Um novo método criptográfico baseado no cálculo de pré-imagens de autômatos celulares caóticos, não-homogêneos e não-aditivos

Heverton Barros de Macêdo

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Uberlândia

Programa de Pós-graduação em Ciência da Computação
Faculdade de Computação
Universidade Federal de Uberlândia
Minas Gerais – Brasil

Setembro de 2007

Heverton Barros de Macêdo

UM NOVO MÉTODO CRIPTOGRÁFICO BASEADO NO CÁLCULO DE PRÉ-IMAGENS DE AUTÔMATOS CELULARES CAÓTICOS, NÃO-HOMOGÊNEOS E NÃO-ADITIVOS

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Uberlândia, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Inteligência Artificial.

Orientadora: Prof. Dra. Gina Maira Barbosa de Oliveira.

UBERLÂNDIA – MINAS GERAIS – BRASIL
Universidade Federal de Uberlândia – UFU
Setembro de 2007

Dados Internacionais de Catalogação na Publicação (CIP)

M134n Macêdo, Heverton Barros de, 1982-
Um novo método criptográfico baseado no cálculo de pré-imagens de autômatos celulares caóticos, não-homogêneos e não-aditivos / Heverton Barros de Macêdo. - 2007.
184 f. : il.

Orientadora: Gina Maira Barbosa de Oliveira.
Dissertação (mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Ciência da Computação.
Inclui bibliografia.

1. Inteligência artificial - Teses. 2. Criptografia - Teses. I. Oliveira, Gina Maira Barbosa de. II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU: 681.3:007.52

Elaborada pelo Sistema de Bibliotecas da UFU / Setor de Catalogação e Classificação

Heverton Barros de Macêdo

UM NOVO MÉTODO CRIPTOGRÁFICO BASEADO NO CÁLCULO DE PRÉ-IMAGENS DE AUTÔMATOS CELULARES CAÓTICOS, NÃO-HOMOGÊNEOS E NÃO-ADITIVOS

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação da Universidade Federal de Uberlândia, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Inteligência Artificial.

Banca Examinadora:

Uberlândia, 12 de setembro de 2007.

_____

Prof. Dra. Gina Maira Barbosa de Oliveira – UFU (orientadora)

_____

Prof. Dr. Zhao Liang – ICMC/USP – São Carlos

_____

Prof. Dr. João Nunes de Souza - UFU

Um novo método criptográfico baseado no cálculo de pré-imagens de autômatos celulares caóticos, não-homogêneos e não-aditivos

Heverton Barros de Macêdo

Resumo

O presente trabalho investiga um novo método criptográfico baseado em autômatos celulares (ACs). Neste método o processo de cifragem é realizado através do cálculo de pré-imagens enquanto a decifragem é realizada através da evolução temporal dos ACs. Para que o cálculo de pré-imagem possa ser utilizado em um sistema criptográfico, é necessário que todos os reticulados possíveis de um AC possuam pelo menos uma pré-imagem. Em um método anterior, proposto por Gutowitz, essa garantia de existência de pré-imagem foi conseguida graças à propriedade de sensitividade das regras empregadas como chaves e à utilização de bits adicionais. Nessa dissertação, um novo método é proposto onde dois tipos distintos de regras fazem com que não seja necessário acrescentar bits adicionais, tornando o texto cifrado e texto original do mesmo tamanho. Uma das regras utilizadas também possui a propriedade de sensitividade e é responsável pelo comportamento dinâmico médio do AC. A outra regra é responsável por garantir que sempre exista uma pré-imagem, sem a necessidade dos bits adicionais. Esse método também difere de outros modelos criptográficos publicados anteriormente que utilizam ACs com regras aditivas. Uma das características do modelo aqui proposto é a sua resistência contra um tipo de ataque conhecido como criptoanálise diferencial, além da possibilidade de implementação eficiente em hardware, usufruindo do paralelismo do modelo.

Palavras-chave: Autômato celular, criptografia, cálculo de pré-imagem, sistema dinâmico.

A new cryptography method based on the pre-image calculus of chaotic, non-homogeneous and non-additive cellular automata

Heverton Barros de Macêdo

Abstract

A new cryptographic method based on cellular automata (CA) has been investigated. In this method, the ciphering process is performed by preimages computation while deciphering is performed by CA temporal evolution. The preimages calculus is able to be used in a cryptographic system only if any arbitrary CA lattice has at least one preimage. In a previous method proposed by Gutowitz, this guarantee of pre-image was obtained due to the toggle property of the rules used as keys and some additional bits used in the calculus. Here a new method is proposed in which two distinct types of toggle rules are used without the need of additional bits. As a consequence, the cipher text and the plain text have the same size. One of the rules has the toggle property and it is responsible for CA dynamical behavior. The other rule is responsible to guarantee preimage existence, without need of additional bits. This method also differs from other cryptographic models based on additive rules previously published. The model is resistant against an attack known as differential cryptanalysis. Besides, it is efficient considering hardware implementation due to the intrinsic parallelism of the model.

Keywords: Cellular automata, cryptography, preimage calculus, dynamical system.

Agradecimentos

Aos meus pais Enio e Marlene, ao meu irmão Enio Júnior e à minha irmã Lilian, pelo apóio incondicional em todos os momentos da minha vida.

À minha orientadora de dissertação, prof. Dra. Gina Maira Barbosa de Oliveira, pela oportunidade que me concedeu em desfrutar de sua companhia e compartilhar de seus conhecimentos ao longo deste trabalho, além da dedicação e confiança por ela depositada.

À minha namorada Isabelle, pela paciência, compreensão e apoio no decorrer deste trabalho.

Aos meus amigos do mestrado, pela troca de conhecimentos e os momentos de descontração. Em especial ao Everton, Rogério, Stéfano, Victor, Ítalo, Vinícius, Welter, Klérisson, Felipe, Robson, Tauller, Laurence, Lucas, Marcelo (Gremo) e aos monitores Rodrigo (Digão) e Daniel (Coelho).

Aos meus professores da Universidade Federal de Uberlândia, que contribuíram para o meu aprendizado.

Ao prof. Dr. Carlos Roberto Lopes, pela dedicação como coordenador do programa de pós-graduação, e em particular, por não medir esforços em benefício dos alunos do programa.

À CAPES, pelo apóio financeiro.

E a todas as pessoas que, de forma direta ou indireta, contribuíram para a conclusão deste trabalho.

Sumário

Lista de Abreviaturas

Abreviatura	Detalhes
DES	*Data Encryption Standard*
AES	*Advanced Encryption Standard*
AC	Autômato Celular
CA	*Cellular Automata*
SPN	*Substitution Permutation Network*
NIST	*National Institute of Standards and Technology*
RSA	Rivest, Shamir e Adleman.
GF	Galois *Field*
AG	Algoritmo Genético
CAC	*Cellular Automata Cryptosystem*

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Tornar a comunicação entre entidades remotas segura é a principal função de sistemas que fazem a troca de informações sigilosas, tais como, sistemas militares, sistemas bancários ou até mesmo no envio de um e-mail cujo conteúdo o remetente deseja que seja acessado somente por um ou um grupo de destinatários específico.

Existem vários modelos de sistemas com o propósito de segurança no tráfego e armazenamento de informações. Porém, por mais que um sistema possa parecer seguro, é comum que, após algum tempo, seja descoberta alguma fraqueza comprometendo a segurança das informações. Contudo, o desafio de proteger dados sigilosos ao serem armazenados ou enquanto trafegam até o destino correspondem às maiores motivações para o estudo de técnicas de criptografia.

Sistemas dinâmicos têm despertado o interesse de diversos pesquisadores por sua possibilidade de utilização em métodos criptográficos. A idéia é aproveitar a aleatoriedade inerente ao comportamento caótico de um sistema dinâmico, com o intuito de criar um método criptográfico capaz de produzir uma seqüência de bits o mais aleatória possível, dada uma mensagem ou texto qualquer, juntamente com uma chave criptográfica.

Autômatos celulares (ACs) são sistemas dinâmicos que possuem variáveis discretas na representação do tempo, do espaço e dos estados. Além disso, ACs são intrinsecamente paralelos e podem ser facilmente implementados em hardware, proporcionando características que os tornam objeto de investigação na criação de um sistema criptográfico.

Diversos pesquisadores, tais como Wolfram (1986), Gutowiz (1995), Ganguly, Das, Sikdar e Chaudhuri (2000), Sen, Shaw, Chowdhuri (2002), Seredynski, Bouvry e Zomaya (2003), Oliveira, Coelho e Monteiro (2004), Benkiniouar e Benmohamed (2004), dentre outros, investigaram o uso de ACs aplicados a criptografia. Os métodos criptográficos propostos por estes pesquisadores serão apresentados no decorrer desta dissertação, mostrando suas principais características, além das vantagens e desvantagens de cada um.

1.1 Objetivo

Esta dissertação tem como principal investigação a aplicação de autômatos celulares como método de criptografia, tendo como objetivo geral criar um novo sistema criptográfico baseado no cálculo de pré-imagens de autômatos celulares. É desejado que, em tal modelo, o texto cifrado seja do mesmo tamanho que o texto original, além de prover segurança contra alguns ataques de criptoanálise conhecidos como, por exemplo, a criptoanálise diferencial.

A criação de um sistema criptográfico envolve diversos fatores, como: o tamanho do espaço de chaves, o desempenho do método ao cifrar e decifrar, a quantidade de recursos necessários no processo de cifragem e decifragem, a segurança contra algum método de criptoanálise, dentre outros. Estes fatores serão discutidos no decorrer dos capítulos e analisados no método aqui proposto.

1.2 Organização do trabalho

A dissertação foi dividida em oito capítulos com o objetivo de transmitir as informações necessárias para a contextualização, definição e entendimento do sistema criptográfico proposto, além de apresentar resultados de alguns testes efetuados.

No Capítulo 2, são apresentados os principais conceitos envolvendo a criptografia, além de uma revisão histórica dos métodos mais conhecidos. A definição de um sistema criptográfico, assim como a criptoanálise e suas principais técnicas, são também apresentadas neste capítulo.

Uma visão geral sobre autômatos celulares, contendo as definições e conceitos necessários para a execução do cálculo de pré-imagens, é encontrada no Capítulo 3.

O Capítulo 4 revisa alguns dos principais sistemas criptográficos empregando autômatos celulares. Dentre estes modelos, merece destaque o sistema proposto por Gutowitz (1995), no qual o método aqui descrito foi inspirado.

No Capítulo 5, são apresentadas as investigações que delinearam a criação do sistema criptográfico aqui proposto, além de prover os conceitos básicos, tais como, geração de regras utilizadas como chave criptográfica e o esboço do sistema criptográfico de forma geral.

O Capítulo 6 é responsável por mostrar os resultados de alguns experimentos para avaliar a robustez do método e identificar alguma característica que eventualmente possa ou deva ser evitada.

A descrição do método como um sistema criptográfico, assim como os resultados de alguns experimentos após a definição dos padrões empregados no sistema criptográfico, são apresentandos no Capítulo 7.

No Capítulo 8, são relatadas as conclusões e também algumas investigações não contempladas neste trabalho que são sugeridas como continuidade dessa dissertação.

Capítulo 2

Criptografia

Criptografia (do Grego *kriptos*, "escondido", e *grapho*, "escrever") é geralmente entendida como sendo o estudo dos princípios e das técnicas pelas quais a informação pode ser transformada da sua forma original para outra ilegível. Na criptografia moderna, tal transformação utiliza uma chave criptográfica, assim, somente quem possuir a correta chave criptográfica usada na transformação inversa, pode compreender a informação com facilidade [Wikipédia 2007a].

É comum que sejam encontrados na literatura os termos criptoanálise e criptologia quando se estuda criptografia. Criptoanálise (*kriptos* = escondido, oculto; *análysis* = decomposição) é a arte ou ciência de determinar a chave ou decifrar mensagens sem conhecer a chave. Uma tentativa de criptoanálise é chamada de ataque. Criptologia (*kriptos* = escondido, oculto; *logo* = estudo, ciência) é a ciência que reúne a criptografia e a criptoanálise.

Neste capítulo, será apresentada uma pequena revisão sobre a história da criptografia e suas principais técnicas.

2.1 Terminologia

Nesta seção, apresentamos os principais termos utilizados em criptografia.

Mensagem: informação, textual ou não, a ser enviada de uma pessoa a outra através de um meio de comunicação não seguro.

Texto claro: mensagem na forma original (legível). A terminologia usada no inglês é "*plain text*", o que leva a uma tradução usual em português para "texto plano". Como existe um esforço da comunidade brasileira de criptografia em coibir a utilização desse termo, por considerá-lo uma tradução errônea, utilizaremos nessa dissertação o termo "texto claro", embora "texto plano" seja o mais usual.

Texto cifrado: mensagem que passou por uma processo de cifragem, tornando-se incompreensível para pessoas não autorizadas.

Cifragem: processo aplicado ao texto claro, que assume a nova forma de texto

cifrado.

Decifragem: processo aplicado ao texto cifrado, que resulta no texto claro.

Cifrar: ato da cifragem.

Decifrar: ato da decifragem.

Chave: cadeia de bits, utilizada no processo de cifragem e decifragem.

Criptoanálise: arte ou ciência de tentar encontrar fraquezas em sistemas criptográficos com o intuito de descobrir a chave criptográfica ou o texto claro.

Criptoanalista: aquele que pratica a criptoanálise.

2.2 História da criptografia

Para Pfaffenberger (1998), a criptografia teve origem na Roma antiga. Porém, há indícios que nos conflitos greco-persas já havia ocultamento de mensagens através de escritas na cabeça raspada de um servo. Após algum período, onde as mensagens já não mais poderiam ser vistas em consequência do crescimento do cabelo, o servo era enviado para outra região, o que pode ser considerada uma forma de criptografia [Carvalho, 2000]. Há quem afirme que a criptografia é quase tão antiga quanto a própria escrita [Sancese, 1998]. Um exemplo é um texto encontrado no túmulo do faraó Knumotete II, por volta de 1900 a.C., que parece ter sido modificado propositadamente.

Existem diversos modelos de cifragem relatados no decorrer da história, os quais utilizam basicamente susbtiuição ou transposição de letras.

2.2.1 Métodos de substituição

Nos métodos de substituição, letras do texto claro (texto original) são substituídas por outras letras, números ou símbolos.

Diversos casos de cifras com simples substituição aparecem na história da criptografia, sendo que um desses casos em especial é conhecido como cifra de deslocamento (*shift cipher*). Neste modelo de substituição simples, a chave do sistema é o número de deslocamentos que será usado no alfabeto. Por exemplo: se considerarmos o deslocamento de uma posição no alfabeto, a letra A será substituída por B, B por C, e assim sucessivamente, até a substituição de Z por A, como mostra a Tabela 1.

Neste caso, o texto claro “ATAQUE ADIADO” teria como texto cifrado correspondente “BUBRVF BEJBEP”. Para decifrar, o receptor deve realizar o deslocamento do alfabeto, com a mesma quantidade de posições, na direção contrária ao utilizado no processo de cifragem.

Tabela 1 - Mapeamento de cifragem para *shift cipher* – Deslocamento de uma posição.

A	B	C	D	E	F	G	H	I	J	K	L	M
B	C	D	E	F	G	H	I	J	K	L	M	N

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
O	P	Q	R	S	T	U	V	W	X	Y	Z	A

Um caso especial de *shift cipher* é conhecido como *César cipher*, criada pelo imperador romano Júlio César para comunicar planos de batalha, onde a chave nesse caso é o deslocamento de três posições no alfabeto [Mao 2003].

Se considerarmos apenas deslocamentos no alfabeto, o número de chaves neste modelo é igual a 26. Sendo assim, um criptoanalista pode testar todas as possíveis chaves, já que existem poucas, e descobrir o texto claro. Uma generalização desta técnica pode aumentar o número de chaves consideravelmente, como mostra o exemplo a seguir. Considere o alfabeto ocidental atual compondo o conjunto de mensagens e cifras possíveis. A chave neste modelo corresponde à disposição das letras na tabela de cifragem. A Tabela 2 mostra um mapeamento possível para este método.

Tabela 2 - Substituição monoalfabética: mapeamento usado na cifragem.

A	B	C	D	E	F	G	H	I	J	K	L	M
H	P	U	G	C	N	Z	F	E	T	M	W	D

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
I	Q	J	B	K	O	A	L	S	Y	R	X	V

O texto claro “ATAQUE ADIADO” após sofrer as substituições direcionadas pela Tabela 2 produz o texto cifrado: “HAHBLC HGEHGQ”. Para decifrar, a Tabela 3, que é apenas o inverso da tabela de cifragem, deve ser consultada ao executar as substituições no texto cifrado, retornando ao texto original.

Tabela 3 - Substituição monoalfabética: mapeamento usado na decifragem.

A	B	C	D	E	F	G	H	I	J	K	L	M
T	Q	E	M	I	H	D	A	N	P	R	U	K

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
F	S	B	O	X	V	J	C	Z	L	Y	W	G

O exemplo citado acima é uma cifra de substituição conhecida como monoalfabética, onde cada letra do texto claro corresponde a uma letra do texto cifrado. Neste esquema é possível ter 26! chaves diferentes. Porém, apesar do grande número de chaves possíveis, este método é vulnerável a um ataque conhecido como análise de freqüência. O ataque explora o fato da linguagem natural possuir um elevado volume de redundância [Mao 2003]. Por exemplo, na língua portuguesa, sabe-se que a freqüência média da letra "*A*" ocorrer em um texto é bem maior do que a ocorrência da letra "*B*", como mostra a Figura 1. Assim, o ataque é feito ao se fazer uma análise de freqüência no texto cifrado e em seguida associar letras do texto claro às letras do texto cifrado. Um texto cifrado com uma alta freqüência de ocorrência da letra "*X*", por exemplo, sugere que "*X*" corresponda à letra "*A*", de acordo com o histograma da freqüência relativa das letras na língua portuguesa (Figura 1).

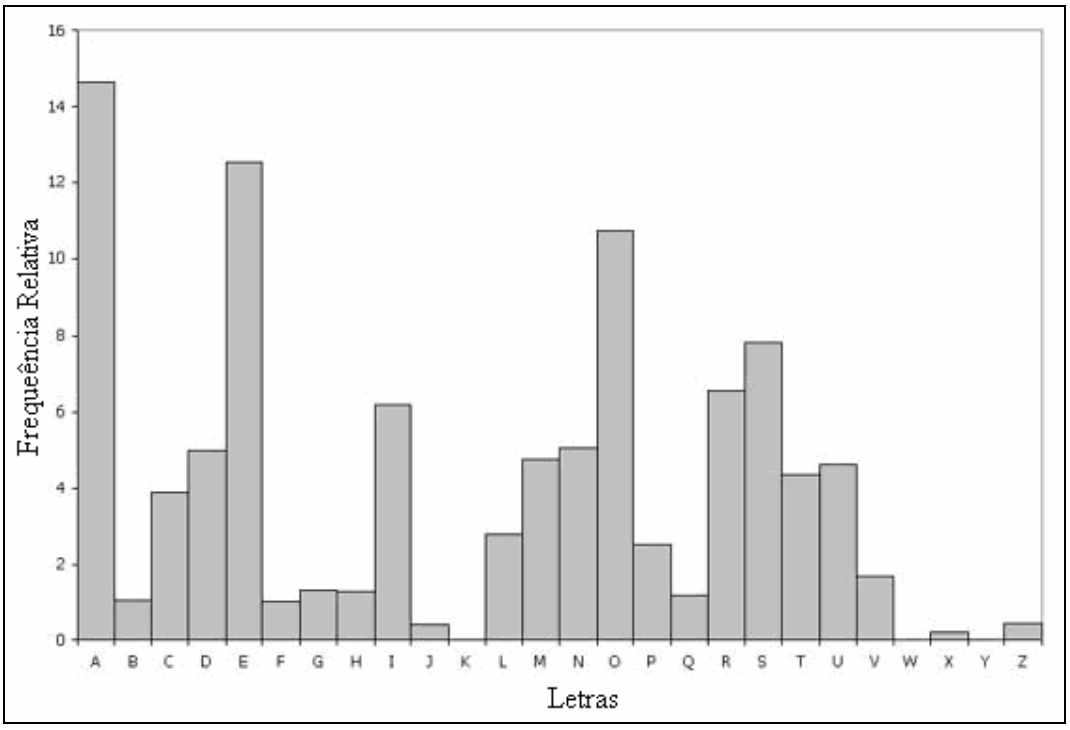


Figura 1 - Freqüência do uso das letras na língua portuguesa [Wikipédia 2007b].

Outro tipo de cifra, conhecida como poli-alfabética, também usa substituição. Porém,

neste modelo, um elemento no espaço de texto claro pode ser substituído por diferentes elementos no espaço de texto cifrado, como é o caso da cifra de Vigenère.

No modelo de Vigenère é realizada uma operação de adição módulo 26 entre cada caractere do texto claro com um respectivo caractere da chave. Neste modelo, geralmente é utilizada uma chave de tamanho menor do que o texto a ser cifrado. Assim, a chave é repetida até o final do texto claro. Para realizar a adição entre os caracteres da chave e do texto claro é feita uma conversão dos mesmos em números. Por exemplo: O texto claro "ATAQUE ADIADO" cifrado com a chave "XYZ" pode ser obtido com a conversão de A=0, B=1,...,Z=25 (veja Tabela 4), produzindo o texto claro "0-19-0-16-20-4 0-3-8-0-3-14" e a chave "23-24-25", que são aplicados a operação de adição modulo 26, como mostra a seqüência abaixo:

$$\begin{array}{cccccc} +0 & +19 & +0 & +16 & +20 & +4 \\ \underline{23} & \underline{24} & \underline{25} & \underline{23} & \underline{24} & \underline{25} \\ 23 & 17 & 25 & 13 & 18 & 3 \end{array} \qquad \begin{array}{cccccc} +0 & +3 & +8 & +0 & +3 & +14 \\ \underline{23} & \underline{24} & \underline{25} & \underline{23} & \underline{24} & \underline{25} \\ 23 & 1 & 7 & 23 & 1 & 13 \end{array}$$

Após a adição modulo 26, o resultado é convertido em letras, usando novamente a Tabela 4, resultando no texto cifrado: "XRZNSD XBHXBN". Para decifrar, o texto cifrado deve sofrer a operação de subtração módulo 26 com a mesma chave empregada no processo de cifragem, como apresentado abaixo:

$$\begin{array}{cccccc} -23 & -17 & -25 & -13 & -18 & -3 \\ \underline{23} & \underline{24} & \underline{25} & \underline{23} & \underline{24} & \underline{25} \\ 0 & 19 & 0 & 16 & 20 & 4 \end{array} \qquad \begin{array}{cccccc} -23 & -1 & -7 & -23 & -1 & -13 \\ \underline{23} & \underline{24} & \underline{25} & \underline{23} & \underline{24} & \underline{25} \\ 0 & 3 & 8 & 0 & 3 & 14 \end{array}$$

Depois de concluída a subtração módulo 26 entre o texto cifrado e a chave, o resultado deve ser novamente convertido em caracteres, resultando no texto original.

Tabela 4 - Conversão dos caracteres do alfabeto em números.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Embora, as ocorrências de caracteres repetidos no texto cifrado não sejam as mesmas do texto claro, alguma estrutura do texto pode ser utilizada na tentativa de descobrir a chave. Uma vez que a chave sempre se repete para completar todo o texto claro, substituições monoalfabéticas do tamanho da chave são executadas. Dessa forma, o sistema pode deixar

aparecer estruturas no texto cifrado que revele o tamanho da chave, e assim, de posse do tamanho da chave e um volume de texto cifrado, o criptoanalista será capaz de deduzir a chave do sistema sem muita dificuldade [Stallings 2003].

Em 1918, Vernam construiu um sistema criptográfico em que o texto claro, a chave criptográfica e o texto cifrado são representados por uma cadeia de bits. A cada bit do texto claro é feita uma operação XOR (adição modulo 2) com um bit correspondente da chave secreta, resultando no texto cifrado. Dessa forma, o método pode ser representado pela equação (1).

$$c_i = m_i \text{ XOR } k_i \qquad (1)$$

Para que este método de cifra se torne seguro, a chave do sistema criptográfico é usada apenas uma vez, e por isso, este modelo também é conhecido como *one-time-key Vernam cipher*. Caso a chave nesse sistema seja usada apenas uma vez e gerada de forma aleatória, tal que a chave e o texto claro sejam do mesmo tamanho, esta cifra recebe o nome de *on-time pad cipher,* e é conhecida por ser incondicionalmente segura [Mao 2003].

Em comparação com o modelo de Vigenère (adição módulo 26), o modelo de Vernam (adição módulo 2) pode ser facilmente implementado em circuitos eletrônicos. Em função da segurança encontrada no texto cifrado (caso uma chave apropriada tenha sido escolhida) e a facilidade de implementação, a operação XOR bit a bit é muito usada nos sistemas criptográficos atuais.

2.2.2 Método de transposição

Outro tipo de cifra clássica é o método conhecido como transposição, onde é feita uma permutação entre as letras do texto claro. Por exemplo, o texto claro "RETORNE HOJE PARA BASE UM" é estruturado em forma de matriz, onde o número de colunas é dado pelo tamanho da chave e são utilizadas tantas linhas quanto forem necessárias para representar o texto claro. Seja a chave "4312567", o texto cifrado é dado pela leitura dos caracteres nas colunas, de acordo com a ordenação da chave.

Chave:	4	3	1	2	5	6	7
Texto claro:	R	E	T	O	R	N	E
		H	O	J	E		P
	A	R	A		B	A	S
	E		U	M	X	Y	Z

Neste caso o texto cifrado será: “TOAUOJ MEHR R AEREBXN AYESPZ”.

De acordo com Stallings (2003), cifras realizadas somente com transposição são facilmente identificadas porque o texto claro e o texto cifrado possuem a mesma freqüência de letras. Uma criptoanálise para o exemplo de transposição citado acima consistiria em ajustar o texto cifrado em uma matriz e tentar reorganizar o texto claro fazendo uso de pares de letras que ocorrem com freqüência, como por exemplo, dígrafos, e então analisar os possíveis anagramas. Estas cifras podem ser mais significantemente seguras se executadas por mais de um estágio de transposição. Assim o texto “TOAUOJ MEHR R AEREBXN AYESPZ” seria cifrado mais de uma vez usando a mesma técnica, de preferência com chaves diferentes, o que resultaria em uma permutação mais complexa.

As técnicas de substituição e transposição aqui apresentadas são conhecidas como cifras clássicas. Exceto a *on-time pad cipher*, todas elas possuem vulnerabilidades se usadas sozinhas, mas quando usadas em conjunto, são ferramentas poderosas na construção de um sistema criptográfico como, por exemplo, o sistema DES (Data Encryption Standard) [NIST 1977] e o sistema AES (Advanced Encryption Standard) [NIST 2001].

2.3 Sistema criptográfico

Um sistema criptográfico consiste basicamente de:

- um espaço de mensagens em texto claro M: conjunto de todas as cadeias de caracteres escrita sob algum alfabeto;
- um espaço de mensagens cifradas C: conjunto de todas possíveis mensagens cifradas;
- um espaço de chaves K e $K´$: conjunto das possíveis chaves Ke usadas para cifrar e chaves Kd para decifrar, respectivamente;
- um eficiente algoritmo para cifrar: E: $M \times K \rightarrow C$;
- e um eficiente algoritmo para decifrar: D: $C \times K´ \rightarrow M$.

A cifragem é definida por: $c = E_{ke}(m)$, e a decifragem e definida por: $m = D_{kd}(c)$.

É necessário que $c \in C$, $m \in M$, $Ke \in K$, e existe um $Kd \in K´$ para qualquer $E_{ke}(m)$. A Figura 2 mostra um esquema onde uma mensagem é cifrada e enviada por um canal de comunicação não seguro, até chegar ao destino e passar pelo processo de decifragem.

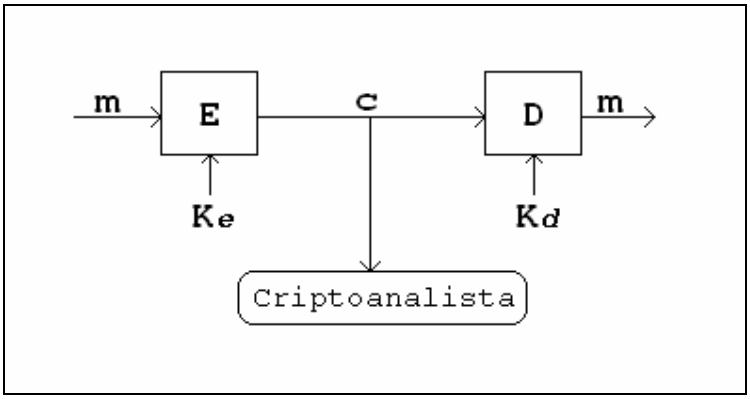


Figura 2 - Modelo de um sistema criptográfico (cifragem e decifragem).

Se *Ke* = *Kd*, o sistema criptográfico é classificado como simétrico, chave simples ou chave secreta. Caso *Ke* ≠ *Kd*, o sistema criptográfico é considerado assimétrico, chave composta ou chave pública. Dentre os principais métodos simétricos podemos citar: DES (*Data Encryption Standard*) [NIST 1977] e AES (*Advanced Encryption Standard*) [NIST 2001]. Os métodos assimétricos mais conhecidos são o RSA [Rivest, Shamir e Adleman 1977] e o ElGamal [ElGamal 1985].

A maioria dos algoritmos de criptografia assimétrica utiliza os conceitos empregados na teoria dos números, enquanto que os métodos simétricos geralmente possuem seu núcleo baseado em substituição e transposição. Métodos assimétricos costumam ser mais lentos que os simétricos e, por isso, os protocolos de segurança geralmente utilizam as duas formas, onde os sistemas assimétricos são utilizados para trocar a chave criptográfica de um sistema simétrico e, em seguida, passam a se comunicar apenas com o método simétrico.

Quanto à forma em que o texto claro é processado, os algoritmos criptográficos podem se dividir em duas categorias: cifragem de bloco (*block cipher*) e cifragem de fluxo (*stream cipher).* Na cifragem de bloco, o texto claro é dividido em blocos de tamanho fixo; cada bloco é processado e produz a saída em blocos correspondentes a cada bloco de entrada. Na cifragem de fluxo, também conhecida como cifragem por demanda, o texto claro é processado continuamente, produzindo como saída um elemento por vez, conforme o texto claro é processado.

Se um algoritmo criptográfico é conhecido apenas pelas pessoas que trabalharam na sua construção, o mesmo é nomeado de restrito. Se a segurança do algoritmo se baseia apenas no sigilo de seu desenvolvimento, e algum membro deixar a equipe, o sistema estará comprometido. Seguem abaixo algumas características desejáveis em um sistema

criptográfico:

- Os algoritmos "*E*" e "*D*" não podem conter nenhum componente ou parte do desenvolvimento mantido em segredo;
- A mensagem de entrada ao passar pela transformação "*E*" é distribuída de forma razoavelmente uniforme sob o espaço de possíveis mensagens cifradas;
- Com a chave criptográfica correta, "*E*" e "*D*" devem ser eficientes;
- Sem o conhecimento da chave criptográfica, a tarefa de recuperar o texto claro a partir do texto cifrado é um problema de dificuldade determinada somente pelo tamanho da chave, que usualmente tem um tamanho *s*, que para ser resolvido requer recursos computacionais de uma medida quantitativa por volta de $p(s)$, sendo p algum polinômio.

2.3.1 Métodos envolvendo múltiplas técnicas de cifragem

- **Enigma e Colosso**

Durante a segunda guerra mundial, os alemães construíram uma máquina que utiliza o conceito de múltiplos estágios em seu processo de cifragem. Tal máquina era conhecida como Enigma, sendo composta de um teclado e três rotores contendo o alfabeto.

Cada rotor tinha 26 pinos de entrada e 26 pinos de saída, onde cada pino da entrada era conectado a um único pino da saída. Se for associado cada pino de entrada e saída com alguma letra, então um rotor pode ser considerado uma cifra de substituição monoalfabética. Além disso, a cada letra pressionada, o rotor gira por uma posição, então, as conexões internas se deslocam da mesma maneira. Sendo assim, outro tipo de cifra monoalfabética está sendo usado. Após 26 letras do texto claro serem digitadas, o cilindro retornará a posição inicial. Portanto, existe também embutido no processo de cifragem, um tipo de cifra polialfabética com um período de tamanho 26 [Stallings 2003]. Para conexão entre os rotores, os pinos de saída de um rotor eram conectados aos pinos de entrada do rotor seguinte, e por fim, o pino de saída do último rotor era associado a uma letra do alfabeto. Cada um dos rotores girava em velocidades diferentes, sendo que o rotor anterior girava somente após o rotor da frente ter completado uma volta completa. No entanto, o matemático Alan Turing e seus colaboradores desenvolveram uma máquina chamada "Colosso", capaz de quebrar os códigos gerados pela máquina Enigma.

- **Estrutura de Feistel**

Após a segunda guerra mundial, outra maneira de cifrar mensagens passou a ser bastante empregada nos métodos de cifragem de blocos. Tal processo ficou conhecido como estrutura de Feistel, homenageando o seu criador. No modelo de Feistel, o bloco de mensagem do texto claro, convertido em bits, é dividido em duas partes (L_1, R_1). As duas metades passam por *n* etapas (*rounds*) e então são novamente combinadas para produzir um bloco contendo o texto cifrado. Em cada etapa é executada uma operação (geralmente substituição e permutação) aplicada na metade direita do bloco de texto (R_1), juntamente com a função *F* e a sub-chave correspondente àquela etapa. Na seqüência, é realizada a operação OR-exclusivo (XOR) entre a metade esquerda do bloco de texto (L_1) e a saída da função *F*, conforme é ilustrado na parte esquerda da Figura 3. No final de cada etapa, os bits da metade esquerda são trocados de lado com a parte direita, que então, servem de entrada para a próxima etapa. Todas as etapas possuem a mesma estrutura, e geralmente, a função *F* é parametrizada com sub-chaves diferentes para cada etapa. Tais sub-chaves são derivadas da chave do sistema criptográfico.

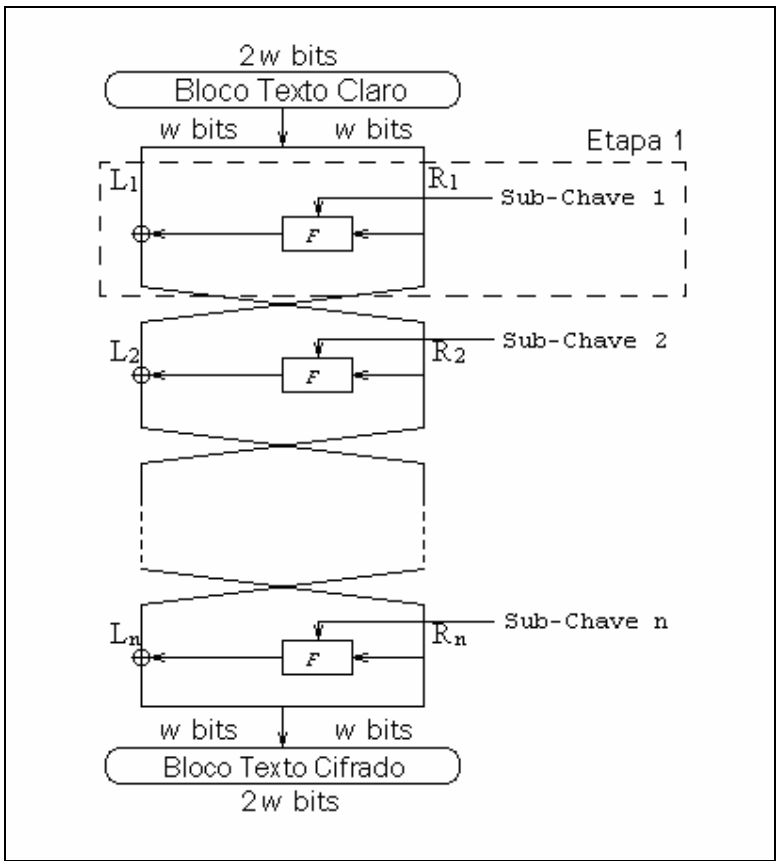


Figura 3 - Estrutura de Feistel

Os parâmetros referentes ao tamanho do bloco, tamanho da chave, número de etapas, algoritmo para gerar as sub-chaves e a função *F*, variam de acordo com o projeto do sistema criptográfico. Na forma como a estrutura de Feistel foi proposta não é necessária a construção de um algoritmo para cifrar e outro para decifrar, pois o mesmo algoritmo pode ser usado nas duas etapas. No processo de decifragem, é preciso apenas utilizar as sub-chaves na ordem inversa à que foi aplicada para cifrar.

- **Sistema criptográfico DES**

Desenvolvido no período de 1973-1974 pela equipe da IBM, o sistema criptográfico DES (Data Encryption Standard [NIST 1977] e [Stallings 2003]) é sem dúvida o mais estudado e citado até hoje em diversas pesquisas a título de comparação. O algoritmo utilizado no DES se baseia na estrutura de Feistel, com a diferença de possuir uma permutação no início e a correspondente permutação inversa no final.

No sistema criptográfico DES, o bloco de entrada contém um texto claro de 64 bits que é transformado, por uma série de passos, em um bloco de saída com 64 bits, utilizando-se uma chave de 56 bits. São efetuadas 16 etapas, onde cada etapa possui uma sub-chave diferente, de tamanho igual a 48 bits, a qual é produzida com a chave original (56 bits). Para gerar as sub-chaves, inicialmente é feita uma permutação entre os 56 bits da chave. Ao final da permutação, os 56 bits resultantes são separados em duas metades E_0 e D_0 de 28 bits cada. Em cada etapa, E_i e D_i sofrem, separadamente, deslocamento circular de duas posições, exceto as etapas de número 1, 2, 9 e 16, que possuem deslocamento circular de apenas uma posição. O resultado do deslocamento efetuado nas duas metades servem de entrada para a geração da sub-chave da próxima etapa, além de ser utilizado como entrada para uma segunda permutação, que produz 48 bits de saída. Estes 48 bits correspondem a uma sub-chave que é usada como parâmetro para função *F*. O lado direito da Figura 4 apresenta a geração das sub-chaves.

O texto claro no DES passa por uma permutação inicial, antes de ser processado pelas dezesseis etapas, que envolve funções contendo permutação e substituição. Após as dezesseis etapas ocorre uma troca entre a metade dos bits do lado direito com a outra metade dos bits do lado esquerdo. Esta troca é utilizada para que o mesmo algoritmo seja utilizado no processo de decifragem. Por fim, é efetuada a permutação inversa à executada na permutação inicial.

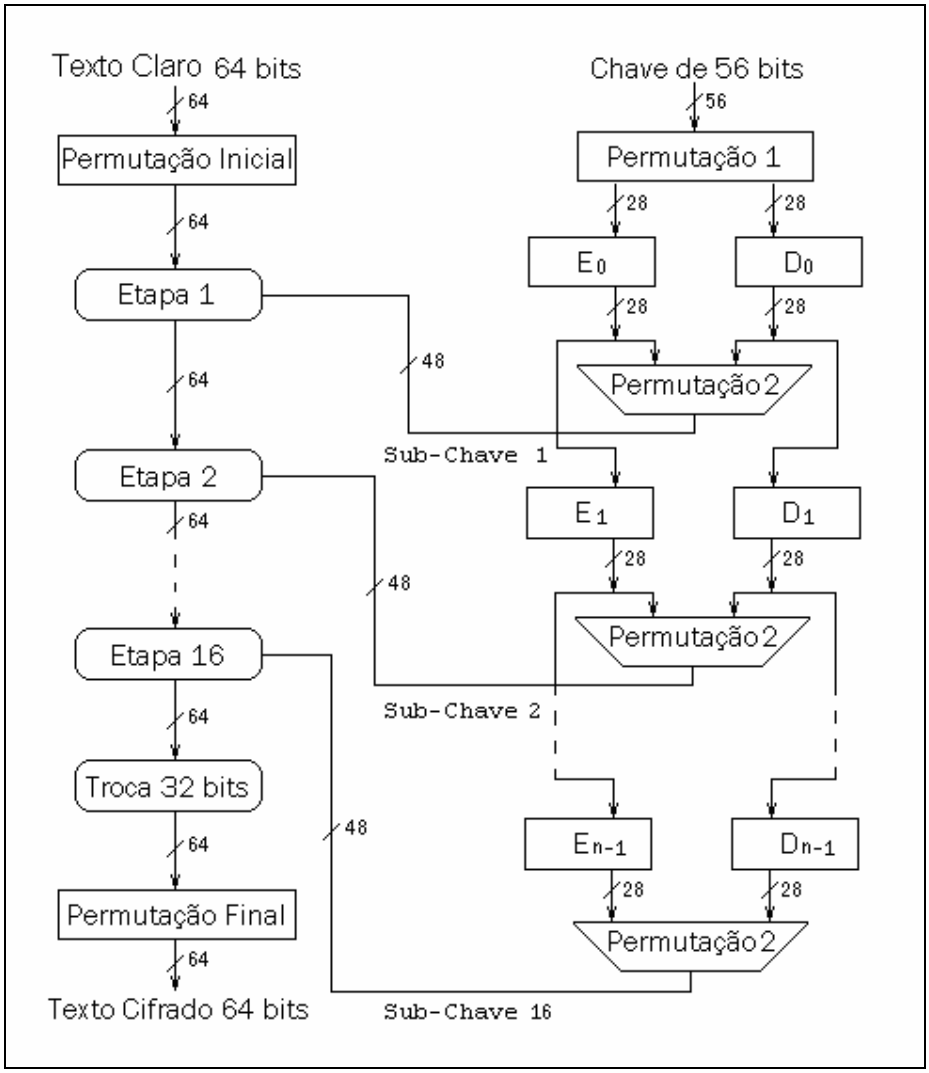


Figura 4 - Etapas do DES detalhando a geração das sub-chaves.

A estrutura de cada etapa pode ser visualizada na Figura 5. O bloco de texto de 64 bits é dividido em duas partes de 32 bits (*L*, *R*). Uma vez que *R* possui 32 bits, uma tabela de permutação/expansão é utilizada, onde 16 bits são duplicados, resultando em 48 bits que sofrem a operação XOR com os 48 bits da sub-chave. Os 48 bits resultantes da operação XOR são aplicados a 8 funções conhecidas por tabelas *S-box*. As tabelas *S-box* possuem como entrada 6 bits e como saída 4 bits. Dessa forma, os 48 bits serão consumidos pelas 8 tabelas *S-box* resultando em 32 bits de saída. Os 32 bits de saída das tabelas *S-box* passam por uma permutação e então sofrem a operação XOR com os bits de *L*, produzindo assim os bits correspondente a *R*. A função *F* está delimitada com uma linha pontilhada. Assim como na estrutura de Feistel, todo o processamento pode ser resumido nas equações (2) e (3):

$$L_i = R_{i-1} \quad (2)$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i) \quad (3)$$

A Tabela 5 apresenta um exemplo de *S-box*, que será explicado a seguir e a Figura 6 apresenta o esquema de transformação de 48 bits em 32 bits através da aplicação das 8 tabelas *S-box*.

Tabela 5 - Valores definidos pela *S-box* 6 (S_6) no DES.

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

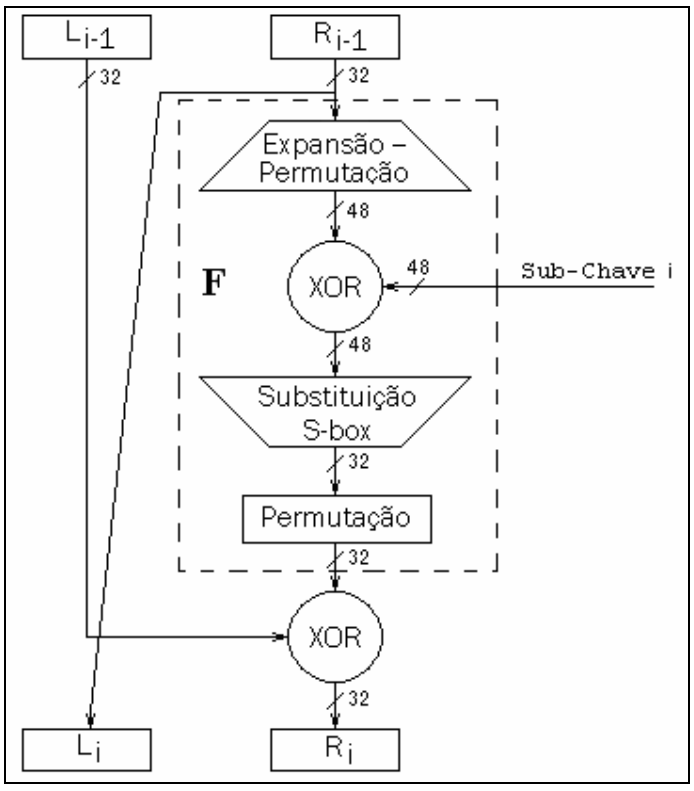


Figura 5 - Uma etapa do DES.

O valor de saída da tabela *S-box* é obtido através de uma consulta na linha e coluna desta tabela. Cada uma das 8 tabelas *S-box* possuem 4 linhas e 16 colunas, sendo que, os 4 bits de saída são obtidos a partir de 6 bits de entrada (Figura 6). A linha é definida pelo primeiro e o último bit da entrada (2 bits), que combinados formam um número na base

decimal. Dentre os 6 bits de entrada, o valor da coluna é obtido pelos 4 bits do meio convertidos na base decimal. Os números contidos nas tabelas *S-box* variam de 0 a 15, sendo assim, os bits de saída podem variar de 0000 a 1111. Como exemplo do processo efetuado em uma tabela *S-box*, considere os 6 bits de entrada {001001} sendo aplicados à tabela *S-box* 6 (Tabela 5). Os bits da extremidade {01}, convertidos para base decimal, correspondem à linha 1 da tabela *S-box*. Os 4 bits do meio, referentes aos 6 bits de entrada, são os bits {0100}, que na base decimal representam a coluna 4. Ao consultar a linha 1 e coluna 4 na tabela *S-box* é obtido o número 15, que convertido para base binária produz os 4 bits de saída {1111}. Portanto, a entrada {001001} é convertida na saída {1111}, se submetida à tabela *S-box* 6 do DES.

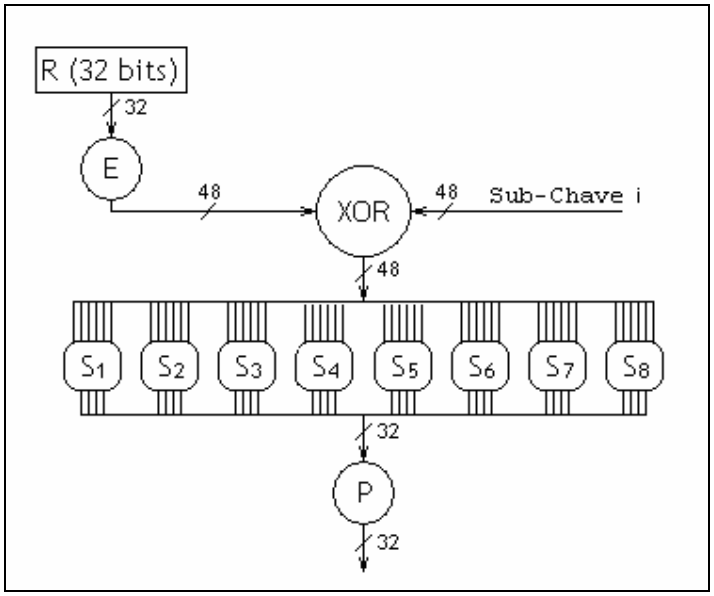


Figura 6 - Função *F* detalhada.

As tabelas de permutação, assim como os valores das *S-boxes*, podem ser encontrados em [NIST 1977] e [Stallings 2003, cap. 3].

O tamanho da chave utilizada no DES não é mais adequado para os padrões atuais, visto que existem máquinas capazes de quebrar o sistema em minutos com um ataque de força bruta. Dessa forma, o DES foi atualizado para versões com chaves maiores, conhecidas como 2DES e 3DES. Na versão 2DES (3DES) o texto claro é cifrado por duas (três) vezes com o DES padrão, porém, podendo empregar chaves diferentes a cada execução do DES. Dessa forma, a chave é estendida para 112 (168) bits. O inconveniente nesses novos modelos é que o sistema criptográfico acaba ficando mais lento, pois o mesmo método é executado mais vezes.

- **Sistema criptográfico AES**

Em novembro de 2001 foi publicado pela *Federal Information Processing Standards Publications* (FIPS PUBS 197) um novo padrão denominado AES (*Advanced Encryption Standard* [NIST 2001] e [Stallings 2003]). O padrão especifica o algoritmo selecionado pelo processo seletivo do AES conhecido como Rijndael, uma combinação do nome de seus inventores: Joan Daemen e Vincent Rijmen.

Rijndael é um sistema criptográfico simétrico que utiliza cifragem de blocos, podendo processar blocos de 128 bits, com chaves criptográficas de 128, 192 e 256 bits. Este algoritmo foi projetado para permitir outros tamanhos de bloco e chave, porém estes não foram adotados no padrão. Após se tornar padrão, o algoritmo Rijndael passou a ser referenciado como AES quando se refere ao algoritmo em geral, e de forma mais específica como: AES-128, AES-192 e AES-256, quando o tamanho da chave é mencionado. Diferentemente de seu predecessor DES, que utiliza a estrutura proposta por Feistel, o AES faz uso de uma estrutura conhecida como rede de substituição e permutação (SPN). Uma SPN utiliza etapas (*rounds*) contendo caixas de substituição conhecidas como *S-box* e caixas de permutação denominadas *P-box*, onde cada etapa geralmente é combinada com a chave através de alguma operação de grupo, como por exemplo, a operação XOR. A Figura 7 apresenta o modelo geral de uma rede SPN. As caixas *S* correspondem as *S-box* e os blocos contendo a letra *P* representam as *P-box*.

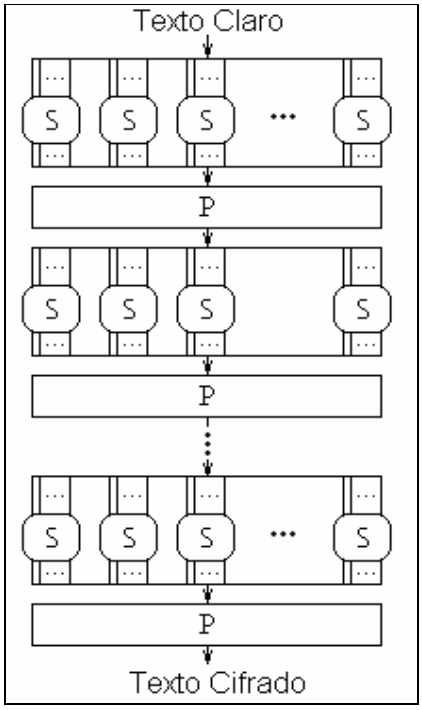


Figura 7 - Estrutura de uma rede de substituição e permutação (SPN).

No AES o número de etapas executadas depende do tamanho da chave. Para chave de 128, 192 ou 256 bits, são executadas 10, 12 e 14 etapas, respectivamente. O bloco de texto é formado por 128 bits que será copiado para uma matriz S_{4x4} (*State array*), onde cada célula da matriz possui 8 bits. As quatro operações executadas pelo AES no processo de cifragem (*SubBytes*(), *ShiftRows*(), *MixColumns*(), e *AddRoundKey*()), e as quatro operações responsáveis pela decifragem (*InvSubBytes*(), *InvShiftRows*(), *InvMixColumns*() e *AddRoundKey*()), são realizadas sobre S_{4x4}, que ao final do processo de cifragem ou decifragem faz a cópia inversa à efetuada inicialmente. A Figura 8 mostra um diagrama com uma visão geral das etapas efetuadas pelo AES-128. Todas as etapas são formadas pelas mesmas operações, exceto a última que não contém a operação *MixColumns*(). Esta afirmação vale para os três tamanhos de chave do AES.

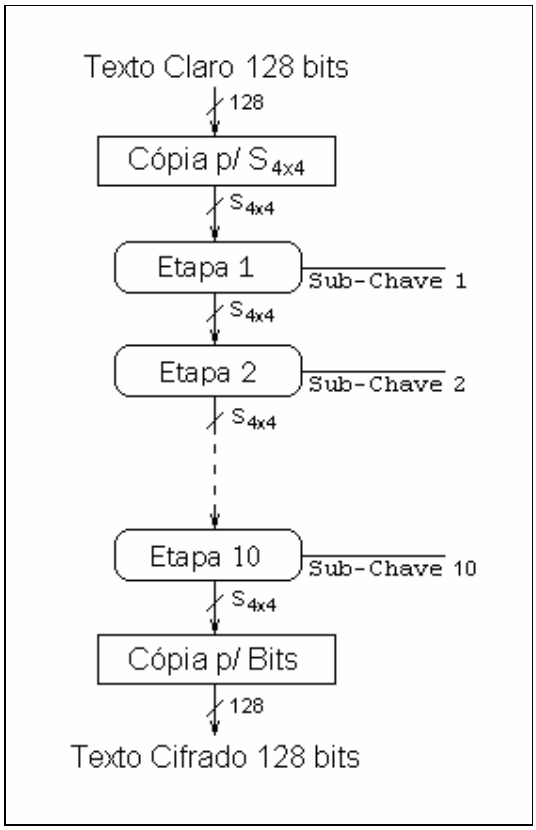


Figura 8 - Diagrama representando as etapas do AES-128.

A Figura 9 apresenta um diagrama detalhando a seqüência de passos efetuada nas etapas do AES. Todas as etapas, exceto a última, seguem a estrutura da Figura 9 (a); a última etapa é indicada pela Figura 9 (b).

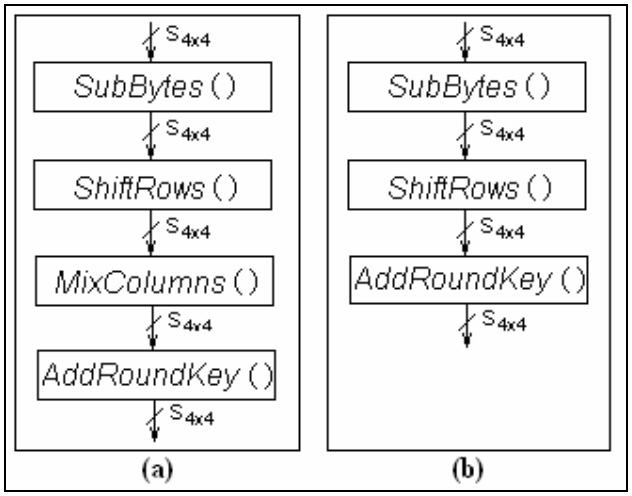


Figura 9 - (a) Seqüência de passos executados em cada etapa do AES, exceto a última. (b) Seqüência de passos executados apenas na última etapa.

Detalhes sobre cada uma das funções executadas no AES podem ser encontrados em [NIST 2001]. O sistema criptográfico AES é rápido ao ser implementado tanto em software como em hardware, além de ser relativamente fácil de implementar e exigir pouca memória [Wikipédia 2007a].

2.4 Criptoanálise

O processo de tentar descobrir o texto claro ou chave secreta ou ambos é conhecido como criptoanálise. A estratégia usada pelo criptoanalista depende da natureza do esquema de cifragem e da informação disponível para o criptoanalista [Stallings 2003]. Os tipos de ataque podem ser classificados de acordo com a disponibilidade de informações que podem ser obtidas pelo criptoanalista, conforme apresenta a Tabela 6.

Tabela 6 - Principais tipos de ataques em mensagens cifradas.

Tipo de Ataque	Informação Disponível
Somente texto cifrado	Algoritmo criptográfico; Texto cifrado.
Texto claro conhecido	Algoritmo criptográfico; Texto cifrado; Um ou mais pares de texto claro e texto cifrado usando a chave secreta.
Texto claro escolhido	Algoritmo criptográfico; Texto cifrado; Texto claro escolhido pelo criptoanalista, junto com seu correspondente texto cifrado gerado com a chave secreta.

O ataque feito quando se tem conhecimento apenas do algoritmo criptográfico e do texto cifrado (*ciphertext only*) é considerado um dos mais ineficientes. Geralmente estas técnicas envolvem a aplicação de vários testes estatísticos no texto cifrado, como por exemplo, a análise de freqüência ou tentativa de descobrir algum dígrafo do texto claro, dentre outros. Dado o reduzido número de informações, uma das possíveis tentativas (muitas vezes nem considerado um ataque por alguns autores) é conhecida como força bruta, onde todas as chaves plausíveis são testadas. Se o espaço de chaves é grande o suficiente, este ataque se torna impraticável.

O criptoanalista pode ser capaz de capturar alguns pares de texto claro e texto cifrado, e com isso, tentar descobrir a chave criptográfica explorando a forma como o texto claro foi transformado em texto cifrado. Este ataque é conhecido como texto claro conhecido (*known plaintext*).

Se de alguma forma o criptoanalista for capaz de escolher o texto claro ou inserir em uma mensagem algum texto escolhido por ele, o ataque recebe o nome de texto claro escolhido (*chosen plaintext*).

De acordo com Stallings (2003), um esquema de cifragem é computacionalmente seguro se dois critérios são atendidos:

- O custo de quebrar o texto cifrado excede o valor da informação cifrada;
- O tempo requerido para quebrar a cifra excede a vida útil da informação.

Porém, é muito difícil estimar qual será o esforço requerido para fazer uma criptoanálise bem sucedida em um texto cifrado.

No estudo da criptoanálise dois métodos merecem destaque, são eles a criptoanálise linear [Matsui 1994] e a criptoanálise diferencial [Biham and Shamir 1991].

2.4.1 Criptoanálise diferencial

A técnica conhecida como criptoanálise diferencial foi publicada em 1991 pelos pesquisadores Eli Biham e Adi Shamir, porém, o método já era conhecido em 1974 pelos projetistas do DES. Segundo um dos membros da equipe a técnica não foi divulgada por motivo de segurança nacional dos Estados Unidos [Stallings 2003].

A criptoanálise diferencial foi o primeiro método que reduziu a complexidade de ataque ao DES abaixo da prática de força bruta. Inicialmente, foi definido na categoria de

texto claro escolhido (*chosen plaintext)* e posteriormente sofreu adaptações para texto claro conhecido (*known plaintext*).

Como visto na seção 2.3.1, cada etapa do algoritmo DES faz o processamento de apenas uma metade de um bloco de 64 bits. Assim, a metade direita de uma etapa, recebe como entrada a metade esquerda do bloco anterior, que é processada com a sub-chave daquela etapa. A metade esquerda recebe como entrada a metade direita do bloco anterior. Assim, cada etapa intermediária é calculada como mostra a equação (4).

$$m_{i+1} = m_{i-1} \oplus f(m_i, k_i), \qquad i = 1,\ 2,\ ...,16 \tag{4}$$

Na criptoanálise diferencial, o ataque é iniciado com duas mensagens m e $m`$ com uma diferença conhecida entre as duas mensagens dada por $\Delta m = m \oplus m`$. Considere a diferença entre a metade da mensagem intermediária como: $\Delta m_i = m_i \oplus m_i`$. Dessa forma, pode-se obter a equação (5).

$$\begin{aligned} \Delta m_{i+1} &= m_{i+1} \oplus m`_{i+1} \\ &= [m_{i-1} \oplus f(m_i, k_i)] \oplus [m`_{i-1} \oplus f(m`_i, k_i)] \\ &= \Delta m_{i-1} \oplus [f(m_i, k_i) \oplus f(m`_i, k_i)] \end{aligned} \tag{5}$$

Suponha que diversos pares de entrada para a função f, com a mesma diferença, resulta na mesma diferença nos pares de saída, se a mesma sub-chave é usada. Mais precisamente, vamos dizer que *X* pode causar *Y* com probabilidade *p*. O ataque supõe que existe um número de valores de *X* que tem alta probabilidade de causar uma particular diferença na saída. Se o valor de Δm_{i-1} e Δm_i possui uma alta probabilidade de ser conhecido, então o valor de Δm_{i+1} pode ser conhecido com alta probabilidade. Além do mais, se um número destas diferenças são determinados, é provável determinar a sub-chave utilizada em f [Stallings 2003].

Toda a estratégia do ataque de criptoanálise diferencial é baseada nestas considerações para uma simples etapa (*round*).

Apesar da criptoanálise diferencial ter sido projetada inicialmente para sistemas criptográficos que utilizam uma estrutura como a de Feistel, Sen e colaboradores (2002) submeteram seu sistema criptográfico, denominado CAC (*Cellular Automata Based Cryptosystem*), que será detalhado na seção 4.2, a um teste de criptoanálise diferencial.

O ataque se baseia na diferença *D* entre pares do texto claro (*X*, *X'*), e a diferença *D'* resultante do texto cifrado correspondente (*Y*, *Y'*). Para cada par analisado, contam-se quantos bits diferentes foram obtidos na saída, assim como o desvio padrão entre os vários pares analisados. Se o valor obtido pelo desvio padrão estiver abaixo de 10%, o sistema pode ser considerado seguro contra o ataque de criptoanálise diferencial [Sen *et al*. 2002].

Esta mesma técnica de criptoanálise diferencial será utilizada para avaliar o método proposto nessa dissertação. Informações sobre a criptoanálise diferencial aplicada sobre o DES podem ser encontradas em [Howard 2002] e [Biham and Shamir 1991].

2.4.2 Criptoanálise linear

A criptoanálise linear foi criada por Matsui (1994) e teve como alvo o algoritmo criptográfico DES. Esta técnica é caracterizada como um ataque de texto claro conhecido.

O método tenta tirar vantagem da alta probabilidade de ocorrer uma expressão linear entre os bits do texto claro, texto cifrado e sub-chave.

A idéia básica é aproximar uma porção de texto cifrado e texto claro com uma expressão linear, onde a linearidade diz respeito a operação bit a bit mod-2 (OR exclusivo representada aqui por $\oplus$), tal como a expressão abaixo:

$$X_{i_1} \oplus X_{i_2} \oplus \ldots \oplus X_{i_u} \oplus Y_{j_1} \oplus Y_{j_2} \oplus \ldots \oplus Y_{j_v} = 0 \qquad (6)$$

Onde X_i representa o i-ésimo bit da entrada $X = [X_1, X_2, ...]$ e Y_j representa o j-ésimo bit da saída $Y = [Y_1, Y_2, ...]$. Esta equação indica a soma do operador OU-exclusivo entre os "*u*" bits de entrada e os "*v*" bits de saída.

Se o sistema criptográfico apresentar uma tendência para a equação (6) ocorrer com alta ou baixa probabilidade, este sistema possui grande dificuldade em tornar o texto cifrado aleatório, tornando-se vulnerável ao ataque. Detalhes da criptoanálise linear sobre o DES podem ser encontrados em [Howard 2002] e [Matsui 1994].

Capítulo 3

Autômatos celulares

Um autômato celular é um modelo discreto estudado em diferentes áreas, tais como a teoria da computabilidade, a matemática, e a biologia. Os autômatos celulares (ACs) foram introduzidos nos anos 50 pelo matemático John Von Neumann, levando em conta sugestões do físico Stanislaw Ulam [Wikipédia 2006]. Von Neumann estava interessado nas conexões entre a biologia e a teoria dos autômatos. Nos seus estudos, predominava a idéia do fenômeno biológico da auto-reprodução. A questão que ele apresentava era: "Que tipo de organização lógica é suficiente para um autômato ser capaz de reproduzir a si próprio?" [Aguiar e Costa 2001].

Autômatos celulares são sistemas dinâmicos totalmente discretos, ou seja, possuem variáveis discretas na representação do tempo, do espaço e dos estados. Podem ser aplicados a diversas áreas, e nesta pesquisa, serão empregados como método de criptografia.

De forma geral, um AC é definido por seu espaço celular e por sua regra de transição. O espaço celular é um reticulado de N células idênticas dispostas em um arranjo d-dimensional, cada uma com um padrão idêntico de conexões locais para outras células, e com condições de contorno [Oliveira 2003]. Cada célula do reticulado assume um estado a cada passo de tempo, dentre um conjunto finito de estados possíveis. A regra de transição faz um mapeamento entre células vizinhas para determinar o novo estado da célula central da vizinhança. Após a aplicação da regra de transição em todas as células do reticulado, que pode ser de forma síncrona ou não, é contado um passo de tempo. Normalmente, o reticulado do AC é submetido à regra de transição por vários passos de tempo resultando no que chamamos de evolução temporal do AC. Os ACs mais pesquisados são binários (dois estados possíveis), unidimensionais e evoluem de forma síncrona.

3.1 AC unidimensional

Um AC unidimensional consiste de uma coleção de variáveis discretas dependentes do tempo s_t^i, chamadas de estados locais, e um reticulado unidimensional de N células, $i = 0, 1, \cdots, N-1$. Cada célula será considerada como uma variável booleana: $s_t^i \in \{0,1\}$. A coleção de todos os estados locais das células que formam o reticulado em um determinado

instante de tempo *t* é chamada de *configuração do reticulado*: $s_t = s_t^0 s_t^1 \cdots s_t^{N-1}$, sendo que s_0 denota a configuração inicial (CI). Tipicamente, a equação dinâmica para um AC é especificada pela regra de transição de estados ϕ, responsável por mapear a vizinhança de uma célula s_t^i, chamada η_t^i, para a célula em questão, no próximo espaço de tempo: $s_{t+1}^i = \phi(\eta_t^i)$, onde $\eta_t^i = s_t^{i-r} \cdots s_t^i \cdots s_t^{i+r}$ e *r* é chamado de raio do AC. Usaremos simplesmente η (sem *i* ou *t*) para denotar a vizinhança. A equação dinâmica global Φ mapeia a configuração de um passo de tempo para o próximo: $s_{t+1} = \Phi(s_t)$, onde é entendido que a função local ϕ é aplicada simultaneamente em todas as células do reticulado [Das *et al.* 1995].

Com relação às células extremas do reticulado, deve-se definir a condição de contorno do reticulado. A condição de contorno periódica é definida como $s_t^i = s_t^{i+N}$, ou seja, neste caso o reticulado possui as células da extremidade conectadas formando um anel. Outra condição de contorno comum é determinar que os bits restantes da vizinhança (das células extremas) tenham sempre o valor 0, que recebe o nome de condição de contorno nula.

Se cada célula de um AC evolui com a mesma regra de transição de estados, este AC é chamado de homogêneo, caso contrário, o AC é não-homogêneo ou híbrido.

A Figura 10 mostra um AC binário, homogêneo, unidimensional, contendo dez células ($N = 10$), com reticulado inicial $s_0 = \{1001001110\}$, raio 1 e condição de contorno periódica. Na parte inferior da figura encontra-se o mapeamento de transição ϕ para uma vizinhança de tamanho três, que resulta na regra de transição $\Phi = \{10010101\}$, a qual está sendo representada com o bit menos significativo (LSB) à esquerda. Em destaque na Figura 10, pode ser vista a atualização da terceira célula de acordo com a configuração da vizinhança e a regra de transição: $s_1^3 = \phi(s_0^2, s_0^3, s_0^4) = \phi(0,0,1) = 0$. Esse processo é repetido para todas as outras células, resultando na transição do reticulado $\Phi(1001001110) = (0000001101)$ em um passo de tempo.

A evolução dos estados de um AC por diversos passos de tempo pode ser visualizada por um diagrama espaço-temporal. Esta representação possui um formato de matriz em que cada linha corresponde ao espaço celular em uma determinada evolução, e o número de colunas representa a quantidade de células do espaço celular. A evolução no tempo pode ser visualizada de cima para baixo ou de baixo para cima. Nesta dissertação, o diagrama espaço-

temporal será representado com a evolução de cima para baixo, exceto se informado o contrário.

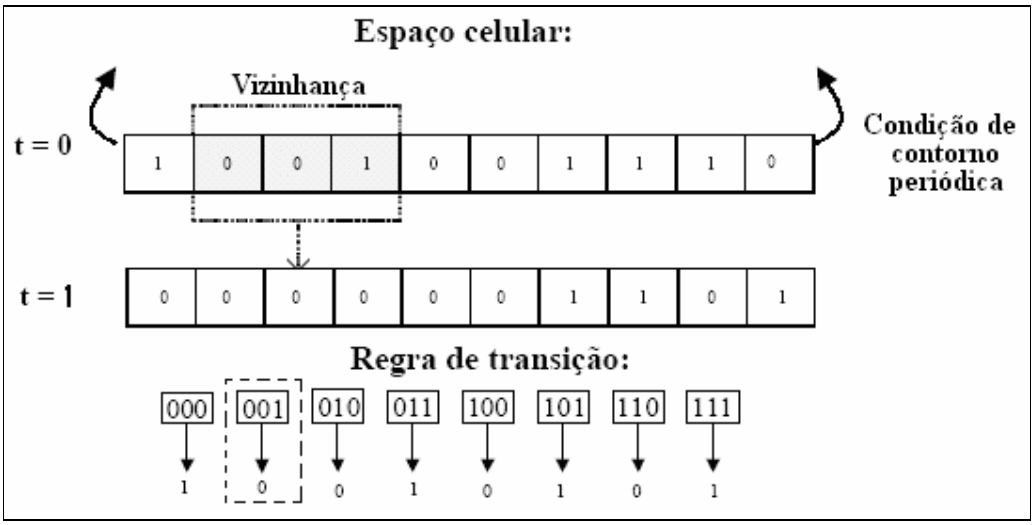


Figura 10 - Exemplo de AC unidimensional.

A Figura 11 (a) apresenta um diagrama espaço-temporal onde as 10 células do reticulado evoluem seguindo a mesma regra por 15 passos de tempo. A regra e o reticulado inicial são os mesmos empregados na Figura 10. A Figura 11 (b) corresponde ao mesmo diagrama da Figura 11 (a) com as células contendo o estado 1 preenchidas e as células com o estado 0 sem preenchimento. Este tipo de visualização da evolução espaço-temporal das células é largamente utilizado, pois facilita a identificação de padrões associados à dinâmica dos ACs.

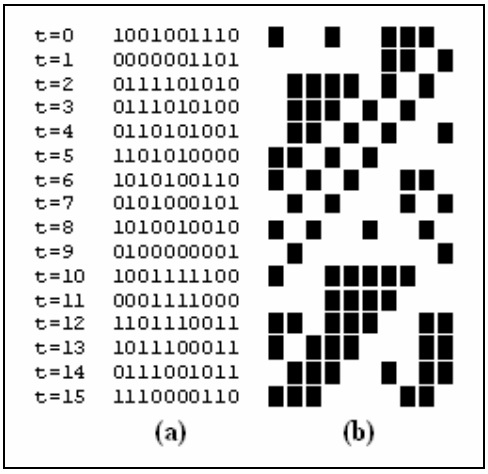


Figura 11 - Diagrama espaço-temporal representando a evolução de um AC.

ACs elementares são aqueles que possuem a configuração mais simples possível (não

trivial), sendo assim, eles são unidimensionais, homogêneos, binários, e com tamanho de raio igual a 1. Nesta configuração, a vizinhança possui três células, e portanto, existem oito (2^3) combinações de vizinhança, totalizando 256 (2^{2^3}) regras de transição possíveis. Wolfram propôs que os bits de saída da regra de transição fossem ordenados lexicograficamente como indicado na Figura 10, e lidos da direita para esquerda formando um número entre 0 e 255 na base decimal. Dessa forma, o AC apresentado na Figura 10 é um AC elementar que possui a regra de transição corresponde ao número 169 na base decimal. Para ACs com raios maiores a quantidade de regras aumenta significativamente (raio 2: 2^{2^5} regras; raio 3: 2^{2^7} regras) e elas são normalmente representadas na base hexadecimal.

Diversas variações de ACs podem ser encontradas na literatura: os reticulados podem ser definidos em *N* dimensões; maior número de estados por célula (não apenas binário); ter uma condição de contorno diferente, além da nula ou periódica citada anteriormente; ou até conter uma ou mais células que evoluem com uma regra de transição distinta (ACs não-homogêneos). Os ACs aqui pesquisados são binários, unidimensionais e evoluem de forma síncrona. Diferentes tamanhos de raio e de reticulado foram analisados. Serão apresentados, no Capítulo 4, tanto ACs homogêneos quanto não-homogêneos, podendo possuir condição de contorno nula ou periódica.

3.2 Classificação dinâmica das regras

Existem diferentes formas para classificar o comportamento dinâmico observado dos ACs, dependendo do grau de refinamento desejado [Oliveira 2003]. A classificação dinâmica mais conhecida foi proposta por Wolfram (1984) e divide os ACs em 4 classes:

Classe 1: quase todas as configurações iniciais convergem, após um período transiente, para uma configuração homogênea, como mostra a Figura 12.

Classe 2: quase todas configurações iniciais convergem, após um período transiente, como mostra a Figura 13; (a) para algum ponto-fixo, (b) para algum ciclo periódico de configurações, podendo ocorrer deslocamento para a direita (c) ou esquerda (d).

Classe 3: quase todas configurações iniciais resultam, após um período transiente, em um comportamento caótico, como nos dois exemplos da Figura 14.

Classe 4: algumas configurações iniciais resultam em estruturas localizadas complexas, algumas vezes bastante duradouras (Figura 15).

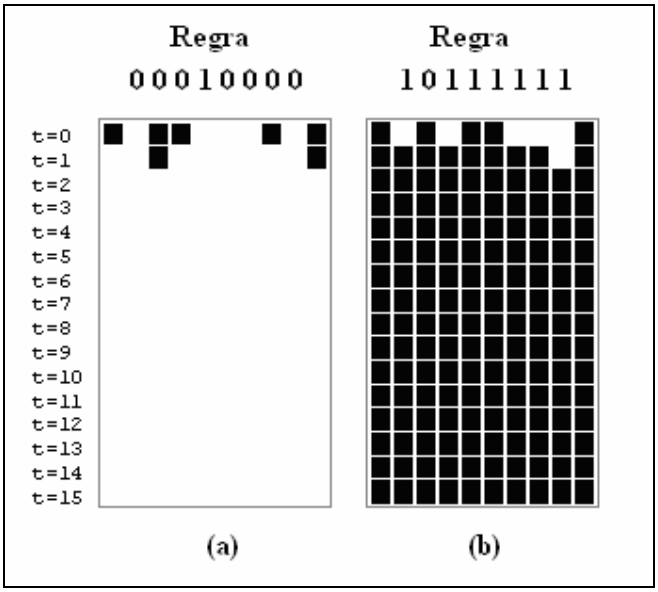


Figura 12 - Exemplo de regras pertencentes à classe 1: (a) regra 8; (b) regra 253.

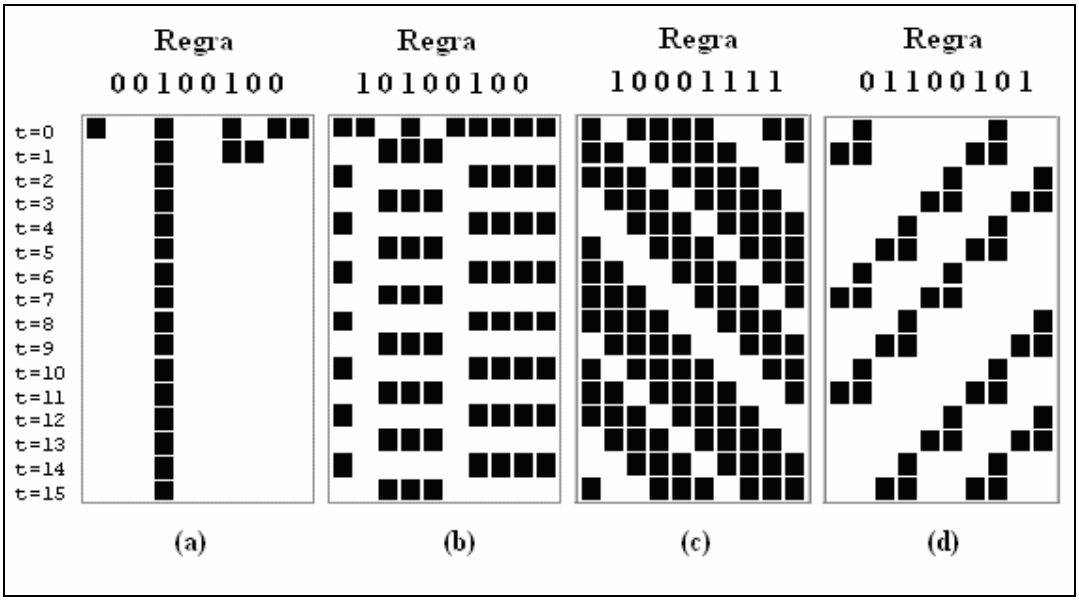


Figura 13 - Exemplo de regras pertencentes à classe 2: (a) regra 36, (b) regra 37, (c) regra 241 e (d) regra 166.

Regras pertencentes à classe 3 são muito sensíveis à configuração inicial do reticulado, ou seja, alterando-se poucos bits no reticulado inicial, os reticulados resultantes da evolução temporal sofrem grandes alterações. Uma regra pertencente a esta classe possui um alto grau de aleatoriedade em sua evolução. Por isso, regras caóticas são empregadas em diversos sistemas criptográficos.

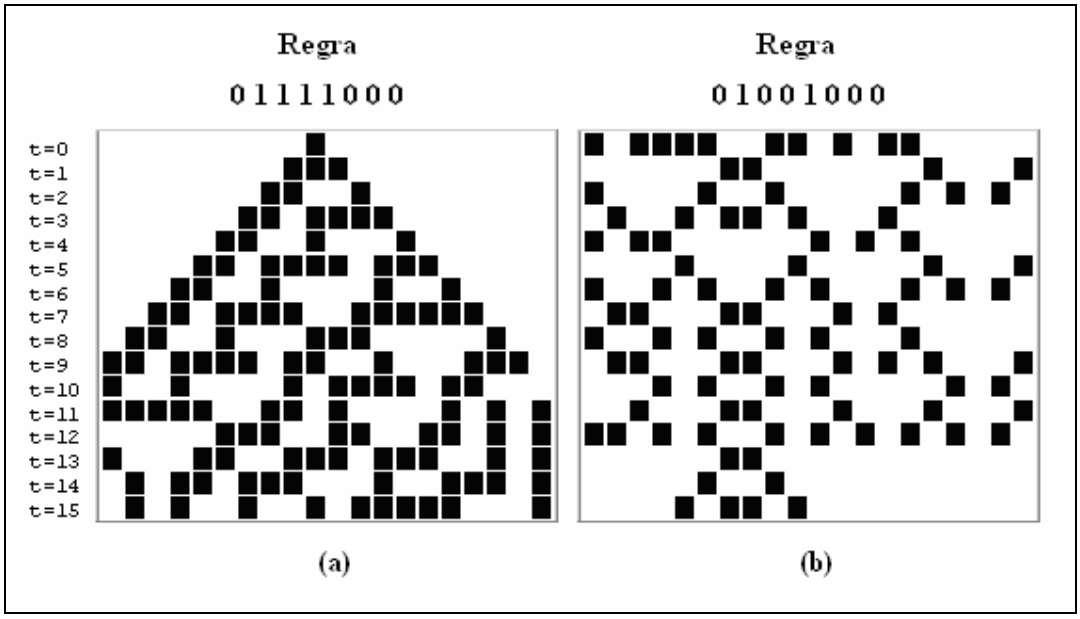


Figura 14 - Exemplo de regras pertencentes à classe 3: (a) regra 30, (b) regra 18.

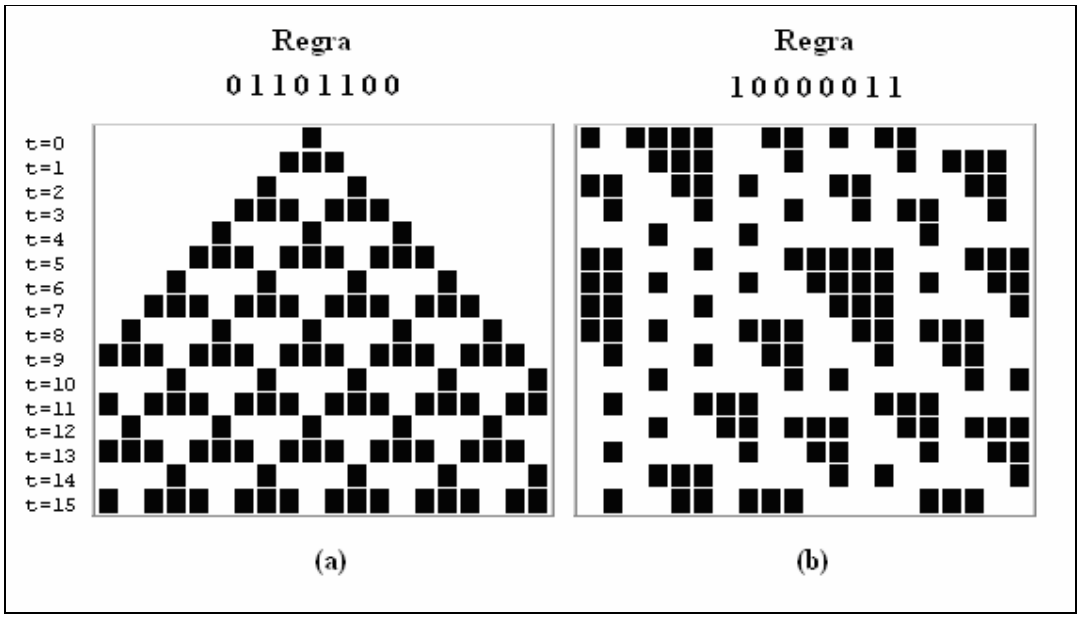


Figura 15 - Exemplo de regras pertencentes à classe 4: (a) regra 54, (b) regra 193.

A Tabela 7 apresenta as 256 regras elementares (binárias, unidimensionais, raio 1) e a classificação de seus respectivos comportamentos dinâmicos.

Tabela 7 - Classificação dinâmica das regras elementares [Lima 2005].

Classe	Comportamento	Regras
1	Nulo	0, 8, 32, 40, 64, 96, 128, 136, 160, 168, 192, 224, 234, 235, 238, 239, 248, 249, 250, 251, 252, 253, 254, 255
2	Ponto Fixo	4, 12, 13, 36, 44, 68, 69, 72, 76, 77, 78, 79, 92, 93, 100, 104, 132, 140, 141, 164, 172, 196, 197, 200, 202, 203, 204, 205, 206, 207, 216, 217, 218, 219, 220, 221, 222, 223, 226, 228, 232, 233, 236, 237 2*, 10*, 16*, 24*, 34*, 42*, 46*, 48*, 56*, 57*, 58*, 66*, 80*, 98*, 99*, 112*, 116*, 114*, 130*, 138*, 139*, 144*, 152*, 162*, 163*, 170*, 171*, 174*, 175*, 176*, 177*, 184*, 185*, 186*, 187*, 188*, 189*, 190*, 191*, 194*, 208*, 209*, 227*, 230*, 231*, 240*, 241*, 242*, 243*, 244*, 245*, 246*, 247*
2	Ciclo Duplo	1, 5, 19, 21, 23, 25, 28, 29, 31, 33, 37, 50, 51, 55, 61, 67, 70, 71, 87, 91, 95, 103, 108, 123, 127, 156, 157, 179, 198, 199, 201 3*, 6*, 7*, 9*, 11*, 14*, 15*, 17*, 20*, 27*, 35*, 38*, 39*, 43*, 47*, 49*, 52*, 53*, 59*, 63*, 65*, 74*, 81*, 83*, 84*, 85*, 88*, 111*, 113*, 115*, 117*, 119*, 125*, 134*, 142*, 143*, 148*, 155*, 158*, 159*, 173*, 178*, 211*, 212*, 213*, 214*, 215*, 229*
2	Periódico**	26, 41, 62, 82, 94, 97, 107, 118, 121, 131, 133, 145, 154, 166, 167, 180, 181, 210
3	Caótico	18, 22, 30, 45, 60, 73, 75, 86, 89, 90, 101, 102, 105, 106, 109, 120, 122, 126, 129, 135, 146, 149, 150, 151, 153, 161, 165, 169, 182, 183, 195, 225
4	Complexo	54, 110, 124, 137, 147, 193,
* regras que possuem deslocamento espacial para direita ou para esquerda ** regras cujo reticulado se repete em ciclos pequenos ($\neq 2$), como por exemplo, ciclo triplo ou quádruplo.		

3.3 Propriedades algébricas de ACs aditivos

Além da classificação dinâmica das regras, apresentada na seção 3.2, ferramentas algébricas também podem ser utilizadas na análise das propriedades globais dos ACs. Tais ferramentas são amplamente aplicadas ao uso de criptografia, existindo diversos sistemas criptográficos baseados em ACs que empregam os conceitos que são brevemente descritos nessa seção.

Regras de ACs podem ser representadas por um número decimal ou hexadecimal correspondente à sua cadeia de bits, e também, quando possível, por operações lógicas.

Assim, a regra <01011010> pode ser representada pelo número 90_d ou $5A_h$, além de poder ser escrita na forma lógica: $s_{t+1}^i = s_t^{i-1} \oplus s_t^{i+1}$, onde $\oplus$ representa a operação XOR, conforme apresenta a Tabela 8.

Tabela 8 - Representação da regra <01011010> na forma lógica: $S_{t+1}^i = S_t^{i-1} \oplus S_t^{i+1}$.

Configuração	000	001	010	011	100	101	110	111
$S_t^{i-1} \oplus S_t^{i+1}$	$0 \oplus 0$	$0 \oplus 1$	$0 \oplus 0$	$0 \oplus 1$	$1 \oplus 0$	$1 \oplus 1$	$1 \oplus 0$	$1 \oplus 1$
S_{t+1}^i	0	1	0	1	1	0	1	0

Se a regra do AC pode ser representada utilizando-se apenas operações lógicas XOR, esta regra é chamada de linear. Regras envolvendo exclusivamente operações lógicas XNOR são referidas como regras complementares. As regras que envolvam operação lógica AND ou OR são regras não-aditivas [Chaudhuri *et al*. 1997]. As dezesseis regras elementares lineares e complementares que podem ser empregadas em ACs aditivos são representadas na Tabela 9. Um AC não-homogêneo empregando apenas regras lineares também são chamados de ACs lineares, enquanto que ACs não-homogêneos que empregam regras tanto lineares quanto complementares são chamados de ACs aditivos.

Tabela 9 - Regras elementares lineares e complementares.

Operação XOR (AC linear)	**Operação XNOR (regras complementares)**
regra 0: $s_{t+1}^i = s_t^i \oplus s_t^i$	regra 255: $s_{t+1}^i = \overline{s_t^i \oplus s_t^i}$
regra 60: $s_{t+1}^i = s_t^{i-1} \oplus s_t^i$	regra 195: $s_{t+1}^i = \overline{s_t^{i-1} \oplus s_t^i}$
regra 90: $s_{t+1}^i = s_t^{i-1} \oplus s_t^{i+1}$	regra 165: $s_{t+1}^i = \overline{s_t^{i-1} \oplus s_t^{i+1}}$
regra 102: $s_{t+1}^i = s_t^i \oplus s_t^{i+1}$	regra 153: $s_{t+1}^i = \overline{s_t^i \oplus s_t^{i+1}}$
regra 150: $s_{t+1}^i = s_t^{i-1} \oplus s_t^i \oplus s_t^{i+1}$	regra 105: $s_{t+1}^i = \overline{s_t^{i-1} \oplus s_t^i \oplus s_t^{i+1}}$
regra 170: $s_{t+1}^i = s_t^{i+1}$	regra 85: $s_{t+1}^i = \overline{s_t^{i+1}}$
regra 204: $s_{t+1}^i = s_t^i$	regra 51: $s_{t+1}^i = \overline{s_t^i}$
regra 240: $s_{t+1}^i = s_t^{i-1}$	regra 15: $s_{t+1}^i = \overline{s_t^{i-1}}$

Além da representação da evolução de um AC através do diagrama espaço-temporal, como o apresentado na Figura 11, a evolução pode ser visualizada por um diagrama de transição de estados, onde cada estado representa uma configuração do reticulado. Por exemplo, a Figura 16 mostra a representação de um AC homogêneo formado com quatro células, através do diagrama de transição de estado. Como são quatro células no reticulado,

existem dezesseis estados possíveis: 0 (0000), 1 (0001), 2 (0010), 3 (0011), 4 (0100), 5 (0101), ..., 15 (1111). O AC possui condição de contorno periódica e todas as células são evoluídas pela regra 90. Cada nó representa uma configuração do reticulado e os arcos direcionados indicam a evolução do AC ao ser aplicada a regra de transição.

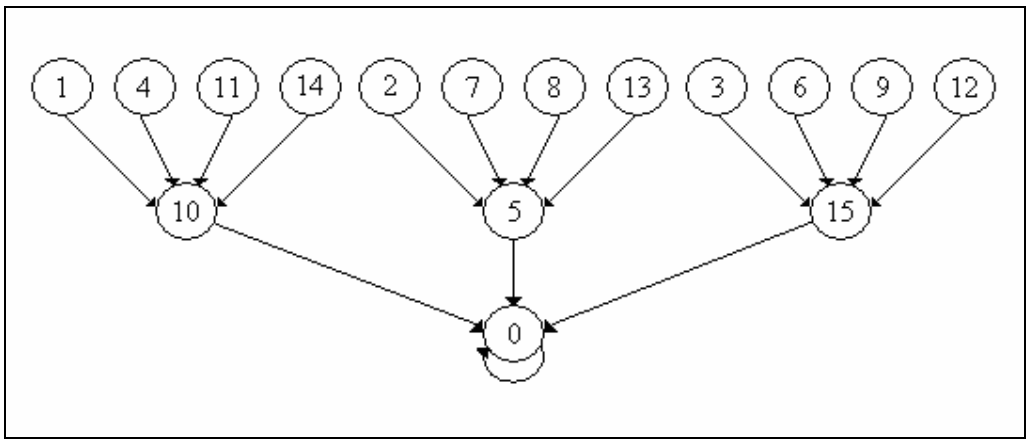


Figura 16 - Diagrama de transição de estados de um AC com quatro estados e condição de contorno periódica. Evolução uniforme com a regra 90.

Uma outra forma de representar o diagrama indicado na Figura 16 é pela matriz de transição apresentada abaixo.

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 0 & 10 & 5 & 15 & 10 & 0 & 15 & 5 & 5 & 15 & 0 & 10 & 15 & 5 & 10 & 0 \end{bmatrix}$$

Os valores referentes à primeira linha da matriz correspondem à configuração do reticulado inicial do AC. Os valores da segunda linha representam a configuração sucessora desse reticulado ao evoluir o AC por um passo de tempo.

É possível notar que não ocorre uma permutação entre os estados do diagrama mostrado na Figura 16, pois todos os estados possuem zero ou mais que um predecessor. Os estados que não possuem pelo menos um predecessor são chamados de "Jardim do Éden". Os estados do tipo Jardim do Éden nunca podem ser alcançados por uma evolução para frente, e por isso, são também conhecidos como estados inalcançáveis [Chaudhuri *et al*. 1997].

A Figura 17 corresponde ao diagrama de transição de estado de um AC com a mesma configuração do AC apresentado na Figura 16, porém, a condição de contorno aqui é nula. Cada nó deste diagrama possui um único predecessor e, portanto, a transição entre os estados do AC forma um ciclo.

Um AC linear, complementar ou aditivo, de *n* células (homogêneo ou não) pode ser

representado por uma matriz característica $n \times n$. A matriz característica é construída da seguinte forma: Se o próximo estado da *i*-ésima célula do reticulado depende da *j*-ésima célula do reticulado na evolução do AC, o elemento da matriz *T*[*i*,*j*] terá o valor 1, caso contrário, o elemento da matriz *T*[*i*,*j*] possuirá o valor 0.

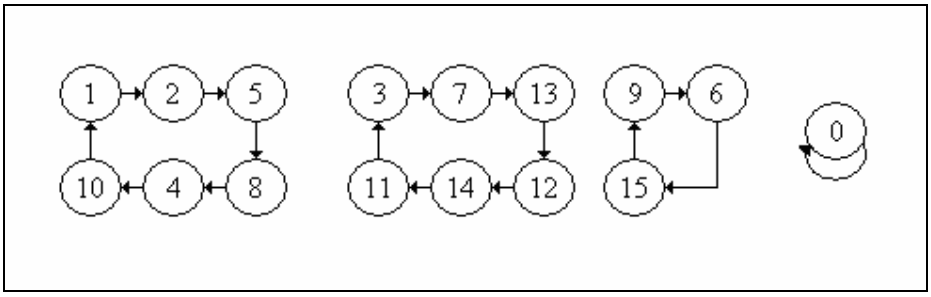


Figura 17 - Diagrama de transição de estados de um AC com quatro estados e condição de contorno nula. Evolução uniforme com a regra 90.

Como exemplo, considere um AC híbrido com quatro células e condição de contorno nula. O vetor de regras aplicado a cada célula do reticulado, da esquerda para direita, é <102, 150, 90, 150>, que convertido em suas respectivas combinações lógicas resulta em $< s_t^i \oplus s_t^{i+1}$, $s_t^{i-1} \oplus s_t^i \oplus s_t^{i+1}$, $s_t^{i-1} \oplus s_t^{i+1}$, $s_t^{i-1} \oplus s_t^i \oplus s_t^{i+1} >$. Uma vez que, neste exemplo, o AC é formado por quatro células, a matriz característica possui quatro linhas e quatro colunas. Cada uma das linhas da matriz é preenchida de acordo com uma regra do vetor de regras, da esquerda para direta, seguindo a definição da construção da matriz característica indicada no final do parágrafo anterior. Dessa forma a primeira linha da matriz é preenchida com a regra 102 ($s_t^i \oplus s_t^{i+1}$), a segunda com a regra 150 ($s_t^{i-1} \oplus s_t^i \oplus s_t^{i+1}$), a terceira com a regra 90 ($s_t^{i-1} \oplus s_t^{i+1}$) e a última com a regra 150 ($s_t^{i-1} \oplus s_t^i \oplus s_t^{i+1}$). A composição da primeira linha da matriz, referente à aplicação da regra 102 à célula ϕ do reticulado, é realizado da seguinte forma:

- O novo valor da célula 0 depende da célula 0, pois de acordo com a regra 102, a própria célula influencia na evolução, portanto *T*[0,0] = 1;
- O novo valor da célula 0 depende da célula 1, pois na regra 102, a célula vizinha direita influencia na evolução, portanto *T*[0,1] = 1;
- O novo valor da célula 0 não depende da célula 2, pois a célula 2 não está no escopo de vizinhança da célula 0, portanto *T*[0,2] = 0;
- O novo valor da célula 0 não depende da célula 3, pois a célula 3 não está no escopo de vizinhança da célula 0, portanto *T*[0,3] = 0;

De forma geral, temos:

$$T[i,j] = \begin{cases} 1, \text{ Se a célula } i \text{ depende da célula } j \text{ na evolução do AC;} \\ 0, \text{ caso contrário.} \end{cases} \quad (7)$$

Assim, a primeira linha da matriz corresponde a {1, 1, 0, 0}. As outras linhas também são preenchidas conforme a condição mostrada em (7), de acordo com a regra e a célula em questão. Assim, a matriz característica correspondente ao AC não-homogêneo <102, 150, 90,150>, com contorno nulo, é dada abaixo:

$$[T] = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Dessa forma, o próximo estado s_{t+1} de um autômato celular aditivo é dado por $s_{t+1} = T \times s_t$. Similarmente, *x* evoluções podem ser calculadas como: $s_{t+x} = T^x \times s_t$, onde *s* e *t* correspondem a configuração do reticulado e o tempo, respectivamente.

As regras complementares também podem ser usadas em conjunto com as regras lineares para gerar a matriz característica de ACs não-homogêneos aditivos. Porém, para evoluir este tipo de AC é necessário construir um vetor *F*, responsável pela inversão das regras complementares utilizadas. Na construção de *F*, as células que possuírem regras complementares devem ser marcadas com o estado 1. A operação XOR entre o vetor e a matriz característica fará com que as regras complementares sejam invertidas.

Como exemplo, considere um AC não-homogêneo com condição de contorno nula e as regras $< x^{-1} \oplus x^0, \overline{x^0}, \overline{x^0 \oplus x^1}, \overline{x^0 \oplus x^1} >$ aplicadas nas células do reticulado, da esquerda para direta. O vetor *F* é definido como $[0,1,1,1]^t$. Sendo assim, o próximo estado de um AC pode ser definido como:

$$s_{t+1} = F \oplus T \times s_t = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \oplus \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times s_t$$

A equação geral da evolução de um AC aditivo pode ser definida como: $s_{t+x} = F \oplus T^x \times s_t$. Se as regras empregadas são somente do tipo linear, o vetor *F* será

formado apenas por zeros, e caso as regras sejam todas do tipo complementar, o vetor conterá apenas uns.

Um AC possui propriedades de grupo se $T^p = I$, tal que I corresponde à matriz identidade e p é um inteiro positivo [Nandi 1994].

Dessa forma, se $T^p = I$, podemos definir a inversão da matriz de transição como: $T^{-1} = T^{p-1}$. Assim, a inversão de um reticulado no tempo *t-1* pode ser dada como: $s_{t-1} = F \oplus T^{-1} s_t = F \oplus T^{p-1} s_t$. De forma mais geral, temos: $s_{t-x} = F \oplus T^{p-x} s_t$.

Se a determinante da matriz característica T é igual a 1, então $\exists p \mid T^p = I$ e o AC possui propriedades de grupo (*group CA*) [Chaudhuri *et al*. 1997]. O polinômio característico associado a uma matriz característica T pode ser obtido calculando-se a matriz $T + \lambda I$ e o cálculo do determinante dessa matriz T. Por exemplo, para a matriz característica T apresentada anteriormente, a matriz $T + \lambda I$ é dada abaixo:

$$T + \lambda I = \begin{bmatrix} 1+\lambda & 1 & 0 & 0 \\ 1 & 1+\lambda & 1 & 0 \\ 0 & 1 & \lambda & 1 \\ 0 & 0 & 1 & 1+\lambda \end{bmatrix}$$

O polinômio característico associado à matriz característica exemplificada acima é $\lambda^4 + \lambda^3 + 1$. Se o polinômio característico do AC é primitivo ou irredutível (coeficientes não fatoráveis), então ele é referenciado como AC de duração máxima (*maximal length CA*) [Nandi 1994]. Os ACs de duração máxima possuem ciclos ininterruptos percorrendo todos os estados possíveis para o tamanho do reticulado, exceto o estado corresponde ao zero. Se um AC possui N células, então existem 2^N estados possíveis, sendo assim, os AC de duração máxima possuem ciclo de tamanho $2^N - 1$.

As principais técnicas de criptografia que empregam autômatos celulares aditivos serão apresentadas na seção 4.2. Maiores detalhes referentes aos ACs aditivos podem ser encontrados em [Chaudhuri *et al*. 1997].

3.4 Sensitividade das regras

Algumas regras de ACs possuem uma característica conhecida como sensitividade. Uma regra é dita sensível a uma célula específica da vizinhança, se uma alteração no estado dessa célula provoca a alteração do estado da célula central, considerando todas as

possibilidades de vizinhança da regra. Como exemplo, considere um AC binário unidimensional. Uma regra de transição pode ser sensível à esquerda se $\phi\left(s_{i-1}^t, s_i^t, s_{i+1}^t\right) \neq \phi\left(\overline{s_{i-1}^t}, s_i^t, s_{i+1}^t\right), \ \forall s_{i-1}^t \neq \overline{s_{i-1}^t}$; à direita se $\phi\left(s_{i-1}^t, s_i^t, s_{i+1}^t\right) \neq \phi\left(s_{i-1}^t, s_i^t, \overline{s_{i+1}^t}\right), \ \forall s_{i+1}^t \neq \overline{s_{i+1}^t}$; em ambas as direções se $\phi\left(s_{i-1}^t, s_i^t, s_{i+1}^t\right) \neq \phi\left(\overline{s_{i-1}^t}, s_i^t, s_{i+1}^t\right) \wedge \phi\left(s_{i-1}^t, s_i^t, s_{i+1}^t\right) \neq \phi\left(s_{i-1}^t, s_i^t, \overline{s_{i+1}^t}\right), \forall s_{i-1}^t \neq \overline{s_{i-1}^t} \wedge s_{i+1}^t \neq \overline{s_{i+1}^t}$, onde "$s$" corresponde ao estado 0 ou 1 de uma célula no reticulado, "$\bar{s}$" é o complemento de "s", "t" indica o tempo, "i" representa o índice da célula no reticulado, e "ϕ" é a função que faz o mapeamento da vizinhança para o bit de saída.

A Figura 18 apresenta um exemplo de uma regra elementar com sensitividade (a) à direita, (b) à esquerda, e (c) bidirecional. Note que os bits de saída da Figura 18 (a) sofrem influência direta do bit direito da vizinhança. Dessa forma, considere o par de vizinhanças 000 e 001 (agrupadas na figura como par 1): a vizinhança 000 tem como bit de saída 0 e alterando-se o bit mais à direita da vizinhança temos 001, que possui o bit de saída 1. Esta análise deve ser feita para todos os pares de vizinhança restantes, 2, 3 e 4. Se o bit de saída para todos estes pares de vizinhança são complementares, então a regra é sensível à direita. Da mesma forma, esta análise deve ser feita na Figura 18 (b), porém o bit da vizinhança a ser analisado é o bit esquerdo, portanto, os pares a serem analisados são: par 1 (000 e 100), par 2 (001 e 101), par 3 (010 e 110) e, por fim, o par 4 (011 e 110). Vemos que na Figura 18 (b) todos os pares são sensíveis ao bit da esquerda. As regras bidirecionais possuem as duas características citadas anteriormente, um exemplo desse tipo de regra é apresentado na Figura 18 (c), onde os pares marcados com pontilhado representam os pares sensíveis à direita e os demais pares representam a sensitividade à esquerda.

Para os ACs elementares, existem dezesseis regras com sensitividade à esquerda e dezesseis regras com sensitividade à direita. Dentre estas regras, quatro delas possuem sensitividade bidirecional. A Tabela 10 mostra todas as regras elementares com sensitividade a alguma das extremidades (ou a ambas). A tabela também apresenta a classificação dinâmica e as propriedades algébricas dessas regras. A maioria das regras com sensitividade a pelo menos uma das extremidades possui comportamento caótico, com exceção das regras sensíveis à direita 85, 154, 166 e 170, e das regras sensíveis à esquerda 15, 180, 210 e 240. De fato, o comportamento dinâmico mais provável de uma regra de AC com sensitividade exibir é o caótico. Wuensche (1999) propôs um indicador para previsão do comportamento dinâmico de uma regra de AC chamado de parâmetro Z. Uma das principais conclusões do autor sobre esse parâmetro, é a de que regras com Z próximo ou igual a 1 têm grande probabilidade de

exibir uma dinâmica caótica. Regras com sensitividade a pelo menos uma das células extremas têm Z igual a 1, independentemente do raio do AC. Portanto, o comportamento dinâmico mais provável desse tipo de regra é o caótico. Esta alta tendência das regras com sensitividade a exibir um comportamento caótico também é fundamental no método criptográfico investigado, pois essa característica faz com que o AC propague qualquer perturbação em um reticulado inicial. Dessa forma, a cifragem de dois textos extremamente similares gera dois textos completamente distintos. Maiores detalhes sobre o cálculo do parâmetro Z podem ser obtidos em [Wuensche 1999] e [Lima 2005].

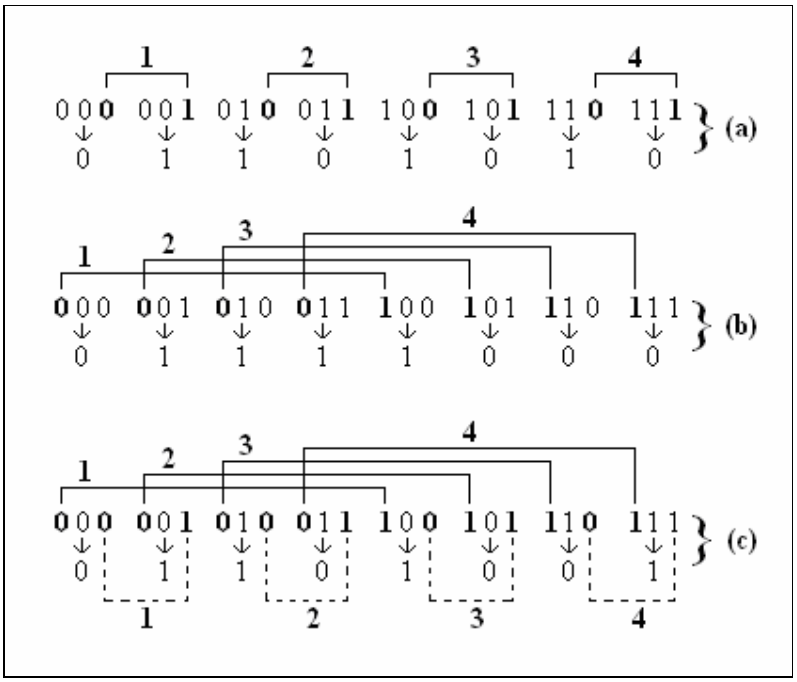


Figura 18 - Regras com sensitividade à direita (a), à esquerda (b) e bidirecional (c).

A sensitividade à esquerda ou à direita de uma regra possui um papel de extrema importância no método de criptografia investigado nesta dissertação, pois essa característica faz com que a evolução de um AC para trás seja possível.

3.5 ACs reversíveis e irreversíveis

Embora seja possível definir ACs com regras de transição probabilistas, os modelos de ACs mais comuns são os deterministas na evolução temporal, obtida pela aplicação da regra de transição Φ a cada passo de tempo. Essa evolução temporal será chamada a partir daqui de evolução "para frente" do AC. Outra evolução possível de se analisar é a evolução "para trás", onde a partir de um reticulado inicial s_t, obtém-se o reticulado s_{t-1} que teria dado origem a

s_t, com a aplicação da regra Φ. Esse reticulado anterior s_{t-1} é chamado de pré-imagem de s_t.

Tabela 10- Regras com sensibilidade a pelo menos uma das extremidades.

Regra	**Decimal**	**Comportamento Dinâmico**	**Propriedade Algébrica**	**Sensitividade**
11110000	15	Ciclo duplo com deslocamento	Complementar	Esquerda
01111000	30	Caótico	Não-aditivo	Esquerda
10110100	45	Caótico	Não-aditivo	Esquerda
00111100	60	Caótico	Linear	Esquerda
11010010	75	Caótico	Não-aditivo	Esquerda
10101010	85	Ciclo duplo com deslocamento	Complementar	Direita
01101010	86	Caótico	Não-aditivo	Direita
10011010	89	Caótico	Não-aditivo	Direita
01011010	90	Caótico	Linear	Bidirecional
10100110	101	Caótico	Não-aditivo	Direita
01100110	102	Caótico	Linear	Direita
10010110	105	Caótico	Complementar	Bidirecional
01010110	106	Caótico	Não-aditivo	Direita
00011110	120	Caótico	Não-aditivo	Esquerda
11100001	135	Caótico	Não-aditivo	Esquerda
10101001	149	Caótico	Não-aditivo	Direita
01101001	150	Caótico	Linear	Bidirecional
10011001	153	Caótico	Complementar	Direita
01011001	154	Periódico	Não-aditivo	Direita
10100101	165	Caótico	Complementar	Bidirecional
01100101	166	Periódico	Não-aditivo	Direita
10010101	169	Caótico	Não-aditivo	Direita
01010101	170	Ponto fixo com deslocamento	Linear	Direita
00101101	180	Periódico	Não-aditivo	Esquerda
11000011	195	Caótico	Complementar	Esquerda
01001011	210	Periódico	Não-aditivo	Esquerda
10000111	225	Caótico	Não-aditivo	Esquerda
00001111	240	Ponto fixo com deslocamento	Linear	Esquerda

Um AC é considerado irreversível se algum reticulado possuir mais de uma ou nenhuma pré-imagem, de acordo com sua regra de transição. Dessa forma, existem algumas configurações de reticulado que são inalcançáveis por uma evolução para frente. Esses reticulados são chamados de "jardim do Éden" A Figura 16 apresenta um exemplo de AC irreversível. Neste caso, os estados "jardim do Éden" (1, 4, 11, 14, 2, 7, 8, 13, 3, 6, 9 e 12) só podem surgir como reticulados iniciais, por não possuírem pré-imagens.

Os ACs reversíveis são aqueles deterministas também na evolução para trás. Nesse modelo de AC, cada configuração do reticulado possui exatamente uma única pré-imagem. A Figura 17 mostra um exemplo de AC reversível. Tanto modelos de ACs reversíveis quanto

modelos de ACs irreversíveis foram aplicados anteriormente à criptografia. Alguns desses modelos serão apresentados no Capítulo 4.

3.6 Algoritmo de cálculo de pré-imagem proposto por Wuensche e Lesser

Dado um reticulado inicial, o cálculo de pré-imagem tenta encontrar a configuração de um reticulado anterior ao reticulado inicial, ou seja, encontrar o reticulado que gerou o reticulado inicial, dada uma regra de transição. Porém, existem algumas configurações de reticulados iniciais que não possuem uma pré-imagem.

Um algoritmo para o cálculo de pré-imagens de ACs foi proposto por Wuensche e Lesser (1992). Esse algoritmo utiliza os conceitos de vizinhança parcial determinista, impossível e ambígua.

Para dar início ao cálculo de pré-imagem é necessário definir alguns bits da pré-imagem (bits iniciais). No cálculo de pré-imagem de Wuensche e Lesser, os bits iniciais são escolhidos de forma aleatória. Após escolhidos os bits iniciais, que no total correspondem a $2 \times raio$, é então verificado qual bit (0 ou 1) poderia completar a vizinhança, dado o bit de saída da regra de transição, surgindo então os conceitos de vizinhança parcial determinística, impossível e ambígua.

A vizinhança parcial determinística ocorre quando a formação da vizinhança é determinada de forma única pela regra de transição e pelo bit do reticulado inicial que está sendo analisado. A Figura 19 exemplifica a situação em que a vizinhança parcial da pré-imagem é formada pelas células 1 e 2 preenchidas com {00} e o bit do reticulado inicial {0} (célula 2). O caractere interrogação {?} representa o valor da célula 3 da pré-imagem que se deseja identificar. A regra de transição empregada é formada pelos bits {01001011}. Em destaque na regra de transição da Figura 19 está a única vizinhança que poderia fazer parte dos bits da pré-imagem. Dessa forma, a interrogação {?} é preenchida com o bit {0}, marcado em negrito na figura (célula 3). Veja que o bit {1} não pode fazer parte da vizinhança {00} porque a regra indica que para vizinhança {001} o bit de saída é {1}, que é diferente do bit indicado no reticulado inicial (célula 2).

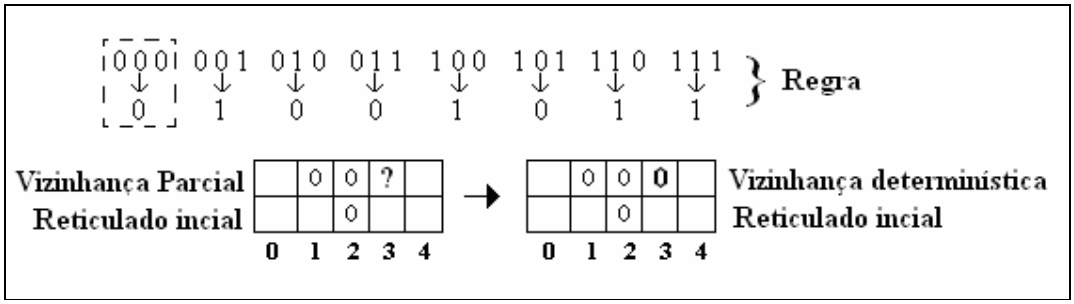


Figura 19 - Vizinhança parcial determinística.

Dada uma vizinhança parcial, é possível que a regra de transição não tenha uma vizinhança que atenda o bit estabelecido pelo reticulado inicial, formando então uma vizinhança parcial impossível. A Figura 20 mostra o caso em que dado os bits da vizinhança parcial {01} (célula 1 e 2), e o bit do reticulado inicial {1} (célula 2) juntamente com a regra de transição {01001011}, é impossível de determinar o próximo bit (célula 3) da vizinhança completa da pré-imagem.

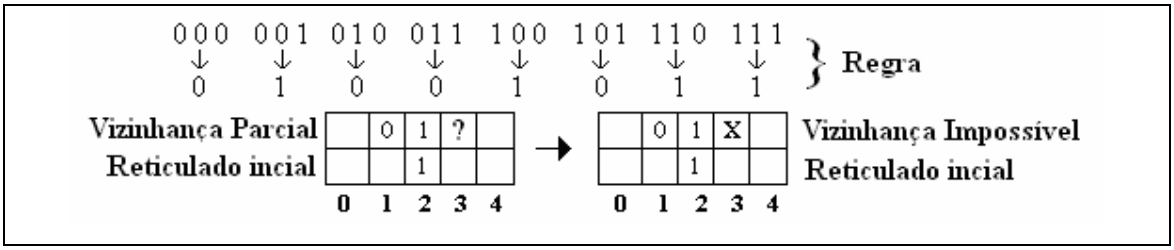


Figura 20 - Vizinhança parcial impossível.

A vizinhança parcial ambígua ocorre quando a última célula da vizinhança da pré-imagem parcial pode ser composta com qualquer um dos bits (0 ou 1), dado o reticulado inicial e a regra de transição. A Figura 21 exemplifica este caso, onde a vizinhança parcial da pré-imagem é formada pelos bits {01} (célula 1 e 2) e o bit do reticulado inicial {0} (célula 2). A regra de transição é a mesma dos exemplos anteriores. Note neste caso, que qualquer um dos bits pode ser empregado como vizinhança completa da pré-imagem, de acordo a regra de transição.

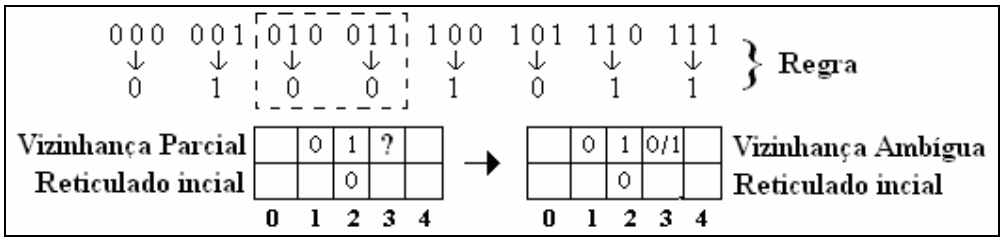


Figura 21 - Vizinhança parcial ambígua.

Os exemplos de vizinhança parcial, citados acima, foram elaborados da esquerda para direita, ou seja, dado os bits da vizinhança parcial, é verificada a possibilidade de encontrar o próximo bit direito da vizinhança. Porém, pode-se executar o mesmo procedimento da direita para esquerda, onde o bit a ser encontrado pela regra de transição passa a ser o bit esquerdo da vizinhança completa da pré-imagem.

No algoritmo proposto por Wuensche e Lesser, a pré-imagem é calculada da célula mais à esquerda do reticulado até a mais à direita, verificando-se a cada passo o tipo de vizinhança. O algoritmo utiliza bits extras nos dois lados do reticulado referentes à pré-imagem, conforme exemplifica a Figura 22. No caso do AC possuir condição de contorno periódica, os bits extras passam por uma verificação e são descartados ao final do cálculo, se a pré-imagem for encontrada.

Para iniciar o algoritmo do cálculo de pré-imagem, as células mais à esquerda do reticulado são iniciadas aleatoriamente (bits iniciais). Dessa forma, uma das três vizinhanças (determinista, ambígua ou impossível) será encontrada.

Se o bit a ser calculado formar uma vizinhança determinística, o bit referente à vizinhança da regra é copiado para o bit que está sendo calculado.

Se a vizinhança do bit a ser calculado for ambígua, o método de Wuensche e Lesser escolhe um dos dois bits para seguir a seqüência do cálculo de pré-imagem, além de armazenar a célula e o bit não utilizado em uma pilha, e então, o cálculo segue para encontrar o próximo bit da pré-imagem.

Se a vizinhança é impossível, como não existe o próximo bit para aquela seqüência de pré-imagem, o algoritmo retorna no reticulado e um elemento da pilha será retirado. Uma vez que foi retirado o elemento da pilha, contendo a célula e o bit de alguma vizinhança ambígua encontrada anteriormente, o processo volta para a célula indicada pelo elemento removido da pilha utilizando o bit que consta no nó removido. O cálculo segue então a partir da célula alterada.

Dessa forma, o método de Wensche e Lesser testa todas as possibilidades de pré-imagem para o reticulado em questão. Ou seja, se o reticulado possuir pelo menos uma pré-imagem, esta pré-imagem será encontrada, mas é bom lembrar que dependendo da regra e reticulado a ser analisado, nem sempre é possível encontrar uma pré-imagem (jardim do Éden).

Se o AC possuir condição de contorno periódica os bits extras precisam ser testados

para que a pré-imagem que está sendo calculada seja válida. Para exemplificar a validação efetuada na região de contorno, considere um AC binário, unidimensional e um reticulado de oito bits, conforme mostra a Figura 22, tal que a regra empregada neste exemplo é a mesma descrita na Figura 21. O bit P0 necessariamente precisa ser igual a P8, assim como P1 precisa ser igual a P9. Neste exemplo a condição de contorno é satisfeita, mas caso esta restrição não seja atendida, um nó da pilha é removido, e então o cálculo segue a partir da célula indicada pelo nó removido da pilha.

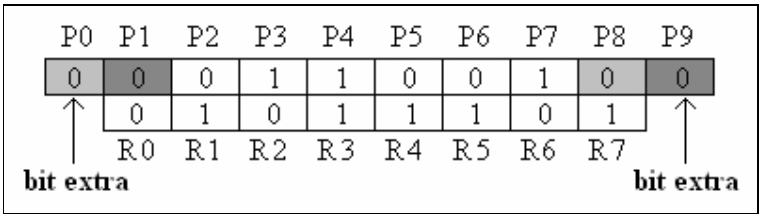


Figura 22 - Pré-imagem destacando a região de contorno P0 = P8 e P1 = P9.

Para executar o calculo por vários passos de tempo, o reticulado referente à pré-imagem que acabou de ser calculada passa a ser utilizado para que outra pré-imagem seja encontrada. O processo continua por quantos passos de tempo se desejar, realizando a evolução "para trás" do reticulado. A Figura 23 ilustra o cálculo consecutivo de pré-imagem sendo executado por 10 passos de tempo, onde as pré-imagens são calculadas de baixo para cima. O reticulado inicial é o mesmo apresentado na Figura 22 e a regra de transição é igual à utilizada no exemplo da Figura 21.

```
P[10] 01100001
P[9]  00110010
P[8]  01011101
P[7]  00001100
P[6]  00010110
P[5]  00100011
P[4]  11010101
P[3]  11000000
P[2]  01100001
P[1]  00110010
R-->  01011101
```

Figura 23 - Cálculo de pré-imagem executado por 10 passos de tempo.

Podem ocorrer casos em que a pilha está vazia e a condição de contorno não é atendida ou uma vizinhança impossível é encontrada. Nestes casos simplesmente não existe uma pré-imagem para o reticulado que está sendo calculado.

A restrição imposta pela região de contorno periódica, onde os bits extras necessitam ser validados, faz com que apenas um pequeno número de regras seja capaz de encontrar alguma pré-imagem, para qualquer que seja a configuração do reticulado inicial. Contudo, Lima (2005) investigou se é possível utilizar o cálculo de pré-imagem de Wuensche e Lesser em um sistema criptográfico. Detalhes desta investigação são apontados na seção 4.5.

Capítulo 4

Autômatos celulares aplicados a criptografia

Neste capítulo, serão apresentados os principais métodos criptográficos encontrados na literatura baseados em autômatos celulares. Serão abordados tanto métodos baseados em ACs reversíveis quanto aqueles que utilizam ACs irreversíveis. Dentre os métodos, existem sistemas criptográficos de blocos e de fluxos, porém será dado um enfoque maior nos modelos de mensagens cifradas em blocos, uma vez que o tema aqui proposto envolve a criação de um sistema criptográfico de blocos.

4.1 Modelo baseado em uma regra de AC não-linear [Wolfram 1986]

Wolfram (1986) foi o primeiro a sugerir a utilização de ACs em criptografia. O modelo proposto por Wolfram se classifica na categoria de cifragem por fluxo (*stream cipher*). A idéia principal é utilizar um autômato celular binário unidimensional com condições de contorno periódica, capaz de gerar seqüências de bits o mais aleatório o possível, dado um reticulado inicial de tamanho *N*. A chave criptográfica deste sistema criptográfico são os *N* bits que correspondem ao reticulado inicial.

Os valores do reticulado são atualizados de forma síncrona em passos de tempo discreto de acordo com a regra elementar 30. Esta regra é não-aditiva e tem uma dinâmica classificada como caótica. A regra 30 pode ser descrita com combinações lógicas como mostra a equação (8).

$$s_{t+1}^{i} = s_{t}^{i-1} \oplus \left(s_{t}^{i} \vee s_{t}^{i+1} \right) \qquad (8)$$

Wolfram, utilizando várias medidas estatísticas, mostrou que os bits obtidos a cada passo de tempo *t*, por uma determinada célula *i* (s_t^i), formam uma seqüência de bits aleatória. Essa seqüência pode ser aplicada bit a bit a um texto claro *P* com a operação XOR, produzindo o texto cifrado *C*. Suponha que o texto claro tenha tamanho *N* e a seqüência temporal da célula na posição *i* seja utilizada para gerar a seqüência s_t^i para $t = 1, \cdots, N$. Assim, a criptografia é realizada através da equação (9).

$$C_j = P_j \oplus s_j^i \text{, para } j \text{ variando de 1 a } N. \qquad (9)$$

Por exemplo, a Figura 24 mostra um AC com reticulado inicial de tamanho 25 evoluindo por 15 passos de tempo com a regra 30 sugerida por Wolfram. Os bits representados pela célula 10 (i=10) estão em destaque indicando uma possível seqüência que poderia ser aplicada na operação XOR junto com os bits do texto claro de tamanho 15. O texto claro P pode ser recuperado ao aplicar a operação XOR entre o texto cifrado C e a mesma seqüência obtida em s_t^i, para i = 10.

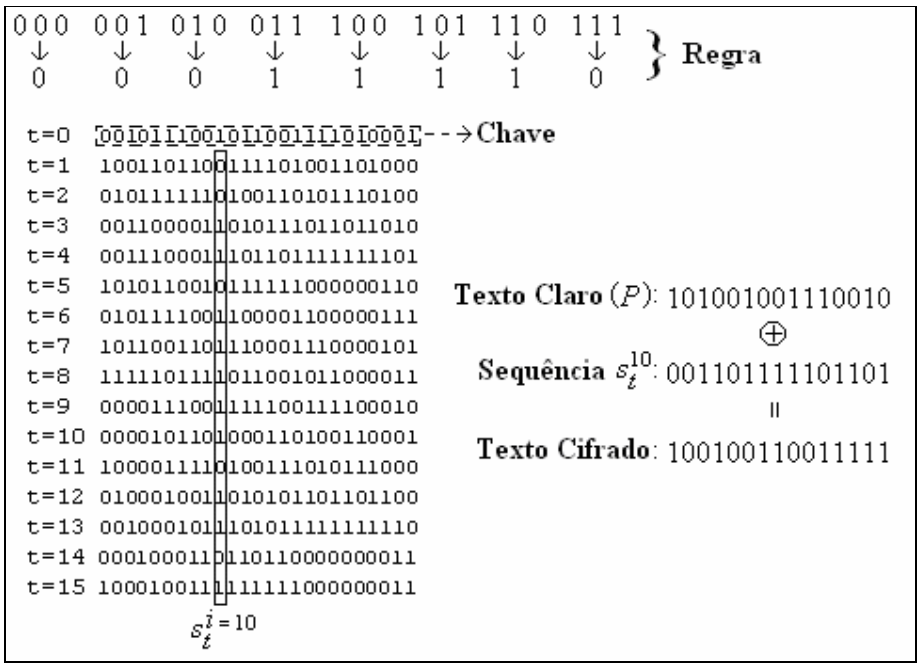


Figura 24 - Evolução de um reticulado inicial com a regra 30 por 15 passos de tempo.

A força deste método se baseia na dificuldade de um criptoanalista encontrar o reticulado inicial (chave) capaz de produzir a seqüência de bits correta. Portanto, neste método, a evolução temporal dos ACs é usada apenas para gerar uma seqüência suficientemente aleatória para ser utilizada em uma cifragem com o texto claro e não para realizar a cifragem propriamente dita, que nesse caso é feita pela operação XOR.

4.2 Modelo utilizando ACs aditivos, não-homogêneos e reversíveis.

A seção 3.3 definiu as principais propriedades algébricas envolvendo ACs aditivos. Nesta seção, será brevemente descrito um modelo de ACs que utiliza propriedades de grupo aplicadas à criptografia.

A maioria dos modelos de sistemas criptográficos que utilizam ACs encontrados na literatura são baseados nas propriedades algébricas dos ACs aditivos. Um destes sistemas, conhecido como CAC (*Cellular Automata Cryptosystem*) foi proposto por Nandi, Kar e Chaudhuri (1994), e com o passar do tempo sofreu algumas modificações propostas por Ganguly e colaboradores (2000), chegando então, ao sistema criptográfico proposto por Sen e colaboradores (2002).

O modelo de Nandi, Kar e Chaudhuri (1994) utiliza ACs aditivos de grupo com ciclo de duração não-máxima (seção3.3) com *n* ciclos de tamanho *x*, onde *x* é par e é uma potência de 2 (2^x). Em [Chaudhuri *et al*. 1997] foi demonstrando que qualquer arranjo das regras elementares 51, 153 e 195 com condição de contorno nula, gera ACs não-homogêneos que possuem a propriedade acima. Essa característica é aproveitada para elaborar um método de criptografia que utiliza p funções de cifragem $E_i, i = 1,..., p$. Cada uma dessas funções é composta por q transformações fundamentais, sendo que uma transformação fundamental corresponde a um arranjo entre as regras elementares 51, 153 e 191, formando assim um vetor de regras de tamanho *N*. Por exemplo: considere as transformações fundamentais *s, y, z, u, v* ($q = 5$) para um vetor de regras formado por oito células ($N = 8$) e com condição de contorno nula:

s = (153, 153, 153, 153, 51, 51, 51, 51)

y = (195, 195, 195, 195, 51, 51, 51, 51)

z = (51, 51, 153, 153, 153, 153, 51, 51)

u = (51, 51, 195, 195, 195, 195, 51, 51)

v = (51, 153, 153, 153, 153, 153, 153, 51)

Cada uma destas transformações geram ciclo de tamanho 8. Dessa forma, o processo de cifragem é alcançado por evoluir uma configuração do reticulado inicial *M* (correspondente ao texto claro) por 4 passos de tempo, para cada transformação fundamental. Dessa forma temos que o texto cifrado *C* é obtido por: $C = (v(u(z(y(s(M))))))$. No processo de decifragem, basta evoluir cada uma das transformações fundamentais a partir da configuração do reticulado *C* por mais 4 passos de tempo na ordem inversa, assim: $M = (s(y(z(u(v(C))))))$. Neste caso, tanto *M* quanto *C* são formados por oito bits. Uma vez que as transformações fundamentais possuem tamanho de ciclo igual a 8, a configuração do reticulado inicial será encontrada novamente após a execução do processo de decifragem.

Dessa forma, o bloco de texto M_0 é cifrado pela função de cifragem E_0. De forma geral o bloco M_i é cifrado por uma função $E_{i \bmod p}$. Para este modelo, a chave criptográfica corresponde às *p* funções de cifragem, e o bloco de texto claro é um reticulado inicial *N* que será evoluído de acordo com estas funções.

O modelo de Nandi, Kar e Chaudhuri (1994) é baseado em permutações no espaço de vetores V_N sobre GF(2), e foi criticado por Blackburn, Merphy e Paterson (1997). Segundo os autores, as permutações efetuadas são do tipo *affine*, um pequeno sub grupo no espaço de vetores V_N, onde todas as funções E_i geradas pelo método produzem a propriedade *affine group*, e podem ser quebradas facilmente por um ataque de texto claro escolhido ou um ataque de texto claro conhecido.

O método proposto por Ganguly (2000) e a alteração feita por Sen (2002) são apresentadas com um pouco mais de detalhes a seguir.

Uma nova versão do sistema criptográfico CAC, pertencente aos modelos de cifragem de blocos, foi proposta por Ganguly (2000). Para cifrar e decifrar um texto este modelo emprega ACs reversíveis operando sobre GF(2^p), que é uma generalização de GF(2) no sentido de que cada célula é capaz de processar um símbolo de $\{0,1,\cdots 2^p-1\} \in$ GF(2^p).

Neste método dois modelos de ACs aditivos baseados em grupo são utilizados. Um deles, denominado *Major CA,* é formado por um AC aditivo de ciclo não máximo possuindo tamanho de ciclo igual a 32. O outro, chamado de *Minor CA,* é composto por um AC aditivo de ciclo máximo. Tanto o *Major CA* quanto *Minor CA* possuem 16 células, e o estado de cada célula é representado por 8 bits. Assim, os ACs operam sobre GF(2^8), podendo então cifrar um bloco de texto de tamanho $16 \times 8 = 128$ bits a cada execução do método. O texto claro e a chave criptográfica possuem 128 bits, produzindo um texto cifrado de tamanho também igual a 128 bits.

As operações executadas no processo de cifragem e decifragem são apresentadas na Figura 25 e explicadas a seguir.

Geração da chave: A chave (*K*) é utilizada como semente para a execução do *Minor CA*. A chave é formada por uma string de bits com o mesmo número de bits usado em *Minor CA*, ou seja, 128 bits.

Papel do *Minor CA*: O *Minor CA* é um AC periódico de ciclo máximo. Nesse caso,

como o reticulado tem 128 bits, o ciclo do *Minor CA* é $2^{128}-1$. O AC é executado por um número fixo de passos (*d*) para cada bloco de texto de entrada. O reticulado inicial, ou semente, do *Minor CA* (S_0) é a chave *K* para o primeiro bloco cifrado, marcada como I na Figura 25. Para cada bloco de texto sucessivo, *Minor CA* gera um novo reticulado (marcado como S_N) após evoluir o reticulado por *d* número de passos a partir do reticulado atual (mostrado como II na Figura 25). Portanto, para cada bloco de texto é utilizado um reticulado S_N diferente, evoluído a partir da configuração do reticulado S_N utilizado no bloco anterior. O reticulado S_N é utilizado para quatro diferentes propósitos:

- Prover o valor de x, que determina a quantidade de deslocamentos a ser executada nos bits do bloco de entrada;
- Prover a semente para a especificação do *Major CA*;
- Prover o número de ciclos (Δ) a ser executado pelo *Major CA* no processo de cifragem;
- É submetido à operação XOR com o texto intermediário (T_2);

Bloco de texto e seu deslocamento: Inicialmente o bloco de texto claro é submetido a um deslocamento por x posições resultando em T_1. O valor de x é derivado de *p* bits selecionados a partir do reticulado resultante do *Minor CA* (S_N). A operação inversa é executada no processo de decifragem, ou seja, o bloco de texto gerado pelo *Major CA* é submetido à mesma quantidade de deslocamentos na direção oposta à executada no processo de cifragem.

Papel do *Major CA*: O *Major CA* é selecionado a partir de um banco de dados composto por regras de ACs com ciclo de tamanho 32. Esta seleção é feita utilizando o reticulado final do *Minor CA* (S_N) como entrada. O Major CA utiliza o bloco de texto T_1 (resultante do deslocamento do bloco de texto claro) como semente e é executado por Δ número de passos para gerar o texto cifrado T_2 (IV na Figura 25). O *Major CA* invariavelmente retorna ao bloco de texto T_1 após ser executado por 32 números de passos. Sendo assim, no ato da cifragem, o reticulado do Major CA referente ao texto intermediário (T_2) é obtido após Δ número de passos. No ato da decifragem, o bloco de texto T_2 resulta no bloco de texto T_1 ao ser executar por $32-\Delta$ número de passos. O número de passos (Δ) de *Major CA* é obtido a partir do reticulado final do *Minor CA* (S_N).

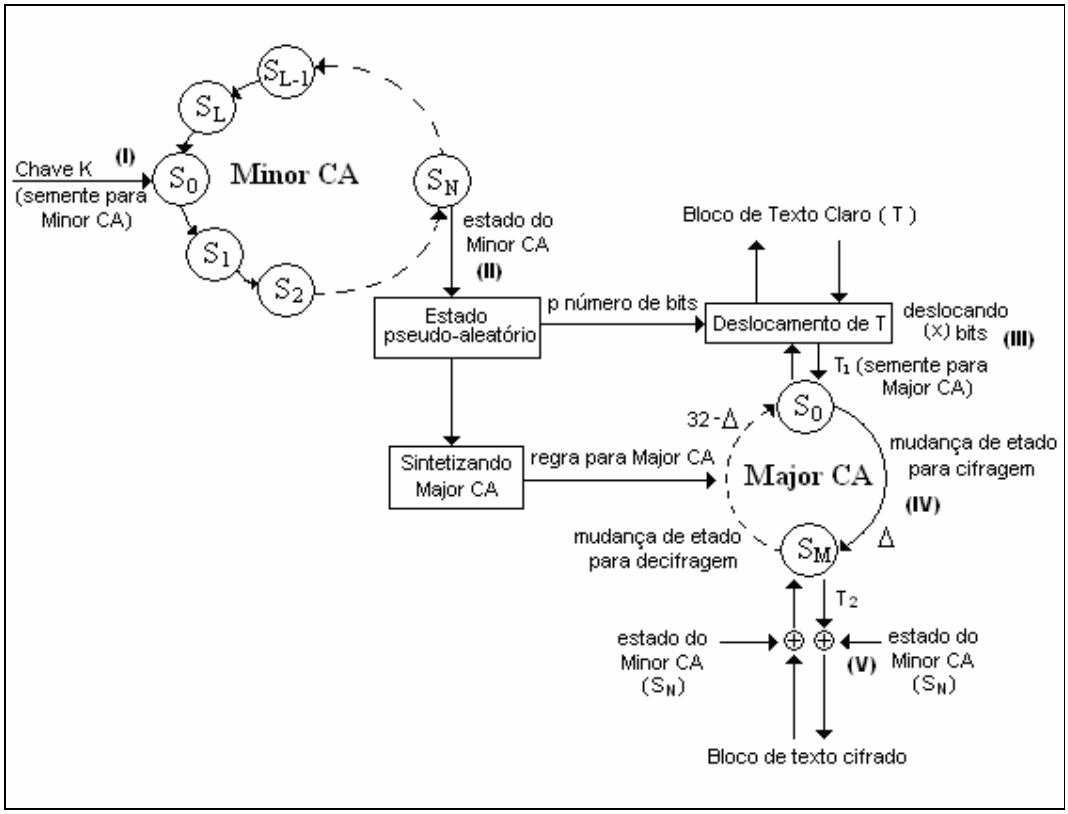


Figura 25 - Esquema de cifragem e decifragem no modelo de Ganguly (2000).

Operação XOR entre o bloco de texto e o reticulado final do *Minor CA*: O texto cifrado é obtido ao executar a operação XOR entre o reticulado final do *Minor CA* (S_N) e do bloco de texto T_2, correspondente ao reticulado resultante do *Major CA* (V na Figura 25). Na decifragem, a operação XOR é executada entre o bloco de texto cifrado e o reticulado final do *Minor CA* (S_N), e então, o resultado desta operação é utilizado como reticulado inicial para evoluir $32-\Delta$ passos no *Major CA*.

Sen (2002) propôs uma alteração no método de Ganguly (2000) adicionando uma etapa a mais, responsável por eliminar uma fraqueza encontrada no método anterior. O problema no método de Ganguly (2000) está na propriedade conhecida como *affine property* encontrada nos ACs que têm ciclos não máximo de tamanho par, como o caso do M*ajor CA*. A adição de mais uma etapa faz com que AC com *affine property* seja misturado com *nonaffine transformations*, para gerar o bloco de texto cifrado. A etapa adicionada emprega o uso de uma porta lógica denominada *Control Majority Not* (CMN). CMN é uma porta lógica não linear reversível com quatro entradas (1 de dados e 3 de controles) e uma saída. A função é definida pela equação (10), sendo que "$\oplus$" corresponde à operação XOR, "$\wedge$" representa a operação lógica AND e c_1, c_2 e c_3, denotam os bits de controle que são extraídos de S_N.

$$y = x \oplus \{(c_1 \wedge c_2) \oplus (c_2 \wedge c_3) \oplus (c_3 \wedge c_1)\} \quad (10)$$

Bao (2004) analisou o sistema proposto por Sen e colaboradores e constatou que mesmo com a utilização de uma nova etapa o sistema criptográfico possui algumas fraquezas e pode ser quebrado com a técnica de criptoanálise conhecida como texto claro escolhido (*chosen plan text*) [Bao 2004].

4.3 Modelo utilizando ACs homogêneos irreversíveis [Gutowitz, 1995]

O sistema criptográfico proposto por Gutowitz (1995) é baseado na teoria dos sistemas dinâmicos e pertence aos modelos de cifragem de blocos. Este modelo foi desenvolvido para ser implementado em circuitos integrados trabalhando de forma paralela.

Ao contrário do modelo proposto por Wolfram (seção 4.1), que utiliza apenas a evolução de um AC para frente, gerando bits de forma pseudo-aleatória por um sistema dinâmico (a evolução de um AC com a regra 30), o modelo de Gutowitz emprega tanto a evolução para frente, quanto a evolução para trás, sendo que o próprio sistema dinâmico é utilizado para cifrar e decifrar uma mensagem. Dessa forma, a chave do sistema criptográfico não é o reticulado inicial, como no modelo de Wolfram, mas sim a regra de transição do AC. Essa regra é utilizada na evolução do reticulado inicial, que corresponde a um bloco da mensagem a ser cifrada.

Gutowitz utilizou tanto ACs reversíveis quanto irreversíveis em seu sistema criptográfico. Entretanto, a etapa principal de cifragem é realizada por ACs irreversíveis que possuem a característica conhecida como sensitividade. Regras com sensitividade a alguma das extremidades ou em ambas as direções (bidirecionais) foram definidas na seção 3.4. Essas regras são utilizadas no sistema criptográfico devido à sua capacidade de sempre gerar pré-imagens em um método de cálculo proposto pelo próprio Gutowitz.

Para cifrar e decifrar um bloco de texto no modelo de Gutowitz são necessárias duas etapas, cada uma delas divididas em uma sub-etapa esquerda e uma sub-etapa direita. Cada sub-etapa é composta por duas fases: difusão e substituição.

4.3.1 Fase de difusão

Na fase de difusão, ACs irreversíveis são empregados efetuando-se um cálculo de pré-

imagens específico para o modelo criptográfico de Gutowitz. As regras utilizadas no cálculo possuem a característica de sensitividade a alguma das extremidades. A fase de difusão da sub-etapa esquerda emprega regras sensíveis à esquerda. Assim, na sub-etapa direita são empregadas regras sensíveis à direita. Para facilitar o entendimento, essa fase será explicada usando um AC elementar, mas a idéia é similar para ACs de raio maior.

Considere um AC unidimensional binário de raio 1, onde a vizinhança possui tamanho três. A evolução para frente é definida como $s_{t+1}^{i} = \phi\left(s_t^{i-1}, s_t^{i}, s_t^{i+1}\right)$. A evolução para trás, na obtenção da pré-imagem, necessita que alguns bits iniciais sejam escolhidos para que a regra seja aplicada, assim como no método de Wuensche e Lesser, apresentado na seção 3.6. Entretanto, quaisquer que sejam os bits iniciais escolhidos, o restante da pré-imagem é calculado de forma determinista, sem a possibilidade do cálculo ser interrompido por uma vizinhança parcial impossível, devido à sensitividade da regra. Como a quantidade de bits iniciais é de duas vezes o tamanho do raio da regra, a cada passo do calculo de pré-imagem é possível termos $2^{2\times raio}$ pré-imagens diferentes.

A Figura 26 mostra um exemplo de cálculo de pré-imagem. Note-se que a pré-imagem está sendo calculada da esquerda para direita. Este fato ocorre porque a regra utilizada é sensível à direita. Note-se também, que regras sensíveis à direita são sempre deterministas ao definir o bit mais à direita da vizinhança. De forma contrária, regras sensíveis à esquerda são calculadas da direita para a esquerda.

A Figura 26 (a) indica os dois bits iniciais escolhidos de forma aleatória nas células do lado esquerdo da pré-imagem. Na Figura 26 (b), o próximo bit foi definido pela regra de transição a partir dos bits iniciais. Cada bit pode ser definido deterministicamente devido à propriedade da regra. Ou seja, não existe a possibilidade de surgir uma vizinhança ambígua ou impossível e é sempre possível concluir o cálculo da pré-imagem, como na Figura 26 (c). Entretanto, ao contrário do método visto na seção 3.6, os bits extras não são validados e nem descartados ao final do cálculo, pois o reticulado não possui condição de contorno periódica. Essa característica faz com que a pré-imagem sempre possa ser obtida a partir de qualquer configuração de bits iniciais, mas causa um aumento no reticulado da pré-imagem, igual ao número de bits iniciais.

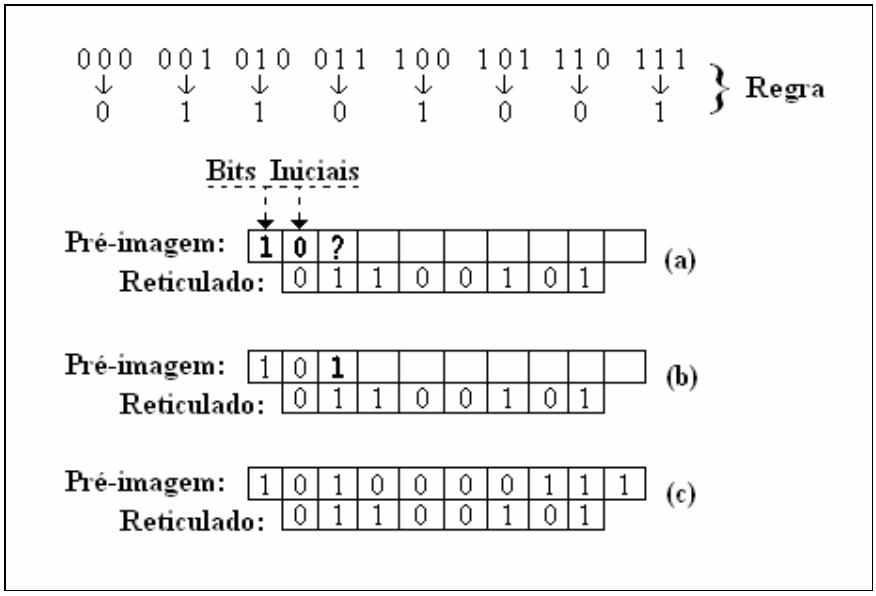


Figura 26 - Cálculo passo a passo de apenas uma pré-imagem utilizando uma regra sensível à direita.

Esse processo pode ser empregado sucessivamente, por quantos passos se desejar. Seja P o número de passos, N o tamanho do reticulado inicial e r o raio da regra. O método adiciona $2\times r$ bits a cada pré-imagem e o tamanho do reticulado final é dado por $(2\times r\times P)+N$. Por exemplo, se fossem calculadas 10 pré-imagens (P = 10) utilizando uma regra de raio 1 (r = 1) a partir de um texto claro de 10 bits (N = 10), a pré-imagem final seria formada por 30 bits, conforme apresentado na Figura 27.

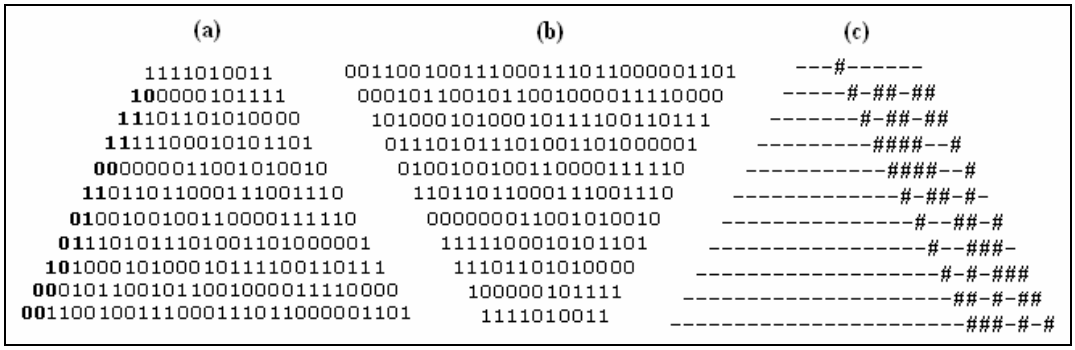


Figura 27 - Método proposto por Gutowitz (a) Cifragem a partir de um texto claro (b) Decifragem a partir do texto cifrado (c) Diferenças provocadas pela perturbação.

A Figura 27 (a) apresenta uma configuração de reticulado inicial aleatória que representa um texto claro e suas respectivas pré-imagens calculadas consecutivamente de cima para baixo, pela regra sensível à direita {10101001}. Os bits em negrito são os bits iniciais escolhidos de forma aleatória. O último reticulado obtido equivale ao texto cifrado. Na Figura 27 (b), é possível verificar a evolução para frente, partindo-se do texto cifrado, de

cima para baixo, onde os bits extras são removidos a cada passo da evolução, até alcançar o número de passos em que o reticulado corresponda ao texto claro. Um fato observado por Gutowitz é que ao se alterar um bit no texto claro, o texto cifrado permanece inalterado apenas do lado oposto à sensitividade da regra, como pode ser observado pela Figura 27 (c). Na figura, foi utilizado "#" para indicar os bits alterados e "-" para os bits que permaneceram iguais. Esse comportamento é considerado insatisfatório, pois facilitaria a criptoanálise e uma eventual quebra da chave secreta.

Como solução para este problema, o modelo de Gutowitz utiliza duas sub-etapas, sendo uma delas empregando regras irreversíveis sensíveis à esquerda e outra com regras irreversíveis sensíveis à direita. Assim, o problema da propagação da perturbação apresentado na Figura 27 (c) não ocorre neste modelo. Porém, além de serem necessárias duas sub-etapas para cifrar ou decifrar um bloco, a utilização do cálculo de pré-imagem com bits adicionais torna o texto cifrado consideravelmente maior que o texto claro.

4.3.2 Fase de substituição

Na fase de substituição cada bloco de texto é transformado em *N frames* compostos por *n* bits, e então, são aplicadas permutações simples entre estes *frames*. Desta forma, cada um destes *frames* corresponde a uma célula do AC possuindo 2^n estados, com um total de *N* células. Como exemplo, considere que existem 64 *frames*, nos quais possuem 6 bits cada (veja Figura 28). A permutação será realizada entre os 64 *frames*, ou seja, serão feitas trocas de posições entre estes *frames*. A forma mais simples de realizar uma permutação entre 64 posições é definir que cada posição seja representada por um número de 0 a 63 e então definir em seqüência 64 números entre 0 e 63 sem repetições, para então reorganizar as posições conforme a seqüência indicada. Sendo assim, seriam necessários 384 bits responsáveis por definir a seqüência de permutação: para representar 64 números são necessários 6 bits, e como existem 64 posições, tem-se $64 \times 6 = 384$ bits. De acordo com Gutowiz, existem outras formas de representar esta permutação, onde é necessária uma menor quantidade de bits na representação de cada posição, por exemplo, a representação na forma de árvore binária. Embora a solução em árvore seja mais econômica, a maneira de realizar a permutação no sistema criptográfico é a mesma da explicada anteriormente.

A Figura 28 ilustra como os blocos são transformados em *frames* e também mostra um exemplo onde estes *frames* são trocados de posição. Neste exemplo está em destaque o frame da posição 63, que passa a ocupar a posição (célula) 1, de acordo com a seqüência de bits para

permutação apresentada na figura: 58, 63, ..., 19.

Além das permutações entre os *frames* são executados também deslocamentos circulares entre os bits dentro de cada *frame*. Estes deslocamentos são determinados por algumas referências fixas. No processo de decifragem são feitos deslocamentos e permutações na ordem inversa. As etapas de permutação e deslocamento são reversíveis. Segundo Gutowitz, essas operações são executadas por ACs reversíveis de raio 0.

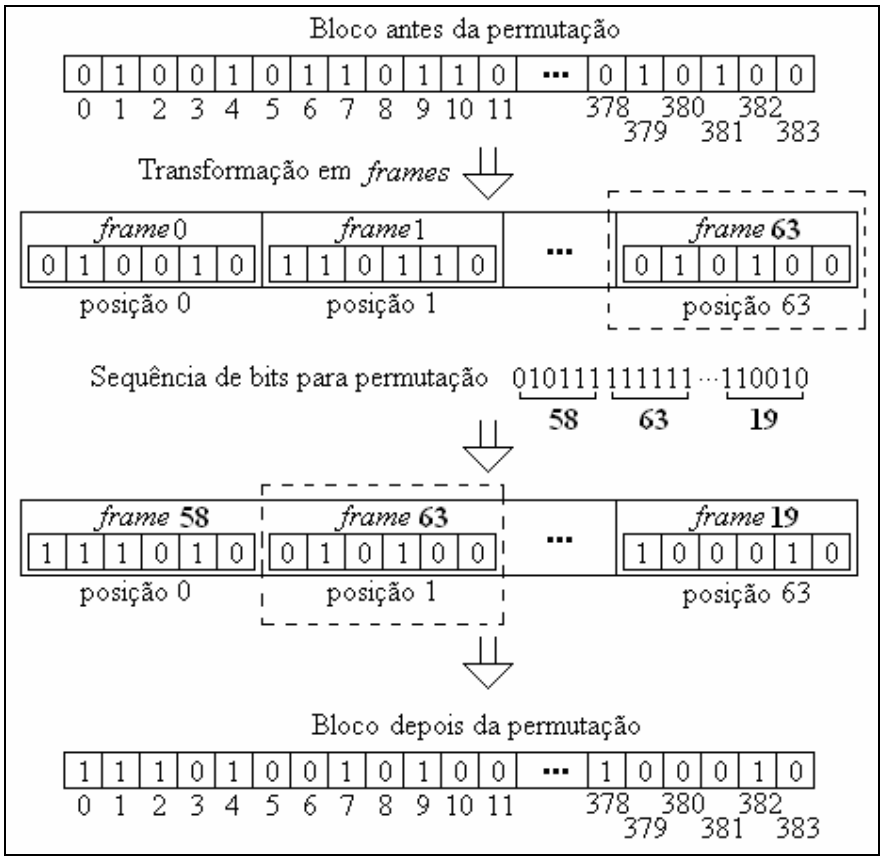


Figura 28 - Permutação entre 64 *frames* efetuada por de uma seqüência de 384 bits.

4.3.3 Visão geral do método

A chave criptográfica utilizada neste método possui 1088 bits e é dividida em duas partes: o *block key* formado por 1024 bits e o *link key* que possui 64 bits. As regras irreversíveis são obtidas diretamente a partir do *block key*. As regras reversíveis são geradas a partir de um processamento do *link key*, cujo resultado é chamado de *link* e possui 320 bits.

Este método, denominado *block-link cryptosystem*, utiliza blocos de 384 bits e *links* de 320 bits. As mensagens a serem cifradas são transformadas em blocos. Os links são utilizados tanto para gerar a seqüência responsável pela permutação na fase de substituição, como

também correspondem aos bits que são adicionas a cada passo do cálculo de pré-imagem (bits iniciais). A decifragem recupera os blocos e destrói os *links.*

Na especificação da fase de difusão, são realizados 32 passos de execução do cálculo de pré-imagem empregando uma regra de raio 5 (*block key*) para os blocos. Na fase de substituição os *frames* referentes a cada bloco são formados por 6 bits.

A Figura 29 mostra uma visão geral do método. A fase de difusão das sub-etapas esquerdas são evoluídas com uma regra sensível à esquerda, enquanto as sub-etapas direitas são evoluídas com uma regra sensível à direita. Note que a fase de substituição, efetuada na primeira etapa da sub-fase esquerda está marcada, significando que esta fase não é executada nesta etapa. Esta escolha é realizada para que o processo de decifragem realize as mesmas operações do processo de cifragem na ordem inversa. Ou seja, tanto na cifragem como na decifragem são executados os seguintes passos: **D**, **S**, **D**, **S**, **D**, **S** e **D**, onde **D** corresponde à difusão e **S** substituição.

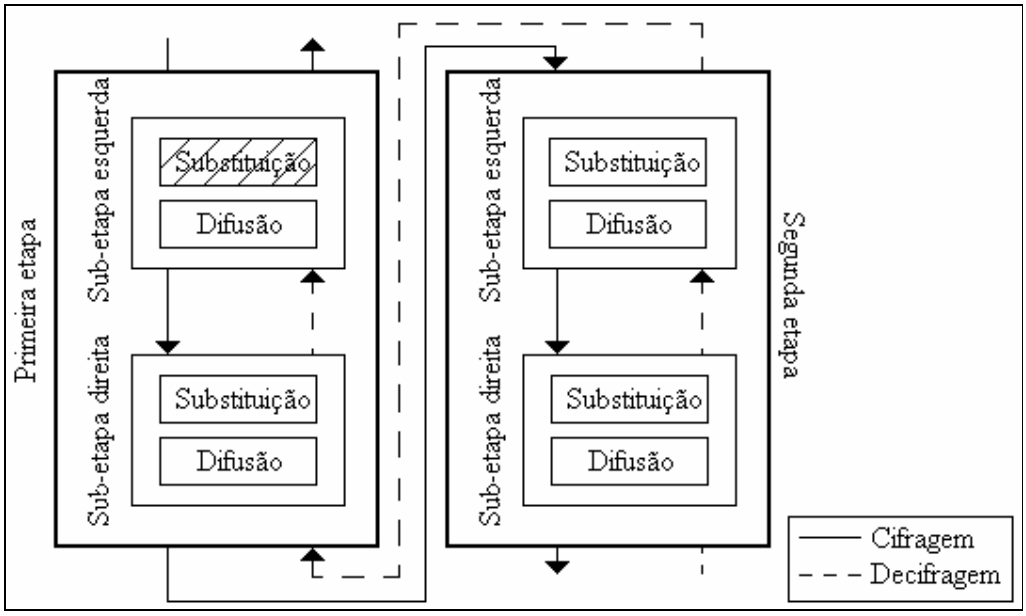


Figura 29 - Visão geral do método de Gutowitz.

Os detalhes da geração do *link* de 320 bits a partir do *link key* de 64 bits serão omitidos. Mas basicamente, o processo é similar à própria cifragem dos blocos. Porém as regras irreversíveis possuem raio 3, e a permutação, realizada pelas regras reversíveis, utilizam *frames* de 4 bits. Mais detalhes referentes a este sistema criptográfico podem ser encontrados em [Gutowitz 1995].

4.4 Alterações no modelo de Gutowitz propostas em [Oliveira et al. 2004]

Oliveira, Coelho e Monteiro (2004) apresentaram uma proposta para resolver o problema encontrado no modelo de Gutowitz ao se fazer uma análise de perturbação na etapa de difusão. A solução é utilizar regras com sensitividade bidirecional no lugar de regras com sensitividade em apenas uma das extremidades. Neste caso o cálculo de pré-imagem pode começar com bits iniciais consecutivos em qualquer célula da pré-imagem, e então, os demais bits decorrem da utilização da regra.

É importante que a cada pré-imagem calculada, os bits iniciais utilizem células em posições distintas, pois, se forem usadas sempre as células mais a esquerda ou mais a direita como bits iniciais, este método se comportaria exatamente como o modelo de Gutowitz.

A Figura 30 representa uma análise de perturbação sobre o método proposto por Oliveira e colaboradores. A Figura 30 (a) mostra o cálculo de pré-imagem evoluindo de cima para baixo por 10 passos de tempo (cifragem), e sua correspondente evolução para frente pode ser observada na Figura 30 (b) (decifragem). A Figura 30 (c) indica a perturbação provocada por apenas um bit se propagando por todo o reticulado até o final do cálculo da pré-imagem.

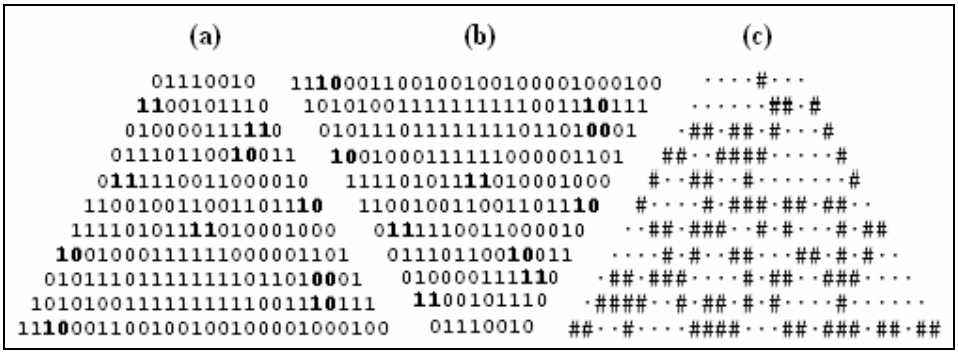


Figura 30 - Método proposto por Oliveira, Coelho e Monteiro: (a) Cifragem a partir de um texto claro (b) Diferenças provocadas pela perturbação [Oliveira *et al*. 2004].

Dessa forma, o método proposto por Oliveira e colaboradores (2004) resolve o problema referente à propagação da perturbação pelo reticulado, porém, ainda é necessário adicionar bits em cada pré-imagem calculada, tornando o texto cifrado maior que o texto claro.

4.5 Investigação do cálculo de pré-imagem proposto por Wuensche e Lesser para aplicação em criptografia

No método para calcular pré-imagens de Wuensche e Lesser (1992) não é preciso utilizar bits adicionais e o cálculo é efetuado para qualquer tipo de regra, com ou sem sensitividade. Esse método foi apresentado na seção 3.6.

Na tentativa de resolver o problema relacionado ao acréscimo de bits no bloco de texto cifrado, encontrado nos modelos de Gutowitz (1995) e de Oliveira e colaboradores (2004), Lima (2005) analisou se o método de cálculo de pré-imagens proposto por Wuensche e Lesser poderia ser utilizado em um novo modelo de criptografia baseado em ACs.

Segundo Lima, umas das vantagens de se empregar este método é que ele pode ser aplicado a regras de transição mais genéricas, mesmo que elas não tenham a característica da sensitividade à esquerda ou à direita, além de tornar o reticulado final (texto cifrado) do mesmo tamanho que o reticulado inicial (texto claro).

Uma dos problemas encontrados por Lima em sua investigação foi descobrir regras capazes de calcular pelo menos uma pré-imagem para qualquer que seja o reticulado inicial. No cálculo de pré-imagens de Wuensche e Lesser, nem todos os textos claros poderiam ser capazes de sofrer a transformação para textos cifrados, porque para determinados reticulados, simplesmente não existe uma pré-imagem.

Para realizar esta investigação, Lima utilizou a técnica evolutiva conhecida como algoritmo genético (AG) [Lima, 2005] e a utilização de parâmetros de comportamento dinâmico para guiar o AG na busca de regras com uma determinada característica dinâmica.

A dinâmica de um autômato celular está associada à sua regra de transição. Com o intuito de ajudar na previsão do comportamento dinâmico dos ACs, vários parâmetros calculados diretamente sobre as regras de transição foram propostos [Oliveira 1999]. A idéia desses parâmetros é realizar um cálculo sobre os bits de saída da regra de transição, cujo resultado fornece uma estimativa de comportamento do AC quando a regra for aplicada em um reticulado inicial arbitrário. Diversos parâmetros foram propostos na literatura, tais como, lambda (λ), Z, sensitividade, domínio da vizinhança e atividade absoluta [Oliveira 1999].

Lima empregou o parâmetro Z, pelo fato deste parâmetro ser capaz de quantificar a capacidade que uma regra de transição possui em perturbar uma célula qualquer do reticulado à medida que as pré-imagens são calculadas. Em [Wuensche 1999] e [Oliveira *et al*. 2001] são apresentadas análises do parâmetro Z e as principais conclusões são: (i) ele é um excelente

discriminador do comportamento caótico; (ii) regras com um valor de Z igual ou próximo de 1 têm alta probabilidade de serem caóticas. Esperava-se que com a utilização de regras com valor de Z alto haveria uma alta probabilidade de existir pelo menos uma pré-imagem para a maioria dos reticulados possíveis.

Como resultado de sua investigação, Lima mostrou que uma variação do algoritmo de Wuensche e Lesser, que adiciona bits ao reticulado a cada passo de cifragem, mostrou ser viável para aplicação em criptografia, bastando que as regras utilizadas como chave tenham Z igual a 1. Um AG pode ser facilmente empregado na geração de chaves com essa característica. Em relação aos métodos propostos anteriormente [Gutowitz 1993] (seção 4.3) e [Oliveira *et al*. 2004] (seção 4.4), esse método teria a vantagem de empregar regras mais gerais. Entretanto, permanece a mesma desvantagem, que é um aumento significativo do texto cifrado em relação ao texto original.

Com relação à aplicação do algoritmo original de Wuensche e Lesser, que não teria a desvantagem associada ao aumento do texto, os resultados se mostraram promissores. Porém, ainda permaneceram questões em aberto, por exemplo, como encontrar um conjunto de parâmetros que garanta a característica de uma regra (chave do sistema) ser capaz de cifrar qualquer reticulado inicial?

4.6 Outros modelos encontrados na literatura

Além dos modelos apresentados nas seções anteriores, podemos citar os sistemas criptográficos baseados no conceito de chave pública propostos em [Kari 1992] e em [Guan 1987], que empregam o uso de ACs bidimensionais. Nestes modelos são executadas apenas evoluções para frente, onde os ACs utilizados no processo de cifragem são diferentes daqueles empregados na decifragem. O maior problema encontrado nestes modelos é encontrar um autômato celular inverso, uma vez que não existe um método geral para encontrar ACs inversos.

Outras pesquisas na área de ACs aplicados a criptografia que podemos citar são os modelos baseados em cifragem por fluxo (*stream cipher*) ([Tomassini and Perrenoud 2000] e [Benkiniouar and Benmohamed 2004]), que geralmente possuem como principal meta a geração de números pseudo-aleatórios, como no modelo pioneiro de Wolfram (seção 4.1). No modelo pesquisado em [Seredynski, Bouvry and Zomaya 2003] são empregados algoritmos genéticos na busca de ACs unidimensionais não-homogêneos capazes de produzir uma

seqüência de bits aleatória, estatisticamente de boa qualidade.

Em geral, os modelos criptográficos baseados em ACs procuram explorar ao máximo o paralelismo inerente aos ACs e, na sua maioria, tentam aproveitar o fato de que ACs são bons geradores de números pseudo-aleatórios, usando esta característica combinada a algumas outras operações, como por exemplo XOR. Dentre os pesquisadores que exploram o comportamento dinâmico dos ACs no processo de cifragem, alguns aproveitam os conceitos das propriedades algébricas, fazendo uso de regras aditivas, e outros, investigam a possibilidade de evolução para trás, empregando regras irreversíveis. O método proposto nesta dissertação utiliza o comportamento dinâmico dos ACs como o próprio sistema criptográfico, fazendo uso de regras reversíveis, não-aditivas com ACs não-homogêneos.

Capítulo 5

Investigações para definição de um novo método para cálculo de pré-imagem

A criação do método criptográfico que será descrito adiante nesta dissertação teve como principal inspiração os modelos criptográficos propostos por Gutowitz (1995) e Oliveira e colaboradores (2004), que foram brevemente discutidos na seção 4.3 e 4.4, respectivamente. O modelo apresentado nesta dissertação pertence ao conjunto de sistemas criptográficos simétricos baseados em cifragem por blocos. Cada bloco de texto claro é transformado em bloco de texto cifrado através de um método de cálculo de pré-imagem desenvolvido especificamente para o modelo. Assim como no cálculo de pré-imagem de Wuensche e Lesser, não é necessário adicionar bits extras a cada passo do cálculo efetuado, tornando o tamanho do bloco de texto cifrado igual ao tamanho do bloco de texto claro. Além disso, o método garante que existe uma pré-imagem para qualquer configuração de reticulado inicial possível.

A construção do cálculo de pré-imagem foi alcançada a partir da reprodução e análise dos experimentos realizados na investigação feita por Lima (2005), e descrita na seção 4.5. Embora os resultados apresentados pelo autor sejam bem próximos daqueles encontrados na reprodução, testes adicionais foram executados revelando que apenas um conjunto muito restritivo de regras pode ser usado em um sistema criptográfico baseado no cálculo de pré-imagem de Wuensche e Lesser. A característica deste conjunto de regras é apresentada na seção seguinte.

5.1 Regras capazes de encontrar pré-imagem para qualquer reticulado inicial

Ao reproduzir os experimentos realizados por Lima (2005), uma das questões adicionais a ser investigada foi a seguinte: existem regras capazes de encontrar pré-imagem para qualquer que seja o reticulado inicial?

Como citado na seção 4.5, a investigação em [Lima 2005] se concentrou na utilização do parâmetro Z [Wuensche 1999] para tentar responder a esta pergunta, mas tal investigação não retornou resultados conclusivos. Por outro lado, conseguimos responder a esta pergunta

ao aplicar o método de Wuensche e Lesser no conjunto formado pelas regras elementares com parâmetro Z igual a 1. O experimento foi realizado com o conjunto de todas as configurações de estados possíveis para um dado tamanho de reticulado. Para um reticulado contendo 5 células, foram analisados um total de 2^5 reticulados, na tentativa de encontrar pelo menos uma pré-imagem por reticulado. Foram testados reticulados de tamanho 5, 6, 7, 8, 9 e 10, e a partir daí, analisadas todas as regras que conseguiram encontrar pelo menos uma pré-imagem para todos os reticulados possíveis naquele tamanho de reticulado.

O resultado do teste indica que apenas seis regras conseguem encontrar pré-imagens para qualquer que seja o reticulado inicial (Apêndice B). As regras elementares com parâmetro Z igual a 1 que conseguem encontrar pré-imagem são: 15, 51, 85, 170, 204, 240.

Nos concentramos então na seguinte questão: que característica estas regras possuem que as tornam capazes de encontrar pré-imagem para qualquer configuração inicial de reticulados?

É sabido que todas as regras elementares que possuem o parâmetro Z igual a 1 são sensíveis a algum dos bits da vizinhança e são formadas por uma vizinhança de tamanho 3 [Lima 2005]. Dessa forma, as regras citadas acima possuem a característica da sensitividade em algum dos 3 bits de vizinhança, sendo que as regras 15 e 240 são sensíveis ao bit da esquerda, as regras 51 e 204 ao bit central e as regras 85 e 170 ao bit direito da vizinhança.

Em uma análise mais detalhada foi possível constatar que as regras sensíveis ao bit central apenas congelam (repetem) o reticulado inicial. E as regras com sensibilidade às extremidades fazem deslocamentos à direita ou à esquerda, dependendo do sentido da sensitividade. Além disso, existem duas regras para cada um dos comportamentos citados, sendo que uma é o complemento da outra. Por exemplo: a regra 15 possui a disposição dos bits da seguinte forma: 11110000; a regra 240 possui a formação dos bits representada pelo complemento dos bits referente à regra 15, sendo assim, os bits correspondentes à regra 240 são: 00001111.

Além das observações feitas acima, a mais importante delas refere-se à seguinte propriedade apresentada por estas regras: o bit sensível da vizinhança define de forma única o bit de saída da regra de transição. Essa propriedade faz com que, no cálculo de pré-imagem de Wuensche e Lesser, além de sempre ser possível calcular uma pré-imagem da primeira à última célula, existe sempre uma única pré-imagem que resolve a condição de contorno. A Figura 31 apresenta as regras 15 e 240, sensíveis ao bit esquerdo, além de mostrar as regras 85

e 170, sensíveis ao bit direito.

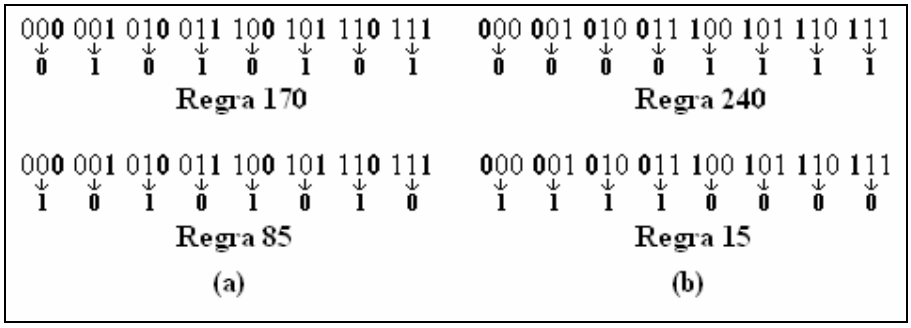


Figura 31 - Regras capazes de resolver a condição imposta pela região de contorno do cálculo de pré-imagem proposto por Wuensche e Lesser; (a) regras sensíveis à direita e (b) regras sensíveis à esquerda.

Para exemplificar como estas regras são capazes de resolver a condição de contorno, considere as regras 15 e 240 sensíveis ao bit esquerdo, conforme mostra a Figura 32. De acordo com o cálculo apresentado na seção 3.6, é necessário adicionar uma célula de cada lado do reticulado referente à pré-imagem e escolher os valores iniciais das duas primeiras células da pré-imagem para dar início ao cálculo. A Figura 32 mostra as células preenchidas por P0, P1, ..., P9 representando a pré-imagem a ser calculada, e as células contendo R0, R1, ..., R7 como os bits referente ao reticulado inicial. Neste exemplo, pode ser usada a regra 15 ou a regra 240. Uma vez que as duas regras são sensíveis à esquerda, o cálculo deve ser iniciado da direita para esquerda. Ao contrário do modelo apresentado na seção 3.6, que sorteia os dois bits iniciais de forma aleatória, para estas regras é mais adequado definir estes bits de acordo com as células do reticulado R0 e R1, já que as duas regras permitem que estes bits sejam definidos antecipadamente, como pode ser observado na Figura 32. O motivo de não usar o sorteio aleatório nos bits iniciais é que o cálculo somente chegará até o final se o bit P0 for igual a P8, e P1 for igual a P9. Sendo assim, não adianta ficar sorteando bits iniciais que o cálculo terá que voltar (através do acesso a pilha), até que os bits iniciais P8 e P9 estejam configurados de acordo com os bits do reticulado R0 e R1 que definem P0 e P1 de forma única. Após encontrar os bits iniciais o cálculo segue normalmente de P7 até P0, da direita para esquerda, pelo fato das regras serem sensíveis à esquerda. No momento em que o cálculo chega ao final não é necessário confirmar os bits da região de contorno, pois os bits referentes ao contorno já haviam sido definidos no início do cálculo. As regras que possuem a característica citada no parágrafo anterior permitem que os bits P0 e P1 definidos por R0 e R1 tenham qualquer configuração de vizinhança e, portanto, estas regras sempre encontram uma pré-imagem para qualquer que seja o reticulado inicial, algo que nem sempre ocorre com regras sem a característica mencionada.

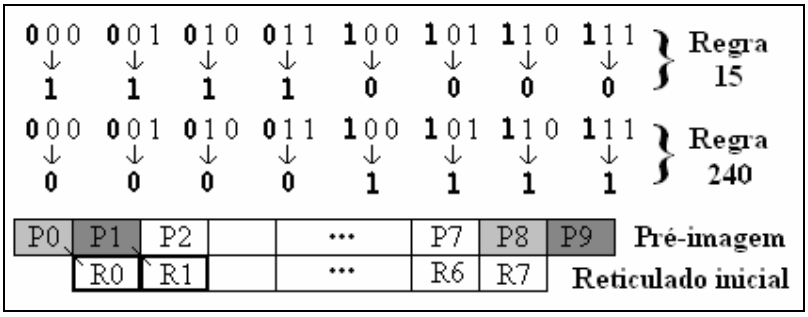


Figura 32 - Cálculo de pré-imagem proposto por Wuensche e Lesser utilizando as duas regras sensíveis à esquerda que sempre conseguem encontrar pré-imagem.

De acordo com a análise feita nas regras aqui citadas é possível concluir que, independentemente do tamanho do reticulado, estas regras sempre encontram pré-imagem e esta pré-imagem é única.

Uma análise similar pode ser feita para regras sensíveis ao bit direito, a única diferença está na direção em que o cálculo é executado, da esquerda para a esquerda.

5.2 Conclusões sobre o cálculo de pré-imagem proposto por Wuensche e Lesser como método de cifragem

O cálculo de pré-imagem de Wuensche e Lesser é um método de encontrar pré-imagens de uso geral, definido para ACs homogêneos e com condição de contorno periódica. Se existe uma pré-imagem para um dado reticulado e regra, o método irá encontrá-la. Dois problemas foram encontrados para a aplicação direta desse método em um sistema criptográfico: primeiro, existe um conjunto muito restrito de regras capazes de encontrar pré-imagem para qualquer que seja o reticulado e a cardinalidade desse conjunto define o número de chaves possíveis. Na medida em que o raio é aumentado a cardinalidade do conjunto de regras também aumenta, porém, este aumento não é grande o suficiente, sendo que a quantidade de regras é dada por $2\times(2\times r+1)$, onde r é o raio do AC. O segundo problema, diz respeito ao comportamento dinâmico de um reticulado ao sofrer evolução por alguma das regras que possua a característica indicada na seção 5.1. Dentre as regras elementares que possuem esta característica, algumas são classificadas como ponto fixo e as demais como ciclo periódico. Mesmo empregando-se ACs com raio de tamanho maior, o comportamento dinâmico das regras com a característica indicada também não será caótico. Dessa forma, estas regras são inadequadas para aplicação em um sistema criptográfico baseado em ACs homogêneos.

5.3 Características das regras empregadas no cálculo de pré-imagem aqui proposto

A principal idéia do cálculo de pré-imagem aqui proposto está na utilização de uma regra com a característica de garantia de existência de pré-imagem, apresentada na seção 5.1, juntamente com uma regra com sensitividade (à esquerda, à direita ou em ambas as direções), como as utilizadas em Gutowitz (1995) e Oliveira e colaboradores (2004). Portanto, ao contrário do cálculo de pré-imagem de Wuensche e Lesser, esse método emprega ACs não-homogêneos para efetuar o cálculo de pré-imagem. Isto é, duas regras diferentes são utilizadas na transição dos estados do reticulado.

A diferença no cálculo aqui proposto está fortemente relacionada à regra utilizada na região de contorno, que passamos a chamar de regra de contorno. A regra de contorno é responsável por fazer com que não sejam necessários bits adicionais a cada passo do cálculo de pré-imagem, sendo assim, esta regra garante que qualquer reticulado inicial possua uma pré-imagem de mesmo tamanho. A segunda regra é responsável pela caoticidade do AC, fazendo com que a mensagem seja "embaralhada" na evolução do AC para trás. Além disso, a forma como a regra com sensitividade é gerada faz com que o espaço de chaves tenha uma cardinalidade o suficiente para a segurança do método. Passaremos a chamar esta última regra de regra principal.

No método para cálculo de pré-imagem proposto por Gutowitz (seção 4.3.1), a garantia de encontrar pré-imagens foi alcançada ao se utilizar bits adicionais em conjunto com uma regra sensível a alguma das extremidades, empregando ACs homogêneos. O método proposto nesta dissertação garante a existência de uma pré-imagem utilizando ACs não-homogêneos evoluindo com dois tipos de regras, sendo que uma das regras possui sensitividade a alguma das extremidades e a outra é responsável por resolver a condição de contorno do reticulado, garantindo assim a existência de uma pré-imagem.

5.3.1 Escolha da regra principal

A regra principal é aplicada na atualização de todas as células do reticulado, exceto as células que estão localizadas na região de contorno. O número de células que representam a região de contorno é dado por $2 \times raio$. No caso de um AC com raio igual a 1, apenas dois bits serão calculados com a regra de contorno e os demais pela regra principal.

A regra principal possui influência direta no comportamento dinâmico do AC e,

portanto, ela é responsável por executar de fato o processo de cifragem através do cálculo de pré-imagem. Assim como nos métodos de Gutowitz (1995) e Oliveira e colaboradores (2004), a regra principal aqui empregada deve ser sensível a alguma das extremidades, pois esta característica faz com que o bit mais à esquerda ou à direita de uma vizinhança seja determinístico ao executar o cálculo de pré-imagem em uma direção (esquerda para direita ou vice-versa). A direção em que o cálculo é executado depende do lado em que a regra principal possui sensitividade. Além disso, a sensitividade da regra aumenta a probabilidade do comportamento dinâmico do AC ser caótico.

Regras sensíveis à esquerda ou à direita podem ser criadas a partir de um conjunto menor de bits, denominado núcleo da regra. Por exemplo, a regra de raio 1 sensível à esquerda {**0100**1011} pode ser criada a partir dos bits {**0100**} que definem as transições das vizinhanças 000, 001, 010 e 011. Os valores das transições restantes da regra (100, 101, 110 e 111) decorrem dos quatro bits do núcleo, pois para atender a propriedade de sensitividade à esquerda, eles devem ser o complemento do núcleo (1011). Caso fosse usado o mesmo núcleo para criar uma regra sensível à direita, esta regra seria {**0**1**1**0**0**1**0**1}. Assim, o núcleo de uma regra sensível a alguma das extremidades é formado pela metade dos bits que uma regra possui, sendo que os demais bits decorrem da sensitividade da regra.

Os núcleos das regras com sensitividade representam o conjunto de chaves criptográficas possíveis em todas as versões dos métodos aqui propostos, sendo que a quantidade de núcleos é determinada pelo raio da regra utilizada no sistema criptográfico. Para um raio específico, o tamanho do núcleo que gera essas regras é dado por $2^{2\times raio}$ e a cardinalidade do conjunto de núcleos com sensitividade é dado por: $2\times 2^{2^{2\times raio}}$. A Tabela 11 demonstra a quantidade de núcleos possíveis de acordo com o raio da regra utilizada, além do tamanho do núcleo.

Tabela 11- Raio de uma regra e a cardinalidade do conjunto de regras possíveis.

Raio	**Tamanho do núcleo (bits)**	**Quantidade de núcleos (≅)**
1	4	32
2	16	131072
3	64	3×10^{19}
4	256	$2{,}3\times10^{77}$
5	1024	$2{,}3\times10^{308}$

5.3.2 Escolha da regra de contorno

O conjunto das regras de contorno é formado por apenas 4 regras, independentemente do tamanho do raio utilizado, sendo que duas delas são sensíveis à esquerda e as outras duas são sensíveis à direita. Estas regras possuem como função apenas garantir que todos os reticulados possíveis possuam uma pré-imagem e, portanto, não são responsáveis diretas pelo comportamento dinâmico global dos ACs. As regras de contorno apenas repassam os bits que chegam até a borda do reticulado para o outro lado do reticulado. Apenas as regras com sensitividade a alguma das extremidades são usadas como contorno, como as quatro apresentadas na Figura 31.

O critério de escolha da regra de contorno é definido de acordo com o primeiro bit da regra principal, sendo que a regra de contorno deve obedecer à mesma sensitividade da regra principal. Caso o primeiro bit da regra principal seja o bit 0, a regra de contorno deverá começar com o primeiro bit igual a 1, sendo que os demais bits decorrem da própria característica destas regras. Suponha que a regra principal seja sensível à esquerda. Como só existe uma regra de contorno sensível à esquerda com primeiro bit igual a 1, estas regras são definidas unicamente pela regra principal.

Por exemplo, considere {01001011} que é uma regra sensível à esquerda como sendo a regra principal, a regra de contorno também deve ser sensível à esquerda e deve iniciar com o bit 1, resultando na regra {11110000}. Caso a regra principal seja {10100110} (sensível à direita), a regra de contorno será {01010101}.

A escolha da regra de contorno possuindo o primeiro bit contrário ao primeiro bit da regra principal é utilizada para poder inserir um ruído emitido pela borda do reticulado, que ao chegar até as células executadas pela regra principal provocará uma maior perturbação em todo o reticulado, fazendo com que um reticulado muito homogêneo sofra grandes modificações ao final de várias execuções do cálculo de pré-imagem. Estas regras são escolhidas seguindo esse mesmo critério para todas as três versões do método de cálculo de pré-imagem que serão vistos nas próximas seções.

5.4 Primeira versão para o método de cálculo de pré-imagem: modelo básico

Na primeira versão, que serve como base para as demais, a idéia é utilizar a regra principal em todas as células do reticulado, exceto naqueles pertencentes à região de contorno.

A Figura 33 mostra um exemplo do cálculo utilizando-se um AC de raio 1. Na Figura 33 (a) temos o reticulado inicial {0100101}, e na pré-imagem, *P1*, *P2*, ..., *P7*, denotam os bits a serem encontrados. O AC não-homogêneo é definido de forma que a regra de contorno é aplicada nas duas primeiras células do reticulado (no caso de raio 1), e a regra principal nas demais células. Neste exemplo, a regra principal é sensível à esquerda, portanto, o cálculo da pré-imagem deve ser seguido da direita para esquerda. As duas células do reticulado que são atualizadas pela regra de contorno estão destacadas em negrito (Figura 33 (a)). Para iniciar o cálculo são determinados os bits *P1* e *P7* pela **regra de contorno** {11110000}, que também é sensível à esquerda e depende apenas do bit de saída (Figura 33 (b)). Encontrados os bits iniciais, que neste exemplo estão localizados nas extremidades do reticulado referente à pré-imagem, o cálculo segue da direita para esquerda (de *P6* à *P2*) utilizando a **regra principal**, que é sempre determinista ao definir o próximo bit da vizinhança esquerda. Ao encontrar todos os bits da pré-imagem, Figura 33 (g), o reticulado referente à pré-imagem, que acabamos de calcular, passa a ser o reticulado inicial para que outra pré-imagem seja encontrada. O processo termina quando for atingida a quantidade de evoluções desejada.

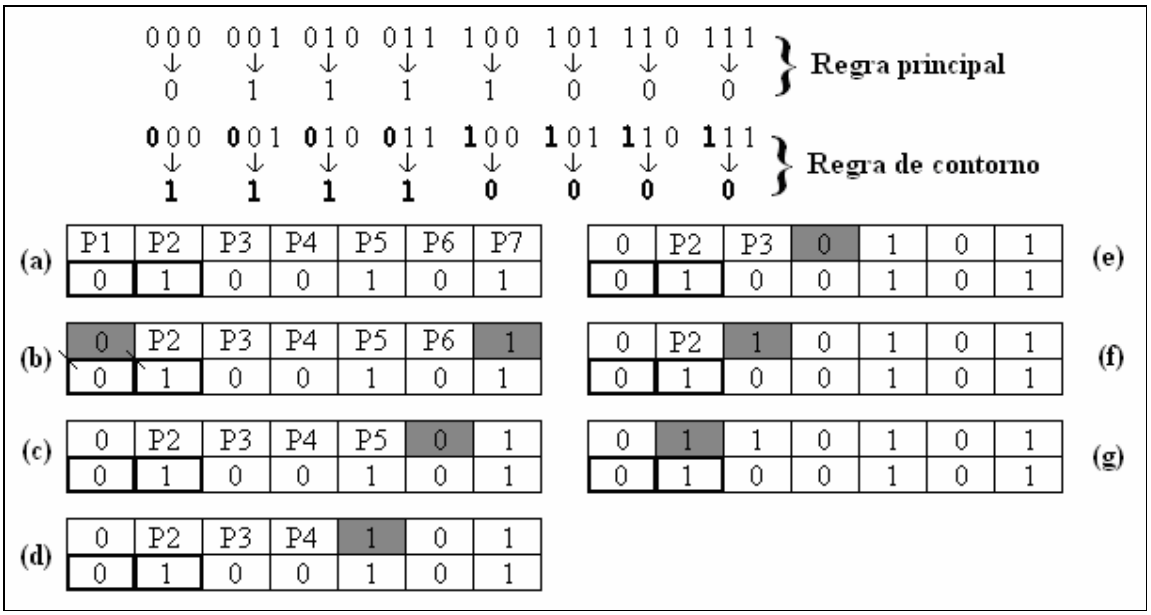


Figura 33 - Cálculo de pré-imagem proposto evoluindo passo a passo.

Ao evoluir o AC para frente o método deverá usar a regra principal para evoluir todas as células, exceto as *n* células situadas mais à esquerda ou mais à direita do reticulado, dependendo da sensitividade da regra em questão, sendo que estes bits devem ser evoluídos pela regra de contorno. O número *n* de células que devem ser atualizadas pela regra de contorno é dado por $2 \times raio$.

Como pode ser observado na Figura 33, para encontrar cada bit da pré-imagem é indispensável ter conhecimento dos outros bits da vizinhança (pré-imagem), além de ser preciso conhecer o bit de saída (reticulado). Sendo assim, este processo é efetuado de forma seqüencial, ou seja, cada bit da pré-imagem depende dos bits anteriores para que este seja determinado.

Para ilustrar a quantidade de tempo necessária para que uma pré-imagem seja calculada, considere que uma unidade de tempo é o suficiente para que o bit mais a esquerda ou mais a direita de uma vizinhança seja encontrado, dada a vizinhança parcial (faltando o elemento mais à esquerda ou mais à direita) e o bit de saída do reticulado inicial referente a esta vizinhança. Considere também que a cópia dos bits iniciais pela regra de contorno equivale a uma unidade de tempo. Dessa forma, a quantidade de tempo necessária (t) para que um passo de pré-imagem seja executado pode ser dada pela equação (11), onde N é o tamanho do reticulado (número de células) e r o raio do AC.

$$t = N - (2 \times r) + 1 \qquad (11)$$

Para o exemplo apresentado na Figura 33 são necessárias seis unidades de tempo para que apenas uma pré-imagem seja encontrada.

A Figura 34 (a) mostra 20 passos de pré-imagem sendo calculados em seqüência, ou seja, a pré-imagem que acaba de ser encontrada passa a ser o reticulado inicial para que outra pré-imagem seja encontrada. As pré-imagens estão sendo calculadas de cima para baixo com a regra principal {10010110} e a regra de contorno {01010101}, sendo que os quadrados preenchidos equivalem ao bit 1. O tempo total gasto (t) para se calcular todas as pré-imagens, partindo-se do reticulado inicial até chegar no reticulado final é dado pela equação (12), onde P corresponde ao número de passos utilizados no cálculo da pré-imagem, N é o tamanho do reticulado e r o raio.

$$t = P \times [N - (2 \times r) + 1] \qquad (12)$$

No exemplo da Figura 34, todas as pré-imagens foram calculadas sem que haja paralelismo. Portanto, a quantidade de tempo gasta para encontrar o reticulado final (na base da figura) foi de vinte vezes o tempo necessário para se calcular apenas uma pré-imagem (equação (11)), neste caso, $t = 20 \times [N - (2 \times r) + 1] = 20 \times [20 - 2 + 1] = 380$ unidades de tempo.

No diagrama espaço-temporal da Figura 34 (b) apenas um bit (célula 10) foi modificado no reticulado inicial, em relação à Figura 34 (a), e o cálculo das 20 pré-imagens consecutivas foi refeito. A diferença entre os dois diagramas da Figura 34 (a) e (b) são representados na Figura 34 (c), onde os quadrados preenchidos representam esta diferença, ou seja, os bits preenchidos em (c) correspondem aos bits 1 ao realizar a operação XOR entre (a) e (b). É possível observar que uma simples perturbação no reticulado inicial se propaga por todo o reticulado.

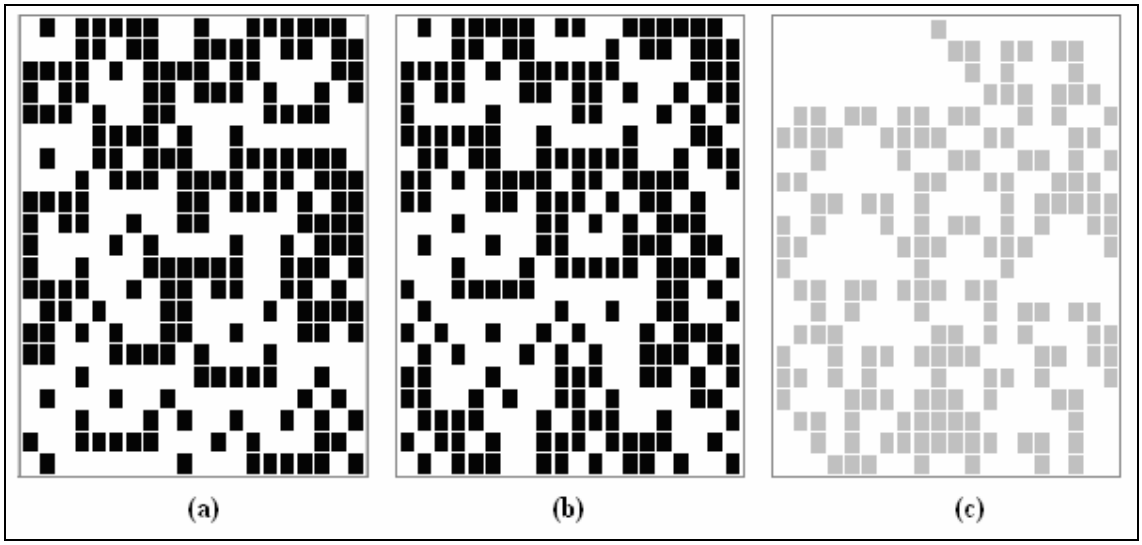


Figura 34 - Evolução temporal da primeira versão (a) Calculo de pré-imagem sendo executado de cima para baixo por 20 passos de tempo (b) Nova evolução ao alterar apenas um bit (célula 10) no reticulado inicial (c) Diferença entre os diagramas (a) e (b).

5.5 Segunda versão para o método do cálculo de pré-imagem: modelo com rotação na borda

Conforme mencionado anteriormente, uma das maiores motivações para se empregar ACs em criptografia está no paralelismo intrínseco dos ACs, que tem um grande potencial para ser aproveitado em *hardware*. No modelo descrito na seção anterior, o cálculo da pré-imagem é realizado de forma seqüencial, sem possibilidade de implementação paralela.

Baseado no modelo descrito na seção anterior, um novo modelo de cálculo de pré-imagem foi elaborado capaz de prover um paralelismo na evolução para trás. Para isso, é necessária uma pequena modificação na forma como os bits da região de contorno são definidos no reticulado. Assim como na primeira versão, apenas uma regra principal e sua respectiva regra de contorno serão utilizadas. A idéia neste novo modelo é alterar, a cada passo de tempo, as posições das células que definem a região de contorno. A cada iteração, a

borda, ou região de contorno, se desloca na mesma direção da sensitividade da regra principal, ao ser evoluído para trás. Sendo que na evolução para frente, a borda se desloca na direção oposta à da sensitividade.

A Figura 35 apresenta um exemplo onde três pré-imagens são calculadas de forma paralela. Foi utilizada na construção do exemplo a mesma regra e a mesma configuração do reticulado inicial da Figura 33.

Na Figura 35 (a) vemos o reticulado inicial e as células que deverão ser calculadas para o cálculo de três pré-imagens consecutivas (de baixo para cima). Além disso, a figura destaca em negrito as posições que são definidas como a borda do reticulado (região de contorno) a cada passo. Ou seja, as células destacadas são as que devem ser submetidas à regra de contorno, a cada passo da evolução temporal do AC (evolução para frente). É possível notar que a borda se desloca em duas posições para a esquerda a cada passo do cálculo de pré-imagem, o que equivale a deslocar duas posições para a direita na evolução para frente.

A Figura 35 (b) apresenta o início do cálculo da primeira pré-imagem, que é igual ao que é feito no modelo da primeira versão (Figura 33 (b)): define-se o valor das duas células que dependem apenas do reticulado inicial, devido à regra de contorno. O cálculo nessa primeira pré-imagem prossegue de forma similar à Figura 33, como é possível observar na Figura 35 (c). Entretanto, no mesmo momento em que a quarta célula da primeira pré-imagem é calculada, os bits iniciais da segunda pré-imagem também já podem ser definidos, pois, eles dependem apenas dos bits que já foram definidos na primeira pré-imagem, nos passos anteriores (Figura 35 (d)).

Na Figura 35(e) vemos que a quarta célula da primeira pré-imagem e a terceira célula da segunda pré-imagem também podem ser calculadas em paralelo, utilizando-se a regra principal. Na Figura 35 (f) vemos que a terceira pré-imagem define os bits inicias com a regra de contorno no mesmo momento em que são calculados um bit da segunda e primeira pré-imagem com a regra principal.

A partir deste ponto (Figura 35 (g)), as próximas células de cada pré-imagem podem ser calculadas em paralelo, dependendo unicamente da regra principal. Assim, pode-se perceber que com a rotação da borda, é possível calcular quantas pré-imagens se desejar, com um atraso de duas unidades de tempo do início de uma pré-imagem para outra.

Na Figura 35 são apresentadas passo a passo dez unidades de tempo para que o

cálculo seja completamente concluído. A quantidade total de unidades de tempo (*t*) utilizada para o cálculo de pré-imagens para segunda versão é dado pela equação (13), onde *P* corresponde ao número de passos utilizados no cálculo da pré-imagem, *N* é o tamanho do reticulado e *r* o raio.

$$t = \{[N - (2 \times r) + 1] + [2 \times (P - 1)]\} \tag{13}$$

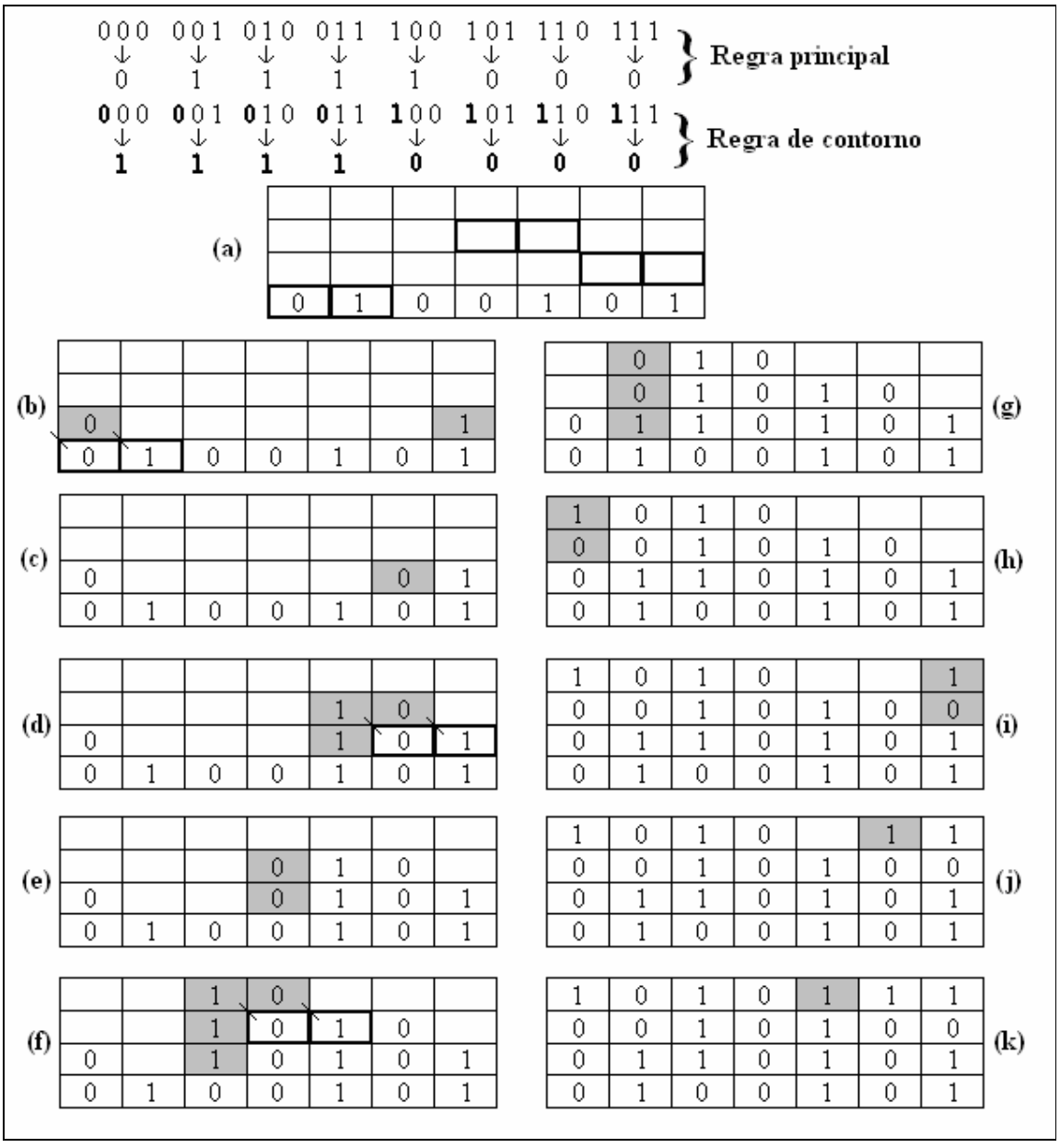


Figura 35 - Exemplo de execução para segunda versão com três pré-imagens consecutivas calculadas passo a passo de forma paralela a partir do reticulado inicial {0100101}.

Por exemplo, caso os passos da Figura 34, que requer 380 unidades de tempo na

execução seqüencial, fossem executados em paralelo, seriam necessárias apenas 57 unidades de tempo para que os 20 passos do cálculo fossem completamente executados.

A Figura 36 (a) apresenta o diagrama espaço-temporal para o segundo método de cálculo de pré-imagem em que foram utilizadas as mesmas regras e reticulado no exemplo da primeira versão (Figura 34). A Figura 36 (b) apresenta a evolução caso a décima célula do reticulado fosse complementada. Na Figura 36 (c) é realizada a análise de perturbação provocada por essa alteração. Como é possível observar, no caso do modelo com borda variável, no início, a perturbação não ocorre em toda a extensão do reticulado. Assim, é necessário um número maior de passos de cálculo de pré-imagem, para que a perturbação se propague por todo o reticulado.

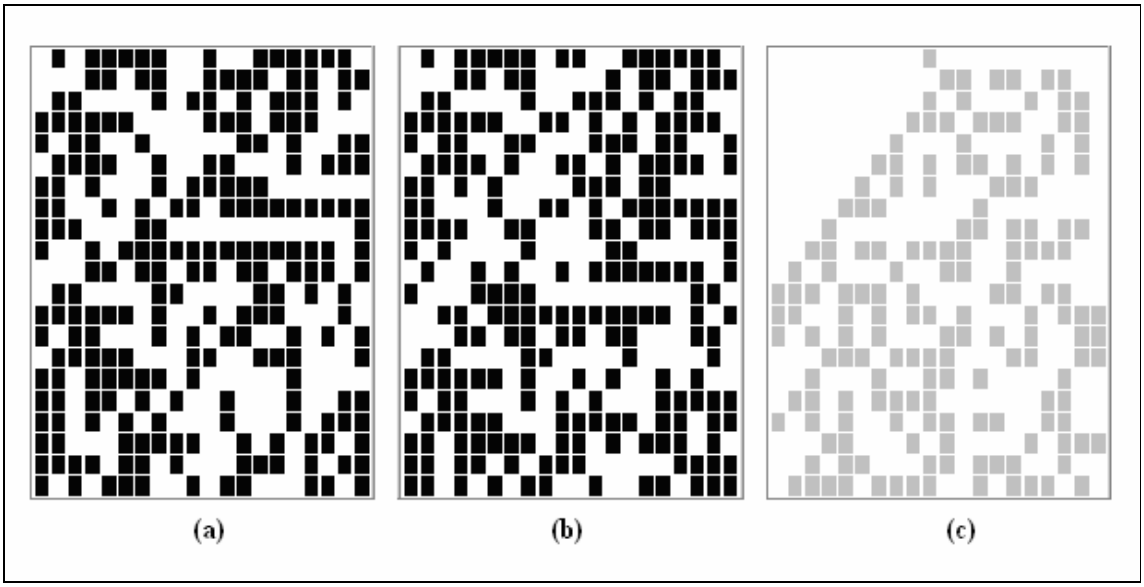


Figura 36 - Evolução temporal da segunda versão (a) Calculo de pré-imagem sendo executado por 20 passos de tempo (b) Nova evolução ao alterar apenas um bit (célula 10) no reticulado inicial (c) Diferença entre os diagramas (a) e (b).

A Figura 37 mostra uma comparação entre as duas versões do cálculo de pré-imagem. O gráfico apresenta a quantidade de unidades de tempo (t), com e sem paralelismo, para raio 1, 3 e 5. O número de pré-imagens é igual ao tamanho do reticulado ($N = P$) e é apresentado no eixo x.

Como pode ser observado na Figura 37, quanto maior é o reticulado inicial e a quantidade de passos consecutivos para o cálculo de pré-imagem, mais significativa é a diferença de unidades de tempo gasta entre as duas versões, onde a segunda versão, por ser executada em paralelo, consome bem menos unidades de tempo do que a primeira.

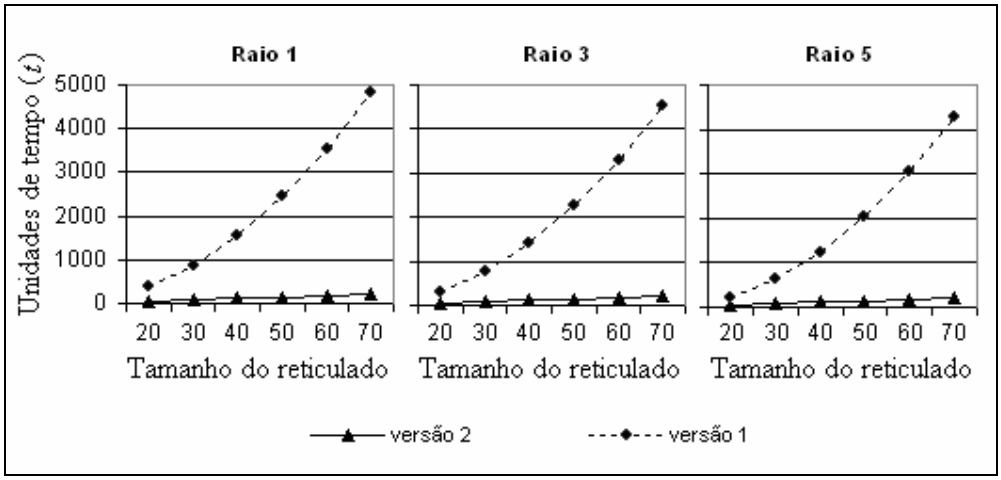


Figura 37 - Comparação entre a quantidade de unidade de tempo gasta para a primeira e segunda versão empregando raio 1, 3 e 5.

O paralelismo na execução do cálculo de pré-imagem pode ser alcançado a cada passo de execução (paralelismo no tempo). Na evolução para frente, é possível efetuar paralelismo no espaço, ou seja, cada célula do reticulado pode ser executada de forma paralela, sendo que a quantidade de tempo para evolução de cada reticulado é dada por apenas uma unidade de tempo. Como são efetuados vários passos de evolução, a quantidade de tempo gasta na evolução para frente é dada pelo número de evoluções a serem realizadas.

5.6 Versão final do método para cálculo de pré-imagem: modelo com rotação na borda e no núcleo

Nas análises que serão apresentadas no próximo capítulo, será possível perceber que as duas versões anteriores possuem uma limitação: para alguns reticulados iniciais, o tamanho do ciclo do AC pode ser extremamente pequeno. Essa característica faz com que a cifragem de alguns textos claros sejam prejudicadas. Após uma análise criteriosa dos modelos anteriores, percebemos que esse problema ocorre devido ao fato de que a mesma regra principal é aplicada em todos os passos do cálculo de pré-imagem.

Nesta última versão, o objetivo é manter o paralelismo apresentado na segunda versão e fazer com que o reticulado seja evoluído a cada passo do cálculo de pré-imagem por uma regra diferente. O conjunto de regras utilizado deve obedecer a mesma direção da sensitividade, ou seja, todas as regras devem ser sensíveis à mesma direção. Este conjunto de regras pode ser definido por um único núcleo, que gera regras sensíveis à esquerda ou à direita. A idéia é usar o núcleo inicial, definido pela chave, para gerar a regra principal da

primeira pré-imagem calculada. A cada passo do cálculo, a configuração da regra principal será gerada pelo núcleo da etapa anterior rotacionado à esquerda em um bit. Dessa forma, a chave do sistema criptográfico define o núcleo da regra que gera uma regra principal a cada passo de execução do cálculo de pré-imagem.

Por exemplo, se usarmos o núcleo {**0111**} para gerar regras sensíveis à esquerda, o passo de cálculo da primeira pré-imagem será efetuado com a regra principal {**0111**1000} e a regra de contorno {**1**1110000}. No cálculo da segunda pré-imagem, o núcleo é rotacionado para esquerda {**1110**}, gerando a regra principal {**1**0**1**0**1**0**0**1} e a regra de contorno {**0**0001111}. O restante do cálculo seguirá esta mesma seqüência até que sejam concluídos todos os passos do cálculo de pré-imagem. Dessa forma as regras de raio 1, que possuem o núcleo formado por 4 bits, resultam em quatro regras principais diferentes que são aplicadas em passos diferentes de pré-imagem. No quinto passo de cálculo de pré-imagem o núcleo retorna ao inicial e a seqüência recomeça. A idéia é a mesma para raios maiores. Ou seja, se fosse utilizada uma regra de raio 2, que possui o núcleo formado por 16 bits, teríamos dezesseis regras principais diferentes para cada passo de evolução. Assim, se desejarmos garantir regras principais diferentes aplicadas a cada passo de pré-imagem, basta escolhermos regras com núcleos maior ou igual ao número de passos que desejamos calcular. Por exemplo, se desejarmos calcular 50 pré-imagens consecutivas (P=50), basta usarmos regras de raio 3, que possuem o núcleo formado por 64 bits, permitindo 64 rotações do mesmo. Na realidade, um mesmo núcleo pode ser utilizado para gerar regras sensíveis à esquerda ou à direita. Passaremos a descrever o caractere "#" e o bit 0 ou 1 no final, por exemplo: {0111#0}, podendo assim identificar quando os núcleos serão utilizados para produzir regras sensíveis à esquerda ou à direita, respectivamente.

Com exceção da utilização da chave, o processo de cálculo de pré-imagem para esta versão é igual ao anterior, inclusive o ganho com paralelismo é o mesmo que o alcançado pelo método anterior.

A partir do exemplo apresentado para a segunda versão do cálculo de pré-imagem (Figura 35), a Figura 38 mostra o mesmo exemplo para a última versão, porém cada uma das três pré-imagens calculadas possuem uma regra de transição diferente. Uma vez que o núcleo aqui empregado é {0111#0}, a primeira pré-imagem é executada pelo núcleo {0111#0}, a segunda pelo núcleo {1110#0} e a terceira pelo núcleo {1101#0}. Estes núcleos são representados pela figura por **A, B** e **C**, respectivamente. Já a regra de contorno é dependente da regra principal gerada pelo núcleo. Por exemplo, se o núcleo for definido pelos bits

{0111#0}, que gera a regra principal {01111000}, a regra de contorno será {11110000}, pois o primeiro bit da regra principal é {0}, sendo que o primeiro bit da regra de contorno corresponde ao seu complemento. As regras de contorno {00001111} e {11110000} são representados por **D** e **E**, respectivamente.

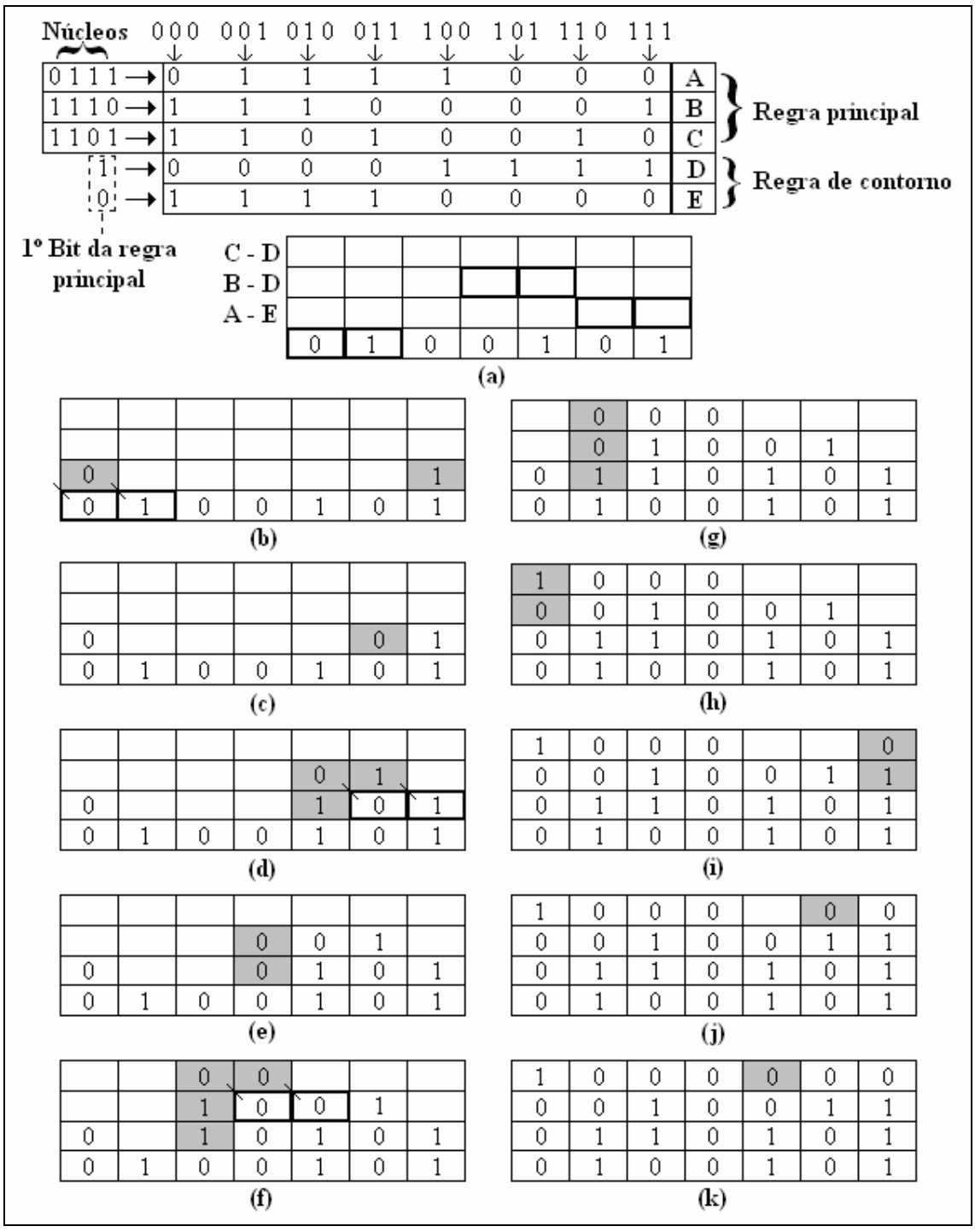


Figura 38 - Exemplo de execução para última versão com três pré-imagens consecutivas calculadas passo a passo de forma paralela a partir do reticulado inicial {0100101}.

A Figura 38 (a) apresenta o reticulado inicial {0100101} e os três reticulados referentes às pré-imagens, que são calculadas de baixo para cima. Assim como no exemplo da Figura 35, as células da região de contorno são apresentadas em negrito. Note do lado esquerdo dos reticulados referente às pré-imagens, as letras informando quais são as regras que operam sobre aquele reticulado, onde a primeira letra define a regra principal e a segunda letra corresponde à regra de contorno. Para a primeira pré-imagem são utilizadas as regras **A** e **E**, na segunda pré-imagem temos **B** e **D**, e a terceira pré-imagem é calculada com as regras **C** e **D**.

Definidas as regras para serem aplicadas a cada pré-imagem, o cálculo segue de forma similar ao apresentado na segunda versão, porém utilizando-se regras principais diferentes, para cada pré-imagem, e suas regras de contorno correspondentes.

A Figura 39 (a) mostra o diagrama espaço-temporal para a versão final onde o mesmo reticulado inicial da Figura 36 foi empregado com o núcleo {0111#0}. Na Figura 39 (b) é apresentado o diagrama espaço-temporal ao se modificar apenas um bit no reticulado inicial da Figura 39 (a). A Figura 39 (c) apresenta a diferença provocada entre os diagramas da Figura 39 (a) e (b). Vemos que são necessários alguns passos de cálculo de pré-imagem para que a perturbação se propague por todo o reticulado.

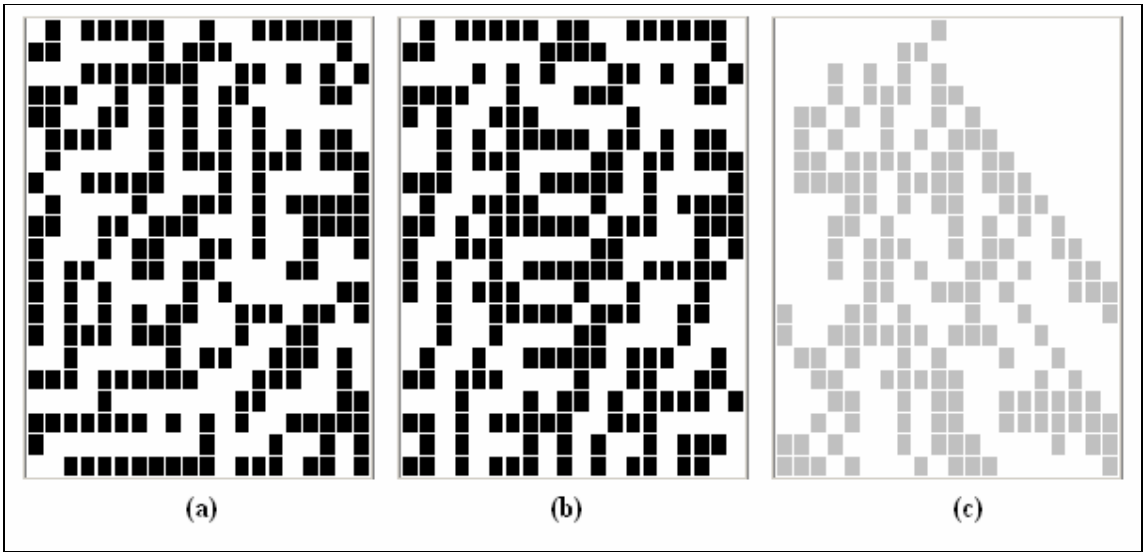


Figura 39 - Evolução temporal da última versão (a) Calculo de pré-imagem sendo executado por 20 passos de tempo (b) Nova evolução ao alterar apenas um bit (célula 10) no reticulado inicial (c) Diferença entre os diagramas (a) e (b).

5.7 Análise preliminar das três versões para cálculo de pré-imagem

Nessa seção, serão discutidas as vantagens, as desvantagens, os possíveis problemas e as modificações para cada uma das versões dos métodos de cálculo de pré-imagem apresentadas nas seções anteriores.

Nos cálculos de pré-imagens com bits adicionais vistos anteriormente ao apresentar os modelos de Gutowitz (1995) e Oliveira e colaboradores (2004), regras com sensitividade a alguma das extremidades possuem $2^{2\times raio}$ pré-imagens diferentes a cada passo de execução do cálculo. Esta quantidade de pré-imagens está relacionada com o número de bits iniciais possíveis, ou seja, ao se utilizar o método de cálculo de pré-imagem proposto por Gutowitz e alguma regra sensível ao bit extremo, qualquer combinação possível de bits iniciais sempre encontrará uma pré-imagem. Fazendo uma análise mais detalhada nas três versões descritas anteriormente é possível notar que, dada a restrição imposta pela condição de contorno, apenas uma pré-imagem é possível de ser calculada com esse tipo de regra. Uma vez que os métodos aqui propostos utilizam uma regra sensível a alguma das extremidades para encontrar uma pré-imagem, juntamente com uma regra de contorno que possui a característica de encontrar apenas uma pré-imagem a cada passo de execução do cálculo, é possível concluir os métodos aqui apresentados sempre encontram alguma pré-imagem, para qualquer que seja o reticulado, e esta pré-imagem é única.

Como visto na seção 3.5, que trata de ACs reversíveis e irreversíveis, um AC que possui exatamente uma pré-imagem para qualquer que seja o reticulado inicial é considerado um AC reversível. Dessa forma, diferentemente dos modelos de Gutowitz (1995) e Oliveira e colaboradores (2004) que são formados por ACs irreversíveis, os ACs aqui empregados são todos reversíveis. Em algum momento estes ACs retornarão à configuração do reticulado inicial após um número indeterminado de passos de tempo, formando então um ciclo. O ciclo possui fundamental importância nos modelos de sistemas criptográficos apresentados na seção 4.2, que utilizam as propriedades algébricas dos ACs aditivos sobre as quais tem-se um controle sobre o ciclo que o AC irá operar. Diferentemente dos métodos discutidos na seção 4.2, os modelos aqui propostos empregam tanto a evolução para frente quanto a evolução para trás (através do cálculo de pré-imagem), além de não possuírem a característica de serem aditivos.

De fato, os ACs empregados nos métodos aqui apresentados possuem a característica

de formar um ciclo, sendo assim, uma das preocupações passou a ser qual o tamanho do ciclo que estes ACs possuem. Se este ciclo fosse muito pequeno, poderia ocasionalmente ocorrer que um reticulado inicial, após ser evoluído por um determinado número de pré-imagens, pare exatamente no reticulado inicial, portanto, deixando o texto cifrado igual ao texto claro. Experimentos acerca do tamanho dos ciclos serão apresentados no próximo capítulo de forma detalhada.

Assim como no modelo de Oliveira e colaboradores (2004), o problema relacionado à propagação da perturbação apenas em uma direção, encontrado ao executar uma etapa do método de Gutowitz (Figura 27 (c)), não ocorre no método aqui proposto, desde que seja utilizada uma quantidade de evoluções o suficiente para que a perturbação se propague por todo o reticulado (Figura 34 (c), Figura 36 (c) e Figura 39 (c)). Serão apresentados experimentos que analisam a propagação da perturbação, de uma forma quantitativa, no próximo capítulo.

Conforme dito na seção 5.3.2, existe um número muito reduzido de regras para a região de contorno. Sendo assim, um criptoanalista poderia testar todas as regras de contorno possíveis. Com base nessa hipótese, considere que a regra de contorno é conhecida por um criptoanalista e o mesmo pretende descobrir alguma característica do texto claro ou da regra principal. O Apêndice A mostra que se o número de passos de cálculo de pré-imagem for acima de $(2 \times raio) + 1$, nenhuma informação sobre os bits do reticulado podem ser recuperadas, tal que a regra principal é mantida secreta e a regra de contorno é conhecida.

A primeira versão possui a vantagem de ser bem simples, tendo como desvantagem em relação às demais versões, a impossibilidade de alcançar paralelismo. Além disso, a última versão possui a vantagem de utilizar várias regras na evolução do cálculo de pré-imagem. Como veremos no próximo capítulo, essa vantagem torna a última versão capaz de aumentar o tamanho do ciclo mínimo.

Outra preocupação ao se analisar os métodos aqui propostos foi tentar verificar o quão aleatório são os textos cifrados produzidos por estes métodos. Experimentos que tratam desse aspecto também serão discutidos mais à frente. Para várias das análises que realizamos em relação aos modelos propostos, foi necessário utilizar uma medida que quantificasse a aleatoriedade de zeros e uns em uma seqüência binária. Por exemplo, a Tabela 12 apresenta vários exemplos de seqüências binárias de 16 bits. Claramente, as quatro primeiras seqüências da figura têm uma estrutura geral muito ordenada, enquanto as quatro últimas têm uma

estrutura bem mais desordenada.

A medida mais usual para identificar aleatoriedade em uma seqüência de eventos é a entropia [Shannon, 1948]. A entropia de uma seqüência de *k* eventos é definida pela equação (14), onde p_i é a probabilidade de ocorrência do evento *i*:

$$S = -\sum_{i=1}^{k} p_i \times \log_2 p_i \tag{14}$$

Tabela 12 - Seqüências binárias ordenadas e desordenadas com seus os valores de ***S*** e ***s***.

Seqüências ordenadas		
Seqüência binária	***S***	***s***
0101010101010101	1,000000	0.250000
0000000100000000	1,311278	0.327820
0110000000000000	1,621641	0.405410
0011001100110011	2,000000	0.500000
Seqüências desordenadas		
Seqüência binária	***S***	***s***
0111010101000011	3,452820	0.863205
1101001011000101	3,625000	0.906250
0111110101001000	3,625000	0.906250
1010110000100111	4,000000	1.000000

Assim, dentre o conjunto de eventos, se apenas alguns deles possuem alta probabilidade de ocorrência, o valor de *S* será baixo, e quanto maior for a variação na probabilidade dos eventos, maior o valor de *S*.

Para adaptar essa medida ao nosso propósito, definimos a entropia espacial de uma palavra binária de *N* bits como sendo a entropia da ocorrência de *N* janelas de tamanho *j*, com $j < N$. Por exemplo, na Figura 40 usamos uma palavra binária de 16 bits e uma janela de 4 bits ($N = 16$ e $j = 4$). No total, são 16 janelas de 4 bits que devem ser analisadas. Usaremos como a probabilidade de ocorrência da janela *i*, o número de ocorrências da mesma nas 16 janelas analisadas. No caso da seqüência binária {0100101110011101} apresentada na figura, ocorrem 11 janelas diferentes, distribuídas como indicado pela Tabela 13.

Tabela 13- Ocorrências de janelas de tamanho 4 encontradas na seqüência {0100101110011101}.

Janela	Quantidade de Ocorrências	Janela	Quantidade de Ocorrências
0000	0	1000	0
0001	0	1001	2
0010	1	1010	2
0011	1	1011	1
0100	1	1100	1
0101	2	1101	1
0110	0	1110	2
0111	2	1111	0

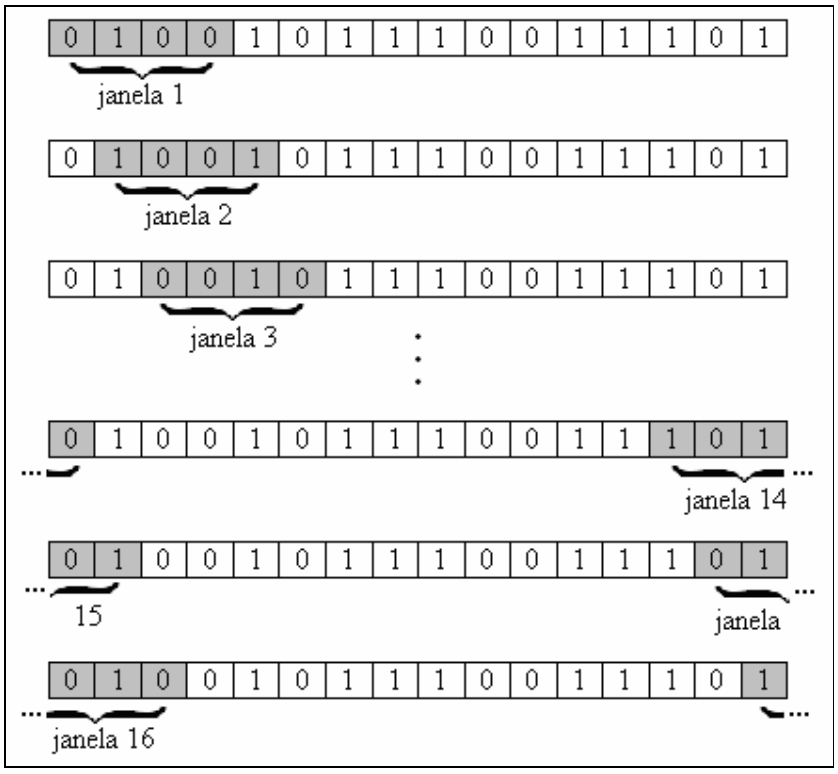


Figura 40 - Janelas de 4 bits em um palavra binária de 16 bits.

Assim, de acordo com a Tabela 13 a entropia espacial é dada pela equação (15):

$$S = -\sum_{i=1}^{N} p_i \times \log_2 p_i = -\begin{pmatrix} \left(\frac{1}{16}\log_2\frac{1}{16}\right)+\left(\frac{1}{16}\log_2\frac{1}{16}\right)+\left(\frac{1}{16}\log_2\frac{1}{16}\right)+\left(\frac{2}{16}\log_2\frac{2}{16}\right)+ \\ \left(\frac{2}{16}\log_2\frac{2}{16}\right)+\left(\frac{2}{16}\log_2\frac{2}{16}\right)+\left(\frac{2}{16}\log_2\frac{2}{16}\right)+\left(\frac{1}{16}\log_2\frac{1}{16}\right)+ \\ \left(\frac{1}{16}\log_2\frac{1}{16}\right)+\left(\frac{1}{16}\log_2\frac{1}{16}\right)+\left(\frac{2}{16}\log_2\frac{2}{16}\right) \end{pmatrix} = 3{,}375 \quad (15)$$

A definição do tamanho da janela pode ser escolhida arbitrariamente. Entretanto, para

que fosse possível normalizar o valor da entropia (entre 0 e 1), para qualquer tamanho de palavra binária, estabelecemos que o tamanho da janela é definido como o número de bits *j* tal que o número de janelas possíveis de ocorrer na palavra seja igual a *N*. Assim, o tamanho da janela utilizada para a análise de uma palavra de tamanho *N* é dada pela equação (16).

$$j = \log_2 N \tag{16}$$

Como cada janela é formada por *j* bits, o número máximo de janelas diferentes que podem ocorrer é *N*. Nesse caso, a entropia máxima é dada por uma palavra que possui 1 ocorrência de cada uma das *N* janelas. Assim, $\forall i,\ p_i = \frac{1}{N}$ e a entropia é dada pela equação (17).

$$S = -\left(\frac{1}{N}\log_2\frac{1}{N} + \frac{1}{N}\log_2\frac{1}{N} + \frac{1}{N}\log_2\frac{1}{N} + ...\right) = -\sum_{i=1}^{N}\frac{1}{N}\log_2\frac{1}{N} \tag{17}$$

Como $N = 2^j$, temos:

$$S = -\sum_{i=1}^{N}\frac{1}{N}\log_2\frac{1}{2^j} = -\left(N\frac{1}{N}(-j)\right) = j \tag{18}$$

Por outro lado, a palavra com entropia mínima é aquela que possui uma única janela *k* que ocorre *N* vezes. Nesse caso, $p_i = \frac{N}{N} = 1$ para i = k e $p_i = 0$ para $i \neq k$.

$$S = -\frac{N}{N}\log_2\frac{N}{N} = -1\log_2 1 = 0 \tag{19}$$

Assim, a forma da entropia normalizada é dada pela equação (20) para $j = \log_2 N$ e sendo *p* a probabilidade de ocorrência de uma janela, que é dado pelo número de ocorrências da janela dividido por *N*.

$$s = \frac{-\sum_{i=1}^{N} p_i \log_2 p_i}{j} \tag{20}$$

Quanto maior o valor da entropia (*s)*, maior é a garantia de que o reticulado não possui alguma regularidade na disposição dos bits. Por exemplo, a Tabela 12 apresenta os valores da

entropia não normalizada (S) e da entropia normalizada (s) para cada uma das palavras binárias de 16 bits, considerando uma janela de 4 bits. (N = 16 e j = 4). Através de diversos testes, percebemos que cadeias com características visualmente aleatórias, apresentam entropia acima de 0,8.

Capítulo 6

Experimentos e resultados

No Capítulo 5 foram apresentados três modelos de como evoluir para trás um AC não-homogêneo, sem a necessidade de se acrescentar bits adicionais a cada passo de cálculo da pré-imagem. Neste capítulo, são apresentados os resultados de alguns experimentos investigando a possibilidade de aplicação de dois destes modelos (básico e com rotação na borda e no núcleo) para a criptografia. A partir desse ponto o "modelo com rotação" refere-se à terceira versão (rotação na borda e no núcleo). Os resultados para o modelo com rotação apenas na borda não são apresentados pelo fato que a modificação em relação ao seu predecessor difere somente na possibilidade de execução em paralelo, portanto, não influenciando diretamente no comportamento dinâmico do AC, sendo que seus resultados são bem próximos daquele operando sequencialmente (modelo básico).

6.1 Análise de ciclo

Em todas as versões do cálculo de pré-imagem aqui propostas, apenas uma pré-imagem é encontrada para qualquer que seja o reticulado inicial. Uma vez que são efetuados cálculos consecutivos de pré-imagens, a configuração do reticulado inicial será encontrada novamente após alguns ou vários passos do cálculo, formando assim um ciclo. A quantidade de passos executadas até que o reticulado inicial se repita é chamado de tamanho do ciclo. A importância de se analisar o tamanho do ciclo deve-se ao fato de que, caso a quantidade de evoluções estipulada pelo sistema criptográfico seja a mesma do tamanho de algum ciclo, a configuração de um reticulado inicial será a mesma da configuração do reticulado final. Além disso, se o tamanho de um ciclo for menor ou múltiplo da quantidade de evoluções estipulada pelo sistema criptográfico, também pode ocorrer a situação em que os reticulados iniciais e finais são os mesmos.

6.1.1 Análise de ciclo para o modelo básico

A primeira versão do método para cálculo de pré-imagem foi analisada para regras de raio 1, 2, 3 e 4, aplicadas a todos os reticulados de tamanho 16, 20 e 22 bits, além de 200 amostras de reticulados com tamanho 32, 64 e 128 bits.

Para cada regra analisada, foi verificado o tamanho de ciclo mínimo aplicado aos tamanhos de reticulados estabelecidos anteriormente. Ou seja, cada regra foi submetida a reticulados de 16, 20, 22, 32, 64 e 128 bits, verificando o ciclo mínimo daquela regra em relação ao tamanho do reticulado.

A primeira análise foi feita utilizando-se todas as regras de raio 1 com a característica da sensitividade à esquerda ou à direita, que totalizam 32 regras. Inicialmente, essas regras foram aplicadas e avaliadas utilizando-se todos os reticulados possíveis de 16 bits (2^{16} reticulados), todos os reticulados de 20 bits (2^{20} reticulados) e todos os reticulados de 22 bits (2^{22} reticulados). Posteriormente elas foram aplicadas a amostras de 200 reticulados de tamanho 32 bits, 200 reticulados de 64 bits e 200 reticulados de 128 bits. A abordagem de amostras foi utilizada devido à impraticabilidade de se fazer uma análise completa nesses espaços de reticulados ($2^{32}, 2^{64}, 2^{128}$).

Os resultados para as 32 regras de raio 1, aplicadas a todo o espaço de reticulados de tamanho 16, 20 e 22 bits, apresentaram em sua maioria o ciclo mínimo igual a 1. Ou seja, para a maioria das regras de raio 1 existe pelo menos um reticulado inicial de 16 bits, um reticulado inicial de 20 bits e um reticulado inicial de 22 bits, que exibe um ciclo unitário, onde a pré-imagem é o próprio reticulado inicial. Para os reticulados de 16 bits, 28 regras apresentaram ciclo mínimo igual a 1. Dentre os reticulados de 20 bits, 30 regras obtiveram ciclo igual mínimo a 1. Para reticulados de 22 bits, também 30 regras obtiveram ciclo mínimo igual a 1. Posteriormente, foram avaliados os reticulados de tamanho 32, 64 e 128 (amostras), o ciclo mínimo unitário teve uma menor freqüência, apesar de ocorrer em diversas regras. Este resultado foi justificado ao analisar algumas destas regras e constatar que de fato existem algumas configurações de reticulados que utilizam poucos bits das regras, ou ainda, existem regras que não possuem característica dinâmica caótica.

Dentre as regras que não possuem característica caótica já era esperada a existência de ciclos pequenos, porém existem outros casos em que as regras são caóticas e também foram encontrados ciclos unitários. Para exemplificar esta situação, considere um reticulado inicial formado apenas por 0s. Apesar da regra de contorno ser diferente da regra principal, em diversos casos, isso não é suficiente para que a regra principal exiba seu comportamento caótico, pois apenas um bit da regra é empregado de acordo com a vizinhança formada por 0s. Um outro exemplo que explora a vizinhança das regras pode ser dado pela seguinte configuração de um reticulado: 0101...01. Esta configuração apresenta apenas dois tipos de

vizinhança. Por exemplo, considere um AC de raio 1 com a configuração do reticulado inicial citado anteriormente. Neste caso só existe a vizinhança 010 e 101, portanto, apenas dois bits da regra principal serão utilizados.

Posteriormente, regras de raio 2, 3 e 4 foram submetidas a uma avaliação similar, obtendo-se os ciclos mínimos nos conjuntos formados por todos os reticulados de 16, 20 e 22 bits, além de serem avaliadas 200 amostras de reticulados de tamanho 32, 64 e 128 bits. Entretanto, devido ao espaço de regras (2×2^{16} para raio 2, 2×2^{64} para raio 3 e 2×2^{256} para raio 4), também foram avaliadas apenas amostras de 100 regras para cada raio. A Tabela 14 apresenta a quantidade de regras que possuem ciclo mínimo igual a 1, para os raios 2, 3 e 4 de acordo com o tamanho do reticulado (*N*).

Tabela 14 - Quantidade de regras com ciclo mínimo igual a 1.

Raio	**N = 16**	**N = 20**	**N = 22**	**N = 32**	**N = 64**	**N = 128**
2	92	80	89	20	18	18
3	82	74	82	18	18	18
4	75	78	59	20	16	16

Note na Tabela 14 que os resultados para reticulados de 32, 64 e 128 apresentaram um menor número de regras com ciclos mínimo igual a 1. É importante salientar que para estes reticulados são empregadas apenas 200 amostras de reticulados, além de ser usada uma amostra de 100 regras para cada tamanho de raio. Os resultados dos experimentos com reticulados de tamanho 16, 20 e 22, apesar de usarem uma amostra de 100 regras, empregam todos os reticulados possíveis para cada tamanho e, portanto, representam melhor a quantidade de regras com ciclo mínimo igual a 1. É possível constatar que a maioria das regras possui pelo menos um ciclo unitário para algum reticulado inicial.

Embora, para cada regra analisada, existam poucos reticulados iniciais que possuem o tamanho de ciclo pequeno, para a maioria das regras existe pelo menos um reticulado que possui ciclo unitário. Assim, a primeira versão do cálculo de pré-imagem é muito vulnerável à configuração do reticulado inicial pelo fato de que apenas uma regra principal é empregada no cálculo de pré-imagem.

6.1.2 Análise de ciclo para o modelo com rotação

A análise realizada na primeira versão mostrou que para grande maioria das regras existem alguns reticulados que possuem ciclos muito pequenos, inclusive de tamanho 1. Os

resultados obtidos para o modelo com rotação são apresentados nesta seção.

Para raio 1 (núcleo de 4 bits) e 2 (núcleo de 16 bits) foram analisadas todas as regras possíveis (2×2^4 núcleos de raio 1 e 2×2^{16} núcleos de raio 2), porém, para raio 3 e 4 foram analisadas apenas amostras de 100 núcleos, uma vez que a quantidade de núcleos possíveis para estes raios é muito grande.

Como visto na seção 5.6, nesse modelo de cálculo de pré-imagem, para cada passo de execução é gerada uma regra de acordo com um núcleo tido como chave criptográfica. Dessa forma, quando o cálculo de pré-imagem é executado, o tamanho do ciclo é determinado ao encontrar novamente o reticulado inicial e a regra inicial. As regras que exibem um ciclo menor do que o tamanho do seu núcleo podem representar um risco para o sistema criptográfico, fazendo com que o texto cifrado tenha pouca ou nenhuma diferença do texto claro.

6.1.2.1 Análise para Raio 1

Todas os núcleos de raio 1 foram testados, totalizando 32 núcleos (2^4 com sensitividade à esquerda e 2^4 com sensitividade à direita). Para cada um destes núcleos foram calculados consecutivamente todos os reticulados possíveis de 16 bits, 20 bits, 22 bits e 24 bits, além de uma amostra de 200 reticulados de 32, 64 e 128 bits.

Dentre todos os núcleos analisados para o espaço dos reticulados de 16 bits, a Figura 41 mostra aqueles que obtiveram ciclo mínimo menor do que a quantidade de bits que compõem o seu núcleo, apresentando o tamanho mínimo do ciclo encontrado. Todos os demais núcleos (28 no total), apresentaram ciclos mínimos com tamanho maior ou igual a 4, que equivale à quantidade de bits formada pelo núcleo e, portanto, ao número de regras diferentes utilizadas no cálculo de pré-imagem.

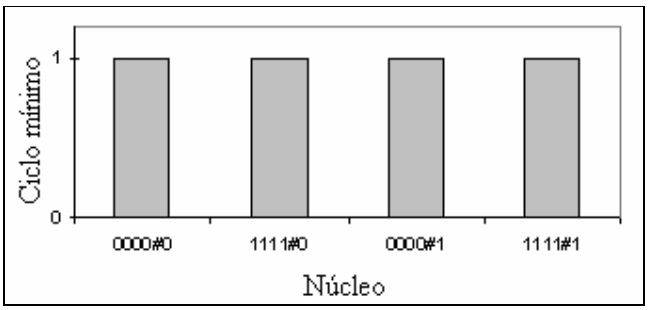


Figura 41 - Ciclo mínimo abaixo de 4 para núcleos de raio 1 com reticulados de tamanho 16 bits.

A Figura 42 indica os núcleos que apresentaram tamanho de ciclo inferior a 4, juntamente com o tamanho mínimo de ciclo encontrado para todos os reticulados de tamanho 20 bits. Todos os outros núcleos (26 no total), apresentam ciclos com tamanho maior ou igual a 4.

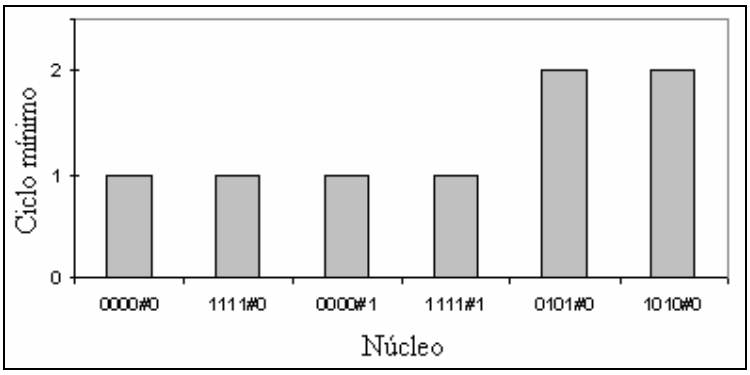


Figura 42 - Ciclo mínimo abaixo de 4 para núcleos de raio 1 com reticulados de tamanho 20 bits.

Na Figura 43, são apresentados os núcleos com ciclo abaixo de 4, juntamente com o tamanho mínimo de ciclo encontrado para todos os reticulados de tamanho 22 bits. Todos os outros núcleos (28 no total), apresentam ciclos com tamanho maior ou igual a 4.

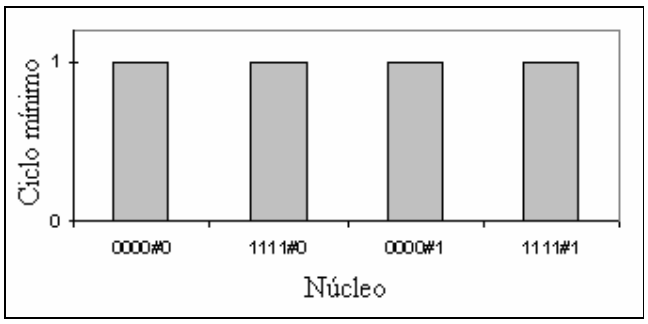


Figura 43 - Ciclo mínimo abaixo de 4 para núcleos de raio 1 com reticulados de tamanho 22 bits.

A Figura 44 apresenta o resultado para o experimento com reticulados de tamanho 24 bits. Os núcleos com ciclo mínimo abaixo do tamanho do núcleo são mostrados, com os seus valores de ciclo mínimo encontrado. Todos os outros núcleos (28 no total), apresentam ciclos com tamanho maior ou igual a 4.

Ao analisar os resultados da Figura 41, Figura 42, Figura 43 e Figura 44 pode-se notar que os núcleos 0000 e 1111, tanto com sensitividade à esquerda quanto à direita, obtiveram tamanho de ciclo mínimo igual a 1 para pelo menos um dos reticulados analisados. Uma vez que o método utiliza rotações do núcleo e estes núcleos são formados apenas por 0s ou 1s, todas as regras acabam sendo as mesmas na evolução de cada passo do cálculo de pré-

imagem. Além disso, os núcleos formados somente por 0s ou 1s não geram regras caóticas, fazendo com que o reticulado final possa apresentar características do reticulado inicial. Já os núcleos 0101 e 1010 obtiveram ciclo 2 para regras sensíveis à esquerda nos resultados do experimento para reticulados de 20 bits. Note que, ao sofrerem duas rotações, estes núcleos retornam ao núcleo inicial. Núcleos com esta característica devem ser evitados e serão discutidos mais adiante.

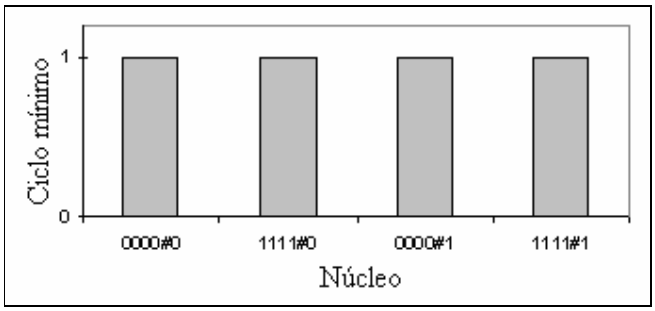


Figura 44 - Ciclo mínimo abaixo de 4 para núcleos de raio 1 com reticulados de tamanho 24 bits.

A Tabela 15 contém os resultados realizados com amostras de 200 reticulados de tamanho 32, 64 e 128 bits apresentando a quantidade de núcleos que possuem o ciclo menor que o tamanho do núcleo (ou seja, 4 bits).

Tabela 15 - Quantidade de núcleos que obtiveram ciclo menor do que o tamanho de seu núcleo para raio 1 com amostras de reticulados de 32, 64 e 128 bits.

Tamanho do reticulado	**Regras com ciclo menor que o núcleo**
32	0
64	0
128	0

Os resultados da Tabela 15 indicam que na análise de todos os núcleos de raio 1, nenhum deles obteve ciclo menor que o tamanho de seus núcleos. Porém, é muito provável que existam reticulados que façam com que aqueles núcleos apresentem ciclos pequenos, uma vez que os resultados para esta tabela utilizam uma amostra de 200 reticulados para cada um dos tamanhos avaliados.

Embora os resultados apresentados acima possam parecer preocupantes, os núcleos que apresentam a característica de possuir o ciclo pequeno são a minoria, e sua proporção diminui consideravelmente com o aumento do tamanho do raio, conforme veremos a seguir.

6.1.2.2 Análise para Raio 2

Todos os núcleos de raio 2 foram testados, sendo $2^{16} = 65.536$ sensíveis à esquerda e $2^{16} = 65.536$ sensíveis à direita, totalizando 131.072 núcleos. Para cada uma destes núcleos, foram calculados consecutivamente todos os reticulados possíveis para tamanho 12, 16 e 20 bits, além de uma amostra de 200 reticulados para 32, 64 e 128 bits.

A Figura 45 apresenta o ciclo mínimo dentre os núcleos que apresentam um ciclo menor que o tamanho do núcleo (nesse caso, 16 bits) para algum dos 4.096 reticulados de tamanho 12 bits. No total, são 352 núcleos, sendo que a maioria deles possui ciclo mínimo de tamanho 8. Esse total representa 0,27% de todos os núcleos possíveis de 16 bits (131.072).

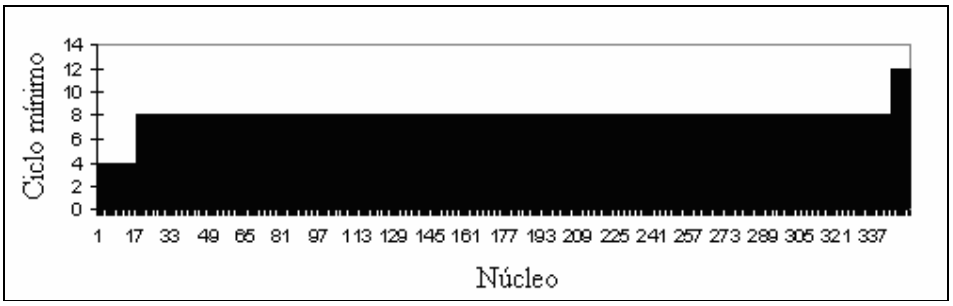


Figura 45 - Ciclo mínimo abaixo de 16 para raio 2 com reticulados de tamanho 12 bits.

A Figura 46 apresenta o ciclo mínimo dos núcleos de raio 2 que obtiveram ciclo menor do que o tamanho do núcleo para algum dos 65.536 reticulados de tamanho 16 bits. A figura apresenta 310 núcleos, onde a maioria possui ciclo mínimo de tamanho 8, totalizando 0,23% do espaço de núcleos.

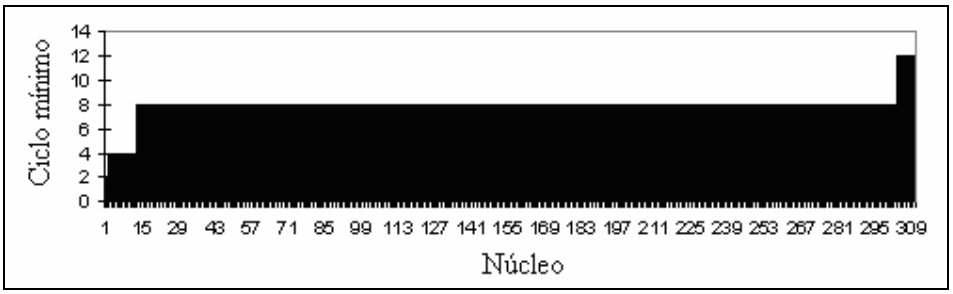


Figura 46 - Ciclo mínimo abaixo de 16 para raio 2 com reticulados de tamanho 16 bits.

A Figura 47 apresenta o resultado do experimento dos núcleos de raio 2 que obtiveram ciclo menor do que o tamanho do núcleo para algum dos 1.048.576 reticulados de tamanho 20 bits. Nesta figura estão 256 núcleos, onde a maioria possui ciclo mínimo de tamanho 8, totalizando 0,19% do espaço de núcleos.

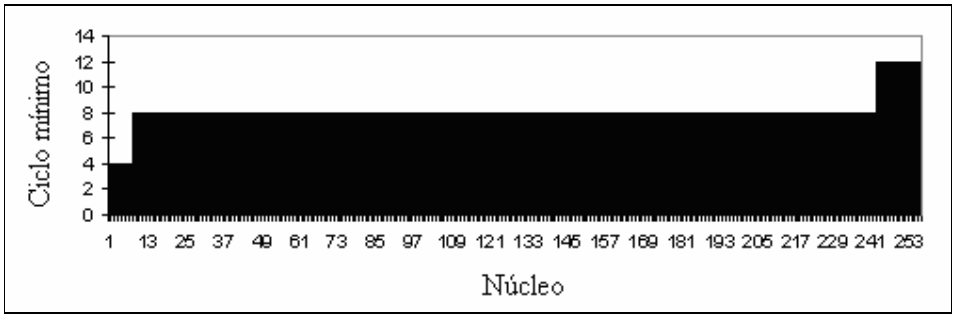


Figura 47 - Ciclo mínimo abaixo de 16 para raio 2 com reticulados de tamanho 20 bits.

Os resultados da Figura 45, Figura 46 e Figura 47 são bem próximos, porém, apresentam uma tendência de que, com o aumento do tamanho do reticulado, a quantidade de núcleos que apresentam ciclo mínimo abaixo do tamanho do núcleo tende a diminuir.

A Tabela 16 contém os resultados realizados com amostras de reticulados de tamanho 32, 64 e 128 bits, apresentando a quantidade de núcleos que possuem o ciclo menor que o tamanho do núcleo.

Tabela 16- Quantidade de núcleos que obtiveram ciclo menor do que o tamanho de seu núcleo para raio 2 com amostras de reticulados de 32, 64 e 128 bits.

Tamanho do reticulado	Regras com ciclo menor que o núcleo
32	0
64	0
128	0

A análise da Tabela 16 é similar à análise da Tabela 15. Ou seja, é muito provável que exista algum reticulado que faça com que algum núcleo exiba um ciclo abaixo do tamanho do núcleo. Entretanto, como o conjunto de amostras é pequeno em relação à quantidade de reticulados possíveis, esse resultado não foi encontrado.

6.1.2.3 Análise para Raio 3 e 4

Devido ao grande número de núcleos para raio 3 e 4, foram realizados experimentos com 100 amostras destes núcleos, sendo 50 sensíveis à esquerda e 50 sensíveis à direita. Foram feitos testes com todos os reticulados de tamanho 16, 20, 22 e 24 bits, além de 200 amostras de 32, 64 e 128 bits. Nenhum dos testes detectou ciclos menores do que o tamanho do núcleo analisado. Entretanto, é muito provável que existam núcleos que para algum reticulado possa obter um ciclo menor que o tamanho do núcleo.

6.1.3 Conclusão da análise de ciclo para o modelo com rotação

No raio 1, 12,5%, 18,75%, 12,5% e 12,5% dos núcleos avaliados com reticulado de tamanho 16, 20, 22 e 24 bits, respectivamente, obtiveram ciclo abaixo do tamanho do núcleo. No raio 2, 0,27%, 0,23% e 0,19% dos núcleos apresentaram ciclo menor que o tamanho do núcleo com reticulados de 12, 16 e 22 bits. Dessa forma, fica claro que a quantidade de núcleos que apresentaram um ciclo problemático (do ponto de vista da criptografia) é a minoria, tendendo a ser cada vez menor com o aumento do raio.

Contudo é muito provável que, para qualquer tamanho de raio, exista um núcleo que faça com que o ciclo de um reticulado seja menor do que a quantidade de bits do seu núcleo, portanto, colocando em risco a informação a ser protegida pelo sistema criptográfico. Dessa forma, é necessário definir qual tipo de núcleo pode ser utilizado na geração da regra e quais devem ser evitados no processo de cifragem. Outros testes foram realizados através da análise da entropia no núcleo, revelando que núcleos com baixa entropia possuem influência nos resultados do tamanho do ciclo. Estes resultados são apresentados a seguir.

6.1.4 Entropia do núcleo e o tamanho do ciclo

Ao executar o cálculo da entropia espacial, apresentado na seção 5.7, nos núcleos que obtiveram ciclos menores que o seu próprio tamanho, foi possível notar que estes núcleos possuem entropia abaixo de 0,75 em todos os raios analisados. Isto significa que estes núcleos possuem algum tipo de regularidade. Esta suspeita foi comprovada ao verificar cada um destes núcleos. Como exemplo, observe os núcleos apresentados na Tabela 17. É possível notar que o primeiro núcleo apresentado possui ciclo mínimo de tamanho igual a 8 ao ser avaliado em reticulados de 16 bits, que é exatamente o número de rotações dos bits do núcleo até que haja uma repetição na geração de uma regra. Ou seja, se este núcleo for dividido ao meio, cada metade será exatamente igual. A análise é similar para o segundo núcleo apresentado, porém, a rotação no segundo caso é de 4 posições até que o núcleo seja o mesmo núcleo inicial. Esta observação ocorre para todos os núcleos que apresentaram problemas na análise de ciclo. A tabela também apresenta o valor da entropia espacial da seqüência binária que representa o núcleo.

Tabela 17 - Exemplo de núcleos de raio 2 que retornam ao núcleo inicial após alguns passos de rotação.

Raio 2		
Núcleo	**Ciclo Mínimo**	**Entropia do Núcleo**
0111010101110101#0	8	0,63
0111011101110111#0	4	0,50

A Figura 48 apresenta o diagrama espaço-temporal referente à evolução das regras da Tabela 17 com um dos seus respectivos reticulados iniciais de 16 bits que obtiveram ciclos pequenos. Os reticulados iniciais estão localizados na base da figura, sendo que cada repetição do reticulado inicial está marcado e indicado por uma seta. Nos dois exemplos a evolução do cálculo de pré-imagem apresentava uma dinâmica caótica até que começaram a se repetir.

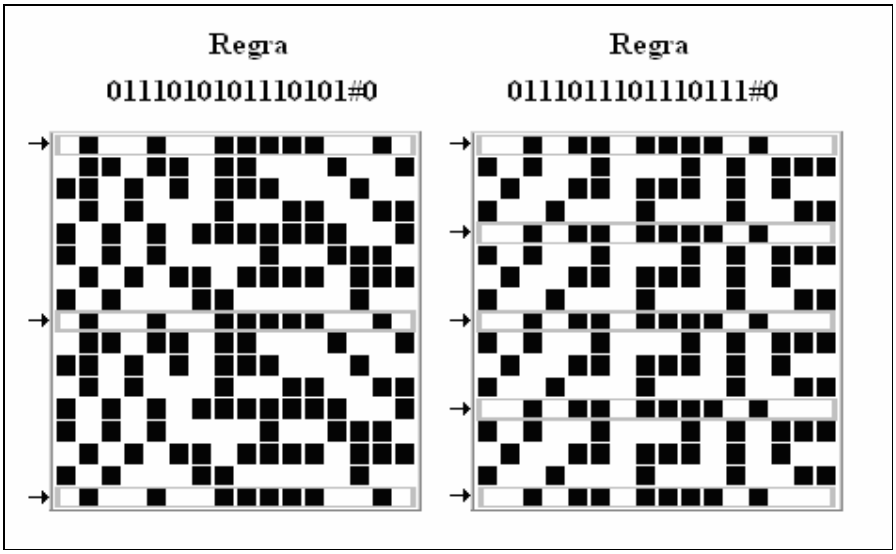


Figura 48 - Diagrama espaço-temporal para os núcleos indicados na Tabela 17.

Os experimentos realizados com a entropia espacial indicam que todos os núcleos que obtiveram problemas no tamanho do ciclo possuem entropia abaixo de 0,75. Por isso, foi analisada a possibilidade de excluir este conjunto núcleos do espaço de chaves.

A Tabela 18 apresenta a quantidade de núcleos para cada faixa de entropia obtida, considerando-se todos os núcleos de raio 2. Note que a quantidade de núcleos abaixo de 0,75 de entropia é formada por 11.328 núcleos, de um total de 131.072 (8,64%). Ao empregar reticulados de tamanho 12 bits apenas 352 destes núcleos obtiveram um resultado insatisfatório; para reticulados de tamanho 16 bits, o número de núcleos insatisfatórios é de 310 regras, e para reticulados de 20 bits, apenas 256 núcleos obtiveram um ciclo menor que o

tamanho do núcleo. Embora vários núcleos com entropia abaixo de 0,75 não tenham apresentado ciclo mínimo pequeno (abaixo do tamanho do próprio núcleo), outros resultados serão apresentados mais adiante reforçando a hipótese de se descartar os núcleos com entropia inferior a 0,75.

Tabela 18- Distribuição da quantidade de núcleos para raio 2, de acordo com as faixas de entropia.

Faixa de entropia "s"	Quantidade de núcleos
$s < 0,5$	164
$0,5 \leq s < 0,6$	876
$0,6 \leq s < 0,7$	4912
$0,7 \leq s < 0,75$	5376
$0,75 \leq s < 0,8$	12064
$s \geq 0,8$	42144

6.2 Análise da entropia no texto cifrado

Qualquer que seja o método de criptografia empregado, é desejável que a etapa de cifragem seja capaz de "embaralhar" a mensagem original o suficiente para que nenhuma semelhança com a mesma seja encontrada.

Para quantificar essa capacidade de "embaralhamento" dos métodos de cálculo de pré-imagem propostos nessa dissertação, resolvemos lançar mão mais uma vez da medida de entropia espacial, apresentada na seção 5.7.

Utilizamos nessa análise os métodos de cálculo de pré-imagem a partir de duas amostras de reticulados iniciais. A primeira amostra contém reticulados com entropia abaixo de 0,5, sendo portanto, caracterizados por reticulados com alta regularidade. Por exemplo: 00000000, 01010101, 00100000, etc. A segunda amostra contém reticulados com entropia acima de 0,7 caracterizados por reticulados com grande aleatoriedade. A idéia central dessa análise é a seguinte: se o embaralhamento do método for bom, a entropia final dos reticulados após a cifragem deve ser similar, tanto para a amostra de baixa entropia inicial, quanto para a amostra de alta entropia inicial. Em ambos os casos, é desejável que os reticulados finais tenham um valor de entropia alto, em média. Nestes experimentos foram testadas todas as regras de raio 1 e 100 amostras de regras de raio 2, 3 e 4. Os testes foram realizados com reticulados iniciais de 16, 32, 64 e 128 bits com 100 amostras de baixa entropia, e outras 100 amostras geradas aleatoriamente (por serem aleatórios tendem a uma alta entropia), para cada

um dos tamanhos de reticulados. Apenas os reticulados de tamanho 16 bits com baixa entropia possuem seu conjunto formado por 50 reticulados iniciais, pois para este tamanho de reticulado não foi possível formar um conjunto de 100 palavras de 16 bits com entropia abaixo de 0,5.

A Tabela 19 apresenta a faixa de entropia e a quantidade de reticulados iniciais para cada um dos tamanhos de reticulados analisados.

Para cada raio e amostra de reticulados iniciais, foram feitos experimentos com evoluções de 25%, 50%, 75% e 100% do tamanho do reticulado. Ou seja, nos experimentos com reticulados de tamanho 16 bits são executados 4, 8, 12 e 16 passos do cálculo de pré-imagem, para cada raio e amostra de reticulados iniciais. Nesta seção, serão apresentados e discutidos apenas os resultados para reticulados de tamanho 128 bits executando 128 cálculos de pré-imagens consecutivos (100% do tamanho reticulado). A título de exemplo, o Apêndice C apresenta todos os resultados empregando o modelo básico em regras de raio 1 e todos os resultados empregando o modelo com rotação em reticulado de 128 bits. O 0 apresenta alguns exemplos dos reticulados utilizados nos testes.

Tabela 19 - Quantidade de reticulados iniciais analisados e suas faixas de entropia de acordo com o tamanho do reticulado em bits.

Reticulados Iniciais com Baixa Entropia		
Tamanho do Reticulado	**Quantidade de Reticulados**	**Faixa de entropia "*s*"**
16	50	$0{,}25 \leq s \leq 0{,}5$
32	100	$0{,}3 < s < 0{,}4$
64	100	$0{,}15 < s < 0{,}25$
128	100	$0{,}09 < s < 0{,}14$
Reticulados Iniciais Escolhidos Aleatoriamente		
Tamanho do Reticulado	**Quantidade de Reticulados**	**Faixa de entropia "*s*"**
16	100	$0{,}6 < s \leq 1$
32	100	$0{,}73 < s < 0{,}95$
64	100	$0{,}75 < s < 0{,}94$
128	100	$0{,}84 < s < 0{,}92$

6.2.1 Análise da entropia no texto cifrado para o modelo básico

Os gráficos da Figura 49 e da Figura 50 apresentam o valor médio da entropia final para cada regra avaliada (eixo *x*), além das entropias mínimas e máximas, após a evolução por 128 passos de cálculo de pré-imagem, utilizando-se o modelo básico.

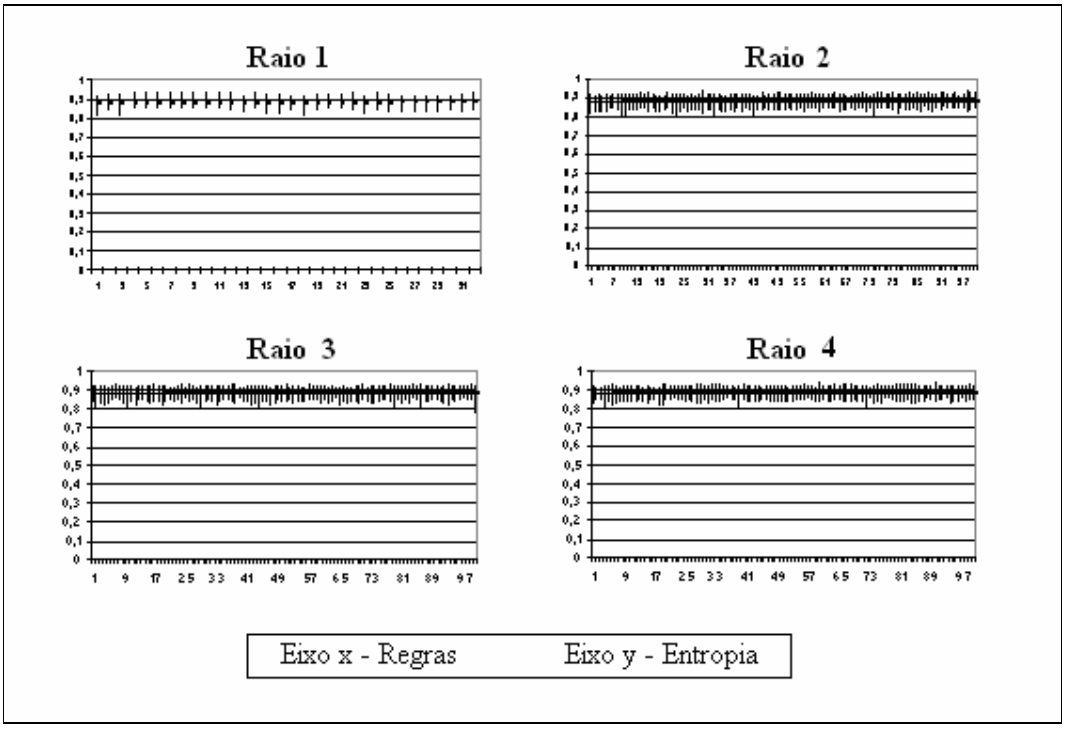


Figura 49 - Modelo básico: entropia média, mínima e máxima em reticulados finais para cada regra avaliada empregando reticulados iniciais aleatórios.

A Figura 49 refere-se ao experimento realizado com reticulados iniciais aleatórios (entropia inicial alta). A partir dos resultados apresentados é possível concluir que para reticulados iniciais aleatórios, todas as regras mantiveram a entropia alta após a evolução do cálculo de pré-imagem. Para os raios 2, 3 e 4 são empregadas apenas amostras destes conjuntos de regras e para raio 1 são testadas todas as regras com sensitividade à esquerda e à direita.

A Figura 50 mostra o resultado dos testes com reticulados iniciais de baixa entropia. É possível notar que várias regras de raio 1 e algumas regras de raio 2 não conseguiram fazer com que os reticulados de baixa entropia tenham um valor de entropia alto, após executado o cálculo de pré-imagem. Porém, já era esperado que algumas regras não fossem apropriadas para utilização em criptografia como, por exemplo, as regras que possuem comportamento dinâmico periódico. Além disso, os reticulados iniciais com baixa entropia aqui empregados acabam fazendo com que poucos bits da regra principal sejam efetivamente utilizados (como comentado na análise de ciclo) e, portanto, não são suficientes para obter um bom resultado, mesmo em regras com comportamento caótico. Este resultado não é tão aparente nos outros tamanhos de raio, pois estamos utilizando apenas algumas amostras de regras. Como existem poucas regras com comportamento periódico nos raios maiores, elas não costumam ser

geradas nas amostras, se estas são criadas de forma aleatória.

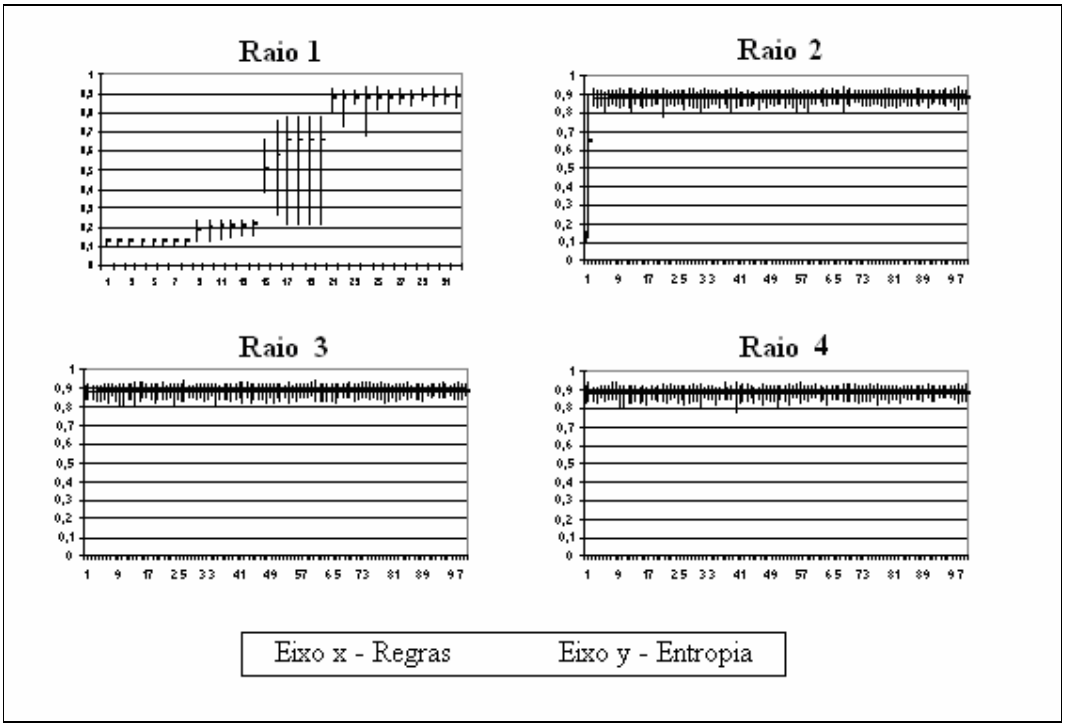


Figura 50 - Método básico: entropia média, mínima e máxima em reticulados finais para cada regra avaliada empregando reticulados iniciais com baixa entropia.

6.2.2 Análise da entropia no texto cifrado para o modelo com rotação

Para a última versão do cálculo de pré-imagem foram executados os mesmos testes realizados na primeira versão, inclusive com o mesmo conjunto de reticulados iniciais. Foram testados todos os núcleos para raio 1 e 100 amostras aleatórias de núcleos para os demais tamanhos de raio. Dentre os núcleos que foram gerados de forma aleatória, dois foram "preparados" com a seguinte característica: todos os bits do núcleo formados com o bit 0, exceto o último bit, sendo um dos núcleos sensível à esquerda e o outro sensível à direita. Por exemplo, para o raio 2 (núcleos com 16 bits) os núcleos: 0000000000000001#0 e 0000000000000001#1 foram incluídos na amostra de núcleos. Núcleos com esta característica de alto desbalanceamento no número de 0s e 1s tendem a ter comportamento periódico.

A Figura 51 apresenta os resultados para o experimento com reticulados iniciais aleatórios. Assim como na primeira versão, todos os núcleos mantiveram o valor de entropia do reticulado alto, após executar o cálculo de pré-imagem.

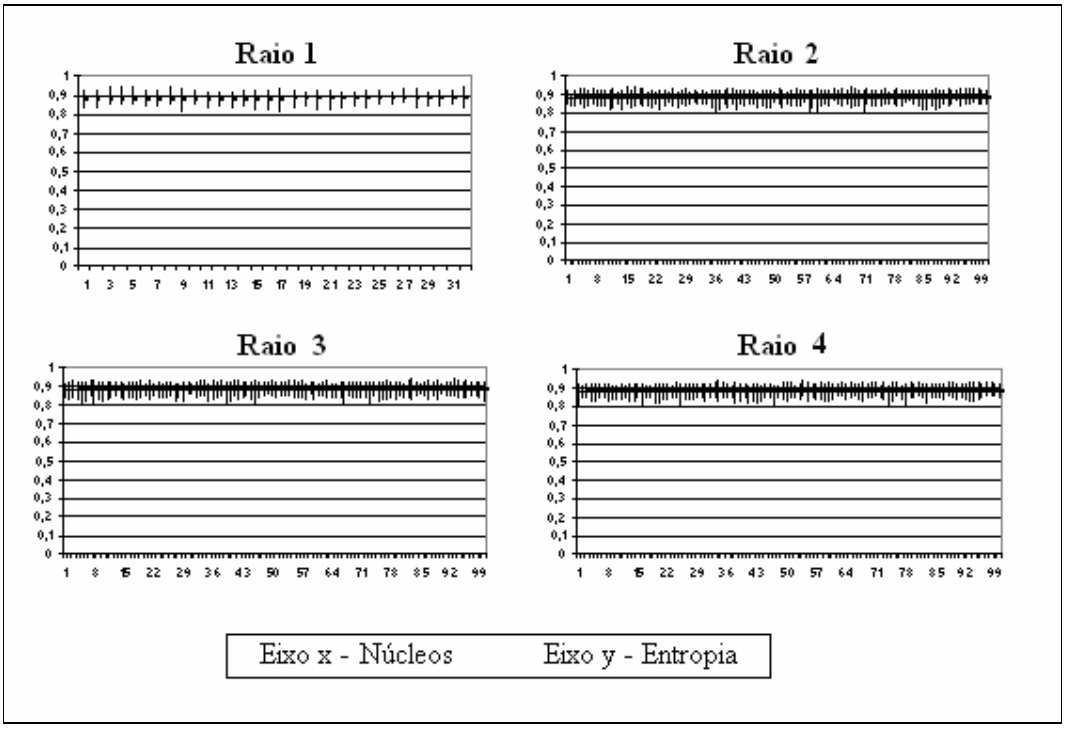


Figura 51 - Método com rotação: entropia média, mínima e máxima em reticulados finais para cada regra avaliada empregando reticulados iniciais aleatórios.

A Figura 52 mostra os resultados para os testes com reticulados iniciais de baixa entropia. Note que apenas núcleos de raio 1 apresentaram problemas em algumas regras. As 8 regras de raio 1 que possuem os valores médios de entropia mais baixos são: 0000#0, 1111#0, 0000#1, 1111#1, 1010#1, 0101#1, 0101#0 e 1010#0. É possível observar que todas estas regras possuem problemas ao rotacionar o núcleo, voltando ao núcleo inicial antes de completar 4 rotações. É possível notar também que no resultado para raio 4, dois núcleos obtiveram resultados inferiores aos demais. Estes núcleos são exatamente aqueles dois "preparados" da amostra de raio 4. Porém nos resultados de raio 2 e raio 3 estes núcleos tiveram bons resultados indicando que quanto maior o raio, mais significativas são as diferenças exibidas pelos núcleos com alto desbalanceamento. Mais adiante são efetuadas comparações entre o comportamento dinâmico dos núcleos com alto desbalanceamento e os aleatórios, comprovando que as regras geradas por núcleos desbalanceados exibem um comportamento dinâmico que as comprometem, pois tendem a ser periódicas.

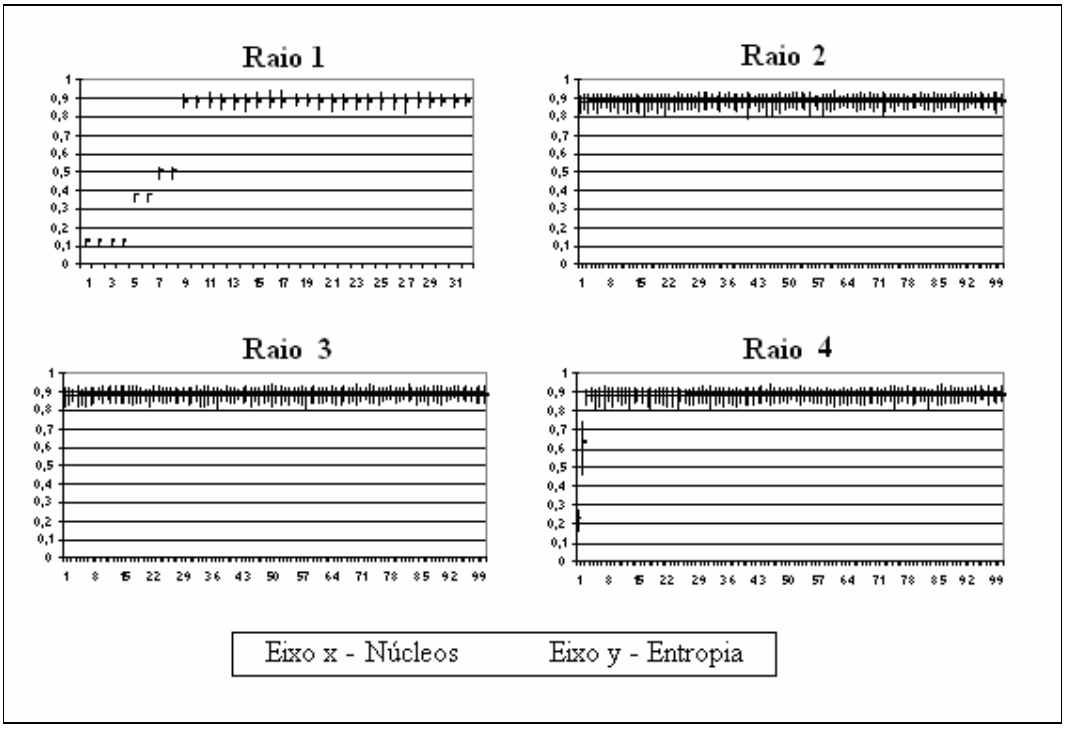


Figura 52 - Método com rotação: entropia média, mínima e máxima em reticulados finais para cada regra avaliada empregando reticulados iniciais com baixa entropia.

6.2.3 Conclusão da análise da entropia no texto cifrado

O resultado da análise de entropia no texto cifrado para as duas versões mostrou que nos experimentos em que os reticulados iniciais são aleatórios, todas as regras (núcleos) mantiveram os reticulados finais aleatórios após a execução do cálculo de pré-imagem. Porém, nem todas as regras (núcleos) foram capazes de provocar um valor de entropia alto em um reticulado inicial com valor de entropia baixo.

No método com rotação, todos os núcleos que não obtiveram um bom resultado, ou tinham a característica de retornar precocemente ao núcleo inicial em poucas rotações ou eram núcleos com alto desbalanceamento. Por outro lado, no modelo básico, existem regras com comportamento caótico que não possuem um bom resultado, isto pode ser explicado pelo fato de que este método utiliza apenas uma regra na evolução de todos os passos, tornando-o sensível a alguns reticulados iniciais (como foi explicado na análise de ciclo).

6.3 Análise da propagação de uma perturbação simples no reticulado inicial

Uma análise usual para se tentar identificar os pontos fortes e/ou fraquezas de um

método criptográfico é a análise da propagação de uma perturbação simples [Gutowitz, 1995], [Sen *et al*. 2002] e [Oliveira *et al*. 2004].

Nesta análise, inicialmente seleciona-se um texto claro *X* e seu respectivo texto cifrado *Y*. Posteriormente, é realizada uma pequena perturbação no texto claro *X* (tipicamente a complementação de um único bit do bloco) obtendo-se *X'*, que após ser cifrado obtém-se o novo texto cifrado *Y'*. Ao final, é analisada a diferença entre *Y* e *Y'*. Embora a diferença inicial entre *X* e *X'* seja pequena, é desejável que a diferença entre *Y* e *Y'* seja significativa. Dessa forma, o método não se torna vulnerável a ataques do tipo criptoanálise diferencial (seção 2.4.1).

A análise do reticulado final ao provocar uma perturbação de apenas um bit no reticulado inicial será apresentada com dois tipos de testes. Um deles é o cálculo do desvio padrão da diferença média entre dois textos cifrados *Y* e *Y'* em uma amostra de pares (*X*, *X'*). O outro se refere a calcular a entropia entre a diferença dos dois textos cifrados. A idéia inicial para os dois testes é a mesma, seguindo os seguintes passos:

- Executar evoluções do cálculo de pré-imagem para um reticulado inicial *X* obtendo o reticulado final *Y*;
- Provocar uma perturbação em apenas um bit do reticulado inicial *X* obtendo o reticulado inicial com a diferença *X'*;
- Executar o cálculo de pré-imagem para o reticulado inicial *X'* obtendo o reticulado final *Y'*;
- Executar a operação XOR entre os reticulados finais *Y* e *Y'* obtendo *Z*.

A partir daqui, o experimento com desvio padrão contabiliza o número de bits 1s de *Z* (diferença entre *Y* e *Y'*), e armazena este valor para vários pares de *Y* e *Y'*, formando uma distribuição para que o desvio padrão seja calculado. Já no segundo experimento, é verificada a entropia de *Z* para diversos pares de *Y* e *Y'*, obtendo ao final do teste a entropia mínima, a entropia máxima e a entropia média para cada um dos tamanhos de reticulados iniciais que são analisados.

Nestes dois testes foram utilizados reticulados iniciais aleatórios com tamanho 16, 32, 64 e 128 bits, e o mesmo conjunto de regras para raio 1, 2, 3 e 4 dos experimentos anteriores. A quantidade de passos do cálculo de pré-imagem nestes testes é de 50%, 75% e 100%. De uma forma geral, os testes com 16 bits apresentaram grandes diferenças em relação aos

demais. Atribuímos essa diferença ao tamanho reduzido do reticulado em relação ao tamanho da região de contorno. Por exemplo, no raio 2, a borda corresponde a 5 bits do reticulado (31,25% do reticulado) e no raio 4, a borda corresponde a 9 bits (56,25%). Assim, o comportamento da borda acaba por distorcer o comportamento dinâmico do AC como um todo, que deveria ser determinado pela regra principal. Contudo, nesta seção, não faremos a análise dos resultados com reticulados de 16 bits e recomendamos que, embora o método possa ser aplicado com qualquer tamanho de bloco, deve-se considerar o tamanho da região de contorno em relação ao tamanho total do bloco.

6.3.1 Análise do desvio padrão na propagação da perturbação

Nesta seção, são apresentados os resultados para os modelos básico e com rotação (borda e núcleo) do cálculo de pré-imagem utilizando regras (núcleos) de raios 1, 2, 3 e 4 com 1000 amostras de reticulados iniciais escolhidas de forma aleatória, para cada um dos tamanhos de reticulado. Os resultados aqui apresentados se referem aos experimentos com evoluções de 100% do tamanho do reticulado para execução do cálculo de pré-imagem. Os demais resultados estão disponíveis no Apêndice D.

6.3.1.1 Análise do modelo básico: desvio padrão na propagação da perturbação

Nos resultados aqui discutidos, para cada tamanho de raio e de reticulado é apresentada a média da distribuição (que se refere ao percentual de 1s em relação ao tamanho do reticulado), normalizada de acordo com o tamanho do reticulado, e o desvio padrão dessa média. O conjunto de regras aqui empregado é o mesmo utilizado nos testes de ciclo e entropia do texto cifrado para primeira versão. O Apêndice G exemplifica algumas dessas regras.

Quanto mais próximo de 50% estiver o valor da média da distribuição, significa que a diferença entre os dois textos cifrados (Z) possui a mesma quantidade de 0s e 1s. Portanto, essa é a média desejada na análise de perturbação. Além disso, segundo Sen *et al*. (2002), um sistema criptográfico que possui o valor do desvio padrão dessa média abaixo de 10% pode ser considerado seguro contra um ataque de criptoanálise diferencial.

A Tabela 20 mostra os resultados obtidos para os experimentos de raio 1, 2, 3 e 4 ao calcular o desvio padrão em pares de reticulados finais, cujo reticulados iniciais sofreram alteração de apenas um bit.

Tabela 20 - Modelo básico: desvio padrão ao alterar um bit nos reticulados iniciais.

N	Raio 1		Raio 2		Raio 3		Raio 4	
	Média (%)	Desvio (%)	Média (%)	Desvio (%)	Média (%)	Desvio (%)	Média (%)	Desvio (%)
32	34,03047	20,97531	51,01378	9,414386	51,5145	8,665817	51,56875	8,694179
64	31,74056	22,95171	50,20157	7,274188	50,76155	6,196444	50,78557	6,189435
128	30,52559	23,99493	49,77151	6,062411	50,3909	4,393901	50,3766	4,412804
Média	32,09887	22,64065	50,32895	7,583662	50,88898	6,418721	50,91031	6,432139

De acordo com os resultados apresentados na Tabela 20, na maioria dos casos, é possível notar que com o aumento do tamanho do raio, tanto na média da distribuição quanto no valor do desvio padrão, resultados melhores são encontrados. A diferença entre os raios 3 e 4 é muito sutil. Os valores obtidos a partir do raio 2 são aceitáveis pois a média tende a 50% e o desvio padrão está abaixo de 10%.

6.3.1.2 Análise do modelo com rotação: desvio padrão na propagação da perturbação

Nos experimentos realizados para a última versão do método de cálculo de pré-imagem, foram realizados os mesmos testes da análise para a primeira versão. Porém, aqui os núcleos são rotacionados gerando uma regra a cada passo. Os núcleos aqui analisados são os mesmos empregados nos testes de ciclo e entropia no texto cifrado para o modelo com rotação. O Apêndice H apresenta alguns exemplos desses núcleos.

Conforme pode ser observado na Tabela 21, o modelo com rotação também apresenta melhores resultados à medida em que o tamanho do raio é aumentado, exceto para o desvio padrão que apresentou piores resultados a partir do raio 3. De fato, os experimentos com o modelo básico apresentaram melhores resultados, apesar de não terem se saído bem nos testes de ciclo e entropia do reticulado final. Como comentado anteriormente, para o conjunto de núcleos da última versão, dois núcleos foram preparados para intencionalmente produzirem um comportamento periódico. Essa inclusão acabou influenciando nos resultados deste experimento, pelo fato destes núcleos não serem adequados para criptografia. Mais adiante serão apresentados experimentos mostrando o comportamento dinâmico destes núcleos confirmando a necessidade de excluí-los do espaço de chaves possíveis.

Tabela 21 - Modelo com rotação: desvio padrão ao alterar um bit nos reticulados iniciais.

N	Raio 1		Raio 2		Raio 3		Raio 4	
	Média (%)	Desvio (%)	Média (%)	Desvio (%)	Média (%)	Desvio (%)	Média (%)	Desvio (%)
32	39,61367	21,01367	51,50444	8,774866	51,00753	9,699211	50,74519	10,40402
64	38,74951	21,36819	50,73297	6,245511	50,29411	7,297249	49,93303	8,687604
128	38,17253	21,52002	50,37423	4,415558	49,9252	5,697839	49,50321	7,526135
Média	38,84524	21,30063	50,87055	6,478645	50,40895	7,564766	50,06048	8,872585

6.3.2 Análise da entropia na propagação da perturbação

A maioria das análises de perturbação em sistemas criptográficos encontradas na literatura considera apenas a distribuição de 0s e 1s na diferença dos reticulados finais *Y* e *Y'*, sendo que o comportamento ideal é obter uma média em torno de 50% de 0s e 50% de 1s com um pequeno desvio padrão em relação a essa média. Entretanto, nos trabalhos de Gutowitz (1995) e Oliveira e colaboradores (2004), uma análise qualitativa dessa perturbação também é realizada, e mostra-se que uma perturbação tendenciosa (a uma região específica do reticulado) também é um comportamento indesejado. Por exemplo, suponha que a aplicação da operação XOR entre dois reticulados finais *Y* e *Y'* de 16 bits resulte na seqüência binária {1111111100000000}. Embora a distribuição dessa diferença apresente exatamente 50% de 0s e 1s, ela é claramente indesejada por apresentar um padrão que pode ser utilizado para algum tipo de criptoanálise. Assim, nos deparamos com o problema de quantificar não só a distribuição percentual dos 0s e 1s na diferença XOR dos reticulados finais *Y* e *Y'*, mas também quantificar a distribuição espacial de 0s e 1s. A entropia espacial apresentada na seção 5.7, torna essa quantificação possível através de uma única medida. Como ela quantifica a aleatoriedade espacial de uma seqüência binária, para que uma seqüência apresente um alto valor de entropia (acima de 0,8) é necessário não apenas uma distribuição de 0s e 1s em torno de 50%, mas também uma distribuição espacial aleatória desses bits, conforme foi observado nos exemplos da Tabela 12.

Os experimentos para a análise de entropia da diferença entre os reticulados finais ao se alterar um bit no texto claro seguiram as mesmas especificações dos experimentos realizados com o desvio padrão. São apresentados os resultados das regras (núcleos) de raios 1, 2, 3 e 4 com 1000 amostras de reticulados iniciais (32, 64 e 128 bits) escolhidas de forma aleatória, onde 100% do tamanho do reticulado equivale a quantidade de passos a serem executadas pelo cálculo de pré-imagem. Os resultados completos desses testes também estão disponíveis no Apêndice D.

6.3.2.1 Análise do modelo básico: entropia na propagação da perturbação

Os resultados para análise de entropia entre os reticulados finais ao se alterar um bit nos reticulados iniciais são apresentados para cada raio de acordo com o tamanho do reticulado. São apresentadas a entropia mínima, entropia máxima e a entropia média, dado o conjunto de regras e reticulados analisados. Quanto maior o valor da entropia, menor é a regularidade encontrada na diferença entre os dois reticulados finais. A Tabela 22 mostra os resultados obtidos com o modelo básico.

Tabela 22 - Modelo básico: entropia da diferença dos reticulados finais, ao se alterar um bit nos reticulados iniciais.

N	Raio 1			Raio 2			Raio 3			Raio 4		
	Min	**Máx**	**Méd**	**Min**	**Max**	**Méd**	**Min**	**Max**	**Méd**	**Min**	**Max**	**Med**
32	0,20	0,98	0,65	0,20	1,00	0,84	0,47	0,99	0,84	0,32	1,00	0,84
64	0,12	0,95	0,59	0,12	0,95	0,86	0,68	0,96	0,87	0,69	0,96	0,87
128	0,07	0,94	0,55	0,07	0,94	0,88	0,78	0,94	0,88	0,78	0,94	0,88
Média	0,13	0,96	0,60	0,13	0,96	0,86	0,64	0,96	0,86	0,60	0,97	0,86

Vemos que apenas nos experimentos de raio 1 a entropia média está abaixo do desejado (0,8). Tal comportamento é justificado pelo fato dessa vizinhança possuir um alcance menor que as demais a cada passo, necessitando de um número maior de passos de pré-imagem para propagar qualquer perturbação. Além disso, como os núcleos de raio 1 são formados por apenas 4 bits, é viável testar todas as regras possíveis. Nesse caso, são incluídas regras com comportamento indesejado (periódico) ou com problemas de rotação, conforme discutido na seção 6.1.4. Quando as regras são criadas aleatoriamente nos raios 2, 3 e 4, esse comportamento é mais improvável.

Nos experimentos de raio 3 e 4, o aumento do tamanho do reticulado faz com que o valor da entropia mínima aumente. O valor de entropia máxima se mostrou alto para todos os raios. Com o valor da entropia média é possível constatar que a maioria dos experimentos realizados tende a não apresentar padrões entre a diferença de dois reticulados finais ao efetuar a modificação de um bit no reticulado inicial.

6.3.2.2 Análise do modelo com rotação: entropia na propagação da perturbação

Nos experimentos com a última versão do método de cálculo de pré-imagem foram

realizados os mesmos testes apresentados para primeira versão. A diferença é que na última versão, além das bordas, os núcleos rotacionam gerando uma regra diferente a cada passo. Alguns exemplos dos núcleos analisados estão disponíveis no Apêndice H. Os resultados são apresentados na Tabela 23.

Tabela 23 - Modelo com rotação: entropia da diferença dos reticulados finais ao se alterar um bit nos reticulados iniciais.

N	Raio 1			Raio 2			Raio 3			Raio 4		
	Min	**Máx**	**Méd**	**Min**	**Max**	**Méd**	**Min**	**Max**	**Méd**	**Min**	**Max**	**Med**
32	0,20	0,99	0,70	0,24	1,00	0,84	0,20	0,99	0,84	0,20	1,00	0,83
64	0,12	0,95	0,69	0,48	0,95	0,86	0,15	0,95	0,86	0,12	0,95	0,85
128	0,07	0,94	0,69	0,55	0,95	0,88	0,13	0,94	0,88	0,07	0,94	0,87
Média	0,13	0,96	0,70	0,42	0,97	0,86	0,16	0,96	0,86	0,13	0,96	0,85

Os valores de entropia máxima e média são bem próximos daqueles apresentados no modelo básico. Entretanto, o valor de entropia mínima se mostrou bem mais baixo para os raios 3 e 4.

Apesar do valor de entropia média ter sido alto, na maioria dos casos acima de 0,8, o resultado do valor de entropia mínimo é preocupante nas duas versões do cálculo de pré-imagem, pois a diferença entre reticulados iniciais ao sofrerem modificações de apenas um bit, produzem reticulados finais muito próximos para alguns experimentos. Porém, já havia sido indicado em experimentos anteriores que, as amostras de núcleos utilizadas na versão com rotação do cálculo de pré-imagem, possuem núcleos que não são aconselhadas para utilização em um sistema criptográfico. Uma análise mais detalhada apresentando a diferença entre o comportamento dinâmico entre tipos diferentes de núcleos e como evitar aqueles inadequados é apresentada a seguir.

6.4 Método com rotação empregando filtragem na geração dos núcleos

Os resultados apresentados anteriormente para o método básico de cálculo de pré-imagem, mostraram que a utilização de apenas uma única regra principal faz com que o método possa apresentar ciclo unitário ou um ciclo pequeno (abaixo do tamanho do núcleo). Essa característica torna o método vulnerável a alguns reticulados iniciais específicos, tornando-o perigoso para utilização como um sistema criptográfico. O modelo com rotação (de borda e de núcleo) se mostrou mais promissor, porém alguns núcleos não apresentaram

bons resultados, como é o caso dos dois núcleos "preparados" com alto desbalanceamento no conjunto de amostras e aquelas regras que voltam ao núcleo inicial precocemente ao sofrer rotação (seção 6.1.4).

A análise a seguir apresenta o comportamento dinâmico de amostras de núcleos gerados aleatoriamente em comparação com os núcleos que foram preparados com um alto desbalanceamento. Posteriormente é apresentada uma solução que filtra núcleos com essa característica. Ao final, refazemos as análises da seção 6.3 para o conjunto de núcleos gerados com essa filtragem.

6.4.1 Comparação do comportamento dinâmico entre os núcleos desbalanceados e os núcleos aleatórios

Dentre os experimentos para verificar o comportamento dinâmico dos núcleos, serão apresentados alguns resultados para núcleos sensíveis à direita de raio 1, 2 e 4. Os resultados para núcleos de raio 3 são muito próximos dos obtidos com raio 4. Núcleos sensíveis à esquerda possuem análise similar às descritas nesta seção. Para cada raio analisado, foram avaliados dois núcleos, um desbalanceado propositalmente e outro aleatório. Para cada núcleo, também são empregados dois tipos de reticulados iniciais, sendo um deles com a configuração homogênea formada somente com o bit 0 e o outro aleatório. A evolução do cálculo de pré-imagem é visualizada de baixo (reticulado inicial) para cima (reticulado final) pelo diagrama espaço-temporal que possui 20 passos de evolução com um reticulado de 32 bits.

A Figura 53 apresenta as evoluções geradas utilizando-se um reticulado inicial nulo (todos as células com o bit 0) e a Figura 54 apresenta a evolução a partir de um reticulado inicial aleatório para raio 1. Em cada figura, são comparadas as evoluções utilizando-se um núcleo com um padrão mais aleatório (1001) e um núcleo mais desbalanceado (1000). Nas duas figuras, os núcleos aleatório e desbalanceado apresentaram um comportamento similar que pode ser classificado como caótico.

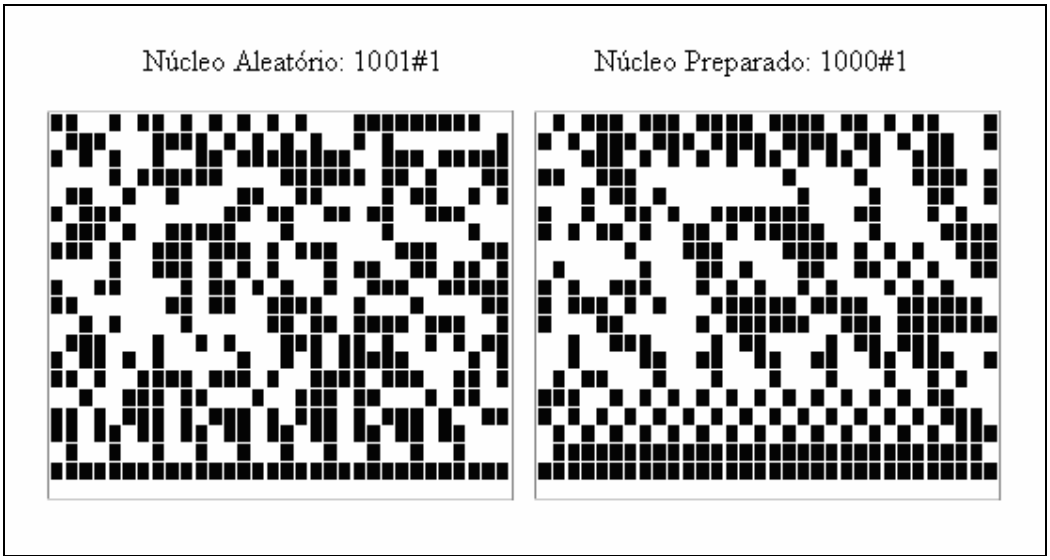


Figura 53 - Diagrama espaço-temporal para regras de raio 1 evoluindo com reticulado inicial nulo.

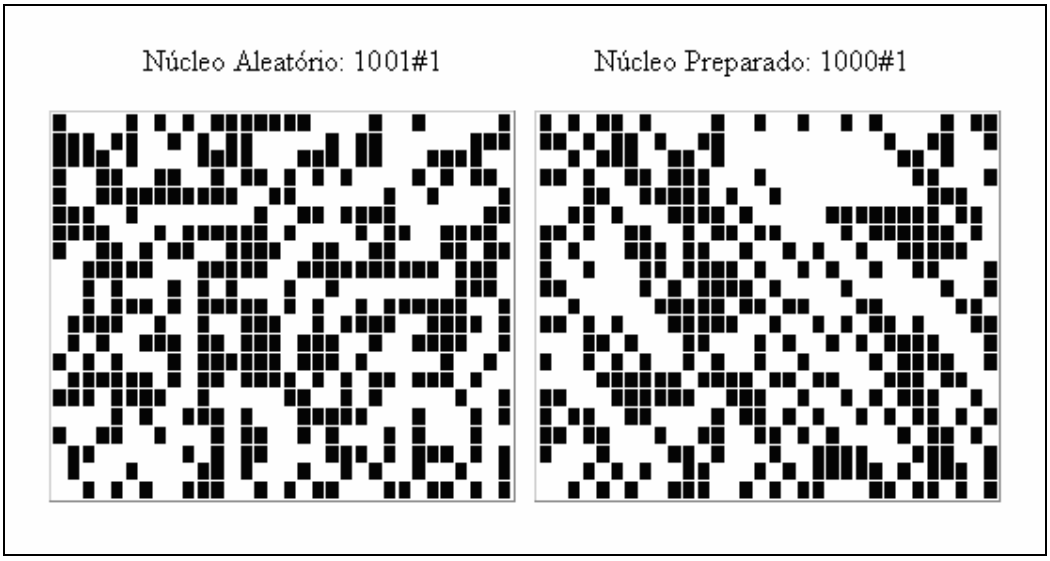


Figura 54 - Diagrama espaço-temporal para regras de raio 1 evoluindo com reticulado inicial aleatório.

A Figura 55 apresenta as evoluções geradas utilizando-se um reticulado inicial nulo e a Figura 56 apresenta a evolução a partir de um reticulado inicial aleatório, utilizando-se um núcleo com um padrão aleatório (1011110011010110) e um núcleo desbalanceado (0000000000000001) para raio 2. A partir desse valor de raio, o comportamento da regra gerada pelo núcleo desbalanceado já começa a se diferenciar.

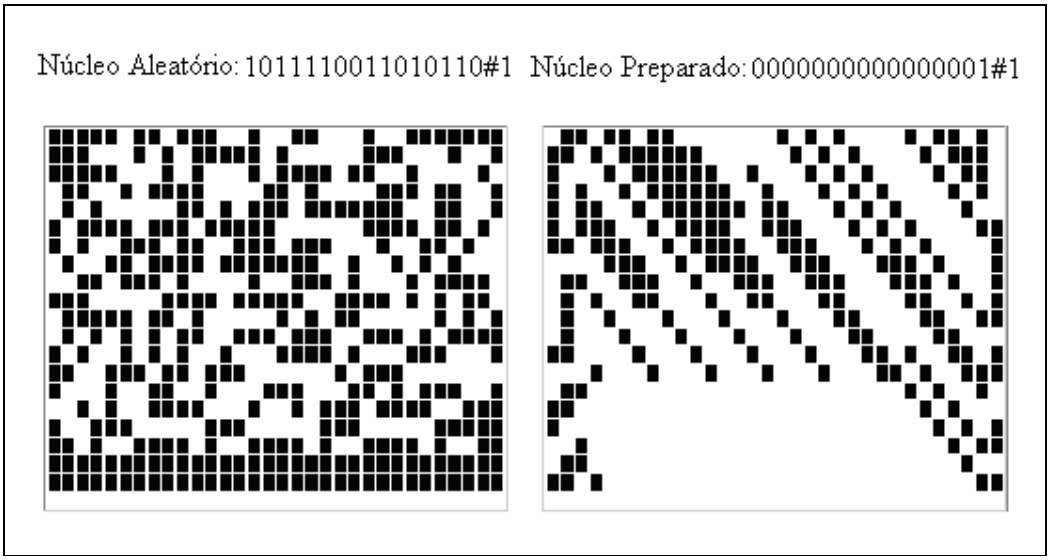


Figura 55 - Diagrama espaço-temporal para regras de raio 2 evoluindo com reticulado inicial nulo.

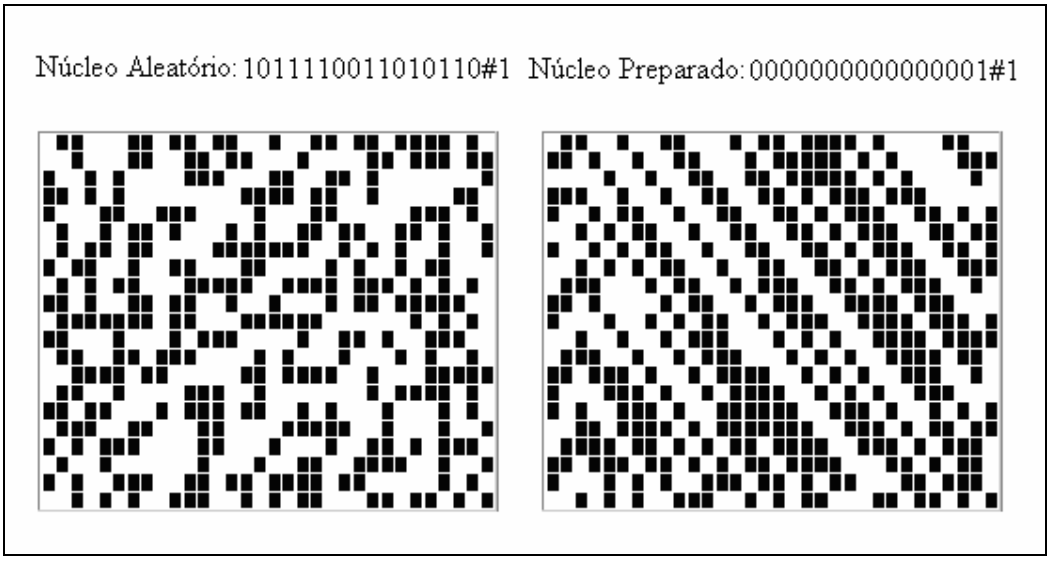


Figura 56 - Diagrama espaço-temporal para regras de raio 2 evoluindo com reticulado inicial aleatório.

Por fim, a Figura 57 apresenta as evoluções geradas utilizando-se um reticulado inicial nulo e a Figura 58 apresenta a evolução a partir de um reticulado inicial aleatório, com um núcleo aleatório e um núcleo preparado para raio 4 (256 bits).

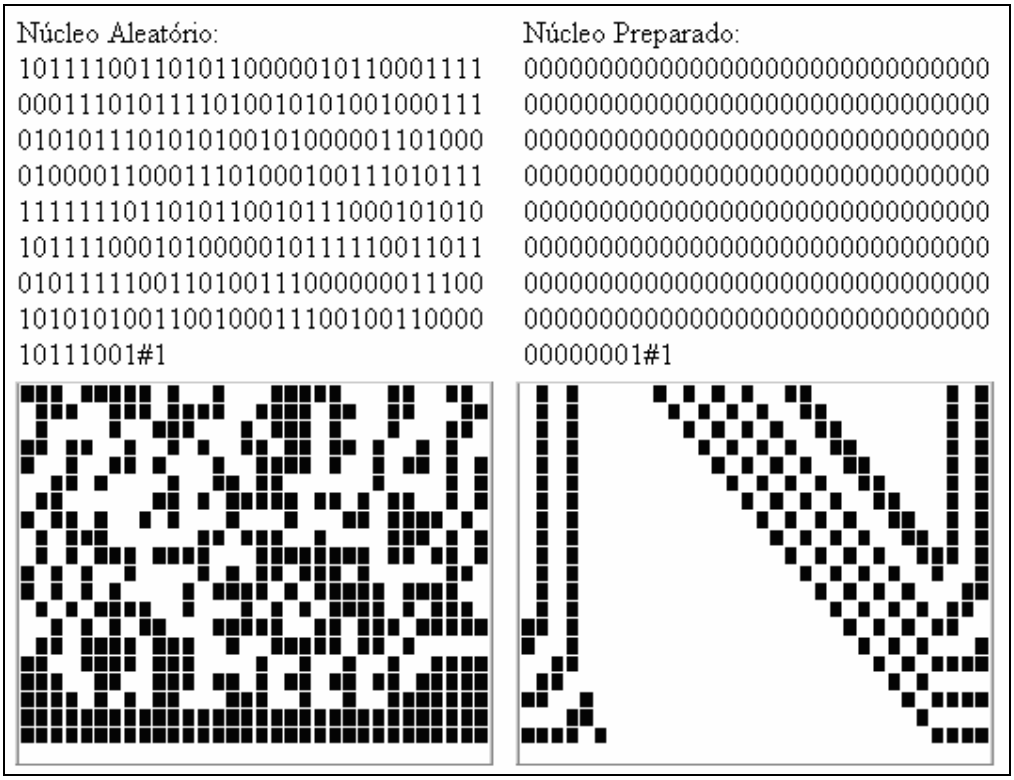


Figura 57 - Diagrama espaço-temporal para regras de raio 4 evoluindo com reticulado inicial nulo.

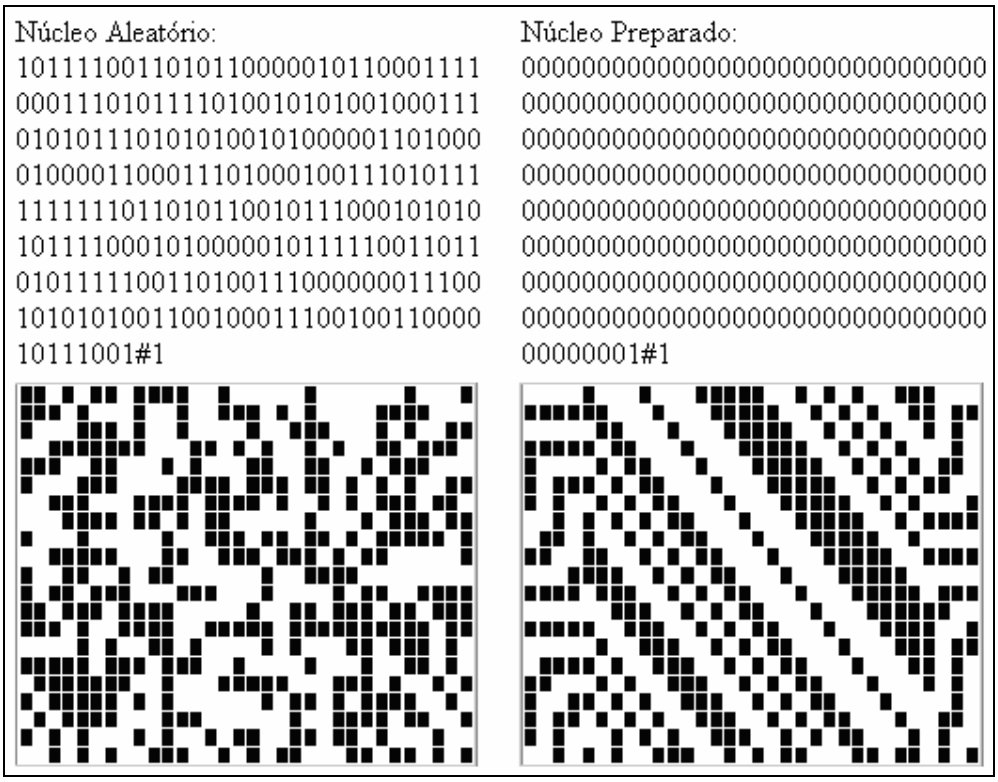


Figura 58 - Diagrama espaço-temporal para regras de raio 4 evoluindo com reticulado inicial aleatório.

Os resultados para raio 1 apresentam um comportamento próximo para os dois núcleos. Porém, à medida que o tamanho do raio aumenta, fica muito claro que o diagrama espaço-temporal apresenta um comportamento muito mais ordenado para os núcleos desbalanceados. Esse comportamento se justifica pela dinâmica periódica apresentada por regras com alto desbalanceamento de 0s e 1s, mesmo em ACs homogêneos. Na composição da regra principal com núcleo desbalanceado e a utilização de uma regra de contorno, esse comportamento não é perfeitamente periódico, mas apresenta uma maior regularidade. Portanto, essa análise confirma que núcleos com alto desbalanceamento realmente precisam ser evitados na geração de uma chave criptográfica.

Além disso, como apresentado anteriormente na análise de ciclo (seção 6.1.2), dentre os núcleos analisados, todos que apresentaram um ciclo unitário ou abaixo do tamanho do núcleo possuem entropia do núcleo abaixo de 0,75. Tal comportamento pode ser evitado, portanto, filtrando-se (excluindo) núcleos com entropia abaixo de 0,75. A vantagem dessa filtragem é que ao excluir núcleos com baixa entropia, também se evita núcleos com alto desbalanceamento, que apresentam o comportamento dinâmico ordenado indesejado. Dessa forma, estabelecemos que, além de empregar o modelo com rotação do núcleo, o ideal é que se faça uma filtragem nos núcleos utilizados impedindo-se a utilização de chaves com entropia abaixo de 0,75. Embora chaves com esses núcleos sejam improváveis de ocorrer, a filtragem diminui as chances desses comportamentos indesejáveis ocorrerem. Foram realizados novos experimentos com uma amostra de 100 núcleos para cada raio, sendo que todos possuem entropia maior que 0,75, exceto para raio 1 que será analisado com núcleos maiores ou iguais a 0,75 de entropia, devido a baixa quantidade de núcleos com entropia alta para este raio.

6.4.2 Modelo com rotação e filtragem do núcleo: desvio padrão na propagação da perturbação

Os experimentos aqui realizados seguiram a mesma configuração daqueles apresentados na seção 6.3 para a análise de desvio padrão. Nesses experimentos, foi aplicado um novo conjunto de núcleos, disponíveis no Apêndice I, que passaram por uma filtragem nos núcleos abaixo de 0,75. Os resultados são apresentados na Tabela 24.

Tabela 24 - Desvio padrão da perturbação propagada, empregando-se o modelo com rotação e núcleos filtrados ($s > 0,75$).

N	Raio 1		Raio 2		Raio 3		Raio 4	
	Média (%)	Desvio (%)	Média (%)	Desvio (%)	Média (%)	Desvio (%)	Média (%)	Desvio (%)
32	50,75781	9,58617	51,62125	8,70505	51,59153	8,666019	51,60175	8,728055
64	50,63008	6,647073	50,79058	6,197615	50,80152	6,211798	50,81086	6,214282
128	50,37797	4,642778	50,3943	4,403376	50,38629	4,419201	50,38947	4,387211
Média	50,58862	6,958674	50,93538	6,435347	50,92645	6,432339	50,93403	6,443183

Comparando o resultado do desvio padrão deste experimento com os experimentos efetuados com o mesmo modelo sem a filtragem dos núcleos (Tabela 21), em que duas regras com alto desbalanceamento haviam sido preparadas e incluídas na amostra de regras, é possível notar que os valores do desvio padrão foram mais baixos, especialmente nos resultados para raio 1. Note também que para todos os raios analisados, a média da distribuição ficou bem próxima de 50%, e em todos os casos, houve uma diminuição no valor do desvio padrão, tal que na média geral são obtidos valores abaixo de 7%. É possível observar que quanto maior o tamanho do reticulado, melhores são os resultados: média mais próxima de 50% e desvio padrão mais baixo.

6.4.3 Modelo com rotação e filtragem do núcleo: entropia na propagação da perturbação

A mesma configuração dos experimentos da seção 6.3.2.2 foi empregada na análise da entropia da diferença entre os reticulados finais ao se alterar um único bit no reticulado inicial, utilizando-se o novo conjunto de núcleos com entropia acima de 0,75 (Apêndice I). Os resultados são apresentados na Tabela 25.

Tabela 25 - Entropia da perturbação propagada, empregando-se o modelo com rotação e núcleos filtrados.

N	Raio 1			Raio 2			Raio 3			Raio 4		
	Min	Máx	Méd	Min	Max	Méd	Min	Max	Méd	Min	Max	Med
32	0,30	0,99	0,84	0,50	1,00	0,84	0,46	1,00	0,84	0,53	1,00	0,84
64	0,31	0,95	0,86	0,66	0,96	0,87	0,66	0,96	0,87	0,65	0,97	0,87
128	0,35	0,94	0,88	0,78	0,94	0,88	0,77	0,94	0,88	0,77	0,94	0,88
Média	0,32	0,96	0,86	0,65	0,97	0,86	0,63	0,97	0,86	0,65	0,97	0,86

Comparando-se esses resultados com aqueles da Tabela 23, é possível notar que os valores de entropia média nos dois experimentos são significativamente melhores para o raio

1 e discretamente melhores para os outros raios. Os valores de entropia mínima realmente sofreram influência direta dos núcleos desbalanceados que foram incluídos nos experimentos da Tabela 23. Assim, com a filtragem desses núcleos, os resultados de entropia mínima são bem melhores, especialmente para os raios 2, 3 e 4. Também é possível notar que os resultados da entropia tendem a ser melhores, assim como observado no desvio padrão, quanto maior o tamanho do reticulado avaliado.

Dessa forma, é possível concluir que tanto na análise da média e desvio padrão da distribuição de 0s e 1s, quanto na análise de entropia, os resultados de propagação de uma pequena perturbação no reticulado inicial apresentaram melhorias significativas após serem descartados os núcleos com entropia abaixo de 0,75.

6.5 Avaliação da perturbação na chave

Um bom sistema criptográfico deve apresentar blocos de texto cifrados diferentes ao ser executado com chaves criptográficas diferentes. Caso contrário, um ataque em cima da chave pode ser empregado para nortear qual a chave correta, baseado dos textos cifrados que o sistema produz com aquela chave. Ou seja, se o mesmo bloco de texto claro é cifrado por duas chaves semelhantes (diferença de poucos bits) e produz blocos de texto cifrado também semelhantes, o sistema criptográfico pode ser explorado por um criptoanalista, que utilizará esta informação para testar algumas chaves. Neste caso, o sistema criptográfico irá dizer para o criptoanalista qual chave está mais próxima ou mais distante de ser a chave correta. Dessa forma, é desejável que ao utilizar duas chaves similares, blocos de textos cifrados completamente diferentes sejam produzidos, não revelando nenhuma característica da chave. Usaremos o cálculo do desvio padrão e da entropia para realizar experimentos sobre o modelo com rotação. Assim como nos experimentos anteriores, aqui também são apresentados apenas os resultados dos experimentos com evolução de 100% do tamanho do reticulado. Os demais resultados estão disponíveis no Apêndice E.

Nas seções anteriores, foram apresentados os resultados dos testes com o desvio padrão entre os reticulados finais ao sofrerem alteração de um bit no reticulado inicial. Os resultados apresentados a seguir referem-se a experimentos similares, porém, a alteração é realizada em um único bit no núcleo, verificando o desvio padrão entre dois reticulados iniciais idênticos, mas evoluídos com núcleos próximos. Os resultados são apresentados na Tabela 26.

Tabela 26 - Desvio padrão ao se alterar um bit na chave, empregando-se o modelo com rotação e os núcleos com entropia alta.

N	Raio 1		Raio 2		Raio 3		Raio 4	
	Média (%)	Desvio (%)	Média (%)	Desvio (%)	Média (%)	Desvio (%)	Média (%)	Desvio (%)
32	51,54219	8,65899	51,54753	8,664525	51,55031	8,64934	47,10613	14,02806
64	50,775	6,191007	50,77202	6,197218	50,80788	6,230347	50,77106	6,202734
128	50,4251	4,394387	50,37401	4,408169	50,4045	4,401129	50,38982	4,409335
Média	50,91409	6,414795	50,89785	6,423304	50,9209	6,426939	49,42234	8,213377

É possível notar que o método é mais sensível a uma alteração em um bit do núcleo do que em um bit no reticulado inicial. Além disso, quanto maior o tamanho do reticulado, melhores são os resultados para cada raio analisado, tanto para a média da distribuição quanto para o desvio padrão. O resultado para reticulado de 32 bits e raio 4 ficou abaixo do desejado.

Os experimentos para análise de entropia entre reticulados finais ao se alterar um bit no núcleo são similares aos apresentados anteriormente. Os resultados são apresentados na Tabela 27.

Tabela 27 - Entropia ao se alterar um bit na chave, empregando-se o modelo com rotação e os núcleos com entropia alta.

N	Raio 1			Raio 2			Raio 3			Raio 4		
	Min	Máx	Méd	Min	Max	Méd	Min	Max	Méd	Min	Max	Med
32	0,57	0,99	0,84	0,47	1,00	0,84	0,45	1,00	0,84	0,20	1,00	0,80
64	0,66	0,95	0,86	0,65	0,96	0,87	0,67	0,96	0,86	0,49	0,96	0,87
128	0,77	0,94	0,88	0,78	0,94	0,88	0,76	0,94	0,88	0,78	0,94	0,88
Média	0,67	0,96	0,86	0,63	0,97	0,86	0,63	0,97	0,86	0,49	0,97	0,85

Os resultados apresentados mostram que em todos os raios a entropia média ficou acima de 0,8, portanto, indicando que os blocos produzidos não apresentam nenhum padrão. Para os valores de entropia mínima, é possível concluir que os reticulados de 32 bits apresentaram resultados mais baixos, que tendem a melhorar com o aumento do reticulado.

6.6 Análise das propriedades algébricas

As três versões do sistema criptográfico CAC propostas na literatura [Nandi *et al.* 1994], [Ganguly *et al.* 2000] e [Sen *et al.* 2002], discutidas na seção 4.2, foram submetidas a estudos de criptoanálise específicos que identificaram fraquezas do sistema [Blackburn *et al.* 1997] e [Bao 2004].

Uma das preocupações de nosso estudo foi investigar se uma criptoanálise do mesmo tipo poderia ser empregada sobre o sistema proposto nessa dissertação.

Nos métodos empregados nas três versões do CAC, os ACs empregados devem ser aditivos e apresentar alguma das seguintes propriedades:

- Serem de ciclo pequeno e conhecido, como os ACs empregados no *Major CA*, que têm ciclo de tamanho 32 (seção 4.2). Assim, na cifragem deve-se evoluir o AC para frente por Δ passos, e na decifragem, basta rodar o AC para frente por 32-Δ passos. Por exemplo, em [Chaudhuri *et al*. 1997] foi demonstrando que qualquer arranjo das regras elementares 51, 153 e 195 com condição de contorno nula, gera ACs não-homogêneos de duração não-máxima com *n* ciclos de tamanho *p*, onde *p* é par e é uma potência de 2 (2^p).

- Serem de ciclo máximo, como os ACs empregados no *Minor CA* que têm ciclo igual a $2^{128}-1$ (seção 4.2). Nesse caso, tanto na cifragem como na decifragem eles são utilizados para gerar alguma seqüência aleatória, utilizando a chave como semente (reticulado inicial). Para isso, o AC também é executado para frente e por um número fixo de passos. Para exibir essa propriedade, os ACs devem ser aditivos e possuir o polinômio da matriz característica primitivo ou irredutível (coeficientes não fatoráveis).

Regras com esse comportamento são possíveis de serem especificadas justamente pelo fato dos ACs serem aditivos e poderem ser expressos através de uma matriz característica (seção 3.3). Além disso, a matriz característica deve exibir a propriedade algébrica desejada. Nossa conclusão é que, da mesma forma que os métodos de cifragem e decifragem do sistema CAC são baseados nas propriedades algébricas dos ACs aditivos, a criptoanálise sobre esses modelos também se baseou nessas mesmas propriedades.

Dessa forma, uma criptoanálise similar às realizadas em [Blackburn *et al*. 1997] e [Bao 2004] não nos parece possível de ser aplicada no novo método, pois:

- Os ACs não-homogêneos empregados no método proposto não são necessariamente aditivos, pois não nos valemos de propriedades algébricas para realizar as etapas de cifragem/decifragem.

- Os ACs não-homogêneos empregados no método proposto podem, eventualmente, serem classificados como aditivos. As regras de contorno são sempre aditivas. Assim,

se a regra principal também for aditiva, o AC não-homogêneo resultante será aditivo. Por exemplo, no caso de ACs de raio 1, a regra 60 {00111100} que pode ser empregada como regra principal é uma regra aditiva e é sensível à esquerda (Tabela 10). A regra de contorno correspondente a essa regra principal é a regra 15 {11110000}. Assim, o AC que utiliza essas duas regras é um AC aditivo. Entretanto, no nosso método não precisamos saber o tamanho do ciclo *a priori*, ou mesmo se esse tamanho é sempre o mesmo independentemente do reticulado inicial. A diferença em nosso método é que aplicamos uma evolução inversa à evolução para frente na etapa de cifragem e não precisamos utilizar o tamanho do ciclo para retornar ao texto original.

- Todas as regras aditivas com sensitividade são formadas por núcleos contendo somente 0s, somente 1s, ou com número igual de 0s e 1s.

Caso fosse desejado eliminar os ACs aditivos de nosso método, bastaria filtrar também os núcleos com numero de 0s igual ao número de 1s (os casos com somente 0s ou somente 1s já são filtrados, pois têm entropia espacial igual a zero). Entretanto, como esses núcleos correspondem a uma parcela significativa do espaço de chaves e não encontramos nenhuma evidência de um ataque baseado nas propriedades algébricas que eventualmente alguns tipos de núcleos poderiam exibir, resolvemos ignorar essa questão e permitir que tanto ACs não-aditivos quanto ACs aditivos pudessem ser potencialmente empregados no método.

Capítulo 7

Especificação padrão do sistema criptográfico HCA

No capítulo anterior, foram relatados os experimentos realizados para investigar se as duas versões do cálculo de pré-imagens apresentados no Capítulo 5 poderiam ser empregados em um sistema criptográfico. Nesta seção, será feita a especificação completa do sistema criptográfico, fixando algumas características, tais como, tamanho de bloco, tamanho da chave, passos de evolução, dentre outras. Com essa especificação, sugerimos valores dos parâmetros que seriam recomendáveis no estágio atual dos sistemas criptográficos usuais e do *hardware* disponível. Além disso, definir as especificações faz com que seja possível que outros pesquisadores possam realizar uma criptoanálise e identificar alguma possível fraqueza não prevista no método proposto.

O sistema criptográfico proposto segundo essa especificação padrão será chamado de HCA (*Hybrid Cellular Automata – based cipher system*).

7.1 Escolha da versão do cálculo de pré-imagem

Através dos experimentos discutidos no Capítulo 6, foi possível concluir que o modelo básico possui uma vulnerabilidade com relação a alguns reticulados iniciais específicos. Nesse modelo, apenas uma regra principal é utilizada em todos os cálculos de pré-imagem, o que torna o método não indicado para criptografia. Da mesma forma, o modelo com rotação na borda também possui esta fraqueza e, portanto, também não é aconselhado para criptografia. O método com rotação na borda e no núcleo, herda a característica de poder operar em paralelo, obtida na segunda versão, e utiliza uma regra principal a cada evolução do cálculo de pré-imagem, apresentando bons resultados para qualquer que seja o reticulado inicial. Assim, o método de cálculo de pré-imagem empregado no sistema HCA é o método com rotação na borda e no núcleo.

7.2 Tamanho do bloco

Se um reticulado evolui a partir de uma regra com núcleo relativamente grande para o seu tamanho, os bits referentes à borda podem ser muito significativos. Por exemplo, seja um núcleo de raio 4, que define que 8 bits do reticulado sejam evoluídos pela regra de contorno.

Se for utilizado um reticulado de 16 bits, diminui a influência da regra principal caótica no comportamento dinâmico resultante do AC. Sendo assim, apesar de não haver restrição, é aconselhado que o tamanho do bloco seja adequado ao tamanho da chave (núcleo).

Nesta dissertação, definimos como tamanho do bloco ideal um reticulado de 128 bits. Além dos bons resultados obtidos para o cálculo de pré-imagem com reticulados de tamanho 128 bits, este é o tamanho de bloco utilizado por um dos sistemas criptográficos mais utilizados atualmente, o AES.

7.3 Tamanho da chave

O tamanho da chave, que equivale ao tamanho do núcleo utilizado para gerar a regra principal, pode variar de acordo com o raio estipulado, como mostra a Tabela 28.

Tabela 28- Relação entre o raio e a quantidade de bits da chave criptográfica.

Raio do AC	Quantidade de bits da chave
1	4
2	16
3	64
4	256
5	1024

Na configuração padrão para o sistema HCA proposta nessa dissertação, será estabelecido que a chave é formada por ACs de raio 4. Ou seja, serão utilizadas chaves de tamanho 256 bits. Além disso, de acordo com os resultados apresentados na seção 6.4, é altamente recomendado que a chave escolhida obtenha valor de entropia maior que 0,75.

Uma chave criptográfica formada por 256 bits corresponde ao espaço formado por 2^{256} chaves diferentes. Como estamos limitando o espaço de chaves, excluindo aquelas que possuem núcleo com entropia abaixo de 0,75, de fato reduzimos o tamanho do espaço de chaves. Ao fazer uma análise para o raio 2, foi constatado que 119.744 núcleos de um total de 131.072, possuem entropia acima de 0,75 (Tabela 18), correspondendo a 91,35% do total de núcleos. Embora ainda não tenhamos encontrado uma forma de calcular a quantidade exata de núcleos possíveis pra raio 4, acreditamos que o número de regras excluídas é inferior a 5%. Assim, o espaço de chaves resultante ainda é grande o bastante para inviabilizar um ataque de força bruta. Apenas para apresentarmos uma evidência a mais dessa afirmação, realizamos um simples teste gerando aleatoriamente 100.000 palavras binárias de 256 bits, sem repetição. A cada palavra gerada, a entropia espacial s foi calculada. Ao final, nenhuma das seqüências

binárias apresentou entropia inferior a 0,75.

7.4 Número de passos de pré-imagem

A quantidade de passos de cálculo de pré-imagem também é flexível, porém, deve-se empregar uma quantidade suficiente para que o AC possa evoluir o seu comportamento dinâmico. Assim, o número de pré-imagens a serem encontradas se baseou nos experimentos realizados com a análise de entropia e desvio padrão entre dois textos cifrados ao efetuar a modificação entre apenas um bit no texto claro.

Uma quantidade de passos aconselhada corresponde a executar pelo menos o tamanho do bloco que está sendo empregado. Como definimos que o bloco tem 128 bits, temos que 128 passos de execução é o suficiente para que uma pequena perturbação se propague por todo o reticulado.

Além disso, verificamos na análise do tamanho do ciclo (seção 6.1), que ao utilizar o método com rotação na borda e no núcleo é pouco provável a existência de um reticulado inicial com ciclo inferior ao tamanho do núcleo. Ao filtrar os núcleos com entropia abaixo de 0,75, essa situação é ainda mais improvável. Assim, como o tamanho da chave é de 256 bits, é altamente improvável a existência de um ciclo menor que 256, para qualquer reticulado inicial.

7.5 Descrição geral do sistema criptográfico HCA

O sistema criptográfico pode ser executado em qualquer um dos modos de operação para sistemas criptográficos baseados em blocos (ECB, CBC, CFB, OFB). Conforme os padrões definidos nas seções anteriores, o sistema criptográfico HCA executa o processo de cifragem de acordo com os seguintes passos:

- Decompor o texto claro em blocos de 128 bits;
- Executar o cálculo de pré-imagem com rotação na borda e no núcleo, por 128 passos com a chave criptográfica de 256 bits;
- Compor os blocos cifrados, de acordo com o modo de operação desejado.

O processo de decifragem é efetuado de acordo com os seguintes passos:

- Decompor o texto cifrado em blocos de 128 bits;

- Executar a evolução temporal do AC com rotação na borda e no núcleo, por 128 passos com a chave criptográfica de 256 bits;
- Compor os blocos obtendo o texto claro, de acordo com o modo de operação efetuado no processo de cifragem.

A geração das regras que devem ser aplicadas a cada passo é realizada a partir da chave (núcleo). Esse processo deve ser feito, tanto da cifragem quanto na decifragem, da seguinte forma:

- Gerar a regra principal a partir da chave (núcleo) de acordo com a sensitividade;
- Gerar a regra de contorno a partir da regra principal;
- Rotacionar a chave em uma posição para que o processo de geração das regras se repita no passo seguinte;
- Repetir este processo até gerar *P* regras, onde *P* é o número de passos de cálculo de pré-imagem.

A Figura 59 mostra um diagrama esquemático do processo de cifragem e decifragem. Note na figura que as regras aplicadas no processo de decifragem estão sendo utilizadas na ordem inversa à empregada no processo de cifragem.

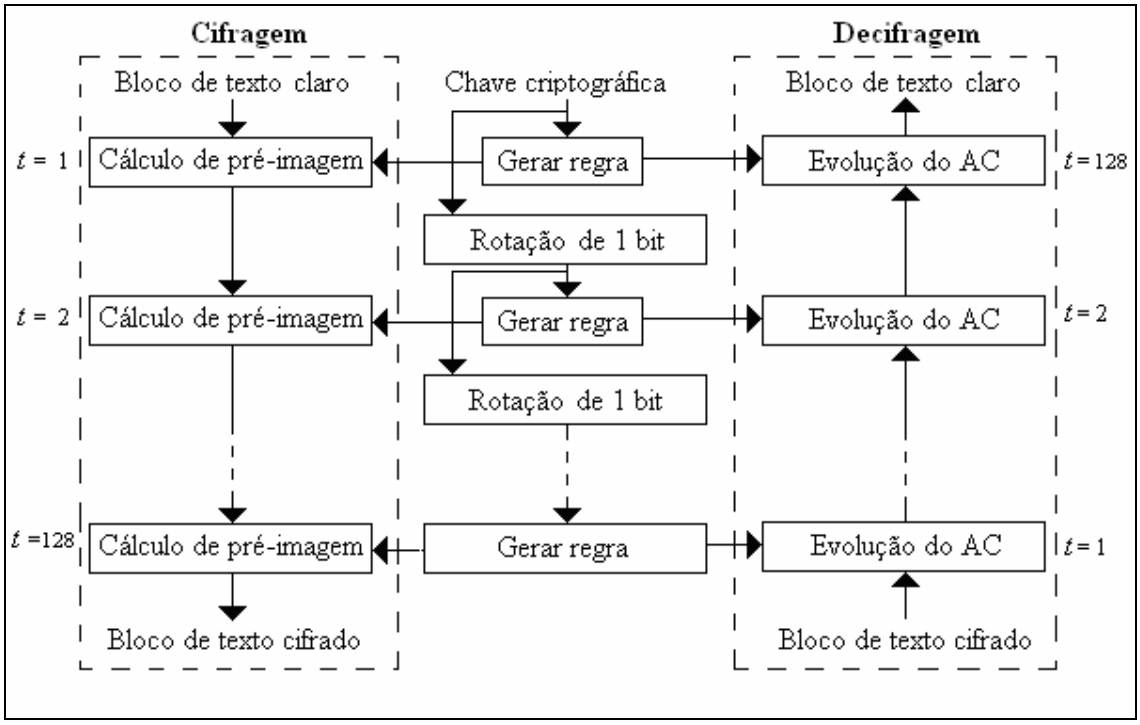


Figura 59 - Diagrama mostrando o esquema de cifragem e decifragem para o sistema criptográfico HCA.

7.6 Experimentos com a configuração padrão

7.6.1 Análise de ciclo no sistema HCA

Os resultados para análise de ciclo efetuado com a especificação descrita nas seções anteriores mostram que, em todos os núcleos analisados, nenhum reticulado apresentou ciclo menor do que a quantidade de passos executada pelo sistema criptográfico. Uma vez que foram utilizadas amostras de núcleos e reticulados, não podemos afirmar que um ciclo menor do que a quantidade de passos nunca irá ocorrer. Porém, como são empregados núcleos de 256 bits, com entropia acima de 0,75, e executados por 128 passos do cálculo de pré-imagem, é muito improvável que para uma dado núcleo e uma dado reticulado, ocorra um ciclo menor ou igual a 256.

7.6.2 Análise da entropia do texto cifrado no sistema HCA

Na análise da entropia no texto cifrado, realizada conforme os experimentos descritos na seção 6.2, foi possível constatar que para todas as amostras de reticulados iniciais, sejam elas com alta ou baixa entropia, os reticulados finais apresentam alta entropia. Os resultados apresentados na Figura 60 referem-se à entropia média (mínima e máxima) calculada sobre uma amostra de 100 reticulados iniciais (para cada grupo), utilizando-se uma amostra de 100 núcleos de 256 bits (eixo *x*).

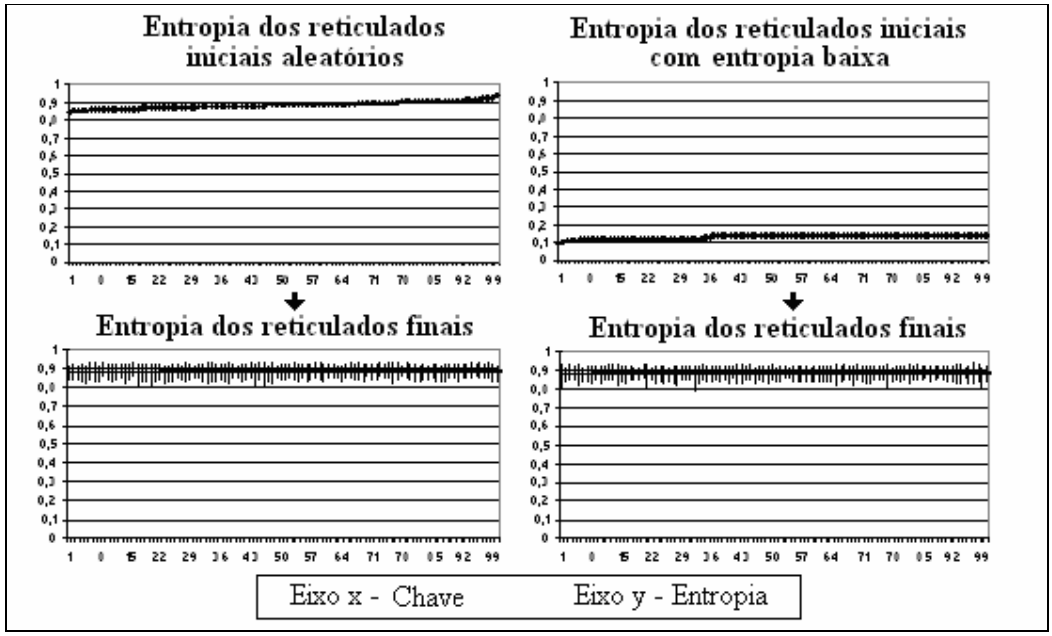


Figura 60 - Resultados da entropia em reticulados finais para a configuração padrão. Lado esquerdo: reticulados iniciais aleatórios. Lado direito: reticulados iniciais com entropia baixa.

A partir desta análise, fica claro que independentemente do texto claro que for submetido ao sistema criptográfico, será produzido um texto cifrado com ausência de um padrão específico, o que torna o sistema HCA confiável. É claro que estamos apresentando resultados de amostras, porém espera-se este comportamento para todos os reticulados iniciais possíveis.

É possível notar na Figura 60 que praticamente não há diferenças nos resultados dos reticulados finais que foram cifrados a partir de reticulados iniciais com baixa entropia ou aleatórios (alta entropia), sendo que todas as entropias finais possuem valores altos (acima de 0,8), conforme desejado.

7.6.3 Análise da propagação da perturbação para a configuração padrão

A Tabela 29 resume os resultados apresentados na seção 6.4, considerando-se a propagação de uma perturbação tanto no texto claro, quanto na chave criptográfica. São apresentados somente os valores referentes aos reticulados de 128 bits e regras de raio 4, uma vez que estes foram estabelecidos como configuração padrão do sistema HCA.

Tabela 29 - Análise da propagação da perturbação para a configuração padrão.

Alteração	*"s"* Mín.	*"s"* Máx.	*"s"* Méd.	Méd. da Distribuição	D. Padrão
Ret. Inicial	0,77	0,94	0,883224	50,389469	4,387211
Chave	0,78	0,94	0,88309	50,38982	4,409335

Os resultados indicam que, a diferença entre os blocos de texto cifrado ao modificar apenas um bit no bloco de texto claro ou chave criptográfica, possui um alto valor de entropia. Ou seja, os reticulados finais não apresentam padrões na diferença provocada pela alteração de apenas um bit no texto claro ou chave criptográfica. O resultado para a média da distribuição de 0s e 1s revela que ao efetuar a diferença entre os textos cifrados, o número de 0s e 1s obtidos é aproximadamente o mesmo, ou seja, próximo de 50%. O valor do desvio padrão mostra que o sistema criptográfico HCA não é vulnerável a um ataque de criptoanálise diferencial, pois está abaixo de 10%, tanto para uma perturbação no texto claro, quanto na chave [Sen *et al*. 2002].

Sen e colaboradores (2002) realizaram experimentos similares avaliando o desvio padrão para os métodos DES, CAC e AES. Em média, os sistemas criptográficos DES, AES e CAC retornaram os seguintes valores 28,8%, 3,6% e 3,9%, respectivamente. De acordo com estes resultados podemos verificar que o sistema HCA possui um valor de desvio padrão para

os textos cifrados bem próximo daqueles encontrados para os sistemas CAC e AES e bem abaixo do valor obtido para o DES. Assim, podemos concluir que a segurança à criptoanálise diferencial do HCA é compatível com o AES e bem superior ao DES.

Capítulo 8

Conclusões

Foi investigada a aplicação de autômatos celulares (ACs) em criptografia. Dentre os modelos analisados que empregam a evolução dos ACs, onde o texto claro corresponde ao reticulado inicial e a chave corresponde à regra (ou ao conjunto de regras) empregada (s), existem basicamente duas abordagens. Uma delas faz uso das propriedades algébricas de algumas regras aditivas [Nandi *et al*. 1994], [Ganguly *et al*. 2000] e [Sen *et al*. 2002]. A outra, refere-se ao uso do cálculo de pré-imagens [Gutowitz 1995] e [Oliveira *et al*. 2004]. Em todos os modelos com propriedades algébricas foi encontrada alguma fraqueza que poderia comprometer o método [Blackburn *et al*. 1997] e [Bao 2004]. Já os métodos que empregam o cálculo de pré-imagem são criticados por possuírem um aumento significativo na quantidade de bits do texto cifrado em relação ao texto claro.

Nesta dissertação, são propostas três formas de se executar o cálculo de pré-imagem, sem a necessidade de se acrescentar bits a cada passo de evolução, tornando o texto cifrado do mesmo tamanho que o texto claro. As três versões se baseiam na mesma idéia: a garantia de existência de uma pré-imagem para qualquer reticulado inicial, sem a necessidade de bits extras, é conseguida através do uso de um AC não-homogêneo ou híbrido. O AC não-homogêneo emprega duas regras, uma chamada de regra principal e a outra chamada de regra de contorno.

A regra principal deve possuir a característica de sensitividade a uma das células da extremidade. Essa característica provê ao AC não-homogêneo um comportamento caótico que faz com que a cifragem realizada pelo AC seja de boa qualidade, o que pode ser comprovado pela análise de entropia do texto cifrado e pelas análises de perturbação de um bit, tanto no texto claro, quanto na chave criptográfica.

A regra de contorno permite que a condição de contorno periódica do reticulado sempre seja satisfeita para encontrar uma pré-imagem, não havendo a necessidade de bits extras. Dessa forma, o texto cifrado é do mesmo tamanho que o texto original, corrigindo a principal falha dos modelos apresentados em [Gutowitz 1995] e [Oliveira *et al*. 2004].

O primeiro método de cálculo de pré-imagem, mais simples dentre os três, é executado de forma seqüencial. O segundo método possui modificações, em relação ao anterior, para

permitir uma execução paralela do cálculo. A principal modificação para a obtenção desse paralelismo está no uso de uma região de contorno variável, que sofre uma rotação a cada pré-imagem calculada. A terceira e última versão, possui a característica de evoluir com regras diferentes a cada passo de execução do cálculo, além de usufruir do mesmo paralelismo da segunda versão. Através dos testes realizados, foi constatado que os métodos referentes à primeira e à segunda versão são muito sensíveis a alguns tipos de reticulado inicial, pelo fato de apenas uma regra ser empregada a cada passo de evolução do cálculo de pré-imagem, tornando-os vulneráveis a um ataque onde o texto claro é escolhido. Porém, esta fraqueza foi corrigida na última versão, onde são geradas regras distintas para cada evolução do cálculo, através de um núcleo gerador de regras que corresponde à chave criptográfica.

A partir dos testes realizados, foi possível constatar que dois tipos de núcleos apresentam um comportamento indesejado na etapa de cifragem:

- Núcleos que, por sua configuração peculiar de bits, retornam ao núcleo original de p bits aplicando-se um número r de rotações, sendo $r < p$. O problema associado a esse tipo de núcleo é que, ao ser empregado como núcleo gerador de regras no modelo de cálculo de pré-imagem com rotação, ele pode fazer com que o AC resultante apresente um tamanho de ciclo menor que o número de passos de cifragem.
- Núcleos com alto desbalanceamento entre o número de 0s e 1s. O problema associado a esse tipo de núcleo é que, ao ser empregado como núcleo gerador de regras, ele pode fazer com que o AC resultante apresente um comportamento ordenado (quase periódico) que prejudique a qualidade da cifragem efetuada pelo AC, que se baseia na pressuposta caoticidade das regras com sensitividade.

Embora os testes tenham identificado esses tipos de núcleos "problemáticos", foi apresentada uma filtragem de núcleos, a partir de uma medida de entropia espacial ($s < 0{,}75$), que diminui consideravelmente a possibilidade de ocorrência desses núcleos, sem causar uma redução drástica do espaço de chaves.

Cabe ressaltar que, as três versões de cálculo de pré-imagem discutidas no Capítulo 6 não foram as únicas avaliadas nesse trabalho. Partindo-se do modelo básico que emprega as duas regras (principal e contorno) no cálculo de pré-imagens de forma seqüencial, diversas variações foram investigadas. Através dos testes apresentados no Capítulo 6, essas versões foram descartadas e refinadas até se chegar ao modelo com rotação na borda e no núcleo apresentado na seção 5.6.

Finalmente, foi especificada uma configuração padrão do sistema criptográfico para efeito de testes e comparações. Esta especificação foi chamada de sistema criptográfico HCA (Hybrid Cellular Automata – based *cipher system*). A especificação do sistema HCA utiliza:

- A terceira versão do cálculo de pré-imagem em ACs não-homogêneos, chamada de modelo com rotação na borda e no núcleo (5.6);
- Blocos de texto de 128 bits;
- Número de passos de cifragem (pré-imagem) e decifragem (evolução para frente) igual a 128.
- Chaves criptográficas de 256 bits, sendo que elas devem ser palavras binárias com entropia espacial, dada pela equação (20), acima de 0,75. Essas chaves equivalem aos núcleos que especificam tanto a regra principal quanto a regra de contorno, a cada passo de cifragem e decifragem.

Os resultados para os testes realizados com o sistema HCA se mostraram promissores. Os experimentos em que foram cifrados dois blocos de texto claro idênticos, alterando-se apenas um bit na chave criptográfica, resultam em um valor de desvio padrão de 4,40% na diferença entre os blocos de texto cifrado produzidos, de acordo com a amostra de textos claros e chaves criptográficas investigada. Além disso, nos experimentos onde dois blocos de texto claro similares são cifrados com a diferença de apenas um bit entre eles, o valor do desvio padrão entre a diferença dos blocos de texto cifrado é de 4,38%, conforme a mesma amostra citada no experimento anterior. Segundo Sen e colaboradores, um valor abaixo de 10% de desvio padrão torna o sistema criptográfico seguro a um ataque de criptoanálise diferencial. Além disso, esse resultado é próximo do obtido para o sistema criptográfico AES em [Sen *et al*. 2002] e bem abaixo do obtido para o sistema criptográfico DES, na mesma referência.

Devido à originalidade e aplicabilidade do método proposto nessa dissertação, solicitamos registro de patente do sistema criptográfico junto ao INPI. O número do protocolo referente ao depósito do pedido de patente efetuado na data 04/09/2007 é INPI 014070006645.

8.1 Perspectivas de trabalhos futuros

Embora o método proposto como padrão nesta dissertação tenha apresentado bons

resultados em todos os experimentos efetuados, mais testes devem ser realizados para confirmar a robustez do mesmo. Por exemplo, podem ser empregados diferentes métodos para avaliar a aleatoriedade produzida nos textos cifrados, além da entropia, assim como outras técnicas de criptoanálise, além da diferencial, como a criptoanálise linear.

Além disso, a abordagem desenvolvida nesse trabalho foi focada em simulações computacionais e no conhecimento acerca dos ACs sob o enfoque dos sistemas dinâmicos, que nos levaram ao modelo proposto como padrão. Uma outra abordagem que pode ser realizada no futuro é através da modelagem e análise matemática do método. Para tal, inicialmente, os processos de cifragem e decifragem devem ser modelados através de funções matemáticas. A segurança de alguns métodos criptográficos foi comprovada a partir da aproximação dos mesmos com problemas matemáticos clássicos. Por exemplo, no caso do algoritmo RSA utilizado na criptografia assimétrica, sabe-se que ele está ligado ao problema da fatoração de números primos que é não-polinomial. Uma abordagem similar poderia ser aplicada buscando aproximar as funções de cifragem/decifragem do método proposto nessa dissertação de algum problema não-polinomial conhecido. Não temos conhecimento de algum método baseado em ACs que tenha sido analisado com esse tipo de abordagem.

Uma outra proposta para trabalhos futuros, refere-se a uma comparação com outros métodos criptográficos, com o intuito de avaliar o tempo de processamento, a quantidade de memória utilizada, a qualidade dos textos cifrados produzidos, dentre outros.

Uma extensão possível para esse método seria a elaboração de uma versão bidimensional do mesmo. Se conseguirmos elaborar um método com paralelismo no plano, podemos conseguir tornar mais eficiente a criptografia de mensagens mais elaboradas. Acreditamos que tal tipo de método teria uma grande aplicabilidade na cifragem de mensagens em duas dimensões, como é o caso das imagens binárias. Uma das aplicações potenciais para esse método é a inserção de marcas d'água em imagens binárias [Pamboukian 2007].

A implementação em hardware do método criptográfico proposto é a continuidade mais direta desse trabalho, uma vez que toda a motivação para o emprego de ACs em criptografia reside na possibilidade de implementação eficiente. Dessa forma, seria empregado de fato o paralelismo alcançado pelo modelo.

Nesta dissertação, foi investigado o uso do cálculo de pré-imagens em um sistema criptográfico simétrico. Acreditamos que existe a possibilidade de se criar um

método baseado no cálculo de pré-imagens para ser empregado em um sistema criptográfico assimétrico.

Apêndice A Análise preliminar

Análise preliminar para o modelo básico

A Figura 61 mostra que ao executar alguns cálculos de pré-imagem consecutivos, os bits do texto claro não são revelados, mesmo conhecendo a regra de contorno. A Figura 61 (a) mostra o reticulado inicial e os bits *P1*, *P2*, ..., *P8* referentes aos bits que terão que ser descobertos. Na Figura 61 (b) apenas os bits da região de contorno são conhecidos. A Figura 61 (c) indica que a segunda pré-imagem calculada possui apenas 1 bit conhecido. Já na Figura 61 (d) que corresponde à terceira pré-imagem calculada, nenhuma informação está disponível para o criptoanalista.

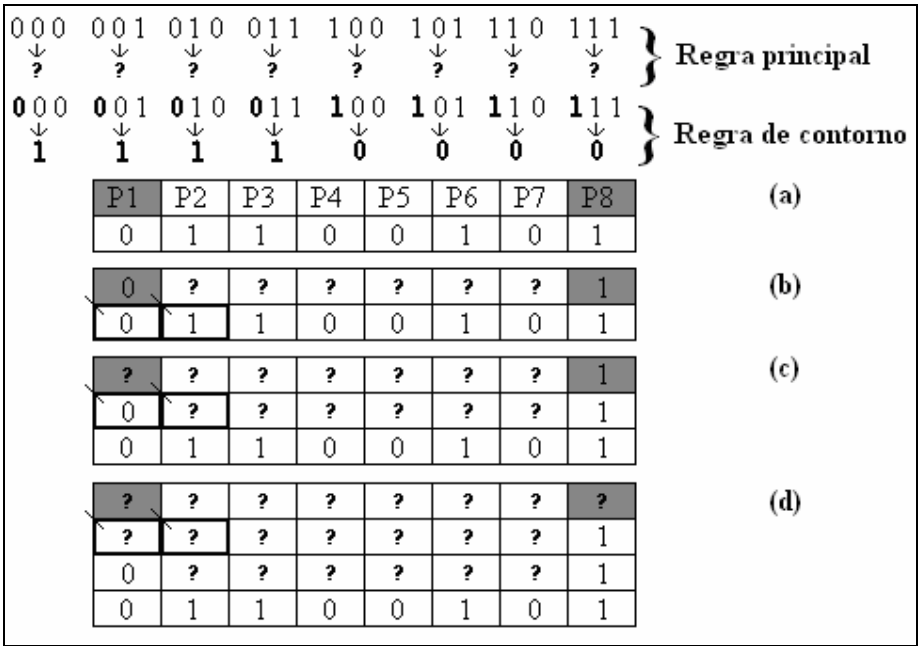


Figura 61 - Tentativa de descobrir alguma característica que possa quebrar o método proposto. Informação desconhecida: regra principal. Informação conhecida: regra de contorno e reticulado inicial. (a) Passo de tempo $t = 0$, (b) Passo de tempo $t = 1$, (c) Passo de tempo $t = 2$, (d) Passo de tempo $t = 3$.

De forma geral, a quantidade de passos para que nenhum bit seja revelado é exatamente o tamanho da vizinhança, ou seja, $(2 \times raio) + 1$.

Análise preliminar para o modelo com rotação da borda

Novamente foi montado o ataque de criptoanálise a partir do conhecimento da regra de contorno para segunda versão do método do cálculo de pré-imagem. A Figura 62 exemplifica essa situação. Os bits que sofrem transformação das regras de contorno são representados pelo

mesmo símbolo por serem conhecidos.

A Figura 62 mostra que ao executar alguns cálculos de pré-imagem consecutivos, os bits do texto claro não são revelados, mesmo conhecendo-se a regra de contorno. Da mesma maneira que para a primeira versão, é necessário $(2\times raio)+1$ passos de execução do cálculo de pré-imagem para que nenhum bit seja revelado.

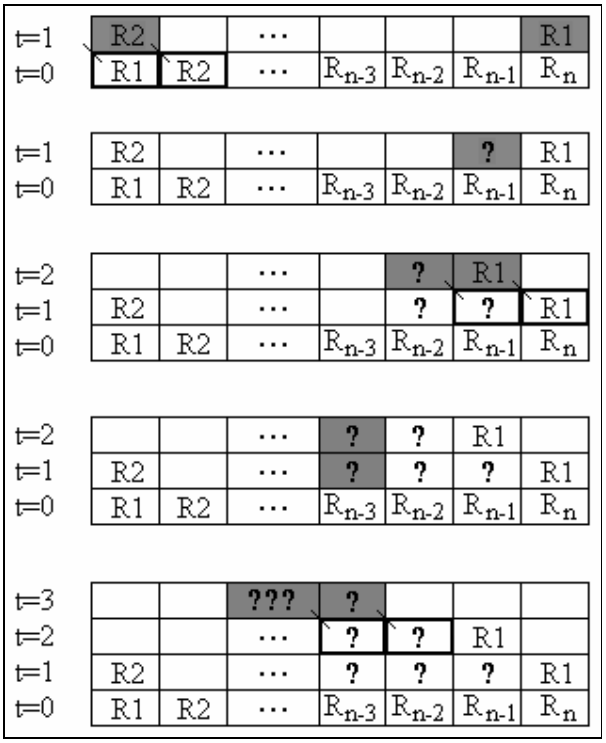


Figura 62 - Criptoanálise com a regra de contorno conhecida.

Análise preliminar para o modelo com rotação na borda e no núcleo

O método de cálculo de pré-imagem com rotação na borda e no núcleo possui análise similar ao modelo em que apenas a borda é rotacionada, quando é investigado o ataque a partir do conhecimento da regra de contorno, porém, para conseguir descobrir quais são todas as regras de contorno utilizadas em cada passo do cálculo de pré-imagem não é uma tarefa tão simples na versão em que é rotacionado a borda e o núcleo.

Uma vez que neste método são empregadas regras diferentes a cada passo de execução do cálculo de pré-imagem, o ciclo mínimo é determinado pela quantidade de bits formada pelo núcleo, e na pior das hipóteses poderíamos ter um núcleo que ao sofrer o deslocamento de um bit ficaria alternando entre uma regra e outra formando um ciclo mínimo de tamanho dois, porém núcleos com formações deste tipo poderiam ser evitados.

Apêndice B Resultado do experimento com o cálculo de pré-imagem proposto por Wuensche e Lesser realizado com regras elementares

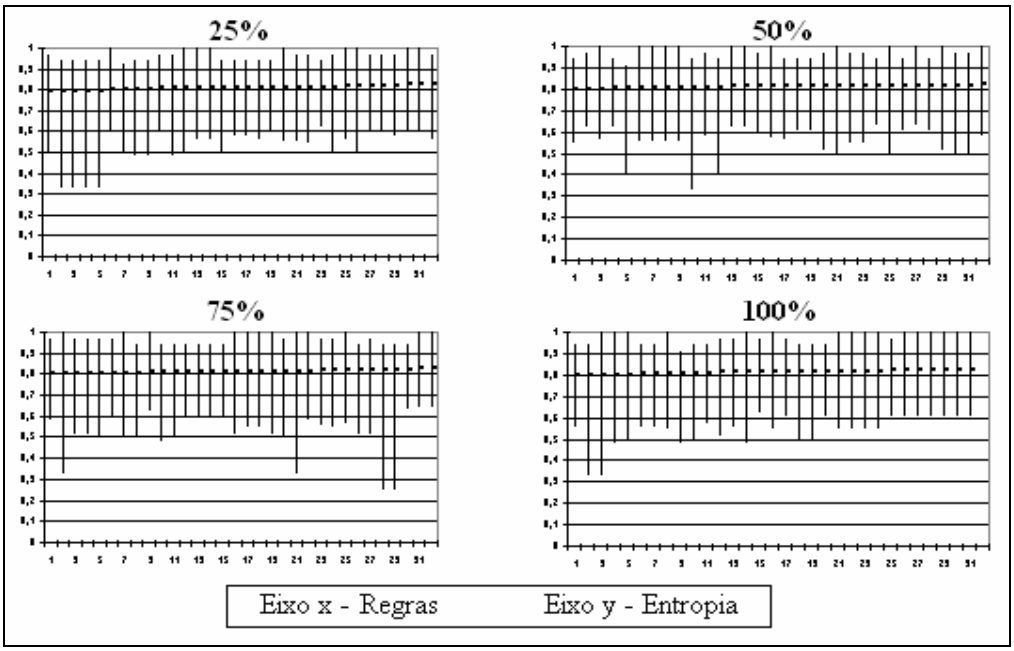
Apenas uma pré-imagem foi calculada por configuração de reticulado. Caso a pré-imagem calculada existir, é acrescentado o contador de pré-imagens calculas. Dessa forma, para reticulados de *N* bits, as regras que possuem valor igual a 2^N são capazes de encontrar pré-imagem para qualquer configuração de estados possíveis deste reticulado.

Regra em decimal	**Tamanho do reticulado**					
	5 bits	**6 bits**	**7 bits**	**8 bits**	**9 bits**	**10 bits**
[15]	[32]	[64]	[128]	[256]	[512]	[1024]
[30]	[26]	[52]	[106]	[223]	[455]	[923]
[45]	[32]	[56]	[128]	[240]	[512]	[992]
[51]	[32]	[64]	[128]	[256]	[512]	[1024]
[60]	[16]	[32]	[64]	[128]	[256]	[512]
[75]	[32]	[56]	[128]	[240]	[512]	[992]
[85]	[32]	[64]	[128]	[256]	[512]	[1024]
[86]	[26]	[52]	[106]	[223]	[455]	[923]
[89]	[32]	[56]	[128]	[240]	[512]	[992]
[90]	[16]	[16]	[64]	[64]	[256]	[256]
[101]	[32]	[56]	[128]	[240]	[512]	[992]
[102]	[16]	[32]	[64]	[128]	[256]	[512]
[105]	[32]	[16]	[128]	[256]	[128]	[1024]
[106]	[26]	[52]	[106]	[223]	[455]	[923]
[120]	[26]	[52]	[106]	[223]	[455]	[923]
[135]	[26]	[52]	[106]	[223]	[455]	[923]
[149]	[26]	[52]	[106]	[223]	[455]	[923]
[150]	[32]	[16]	[128]	[256]	[128]	[1024]
[153]	[16]	[32]	[64]	[128]	[256]	[512]
[154]	[32]	[56]	[128]	[240]	[512]	[992]
[165]	[16]	[16]	[64]	[64]	[256]	[256]
[166]	[32]	[56]	[128]	[240]	[512]	[992]
[169]	[26]	[52]	[106]	[223]	[455]	[923]
[170]	[32]	[64]	[128]	[256]	[512]	[1024]
[180]	[32]	[56]	[128]	[240]	[512]	[992]
[195]	[16]	[32]	[64]	[128]	[256]	[512]
[204]	[32]	[64]	[128]	[256]	[512]	[1024]
[210]	[32]	[56]	[128]	[240]	[512]	[992]
[225]	[26]	[52]	[106]	[223]	[455]	[923]
[240]	[32]	[64]	[128]	[256]	[512]	[1024]

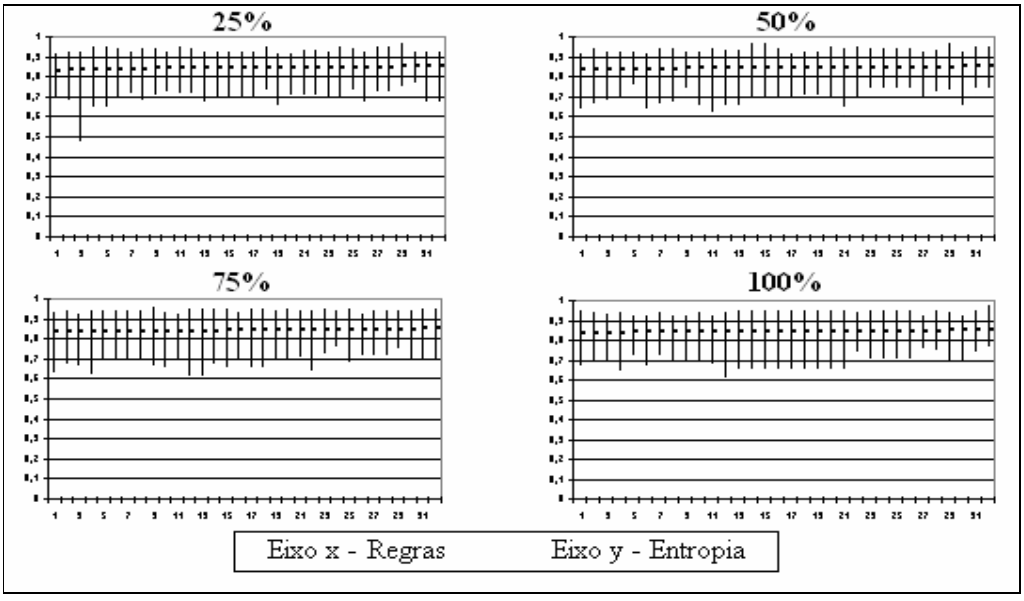
Apêndice C Resultado dos experimentos com análise de entropia no texto cifrado

Modelo básico: resultados da análise de entropia no texto cifrado para raio 1 partindo de reticulados com alta entropia.

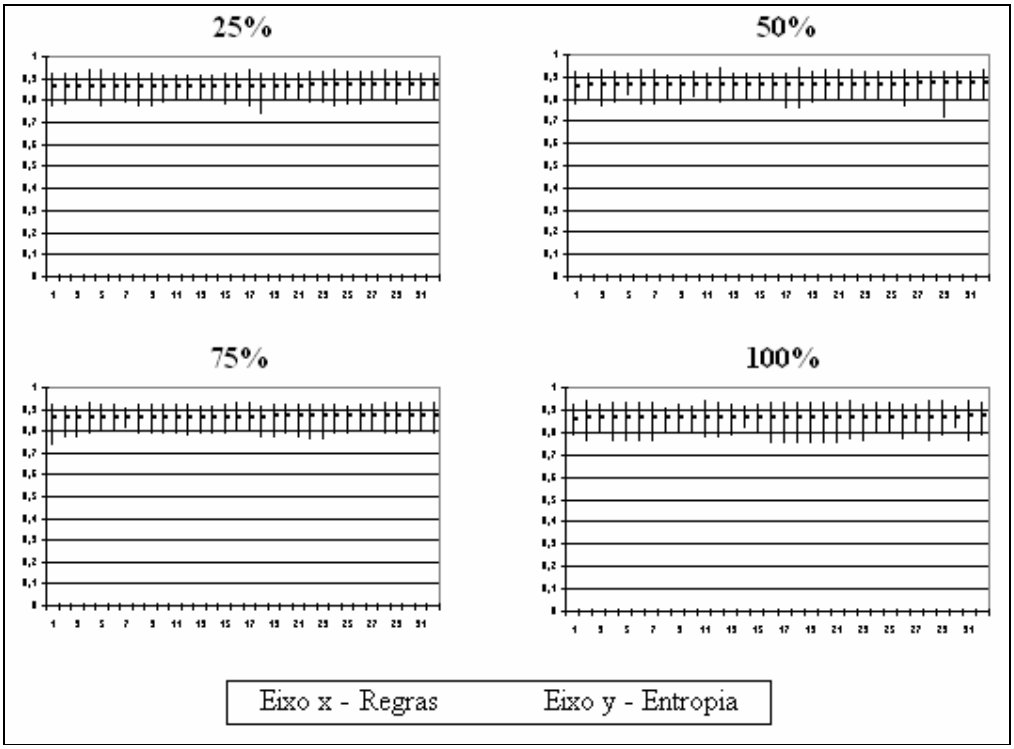
Reticulados de 16 bits



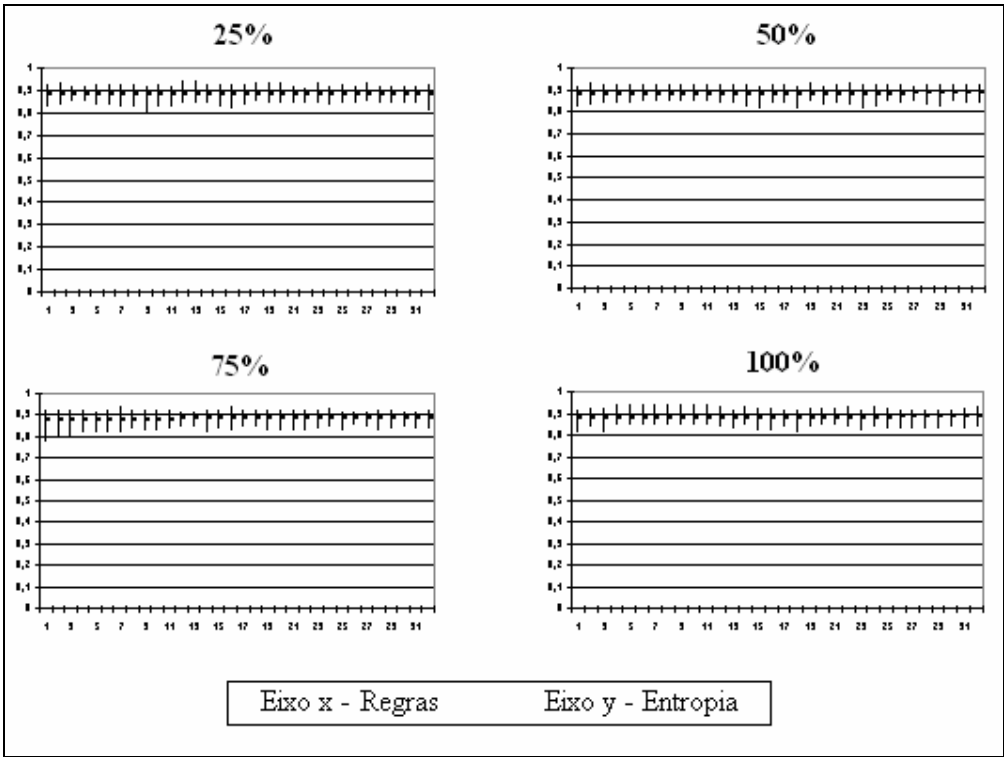
Reticulados de 32 bits



Reticulados de 64 bits

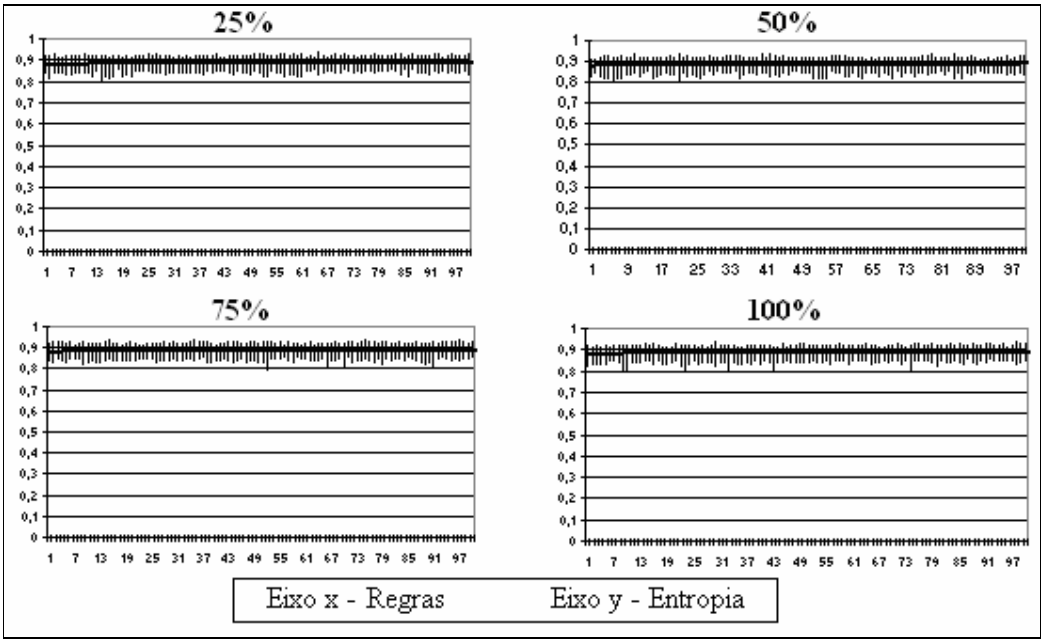


Reticulados de 128 bits



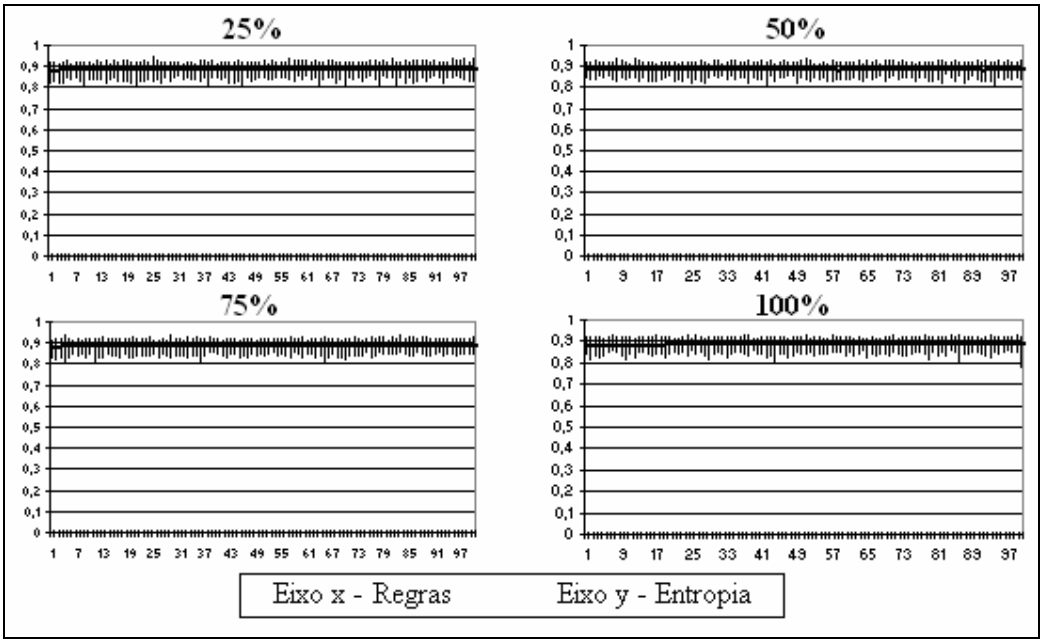
Modelo básico: resultados da análise de entropia no texto cifrado para raio 2 partindo de reticulados com alta entropia.

Reticulados de 128 bits



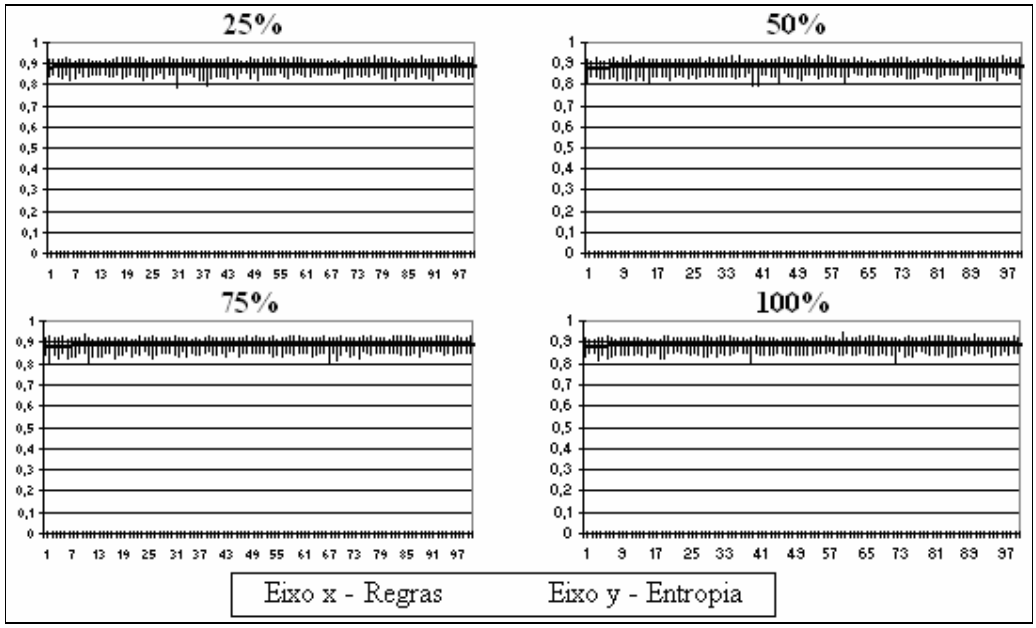
Modelo básico: resultados da análise de entropia no texto cifrado para raio 3 partindo de reticulados com alta entropia.

Reticulados de 128 bits



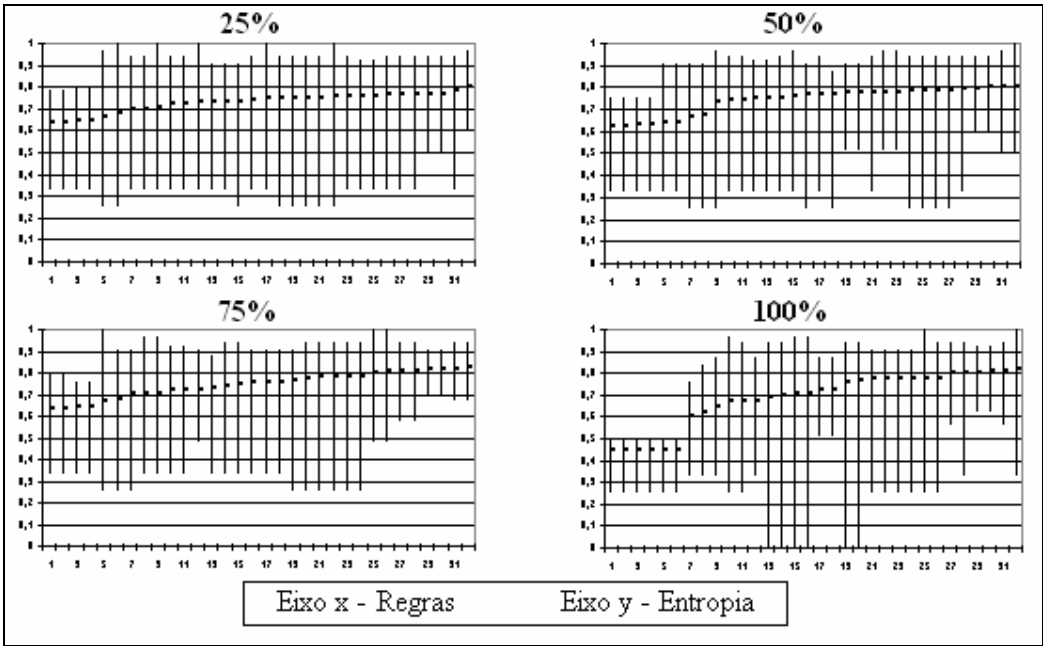
Modelo básico: resultados da análise de entropia no texto cifrado para raio 4 partindo de reticulados com alta entropia.

Reticulados de 128 bits

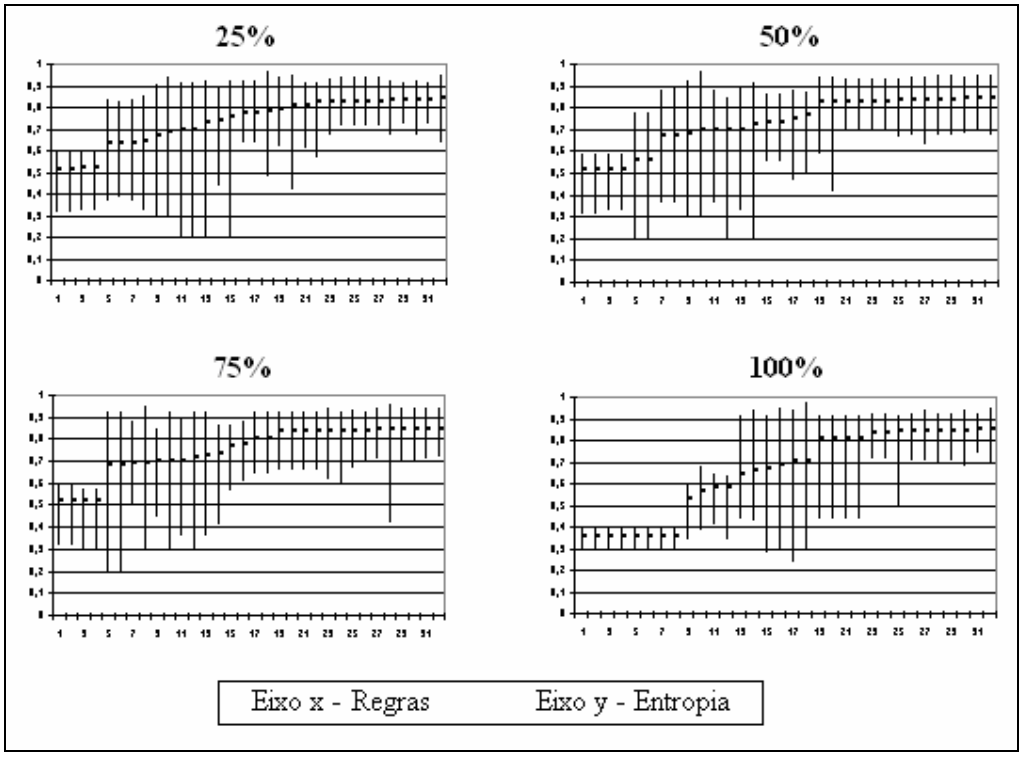


Modelo básico: resultados da análise de entropia no texto cifrado para raio 1 partindo de reticulados com baixa entropia.

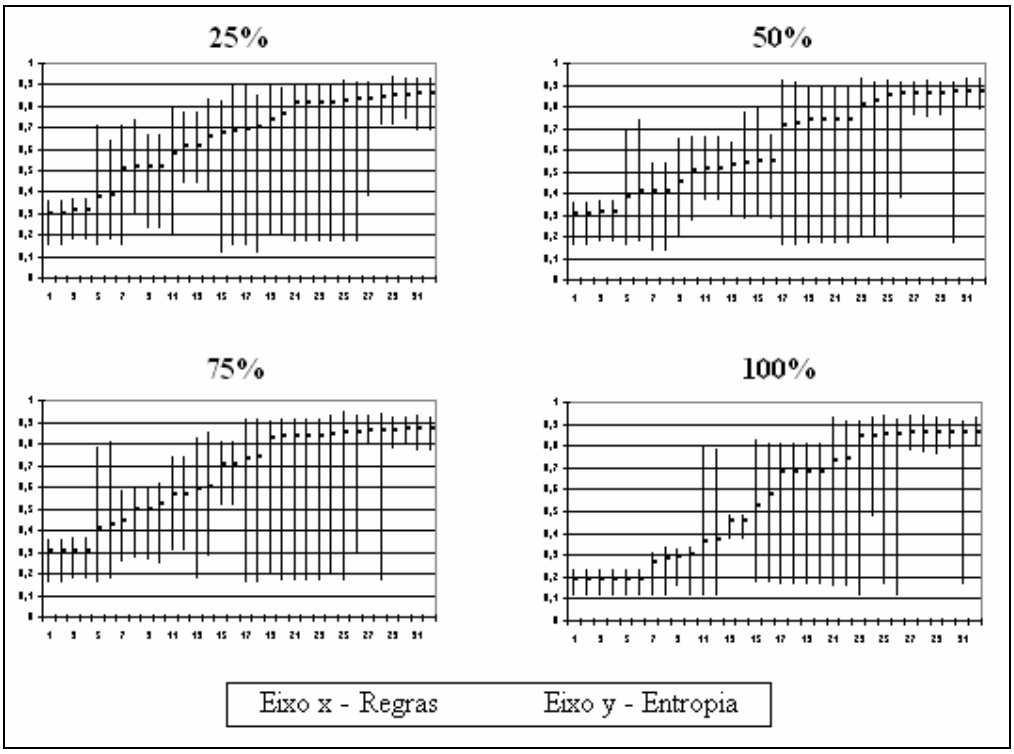
Reticulados de 16 bits



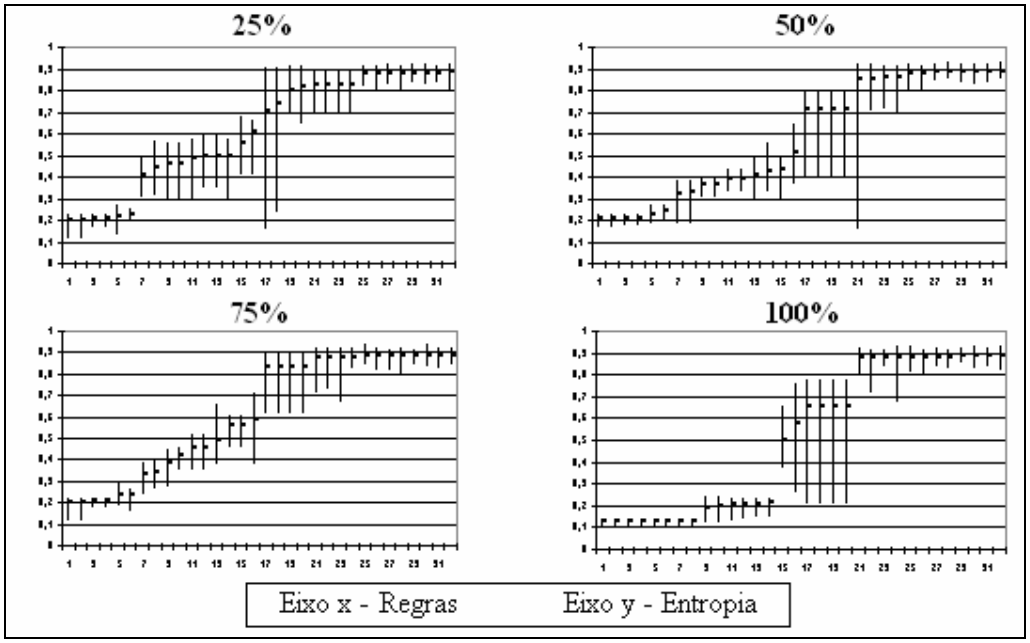
Reticulados de 32 bits



Reticulados de 64 bits

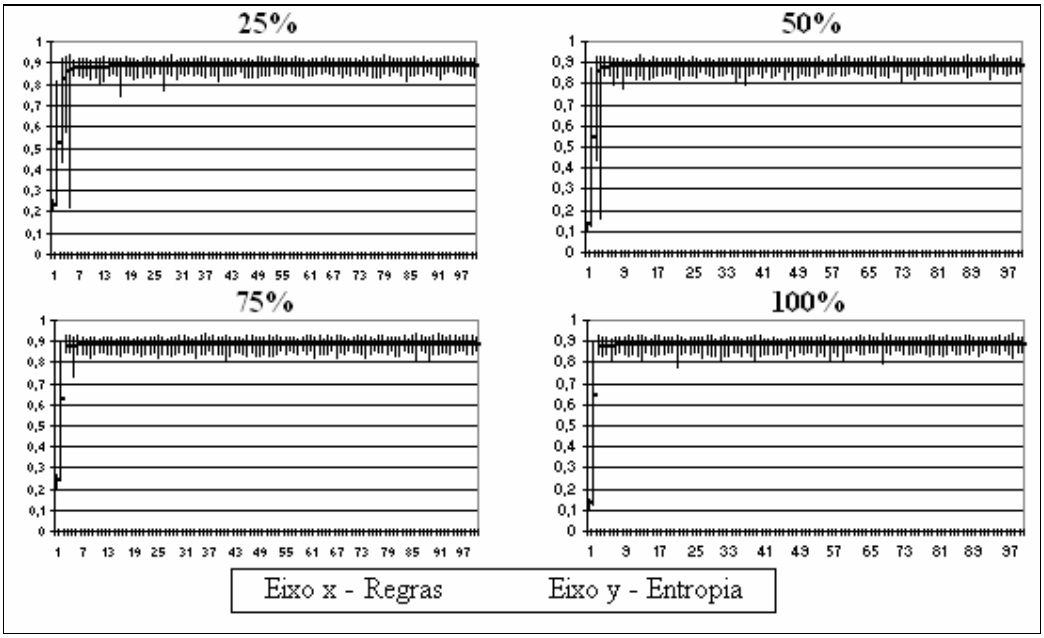


Reticulados de 128 bits



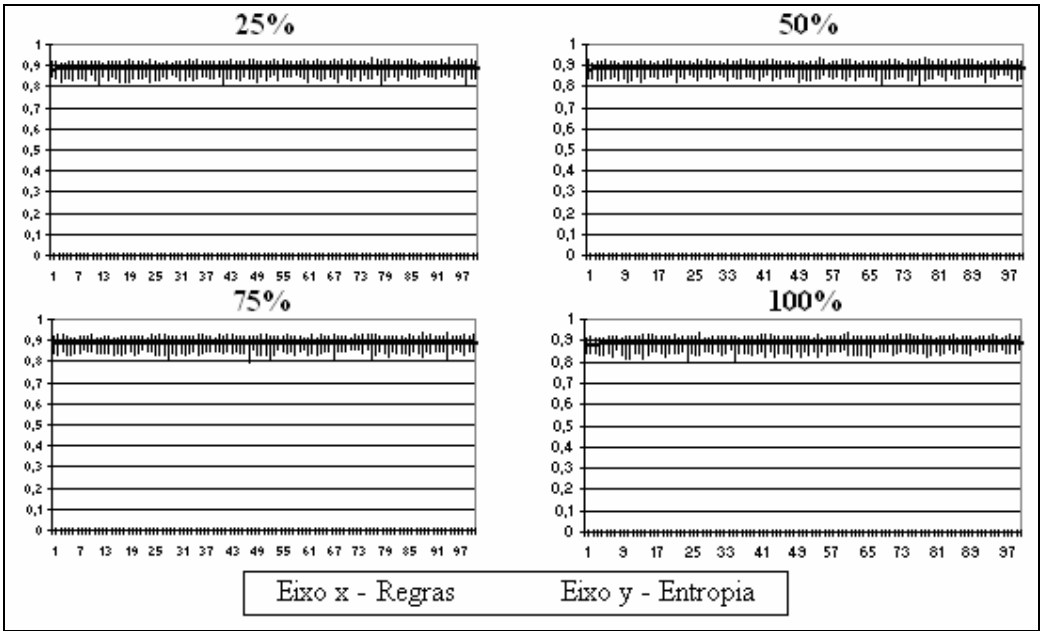
Modelo básico: resultados da análise de entropia no texto cifrado para raio 2 partindo de reticulados com baixa entropia.

Reticulados de 128 bits



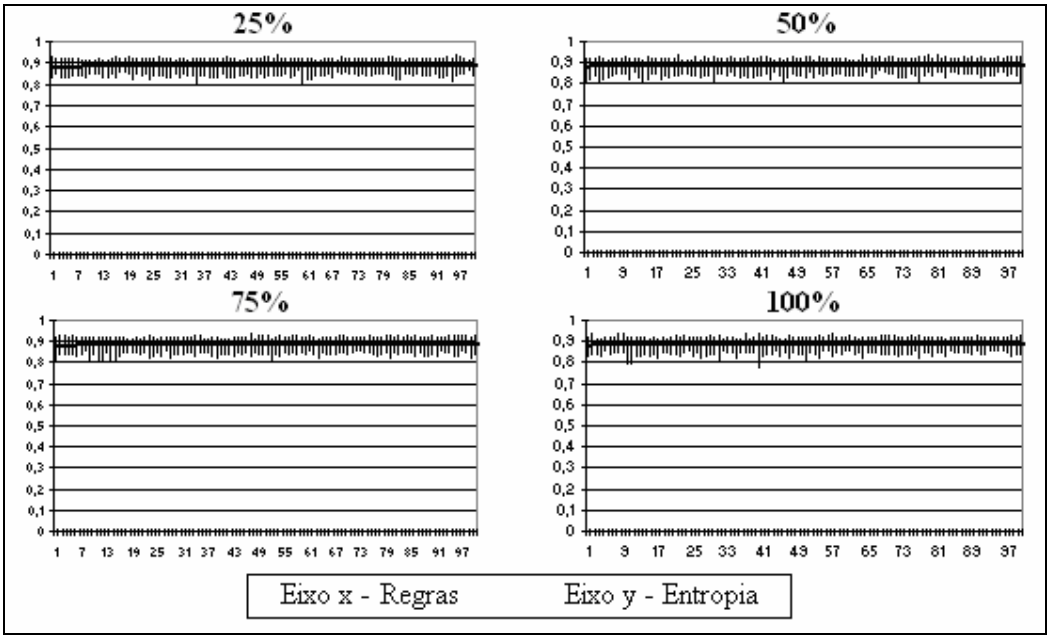
Modelo básico: resultados da análise de entropia no texto cifrado para raio 3 partindo de reticulados com baixa entropia.

Reticulados de 128 bits



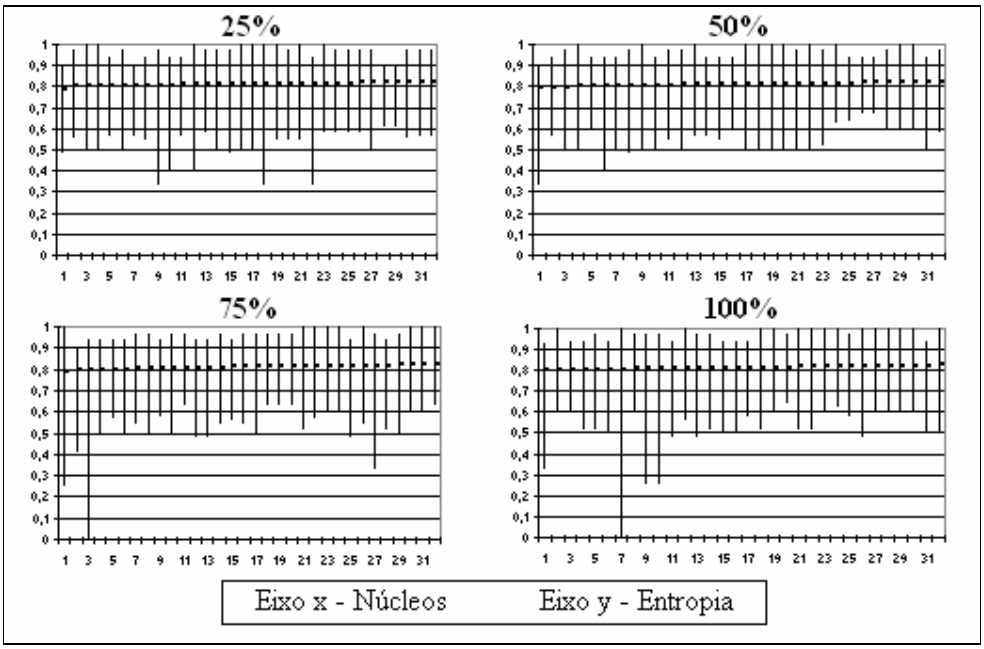
Modelo básico: resultados da análise de entropia no texto cifrado para raio 4 partindo de reticulados com baixa entropia.

Reticulados de 128 bits

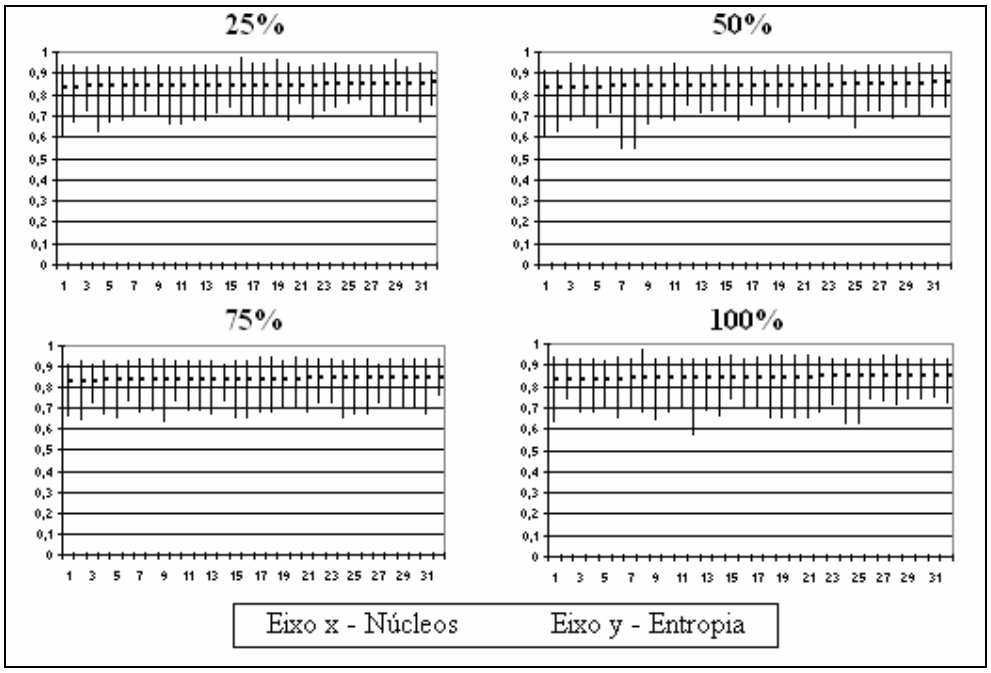


Modelo com rotação: resultados da análise de entropia no texto cifrado para raio 1 partindo de reticulados aleatórios.

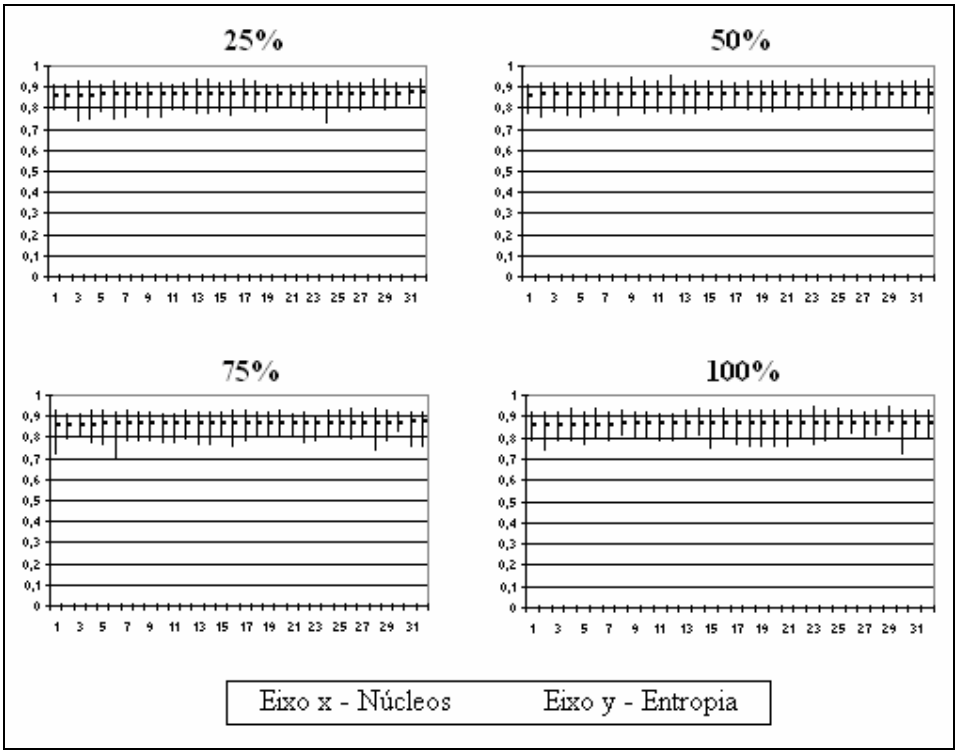
Reticulados de 16 bits



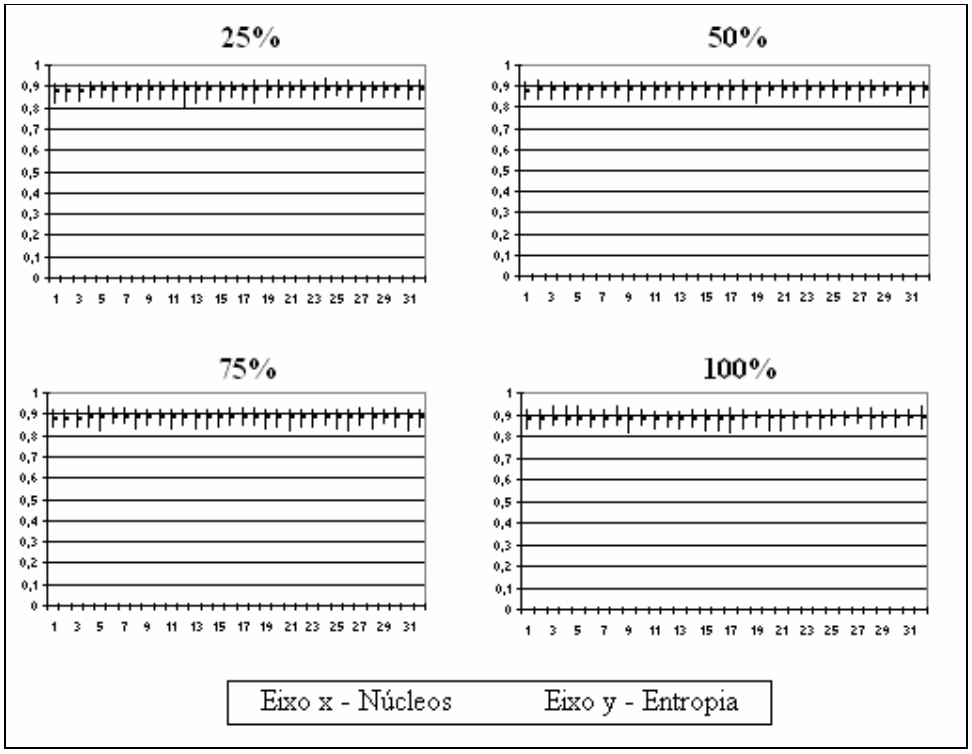
Reticulados de 32 bits



Reticulados de 64 bits

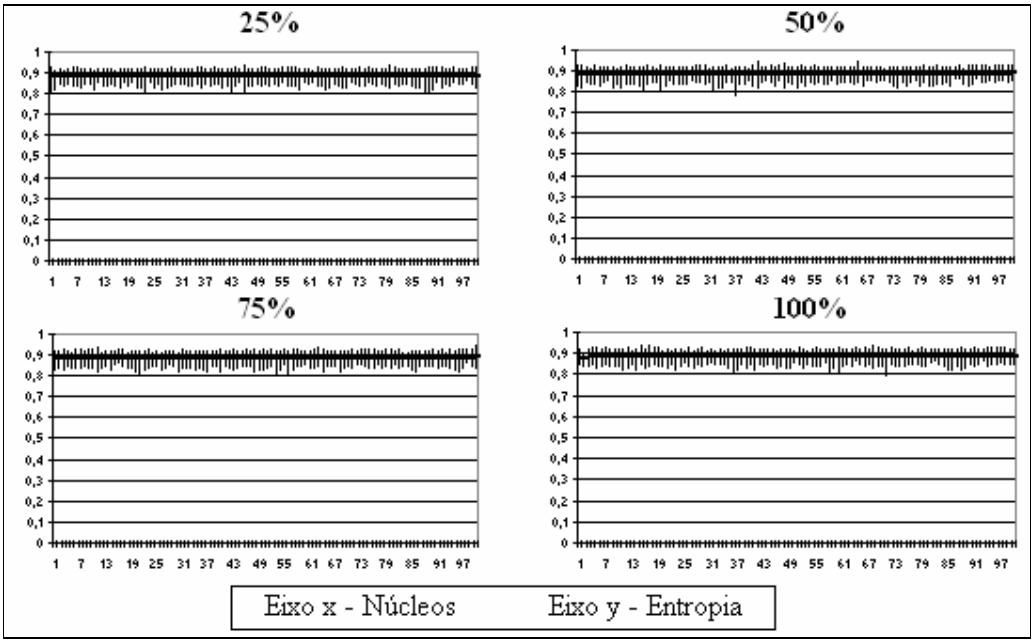


Reticulados de 128 bits



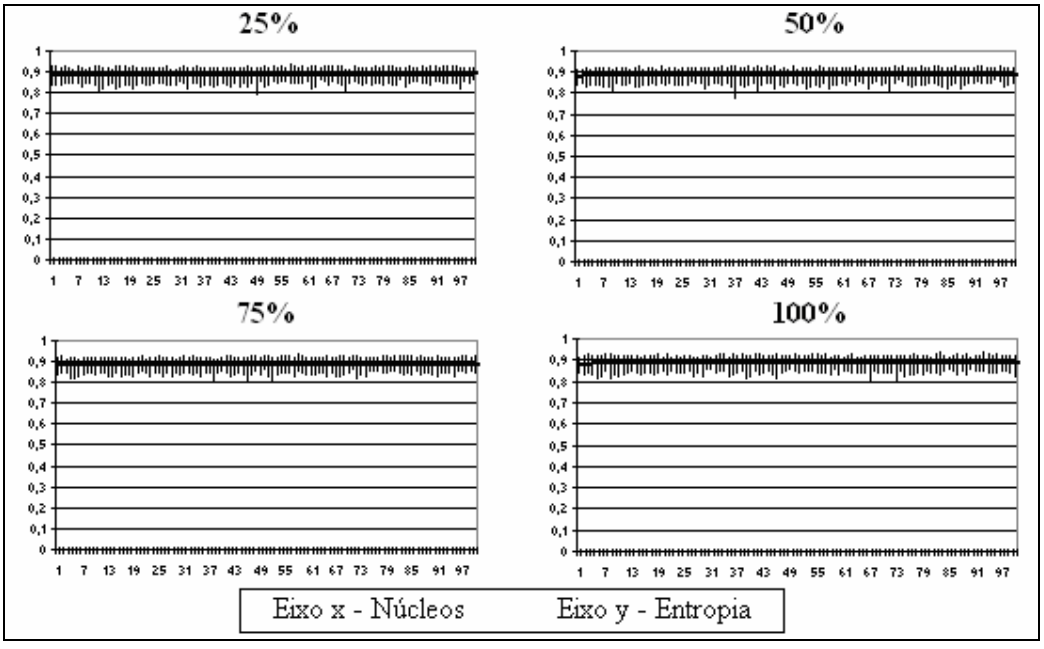
Modelo com rotação: resultados da análise de entropia no texto cifrado para raio 2 partindo de reticulados aleatórios.

Reticulados de 128 bits



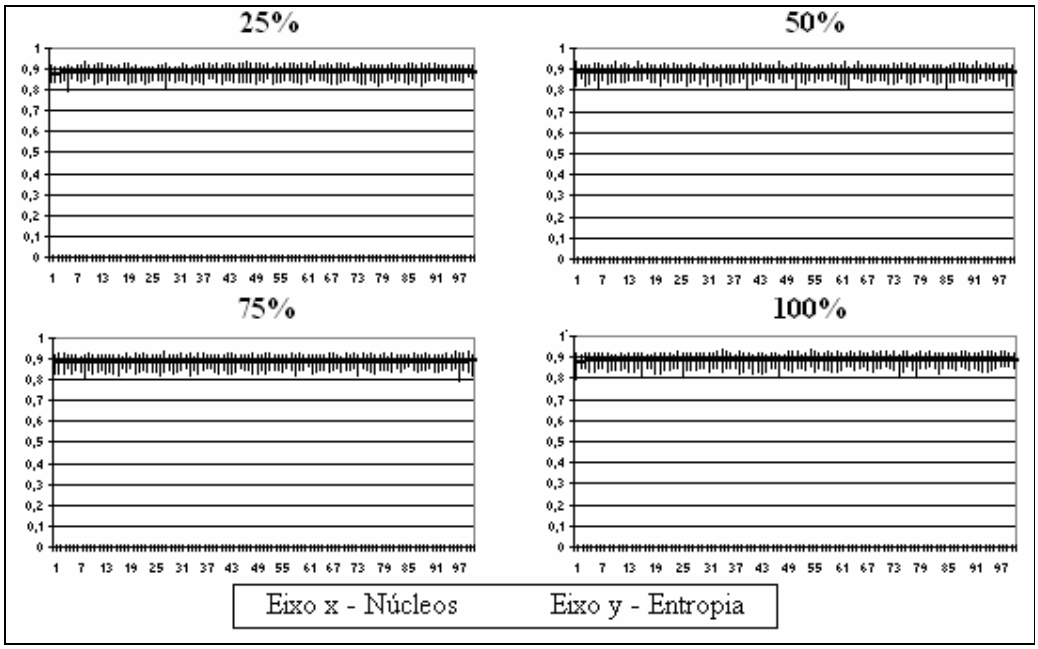
Modelo com rotação: resultados da análise de entropia no texto cifrado para raio 3 partindo de reticulados aleatórios.

Reticulados de 128 bits



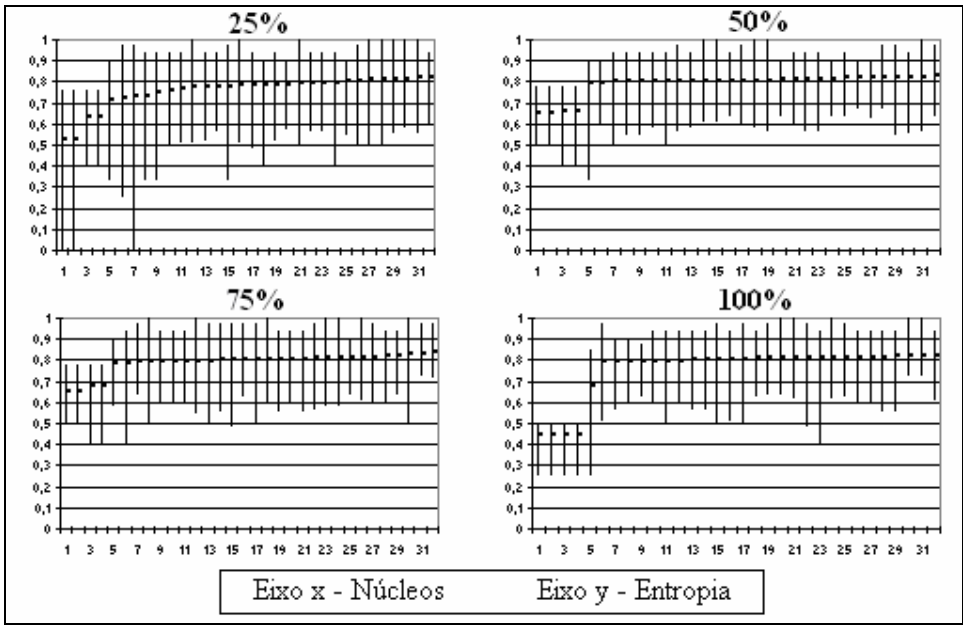
Modelo com rotação: resultados da análise de entropia no texto cifrado para raio 4 partindo de reticulados aleatórios.

Reticulados de 128 bits

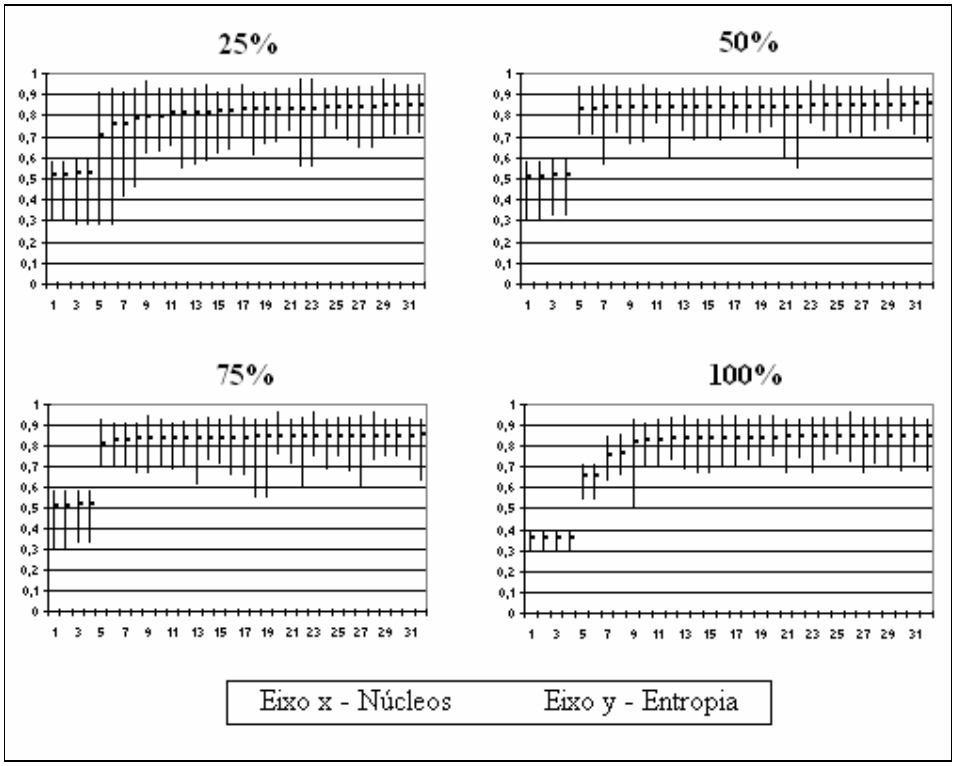


Modelo com rotação: resultados da análise de entropia no texto cifrado para raio 1 partindo de reticulados com baixa entropia.

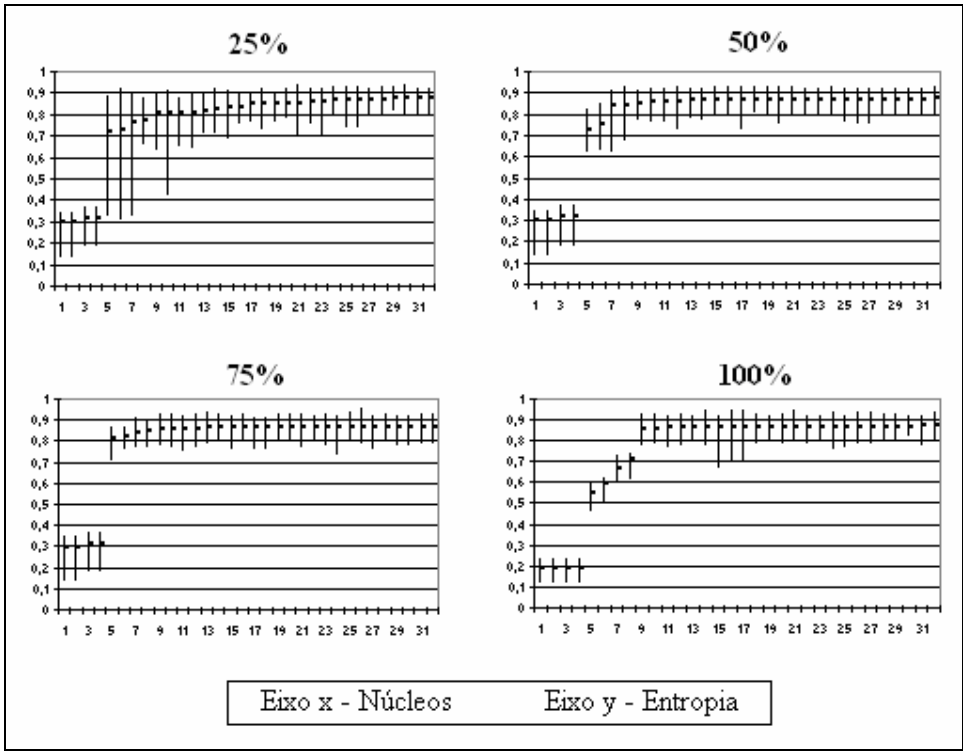
Reticulados de 16 bits



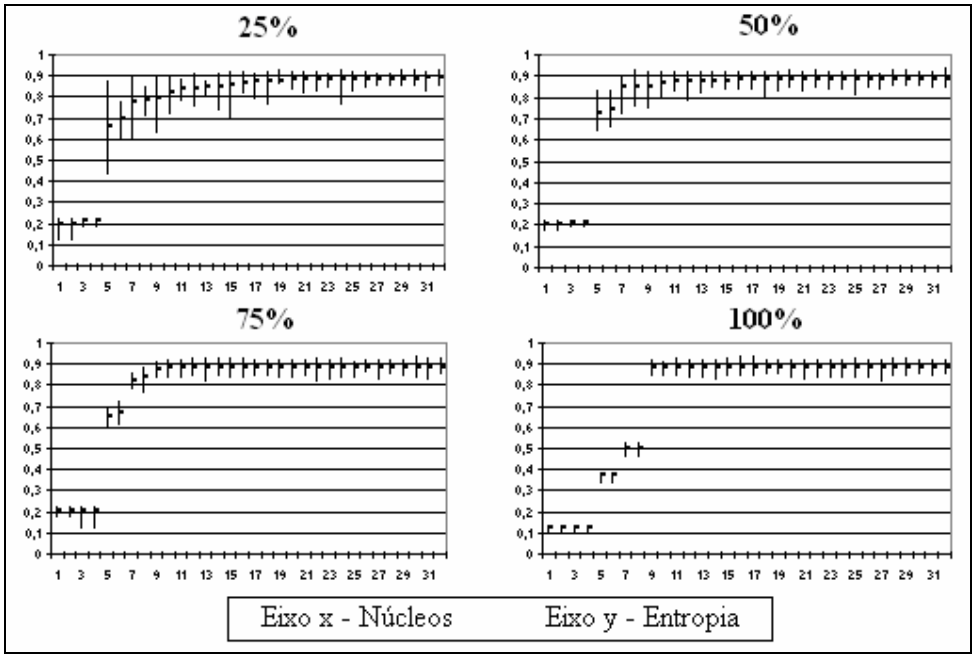
Reticulados de 32 bits



Reticulados de 64 bits

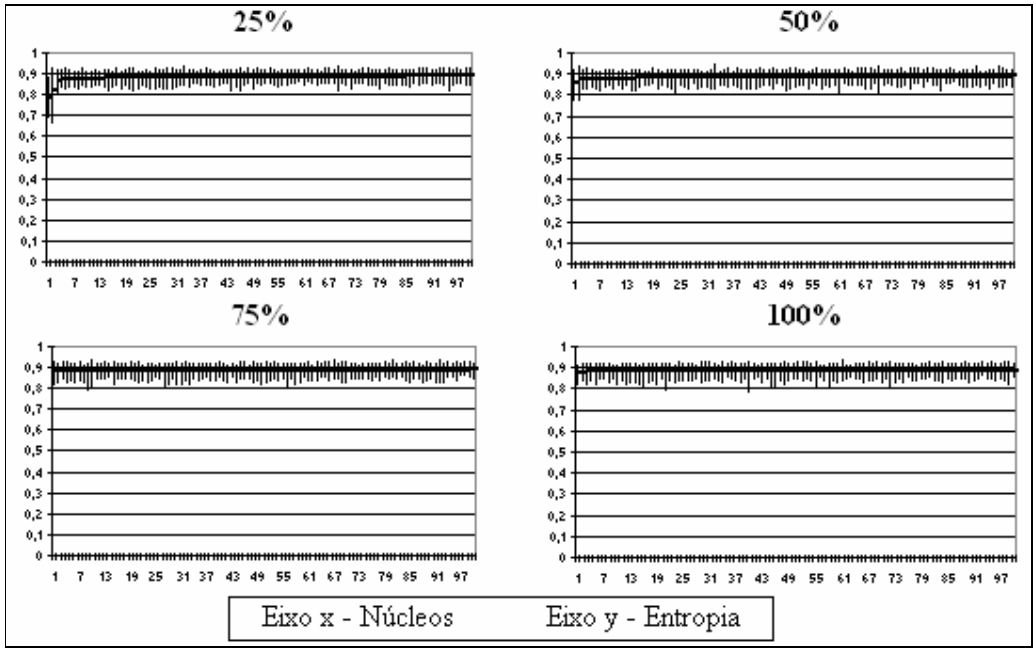


Reticulados de 128 bits



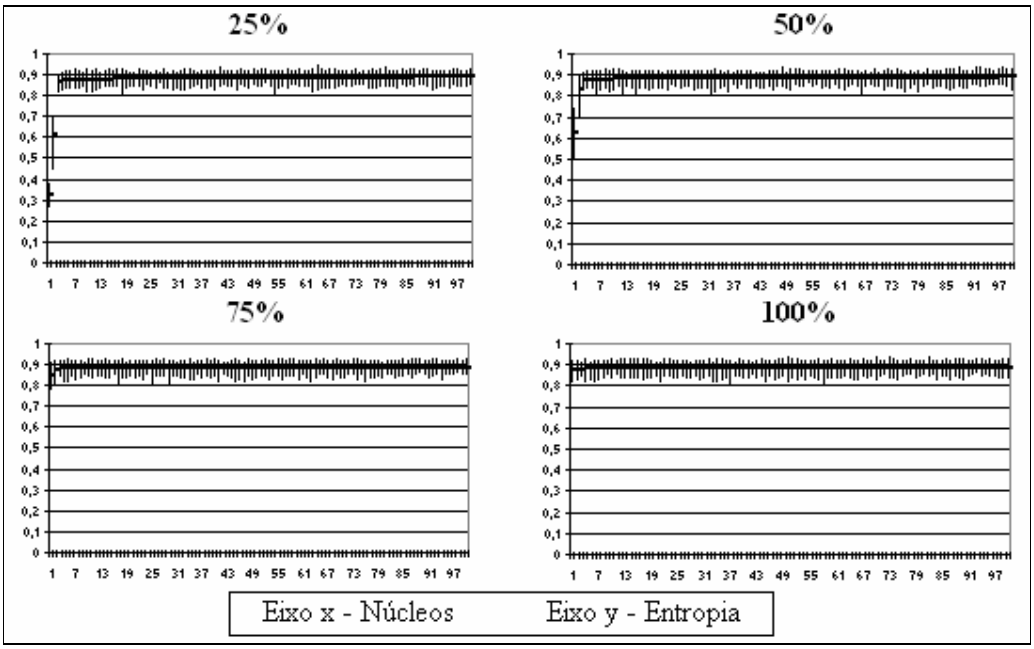
Modelo com rotação: resultados da análise de entropia no texto cifrado para raio 2 partindo de reticulados com baixa entropia.

Reticulados de 128 bits



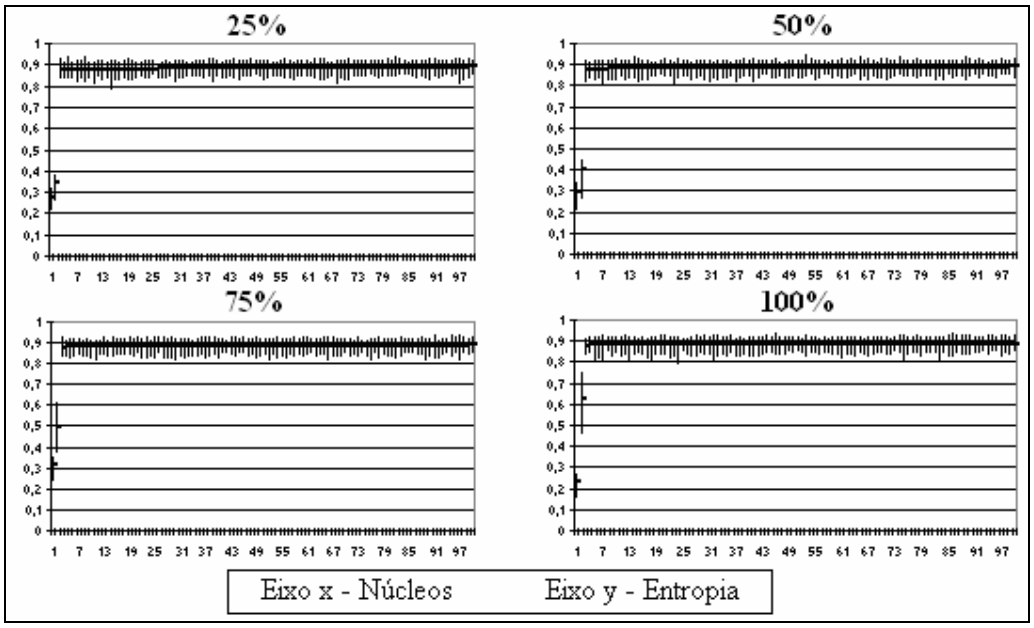
Modelo com rotação: resultados da análise de entropia no texto cifrado para raio 3 partindo de reticulados com baixa entropia.

Reticulados de 128 bits



Modelo com rotação: resultados da análise de entropia no texto cifrado para raio 4 partindo de reticulados com baixa entropia.

Reticulados de 128 bits



Apêndice D Resultado dos experimentos na propagação da perturbação de um bit no reticulado

Modelo básico: resultado da entropia e do desvio padrão na propagação da perturbação.

Raio 1 evoluindo por 50% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0,25	1	0,725511	38,593751	18,640508
32	0,2	0,98	0,648057	32,859245	20,853974
64	0,12	0,95	0,583901	30,251953	22,448502
128	0,07	0,94	0,541887	29,105631	23,270042
Média	0,16	0,9675	0,624839	32,702645	21,3032565

Raio 1 evoluindo por 75% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,741557	40,990105	17,484789
32	0,2	0,99	0,679761	34,728384	20,060699
64	0,12	0,95	0,615274	31,739128	22,098815
128	0,07	0,94	0,570247	30,444661	23,480636
Média	0,0975	0,97	0,65171	34,4755695	20,78123475

Raio 1 evoluindo por 100% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0,33	1	0,723344	39,316928	19,071308
32	0,2	0,98	0,651502	34,030467	20,975314
64	0,12	0,95	0,589585	31,740561	22,951713
128	0,07	0,94	0,547992	30,525586	23,994927
Média	0,18	0,9675	0,628106	33,9033855	21,7483155

Raio 2 evoluindo por 50% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,806197	52,257061	12,780766
32	0,2	1	0,835643	50,782251	9,847263
64	0,12	0,97	0,855967	49,862954	8,087856
128	0,07	0,94	0,870869	49,379554	7,205734
Média	0,0975	0,9775	0,842169	50,570455	9,48040475

Raio 2 evoluindo por 75% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,806817	52,518815	12,68677
32	0,2	1	0,837504	50,91669	9,555093
64	0,12	0,96	0,85855	50,084907	7,566346
128	0,07	0,94	0,874111	49,633336	6,530853
Média	0,0975	0,975	0,844246	50,788437	9,0847655

Raio 2 evoluindo por 100% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,807161	52,538562	12,613812
32	0,2	1	0,838243	51,01378	9,414386
64	0,12	0,95	0,859903	50,201565	7,274188
128	0,07	0,94	0,876228	49,771509	6,062411
Média	0,0975	0,9725	0,845384	50,881354	8,84119925

Raio 3 evoluindo por 50% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,80912	53,179127	12,127959
32	0,48	1	0,841692	51,592499	8,716434
64	0,66	0,96	0,865088	50,778157	6,179321
128	0,77	0,94	0,883217	50,380409	4,387861
Média	0,4775	0,975	0,849779	51,482548	7,85289375

Raio 3 evoluindo por 75% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,809206	53,134751	12,136823
32	0,5	1	0,841756	51,57994	8,685351
64	0,65	0,97	0,865043	50,757718	6,207564
128	0,76	0,94	0,883149	50,388229	4,38594
Média	0,4775	0,9775	0,849789	51,4651595	7,8539195

Raio 3 evoluindo por 100% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,809428	53,149873	12,096643
32	0,47	0,99	0,841939	51,5145	8,665817
64	0,68	0,96	0,865014	50,761545	6,196444
128	0,78	0,94	0,883136	50,390899	4,393901
Média	0,4825	0,9725	0,849879	51,4542043	7,83820125

Raio 4 evoluindo por 50% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,809296	53,192812	12,108375
32	0,52	1	0,841619	51,598126	8,688246
64	0,67	0,96	0,864965	50,767171	6,204828
128	0,77	0,94	0,883131	50,381327	4,403799
Média	0,49	0,975	0,849753	51,484859	7,851312

Raio 4 evoluindo por 75% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,809839	53,161001	12,0729
32	0,5	1	0,841718	51,555032	8,688264
64	0,68	0,95	0,864917	50,791782	6,220085
128	0,79	0,94	0,883049	50,396204	4,402608
Média	0,4925	0,9725	0,849881	51,4760048	7,84596425

Raio 4 evoluindo por 100% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,809509	53,208435	12,079331
32	0,32	1	0,841734	51,568753	8,694179
64	0,69	0,96	0,865253	50,785565	6,189435
128	0,78	0,94	0,883082	50,3766	4,412804
Média	0,4475	0,975	0,849895	51,4848383	7,84393725

Modelo com rotação na borda e no núcleo: resultado da entropia e do desvio padrão na propagação da perturbação.

Raio 1 evoluindo por 50% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0,25	1	0,724579	37,626758	17,891109
32	0,2	0,96	0,693643	33,820802	17,743742
64	0,12	0,95	0,664113	32,424903	17,985572
128	0,07	0,94	0,641897	31,858692	18,425731
Média	0,16	0,9625	0,681058	33,9327888	18,0115385

Raio 1 evoluindo por 75% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0,25	1	0,728345	42,03496	19,742768
32	0,2	0,98	0,713849	38,533789	19,697402
64	0,12	0,95	0,701826	37,075782	19,787946
128	0,07	0,94	0,691602	36,697924	20,189109
Média	0,16	0,9675	0,708906	38,5856138	19,85430625

Raio 1 evoluindo por 100% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0,25	1	0,72225	41,890234	20,284584
32	0,2	0,99	0,704337	39,613673	21,013671
64	0,12	0,95	0,694467	38,749513	21,368189
128	0,07	0,94	0,689068	38,172534	21,520022
Média	0,16	0,97	0,702531	39,6064885	21,0466165

Raio 2 evoluindo por 50% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,804269	46,898249	13,569278
32	0,2	1	0,82359	45,49478	11,2693
64	0,2	0,96	0,829356	44,468439	9,947719
128	0,17	0,94	0,831294	44,0925	9,04295
Média	0,1425	0,975	0,822127	45,238492	10,95731175

Raio 2 evoluindo por 75% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,809269	51,266497	12,619631
32	0,27	1	0,840177	49,94247	9,241939
64	0,34	0,95	0,861704	49,17953	6,961969
128	0,43	0,94	0,876572	48,803493	5,444391
Média	0,26	0,9725	0,846931	49,7979975	8,5669825

Raio 2 evoluindo por 100% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,809359	53,077185	12,162073
32	0,24	1	0,840996	51,504439	8,774866
64	0,48	0,95	0,864807	50,73297	6,245511
128	0,55	0,95	0,882951	50,374228	4,415558
Média	0,3175	0,975	0,849528	51,4222055	7,899502

Raio 3 evoluindo por 50% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0,25	1	0,794112	46,217501	14,099164
32	0,2	1	0,817924	44,947687	11,929721
64	0,13	0,97	0,823023	43,938485	10,803961
128	0,07	0,94	0,824814	43,590656	9,930834
Média	0,1625	0,9775	0,814968	44,6735823	11,69092

Raio 3 evoluindo por 75% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,807556	50,90031	12,959415
32	0,2	1	0,834505	49,287063	10,197418
64	0,15	0,96	0,85582	48,565155	8,121529
128	0,12	0,95	0,869649	48,266515	6,87989
Média	0,1175	0,9775	0,841883	49,2547608	9,539563

Raio 3 evoluindo por 100% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,805743	52,661997	12,790464
32	0,2	0,99	0,836731	51,007533	9,699211
64	0,15	0,95	0,85951	50,294107	7,297249
128	0,13	0,94	0,877291	49,925196	5,697839
Média	0,12	0,97	0,844819	50,9722083	8,87119075

Raio 4 evoluindo por 50% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,788179	44,719374	13,854317
32	0,2	0,99	0,812755	44,764718	12,22425
64	0,12	0,95	0,820021	43,866014	11,160882
128	0,07	0,94	0,821206	43,397319	10,536719
Média	0,0975	0,97	0,81054	44,1868563	11,944042

Raio 4 evoluindo por 75% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,806396	50,682497	13,148612
32	0,2	1	0,830084	49,113688	10,767353
64	0,12	0,96	0,850605	48,361656	9,107567
128	0,07	0,95	0,86443	47,973469	8,088805
Média	0,0975	0,9775	0,837879	49,0328275	10,27808425

Raio 4 evoluindo por 100% do tamanho do reticulado

Tam Ret	**Ent. Min**	**Ent. Max**	**Ent. Med**	**Media 1 bit**	**Des. Padrão**
16	0	1	0,803563	52,538812	13,125543
32	0,2	1	0,832967	50,745189	10,404015
64	0,12	0,95	0,853755	49,933031	8,687604
128	0,07	0,94	0,870867	49,50321	7,526135
Média	0,0975	0,9725	0,840288	50,6800605	9,93582425

Apêndice E Resultado dos experimentos na propagação da perturbação de um bit no núcleo

Modelo com rotação: resultado da entropia e do desvio padrão na propagação da perturbação.

Raio 1 evoluindo por 50% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0,25	1	0,809367	53,273177	12,106664
32	0,54	0,99	0,841192	51,408857	8,782283
64	0,69	0,95	0,864997	50,788409	6,215803
128	0,78	0,93	0,883074	50,417155	4,394079
Média	0,565	0,9675	0,849658	51,4718995	7,87470725

Raio 1 evoluindo por 75% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0,33	1	0,809956	53,221095	12,11367
32	0,43	0,98	0,841192	51,639581	8,764445
64	0,68	0,95	0,865316	50,763804	6,167861
128	0,8	0,95	0,883057	50,385743	4,407828
Média	0,56	0,97	0,84988	51,5025558	7,863451

Raio 1 evoluindo por 100% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0,33	1	0,810244	53,11172	12,117546
32	0,57	0,99	0,841636	51,542187	8,65899
64	0,66	0,95	0,864906	50,774997	6,191007
128	0,77	0,94	0,883323	50,4251	4,394387
Média	0,5825	0,97	0,850027	51,463501	7,8404825

Raio 2 evoluindo por 50% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,807633	52,530253	12,669632
32	0,32	0,99	0,841524	51,628435	8,713034
64	0,67	0,97	0,865027	50,792032	6,204342
128	0,78	0,94	0,883107	50,393826	4,418522
Média	0,4425	0,975	0,849323	51,3361365	8,0013825

Raio 2 evoluindo por 75% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,809109	53,222501	12,156441
32	0,5	1	0,841937	51,586252	8,661262
64	0,66	0,95	0,865014	50,780153	6,186248
128	0,78	0,95	0,883094	50,396758	4,396213
Média	0,485	0,975	0,849789	51,496416	7,850041

Raio 2 evoluindo por 100% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,80959	53,168064	12,107404
32	0,47	1	0,842063	51,547533	8,664525
64	0,65	0,96	0,865068	50,772017	6,197218
128	0,78	0,94	0,883156	50,374007	4,408169
Média	0,475	0,975	0,849969	51,4654053	7,844329

Raio 3 evoluindo por 50% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,668413	34,066936	20,769642
32	0,2	1	0,837904	49,84678	9,615377
64	0,64	0,95	0,864882	50,781596	6,209497
128	0,79	0,95	0,88306	50,422943	4,408538
Média	0,4075	0,975	0,813565	46,2795638	10,2507635

Raio 3 evoluindo por 75% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,740861	43,288186	19,276999
32	0,2	1	0,841643	51,392031	8,749067
64	0,67	0,96	0,865089	50,760186	6,201218
128	0,77	0,94	0,882985	50,382477	4,431439
Média	0,41	0,975	0,832645	48,95572	9,66468075

Raio 3 evoluindo por 100% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,77579	48,1695	16,75199
32	0,45	1	0,841769	51,550311	8,64934
64	0,67	0,96	0,864882	50,807875	6,230347
128	0,76	0,94	0,88316	50,404501	4,401129
Média	0,47	0,975	0,8414	50,2330468	9,0082015

Raio 4 evoluindo por 50% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0	1	0,420622	13,046187	14,40869
32	0,2	0,99	0,655904	31,931436	19,122109
64	0,12	0,96	0,85861	49,180937	7,468845
128	0,77	0,95	0,883115	50,387424	4,417406
Média	0,2725	0,975	0,704563	36,136496	11,3542625

Raio 4 evoluindo por 75% do tamanho do reticulado

Tam Ret	Ent. Min	Ent. Max	Ent. Med	Media 1 bit	Des. Padrão
16	0,25	1	0,466071	16,888937	17,636187
32	0,2	1	0,754136	41,542	17,182218
64	0,12	0,96	0,864675	50,670749	6,297564
128	0,78	0,94	0,883149	50,40881	4,402173
Média	0,3375	0,975	0,742008	39,877624	11,3795355

Raio 4 evoluindo por 100% do tamanho do reticulado

Tam Ret	**Ent. Min**	**Ent. Max**	**Ent. Med**	**Media 1 bit**	**Des. Padrão**
16	0	1	0,507209	21,530312	21,140075
32	0,2	1	0,803109	47,106126	14,028062
64	0,49	0,96	0,865051	50,771064	6,202734
128	0,78	0,94	0,88309	50,38982	4,409335
Média	0,3675	0,975	0,764615	42,4493305	11,4450515

Apêndice F Amostras de reticulados iniciais para o experimento de entropia

Amostra de reticulados com baixa entropia para 16 bits.

0101010101010101	1010101010101010	0010010000000000	0010000000000001	1101111111111011
0000000010000000	1111111101111111	0010001000000000	0110000000000000	1101111111111101
0000000110000000	1111111001111111	0010000100000000	1101101111111111	1101111111111110
1000000000000000	0111111111111111	0010000010000000	1101110111111111	0000000111000000
0000000000000001	1111111111111110	0010000001000000	1101111011111111	1110000000000000
1100000000000000	0011111111111111	0010000000100000	1101111101111111	0000000000000111
0000000000000011	1111111111111100	0010000000010000	1101111110111111	1111111000111111
0000000101000000	1111111010111111	0010000000001000	1101111111011111	0001111111111111

Amostra de reticulados com baixa entropia para 32 bits.

00001000100000000000000000000000	11110111101111111111111111111111	00010001100000000000000000000000	11101111111100111111111111111111
00001000010000000000000000000000	11110111110111111111111111111111	00010000110000000000000000000000	11101111111110011111111111111111
00001000001000000000000000000000	11110111111011111111111111111111	00010000011000000000000000000000	11101111111111001111111111111111
00001000000100000000000000000000	11110111111101111111111111111111	00010000001100000000000000000000	11101111111111100111111111111111
00001000000010000000000000000000	11110111111110111111111111111111	00010000000110000000000000000000	11101111111111110011111111111111
00001000000001000000000000000000	11110111111111011111111111111111	00010000000011000000000000000000	11101111111111111001111111111111
00001000000000100000000000000000	11110111111111101111111111111111	00010000000001100000000000000000	11101111111111111100111111111111
00001000000000010000000000000000	11110111111111110111111111111111	00010000000000110000000000000000	11101111111111111110011111111111

Amostra de reticulados com baixa entropia para 64 bits.

0101010101010101010101010101010101010101010101010101010101010101	0010000000000000000000000000000000000000000000000000001000000000
0010000000000000000000000000000000000000000000000000000000000000	0010000000000000000000000000000000000000000000000000000100000000
0011000000000000000000000000000000000000000000000000000000000000	0010000000000000000000000000000000000000000000000000001000000000
0010100000000000000000000000000000000000000000000000000000000000	0010000000000000000000000000000000000000000000000000010000000000
0010010000000000000000000000000000000000000000000000000000000000	0010000000000000000000000000000000000000000000000000100000000000
0010001000000000000000000000000000000000000000000000000000000000	0010000000000000000000000000000000000000000000000001000000000000
0010000100000000000000000000000000000000000000000000000000000000	0010000000000000000000000000000000000000000000000010000000000000
0010000010000000000000000000000000000000000000000000000000000000	0010000000000000000000000000000000000000000000000100000000000000

Amostra de reticulados com baixa entropia para 128 bits.

00010010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00010001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00010000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00010011000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00010001100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00011000100000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00011000010000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00011000001000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000

Amostra de reticulados aleatórios para 16 bits.

0101111100100111	0010000011111011	0001001110111011	0010110011010101
1110011011101000	0110010100011010	1111000101000010	1011101000000110
0111110110111001	0110000110011101	1000000101001011	0111000100010001
0010110110100000	0101010111000111	1010000010011100	1111011011101110
0010001111001110	0100100001111100	1110000100000111	0110100000110101
1010100111110011	1110111100011011	0101011100100000	1010000001110101
0110101011010010	0001101101111111	0101110101110110	1010010100110000
1111000001010000	1111110011000010	1011110101101011	0101100010101011

Amostra de reticulados aleatórios para 32 bits.

10111100110101100000101100011110	00011111101011111100100000100101	01010000110111110100101100100000	00110000100101010000100011000110
00111010111101001010100100011101	11000011001110010111110011010001	10101100001110011000110111110111	10110111100011111111111001101010
01011101010100101000001101000010	11011101011101000110010110110110	11100101010111101110011000101101	00010101111000011100111110000100
00011000111010001001110101111111	01110010100110100100000101100001	01101011100111100011101010001011	10110011100110001110000100101000
11101101011001011100010101010111	10010101101100101110111100100011	01011100101001101011001111000100	00000101100000111000001001011011
10001010000010111110011011010111	00101111100111000101000001011001	00111101111100000111000110110110	01000010010001111111111101010000
11001101001110000000111001010101	01110101101011110100100000110110	10011001110000000110101000000111	10100000000000111111000001101100
00110010001110010011000010111001	10011011101010010011111110000101	01011000010010011010001011110000	01000010110111010100000011001011

Amostra de reticulados aleatórios para 64 bits.

10111100110101100000101100011110001110101111010010101001000011101	10001100111110101100010111111110010100101110100010011010001100010
01011101010100101000001101000010000110001110100010011101011111111	11101100010110110101111100000010101110110011010111010111101111110111
11101101011001011100010101010111100010100000101111100110110101111	00010010110111011100001010101011110110111000101011000101011100001
11001101001110000000111001010101001100100011100100110000101110001	00010101100111101001100100100110010000001010100001101011001000010
00000100000100000110011110100111101110010000000001101010001100001	11001000010011000001011010011110000000001111101101111001110101000
01101111101101101000001010000001101000100001110010011100100101011	01011001110101100110111010001101000110001000110101100101110000000
11011001000001000100000010111000100100011011111001001011000100001	10101000000000100100010100000001011110000000011110110010010000010
00000111100101101110101101101111100111100110100011001001111110101	01011011010111010001101100100000100001011000111111001011001010101

Amostra de reticulados aleatórios para 128 bits.

01011111001001111111001101110100001111101101110010010110110100000001000111100111010101001111100110110101011010010111100000101000
11100001100001100100101011111011010010100000110101100111010011011100011100100001110111011100101101101111110010011101001011000101
01010001011110100011110010110000010111101010010000100001110101010011001110001101001101010100010010101110001100010011101100010111
11111111100111100010000011111011011001010001101001100001100111010101010111000111010010000111110011101111000110110001101101111111
11111100110000100101001110110000001001001011010011000111100101110101001000101111001100011111101100100101111111011000011010111111
01010011010010111011011001100110101011100110010010011111111110100011000011100101011010000101101100100011111100101000100011010010
01000010100101010010100011111000000100111011101111111000101000010100000010100101110100000100111001110000100000111010101110010000
01011101011101101011110101101011100101100001001110010101010100010011000010000010011110101111001001111011000000000111000100011000

Apêndice G Amostra de regras para o modelo básico

Regras de raio 1

Todas as regras

Regras de raio 2

	Regra		Regra
0	10111100110101100100001100101001	50	10101001000111010101011011100010
1	00001011000111101111010011100001	51	01011101010100101010001010101101
2	00111010111101001100010100001011	52	11001101001110000011001011000111
3	10000011010000100111110010111101	53	01001110010010111011000110110100
4	00011000111010001110011100010111	54	10011110011010000110000110010111
5	10011101011111110110001010000000	55	01000101101101101011101001001001
6	11101101011001010001001010011010	56	00000100111110111111101100000100
7	11000101010101110011101010101000	57	11101010100010100001010101110101
8	10001010000010110111010111110100	58	01011001110101001010011000101011
9	11100110110101110001100100101000	59	11000011001110010011110011000110
10	00001110010101011111000110101010	60	01110010100110101000110101100101
11	00110010001110011100110111000110	61	10010101101100100110101001001101
12	00110000101110011100111101000110	62	10101011101000100101010001011101
13	00000100000100001111101111101111	63	10100000111010110101111100010100
14	01100111101001111001100001011000	64	00100111111001001101100000011011
15	10111001000000000100011011111111	65	00001101011100111111001010001100
16	01101010001100011001010111001110	66	10101100001110010101001111000110
17	01101111101101101001000001001001	67	00111010100010111100010101110100
18	10000001010000000111111010111111	68	01011100101001101010001101011001
19	11010001000011100010111011110001	69	10110011110001000100110000111011
20	11011001000001000010011011111011	70	01110110000110011000100111100110
21	01000000101110001011111101000111	71	01010011111000011010110000011110
22	10010001101111100110111001000001	72	10101110010010010101000110110110
23	01001011000100011011010011101110	73	01011000011001111010011110011000
24	00000111100101101111100001101001	74	10100110000011110101100111110000
25	10011010101001011010011001101001	75	10011001100101100101011010100110
26	01010101100110100101011010101001	76	01100110101001100110011001011001
27	01011010100110011010101001100101	77	10100101101001100101101010010101
28	10010101010110100110010101011001	78	01100101101010010110010110011010
29	01010110100101011010100110010101	79	10010110101010010110100110010101
30	10010110101001100110101010101010	80	01100101011001101001101001101001
31	10101001101001100110100101100110	81	01010101011001011010101010011010
32	10100101011001100110011001101010	82	10101001100110011001010110011001
33	10010101100110010101010110011010	83	01100110100101101010011001100101
34	10101001011010011010011001101010	84	10100101010110100101101010010110
35	01010101101010010110011001100110	85	01101010010110011001011010011001
36	01011010010110010101101010010110	86	10010110011001101001101001011001
37	01011010010101011001101010010110	87	10011001100110101001100101011001
38	01010101011001010101011001010101	88	10011001010101011010100110011010
39	01101001011010101001100101101010	89	01011001011010101010100101100101
40	10011010100101100101010101010101	90	01010101101001100110101001011010

41	01101001100110010101101001010110	91	10011001101001010101101010010110
42	01101001101010101001101001101001	92	01011010100110011001010110011010
43	10010101010101100110010101010101	93	01100110101001011001100101101001
44	10100110010101100101010110101001	94	10011010010110101010010101100101
45	10100110100101100101010101100101	95	01101010011010010101011010010110
46	01100101010101011001101010010101	96	01100110010110101010100101010110
47	10010110010101101001101010101001	97	10011001101010010110010110010110
48	01100101100110100101011001010110	98	01100110100101010110100101101010
49	01010101011010101001011001101001	99	10011001011010010101010110101010

Regras de raio 3

	Regra
0	01011111001001111110011011101000011111011011100100101101101000001010000011011000000110010001011110000010010001101101001001011111
1	11000111001000011101110111001011011011111100100111010010110001010011100011011100010001000110100100100000011011000101101001110
2	01010001011110100011110010110000010111101010010000100001110101011010111010000101110000110100111101000010101101111011111000101010
3	00110011100011010011010101000100101011100011000100111011000101111100110001100101100101010111011010100011100111011000100111010000
4	11111111100111100010000011111011011001010001101001100001100111010000000001100001110111110000010010011010111001011001111001100010
5	01010101110001110100100001111100111011110001101100011011011111111010101000111000101101111000001100010000111001001110010010000000
6	01010010001011110011000111111011001001011111111011000011010111111110101101110100001100111000000100110110100000000100111100101000000
7	01010011010010111101101100110011010101110011001001001111111101010101100101101000010010011001100101010001100110110100000000000101
8	00110000111001010110100001011011001000111111001010001000110100101100111000110101001011110100100110111000000101011011100101011001
9	01000010100101010010100011111000000100111011011111110001010000101011110101101010110101110000011111101000100010000011101011111101
10	10000001010010111010000010011100111000010000011101010111001000000111111010101000101111011000110011110111110001010100011011111
11	01011101011101101011110101101011101011000010011100101010101000101010001010000101001010001010011101100011010101010111101
12	01100001000000100111101011110010011101100000000001110001000110001001111011110110000101000011011000010011111111110001110111001111
13	00001100000010001100101010110000001010001010111010011010001111001111101110011010101001111101011101010001011000101100111111
14	11110000110000010111011000101011001010011110010001001100000010000000111100111101000100111010100110101100001101110110011111101111
15	10100000011101011010010100110000010110001010101110011011001001010101111110001010010110101100111110100111010101000110010011011010
16	01101010111101000000111100100011101000110011100010000100100101100101010000101111100001101110001011001100011100111011101111010
17	11011011101000000110000110100110001111010000111011011101110101111001001000101111110011110011010110011100010111100010010001000101000000
18	00111010111011010111110101001000011001111001001110111111010011111100010100010010100000101011011110011000011011000100000010110000
19	01000110000000000110010110110000111100000111110000110010001101111101110011111111100101001001111000011111000001111001101110010000
20	00000100011011000110000111101011100110001111101111001100000001011111011100100111001111000010100011000111000001000011001111111101
21	01001000110000001111010000011001111111011110101000001100000101010110111001111110000101111100110000001000000101011111001111101011
22	00011110101100100001001000000000010101001001001101010110011010111000010100110111101111111111101010110110101100101010011001010
23	01010110010001000100111001100110001100000010000011111111101100110101001101110111011000110011001110011111101111100000000000100110
24	10010001001011110110000100000000011111010100010011101001111010000110111010100001001110101111111000001010111011000101100000101111
25	01100110101010100101100101101010101010100101101001101010011001010101101010101001101001010011001011001101001101001010101010101
26	10100101011010100101100101010110101001101010011010010011001101001101010101010010110010110101001100101100110100101011001100110
27	01100110010101100110101010011001010110101010010011001101001010101011001101010011001100101001010110010101011010011001100110
28	01011010010110101001010110100110010110100110011001010110010110011001101010010101101001010110010110101001101001010110011101010
29	10101010101010101001011010100101010110010101010110101010100110100110100101100100101101001100101101001010101101001011010010110
30	01100110011001101010010110101001100101100101010110101010100101101010011010100101011010011001011010010110100110101010101010
31	01100110010110010101100110101010010110100101011010101010100110100101100101100110101010101010011010010101011001100110101010101010
32	01100110010110100110010110011010100110100110100101101001101001100110010101001011010011001011010101010101010100110011001
33	01011010010101011010010110011001101001100101010110010100110100101100101011010101010100101100110010101100101011010011001011001
34	01100101010110011001011001100110010110011001010110101010010101010110010110101001101010011010101010100101011001100101010110011001
35	10010101010101100110010110011010100110010101010110010110101001011010100101010110010101010110101001100110011010100101100101010101
36	01100110101001100110101001101001100110101010011001101001100110101001100110100101011001011010100101100110011001100101011001
37	01101001010101100101010101100101101010100110011010101001011001011010101001101001010101010101010101101010010101100101011010010101

38	0101010110100101010101011001010110100101100110011001101001010101010110011001010110011001101010011001011010010011001011010100101010101011001010101
39	1010101001010101101001010101011001101010011010010101100110011010010110011001011010101001011001010110010110100101010101011001010101
40	1001100101010101011010100110011010011001011001011010010101010110011010010101100110011001101010010110100110100101100101100110
41	0110100110011001101010100110010101010101101010100101100101011010100110010101101001011010010101101001010101100101100101101100110
42	1010011010011010100110010101010101101001010101101001100101101001010110101001100101011010100110100110101001101001100110101010
43	0101101010011001101010011010011001101010101001100110011001010101101001011010100101100101101010011010101010100110010110101010
44	0110010101101001010101010101010110100101100101001101001010101101010100101010101101010101001010101101001011001010110100110101010
45	0101010101100101011010011010010101101001010101101010100110011010100101110100101011010101010011010101001011010010101010101011001
46	0110010110010101101001010101010110101010011001010101011010010110101010101001101010101001100110010101010110100101010110011001
47	0101011010101001100110100101100101010110010110010101010101010101011001100110010110011001100101101001100110011010010110100110011001
48	0110011001101001011001010110010110011010100101101001011010010101010110010101010110101010101010101001101001010110
49	1001011001010110010110011010101001101001010101100101010101010101101010101001100110010101100101101010011001011010101001100101001
50	0010001111001110101010011111001101101010110100101111000001010000110111000011000101011000001100100101010010110100001111101011111
51	1110000110000110010010101111101101001010000011010110011101001101000111100111001101010100000100101011011110010100110001011001 0
52	1111110011000010010100111011000000100100101101001100011100101110000000110011101011010110001001111101101101001011001110000110100 0
53	1110000110101001010001100110101110011011001010100010110011010101000111100101011010111001100101000110010011010101110100110010101 0
54	1011101000000110011100010001000111110111011100110100000110101010001011111100110001110111011100000100100010001100101111001010 1
55	0011001110100011010100111001011000101100000111101011010110000111110011000101110010101100011010011101000111110000101001010011110 0
56	1001000000010001100110101011101111111111011000001001010001110011011011111101110011001010100010000000000100111110110101110001100
57	0011100001000011101000001011100111010110101111110111011000000011000111101111000010111110100001100010100101000000100010011111111
58	1001010111100110010111101101100110011010100001100000110110100000011010100001100110100001001001100110001010111001111001001011111
59	1101000101101000000101011101100011111111101001100111101000010000001011101001011111101010001001110000000001011001100001011110111
60	0110110110001000101101111100111000101101000010100001100100011111001001001111011010010000011000111010010011101011110011011100000
61	1111000000110001011101000100100100001111011100000111001100011111000011111100111010000101110110110111100001000111100001100111000000
62	1000100010011011111001010100100010011011111101100010011011010010011101110110010000011010101101110100000001001110110010010110 1
63	0110011100011001110011001100100101001101000111010000111001010100111001100011100110001100110010110101100101110001011100011001010110 0
64	1000010011111101101011000100100100000110111110110010100111100100011110100000100101001110110110111111001000001000110101100001101 1
65	0011110110101111000100000001011010110110100001011000100001111111000010010000111011111101001010010010011110100011101111000000
66	0100100000001111100010011010100101011011000111000011110011111011101111111000001110100101011010100100111001111000011000000100
67	1000101001001101100011010001111010011111111101011000100011100000011101011011001001110010111000010100000000010100111011100011111
68	1011000011101101111011101010101010010100100000011010001000010100111100010010000100001001010101001011010110111100101110111110
69	1100100101110100111001111011000001101100101011000110011001001000011011010001011000011000010011111001001101010011100110011011011
70	0001010101100110100111101001001101001010101111110010001010110100111010101001100101100001011011001011010100000011011101001001 1
71	1100001010111000110110100111000000101010011100011001010111110110001111010100011100100101100011111101010110001110011010100000 1001
72	0111010100111111011100011011011111100000100000000000111001100011100010101000000100011100100100000011111011111111111000110011100
73	0101010001011110001001111111100011101010100100111000101010010001010101110100001110110000000011000010101011011000111010101101111
74	1010000001011000010110010110111100100001101001110111001110110101111110100111101001101001000011011110010110001000110011000100
75	0101100101011010101001011010010011001100100101010101001011010011101001100110010011010011000101100101010100101010101100110010101
76	1010100101010110100101010110100101100101100110010101010011010011001011001001010101101001100110100101101010011001011001110
77	1010101010100101101001010101100101100110101001101001010101010110010110010110011010001100101101010110101010010110011010100
78	1010100101010111010011001100101100101001011010010110010011001101010010110100101100110011001010110011010010110100110011001100110
79	1001101010011001010101010110100101101010010110010101100101011010101010011010011010011010100101101001100101010101101001100110
80	0101101001011010100110010101101001100110010110101001011001101001010110011010010101010110101010011001101001100110100101011010
81	1001011001010101010101100101011010010110100110011001101010011010101010101010101001101001010101011001011001001011010100101101010
82	0101101010010101011001010101101010100110010101010110011010101001011010100110011010011001101010101010011010100110100101010101010
83	1001011001100110101010010110100101100110101010011010011010010110100101100110010101011101001010101011010011010011001010101
84	1010011001010110011010011001010101010110011001101010011010010101101010101010101010011001011010010110101010011001010101100101010
85	0110100110100111010010101100101011001101001101010101001011010100101011001101001101010100110010101011010010110010101110101010
86	1010101001010101010110100101011001101010011001010110010110010110010101011010101001101010010101011010101001011010010101101010
87	1001010110010101100101101001101010101001011001100101100101011001011010011010101010100110100101011001011010011010011001011001
88	0110100101101010010101101001011010100101101001011010010110011001011010011001010110101001100101010110101001011010011001100101
89	1001010101100101101010100110100110011010010101100101100101100101010110100110101010100110101001011001100101101010100101100101

90	01011010101001101001100110101010010101100101010101010110011010011001101001101001101001010101100110101001010110010101011010101010
91	01100101100101010101010110101010100101011001011010011001100101100110011010011010010101101001010101101010100101101010101010011010
92	10010101100110010110010110100110100101011010011001010110101010011001011010101010101010100110011010010101100101011010100101010101
93	10011010010101011010100110100110101010011010101001101001100110011001101001100101100110010110010101010110100110010101100101010110
94	10100101100101100110101001100101101010100101101010100110100101010101101001101001011001100110100101011010010110100101100101100101
95	01010110011001100110100101101001100101101010100110010110010110100110010110011001100110101010101001011001010110011001101001100101
96	10100101010110011001101010010101101001101001100101101010010101010101100110011001011010100101011010010110011001101010101001101001
97	01101010011001100101101010101010011010100101011010011010011010101010100101010101100101010101010101010101101010010110100101011010
98	01100110011001010110011010101001010110010110101010101010100101011010101001100110011001011001011010100101011001100110010110010101
99	10011001010101010110011010010101011001101001011001101001101010100101100101010101101001100101101010011010100101101001011010100110 10

Regras de raio 4

	Regra
0	01011111001001111111001101110100001111101101110010010110110100000000100011110011101010100111110011011010101101001011110000010100001110000110000110 010010101111101101001010000011010110011101001101110001110010000111011101110010110110111111001001110100101100010110100000110110000001100100010111 10000010010001101101001001011111110111000011000101010110000011001001010100101101000011111010111100011110011110011011010100000100101101011110010 10011000101100100011100011011110001000100011010010010000001101100010110100111010
1	010100010111101000111100101100000101111010100100001000011101010100110011100011010011010101000100101011100011000100111011000101111111111110011110 0010000001111101101100101000110100110000110011101010101011100011101001000011111001110111100011011000110110111111110101110100001011100001101001111 1010000101011011110111100010101011001100011100101100101010111011010100011100111011000100111010000000000001100001110111111000001001001101011100101 10011110011000101010101000111000101101111000001100010000111001001110010010000000
2	111111001100001001010011101100000010010010110100110001111001011101010010001011110011000111111011001001011111110110000110101111110101001101001011 110110110011001101010111001100100100111111111010001100001110010101101000010110110010001111110010100010001101001000000011001111011010110001001111 11011011010010110011100001101000101011011101000011001110000000100110110100000000100111100101000000101011001011010000100100110011001010100011001101 101100000000010111001111000110101001011110100100110111000000011010111011100101101
3	01000010100101010010100011111000000010011101110111111100010100001010000000101001011101000000100111001110000100000111010101110010000000101110101110110 101111010110101110101100001001110010101010100010011000010000010011110101111001001111011000000000011100010001100010111101011010101101011100000111 111011000100010000001110101111010111111010110100010111110110001100011110111110001010100011011111101000101000100101000010100101000101001110110000 11010101010111011001111011111011000010100001101100001001111111111000111011100111
4	00001100000001000110010101011000000101000101011101001110100110000111100001100000101110110001010110010100111100100010011000000010001110000110101001 010001100110101110011011001010100010110011010101101110100000011001110001000100011111011011101110011010000011010111110011111101110011010101001111 11010111010100010110001011001111000011110011111010001001110101001101011000011011101100111111011100011110010101101010111001100101000110010011010101 110100110010101001000101111110011000111011101110000001001000100011001011111001010
5	101000000111010110100101001100000101100010101011100110110010010100110011101000110101001110010110001011000001110101101011000011101101010101110100 000011110010001110100011001110001100001001000101110110111010000001100001101001100011101000011101101110111010111101011111100010100101101011001111 101001110101010001100100110110101100110001011100101011000110100111010011111000010100101001111000100101010000101111110000110111000101110011000111 00111101101110100010010001011111100111100101100111000101111000100100010001010000
6	0011101011101101011111101010010000110011110010011101111110100111101000110000000001100101101100001111000001111100001100100011011110000010001101100 01100001111010111001110001111101111001100000000100100100011000000111101000001100111111101111101010000011000001010110001010001001010000001010110111 10011000011011000100000010110000101110011111111110011010010011110000111110000001111001101110010000111110111001001110011110000101000110001110000010 0001100111111101101101110011111100001011111001100000000100000101011111100111110101
7	0001111010110010000100100000000001010100100101001101010110011010010101100100010001001110011001100011000000010000011111111101100110010000000010001 100110101011101111111111101100000100101000111001100111000010000111101000001011100111010110101111110111011000000001110000101001101111011011111111 101010110110101100101010011001011010100110111011101100011001100111001111110111110000000000100110011011111110111001100101010001000000000100111111 011010111000110011000111101111000010111110100001100010100101000000010001001111111
8	1001000100101111011000010000000011111010100010011101001111101000100101011110011001011110110110011001110101000011000001101101000000011111010001110 1111110010110011110001010100010110010010101001010111100100100001010010111110100110110001010010010000010011111101011011101101000010011110111111111 0000010101110110001011000001011101101010000110011010000100100110011000101011110011111001001011111100000101110001000000110100110000111010101110 10 011011010101101010000110110111101011010000010110010011101011011011111101100000010
9	1110111110101001010001011110010001000100110001101001101101011110111100100110001001100101000101110010110011011100110010101011111100001001011010011 110001111111101111110010010111111011111011001100100010101000111111001111010101100110100010110010101101010101011100010000101011010111010000110111 011101100111001011001001010000100001101100111011001101011101000110100110010001100110101010000001111011010010110000111000000001000000110110100000 010000001001100110111010101110000001100001010100110010111010011010100101010101000
10	01100011100010111010010010010000101100101001110000001000011111111001101110000001110110111110011001100100010101011101001001111101111000110001101010 00001100101110111000100101001010110110001101101011011111110011111101111111010000110101100010100101110111111101101010011100011101000101101101101111 010011010110001111101111000000001100100011111000100100000110011001101110101010001011011000001000011100111001010111110011010001000111011010110101 001001110010010100100000000110000000100000001011110010100111010110100010000000100101
11	1100110110010101111111101001000010001100101100010101111001001010010101100010001101110111001001100001011010100110010000000000010011110001100110110 011000100011001010111011010001111101110010101011001111000110110100001010100001100010000001011100001100110111100000001100100110101000000000101101111 0111001101001110101000011011010110101001110111001000100011011001111010010101100110111111111110110000111001100100110011101110011010100010010111000 001000110101010011000011100100101111010101111001110111111010001111001100100001111
12	11010001011010000000101011101100011111111101001100111101000010000010001111110000011111000000100000111010110010011001011101100000000110111100101001 000110110100101010000100010100110011001001111101001011011000111010101110001110011110000101000111101101110110010100101110100101111110101000100111 000000000101100110000101111011111011100000111110000011111011111000101001101100110100010011111111100100001101011011100100101101010111101110101100 11001101100000101101001001110001010100011100011000011110101110000100100010011010
13	0111101101110101011000010001000011000100101010100100110001110101101101001100001101100000011110110000010101001111001101010101001101101010111010100 1111100100001010011101110101000000010100001111000110110110001000101101111110011100010110110000101000011001000111110000100100010101001111011101111 001110110101010110110011100010100100101100111100100111110000100111110101011000011001010101011001001010100010101100000110111101011000100010101111 11101011110000111001001001110111010010000011000111010010011110101111001101110000
14	0001010110101110000010100101000011100000010111011001000101010011011110000001100010111010001001001000011110111000011110011000111111101100110000101 100001100000001001011011111011010101000101001010100111100010100000110011011110010010100101111101110100010000001111101010010100011111010110101111 000111110100010011011101010110010000111111001110100010111011011011110000100011110000110011100000001001100111101001111001111111011010010000010010 101011101011010101100001110101111100110010000110110101101000000100010111011111100
15	10001000100110111111001010100100010011011111110110001001101101001000000000100010100110100011011111101100011110000000100001000100100010110011100011001 11001100110010100110100011101000011100110101001111001111010010101010010001000110111000101001111101001000000101000011101110110010000011010101101111 01100100000010011101100100101101111111110111010110010111001000000100111000011111101111011101101110100110001110011000110011001101011001011100010111 100011001010110000110000010110101010101101110111001000111010110000001011011111010111
16	001001100100011110111111100000000010001100111000001100010011000001010110101101111011001001100110010000000110110101110011001111110000110100101100110 001000111001101000110011111101000101110110110101001011001100100010110100111000010011011000100000101110111001010011011001101110000100000001111111 01110011000111110011101100111101010010100100001001101100110010110111111001001010001100110000000111100101101001100111011100011001011100110000001011 10100010010010101101001100110111010010110001111011001001110111110100010001101011
17	00111000010110101101010001000000101101101100010111100000010000101001110111111111000100110000100110100011101101000101000000011101101000010011111011 01011000100100100000110111110111001010011110010000111101101011110001000000001011010110110110000101110001000011111110001111010010100101011101111111

	01001001001110100001111101111010110001000000000111011001111011001011100010010111010111111110001001011110110000010010100111011011011111001000001000 110101100001101111000010010100001110111111101001010010010011110100011101111100000
18	10010100101000010010001011001000010001011101110010101110011111010101000010111001101100010010001010011100011011001111111110011010011101010100010 00111010110110100011111011100010101011111001000001001000000011111000100110101001010110110001100001111001111110110110101101011110110111010011011 1 10111010001000110101000110000001010101111010001100100111011011110101100011100100110000000011001011000101010111011100010100100101110000010001110 1 01010000001101111101101111111000001110110010101101010010011100111100001100000010 0
19	10011010111111000100100110010000011110101101100110001000001010101111100100011111100010100110011010000000111111101010011011001001001101000001111 1 11000101000110100100010111001110110000011100001110001010010011011000110100011110100111111111010110001000111000000110010100000011101101100110111 1 10000101001001100111011111010101000001101110000000111010110011001011111110000000101011001001101101100101111100000001110101110010110111010001100 01 0011111100011110001110101101100100111001011100001011000000000101001110111000111 11
20	01100110011000001010000011110000001001000110001111010110101001001100110011011100100101111010110010101000100001100011101010110001010011000010000 01 10101101010100100110001111011011111001010000011111111011111101111000110000101100100010110000000111000100010100110010011001100111101011111000011 11 10110111001110000101001010110110011001100100011011010000101001101010111011110011100010101001110101100111101111100101001010101101100111000010010 0 0001101011111000000000100000100001110011110100110111010011111100011101110101100 11
21	11101101000111011111111011111000001000111101001011100011010111010100101011011011101101111110111101101110111100110111011011000101010000010110111 00 01100010010010100101100011010010101000110011100000101010110010000000010001110011011010010111011011101101111000111000100101110001000000000100000 111 11011100001011010001110010100010101101010010010001001000001000010010001000011001000100100111010101111101001000111001110110110101101001110010110 1 0101110011000111110101010011011111111011100011001001011010001001000100100001110 00
22	11000011100001101001010100110000000101001101001101000101101011100011111101111011101100100010100000011000000110010000010000010100011011110101001 01 00010001000010101010101111000101100011110110100101101001000000000101011100101000000001110000111110100011001101011001111000111100101101010110011 11 11101011001011001011101001010001110000001000010001001101110101111110011111100110111110111101011100100010101101011101110111101010101010000111010 0111000010010110100101101111111101010001101011111111000111100000101110011001010 0
23	01111111110110110110001101110000000101100111011110011000111111010101111000011100100100110100101110101100011111001100101110011110111100001011110 10 01111011010000101001011110001001111011011001111101011000001000100011011110100111000011011111001010110111010100001000000000100100100111001000111 11 11010011000100001100111000000101010000111100011011011001011010001010011100000110011010001100001000011110100001011000010010111101011010000111011 0 0001001001100000101001111101110111001000010110001111001000001101010010001010111 1
24	01110001011001100000100001101000010100111110100000010011101100011100110110011110101101100010000011010100000000011010011101011001111011111011001 10 11000100110010101010010010011100111010111101111010001011110000000001100111001100011000001001010001110110010010100100011101001100111110111100101 11 10101100000101111110110001001110001100100110000101001001110111111001010111111111001011000101001100001000000100110010011101100110101010110110110 0011 000101000010000101110100001111111110011000110011100111110110101110001001101101011
25	01100110101010100101100101101010101010010110100110101001100101010110101010100110100110101001011001011001101001101001100101010101010110010101101 0 10100101101010011001100110010110101010100101101001101001100110011010011001011001101010100101010101100110010101011010100101010110100101010110100 1 01100101100110011010101010011010011001011001100101010101101001100110100101101010011001011010011010100101011010100101100101010110101001101010011 0 1010010110011010011010011010101010100101100101101010011001011001101001010110011 0
26	01100110010101100110101010011001010110101010010110011010010101010110011010101001100110010110010101011001010101101010011001100110010110100101101 0 10010101101001100101101001100110011001010110010110011001101010010101101001010110010110101001101001010110011010101010101010101010010110101010100 1 01011001010101011010101010011010011010010110011001010110100110010110100101010110100101101010011001100110011001101010010101101010011001011001010 1 01101010101001011010100110101010010101101001101001010110100110100110101010101010
27	10101010101001011010010101011001011001100101101010011010010101010101100101100101100110100110010110100101011010101001011001101010011001100101100 1 01011001101010100101101001010110101010101001101001011001011001101010101010100110100101010110100110011010101010100110011001011010011001011001101 0 10100110100110100101101001011010011001100110101001011010010110010110010110101010101010101001100101011010010101011010100101100110011010011001010 1 0110011010011010010110010101101010101010010110011001010110010101101001100101100 1
28	01100101010110011001011001100110010110011001010110101010100101010101011001011010100110101001101010101010010101100110010101011001100101010101011 0 01100101100110101001100101010101100101101010010110101001010101100101010101101010011001100110101001011001010101010110011010100110011010100110100 1 10011010101001100110100110011010100110011010010101011001011010100101100110011001100110010101100101101001010101100101010101100101101010100110011 0 1010100101100101101010100110100101010101010101011010100101011001010110100101 01
29	01010101101001010101010110010101101001011001100110011010010101010101100110010101100110011010100110010110101001100101101001010101101010100101010 1 10100101010101100110101001101001010110011001101001011001100101101010100101100101011001011010010101010101100101011010100101010110100110011001011 0 01100101011010010110100110011010100101101001101001011001100110010101100110100101101001100110011010011010100110010101010101101001011010100101011 0 0101011001010110101010100110100110101001101010010110100110010101010110100110011 0
30	10011001010101010110101001100110100110010110011001011010010101010110011010010101100110011001101010010110100110100101100101100110010110100101101 0 10011001010110100110011001011010100101100110100101011001101001010101011010101001100110100110011010010101011010100110100110011001101010100110010 1 01010101101010100101100101011010100110010101101001011010100101011010010101011001011001010110011010100110100110101001100101010101011010010101011 0 1001100101101001010110101001100101010110101001101001101010011010100110011010101 0
31	01011010100110011010100110100110011010101010011001100101100101010110100101101010100101100101101010011010101010100110010110101010011001010110100 1 01010101010101011010010110011010011010010101011010101001010101011010101010010101011010010110010101101001101010100101010101100101011010011010010 1 01101001010101101010100110011010100101101010010101101010101001101010100101101001010101010101100101100101100101011010010101010101101010100110010 1 0101011010010110101010101010011010101010011001100101010101101001010101011001100 1
32	01010110101010011001101001011001010101100101100101010101010101010110011001100101100101100110010110100110011001101001011010011001011001100110100 1 01100101011001010110010110101001011010010110100101011010010101010101100101010101101010101010101010011010010110100101100101010101010110010101 10 10010110100110011001101010011010101010101010101001101001010101011001011001100101011010100101101001011010100101010110010101011010101001100101010 1 0110011010101001011010100110011010011001101010101010011010100110100101010101010 1
33	10010110010101100101100110101010011010010101011001010101010101011010101010011001100101011001011010100110010110101010100110010101100101100110011 0 10101001011010010110011010101001101001101001011010010110101001100110010101011010010101010110100110100110010101010101101010101001100101011010100 1 10101010101001011001101001011010101001010110011001100101011001101001011001011001100110010110011001101010100101100101100101010110011001011001101 0 1010100110010110100110100101011001100101100101100101010101100101101010101010011 0
34	10101001101010100110011001011001100101011001101010100101100101011001010110010110100101011010011001011010011010011001101010100110101010010110010 1 10100101011001011010010110011001010110011010100101100110100101101001101010010110100101100110011001101010101010010101011001011001101001100101101 0 10100101011010101010101010011010101010100101100101100110101010101001101010101001101001011010010110010101100110011001010110101010101001011010101 0 0110011001101001011010011001010110011010010110011001101001100110011001101010
35	01101001010110101001010110011010100110010110010110010110010101011001101001011001100101101010010101010110010101011010101010101010010110100110101 0 01010101011010100110100110101010100101101001011010010110010101100110011001101010011001011001011010101010011010101001010110100101011010011001100 1 01010101101001011001101010011010100101011001011001100101100110011010011010010101101001101001100110100110101010101010010110101010101001101010101 0 100110010101011010011001101001010110011001011001101010011010101010100110100110 01
36	10100101101001101001011001100110101010101010100110010110010101011001010110100101100110100101011001100110101010010110010110011001011001100110100 1 01011001010110100110101001101010010110010110100101010110011010011001100101101001011001010101010101010110010110101010010101101001011010011010011 01 01101001010110010101101001011001100110101001101001100101011010101010011010100101100110011001101001011010101001010110100110100110010101011001100 1 1001010101101001010110010101010110011010100101010110100101101001101010100101010 1
37	10100110010101100110100110010101010101100110011010100110100101011010101010101010100110010110100101101010100110010101011001010101011001010110101 0 10100101010101101010101001010101011001010101011010100110011010010110010110100101100110101001101001010101010101011010011010101001011001100101101 0 01010110100110100110010110011001100101010110010101100110010110100101101001011001011010101010011001011001101001101001010110101001100110011010100 1 0101101010010110101010010101011001100101011010101001101001101010011010010110011 0
38	01101010100110100110101001100110011010010101011001010110010101011010010101100101100110011001100101100101101001010110101001100110100110100110010 1 10100101010110100110100101010101101010100110100101010101100110011001011010101001011010011001100110011001011010011010011001100110101001100110010 1 10101010100101100101010110011001011010100110101001100110010101010101011001100101010110101010010101101001101001101001010110010101100110100110101 0 1010010110101001010110011010011010010101011001100101010110100101100101011010101 0
39	01010110011001101001100110101001010101011001100101100110010101011010100101010101100110101001101001011001010110011001100101101001101010100101010 1 01011010010101100110101001100101011001011001011001010101101010100110101001010101101010100101101001010110101010101010011010010110100101010110011 0 10010101011010010101010101011001011001101001101010101001101001100110011001010110011001011001100110010110101010010101100110010101010110100101101 0 0110101010010110010110011001011001101010101001101010011001010110010101010101101 0
40	10010101100101011001011010011010101010100101100110011001011001010110010110100110101010101010011010010101100101101001101001100101100101010101010101 10 01010110011001011010011001010110100110101010100110100101011010101001010101010110010101011001010110010110010101100110100101101010010101101001011 0 10100101101001011010010110011001011010011001010110101001100101010110101001011010011001100101101010100101101010100110010110011001100110010110010 1 0110010101101001101010010101100110010110101010100110010110010101010110011001010 1
41	01011001011010010110010101101010100110101010101001010101010101011001010110100101101010010101010110100101011001011010010101010110011001101001100 1 10100110101010011010010110010110100101101001011001010101010110100110100110011010100101101001011010101010100101010110100110010110011010010110100 1 01011001010110101001011010011001010110100101101010101010011001010110011010100110100110100110011001011001101001011010010110010101100110100110010 1 1010100101010110010110100110100101011001010101011001101010011010100101100110010 1

42	0101101010010101011001101001100110100110011001010110010101010101100110100110100110100101011001101010100101010101100101010110011001011010100110 10 1010101010100101011001011010010101011001011010011001010110101001101001100101011001100101010101011010100110100110010101011001011010101010011010 0110011010010101100101100101100101010101101001101010101001101010010110011001011010101001011001010101101010100110100110011010101001010110010101 01 0101011001101001100110100110100110100101010110011010100101011001010101101010101 0
43	1001011001100101100110010101011001011001010110011010010110010101011001010110011010100110101001011001100110101001011010101010100110011001100101 01 0110011010100101101001101001010110010110010101100110010110101001010110100110100101101010101010101010010110100110010110101001100110011001010110 01 0101101010011001101001101001100101011010101010011010100101011001100110011010101010010110010101010110010110010101010101011010101010010101100101 10 1001100110010110011001101001101001010110100101010110101010010110101010101010011010
44	1001011010011001101010101010010101100101100101101001011001010101011010101001100110100110100101101001010110010101010110011001100110101010100101 10 0101011010101010100101011001100101101001011010011001010101010101101010101010100110011001011010011010010110010110010110100110010101010110101010 10 1010010101100110010101101001100101100101011001101010010110101001101001010101011010100101010110101001010110011001011001011010011010010101101001 10 0101011010101001100101101010101010101010011001101001010110010101101010010101 01
45	0110100101101001011010010101010110011001010101011010101001010101011001011001010110100101011010101001100110100110011001011001011010010110100101 10 1001101010010110010110011010101001100110100101100110011001010110010101011010010101101010011001100110100101011001100101101001010101100101010101 10 1001100110100110011001100101100101101001010110101010011010011010101010010110011001010101011010101010101010011010101010011010101001010110100101 01 0110011010010110010101100110100101010101011010100101011001010110011001011010 0101
46	1010100110100110010101101010011010101010101010011010101010010101010110010101101010100110010110011010100101011010011001101010011001100101100110 01 1010011010011010100110100110101010100110101010011010011010100110101010010110100110101001101001101001010110011001100101010101100110100110101001 01 0110100101011001011001011001100101100110100101011010011001011001100110010101101001011010100101010101100110011001101001011001010101010101100101 01 1010100101101001101001100101100110101001101001101010011010011010101001010110101 0
47	1010010101011010100101010110100110010110011001100101101001010101010101100110010110100110010110100110010101100110100110011010100101011010101010 10 0110101010011010100110100101100101011001100101010101011010010101010101101001011001010101011001010101011001100101011010011010100110011001011001 10 0101011001010110010101011001100110011001100110101010010101100110100101011010101001101001100101100110100110010110010101010101010110011001101010 01 0110011001010101010101011010100101010110101010100110010101101001011010011001101 0
48	0110101010101010101001101001101001101001010110100110101001010101010110011010010110101001101010100101101001010110101010101001100110011010101001 01 0101101010010110010110010110100110010110011010100110011010010101101010101001011010010110011010100101101010100110101010010101011001101010100110 01 0110101010011010011001010101100110010110011010101001010110010110101010011010011010010110101010100110011010010101010110010101100101011010011010 10 1001100101101010010101011010011010101010010110011001101001101010011001100101010 1
49	0110101001010110011010010110100101010101100101010110100110010101011001100101101010101001100101010101011001011010100110100101011010100101101001 10 1001011010101001100110100110100101011001010101011010011001100101010101010101101001100101101010011001101001011010101001101010101001101001011010 01 1010010101100101101001011001100110010110010110010110101001011010100110011010101001101010100110010101100110101010010101010101011010010110101 0 1001011010010101101001010101011001011001100101011010100110100101100101100110010 1
50	0010110011010100100101101110100111111011110100000011111001000100010111111011110100011000100111101001101110111000011100111100101110000001010101 01 0001000011111010110000101010011000000110011101110100010100000101010100100111011101001000110101010001111110001000110100110010101101101001000101 10 0000001000010111111000001101110111010000000100001011100111011000010110010001000111100011000011010001111010101010111011110000010100111101010110 01 1111100110001000101110101111010101011011000100010110111001010101110000001110111
51	0100010001010101111100010001101010010111000011000011110110101101001110001100110101001110010000011000111010000111111010000011010111111001011100 01 0111110100001011111111000001100011101101010000000101110101101111111101010011110011000100110011000100010100001100010111011101010100000111011100 101 0110100011110011110000100101001011000111001100101011000110111110011100010111100000001011111001010000001101000111010000010111101000000000111110 0111 0001001010111111010001010010000000010101100001100111011001100111011101011110 0111
52	0000010111100101111110010010110000110100100101010011100110100000010000110111000110111100011001000110011101110000110000111011001110111101001100 100 0110010100111000010010111001111111110011101101011100010000111000000101101101110001111000111101010010010010100101011111010000110100000011011010 011 1100101101101010110001100101111101111001000111001000011100110111001100010001111001111000100110001000010110011011100110101100011110110100011000 00 0000110001001010001110111100011111010010010001110000111000010101101101101011 0101
53	0110001010101101011001101000010110100010000111011010111010010001001100011100101110101101111010010101000110001001100010010100111111101100100011 00 1110001000110000100111001110111111100110111111010101101111001000101010101101000100000111100000010101011100000010111001110101010010100110010111 1010 0101110111100010010100010110111011001110001101000101001000010110101011100111011001110110101100000000100110111001100011101110011110110001100010 000 0001100100000010101001000011011101010101001011101111100001111101010100011111 0100
54	1111110110111011110100010010110110010011011010000001011010100111011111010010100011110100111001001011011010101110010000011100100110010001100011 010 1111110100110001101010001001011010101100000000011001100001001010100101011011000101110011010001011000011110110100000000010010001000010111011010 010 0110110010010111110100101011000100000101101011100000101100011011010010010101000110111111000110110011011100111001010000000101100111001010111011 01001 0101001111111110011001111011010101101010010011101000110010111010011110000100 1011
55	1110010011010010000000000110011000111101000011000011011010010110110001111001100001000000011111101000010110000001101010010111110101101001001011 10011 1001110001001011110010010110111111010101001010011111000100100011111101001101101000010011000001100110111111010111000110110010110111111111001100 11 1000010111100111100100101101001001110000110011110111111000000101111010011111001010110100000010100101101101000110001100011101101000011011010010 000 0010101011010110000011101101110000000101100100101111011001111100110010000000 101000
56	1110000100001000000000100010010010110001011011011101001010111010101000100001111101000101110000101110001000011010111101010111101010111111000101 0 1001101000110010000010110111101110101010110111111100010011011001111101101011101011010000010111001101011101000000000111101111011111111101110110 11 0100111010010010001011010100010101011101111000000101110100011110100011101111001010000101010000001010100000011101010110010111001101111101001000 0100 0101010100100000000111011001001100000100101000101001011111010001100101000101 11111
57	0111111101100001000000011111111101000011101010001011111011011100001011101010001100011010011110111001011100011100000010001010110111101000011101 010 0000110111010010011110000001001000110111010110001011010001011100001000101011000111010001000010111011101110100011010000000100111101111111000000 000 1011110001010111010000001001000111101000101011100111001011000010001101000111000111110111010100100001011110001010111100100010110110000111110110 11 1001000101001110100101110100011110111010100111000101110111101000100010001011 1001
58	0100101110010010110110000000111000011011001001001101101101010110110010011011010011101000000101011000000001101100000111011110000111111001011011 0001 1111010010110000101001001101000001000010001111010100100111111001111011100000100110010111010111111011110011110110010110100011011010010011111110 001 1110010011011011001001001010100100110110010010110001011111010100111111100100111110001000011110000000110100100111000000101101001111010110110010 1111 1011110111000010101101100000110000100011111011001101000101000000100001100001 0011
59	0110010011101111110000010110011100100100000101001111111011000011111011000110101110111111001101010011000000001100010101001000000111111110100101 101100 0001111011111000101100001101011100100100011010101010101010110100011000100110100100111101100111100110000010011000101001101100010000011111010011 0001 1011011111010110000000100111100000010011100101000100000110010101100111111110011101010110111111000000000101101001001111100001000001110100111100 101000 1101101110010101010101010010111001110110010110110000100110000110011111011001 1101
60	1101001100101011011010010001011000000010000101111110000011011101110100000001000010111001110110000101100100010001111000110000110100011111010101 010 1110111110000010100111101010110011111100110001000101110101111101010101011011000100010110111001010101110000001110111001011001101010010010110111 01001 1111110111101000000111111001000100010111111011110100011000100111101001101110111000011100111100101110000001010101010001000011111010110000101010 0110 0000011001110111010001010000001010101001001110111010010001101010100011111100 01000
61	1011101110101010000011101110010101101000111100111100001001010010110001110011001010110001101111100111000101111000000101111100101000000110100011 10 1000001011110100000000111110011100010010101111110100010100100000000101011000011001110110011001110111010111100111010001000101010111110001000110 10 1001011100001100001111011010110100111000110011010100111001000001100011101000011111101000001101011111100101110001011111010000101111111100000110 00 1110110101000000101110101101111111101010011110011000100110011000100010100001 1000
62	1111101000011010000000110110100111100101101101010110001100101111101111001000111001000011100110111001100010001111001111000100110001000010110011 011 1001101011000111101101000110000000001100010010100011101111000111110100100100011100001110000101011011011010110101000000101111001011111100100101 100 0011010010010101001110011010000010000110111000110111100011001000110011101110000110000111011001110111101001100100011001010011100001001011100111 11 1111100111011010111000100001110000001011011011100011110001111010100100100101 001010
63	1001110101010010100110010111101001011101111000100101000101101110110011100011010001010010000101101010111001110110011101101010110000000100110111 0011 0001110111001111011000110001000000011001000000101010010000110111010101010010111011111000011111010101000111110100011000101010110101100110100001 01 1010001000011101101011101001000100110001110010111010110111101001010100011000100110001001010011111110110010001100111000100011000010011100111011 11 1110011011111101010110111100100010101010110100010000011110000001010101110000 01011
64	0000001001000100001011101101001001101100100101111101001010110001000000101101011100001011000110110100100101010001101111100011011001101110011100 101 0000001011001110010101110110100101010011111111100110011110110101011010100100111010001100101110100111100001001011111111011011101111010001001011 01 1001001101101000000101101010011101111101001010001111010011100100101101101010111001000001110010011001000110001101011111101001100011010100010010 110 1010110000000000110011000010010101001010110110001011100110100010110000111101 10100
65	0001101100101101111111110011001110000101111001111001001011010010011100001100111101111110000000101111010011110010101101000000101001011011010001 100 0110001110110100001101101001000000101010110101100000111011011100000001011001001011110110011111001100100000001010001110010011010010000000000110 01100 0111101000011000011011010010110110001111001100001000000011111101000010110000001101010010111110101101001001011100111001110001001011110010010110 1111 1101010100101001111100010010001111110100110110100001001100000110011011111101 0111
66	0001111011110111111111101110110110100111010010010001011010100010101011101111000000101110100011110100011101111001010000101010000001010100000011 10101 0110010111001101111110100100001000101010100100000001110110010011000001001010001010010111110100011001010001011111111100001000010000000001000100 100 1011000101101101110100101011101010100010000111110100010111000010111000100001101011110101011110101011111100010101001101000110010000010110111101 1

	1010101011011111110001001101100111110110101110101101000001011100110101110100000
67	1000000010011110111111100000000010111100010101110100000100100011110100010101110011100101100001000110100011100011111011101010010000101111000101 01 1111001000101101100001111101101110010001010011101001011101000111101110101001110001011101111010001000100010111001011111110110000100000000111111111 0100001110101000101111101101110000101110101000110001101001111011100101110001110000010001010110111101000011101010000011011101001001111000001 00100 0110111010110001011010001011100001000101011000111010001000010111011101110100 0110
68	1011010001101101001001111111000111100100110110110010010010101001001101100100101100010111110101001111111001001111100010000111100000001101001001110 0000101101001111010110110010111110111101110000101011011000001100001000111110110011010001010000000100001100001001101001011100100101101100000001110 0001101100100100110110110101011011001001101101001110100000101011000000011011000001110111100001111110010110110001111101001011000010100100110100 00 0100001000111101010010011111001111011100000100110010111010111111011110011110 1100
69	1001101100010000001111101001100011011011111010110000000100111100000010011100101000100000110010101100111111100111010101101111100000000101101001 0011 1110000100000111010011110010100011011011100101010101010100101110011101100101101100001001100001100111110110011101011001001110111110000010110 01110 0100100000101001111111011000011111011000110101110111110011010100110000000110001010100100000111111110100101101100000111101111100010110000110101 11 0010010001101010101010101101000110001001101001001111011001111001100000010011 00010
70	0010110011010100100101101110100111111101111010000001111100100010001011111101111010001100010011110100110111011100001110011110010111000000101010101 0001000011111010110000101010011000000110011101110100010100000101010100100111011101001000110101010001111110001000110100110010101101101001000101 10 0000001000010111111000001101110111010000000100001011100111011000010110010001000111100011000011010001111101010101011101111000000101001111010101 1001 11111001100010001011101011111010101011011000100010110111001010101110000000 1110111
71	0100010001010101111100010001101010010111000011000011110110101101001110001100110101001110010000011000111010000111111010000011010111111001011 10001 0111110100001011111111000001100011101101010000000101110101101111111101010011110011000100110011000100010100001100010111011101010100000111011 100101 0110100011110011110000100101001011000111001100101011000110111110011100010111100000001011111001010000001101000111010000010111101000000000111110 0111 0001001010111111010001010010000000001010110000110011101100110011101110101111 00111
72	0000010111100101111110010010110000110100100101010011100110100000100001101110001101111000110010001100111011100001100001110110011101111010011 00100 0110010100111000010010111001111111110011101101011100010000111000001011011011100011110001111010100100100101001010111110100001101000000110110 10011 1100101101101010110001100101111101111001000111001000011100110111001100010001111001111000100110001000010110011011100110101100011110110100011 00000 00001100010010100011101111000111110100100100011100001110000101011011011010 110101
73	0110001010101101011001101000010110100010000111011010111010010001001100011100101110101101111010010101000110001001100010010100111111101100100 01100 1110001000110000100111001110111111110011011111110101011011110010001010101011010001000001111000001010101110000010111001110101010010100110010111 1010 0101110111100010010100010110111011001110001101000101001000001011010101110011101100111011010110000000100110111001100011101110011110110001100 010000 000110010000000101010010000110111010101010010111011111100001111101010100011 110100
74	1111110110111011110100010010110110010011011010000010110101001110111110100101000111101001110010010110110101011100100000111001001100100011000 11010 1111110100110001101010001001011010101100000000011001100001001010100101011011000101110011010001011000011110110100000000100100010000101110110 10010 0110110010010111110100101011000100000010110101110000101100011011010010010101000110111110001101100110111001110010100000010110011100101011101 101001 01010011111111100110011110110101011010100100111010001100101110100111100001 001011
75	0101100110100101101001100110010110010110011010011010100110010110101010101010011010101001100101010101011010101010010110010101100101011001101 01010 1010011010101001100101011010010101100101101010100110010110100110101001101010010101011010100101101010100101100110101001010101011001100110011 00110 0101011001010101101010101001100110100101010110011001100101101001010101010110100101101010011010100110010101100110010101010110011001100110010 11001 0110101001101010011001011001010110100110011001100101011010101010100101011 0010101
76	0110010101100101011001100110011010101010010101100101011010011001100101100110101001010101101001010101101010100110100110011010011001011010100 10101 1010010110100110011001011010100101100101010101101001010110101001100101010110101010101001100101010101101001100110101010101001011001101010010 10110 0110101010100110010101011001101010101010101001010101011010010101101010011010011001100101010101011001101010011001101001101010101010101001100 11001 011010101001011010010101100101101001011010010101100101011001100101010110 10010101
77	0101010101100110101010010110011010101010100101100101100110100101010110100110010110010110011001100101101010010110100110010101010110010101011 01001 1010100101011010011010101001010110100101100101011010010110101001101010010101011010010101011010100110100101101010011010101001100101101001011 00101 0110100101100110010110101001010101100101100110101001011010101010101010100101101010011010011001101010010101100101010110101001010101011001101 00110 10011010100101011010101001010110101010011001100101100101100101100110010110 011001
78	0110100101011001100110011010011001101001011010011001010101100110100110010101100101010110101001101001100110101001100101100101011001011010010 10110 1010010110011010100110011010011010101001100101100110011001010110100101011001011010010101100101100110010110101010101010011010010110010101101 00101 1010100101011001010110100101010110010110101001011010100110101010101010010110100110101010101001100110011010011010101001011001010110011001100 11001 1010011001010110010101010110101010010101010110011001100110101001010101011 0011010
79	1010101010100110100110101001101010100110010101100101100110100110100101100101101001101001100101010101100110100110011001011010100110101010100 11001 0110011001010110101010011001011010100101100101100110100110100110011001101010010110010101010110101001011001011010010110010101101001010110100 11001 1010101010100110010110100101011010011001100101011001011001101001100110011010010101010101010110100101101001010101100101100110011001011001100 110 1001101001010110011010100101101001100101011001101001010101101010100110100 1100101
80	1010100101100101101001100101100101010101010101011010010110100101011010101001100101010110100101010110100110100110010110011010011010010101101 01010 0101101001010101100101010101011010101010100110010101011001101001010101011010011001100101100110101010100110011010011001011001011001101010010 11010 1001011010100101011001011001101010100101100101100110100110101010101001100110011001011001100101101010101001010110010110010101101010101010011 00101 1010011010011001010101100101101001010101011010010110100110101010101001100 1101010
81	1010100101010110010101011001010101010101010110010101100101100101100110100101011001101001101001101010011001011001100110101001100110011001010 11001 0101011010101010011001010110011010100101010110011010100101011001010101101001100110101010011001100110101010100110011001101010101010010101100 11001 1001011010011001010110100101100101010101100110100110101010011010100110011001100110100110101010101010010101100101101001101001011010101010011 01001 10011010100110011010011001010101011001101010010110100110011010100110010101 010101
82	0110101010101010011010010101011001010101010101101010101010101010011001010101101010011001100101011001101010101001101001101010010101011001101 01001 1001100101011010010101101001100101101010100110101001011001101010010101101010010101010110010101100110011010011010101001100101010110101001100 11001 0101010110100110101001100101100101101010100101010101100101100101011010011010100110011010010101100110100110010101100110101001010101100101011 00110 0110100101011010100110010101100101010110011010100110101001101010011001010110 1001
83	0110010110011010100101100101100110100110100101010101010110101001010101101001101001011001011001011010011010011010011001100110100110100101100 10110 1001101001100101101010011001010101011001100110100101010101010110100110100101010101101010011010101001010101101010101010010110011010011010010 10110 1010101001100101100110100101010110011001011001011010011001010101011001010101100101011010101001100110010110010110101010100101101010100110101 00101 0101011001011010010110011010100110011010101010100110101010010110101010011 0100101
84	0110100101100101101010011010101010010101010110011010010110101001011001011001010101011001100101101010101010100110100101010110101010100110100 10101 1010011001101010011010101010010110100110011001011010010101010101011010010101100110011001011001010101011010101010101010011001011001101001101 00101 0101011010101001101010101001010110011010010101011010011001101010010110010110010101101001100110011001100110011010011001010110100101011001011 0 100110010110010110101010011010010110101010010110100101010101100101101001010 11001
85	1010011001011010010110011001101001101001100101100101011001101001010101010101100101010110011010101010100101010101101001101010011010100110010 10101 0101100101010110011010100101101010011010010101011001101001011001010110010101101010100101011010010101011010011001010110101010100110011001100 11001 1010100110101010010101010110011001011010101001100110011010010110101010101001011010010101100101011001101010011001101010101001100110011001101 00110 1001010110010101100110100110101001011001100110011010100101010101011010100 1101010
86	1001101010011010100110011001100101010101101010011010100101100110011010011001010110101010010110101010010101011001011001100101100110100101011 01010 0101101001011001100110100101011010011010101010010110101001010110011010101001010101010110011010101010010110011001010101010110100110010101101 01001 1001010101011001101010100110010101010101010110101010100101101010010101100101100110011010101010100110010101100110010110010101010101010110011 00110 1001010101101001011010100110100101101001011010100110101001100110101010010 1101010
87	1010101010011001010101101001100101010101011010011010011001011010101001011001101001101001100110011010010101101001011001101010101001101010100 10110 0101011010100101100101010110101001011010011010100101101001010110010101101010100101101010100101011001011010010101100101010110011010010110100 11010 1001011010011001101001010110101010011010011001010110100101010101010101011010010101100101100110010101101010011010101001010110101010100110010 11001 0110010101101010010101011010100101010110011001101001101001101001100110100 1100110
88	1001011010100110011001100101100110010110100101100110101010011001011001101010011010101001010110010110011001010110011010011010100110100101101 01001 0101101001100101011001100101100101010110011010011001100110101001011010100110100101101010011010011001101001010101010101100101101001101010010 11010 0101011010100110101001011010101001101001010110100101011001010101010101101001011001010101010110011001100101100101010110100110101001100110011 00110 01011001101010011010101010010101011010101010011001100110010101101010101010 0110010 1
89	0101010101011001011001010110010101011001101010011010011001011001011010011010010110010110011010101010011001011001100110100101011001010101011 00110 1001100110101001010101100110100101011010011010011001011001011001100110010101101001101010100101011010011010010110100110101001011010100101100 110 0101010101011001101001011010100101100110011010100110100110010110011001100101101010101010101010010110100101101010100110100110011001101001100 11001 0110010110101001100101011010010110011010100110010110101010010101011001011 0011010
90	0101011010011010010110011010011010101010101010100101101001011010100101010110011010101001011010101001011001011001101001100101100101101010010 10101 1010010110101010011010101010100101010101011001101010100110010110101010100101100110011010011001010101011001100101100110100110100110010101101 00101 0110100101011010100110100110010101011010011010011001011001010101010110011001100110100110011010010101010110101001101001101010010101010110011 010 0101100101100110101010011010010110101010100101101001011001010101010110011 0010101

91	010101101010100110101010011010101010101010100110101001101001101001100101101010011001011001011001010110011010011001100101011001100110011010100110 101010010101010110011010100110010101101010100110010101101010011010101001011001100101010110011001100101010101100110011001010101010110101001100110 011010010110011010100101101001101010101001100101100101010110010101100110011001100101100101010101010110101001101001011001011010010101010110010110 01100101011001100101100110101010100110010101101001011001100101011001101010101010
92	100101010101010110010110101010011010101010101001010101010101010110011010101001010110011001101010011001010101011001011001010110101010011001010110 011001101010010110101001011001101001010101100101011010011001010110101001010110101010100110101001100110010110010101011001101010100101011001100110 101010100101100101011001101001101001010101101010101001101001101010010110010101100110010110101001100101100110101001100101011010101001101010011001 10010110101001010110011010100110101010011001010110010101100101011001101010010110
93	100110100110010101101001101001100101100101101010101010100101011010101001011001011010011010011010010110010110010110011001100101100101101001101001 011001011001101001010110011010101010011001100101101010101010100101100101101010101001010110010101011010101001010101010110100110010110010110101001 010101011001101001100101101010100110011010011010010110011010101010011010101001101010010101011001100110100110100101010101101001010101100101011010 10101001101001011010011001010110011001010101010110010101011010010101011001011010
94	100101101001101001010110010101010110101010100110010110100101011010011010011010101010011001101001010101010101100101101010100101010101100101101010 010110011001010110010101010110100101100110011010010110101010101010010110101001100110011010011010101010010101010101010110011010011001011001011010 10101001010101100101010101101010011001011010101001011001100101011010011010011010100101100110011001100110011001011001101010010110101001101001 01100110100110100101010110010110100101010110100101101010101001101001011010100110
95	01011001101001011010011001100101100101100110100110101001100101101010101010101001101010100110010101010101101010101001011001010110010101100110101010 101001101010100110010101101001010110010110101010011001011010011010100110101001010101101010010110101010010110011010100101010101100110011001100110 010101100101010110101010100110011010010101011001100110010110100101010101011010010110101001101010011001010110011001010101011001100110011001011001 0110101001101010011001011001010110100110011001100101011010101010101001010110010101
96	011001010110010101100110011001101010101001010110010101101001100110010110011010100101010110100101010110101010011010011001101001100101101010010101 101001011010011001100101101010010110010101010110100101011010100110010101011010101010100110010101010110100110011010101010100101100110101001010110 011010101010011001010101100110101010101010100101010101101001010110101001101001100110010101010101100110101001100110100110101010101010100110011001 01101010100101101001010110010110100101101001010110010101100110010101011010010101
97	010101010110011010101001011001101010101010010110010110011010010101011010011001011001011001100110010110101001011010011001010101011001010101101001 101010010101101001101010100101011010010110010101101001011010100110101001010101101001010101101010011010010110101001101010100110010110100101100101 011010010110011001011010100101010110010110011010100101101010101010101010010110101001101001100110101001010110010101011010100101010101100110100110 10011010100101011010101001010110101010011001100101100101100101100110010110011001
98	011010010101100110011001101001100110100101101001100101010110011010011001010110010101011010100110100110011010100110010110010101100101101001010110 101001011001101010011001101001101010100110010110011001100101011010010101100101101001010110010110011001011010101010101001101001011001010110100101 101010010101100101011010010101011001011010100101101010011010101010101001011010011010101010100110011001101001101010100101100101011001100110011001 10100110010101100101010101101010100101010101100110011001101010010101010110011010
99	101010101010011010011010100110101010011001010110010110011010011010010110010110100110100110010101010110011010011001100101101010011010101010011001 011001100101011010101001100101101010010110010110011010011010011001100110101001011001010101011010100101100101101001011001010110100101011010011001 1010101010100110010110100101011010011001100101011001011001101001100110011010010101010101010110100101101001010101100101100110011001011001100110 10011010010101100110101001011010011001010110011010010101011010101001101001100101

Apêndice H Amostra de núcleos para o modelo com rotação na borda e no núcleo

Núcleos de raio 1

Todos os núcleos.

Núcleos de raio 2

	Núcleo		Núcleo		Núcleo		Núcleo
0	0101111100100111#0	25	0010000011111011#0	50	0101111100100111#1	75	0010000011111011#1
1	1110011011101000#0	26	0110010100011010#0	51	1110011011101000#1	76	0110010100011010#1
2	0111110110111001#0	27	0110000110011101#0	52	0111110110111001#1	77	0110000110011101#1
3	0010110110100000#0	28	0101010111000111#0	53	0010110110100000#1	78	0101010111000111#1
4	0010001111001110#0	29	0100100001111100#0	54	0010001111001110#1	79	0100100001111100#1
5	1010100111110011#0	30	1110111100011011#0	55	1010100111110011#1	80	1110111100011011#1
6	0110101011010010#0	31	0001101101111111#0	56	0110101011010010#1	81	0001101101111111#1
7	1111000001010000#0	32	1111110011000010#0	57	1111000001010000#1	82	1111110011000010#1
8	1110000110000110#0	33	0101001110110000#0	58	1110000110000110#1	83	0101001110110000#1
9	0100101011111011#0	34	0010010010110100#0	59	0100101011111011#1	84	0010010010110100#1
10	0100101000001101#0	35	1100011110010111#0	60	0100101000001101#1	85	1100011110010111#1
11	0110011101001101#0	36	0101001000101111#0	61	0110011101001101#1	86	0101001000101111#1
12	1100011100100001#0	37	0011000111111011#0	62	1100011100100001#1	87	0011000111111011#1
13	1101110111001011#0	38	0010010111111101#0	63	1101110111001011#1	88	0010010111111101#1
14	0110111111001001#0	39	1000011010111111#0	64	0110111111001001#1	89	1000011010111111#1
15	1101001011000101#0	40	0101001101001011#0	65	1101001011000101#1	90	0101001101001011#1
16	0101000101111010#0	41	1101101100110011#0	66	0101000101111010#1	91	1101101100110011#1
17	0011110010110000#0	42	0101011100110010#0	67	0011110010110000#1	92	0101011100110010#1
18	0101111010100100#0	43	0100111111111010#0	68	0101111010100100#1	93	0100111111111010#1
19	0010000111010101#0	44	0011000011100101#0	69	0010000111010101#1	94	0011000011100101#1
20	0011001110001101#0	45	0110100001011011#0	70	0011001110001101#1	95	0110100001011011#1
21	0011010101000100#0	46	0010001111110010#0	71	0011010101000100#1	96	0010001111110010#1
22	1010111000110001#0	47	1000100011010010#0	72	1010111000110001#1	97	1000100011010010#1
23	0011101100010111#0	48	0100001010010101#0	73	0011101100010111#1	98	0100001010010101#1
24	1111111110011110#0	49	0000000000000001#0	74	1111111110011110#1	99	0000000000000001#1

Núcleos de raio 3

	Núcleo
0	0101111100100111111100110111010000111110110111001001011011010000#0
1	0010001111001110101010011111100110110101011010010111100000101000#0
2	1110000110000110010010101111110110100101000001101011001110100110 1#0
3	1100011100100001110111011100101101101111110010011101001011000101#0
4	0101000101111010001111001011000001011110101001000010000111010101#0
5	0011001110001101001101010100010010101110001100010011101100010111#0
6	1111111110011110001000001111101101100101000110100110000110011101#0
7	0101010111000111010010000111110011101111000110110001101101111111#0
8	1111110011000010010100111011000000100100101101001100011110010111#0

9	0101001000101111001100011111101100100101111111011000011010111111#0
10	0101001101001011110110110011001101010111001100100100111111111010#0
11	0011000011100101011010000101101100100011111100101000100011010010#0
12	0100001010010101001010001111100000010011101110111111000101000010#0
13	1000000101001011101000001001110011100001000001110101011100100000#0
14	0101110101110110101111010110101110101100001001110010101010100010#0
15	0110000100000100111101011110010011110110000000000111000100011000#0
16	0000110000001000110010101011000000101000101011101001110100110000#0
17	1111000011000001011101100010101100101001111001000100110000001000#0
18	1110000110101001010001100110101110011011001010100010110011010101#0
19	1011101000000110011100010001000111110110111011100110100000110101#0
20	1010000001110101101001010011000001011000101010111001101100100101#0
21	0011001110100011010100111001011000101100000111101011010110000111#0
22	0110101011110100000001111001000111010001100111000110000100100010 1#0
23	1101101110100000001100001101001100011101000011101101110111010101111#0
24	0011101011101101011111010100100001100111100100111011111101001111#0
25	0100011000000000110010110110000111100000111110000110010001101111#0
26	0000010001101100011000011110101110011100011111011110011000000010#0
27	0100100011000000111101000001100111111101111101010000011000001010#0
28	0001111010110010000100100000000001010100100101001101010110011010#0
29	0101011001000100010011100110011000110000001000001111111111011001#0
30	1001000000010001100110101011011111111111011000001001010001110011#0
31	0011100001000011110100000101111001110101101011111101110110000000#0
32	1001000100101111011000010000000011111010100010011101001111101000#0
33	1001010111100110010111101101100110011010100001100000110110100 00#0
34	0011111010001110111111001011001111000101010001011001001010100101#0
35	0111100100100001010010111110100110110001010010010000010011111101#0
36	1110111101010010100010111100100010001100011010011010111010111101#0
37	1110010011000100110010100010111001011001101110011001010101111110#0
38	0001001011010011110001111111101111110010010111111011111011001100#0
39	1000101010001111111001111010101100110100010110010101101010101011#0
40	0110001110001011101001001001000010110010100111000001000011111111#0
41	0011011100000111011011111001100110010001010101110100100111110111#0
42	1000110001101010000011001011101110001001010010101101100011011010#0
43	1101111111001111110111111010000110101100010100101110111111011010#0
44	1100110110010101111111101001000010001100101100010101111001001010#0
45	0101011000100011011101110010011000010110101001100100000000001001#0
46	1110001100110110011000100011001010111011010001111101110010101011#0
47	0011110001101101000010101000011000100000101110000110011011110000#0
48	1101000101101000000101011101100011111111101001100111101000010000#0
49	0000000000000000000000000000000000000000000000000000000000000001#0
50	0101111100100111111001101110100001111101101110010010110110100000#1
51	0010001111001110101010011111100110110101011010010111100000101000 0#1
52	1110000110000110010010101111101101001010000011010110011101001101#1
53	1100011100100001110111011100101101101111110010011101001011000101#1
54	0101000101111010001111001011000001011110101001000010000111010101#1
55	0011001110001101001101010100010010101110001100010011101100010111#1
56	1111111110011110001000001111101101100101000110100110000110011101#1
57	0101010111000111010010000111110011101111000110110001101101111111#1
58	1111110011000010010100111011000000100100101101001100011110010111#1
59	0101001000101111001100011111101100100101111111011000011010111111#1
60	0101001101001011110110110011001101010111001100100100111111111010#1

61	00110000111001010110100001011011001000111111001010001000110100 10#1
62	01000010100101010010100011111000000100111011101111110001010000 10#1
63	10000001010010111010000010011100111000010000011101010111001000 00#1
64	01011101011101101011110101101011101100001001110010101010100010#1
65	01100001000001001111010111100100111101100000000001110001000110 00#1
66	00001100000010001100101010110000001010001010111010011101001100 00#1
67	11110000110000010111011000101011001010011111001000100110000001000#1
68	11100001101010010100011001101011100110110010101000101100110101 01#1
69	10111010000001100111000100010001111110110111011100110100000110101#1
70	10100000011101011010010100110000010110001010101110011011001001 01#1
71	00110011101000110101001110010110001011000001111010110101100000111#1
72	01101010111101000000111100100011101000110011100011000010010001 01#1
73	11011011101000000110000110100110001110100001110110111011101101111#1
74	00111010111011010111110101001000011001111001001110111111010011111#1
75	01000110000000001100101101100001111000001111100001100100011011111#1
76	00000100011011000110000111101011100111000111110111001100000000010#1
77	01001000110000001111010000011001111111011111010100000110000010 10#1
78	00011110101100100001001000000000010101001001010011010101100110 10#1
79	01010110010001000100111001100110001100000010000011111111111011001#1
80	10010000000100011001101010111011111111111011000001001010001110011#1
81	00111000010000111101000001011110011101011010111111011101100000000#1
82	10010001001011110110000100000000111110101000100111010011111010000#1
83	10010101111001100101111011011001100111010100001100000110110100000#1
84	00111110100011101111110010110011111000101010001011001001010100101#1
85	01111001001000010100101111101001101100010100100100000100111111101#1
86	11101111010100101000101111001000100010011000110100110110101111101#1
87	11100100110001001100101000101110010110011011100110010101011111110#1
88	00010010110100111100011111111011111100100101111110111110110011100#1
89	10001010100011111100111101010110011010001011001010110101010101111#1
90	01100011100010111010010010010000101100101001110000010000111111111#1
91	00110111000001110110111111001100100010101011101001001111110111#1
92	10001100011010100000110010111011100010010100101011011000110110 10#1
93	11011111110011111101111110100001101011000101001011101111110110 10#1
94	11001101100101011111111010010000100011001011000101011110010010 10#1
95	01010110001000110111011001001100001011010100110010000000000010 01#1
96	11100011001101100110001000110010101110110100011111011001010101 11#1
97	00111100011011010000101010000110001000001011100001100110111100 00#1
98	11010001011010000001010111011000111111111010011001111010000100 00#1
99	00000000000000000000000000000000000000000000000000000000000000 01#1

Núcleos de raio 4

0	0101111100100111111001101110100001111101101110010010110110100000010001111001101010100111110011011010101101001011100000101000011100001100001100100101011111011010010100000110101100111010011011100001110010011010111000010100000111000011000011
1	0101000101111010001111001011000001011110101001000010000111101010100110011100011010001010101000100101011100011000100111011000101111111111111001111000100000011111011011001010001101001100011101010101011100011010010111011000101111111111#0
2	1111111001100001001010011101100000010010011010011000111100101111010100100010111100110001111110110010010111111101100001101011111101010011010011110111001001011111111010000111010111111010100101000100111101010010#0
3	0100001010010101001010001111100000010011101111111000101000010100000010100101111010000010011100111000010000011101010111001000000101110101111010101110101101011010111000010011001010101010001001100001000000100111101011110010011110110000000001110001000110000#0
4	0000011000000100011001010101100000010100010101111010011101001100001111000011000001011011000101101100101001111001000100110000001000111000011010100101000110011010111001011001010100010110011010101101110100000011001110000100111010111011001101000001101010#0
5	1010000000111010110100101001100000101100010101011100110110010010010011001110100010100110010110001010100000111010110101100001110110101011110100000111001000111010001101100111000010010001011101101111010000010100001110100111011010111#0
6	0011101011101101011111010100100000110011110010011101111101001111010001100000000011001011011000011110000011111000011001000110111000001000110100011000111010111001110000111110111110011000000010010010001100000011110100000110011111011010100000110000010#0
7	0001111010110010000100100000000001010100100101001101010110011010010101100100010001100110011000100000010000011111111101100110010000000010001100110101011101111111111101000001001010001110011001110000100001111010000001011100111010110101011111101110100000#0

8	10010001001011110110000100000000111110101000100111010011111010001001010111100110010111101101100110011101010000110000011011010000000111110100011101111110010110011110001010100010110010010101001010111100100100001010010111110100110110001010010010000010011111101#0
9	11101111010100101000101111001000100010011000110100110110101111011110010011000100110010100010111001011001101110011001010101111110000100101101001111000111111110111111001001011111101111110110011001000101010001111110011110101011001101000101100101011010101010111#0
10	01100011100010111010010010010000101100101001110000001000011111111001101110000001110110111110011001100100010101011101001001111011110001100011010100000011001011101110001001010010101101100011011010110111111100111111011111101000011010110001010010111011111011010#0
11	11001101100101011111111010010000100011001011000101011110010010100101011000100011011101110010011000010110101001100100000000000100111100011001101100110001000110010101110110100011111011100101010110011110001101101000010101000011000100000101110000110011011110000#0
12	11010001011010000000101011101100011111111101001100111101000010000010001111100000111110000010000011101011001001100101110110000000001101111001010010001101101001010100001000101001100110010011111010010110110001110101011100011100111100001010001111011011101100101#0
13	01111011011101010110000100010000110001001010101001001100011101011011010011000011011000000111101100000101010011110011010101010011011010101110101001111100100001010011101110101000000001010000111100011011011000100010110111111001110001011011000010100001100100011111#0
14	00010101101011100000101001010000111000000101110110010001010100110111100000011000101110100010010010000111101110000111100110001111111011001100001011000011000000001001011011111011010101000101001010100111100010100000110011011110010010100101111101110100010000000011#0
15	10001000100110111111001010100100010011011111110110001001101101001000000000100010100110100011011111101100011110000000100001000100100010110011100011001110011001100101001101000111010000111001101010011110011110100101010100100010001101110001010011111010010000010100#0
16	00100110010001111011111110000000010001100111000001100010011000001010110101101111011001001100110010000001101101011100110011111100001101001011001100010001110011010001100111111010001011101101101010010110011001000101101001110000100110110001000000101110111001010#0
17	00111000010111010110101000100000010110110110001011110000010000101001110111111111000100110000100110100011101101000101000000011101101000010011111011010110001001001000000110111111011100101001111001000011110110101111000100000000101101011011011000010111000100000111111#0
18	10010100101000010010001011001000010001011101110010101110011111101010100001011100110110001001000101001110001101100111111111001101001110101010001000111010110110100011111101110001010101111110010000010010000000111110001001101010010101101100011000011110011111101#0
19	10011010111111000100100110010000011110101101100110001000001010101111100100011111100010100110011010000000111111101010011011001001001101000001111111000101000110100100010111001110110000001110000111000101001001101100011010001111010011111111101011000100011100000#0
20	01100110011000001010000011110000010010001100011110101101010010011001100110111001001011110101100101010001000011000111010101100010100110000100000110101101010100100110001111011011111001010000011111111011111011110001100001011001000101100000011100010001010011000#0
21	11101101000111011111111011111000001000111101001011100011010111010100101011011011101101111110111110110111011111001101110110110001010100000101101110001100010010010100101100011010010101000110011100000101010110010000000100011100110110100101110110111011011110001110#0
22	11000011100001101001010100110000000101001101001101000101101011100011111101111011101100100010100000011000000110010000010000010100011011101010010100010001000010101010101111000101100011110110100101101001000000001010111001010000000011100001111101000110011010111#0
23	01111111110110110110001101110000000101100111011110011000111110101011110000111001001001101001011101011000111110011001011100111101111000010111101001111011010000101001011110001001111011011001111101011000001000100011011110100111000011011111001010110111010100000#0
24	01110001011001100000100001101000010100111110100000010011101100011100110110011110101101100010000011010100000000110100111010110011110111110110011011000100110010101010010010011100111010111101111010001011110000000001100111001100011000001001010001110110010010100#0
25	10011101001110100010001100100000001100000111101010101110111000101100011000111110010010011010001110001000111000000011101110000000100101001110011011101011101001001011011011101111010100111010101000111011101100011010100110001111110000000111001010011101010011111#0
26	00000011011000011111111011110000101011010111101000111100101010110011111100111110000001101111110001100100100111101010010010101100011111011000101011100101111110010111010011010001100100000101110010011111000100111011101101100100010001100001100011000010000100011#0
27	00101100110101001001011011101001111111011110100000011111001000100010111111011110100011000100111101001101110111000011100111100101110000010101010100010000111110101100001010100110000001100111011110100010100000101010100100111011101001000110101010001111110001000#0
28	10100000100010011110011010110001100001101100111100010011011000011010111010110000100110001011100010100010001001001100000000100101100001011011010000111101000110010111111100101110110101001000000110010001001100011111001111010000000100010000011111111000110000000100#0
29	10011010000101000000100111110001101101001101101010110101100101001101110100110011000010011000011101100010110011110001101111010010001001011011010000000101011100101000111110101010011111000101000010110111111000110100111111001110101010111001101011101011110111 0#0
30	00010110011011100110001010111001100010111101001011010011110011101000100001010011001001010010000011101110000110011000101011111100101100001110110111101111011010101011010010100100000110100010000111100110010000010101100000001100011010001000001101010100011011011#0
31	11110000111100110011110000110001111110001100011111011110110010110110101111110001101100011001011100101111111001101010001011111011001111001010010101001000010001010000000100100010001111010100111000001110001001011001000101011110001100111111010111000010101001#0
32	11101110101011101101011101110001110000000100010101011110010111001001110111011011100110000011010001110001100000001001111001100000110001000000001111110100111110001001111011100101001011110011010110010101011011000001100100000110000101111000011000011110111111100#0
33	01000011000100011010010100001001111101111001110110001010011011001100101000010100001001000101111111111011111101000101010000010101011101100100100100011100111101010010111001000110011010000100010000100000011010100011100010110011111010110101000111000101001110110#0
34	01101101010010011100101000000000110011100101111011110111000011110110010010111010011110011110110000011011001010110001100110010010011100010111010100100000001011001001100110100010111101011001001001101000010101111011110001111000000000100100100000100010011000010 11#0
35	11110011100101000010001011000001101000101010000011100000001011011101110001111011001011011001001111111101010110100110011101000110101001110011101011010111011111001000110101100101101011000000111110111100100011110101100100010101111101111110011101110010001000000 1#0
36	00011111101010010111111011000100110110001101001010100011000100001011010111101000000100111110010000011111100110110100010111000100110101010100111000111001011011101000001010101100111001110100101110100100111001111101110000101010000100111101110111011110010010010 0#0
37	00010101011001101001111010010011010010101011111100100010101101000000111000110010110011100010011110110101100011111000010011110011111110111000010110010011001100100011100010100011111111011010110010100000011011011110110111001111110000011100001001010000000001111 0#0
38	10101001011110111111001001101001101111110010101010001011001111011011011111001100110101101001011000011100010110000101111110111111100000001111101110101111000011100100100001110101100111011011001100100110000011110011100111100111000011001100110111011111011100100 11#0
39	10100111100010111000101100011011010001111010011101001000100000111100010111010001010011111111111110101001100100100110010001010100000001001100010100100001100001010011101001011100101000001111001101011000111011011010011111100011111100110100110000100011001111001 #0
40	00011000100101100010000100010011001001000001001001101010010100111010010011111111111100111010010000010110001111010101101111110101110100101001101110010110100000011010111001111010100100011101101101100001010111000110110100111000000101010011100011001010111110110 #0
41	10010100010010010111101001010011000110000000100000010110000100100011101010011111101110001101101111111000001000000000000111001100011001110110110101011110110110000111010001111110010000100101101000011001011100111100100011100100011111101101101000110000110011001110 #0
42	11111010011101000001111110000101100101011000011010000000100101100000001100001110101110000100000000101000000010001101100010100001100000101001111001110010110100010111001000011001011010000010100000101010100010111100010011111111100011110101010010011100010101001000 #0
43	01000110101011011110010101000011010101000000010001010011101111011110001111101111011100100001001110010110110111101000110011001110010011000101011011111100110010011101010101101111001101111000101100010010111111100101100110001111100110001101101110111011111100001 #0
44	11001100111100011000101011000011011001100001010011010001100010101011010011111000111100011001100011101001010111011000100000110011111000100001000000000010110100111101110111011010001010110110011010110101101111101011010110101000111110000101011010101110000111 11#0
45	11100010011011101110000010001011010111010011011111011101101110010100000001011000010110010110111100100001101001110111001100111011011011100100110101101100100110111111001101100001100001101101010011101011000011011001001000000101110111110010101100101101110001011 0#0
46	01111111000100110011101111111001111110001000100010101110110111101010000010011101011001100010110001011011001001000011010101001010011000000110101100011011100010011110001001101011010100011111110011000011101110001101100000110000010110000001001110010001001011100 0#0
47	00010001100010100101110011110011100000000010111110011101101010110001001010111001110010001110011101110110100110101011101100101100111111101010101111010101010100111111101111101100011100110110011100111010011110011100110011110011101110001100000011001100100110001 #0
48	10011111111010110001001100011101110111011001110111101000101010010110100111101111101011010001000001001101001110000000000001110101101110011011101111011101000000101110001000111011011101010010011110001101011110100110000110101011000101111010111000100000000000001 11#0
49	0001#0
50	01011111001001111110011011101000011111011011100100101101101000000001000111100111010101001111100110110101011010010111100000101000011100001100001100100101011111011010010100000110101100111010011011100011100100001110111011100101101101111111001001110100101100010 1#1
51	01010001011110100011110010110000010111101010010000100001110101010011001110001101001101010100010010101110001100010011101100010111111111111100111100010000001111101101100101000110100110000110011101010101011100011101001000011111001110111100011011000110110111111 1#1
52	11111100110000100101001110110000000100100101101001100011110010111010100100010111100110001111110110010010111111101100001101011111101010011010010111101101100110011010101110011001001001111111110100011000011100101011010000101101100100011111100101000100011010010 #1
53	01000010100101010010100011111000000100111011101111110001010000101000000010100101110100000100111001110000100000111010101110010000001011101011101101011110101101011101011000010011100101010101000100110000100000100111101011110010011110110000000000111000100011000 #1
54	00001100000001000110010101011000000010100010101110100111010011000011110000110000001011101100010101100101001111001000100110000000100011100000110101001010001100110101110011011001010100010110011010101101110100000011001110001000100011111011011101110011010000011010 1#1
55	10100000011101011010010100110000010110001010101110011011001001010011001110100011010100111001011000101100000111101011010110000111011010101111010000001111001000111010001100111000110000100100010111011011101000000110000110100110001110100001110110111011101011111 #1
56	0011101011101101011111010100100001100111100100111011111101001111010001100000000011001011011000011110000001111100001100100011011110000001000110110

	001100001111010111001110001111101111001100000000100100100011000000111101000001100111111101111101010000011000001010#1
57	000111101011001000010010000000000010101001001010011010101100110100101011001000100010011100110011000110000001000001111111110110011001000000001000 110011010101110111111111101100000100101000111001100111000010000111101000001011110011101011010111111011101100000000#1
58	1001000100101111011000010000000001111101010001001110100111110100010010101111001100101111011011001100111010100001100000110110100000001111101000111 0111111001011001111000101010001011001001010100101011110010010000101001011111010011011000101001001000000100111111101#1
59	11101111010100101000101111001000100010011000110100110110101111011110010011000100110010100010111001011001101110011001010101111110000100101101001 111000111111101111111001001011111101111110110011001000101010001111110011110101011001101000101100101011010101010111#1
60	01100011100010111010010010010000101100101001110000010000111111110011011100000111011011111001100110010001010101110100100111110111100011000110101 000001100101110111000100101001010110110001101101011011111110011111101111110100001101011000101001011101111111011010#1
61	11001101100101011111111101001000010001100101100010101111001001010010101100010001101110111001001100001011010100110010000000000100111100011001101 1 001100010001100101011101101000111110111001010101100111100011011010000101010000110001000001011100001100110111110000#1
62	110100010110100000001010111011000111111111010011001111010000100000100011111000001111100000100000111010110010011001011101100000000011011110010100 10001101101001010100001000101001100110010011111010010110110001110101011100011100111100001010001111011011101100101#1
63	011110110111101010110000100010000110001001010101001001100011101011011010011000011011000001111011000001010100111100110101010100110110101011101010 0111110010000101001110111010100000000101000011110001101101100010001011011111001110001011011000010100001100100011111#1
64	00010101101011100000101001010000111000001011101100100010101001101111000000110001011101000100100001111011100001111001100011111110110011000010 1100001100000000100101101111110110101010001010010101001111000101000001100110111100100101001011111101110100010000011#1
65	10001000100110111111001010100100010011011111110110001001101101001000000001000101001101000110111111011000111100000010000100010010001011001110001100 111001100110010100110100011101000011100110101001111001111010010101010010001000110111000101001111101001000000101000#1
66	001001100100011110111111100000000010001100111000001100010011000001010110101101111101100100110011001000000011011010111001100111111000011010010110011 000100011100110100011001111110100010111011011010100101100110010001011010011100001001101100010000010111011100101 00#1
67	00111000010110101101010001000000010110110110001011110000001000010100111011111111100010011000010011010001110110100010100000001110110100001001111101 101011000100100100000110111110111001010011110010000111101101011110001000000001011010110110110000101110001000011111#1
68	1001010010100001001000101100100001000101110111001010111001111110101010000101110011011000100100010100111000110110011111111001101001110101010001 000111010110110100011111011100010101011111001000001001000000001111100010011010100101011011000110000111100111111011#1
69	10011010111111100010010011001000000111101011011001100010000001010101111100100011111100010100110011010000000111111101010011011001001001101000001111 11100010100011010010001011100111011000001110000111000101001001101100011010001111010011111111101011000100011100000#1
70	01100110011000001010000011111000000010010001100011110101101010010011001100110111001001011110101100101010001000011000111010101100010100110000100000 1101011010101001001100011110110111110010100000111111110111110111110001100001011001000101100000001110001000101001100#1
71	11101101000111011111111101111100000100011110100101110001101011101010010101101101110110111110111101101110111100101110110110001010100000101101110 001100010010010100101100011010010101000110011100000101010110010000000010001110011011010010111011011101101111000111#1
72	110000111000011010010101001100000000101001101001101000101101011100011111101111011101100100010100000011000000110010000010000010100011011101010010 100010001000010101010101111000101100011110110100101101001000000000101011100101000000001110000111110100011001101011#1
73	0111111111011011011000110111000000010110011101111001100011111101010111100001110010010011010010111010110001111100110010111001111011110000101111101 001111011010000101001011110001001111011011001111101011000001000100011011110100111000011011111100101011011101010000#1
74	0111000101100110000010000110100001010011111010000000100111011000111001101100111101011011000100000110101000000000110100111010110011110111110110011 011000100110010101010010010011100111010111101111010001011110000000001100111001100011000001001010001110110010010100#1
75	10011101001110100010001100100000001100000011110101010111011100010110001100011111001001001101000111000100011100000001110111000000010010100111001101 110101110100100101101101110111101010011101010100011101110110001101010011000111111000000001110010100111010100111111#1
76	00000011011000011111110111100001010110101111010001111001010101100111111001111100000011011111100011001001001111101010010010101100011111011000101 0111001011111100101110100110100011001000000101110010011111000100111011101101100100010001100001100011000010000100011#1
77	001011001101010010010110111010011111110111101000000011111001000100010111111011110100011000100111101001101110111000011100111100101110000010101010 1000100000111110101100001010100110000001100111011101000101000000101010100100111011101001000110101010001111110001000#1
78	101000000100010011110011010110001100001101100111100010011011000011010111010110000100110001011100010100010001001001100000001001011000010110110100 0011110100011001011111100101110110101001000000110010001001100011111001111010000000100010000011111111000110000000100#1
79	100110100001010000000100111110001101101001101101010110101100101001101110100110011000010011000011101100010110011110001101111010010001001011011010 0000001010111001010001111101010100111110001010000101101111110001101001111110011101010101110011010111010111101110#1
80	00010110011011100110001010111001100010111101001011010011110011101000100001010011001001010010000011101110000110011000101011111100101100001110110 111101111011010101011010010100100000110100010000111100110010000010101100000001100011010001000001101010100011011011#1
81	1111000011110011001111000011000111111000110001111101110110010110110101111110001101100011001011100101111111100110101000101111101100111100101001 0101001000001000101000000010010001000111101010011100000011100010010110010001010111100011001111110101110000101010001#1
82	1110111010101110110101110111000111000000100010101011110010111001001110111011011100110000011010001110001100000010011110011000001100010000000111 111010011111000100111101110010100101111001101011001010101101100000110010000011000010111100001100001111011111111100#1
83	01000011000100011010010100001001111101111001110110001010011011001100101000010100001001000101111111110111111010001010100000101010111011001001001 00011100111101010010111001000110011010000100010000100000011010100011100010110011111010110101000111000101001110110#1
84	011011010100100111001010000000001100111001011110111101110000111101100100101110100111100111101100000110110010101100011001100100100111000101110101 00100000001011001001100110100010111101011001001001101000010101111011110001111000000000100100100000010001001100001011#1
85	111100111001010000100010110000011010001010100000111000000010110111011100011110110010110110010011111111010101101001100111010001101010011100111010 11010111011110010001101011001011010110000001111101111001000111101011001000101011111011111100111011100100010000001#1
86	0001111110101001011111101100010011011000110100101010001100010000010110101111010000001001111110010000011111110011011010001011100010011010101010011100 011100101101111010000010101011001110011101001011101001001110011111101110000101010000100111101110110111011100100100100#1
87	000101010110011010011110100100110100101010111111001000101011010000000111000110010110011100010011110110101100011111000010011110011111110111000010 1100100110011001000111000101000111111101101011001010000000110110111101101110011111100000111000010010100000000011110#1
88	10101001011110111111001001101001101111110010101010001011001111011011011111100110011010110100101100001110001011000010111110111111100000001111101110 101111000011100100100001110101100111011011001100100110000011110011100111100111000011001100110111011111101110010011#1
89	10100111100010111000101100011011010001111010011101001000100000011110001011101000101001111111111111101010011001001001100100010101000000010011000101 00100001100001010011101001011100101000001111001101011000111011011010011111100011111100110100110000100011001111001#1
90	00011000100101100010000100010011001001000001001001101010010100111010010011111111111001110100100000101100011110101011011111101011101001010011011 100101101000000110101110011110101001000111011011011000010101110001101101001110000000101010011100011001010111110110#1
91	1001010001001001011110100101001100011000000010000000101100001001000111010100111111011100011011011111100000100000000000111001100011001110110110101 01111011011000011101000111110010000100101101000011001011100111100100011100100011111101101101000110000110011001110#1
92	111110100111010000011111000010110010101100001101000000001001011000000001100001110101110000100000000101000000010001101100010100001100000101001111001 110010110100010111001000011001011010000010100000101010100010111100010011111111100011110101010010011100010101001000#1
93	0100011010101101111100101010000110101010000000100010100111011110111100011110111110111001000010011100101101101111010001100110011100100110001010110 11111100110010011101010101101111001101111000101100010010111111100101100110001111100110001101101110111011111100001#1
94	11001100111100011000101011000011011001100001010011010001100010101011010011111000111100011001100011101001010111011000100000110011111000100001000 0000000101101001111011101110110100010101101100110101101011011111010110101101010001111100001010110101010111000011111#1
95	11100010011011101110000001000101101011101001101111110111011101110010100000001011000010110010110111100100001101001110111001100111011011011100100110 10110110010011011111100110110000110000110110101001110101100001101100100100000010111011111100101011001011011100010110#1
96	01111111000100110011101111110011111000100010001010111011101111010100000100111010110011000101100010110110010010000110101010010100110000001101011 0001101110001001111000100110101101010001111111001100001110111000110110000011000000101100000010011100100010010111000#1
97	0001000110001010010111001111001110000000000101111110011101101010110001001010111001110010001110011101110110100110101010111011001011001111111010101 11101010101010011111110111101100011100110110011100111010011110011100110011110011101110001100000011001100100110001#1
98	1001111111101011000100110001110111011101100111011110100010101001011010011110111111010110100010000010011010011100000000000011101011011100110111011 110111010000010111000100011101101110101001001111000110101110100110000110101011000101111010111000100000000000000111#1
99	00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000001#1

Apêndice I Amostra de núcleos após a filtragem

Núcleos de raio 1

	Núcleo		Núcleo		Núcleo		Núcleo		Núcleo		Núcleo		Núcleo		Núcleo
0	1011#0	3	0110#0	6	0011#0	9	0010#0	12	1011#1	15	0110#1	18	0011#1	21	0010#1
1	1100#0	4	0001#0	7	0100#0	10	1000#0	13	1100#1	16	0001#1	19	0100#1	22	1000#1
2	1101#0	5	1110#0	8	1001#0	11	0111#0	14	1101#1	17	1110#1	20	1001#1	23	0111#1

Núcleos de raio 2

	Núcleo		Núcleo		Núcleo		Núcleo
0	10111100110101110#0	25	00000111100101110#0	50	10111100110101110#1	75	00000111100101110#1
1	00001011000111110#0	26	10011110011010000#0	51	00001011000111110#1	76	10011110011010000#1
2	00111010111101000#0	27	11001001111101010#0	52	00111010111101000#1	77	11001001111101010#1
3	10101001000111010#0	28	00010010101111000#0	53	10101001000111010#1	78	00010010101111000#1
4	01011101010100010#0	29	01000101101101100#0	54	01011101010100010#1	79	01000101101101100#1
5	10000011010000010#0	30	00000100111110110#0	55	10000011010000010#1	80	00000100111110110#1
6	00011000111010000#0	31	10111110101101000#0	56	00011000111010000#1	81	10111110101101000#1
7	10011101011111111#0	32	11100001111101110#0	57	10011101011111111#1	82	11100001111101110#1
8	11101101011001010#0	33	11100001100011000#0	58	11101101011001010#1	83	11100001100011000#1
9	11000101010101110#0	34	10111000011101100#0	59	11000101010101110#1	84	10111000011101100#1
10	10001010000010110#0	35	01011001110101000#0	60	10001010000010110#1	85	01011001110101000#1
11	11100110110101110#0	36	01011111101100100#0	61	11100110110101110#1	86	01011111101100100#1
12	11001101001110000#0	37	11000011111101100#0	62	11001101001110000#1	87	11000011111101100#1
13	00001110010101010#0	38	01101110101101100#0	63	00001110010101010#1	88	01101110101101100#1
14	00110010001110010#0	39	10001100101010010#0	64	00110010001110010#1	89	10001100101010010#1
15	00110000101110010#0	40	00010000100110100#0	65	00110000101110010#1	90	00010000100110100#1
16	01100111101001110#0	41	00011111101011110#0	66	01100111101001110#1	91	00011111101011110#1
17	10111001000000000#0	42	11001000001001010#0	67	10111001000000000#1	92	11001000001001010#1
18	01101010001100010#0	43	11000011001110010#0	68	01101010001100010#1	93	11000011001110010#1
19	11010001000011100#0	44	01111100110100010#0	69	11010001000011100#1	94	01111100110100010#1
20	01001110010010110#0	45	11011101011101000#0	70	01001110010010110#1	95	11011101011101000#1
21	11011001000001000#0	46	01110010100110100#0	71	11011001000001000#1	96	01110010100110100#1
22	01000000101110000#0	47	01000001011000010#0	72	01000000101110000#1	97	01000001011000010#1
23	10010001101111100#0	48	10010101101100100#0	73	10010001101111100#1	98	10010101101100100#1
24	01001011000100010#0	49	11101111001000110#0	74	01001011000100010#1	99	11101111001000110#1

Núcleos de raio 3

	Núcleo
0	10111100110101100000101100011110001110101111010010101001000111010#0
1	01011101010100101000001101000010000110001110100010011101011111111#0
2	11101101011001011100010101010111100010100000101111100110110101111#0
3	11001101001110000000111001010101001100100011100100110000101110010#0
4	00000100000100000110011110100111011100100000000011010100011000100#0
5	01101111101101101000000101000000110100010000111001001110010010110#0
6	11011001000001000100000010111000100100011011111001001011000100010#0

7	000001111001011011101011011011111001111001101000110010011111010 1#0
8	100001001000000000001001010111100010001011011011000000100111110 11#0
9	101111101011010010011110011111001110000111110111111000011000110 0#0
10	000100011100011110111000011101101110101010001010010110011101010 0#0
11	010111111011001011000011111101100110111010110110010010001100010 0#0
12	100011001010100100010000100110100001111110101111110010000010010 1#0
13	110000110011100101111100110100011101110101110100011001011011011 0#0
14	011100101001101001000001011000011001010110110010111011110010001 1#0
15	001011111001110001010000010110010111010110101111010010000011011 0#0
16	100110111010100100111111100001010001110010000010100011110111011 11#0
17	010111011110001101010111010001000111010101111001101100100100100 0#0
18	111011011100110001101011000010100001100101110010110011110111000 0#0
19	101010111101111000011101101101111000011101011011011111101010101 0#0
20	011110010111100101011110100000101010000011101011110011101111100 1#0
21	011011100000011010001101110010110101010110100001001001111110010 0#0
22	001011100000101110111000011001001110100111101010011111011111010 1#0
23	111011100000000010101100010010101000011011011110101001100111101 0#0
24	110101111000111000001101011100110000110010100000010100010100001 0#0
25	010100001101111101001011001000001010110000111001100011011111011 1#0
26	111001010101111011100110001011010110101110011110001110101000101 1#0
27	010111001010011010110011110001000011110111110000011100011011011 0#0
28	100110011100000001101010000001110101100001001001101000101111000 0#0
29	110010011111100100100001110011000001111000110000000001000000100 1#0
30	111000000011001000001001101010001101101100000000011111000011011 1#0
31	010111001100110101011001010001100111011000011001000110011011111 0#0
32	100100100101000010101110110011100110110100001000110001011010000 0#0
33	010100111110000110101110010010010010001111111100110110111010010 0#0
34	111110101000001101011000011001110001011001000110101001100000111 1#0
35	110011011011001100101001011100001000001100001111110010101010000 1#0
36	000001100111000110111010000000010010111100010100101000110111010 0#0
37	111100010100000101011100011011100011000010010101000010001100011 0#0
38	101101111000111111111110011010100001010111000011100111110000010 0#0
39	101100111001100011100001001010000000010110000011100000100101101 1#0
40	010000100100011111111111101010000101000000000001111111000001101 100#0
41	010000101101110101000000110010110110000100011000000011000011000 0#0
42	100100000000100110101100111110010000111111101110000111100100101 1#0
43	011100101001001000101101001100010100010101001001011100000011010 0#0
44	101101000101101110000001000110100010000010000110110010100110100 1#0
45	100100111111001011000100010101011001110011111110010001100110001 1#0
46	010110101110010001110010111010110011001010011100011010001100000 0#0
47	011010010100000011001011011011010100101111110111010001100000010 #0
48	110001111111000100101000000001111111100011100110000100100110101 0#0
49	101001001101111111011011101000000100110000111000111011101010010 0#0
50	101111001101011000001011000111100011101011110100101010010001101 #1
51	010111010101001010000011010000100001100011101000100111010111111 1#1
52	111011010110010111000101010101111000101000010111110011011010111 1#1
53	110011010011100000001110010101010011001000111001001100001011100 1#1
54	000001000001000001100111101001111011100100000000011010100011000 1#1
55	011011111011011010000001010000001101000100001110010011100100101 1#1
56	110110010000010001000000010111000100100011011111100100101100010 001#1
57	000001111001011011101011011011111001111001101000110010011111010 1#1
58	100001001000000000010010101111000100010110110110000001001111101 1#1

59	10111110101101001001111001111100111000011111011111000011000110 0#1
60	00010001110001111011100001110110111010101000101001011001110101 00#1
61	01011111101100101100001111111011001101110101101100100100011000 100#1
62	10001100101010010001000010011010000111111010111111001000001001 01#1
63	11000011001110010111110011010001110111010111010001100101101101 10#1
64	01110010100110100100000101100001100101011011001011101111001000 11#1
65	00101111100111000101000001011001011101011010111101001000001101 10#1
66	10011011101010010011111110000101000111001000001010001111011011 111#1
67	01011110111100011010101110100010001110101011110011011001001001 00#1
68	11101101110011000110101100001010000110010111001011001111011100 00#1
69	10101011110111000011101101101111000011101011011011111101010101 010#1
70	01111001011100101011110100000101010000011101011110011101111001 #1
71	01101110000001101000110111001011010101011010000100100111111001 00#1
72	00101110000010111011100001100100111010011101010011111011111010 1#1
73	11101110000000001010110001001010100001101101111010100110011110 10#1
74	11010111100011100000110101110011000011001010000001010001010000 10#1
75	01010000110111110100101100100000101011000011100110001101111101 11#1
76	11100101010111101110011000101101011010111001111000111010100010 11#1
77	01011100101001101011001111000100001111011111000001110001101101 10#1
78	10011001110000000110101000000111010110000100100110100010111100 00#1
79	11001001111111001001000011100110000011110001100000000010000001 001#1
80	11100000001100100000100110101000110110110000000001111100001101 11#1
81	01011100110011010101100101000110011101100001100100011001100111 11#1
82	10010010010100001010111011001110011011010000100011000101101000 00#1
83	01010011111000011010111001001001001000111111110011011011101001 00#1
84	11111010100000110101100001100111000101100100011010100110000011 11#1
85	11001101101100110010100101110000100000110000111111001010101000 01#1
86	00000110011100011011010000000010010111100010100101000110111010 0#1
87	11110001010000010101110001101110001100001001010100001000110001 10#1
88	10110111100011111111111001101010000101011110000111001111100000 100#1
89	10110011100110001110000100101000000000101100000111000001001011 011#1
90	01000010010001111111111110101000010100000000000111111000001101 100#1
91	01000010110111010100000011001011011000010001100000001100001100 00#1
92	10010000000010011010110011111001000011111111011100001111001001 011#1
93	01110010100100100010110100110001010001010100100101110000001101 00#1
94	10110100010110111000000100011010001000001000011011001010011010 01#1
95	10010011111100101100010001010101100111001111111001000110011000 11#1
96	01011010111001000111001011101011001100101001110001101000110000 00#1
97	01101001010000001100101101101101010010111111011101000110000000 10#1
98	11000111111100010010100000000111111110001110011000010010011010 10#1
99	10100100110111111101101110100000010011000011100011101110101001 00#1

Núcleos de raio 4

0	1011110011010110000010110001111000111010111101001010100100011101010111010101001010000011010000100001100011101000100111010111111111110110101100101111000101010111100010100000101111001101101011111001101001110000000111001010101001100100011100100110000101111001#0
1	0000010000010000011001111010011110111001000000001101010001100010110111110110101000000101000000111010011100100101111011001000001000010001100010011011100010001011100100100100100010111000100111000100#0
2	1000010010000000000100101011110001000101101101100000010011111011011111010101001011110011111001110000111101111100001100010000010001110001111011110101010001010011011100011101010011011100100100111000100#0
3	1000110010101001000100000100110100000111110101111100100000100101110000110011100101111100101000111011010111010001100101101101100111001010011010010000010110000110010101101100111011100100010000111010010110111001010011100010100100100#0
4	1001101110101001001111111000010100011100100000101000111101101111010111011110111000110101011101000100011101010111100101100100100100111011011100110001101010110000101000011001011100101110111000010101011101111000010101010111111010101010#0
5	0111100101111001010111101000001010100000111010111001111011111001011011100000011010001101110010110101010110100001001001111110010000101110000010111101001111010011101010011101000000111010111000011001001110100111101111010111011100000000001010110001001010100001101101111010100110011110100#0

6	1101011110001110000011010111001100001100101000000101000101000010010100001101111101001011001000001010110000111001100011011111011111100101010111 0111001100010110101101011100111100011101010001011010111001010011010110011110001000011110111110000011100011010110#0
7	1001100111000000011010100000011101011000010010011010001011110000110010011111100100100001110011000001111000110000000001000000100111100000011001 000001001101010001101101100000000011111000011011101011100110011010101100101000110011101100001100100011001100111111#0
8	1001001001010000101011101100111001101101000010001100010110100000001010011111000011010111001001001001000111111110011011011101001001111010100000001 10101100001100111000101100100011010100110000011111100110110110011001010010111000010000011000011111100101010100001#0
9	0000011001110001101110100000000100101111000101001010001101110100111100010100000101011100011011100011000010010101000010001100011010110111100011 1 11111111001101010000101011110000111001111100001001011001110011000111000010010100000000010110000011100000100101 1011#0
10	0100001001000111111111101010000101000000000000111111000001101100010000101101110101000000110010110110000100011000000011000011000010010000000000100 1101011001111100100001111111011100001111001001011011100101001001000101101001100010100010101001001011100000001 10100#0
11	1011010001011011100000001000110100010000010000110110010100110100110010011111100101100010001010101100111001111111001000110011000110101101011100 10 0011100101110101100110010100111000110100011000000011010010100000011001011011011010100101111110111010001100000000 10#0
12	1100011111110001001010000000011111111100011100110000100100110101010100100110111111101101110100000010011000011100011101110101001001000110011111 01 01100010111111110010100101110100010011010001100101110110001011011010111110000010101110110011010111101110111110111#0
13	0001001011011101110000101010101111011011100010101100010111100001000101011001111010011001001001100100000001010100001101011001000101100100001001 10 000010110100111100000000011111011011110011101010000101100111010110011011101000110100011000100011010110010111000000#0
14	1010100000000010010001010000000101111000000001111011001001000001001011011010111010001101100100000100001011000111111001011001010100111010110101 11 00010001011001101010100011010111110110111011110011100111011100011010001000111111101100001111111010000100011110001#0
15	0110110111000100100100111000101110111100101111011001110011001111111010011001011101110001111001110011111001101011001001110110010011001111001110 11 00101111011001011000111010100010010101000000010001000110010110011110111111100001101111010000011101000111011011 1011#0
16	1001001000101011010101000011101111111010000001011111011101111010111100000001001111011000011100011100001001101101000100001110101111111000001110100 1000 000010010101011001100011101001011111111010000111100110101111111000111001101111100100001010001100111111000000 01111011#0
17	1101001100100111011101110010001100100010110100001111101100110100111100011001000011100000001110011101111001000000011010000011011011100011111101 10 010011011111110101000011010111111011101011110100110110011010101000011110011000011100100000101110110001110000101 0101#0
18	0110101110011111010111110000101111101010010111010101011011010001100001011000011001000100001001011010101001000110010100101011111111010001111101 00 1110111001101010111101010000111000101000010111100010101100011111101011010001111011001011010101010001100010101 1000#0
19	1011101100110110100100001110110011111001011100111111101011010001011010111010111100010100101101001110001101110000111101111000010101101101011001 0101011000010001100110011000000001010100010111100111011101011011011011010000110001110101000011110011010010011 11000#0
20	1111000001110101100101010101100011011011101101011001100000111100100001010100001101010110111001111010000100101001001111100110001101001010100001 11100011101001011101101100101010101110100000100010110100111010101000010110100110001011110010000100111100111010010#0
21	1100010000111011001101100110101011000000111001101010000011101000100111100100010100100100010111001010110000000110110100110001100011100100010000 0 1100100000101101010100100010010100010001011101010100111100101101100100001110011101110000011101011111111111011100 01#0
22	0110001110001110001010111001100100100011111010010111000110101111101110000101100110010011101010111000010000100011001000010111000101110110110110 0 10110110111111010001100111010110010101101010111100010001101110110110100001011101100001111001110111101101000110010 1#0
23	1000111100111000000101100110101011001010011101001011101111011111001100001101101000010110011000100000000111100111111010111001010110100111001110001 1001000001111011111000001111110011011001011110101001101000011000100000000110011001010001001011011011011100101001 10#0
24	0010110011111010000110001101110111000000001101101100000110001001101000000110011101101110101101001101100111000111011101101011111001001011111100 01 0101011001101011011001011101010000000010001011010100100110100000100101010111110101101010000010000111010011001011 10#0
25	1000001000010101001010110011011110000101110100000101110100001010010001110100110000101011001101100001100111110110100000000110010110000010101110 0 000010010010000111111001100110111110001000100101010010110001010100100110100010100001010110001110110101111111101 101#0
26	1110000001000101000000101010001010011011001000011111101011111001100001100010010101110001110110000000111110001010101111000110110011010101001001011 00 0110010110001000110010000100000001000111110000011110110001001011111100100100111000010110011110101000110100111 0110#0
27	1100010000101011010111100010111001100100100000000101100100011101110000000101001000011010111000000010010000110100000010110001010010110100101001001 00 0011110011010011001111011111101010101101010010111100000110010111100010000000101010110101001100000100000000001 100110#0
28	1000101000100000011111110101011001011011000000011101110100010110011110111111100110000000101100100001011100110100000001000101100100100010011000 00 0100011001111100100101110010110101000100011101111111000111010101011010110000111011111000100110011000011111110001 110#0
29	0100110100111011010100010101010100100010011110100110100010111100000000100001000000111111011000000000000100100100000101010100110011001010100110011 0011001111100111110000100001010001001100110100011010100110010000100110101011011100111000011011011101111111010 1100#0
30	0110011101000111100101010011101110011100010001100010111110100000101010011001011010010101110000010111100000000001010000111101101111010011011001001 01100011110101100000110011001111011010001000010010000000111100110000110001001101010110110011111011010000011010 10110#0
31	1001011100010010010000000100010111011100000101001010010010010100001101001010100101100110111011010110010111111001110001000101101110111100001001 11 0111011111001110101011111011011000110001101000010101110010011101110100011101100100011111111000010001000111101010 01#0
32	1110001001010010011000101000101010010111000001100000000100110111111000100111001000000010011000110100111010100100100101101101100111100001010000 01 0110000111011100000000111011111001111110011110001001001100010110101100101011111010110010110101111001011000111010000 1#0
33	0111010001101010110111011110111000100000010010011111001111111000011011001100000111001001010110001101001111101110001001100101101111011101001110100 0011111011110000011100101010010011001100000010011000010000100111001110101100001101100101001010111000100010011101 00#0
34	0110010101011101111011010101011011101100101110010101110001100011101101111001100111010111001010101011000110000111010001100101111001101110100111111 11111010001010001001100110001110111010010110110110011000010100011110011011000110111001000101100010011100001110 110#0
35	0101000101001001011111000111001001101110010010110001001011111011111110011111000101011011001111111100001110001010000011101110011101000111110001101 10001010001110110111001101110000000001001010101000100011001011011111101011111101001001111101010111101000000111 01111#0
36	1000001100101010110110000111001111011100100101100110000011100011010100100111001110110110110111011011001100000110101001010001101000000110011000 0 0100110010001010010001001110000001000100000110010010010000000010110110110101111110111010010011001010001010101010 11#0
37	1010010111010111110111100011000101110101100101010001010011111001100010101011100001100100001100110111010111001000100011001111100100101110010011 0 11010110000100010000111100100001100001001001101101011101011001111111000100110010111001100000110101001000011010 001#0
38	1100001111001111010010100110000110010101010001100110000010010011010101000000111010010011011000101011101100101000001000010110110010100010000010 0 1000100001100010111111010100000110101101010001010111110100101100010110011001101100011100110000000011000000101011 0#0
39	0001000101111101010010001000011111101110101011111110000010011000111000101110011001001010111011010110100011010010111000011111010001011010101001010 1101111000010011000000101100011101111101101101011111110100110001010100000110011100011000011011010001101110011100 00#0
40	1110111001100001111011010101011011001100001100101000011000001110011101001101111001100010011100011101000100011101110111000110111000100110110110001 01011010001111011000000101001010001101001100100000000000110111111110011000001101101111010110001110001101001111 00010#0
41	1000111100011011110110000000010110100010011011110101001001000111100111111110011001100000111100011010001111110110001001110000010111111101000010 0 10101110111010011101011111010000010101000101100011110110011111111111101100111101011010110011100011111001001111 1000#0
42	0101101000000000110111100111111101010110100100110001101110101011011000010000101100010101111000101001100100010100101110001101101001111111001100 01 00010000011111111011000001111101001011010101011000100101010110110011110111111001000000101101111000011100011111100 11#0
43	0001101110010011001001100010011110000111110010100110101100110001101100111110100010010001010010001101011100011010001100001001001101100010010101 00 11111111100001101001000110011001010110000000001101011001000010110101111110010100011011100101010000101011010001 110#0
44	0110010111101111011100100001111101100100010111110001001001011101000110101110100100000110111000110101111101111111001001110000101001000111101100 00 01010000000110111111011010110111000100100001011110011111100010100110100100100011000010110110100111101010000000 110#0
45	0110000110110001001001001111001011011101110001101001100111001000111001111011111011011001101001010100010111000110011011101000101011011010011010 0 1111000000000001011000110111101010110010000000101000011100000000011110110010010011000111011011000110001001110111 01#0
46	1100000111011010011011100101000011011000000111010111000110000100001110010001101110001010010100101000001101010101001110110000011101101011000010 1 1010110110100110101110100100001000000011001110100000101100100100001001100111100100101011010100100001011010100111 11#0
47	1010001010000011110011101111000111011110111010111110100001101100101010111011000111100110000100010110001100000011000011011011110100100001001011 0 11011110011000001011101000111010011100101100110011001010000011111111001111000010110111101000101110010101000 00#0
48	1000010100000000111010001001000110111111110000011010000100100111101000111111101000000000010101010110101101011101001011110100111100101011110010 1 1111000110011110111101011001101010110011011100111010111011000101101100101011000101111100001011001111101000100100 1#0
49	1000001011111100001111110000000111111011111000010010000100000000111001110001001111111111111101110100011100000101011000100010111101101011011101010 11 1001000011101110011101011010011111011111000001101110101001101001110101110011111001110111111110111100111110011111 1#0
50	1011110011010110000010110001111000111010111101001010100100011101010111010101001010000011010000100001100011101000100111010111111111101101011001 0 1110001010101011110001010000010111110011011010111110011010011100000000111001010101001100100011100100110000101110 01#1
51	0000010000010000011001111010011110111001000000000110101000110001011011111011011010000000101000000110100010000111001001110010010111101100100000 10 00100000001011100010010001101111100100101100010001000001111001011011101011011011111001111001101000110010011111010 1#1
52	1000010010000000000010010101111000100010110110110000000100111110111011111010110100100111100111110011100001111101111110000110001100000100011100 011 110111000011101101110101010001010010110011101010001011111101100101100001111110110011011101011011001001000110001 00#1
53	1000110010101001000100001001101000011111101011111100100000100101110000110011100101111100110100011101110101110100011001011011011001110010100110 1 0010000001011000011001010110110010111011110010001100101111100111000101000000101100101110101101011110100100000110 110#1
54	1001101110101001001111111000010100011100100000101000111101101111010111101111000110101011101000100011101010111100110110010010010011101101110 0110

	0011010110000101010000110010111001011001111011100001010101111011110000111011011011110000111010110110111111010101010#
55	0111100101111100101011110100000101010000011101011110011101111100101101110000001101000110111001011010101011010000100100111111001000010111000001011101110000110010011101001111010100111101111010111101111010111000000000101011000100101000111010111010#
56	1101011110000111000001101011100013000011001010000010100001010001001000011011110100101100100000101010000111001100011011110111111001010101111011001110001101010110111001111100011010110#
57	1001100111000000011010100000011101011000010010011010001011110000110010011111100100100011100110000011110001100000000010000001001111000000011001000001001101100000000111111000011101101100111000000100101000011010110000000000111111000011101101100111000000100101000011010110#
58	1001001001010000101011101100111001101010000100011000101101000000101001111100001101011100100100100011111111001101101110100100111110101000001101011000011001000110100010100111010100001100001111110011011011010011000011111100100010111010100001#
59	0000011001110001101110100000000100101111000101001010010001011101001111000101000001010111000110111000100101010000100011000110101101111100011111111110011010100000100100111010111011111000111111111001010100001010111000011100111100000100110111001000111000010010110111000011100111100000100110111#
60	0100001001000111111111110101000010100000000001111110000011010001000010110111010100000001100101101100001000110000001100001100000100100000000100110101100111110010000111111011100001111100101101110010010111010111000011100111100000100110111#
61	1011010000101101110000001000110100010000010000110110010100110100110010011111100101100010001010110110011001111111001000110011000110101101011100100011010011000111010100001100111000101100100011010010011000011111100100010111010100001100111000101100100011010010011000011111#
62	1100011111110001001010000000011111111000111001100001001001101010101001001101111111011011010000001001100001110011101101001001000110011111010100011010111000101100100011010010011000011111#
63	0001001011011011100001010101111011011100010101100010111000001001010001111010011001001001100000001010100001101001000101001000010011010011000010100000100110000010110010111011011100010101100100001001100000010110010111011011100010101100100001001100#
64	1010100000000010010001010000000101110000000011101100100100000100101101010110100011011001000010000101100011111100101100101010011101011010111000100001011001101010100110101100111010111011010111000100001011001101010100110101101011#
65	0110110111000100100100111000101110111001011110011000110011111010011001011011100011110011100111100110101100100111011001001100111001110110001011000111010100100100111010011011100111011000101100100111010011011100111011#
66	1001001000101011010100001110111111010000011011111011101100111010111000001001111011000011100011100001100100100111010111100000111010111110000011101001000000100101010110011000111010111111010000111100110101110000111010010000000100101010110011000111010111111010000011110011010111100001110100101111111010000111100110101110000111010010#
67	1101001100100111011101110010001100100010110100001111101100101001111000110010000111000000111001110111100100000001010000011011011000111111101100100110101101110111000111111010#
68	0110101110011111010111110000101111101010010111010101011011010100011000010110001100100010000100101101010100100011001010010101111111010001111101001110011010100111010100101011111111010001111101001110011010100111010100101011111#
69	1011101100101101010000011101100111100101110001111111101011010010110111010111000101001010110011000111011100011110111000010101101101101100100101010100000100011000111001100000000101010001011110011111010110100100111000111010110100100111000#
70	1111000000111010110010101010110011001101110110100100000111001000101010000110101001100011111000100100010011111100100010100100010100001000111100100011010010100010000111100011010010110110100101010100110111001100111000111010010110110100101010100110111001100111000#
71	1100010000111011001101100110101010000001110011010100000111010001001110010001010010010001011100101011000000011010100110001100011100100010000011001000001010101001000100101000010111010101100111010101000111000101101100100011100111010110010001000011001000001010101001000100101000010111010101100111010101000111000101101100100011100#
72	0110000111000111000101011100110010010001111101001011100011010111110111000010110011001001110101011100000100010001100100001011100010111011011011001011010111010010101011100110010100100011111010010111000111010100101010111100010001101110110101000100001000110100001011100010111011011011001011010111010010101011100110010100100011#
73	1000111100111000001010011010101100101001111010001011011111011111001100001101100001011001000000011110011111010111001010101010011100111000110010000011110111110000011111100110010111010111010011000000111101111100000111110011010111010111001010101010011100111000#
74	0010110011111010000110001101110111000000001101101000001100010011010000001100111011011010110100110110011100011101110101011111001001011111000010110011101110101011111001001011111000#
75	1000001000010101001010110011011110000101110100000101110100001010010001110100110000101011001101100001100111101101000000000110010110000010101100010110011010000000110010110000010101100#
76	1110000010001010000010101000101001101100100001111110101111100110000110001001010111000111011000000011111000101010111000110110011010100100100101100010110011001000000100011110000011100100100000010001111100000111101100010011010111110010010011100001011001100111000111010#
77	1100010000101011010111100001011100100100000001011001000100001101110000001010010000110101100000010100100011010010100101101001001000100001111001101000101100100100000100111010111010101010001011000111010#
78	1000101000100000011111101010110010110110000000111011010001011001111011111110011000000101100100001011001010000001000101100100010001100000010001011000100010001100000010001100100100010001100100100010001100100100010001100100100010001100100#
79	0100110100111011010100010101010100100010011110100110100010111100000010000100000011111101100000000000100100100000101010100110011001010100110011000110011000100011000100100011000100100011000100100011000100100011000100100011000100100#
80	0110011101000111001010011101110011100010001100010111101000001010011001010110100101011100000101111000000101000011101101110100110110010011010001110111011010011011001001101000111101110111010011011001001101000111011101110100110110010011010001110111011101001101100100110100011101110111010011011001001101000111#
81	1001011100010010010000000010001011011000001010010010010010000110100101010010110011011101011001011111010010000100010111010110111011100001001110111011100010010111011101001011111100000100111011101110111010011011100100010111011101001#
82	1110001001010010011000101000101010010111000001100000000100110111111000100111001000000100110001101001110101001001001010110110110011101110000101000001011000011101110000000011101111001111100111000100100110111010100101011011100101100011101000001#
83	0111010001101010110111011110110000100000100100111110011111100001101100100000111001001010110001010011110111000100110010110111011101001110100001111101111011000110000111001010100100110011100000100100011110011111011100000#
84	0110010101011101111010101011011101100101101110001011000011101111001100111010111001010101100011000011101000110001101101001101110100111111111101000100010010011001100011101110111010011010001011011010010111011010010110111010001111#
85	0101000101001001011110001110010011011001001010110001001011111011111100111110001010110110011111110000111000101000001110111001101000011110001101000010010011100010100011101101110011011000000100101010010000100011001010110111011100000100110111#
86	1000001100101010110110000111001110111001001011001100000111000101010010011001110110110110111011010011000001101010010100011010000001100110000010010100010001100100010100100100010001100100#
87	1010010111010111110111100010110101100101001000101001111100110010101011100001100100001100110111010111100100001000110011111001001011100100110101000101100111010010111000100100010000111001000011000010010011010110111010010111010010111000100100010000#
88	1100001111001111010010100110000110100100100110011000001001001100001010000011101001001011000101011110010010000001000010110100100100010000100#
89	0001000101111010100100010000111111011101010111111000001001100011100010111001100100101011101101010100010100101110000111101000101101010100100101011010111010011000011111010001011010101001010110111000010011000000111111101100000111101001011010111111010000010100100101100011101100111010111010011010111111010000010100100101100011101100111010111010011010111#
90	1110111001100001111011010101011011001100001100101000011000001110011101001101111001100010011100011010000100011101111001101110001001101101100010101100100110100111010101000101000101000110100111010101000101000110100111010101000101000110100111#
91	1000111100011011110110000000010110100000100110011000001001011110010010001110011111111110011001100000111000101001011110010010000010001011010010010000100000100010000100011000101111110101000001010100010110001110110010111111010000010010110111010011010111110100000010100100101100011101010111101000001010100010110001110110010111111010000#
92	0101101000000000110111100111111010101101001001100011011010101101100001000010100010101110001010011001000101001011100011010100111111001100010001000110001111111101000001111101010101101100010010000101111011010010111001100100101010111000110101001111110011000100010000011111111010000111110100101101010111000100100001010111010010111001100100101010111000110101001111110011000#
93	0001101110010011001001100010011110000111001010011010110011000101101100111101000100100010100100011010111000110100010000100100110110001001010011111111100000101001000110010111010101011100010010000101111010010111001100100101010111000110101001111110011000#
94	0110010111101111011100100001111101100100010111110001001001011010001101011111001010000011011000101011111011111100100110000101001000111101100000101010000001101111110110101101110001001000010111011100111000010100100011101100000#
95	0110000110100001001100100111000101101111000110011000011001110000111101111011011001010100001100111000110101100101101101001001011010110100110010011110000000000101100011011101010110010000001010000111000000001011101100100100011000101101101001001011010110100110010#
96	1100000111010100101110010100001101100000111010111000110000100001110010001101110001010010100100000110101010100111011000001110110101100000101000000100101010110011000111010111010011000000100101010110011000#
97	1010001010000011110011101111000111011110111010111101000110110010101011101000111100110000100010110011000000100001101101111010010000100101100101000000100101100101000000#
98	1000010100000000111010000100100011011111111000001010000100100111101000011111110100000000010101010110101101100101111101001111000101011100100111101000101111001011101000010111010011010110111010011010101100111010011010101100111010011010101100#
99	1000001011111100001111100000011110111110000100100001000000011100111000100111111111111111011101000111000001010110001000101111011010110111011010111010101100100000111011001111001100101001111010111000001101110101001010011110101110011111011101000100100100111010111010111010101100100000111011001111001100101001111010111000001101110101001010011110101110011111011101000100100#

Referências Bibliográficas

Aguiar, M. S. e Costa, A. C. R. (2001) Autômatos Celulares para Análise da Monotonicidade da Declividade de Áreas Geológicas. In: *Anais do III Workshop Brasileiro de GeoInformática*, Rio de Janeiro: 87-94.

Bao, F. (2004) Cryptoanalysis of a Partially Known Cellular Automata Cryptosystem. *IEEE Transactions on Computers*, 53(11): 1493-1497.

Benkiniouar, M., and Benmohamed, M. (2004) Cellular Automata for Cryptography. In: *Proceedings of IEEE Conference Information and Communication Technologies: From Theory to Applications*, Algeria: 423-424.

Biham, E., Shamir, A. (1991) Differential Cryptanalysis of DES-like Cryptosystems. *Journal of Cryptology*, 4(1): 3-72.

Blackburn, S., Merphy, S., and Paterson, K (1997) Comments on Theory and Applications of Cellular Automata in Cryptography. *IEEE Transactions on Computers*, 46(5): 637-638.

Carvalho, D. B. (2000) *Segurança de dados com criptografia: métodos e algoritmos*. ISBN: 8-58-684638-4. Rio de Janeiro: Book Express.

Chaudhuri, P., Chowdhury, D., Nandi, S. e Chattopadhyay, S. (1997) *Additive Cellular Automata*. ISBN: 0-81-867717-1. New Jersey: *IEEE Computer Society Press.*

Das, R., Crutchfield, J. P., Mitchell, M. and Hanson, J. E. (1995) Evolving Globally Synchronized Cellular Automata. In: *Proceedings of the Sixth International Conference on Genetic Algorithms*, San Francisco: 336-343.

ElGamal, T. (1985) A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469-472.

Ganguly, N., Das, A., Sikdar, B. K., and Chaudhuri, P. P. (2000) Cellular Automata Model for Cryptosystem. In: *Proceedings of Cellular Automata Conference,* Yokohama: 120-125.

Guan, P. (1987) Cellular Automaton Public-Key Cryptosystem. *Complex Systems,* 1:51-56.

Gutowitz, H. (1995) Cryptography with Dynamical Systems. In: E. Goles and N. Boccara (Eds) *Cellular Automata and Cooperative Phenomena.* 1: 237-274, Dordrecht: Kluwer Academic Press.

Howard M. H. (2002) A tutorial on linear and differential cryptanalysis. *Cryptologia*, XXVI(3):189–221.

Kari, J. (1992) Cryptosystem based on reversible cellular automata. Personal communication. *Apud in (Seredynski, Bouvry and Zomaya, 2003).*

Lima, M. J. L. (2005) *Criptografia Baseada no Cálculo Genérico de Pré-Imagens de Autômatos Celulares*, Dissertação de Mestrado. Programa de Pós Graduação em Engenharia Elétrica. Universidade Presbiteriana Mackenzie, São Paulo, Brasil.

Mao, W. (2003) *Modern Cryptography: Theory and Practice*. ISBN: 0-13-066943-1. New Jersey: Prentice Hall.

Matsui, M. (1994) Linear Cryptanalysis Method for DES Cipher. In: Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT '93). *Lecture Notes in Computer Science* (Springer-Verlag)*, 765:* 386-397.

Nandi, S., Kar, B. K., and Chaudhuri, P. P. (1994) Theory and Applications of Cellular Automata in Cryptography. *IEEE Transactions on Computers, 43*: 1346-1357.

NIST (1977) National Institute of Standards and Technology, Data Encryption Standard (DES), FIPS-Pub.46-3. *National Bureau of Standards*, U.S. Department of Commerce, Washington D.C.

NIST (2001) National Institute of Standards and Technology, Advanced Encryption Standard (AES), FIPS-Pub.197. *National Bureau of Standards*, U.S. Department of Commerce, Washington D.C.

Oliveira, G. M. B. (2003) Autômatos Celulares: aspectos dinâmicos e computacionais. In: R. Anido e P. Masiero (Eds) *III Jornada de Mini-cursos em Inteligência Artificial (MCIA).* 8:297-345, Campinas: Sociedade Brasileira de Computação.

Oliveira, G. M. B., Coelho, A. R. e Monteiro, L. H. A. (2004) Cellular Automata Cryptographic Model Based on Bi-Directional Toggle Rules. *International Journal of Modern Physics C*, 15:1061-1068.

Pamboukian, S. V. D. (2007) *Marcas d´água de autenticação para imagens binárias: marcas reversíveis e marcas para o padrão JBIG2*, Tese de Doutorado. Escola Politécnica. Universidade de São Paulo, São Paulo, Brasil.

Pfaffenberger, B. (1998) *Webster's new world dicionário de informática*. ISBN: 8-53-520372-9. Rio de Janeiro: Campus.

Rivest, R. L., Shamir, A. and Adleman, L. (1977) On Digital Signatures and Public Key Cryptosystems, MIT Laboratory for Computer Science. *Technical Memorandum* 82.

Sancese (1998) Criptografia: storia della criptografia Disponível em: <http://www.sancese.com/Cripto.html>. Acesso em: 22 de janeiro de 2007.

Sen, S., Shaw, C., Chowdhuri, R., Ganguly, N. and Chaudhuri, P. (2002) Cellular Automata Based Cryptosystem (CAC). In: *Proceedings of Fourth International Conference on Information and Communication Security (ICICS02)*, Singapore: 303-314.

Seredynski, F., Bouvry, P. and Zomaya, A. Y. (2003) Secret key cryptography with cellular automata. In: *Proceedings of Workshop on Nature Inspired Distributed Computing* (IPDPS2003: International Parallel & Distributed Processing Symposium), Nice: 149-155.

Shannon, C. E. (1948) A mathematical theory of communication. *Bell System Technical Journal,* 27: 379-423 and 27: 623-656.

Stallings, W. (2003) *Cryptography and Network Security: Principles and Practice*. ISBN: 0-13-091429-0. New Jersey: Prentice Hall.

Tomassini, M. and Perrenoud, M. (2000) Stream Ciphers with One- and Two-Dimensional Cellular Automata. In: Proceedings of Parallel Problem Solving from Nature (PPSN VI). *Lecture Notes in Computer Science* (Springer-Verlag), 1917: 722-731.

Wikipédia (2006) Autômato celular. Disponível em: <http://pt.wikipedia.org>. Acesso em: 20 agosto 2006.

Wikipédia (2007a) Advanced Encryption Standard. Disponível em: <http://en.wikipedia.org>. Acesso em: 31 de janeiro de 2007.

Wikipédia (2007b) Análise de frequência. Disponível em: <http://pt.wikipedia.org>. Acesso em: 14 de junho de 2007.

Wolfram, S. (1984) Universality and Complexity in Cellular Automata. *Physica D*, 10:1-35.

Wolfram, S. (1986) Cryptography with cellular automata*:* In: Proceedings of International Cryptology Conference (*Crypto '85)*. *Lecture Notes in Computer Science* (Springer-Verlag)*,* 218:429-432.

Wuensche, A. (1999) Classifying Cellular Automata Automatically: Finding gliders, filtering, and relating space-time patterns, attractor basins and Z parameter. *Complexity*, 4 (3):47-66.

Wuensche, A. e Lesser, M. J. (1992) *The Global Dynamics of Cellular Automata*. ISBN: 0-201-55740-1. New Mexico: Addison-Wesley.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo