada cifra de substituição polialfabética. A Enigma não foi utilizada para comunicações durante esse conflito, mas gradualmente foi sendo adotada para comunicações após seu lançamento comercial em 1923. Cifras geradas pela Enigma foram consideradas inquebráveis por vários anos, até que a criptografia de uma versão inicial da máquina foi quebrada por matemáticos poloneses por volta da década de 30 (BATEMAN, 1983).

Na década de 40, versões mais avançadas da Enigma foram utilizadas nas comunicações internas do exército alemão e outras potências do Eixo durante a Segunda Guerra Mundial. A confiança alemã na confidencialidade que a Enigma proporcionava era tão forte que ela foi utilizada até o fim da guerra. Inúmeros esforços por grupos de pesquisadores de outros países resultaram na quebra do segredo da Enigma, fato mantido em sigilo até a década de 70. Entre esses grupos formados na época da Segunda Guerra está o “Ultra”, grupo organizado pela agência britânica de inteligência dedicado à criptoanálise de comunicações inimigas. Os membros do Ultra, entre eles Alan Turing, são considerados responsáveis por adiantar o fim da guerra devido à quebra da criptografia das versões mais avançadas da Enigma (WINTERBOTHAM, 1974).

A.3.1 Cifra de Uso Único (OTP)

A Cifra de Uso Único, mais conhecida como OTP (*One-Time Pad*), é uma forma de criptografia simétrica descrita inicialmente em 1882 (MILLER, 1882) e redescoberta 44 anos depois (VERNAM, 1926). Essa técnica se baseia em: dado um texto em claro P e considerando a existência de uma chave criptográfica K de tamanho superior ou igual a P, temos que o texto cifrado C é formado a partir de operações módulo sobre a representação binária (bit a bit) dos caracteres em P e K. Ou seja, $C = (P + K)\%26$, conforme ilustrado na Figura 29.

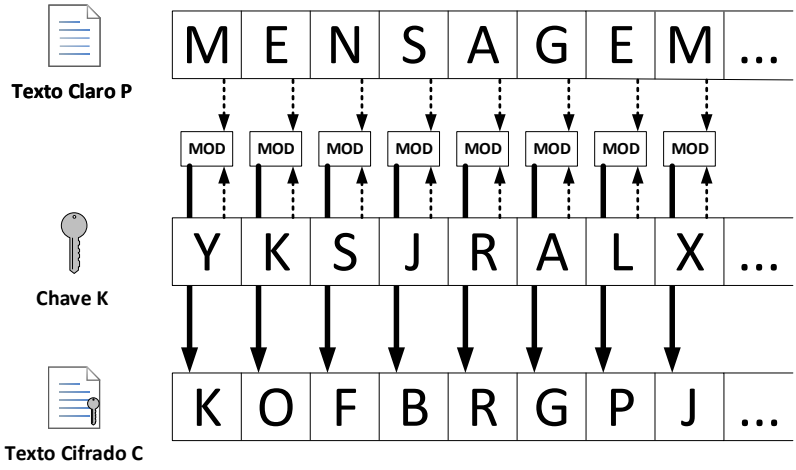


Figura 29 – Exemplo de aplicação do OTP. [Fonte: autor]

Na Figura 29, considera-se apenas o uso dos 26 caracteres alfabéticos; logo, para todo índice i, tem-se que $C[i] = (P[i] + K[i])\%26$. No caso de sequências binárias, a aplicação de um OTP é realizada por operações XOR (ou-exclusivo) entre os bits das sequências em questão ($C[i] = P[i] \oplus K[i]$), ao invés da operação módulo que é utilizada com caracteres alfabéticos. De forma correspondente, em posse do texto binário cifrado C e da chave K, o texto em claro original P pode ser recuperado usando o procedimento reverso $P[i] = C[i] \oplus K[i]$, para cada bit. Dado o baixo custo computacional das operações envolvidas (XOR) na criptografia de textos binários, o alto desempenho (rapidez) ao executar métodos OTP é evidente. Além disso, dadas certas pré-condições, pode-se afirmar que sua criptografia é inquebrável (SHANNON, 1949). Tais pré-condições são:

- ❏ A chave utilizada precisa ser verdadeiramente aleatória, e de tamanho igual ou superior à mensagem;
- ❏ Não pode haver reúso da chave sob qualquer circunstância (justificando o nome “de uso único”);

- ❏ Deve-se evitar o acesso indevido a essa chave, portanto, ela não pode ser comprometida em seu transporte e deve ser descartada corretamente.

Devido à complexidade envolvida no cumprimento de tais condições, os usos práticos do OTP são limitados, apesar de seu alto potencial criptográfico em teoria.

A.3.2 DES

O método criptográfico DES (*Data Encryption Standard*) foi desenvolvido internamente pela IBM sob o codinome "Lucifer" e, após uma série de estudos entre a empresa e o Departamento Nacional de Padrões dos EUA, conhecido como *National Bureau of Standards* (NBS), foi divulgado em 1977 como o padrão de segurança criptográfica recomendado pelo governo dos EUA (STANDARD, 1977). Após a divulgação do referido método, ele foi amplamente adotado pelas mais diversas áreas da indústria, especialmente pelo setor bancário, devido à confidencialidade das informações envolvidas (GRABBE, 2010).

A chave criptográfica do *Data Encryption Standard* (DES) tem 64 bits de comprimento, entretanto sua força criptográfica é de 56 bits, visto que os bits em posições múltiplas de 8 são utilizados para validação interna da própria chave por mecanismo de paridade.

O processo de cifragem é iniciado pela quebra do texto claro em blocos de 64 bits aos quais o algoritmo é separadamente aplicado. A Figura 30 exibe os elementos principais do fluxo da cifragem de um bloco utilizando o algoritmo DES.

Uma etapa inicial de permutação é aplicada ao bloco, modificando a ordenação dos bits para uma configuração fixa e predefinida. Em seguida, os bits resultantes são separados em duas metades de 32 bits, nomeadas "esquerda" e "direita" que são processadas separadamente em 16 etapas (*rounds*) para, em seguida, serem recombinadas. O resultado dessa recombinação, após uma última permutação, é o texto cifrado resultante para o referido bloco.

Os 56 bits efetivos da chave inicial, K, são utilizados para gerar 16 novas chaves: $K_1, K_2, \ldots, K_{16}$. Esse procedimento de geração envolve transformações por meio da rotação circular de bits, bem como permutações e descarte de bits em posições predefinidas para resultar em chaves com 48 bits.

A entrada de 32 bits recebida em cada etapa i de cifragem é expandida utilizando um mecanismo predefinido onde cada subsequência de 4 bits é expandida para 6 bits, resultando em uma sequência de 48 bits que é combinada, via operação XOR (ou-exclusivo), com a chave equivalente à etapa atual, K_i, que tem o mesmo tamanho.

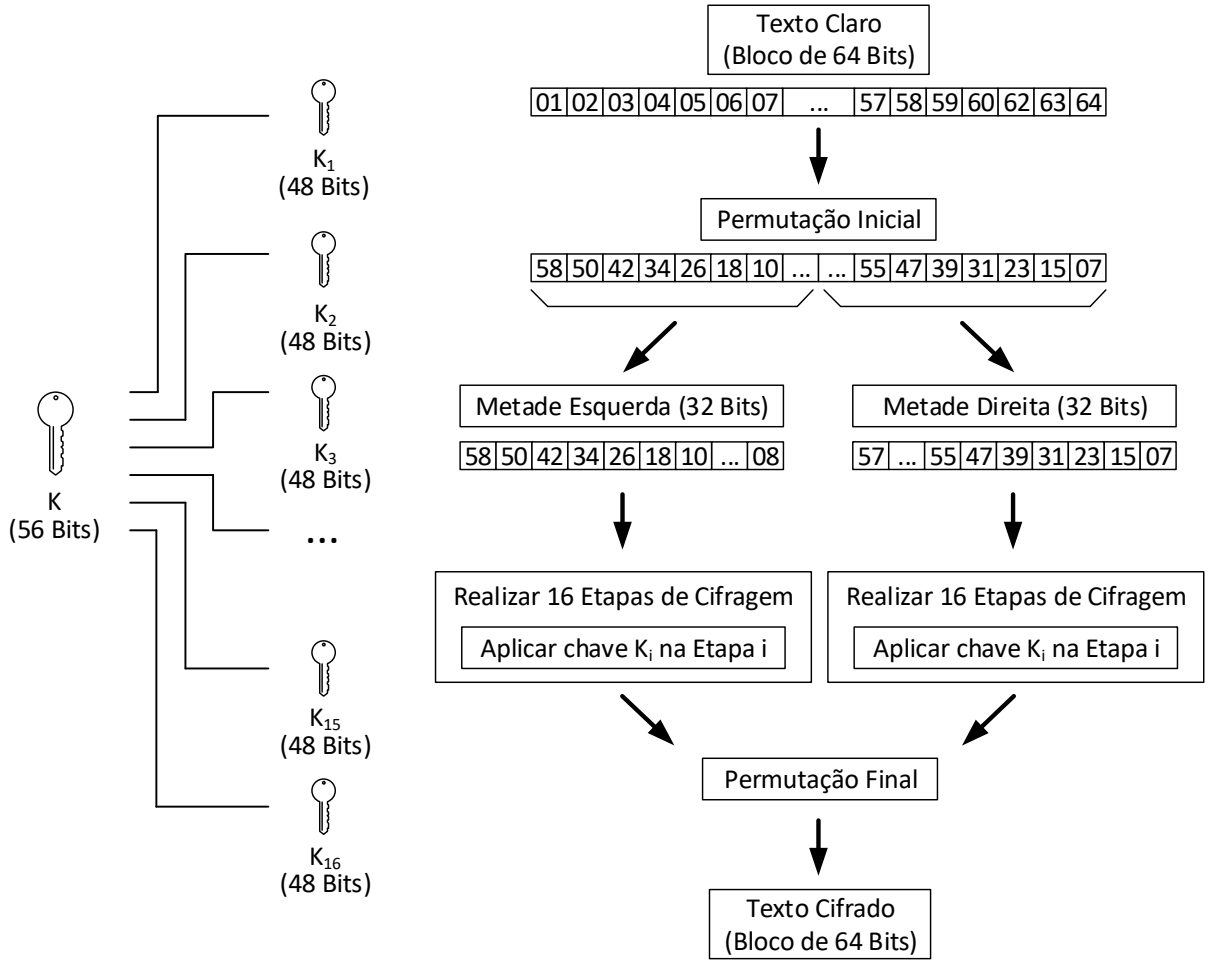


Figura 30 – Diagrama geral do algoritmo DES. [Fonte: autor]

Esse sequência de bits resultante do XOR é submetida a um procedimento de compressão onde cada subsequência de 6 bits indica por qual trecho de 4 bits esta subsequência será substituída. A essa tabela de correspondência dá-se o nome de "S-Box", e seu funcionamento é similar aos mecanismos de substituição utilizados na Idade Média como a cifra de Vigenère (Seção A.2). A Figura 31 exibe o funcionamento geral de uma S-Box.

Os dados apresentados na tabela da S-Box apresentada na Figura 31 são um exemplo didático obtido de uma etapa do próprio algoritmo DES. Em uma S-Box, os bits do valor de entrada são recombinados para obter as coordenadas (linha e coluna) do valor de saída equivalente, que é recuperado da tabela central da S-Box. No caso específico do DES tem-se uma S-Box que faz correspondência entre cada entrada de 6 bits e uma saída de 4 bits, de forma que para a sequência inicial de 48 bits essa substituição resulta em uma nova sequência de 32 bits que servirá de entrada para a etapa subsequente.

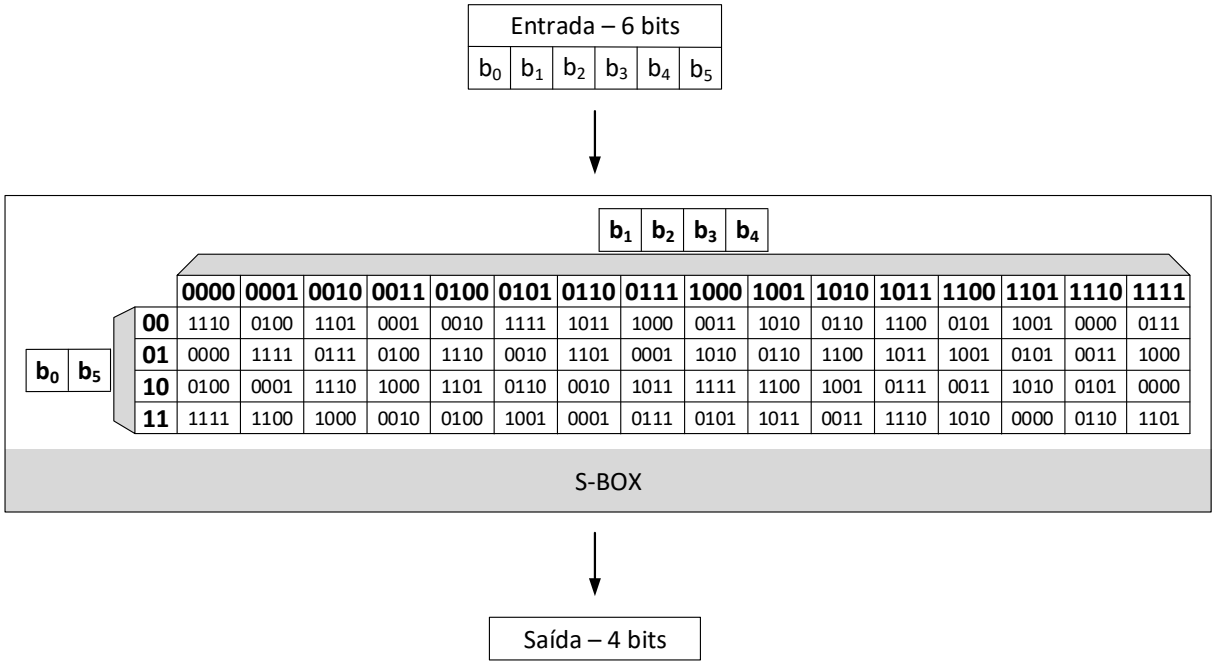


Figura 31 – Exemplo de S-Box do DES. [Fonte: autor]

A evolução tecnológica dos computadores, bem como o estudo de vulnerabilidades do método e o desenvolvimento de hardware dedicado, trouxe uma redução significativa na confiança do público com relação à segurança do DES devido ao tamanho reduzido de sua chave efetiva (56 bits), de forma que em 2005 seu uso passou a ser oficialmente desaconselhado pelo governo dos EUA (SEMERJIAN, 2005).

A.3.3 Triple DES (TDEA)

Uma das principais vulnerabilidades percebidas no DES foi o tamanho reduzido de sua chave criptográfica em relação aos rápidos incrementos na capacidade de processamento computacional. Nesse contexto, o *Triple Data Encryption Algorithm* (TDEA) surgiu como solução temporária à situação. O TDEA trata-se de uma aplicação tripla sequencial do DES, utilizando duas chaves criptográficas do DES no referido processo, conforme a Figura 32.

Na Figura 32 consta o fluxo de execução do TDEA, no qual um texto claro é cifrado com o algoritmo DES utilizando uma chave criptográfica K_1. Em seguida, o resultado dessa cifragem é modificado utilizando o procedimento de decifragem do DES com uma segunda chave K_2. Por último, o resultado dessa segunda etapa é cifrado com o DES fazendo uso da chave K_3. O resultado dessa terceira e última fase é o texto cifrado.

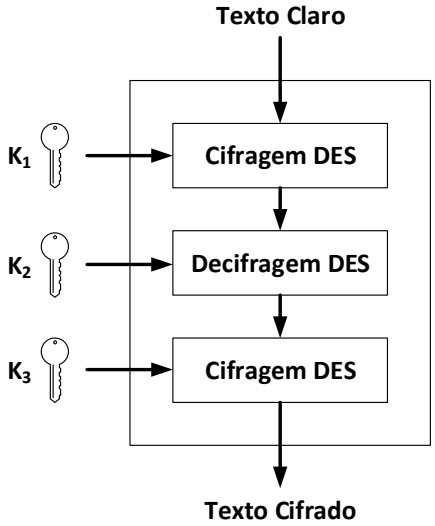


Figura 32 – Diagrama de funcionamento do TDEA. [Fonte: autor]

As chaves K_1, K_2 e K_3 são distintas na versão mais segura do método, e todas são chaves válidas do DES com seus 64 bits de tamanho e 56 bits de força criptográfica. Embora o primeiro pensamento seja de que as forças das chaves seriam somadas, resultando em uma segurança de 168 bits, na verdade a força criptográfica resultante é de 112 bits devido ao ataque meet-in-the-middle (BARKER et al., 2007).

Mesmo depois que o DES era considerado obsoleto, o TDEA ainda era considerado relativamente seguro, fato que mudou após a divulgação do ataque Sweet32 (BHARGAVAN; LEURENT, 2016). A partir de 2018, o algoritmo deixou de ter seu uso recomendado pelo NIST (BARKER; ROGINSKY, 2019).

A.3.4 AES

Em 1997, o Instituto Nacional de Padrões e Tecnologia dos EUA, conhecido como NIST (*National Institute of Standards and Technology*), abriu uma chamada pública solicitando propostas para novos algoritmos criptográficos que cumprissem altos requisitos de desempenho e segurança (BUNTEN-MINES, 1997). Após duas rodadas de seleção e avaliação das propostas recebidas, em 2001, o algoritmo Rijndael foi anunciado como o escolhido (PUB, 2001). A partir desse momento, o algoritmo Rijndael, cujo nome é derivado de seus criadores Joan Daemen e Vincent Rijmen, passou a ser conhecido como AES (*Advanced Encryption Standard*).

O primeiro passo do método AES no procedimento de cifração é transpor o texto em claro para uma matriz de estados. A chave criptográfica original dá origem a 10, 12 ou 14 novas matrizes de chave, dependendo da escolha entre AES-128, AES-192 ou AES-256.

Conforme ilustrado na Figura 33, cada etapa (*round*) da cifração envolve: a permutação de colunas da matriz de estados; a rotação (*shift*) de algumas das linhas da referida matriz; e a substituição de bytes utilizando uma S-Box (entrada: 8 bits - saída: 8 bits), conceito que também foi aplicado no algoritmo DES com S-Boxes de tamanho distinto (entrada: 6 bits - saída: 4 bits). Após tais procedimentos serem aplicados à matriz de estados atual, uma operação de XOR é feita entre ela e a matriz da chave que foi designada ao *round* atual, resultando na nova matriz de estados que será utilizada na etapa subsequente.

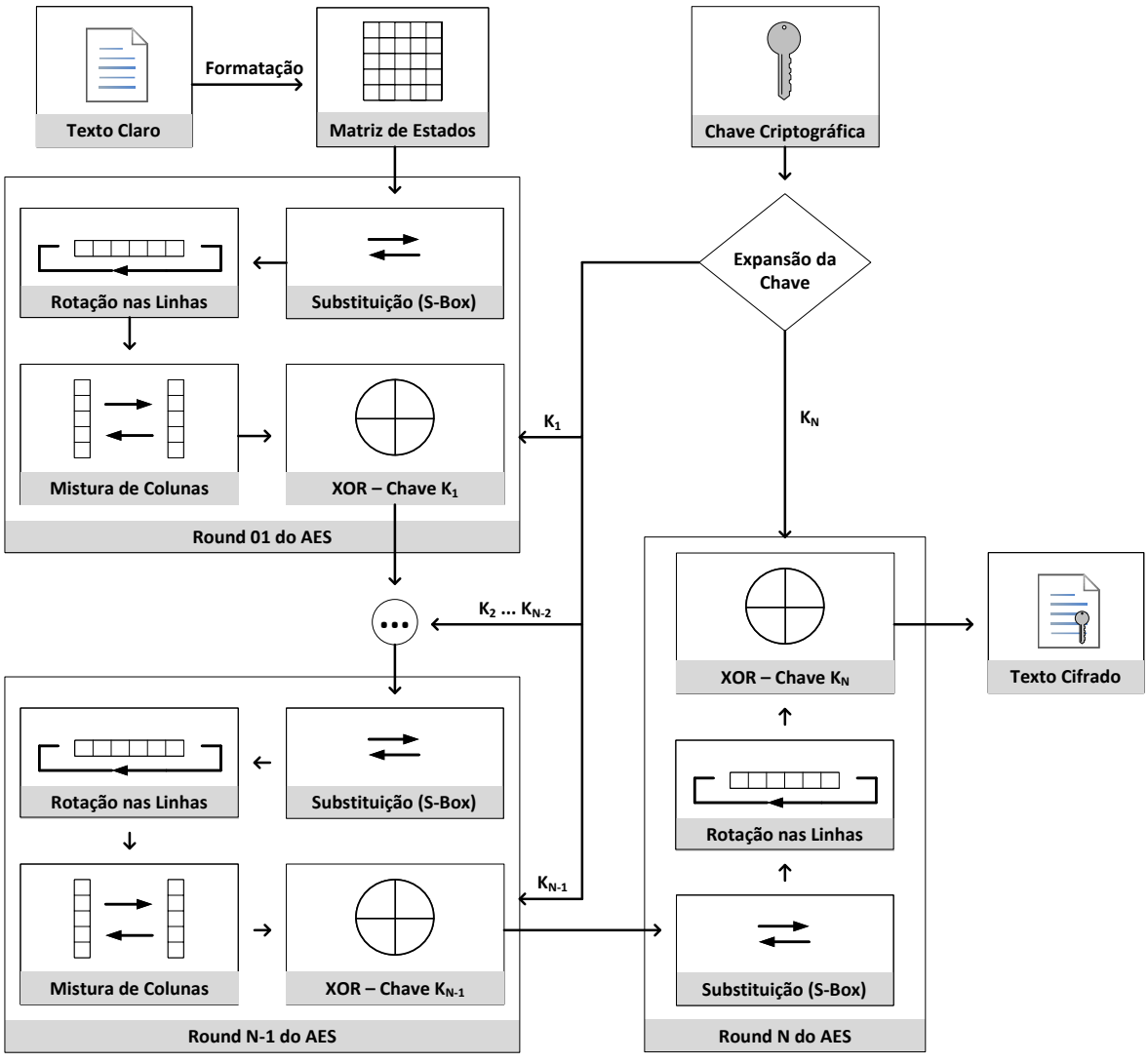


Figura 33 – Diagrama de funcionamento do AES. [Fonte: autor]

Apesar da ideia por trás do algoritmo ser relativamente simples quando comparada a outros da área (o que era um objetivo dos criadores (DAEMEN; RIJMEN, 2013)), isso não indica que a sua segurança criptográfica seja baixa. De fato, até o momento da escrita desta tese, as tentativas públicas mais recentes de redução da complexidade do AES não tiveram êxito a ponto de permitir a quebra de chaves criptográficas em tempo razoável (KARA, 2023).