Autômatos celulares unidimensionais caóticos com borda fixa aplicados à modelagem de um sistema criptográfico para imagens digitais

One-dimensional Chaotic Cellular Automata with fixed border applied to a cryptosystem modeling for digital images

Eduardo C. Silva [1 2]
Jaqueline A. J. P. Soares [1 3]
Danielli A. Lima [1 4]

Data de submissão: 01/03/2016, Data de aceite: 04/05/2016

Resumo: O principal objetivo da criptografia de dados é possibilitar que duas entidades se comuniquem ao longo de um canal inseguro, de tal forma que nenhuma outra entidade consiga decifrar a mensagem que é enviada. Muitos métodos de criptografia clássica já foram investigados para minimizar este problema. Uma nova abordagem para este tema são os Autômatos Celulares (ACs), atualmente estudados por sua capacidade de processar grandes volumes de dados em paralelo. Nesse trabalho é investigado um novo modelo de autômato celular para criptografia de imagens, que tem como característica o uso do cálculo de pré-imagens a partir de chaves caóticas. O modelo é denominado Border Chaotic Cellular Automata (BCCA) para cifragem. Resultados mostraram que o modelo tem grande potencial para a realização de cifragem de grandes volumes de dados.

***P*alavras-chave:** autômatos celulares, cálculo de pré-imagens, criptografia de imagens, processamento paralelo

[1]Instituto Federal de Educação Ciência e Tecnologia do Triângulo Mineiro (IFTM) - Av. Líria Terezinha Lassi Capuano nž 255, Bairro Chácara das Rosas, CEP 38740-000, Tel.:(34)3515-2100 - Patrocínio - MG, Brasil.
[2]{eduardocassiano@iftm.edu.br}
[3]{jaquelinepapini@iftm.edu.br}
[4]{danielli@iftm.edu.br}

Abstract: The main objective of data encryption is allowing two entities to communicate over an insecure channel, such that no other entity can decipher the message that is posted. Many classical methods of cryptography have been investigated to minimize this problem. A new approach to this theme are the Cellular Automata (CA), currently studied for their ability to process large volumes of data in parallel. In this work is investigated a new model of cellular automata for encrypting images that uses calculation of pre-images from chaotic keys. The model is denominated Border Chaotic Cellular Automata (BCCA) cipher model. Results showed that the model has great potential for performing encryption of large data volumes.

***K*eywords:** cellular automata , pre-image calculus, image encryption, parallel processing

1 Introdução

Com o aumento dos sistemas de informação conectados à rede mundial de computadores, e posterior popularização de equipamentos eletrônicos para a captura de imagens digitais, tornou-se mais frequente a troca de dados entre entidades, especialmente a troca de imagens em redes sociais privativas ou correios eletrônicos. Além disso, muitas imagens são capturadas a todo instante por satélites artificiais e as imagens obtidas pelos mesmos devem ser transmitidas eletronicamente. Assim, grande parte dos dados que estão sendo transmitidos devem ser protegidos, pois na maioria das vezes esses dados são confidenciais. Adicionalmente, o mecanismo que realiza a proteção desses dados deve ser rápido o suficiente para que esse procedimento seja factível de ser realizado dentro de um tempo de comunicaçao pré-estabelecido. A ferramenta mais significativa para realizar essa tarefa é a criptografia.

O principal objetivo da criptografia de dados é possibilitar que duas entidades se comuniquem ao longo de um canal de transmissão, de tal forma que nenhuma outra entidade consiga decifrar a mensagem que é enviada. A criptografia também é estudada como abordagem para minimização da vulnerabilidade de dados e recursos, bem como para a garantia de confidenciabilidade, integridade e autenticidade durante a transmissão de dados. Esse campo tem sido muito estudado atualmente devido à necessidade constante de transmissão de informações e por serem importantes, na maioria das vezes, devem ter um tratamento especial. Os algoritmos de criptografia clássica existentes não tratam de forma adequada a grande quantidade e redundância de dados, que são características inerentes às imagens. Esse fato deve-se a pouca disponibilidade de núcleos que os dispositivos de baixo custo apresentam. Além disso, os principais algoritmos de criptografia simétrica AES e DES são sequenciais, dificultando o processamento massivo de dados Daemen [4], Zeghid [29]. Isso motivou a criação de um novo campo na criptografia que estuda algoritmos para a cifragem de imagens Yu [28]. Esse campo consiste na melhoria de algoritmos clássicos Prasad [19], na tentativa de paralelizar algumas etapas custosas ou redundantes do processo de cifragem Le [10].

Uma ferramenta útil na elaboração de sistemas de cifragem pode ser estabelecida por modelos matemáticos conhecidos como Autômatos Celulares (ACs). Existem diversas aplicações no uso de autômatos celulares, dentre elas podemos citar, a modelagem de sistemas naturais ou biológicos Lima [11], Zhang [30], físicos Feliciani [6], Castro [3] e até mesmo na proposição de sistemas de controle de robôs Ferreira [7], que seriam muito difíceis de serem modeladas pelas equações diferenciais, sendo estas as mais utilizadas nesse tipo de tarefa Wolfram [26]. Duas das principais vantagens em se aplicar modelos baseados em ACs reside na sua simplicidade de implementação e também na sua capacidade de processar dados em paralelo Vasantha [24].

Neste trabalho, um novo modelo de criptografia para imagens denominado BCCA (Border Chaotic Cellular Automata), baseado em ACs unidimensionais caóticos é investigado. Essa técnica utiliza o cálculo de pré-imagens (evolução do AC para trás) no processo de cifragem, a exemplo de outros métodos investigados na literatura Gutowitz [9], Lima [12] e Oliveira [17, 18, 15]. Para evitar o problema do aumento de bits apresentado no trabalho de Gutowitz [9] e Oliveira [17], neste trabalho, foi utilizada uma borda fixa para bloquear o crescimento de bits acelerado. Além disso, temos a garantia de que todo texto plano pode ser cifrado, diferentemente do trabalho de Oliveira [18], e de que o procedimento aqui empregado é totalmente paralelo, o que não foi observado no trabalho de Oliveira [15]. Adicionalmente, para a comprovação de que o sistema BCCA é forte contra ataques de criptoanálise, testes clássicos em criptografia foram realizados para validar o modelo aqui investigado, tais como, histograma de cores Prasad [19], análise de perturbação e entropia Stinson [23].

2 Fundamentação Teórica

Esta seção apresentará as principais definições para o entendimento do método de criptografia BCCA. Primeiramente, será apresentado uma definição sobre autômatos celulares unidimensionais e a propriedade de regra denominada sensitividade. Na sequência, os principais trabalhos sobre ACs aplicados à criptografia são apresentados e discutidos.

2.1 Autômatos celulares

Um AC é composto por um reticulado com uma dimensão d dividido em células ou unidades processadoras, sendo que, cada célula C é representada por um estado. As células modificam seus estados a cada passo de iteração de acordo com uma regra de transição. Podemos aplicar a regra de transição por T passos de tempo para obter a evolução espaço-temporal do reticulado do AC. A regra estabelecida por uma função de transição indica o novo símbolo a ser escrito na célula do reticulado de acordo com seu estado atual e dos estados de suas vizinhas (regra local). Em sua definição mais usual, a atualização dos estados se dá de forma síncrona e utiliza uma regra determinística, isto é, a cada passo de tempo todas as N

células do reticulado são atualizadas Castro [3].

A estruturação de um AC unidimensional (1D) é a forma mais estudada. Para um AC com regra de atualização determinística, a mudança de estado de uma célula depende de m vizinhas expressas por $m = (2r + 1)$, sendo r o raio do AC Oliveira [16]. Para ilustrar um AC unidimensional com regra de atualização determinista, considere a Figura 1 (a), que aborda uma modelagem conhecida como regra 30 Wolfram [26], contendo um reticulado de 6 células sendo que o estado inicial de cada célula é apresentado em $t = 0$. Uma regra binária de raio 1 é aplicada, sendo que a vizinhança de cada célula é formada por três elementos: a própria célula e suas duas vizinhas adjacentes (à esquerda e à direita). Como esse AC é binário (2 estados possíveis), existem 8 diferentes vizinhanças, da 000 a 111. A regra em si é dada pelos 8 bits de saída associados a cada vizinhança possível: 01111000. Na Figura 1 (b) vemos a atualização do reticulado (evolução para frente) de $N = 6$ por 2 passos de tempo a partir de sua configuração inicial 101110 em $t = 0$. A cada passo, cada célula do reticulado é atualizada identificando-se sua vizinhança e seu novo estado é dado pelo bit de saída correspondente na regra de transição. Observe como exemplo, a célula de símbolo 1, destacada em $t = 0$, seu próximo estado será 0 em $t = 1$. Essa configuração é dada porque a vizinhança 110 de acordo com a regra de transição retorna 0 ($110 \rightarrow 0$), assim a célula central é atualizada para o estado 0. O reticulado é submetido a condições periódicas de contorno, sendo que a primeira célula é vizinha imediata da última, e vice-versa. Aplicando-se esse procedimento para todas as células do reticulado de forma síncrona por T passos de tempo, tem-se a nova configuração do reticulado a cada passo de tempo.

Algumas propriedades estudadas nos ACs são exploradas em criptografia, sendo uma

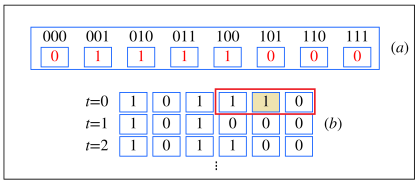

Figura 1. (a) Regra de transição de raio 1. (b) Evolução do AC por *T* = *2* passos de tempo.

dessas propriedades a sensitividade da regra. A sensitividade existe em uma regra quando a alteração de um bit extremo da vizinhança provoca necessariamente alteração do bit de saída. Se a alteração for feita no extremo esquerdo da vizinhança e esta provocar alteração nos bits de saída, temos sensitividade à esquerda. Senão temos sensitividade à direita. A regra da Figura 1 (a) apresenta sensitividade à esquerda; por exemplo, a vizinhança **0**00 leva o estado da célula central a 0, enquanto a vizinhança **1**00 leva a 1. Em todos os quatro pares de vizinhanças similares da regra, diferenciadas apenas pelo primeiro bit da regra, a saída é complementar. Dessa forma, a regra da Figura 1 é sensível à esquerda. O método de criptografia investigado neste trabalho, faz parte de uma família de métodos que utilizam

ACs com regras sensíveis e o cálculo de pré-imagem na etapa de cifragem Gutowitz [9] e Oliveira [17, 18, 15] e Lima [12].

2.2 Criptografia Baseada em Autômatos Celulares

O primeiro trabalho conhecido envolvendo o uso de ACs para a realização da criptografia foi proposto por Wolfram [25]. Nesse modelo precursor, a regra de transição é fixa e apresenta dinâmica caótica Wolfram [26]. A chave é utilizada como reticulado inicial a partir do qual a regra é aplicada por um número fixo de passos. Os modelos que utilizam ACs na criptografia com regras sensíveis foi explorado em Sen [22], no entanto o paralelismo e a segurança desses modelos é limitada, devido ao fato da propriedade aditiva das regras, que não as tornam caóticas. Sistemas criptográficos de boa qualidade devem retornar cifras embaralhadas e caóticas Wolfram [26], Machicao [14], dificultando a criptoanálise.

Para melhor compreensão dos algoritmos baseados no cálculo de pré-imagens de ACs algumas definições serão introduzidas. O texto plano $\mathcal{P}$ apresentado neste trabalho refere-se a qualquer sequência de bits dada como entrada, um texto ou uma imagem binarizada. Esse texto plano $\mathcal{P}$ refere-se ao reticulado L inicial do AC. O texto cifrado $\mathcal{C}$ é a sequência de bits final encontrada (borda, bits extras e o próprio reticulado L' final), após o cálculo de pré-imagens baseado em ACs através da chave criptográfica $k \in \mathcal{K}$ por T passos de tempo. O conjunto de chaves criptográficas refere-se às regras de atualização do AC. Neste caso, $\mathcal{K}$ é o conjunto de todas as regras com sensitividade a um dos extremos da vizinhança e caóticas (alta entropia). O processo de decifragem refere-se à evolução para frente do AC sobre $\mathcal{C}$ com a chave k por T passos de tempo.

A criptografia baseada no cálculo de pré-imagens de ACs começou a ser estudada depois que alguns pesquisadores observaram que após as células de um reticulado serem submetidas a T passos de tempo, essas poderiam resultar em uma sequência caótica de configurações. A evolução de um AC pode ser realizada com a aplicação direta da função de transição (evolução para frente) ou a partir do cálculo da pré-imagem (evolução para trás). A evolução para frente (*forward*) é obtida através da configuração inicial do reticulado no instante t, após aplicar a regra de transição por um passo de tempo, obtém uma nova configuração de reticulado no tempo $t + 1$. Esse procedimento pode ser repetido por quantos passos de tempo forem necessários. A evolução para trás (*backward* ou cálculo da pré-imagem) é obtida a partir de uma configuração inicial do reticulado L no instante t, a finalidade é descobrir qual reticulado L' no instante $t - 1$ pode dar origem ao reticulado L no instante t, após aplicar a regra de transição. Também nesse caso, esse procedimento pode ser repetido por T passos de tempo. Se um AC tem exatamente uma pré-imagem para todos os reticulados possíveis, então esse é um AC reversível. A grande vantagem de se utilizar o cálculo de pré-imagens baseado em ACs reside no fato de que o procedimento é paralelizável, quando existe uma unidade de processamento paralela para o cálculo de cada célula Anghelescu [1].

Um dos primeiros modelos que empregam o cálculo de pré-imagem com regras sensí-

veis e com dinâmica caótica foi proposto em Gutowitz [9]. Nesse modelo, a regra de transição corresponde à chave criptográfica, o texto original define o reticulado inicial e o cálculo de pré-imagens consecutivas corresponde à etapa de cifragem. Enquanto a decifragem é feita utilizando-se a evolução tradicional do AC (forward). Para a garantia de existência de pré-imagem para qualquer reticulado, regras com a propriedade de sensitividade são empregadas. Entretanto, esse método gera um texto cifrado maior que o texto plano. O método de cifragem proposto por Gutowitz [9] será detalhado e o mesmo está ilustrado na Figura 2. Para demonstrar esse método, será utilizada a regra 30 (chave) com sensitividade à esquerda. A cada passo de tempo, será necessário iniciar os $m - 1$ bits à direita do novo reticulado (representado na Figura 2 como células de coloração azul). Uma vez que os bits tenham sido inicializados, começa a etapa de evolução dos bits através da regra de transição juntamente ao valor da vizinhança gerada, preenchendo de forma síncrona e paralela todas as demais células posicionadas à esquerda no reticulado. Além disso, pode-se iniciar paralelamente o preenchimento de instantes de tempo $t + i$, com $i > 0$, desde que os bits dependentes de $t + i$ já tenham sido calculados. Esse procedimento acelera muito o processo, pois cada célula é calculada de forma independente das demais. O preenchimento baseia-se no seguinte processo: dada uma vizinhança $?01 \rightarrow 1$ ($?$ é o bit vermelho, 01 são os bits em azul e $\rightarrow 1$ é o bit verde) no passo $t = 1$, anexa-se na célula adjacente à cadeia já calculada, o bit de vizinhança extremo à esquerda que provoque como saída o valor 1 em $t = 0$. Logo, o valor encontrado no exemplo será 0, pois a regra $001 \rightarrow 1$ (isto é, 001 é a vizinhança que gera o valor de saída 1). O cálculo de pré-imagem será executado até alcançar o número de passos de tempo T estipulado. Uma desvantagem do modelo é a propagação da perturbação, que neste caso aconteceu apenas do lado da sensitividade da regra. Sabe-se que um bom método criptográfico deve propagar a perturbação ao longo de todo o reticulado.

Posteriormente, em Oliveira [17] também é apresentado um modelo que aumenta o

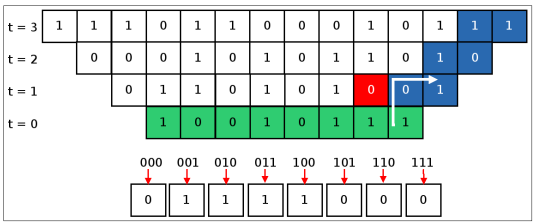

Figura 2. Metodologia aplicada à cifragem de regra sensível à esquerda no modelo de Gutowitz [9].

tamanho do reticulado. No entanto, neste modelo a propagação da perturbação dos bits é propagada ao longo de todo o reticulado por utilizar regras com sensitividade bidirecional, diferentemente do modelo de Gutowitz [9]. Em Lima [12] e Oliveira [17], uma abordagem que preserva o tamanho do reticulado, tornando o texto cifrado do mesmo tamanho que o

original, foi investigada a partir do algoritmo proposto em Wuensche [27]. Contudo, ela tem a desvantagem de que nem todo texto fornecido como entrada pode ser criptografado. No trabalho de Oliveira [15], uma abordagem que faz uso do aumento do texto plano apenas quando é necessário foi elaborado. No entanto, esse modelo apresenta a desvantagem do paralelismo ser quebrado devido à utilização de pilhas de armazenamento global e local que guarda as possíveis pré-imagens. Outros modelos foram posteriormente estudados nessa temática Macedo [13], Barros [2], no entanto, não são estratégias puramente baseadas no cálculo de pré-imagens altamente paralelizáveis de ACs.

No modelo apresentado neste trabalho, a adição de bits extras se faz necessária, no entanto, a quantidade necessária é menor que nos trabalhos de Gutowitz [9] e Oliveira [17], e todo texto de entrada retorna uma pré-imagem válida, o que não acontece no trabalho de Oliveira [18]. Além disso, o método BCCA proposto neste trabalho é altamente paralelizável, diferentemente do modelo de de controle por memórias do tipo pilha em Oliveira [15].

3 Modelo Proposto

Nesta seção serão apresentadas duas abordagens para a cifragem de imagens a partir do método BCCA. A primeira refere-se ao modelo geral de cifragem e aplica-se diretamente às imagens preto e branco no formato *.pbm* (Portable Bitmap Format), onde cada pixel (bit) representa uma cor diretamente. Ou seja, a entrada é uma sequência de bits, onde o bit 0 representa a cor branca e o bit 1 representa a cor preta. Em seguida uma abordagem para a quebra de blocos e embaralhamento em imagens coloridas no formato *.ppm* (Portable Pixmap Format) no padrão RGB é utilizado. Essa quebra em blocos se faz necessária porque as imagens nesse formato apresentam o sistema de cores no padrão RGB formando-se blocos muito grandes. Após o embaralhamento e transformação em blocos menores de bits, a cifragem é realizada através do algoritmo BCCA.

3.1 Abordagem para cifragem de imagens preto e branco

O método de cifragem BCCA é um método para cifragem de blocos unidimensionais de bits. Uma imagem preto e branco é uma matriz de bits, onde cada linha é uma sequência de bits. O BCCA consiste na evolução para trás (*backward*) do AC e é obtido a partir de uma configuração inicial do reticulado L, que é considerado o texto plano $\mathcal{P}$, no instante t, a finalidade é descobrir qual reticulado L' no instante $t-1$ pode dar origem ao reticulado L no instante t, após aplicar a regra de transição, que representa a chave criptográfica. Esse procedimento pode ser repetido por T passos de tempo resultando num padrão caótico (texto cifrado $\mathcal{C}$). Para evitar que o texto cifrado aumente de tamanho Gutowitz [9], neste trabalho foi proposta a criação de uma borda fixa, que é a responsável por limitar esse crescimento de bits. Qualquer sequência pseudo-aleatória pode ser utilizada na composição dessa borda,

inclusive a sequência proposta por Wolfram [26]. No entanto, neste trabalho, a borda refere-se à sequência de bits que corresponde a metade da chave criptográfica k de $r = 1$. A Figura 3 apresenta a chave criptográfica dada por $\{01111000\}$ e a borda b em azul escuro $\{1000\}$, que é a metade dessa chave criptográfica. Se mais bits precisarem ser enviados, eles serão concatenados à sequência de bits da borda b. O próximo bit a ser concatenado é o bit 0 e a borda torna-se 01000. A borda deve ser armazenada e enviada junto ao texto cifrado.

O processo de cifragem de uma linha $L = \{10010111\}$ da imagem preto e branco com $N = 8$ pixels está apresentado na Figura 3. Para detalhar o procedimento, um exemplo será apresentado para a atualização do bit em vermelho no tempo $t = 1$, considerando-se o preenchimento do bit ?, de acordo com a seguinte vizinhança $?01 \rightarrow 0$. A partir da regra de transição utilizada como chave criptográfica, temos que $101 \rightarrow 0$. Assim, o primeiro bit é atualizado e todos os demais bits também o são, partindo-se da mesma definição. Para a aplicação desse método em imagens, essas devem ser quebradas em blocos unidimensionais. O blocos são lineares e representam uma linha ou uma coluna inteira de bits (pixels) da imagem. Assim, o método pode ser aplicado em cada bloco da imagem por T passos de tempo. O custo de processamento P_{CB} para a realização da cifragem de um bloco de tamanho N da imagem é realizado em $P_{CB} = T + N + 2 \times (r - 1)$ ciclos de *clock*, sendo que $N = \max(m, n)$), tal que, $N = n$. No exemplo da Figura 3 temos que $P_{CB} = 3 + N + 2 \times (1 - 1) = 11$ ciclos de *clock* se existirem T núcleos de processamento (3 núcleos). A cifragem de uma imagem de tamanho $n \times m$ é realizada com um custo de processamento de $P_{CI} = m \times (T + N + 2 \times (r - 1))$ ciclos de *clock* do computador.

Se uma cifragem a partir do AES de uma imagem com as mesmas dimensões for comparada com o modelo proposto neste trabalho, o tempo medido em ciclos de *clock* seria de $P_{CB} = (T \times (2 \times N + N^2 + N^3))$ para um bloco de linha. Para o tempo de cálculo de uma imagem teríamos, $P_{CI} = m \times (T \times (2 \times N + N^2 + N^3))$, tal que T é o número de *rounds* necessários para cifrar um bloco de tamanho N bits Daemen e Rijmen [4] - neste caso, uma linha da imagem. Assim, temos que para a cifragem da linha da Figura 3 tem-se que $P_{CB} = 3 \times (2 \times 8 + 8^2 + 8^3) = 3288$ ciclos de *clock*. Esse atraso deve-se ao fato de que todos os blocos são concatenados através de uma função $\odot$ XOR.

O método de decifragem, adotado neste trabalho, baseia-se na evolução de um AC e é realizada com a aplicação direta da função de transição na evolução para frente (*forward*) do AC junto ao texto cifrado e a borda (armazenada), por T passos de tempo. Dessa maneira, o texto plano é obtido novamente. O custo de processamento P_{DB} para a realização da decifragem de um bloco de tamanho N da imagem é realizado em $P_{DB} = T$ ciclos de *clock*, sendo que $N = \max(m, n))$ e vamos supor que $N = n$. No exemplo da Figura 3 temos que a linha de tamanho N seria decifrada em $P_{DB} = 3$, se existirem 8 núcleos de processamento - um para cada célula do reticulado N. A decifragem completa de uma imagem de tamanho $n \times m$ é realizada com um custo de processamento de $P_{DI} = m \times T$ ciclos de *clock* do computador. Para efeito de comparação, o algoritmo AES levaria a mesma quantidade de tempo da cifragem para decifrar uma imagem de tamanho, ou seja, $P_{DI} =$

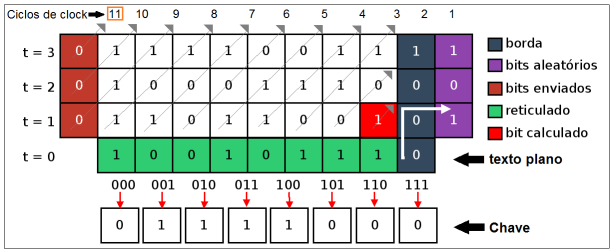

Figura 3. Exemplificação do processo de cifragem do método proposto.

$m \times (T \times (2 \times N + N^2 + N^3))$ ciclos de *clock*. Assim, o algoritmo AES levaria para cifrar uma linha $P_{CB} = 3288$ ciclos de *clock*. Portanto, a investigação da abordagem paralela baseada em ACs faz-se necessária, uma vez que reduz significativamente o tempo de cifragem e decifragem, quando implementado num hardware paralelo. Isso significa que se existir um hardware capaz de processar cada célula do AC ele vai levar menos tempo para cifrar que o algoritmo AES, uma vez que o algoritmo AES possui diversas partes (rounds) que são dependentes entre si e sequenciais Daemen e Rijmen [4].

3.2 Abordagem para a cifragem de imagens coloridas

A aplicação de uma etapa de pré-processamento no modelo para as imagens coloridas *.ppm* antes da aplicação do modelo de criptografia BCCA aqui proposto, se faz necessária, uma vez que a codificação empregada em tais arquivos é mais complexa do que imagens em preto e branco do tipo *.ppm*. Esse fato é verificado uma vez que o algoritmo BCCA, proposto neste trabalho é capaz de cifrar apenas sequências de bits. Lembrando que qualquer sequência de bits poderia ser cifrada através do algoritmo proposto, incluindo-se arquivos de voz, de vídeos ou de texto. Como o objeto de estudo para validação do algoritmo BCCA escolhido foi as imagens digitais coloridas, então, uma pré-processamento foi necessário para atingir uma cifra segura e sem texturas na imagem cifrada, que a qualidade da cifragem.

A principal diferença entre uma imagem *.pbm* e uma imagem *.ppm*, deve-se ao fato que uma imagem colorida possui mais de um único descritor de cor para cada pixel. O esquema de cores utilizado neste trabalho é constituído do padrão RGB (*red*, *green* e *blue*), isto é, a união de três descritores de cor que variam seus valores entre 0 e 255, originando 16.777.216 possibilidades de cor para um único pixel. Utilizando-se do RGB é possível reproduzir figuras possuindo paletas de cores complexas.

Comparada à versão preto e branco, a implementação do novo modelo difere apenas na etapa de formação dos blocos para cifragem. Uma vez que um novo bloco é criado, a cifragem e decifragem é realizada da mesma maneira, ou seja, utilizando-se o algoritmo BCCA. De acordo com o novo modelo as figuras coloridas empregadas no processo recebem um

pré-embaralhamento nos descritores de cor RGB, presentes nos pixels da imagem. A técnica consiste em separar os bits de cada descritor e realizar concatenações formando novos blocos destinados à encriptação. Realizar esse procedimento eleva o nível de segurança do modelo, já que é proporcionado inicialmente um certo nível de embaralhamento, evitando zonas de texturas Prasad [19]. No processo de decifragem esse embaralhamento é realizado da mesma maneira, recuperando-se cada um dos blocos de cor novamente.

A Figura 4 representa um exemplo sobre as primeiras etapas do embaralhamento dos canais RGB. Primeiramente, divide-se a imagem original representada na Figura 4 (a) em 4 partes distintas ilustradas na Figura 4 (b). Em seguida, o procedimento extrai os canais RGB de cada um dos blocos, separando-os por tipo e consequentemente formando seções que possuem todos os bits identificadores das cores. A Figura 4 (a) apresenta o resultado do primeiro bloco da Figura 4 (c), que resultou em 3 blocos de matrizes de $32 \times 32 \times 24$ bits. Ou seja, 24 matrizes de 32×32 bits cada. Dessa forma, são formados 12 blocos, contendo todos os descritores de cor contidos na imagem.

As próximas etapas caracterizadas pela Figura 5, demonstram a fase de formação

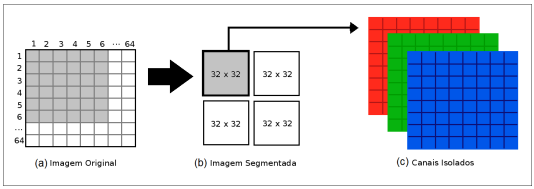

Figura 4. Primeiros passos da quebra em blocos na cifragem de imagens coloridas.

dos blocos finais que serão utilizados na cifragem. A Figura 5 (a) representa o bloco de $32 \times 32 \times 24$ bits da Figura 4 (b). Inicialmente, cada coluna de cada uma das 24 matrizes de bits de 32×32 bits são concatenadas, conforme é apresentado na Figura 5 (b). O novo bloco resultante pronto para a aplicação do modelo BCCA possui 24×32 bits. Posteriormente, o procedimento é repetido para as demais colunas, conforme apresentado na Figura 5 (c). Por fim, o procedimento é repetido para a última coluna, conforme é apresentado na Figura 5 (d). Assim, tem-se 32 blocos de 24×32 que serão cifrados utilizando-se o algoritmo BCCA.

A Figura 6 apresenta um exemplo mais detalhado sobre o embaralhamento. A imagem 2×2 é segmentada em 4 partes, contendo cada uma, 3 blocos descritores de cor. Inicialmente, cada bloco apresenta o tamanho de $1 \times 1 \times 24$ bits, que servirá na formação de uma linha que será utilizado como bloco de cifragem. Na sequencia, cada descritor de cor de tamanho 8 será dividido, formando para cada pixel $3 \times (8 \times 1 \times 1)$ bits. Em seguida cada um desses bits serão concatenados à formando um bloco de $1 \times 1 \times 24$, com os bits alternados representando cada um dos descritores de canal RGB. Para detalhamento da formação do bloco do pixel vermelho, tem-se o primeiro bit do bloco representado pelo primeiro bit do canal R. O segundo bit do bloco será representado pelo primeiro pixel do bloco G. Posteriormente, o

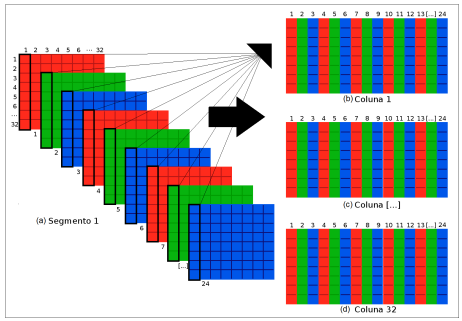

Figura 5. Método aplicado à composição dos blocos de cifragem para imagens coloridas.

terceiro bit do bloco de cifragem será o primeiro pixel do bloco B. Assim, todos os demais bits são concatenados ao bloco (linha) que será cifrado para o pixel vermelho. Em seguida, o procedimento é repetido para os demais pixels da imagem 2×2, de cores verde, azul e branco. Assim, tem-se um bloco de tamanho 4×24 bits para ser cifrado por linhas através do algoritmo BCCA.

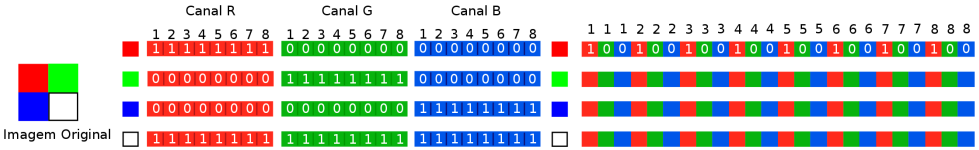

Figura 6. Exemplo de aplicação utilizando o embaralhamento proposto neste trabalho para a cifragem de imagens coloridas.

4 Metodologia para validação de resultados

Algumas das definições apresentadas a seguir serão importantes para a compreensão dos experimentos realizados neste trabalho para demonstrar que o modelo de criptografia baseado em ACs unidimensionais caóticos BCCA aqui apresentado é seguro à ataques de oponentes. Os métodos aqui apresentados fazem parte dos métodos clássicos para validação de modelos criptográficos, dentre eles, entropia, análise de propagação da perturbação e histograma de cores. Toda a implementação dos métodos de criptografia foi realizada de maneira sequencial, uma vez que neste trabalho queremos apenas testar a qualidade de cifragem final. Uma implementação paralela, apenas reduziria o tempo da cifragem, mas não o

resultado final da qualidade da imagem cifrada. Assim, testes de tempo de uma implementação sequencial foram realizados apenas para contrastar o tempo de cifragem de uma imagem colorida e de uma imagem preto e branco com a mesma quantidade de pixels.

4.1 Entropia

Definir através de uma operação matemática a capacidade de confusão presente em uma regra de transição é a principal tarefa do cálculo da entropia. O procedimento baseia-se no mapeamento de janelas contidas em uma determinada cadeia binária (neste caso, a regra de transição). Estas, são abstrações que representam através de seus segmentos cada um dos possíveis valores presentes na cadeia analisada. Com o levantamento da quantidade de manifestações de cada uma das janelas, é possível computar um valor de maneira a definir o grau de confusão (caótica) contido nas evoluções de um AC em conjunto com a regra analisada.

Por exemplo, dada a cadeia binária 01111000 correspondente à regra 30, o tamanho total das possíveis janelas j será obtido por $j = log_2 8$. Em seguida, será executado o mapeamento em busca da quantidade de ocorrências de cada janela. Nesta etapa a regra assume uma composição ligando início e fim, caracterizando uma estrutura em anel, que em função do valor de j, alguns elementos da cadeia deverão se conectar a sua outra extremidade para completar o conjunto imposto, consequentemente alcançando todas as ocorrências possíveis. A Figura 6.5 ilustra o mapeamento com ênfase na busca das janelas analisando o penúltimo e o último elemento, Figuras 7 (a) e 7 (b), respectivamente.

O valor da entropia é dada por $S = \sum_{j=1}^{2^j} \frac{k}{2^j} log_2 \frac{k}{2^j}$, com j sendo o tamanho da

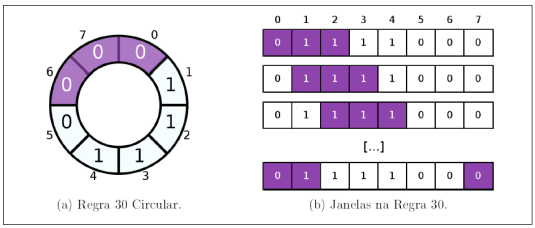

Figura 7. Mapeamento da quantidade de janelas na regra 30 calculado através da entropia s para a verificação da confusão de uma chave criptográfica.

janela e k representa a quantidade de vezes que cada janela aparece na regra. Assim, para o exemplo da Figura 7 temos que $S = 2.5$, pois a janela 000 e 111 aparece 2 vezes, as janelas 001, 011, 100 e 110 aparecem 1 vez e as janelas 010 e 101 não apresentam nenhuma ocorrência. Para promover a normatização dos valores entre 0 e 1, divide-se o valor obtido em $S = 2.5$ pela delimitador da dimensão das janelas ($j = 3$), obtendo o valor de entropia $s = 0.8333$. Quanto mais próximo de 1 estiver o valor de s, maior será o grau de confusão

presente na regra de transição.

Definir a entropia de uma chave é um importante classificador utilizado neste trabalho, uma vez que regras com alta entropia apresentam maior grau de embaralhamento. Logo, regras que possuem baixa entropia não serão interessantes no uso da criptografia baseada em ACs e posteriormente, não farão parte dos experimentos futuros. Dessa forma, o modelo contemplará um conjunto de chaves pré-filtradas, com padrões com maior entropia, onde espera-se obter resultados mais significativos.

4.2 Análise de propagação da perturbação

O nível de embaralhamento ótimo do texto final cifrado é aquele que apresenta média $\bar{x} = 0.5$ de bits 1 (ou bits 0). Ou seja, num texto cifrado ótimo é necessário que se metade dos pixels seja formada por 0s e a outra metade seja formada por 1s. No entanto, essa análise não é suficiente porque os 0s e 1s podem estar agrupados em uma porção da imagem. Assim, a análise da propagação da perturbação se faz necessária. O diagrama ilustrado na Figura 8 representa o experimento da propagação da perturbação. A base do teste é formada por

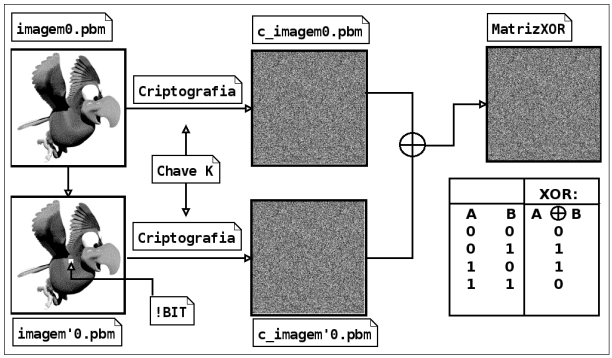

Figura 8. Método de teste denominado propagação da perturbação.

duas imagens idênticas, exceto pela segunda possuir o valor de seu bit central alterado. Em seguida, ambas as figuras passam pelo processo criptográfico empregando a mesma chave e formando respectivamente seus textos cifrados. Posteriormente o experimento utiliza o operador lógico XOR entre as cifras geradas, originando uma matriz que contempla a análise visual e comparativa entre as cifras. A comparação efetuada pelo XOR consegue identificar bits que sofreram modificações no processo de cifragem. Para a avaliação foi utilizado a contagem de 0s na imagem resultante após a aplicação da função XOR. Um cifra de boa qualidade é aquela que apresenta média de 0.5 de 0s e 1s.

4.3 Histograma de cores

Um histograma de uma imagem digital colorida é um gráfico que representa os valores das tonalidades de cores dos pixels nessa imagem. O histograma mostra o número de pixels da imagem com uma cor particular e representa os números no gráfico. Um histograma RGB mostra os três canais combinados, mas um canal de cor individual pode confirmar quais as cores específicas que apresentam picos numa imagem. A escala de cor estende-se ao longo do eixo x horizontal e vai de 0 a 255, a gama da escala é relativa a uma profundidade de cor de 8 bits para cada um dos canais de cores do padrão RGB. O eixo y vertical é o número de pixels contidos numa tonalidade particular. Quanto maior for o valor vertical, mais pixels têm uma tonalidade particular. Em criptografia, dada uma imagem num padrão de cores qualquer, é esperado que ao longo do processo de cifragem seja obtido um histograma de canais individuais equilibrados. Ou seja, se todos os três canais (vermelho, verde e azul) mostrarem gráficos com picos no mesmo ponto com os canais de cores individuais, isso indica que o equilíbrio de cores está corretamente definido e que o algoritmo foi capaz de misturar o padrão de cores da imagem original. Por outro lado, se existir uma variação significativa entre os canais, então é necessário proceder a algum ajustamento para obter o equilíbrio correto.

5 Experimentos e análise de resultados

Esta seção apresenta os experimentos realizados para as imagens preto e branco e coloridas através da cifragem de imagens de diversos tamanhos pelo algoritmo BCCA. As imagens coloridas recebem um tratamento especial, denominado neste artigo de pré-embaralhamento. Além disso, um estudo sobre a qualidade das chaves (regras) é apresentado através do estudo da entropia em Oliveira [15]. Adicionalmente, um estudo elaborado para a qualidade das imagens cifradas é realizado através da metodologia denominada análise da propagação da perturbação. As imagens coloridas também foram avaliadas através do histograma de cores, uma métrica muito importante na definição da segurança dos métodos de criptografia. Por fim, uma análise do tempo de cifragem é analisado para verificar o quanto os parâmetros do modelo afetam o tempo de processamento.

Além disso, é importante considerar que o algoritmo proposto foi implementado na Linguagem C padrão de programação de maneira sequencial, e o Sistema Operacional utilizado foi Debian 7.8 Wheezy, com um Kernel $x86\_64$ Linux 3.2.0-4-amd64 com Processador Intel Core i3 CPU M 370 @ 2.399GHz e Memória RAM 3765 MB. Essa configuração de máquina foi utilizada em todos os experimentos apresentados neste trabalho.

5.1 Análise da quantidade de bits enviados

O primeiro experimento refere-se à análise da quantidade de bits enviados. Considerando-se cada pixel como um bit, os gráficos da Figura 9 (a) e (b) demonstram a quantidade de bits salvos somando os bits originais da imagem 64×64, variando-se os tamanhos de raio (tamanho da chave) por $T = 64$ passos de tempo. Os gráficos da Figuras 10 (a) e (b) apresentam o comportamento da função para cada raio (tamanho de chave) para o modelo de Gutowitz [9] e no proposto, respectivamente. A função que representa esse crescimento no envio de bits é dada por $(m \times n) + (2r \times T \times N)$, para o modelo de Gutowitz [9], sendo que $N = \max(m, n)$. No BCCA esse envio de bits é inferior e é dado por $(m \times n) + (r \times T \times N)$. A quantidade de espaço que precisa para armazenar os bits extras no algoritmo proposto é duas vezes menor que no modelo precursor proposto por Gutowitz [9] e Oliveira [17]. Apesar do novo modelo reter parte dos bits envolvidos no processo, a quantidade total é inferior à apresentada modelo de Gutowitz [9]. Conclui-se que a melhoria proposta neste estudo interrompe a expansão do reticulado no processo criptográfico e reduz a quantidade de bits salvos no texto cifrado.

A Figura 11 apresenta uma comparação entre os modelos de Gutowitz [9] juntamente

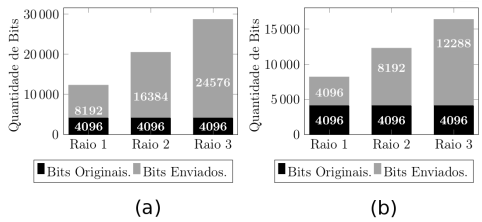

Figura 9. Número de bits enviados: (a) no modelo de Gutowitz [9] e Oliveira [17], e (b) no modelo proposto BCCA.

ao apresentado neste trabalho através da cifragem por $T = 1$ passos de tempo de uma imagem de 16×16 pixels. A Figura 11 (b) e a Figura 11 (c) representam, respectivamente, o modelo Gutowitz [9] e o algoritmo proposto. Conforme foi dito anteriormente, a imagem cifrada pelo algoritmo proposto mostra-se mais reduzida em relação ao método precursor de Gutowitz [9].

5.2 Definição de chaves para a boa qualidade do método de cifragem proposto

O segundo experimento realizado refere-se à análise da qualidade do método para a cifragem de 106 imagens preto e branco de 64×64 pixels no formato *.pbm* (Portable Bitmap Format), que corresponde à 64×64 bits. Para apresentar cifras seguras contra ataques de

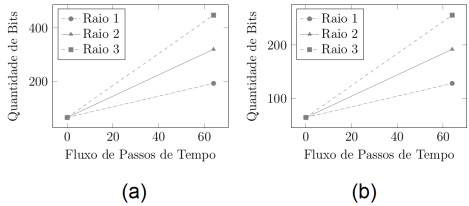

Figura 10. Gráficos das funções de crescimento do número de bits: (a) no modelo de Gutowitz [9] e Oliveira [17], e (b) no modelo proposto BCCA.

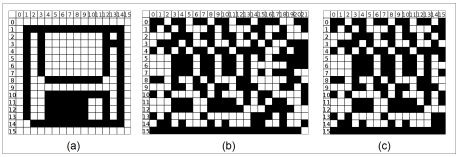

Figura 11. (a) Imagem original. (b) Imagem criptografada resultante a partir do modelo de [9]. (c) Imagem criptografada resultante a partir do modelo proposto neste trabalho.

criptoanálise, é necessário avaliar o embaralhamento do texto cifrado em relação à qualidade das chaves utilizadas na cifragem. Para que um método de criptografia seja seguro, é necessário que o mesmo apresente média de $\bar{x} = 0.5$ de 0s a partir da **análise de propagação de perturbação** entre a cifragem de duas imagens. A primeira imagem representa a imagem original A e a segunda imagem A' se diferencia da primeira através de um bit. Assim, bons métodos de cifragem possuem a mesma quantidade de 0s e 1s na avaliação dessa imagem XOR resultante.

Utilizando-se a entropia s em Oliveira [15], foi possível classificar as 2^{32} chaves de 32 bits que são ao mesmo tempo sensíveis à esquerda Gutowitz [9] e possuem $s \geq 0.7$, resultando em 63872 chaves que satisfazem a esses dois requisitos. Regras com $s = 1.0$ apresentam um nível maior de embaralhamento das chaves criptográficas e portanto tendem a apresentar resultados mais satisfatórios. Em Oliveira [15] foi mostrado que chaves com entropia inferior à $s < 0.7$ não apresentam bom desempenho na cifragem de imagens. A Tabela 1 exibe o resultado referente à cifragem das 106 imagens de 64×64 bits cifradas pelas 63872 regras de raio 2 (chaves de 32 bits) com sensitividade à esquerda, tal que, $0.7 \leq s \leq 1.0$. A faixa de valores da entropia foi dividida em classes para melhor compreensão dos resultados. A média de 0s após a aplicação da **análise da perturbação** está próximo à $\bar{x} = 0.5$ em todas as análises de classes de regras, o que indica que o método pode ser avaliado como caótico, justamente, o que é esperado em algoritmos de criptografia. Além disso, foi mostrado que o

desvio padrão em todos os resultados é inferior à 0.05 em todos os casos, exceto para chaves com entropia mais baixa (Classe 1), onde o resultado do desvio padrão em relação à média dos experimentos foi de 0.1. A variância segue esta mesma análise, ou seja, quanto menor o valor da entropia, maior é a variância em relação à média encontrada e à medida que o valor da entropia aumenta, essa variância diminui. A Tabela 1 também apresenta a variância em relação ao valor ótimo desejado (0.5 de 0s) e o desvio padrão em relação ao valor ótimo é apresentado e mais uma vez, mostra-se que regras com entropia mais baixas (Classe 1) tem dificuldades para cifrar textos planos. Em conclusão aos experimentos das regras sensíveis à esquerda tem-se que o experimento apresentou um bom resultado médio entre todas as classes. Outro fato relevante deve-se à alta entropia da classe 5, alcançando os melhores resultados médios dentre os presentes.

Na segunda fase de experimentos foi possível classificar as 2^{32} chaves de 32 bits que

Sensitividade	Classe	Entropia	Média	Variância	Desvio P.	Var. Ótima	Desv. P. Ótimo
Esquerda	1	0.70 até 0.74	0.542760	0.012041	0.109731	0.013869	0.117768
	2	0.74 até 0.79	0.514253	0.003049	0.055215	0.003252	0.057025
	3	0.80 até 0.84	0.516919	0.003017	0.054931	0.003304	0.057478
	4	0.85 até 0.89	0.511432	0.001475	0.038408	0.001606	0.040073
	5	0.90 até 1.00	0.509390	0.000890	0.029836	0.000978	0.031279
Direita	1	0.70 até 0.74	0.508656	0.000790	0.028100	0.000865	0.029403
	2	0.74 até 0.79	0.505782	0.000368	0.019178	0.000401	0.020031

Tabela 1. Propagação da perturbação em raio 2 e sensitividade à esquerda e à direita.

são ao mesmo tempo sensíveis à direita e possuem $s \geq 0.7$, resultando em 40208 chaves que satisfazem a esses dois requisitos. E essas regras foram classificadas nas Classes 1 e 2 da Tabela 1 e cifraram cada uma das 106 imagens de 64×64 pixels. As análises que podem ser extraídas dos experimentos referem-se também que as regras da Classe 2 realizam cifragens melhores que as regras da Classe 1. Tanto na abordagem com regras de sensitividade à esquerda e à direita os resultados se mostraram de boa qualidade frente ao teste de análise da propagação da perturbação para o algoritmo de criptografia baseado em ACs proposto neste trabalho.

5.3 Análise do tempo de processamento de imagens preto e branco

A implementação empregada neste experimento foi a abordagem sequencial do algoritmo BCCA. Esta implementação tem como objetivo comparar as diferenças no tempo de cifragem em relação à diferenças de parâmetros no próprio modelo. Além disso, a implementação sequencial realizada não tem como objetivo comparar o tempo com outros métodos com abordagem sequencial, tais como, DES e AES. Pois sabe-se que toda a motivação para o emprego dos algoritmos baseados em ACs vislumbram uma implementação paralela Wolfram [26]. Para isso, a implementação deve ser realizada em um hardware de altíssimo desempenho, tais como placas FPGA Das [5]. No entanto, sabe-se que tanto a implementação

paralela quanto a sequencial resultam a mesma imagem cifrada. Portanto, a implementação sequencial pode ser aplicada quando o estudo é apenas para analisar a qualidade da cifragem ou quando não se tem disponível a quantidade de núcleos gasta para a realização do experimento paralelo.

O experimento desta seção resume-se em calcular a quantidade de tempo utilizado na encriptação ou decriptação de diversas dimensões de uma mesma imagem. Para isso, varia-se a quantidade de passos de tempo (isto é, para cada dimensão, o teste avaliará o tempo necessário para cifrar e decifrar uma determinada figura). A regra empregada ao experimento possui raio 2 e é sensitiva à esquerda.

Inicialmente os testes concentraram-se em testar as diversas dimensões das imagens (eixo x) empregando 64 passos de tempo, conforme descrito na Figura 12 (a). O tempo, disposto no eixo y, é medido em segundos e demonstra que imagens pequenas aplicadas ao modelo em 64 passos gastam menos que 1 segundo no processo. Uma figura maior, com dimensão 512×512, obteve um tempo maior que 2 segundos para concluir a cifragem. Na Figura 12 (b) empregando-se o dobro de passos de tempo (isto é, $T = 128$), há um aumento significativo nas imagens maiores e pequenas modificações no tempo total das imagens menores. Antes alcançando pouco mais de 2 segundos, a figura de dimensão 512×512 aplicada a este teste leva mais que 4 segundos para ser cifrada. Ainda assim, este é um tempo pequeno para uma imagem com esta dimensão, haja vista a configuração da máquina.

Ambos os experimentos ilustrados anteriormente, concentraram-se na etapa de ci-

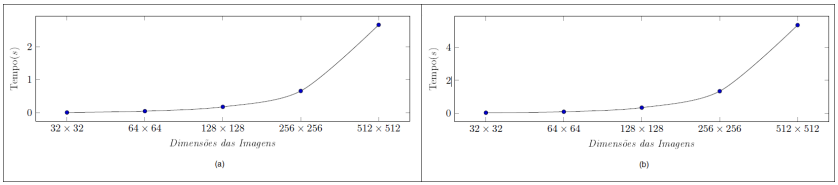

Figura 12. (a) Imagens cifradas em $t = 64$ passos de tempo. (b) Imagens cifradas em $t = 128$ passos de tempo.

fragem. A fase da decifragem também deve ser avaliada uma vez que é constantemente executada nas soluções que envolvam o uso da criptografia. O tempo necessário para decifrar as imagens cifradas no Figura 12 (a) é mostrado na Figura 13 (a), utilizando a mesma quantidade de passos de tempo. Note que a o comportamento do gráfico de tempo de decifragem é similar ao comportamento do tempo da cifragem, porém, a decifragem gasta uma quantidade menor e tempo para ser executada. O mesmo pode ser observado quando são aplicados $T = 128$ passos na decifragem, e a Figura13 (b) mostra esse comportamento.

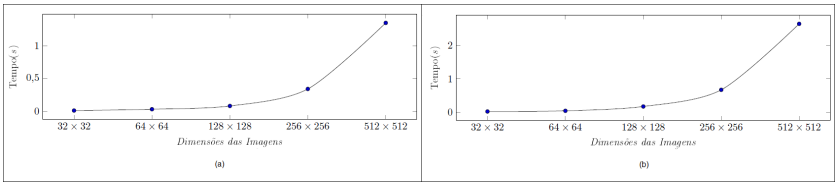

Figura 13. (a) Imagens decifradas em $T = 64$ passos de tempo. (b) Imagens decifradas em $T = 128$ passos de tempo.

5.4 Análise da qualidade da cifragem para imagens coloridas

O histograma é uma ferramenta gráfica capaz de focalizar a tonalidade das cores distribuídas nas imagens. O eixo x e y representam, respectivamente, a tonalidade das cores existentes e a quantidade de pixels empregados a cada cor. Utilizando-se das cores primárias em escalas de 0 a 255, o histograma consegue captar o grau de distribuição de cada um dos canais. A análise do histograma demonstra o esquema de pixels após os processos de pré-embaralhamento e cifragem dos dados. Os histogramas gerados nesta seção foram elaborados através da aplicação do modelo criptográfico em imagens de 64×64, empregando uma regra de raio 2 ($r = 2$) sensitiva à esquerda em 128 passos de tempo ($T = 128$). Os gráficos ilustrados foram elaborados através da ferramenta de manipulação de imagens GIMP [8].

A Figura 14 demonstra a distribuição de cores nos pixels encontrados na primeira imagem analisada. As variações de cada canal estão representadas pela legenda juntamente à coloração representativa da figura. Nota-se que os padrões de cor contidos em cada gráfico formam a imagem original.

Em seguida, os gráficos representados pela Figura 15, demonstram o efeito propor-

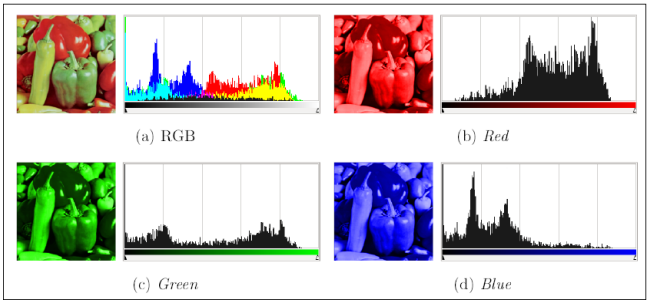

Figura 14. Histogramas de cor e análise de propagação da perturbação na imagem original.

cionado aos pixels, após a execução do pré-embaralhamento descrito anteriormente. Ob-

serve que os níveis de cores apresentados são distorcidos, formando novas figuras. O pré-embaralhamento situa-se no começo do processo de cifragem. Todavia, nota-se uma textura que revela a imagem original o que não é suficiente como forma de cifragem, por isso, o método baseado em ACs unidimensionais caóticos e com borda fixa é aplicado ao pré-embaralhemento para prover a segurança necessária ao mesmo.

O histograma final contido na Figura 16, demonstra o texto cifrado gerado através da

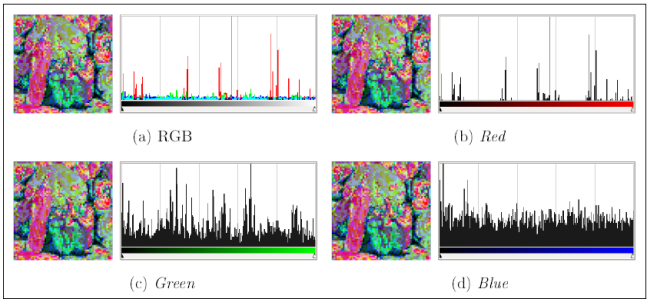

Figura 15. Histogramas de cor e análise de propagação da perturbação em $T = 1$.

abordagem de pré-embaralhamento e posteriormente cifragem de dados através dos algoritmos que foram aqui propostos. Neste caso, é possível observar que a distorção proporcionada pelo embaralhamento anterior, somada à fase de encriptação, acarreta em uma cifra de imagem com padrões divergentes comparados à imagem original. Através desta comparação, é possível expressar com um maior detalhamento a diferença entre o texto claro e o texto cifrado. Essa cifragem apresenta uma boa qualidade, visto que nenhum padrão está associado à imagem original e os histogramas de cores apresentam faixas de valores bem distribuídos em todas os canais de cores do padrão RGB. Embora o canal B esteja com faixas de valores mais esparsas, a cifragem resultante é de boa qualidade, pois alterou muito a dinâmica e a distribuição da imagem original.

5.5 Análise do tempo de processamento de imagens coloridas

O gráfico disposto na Figura 17 (a), demonstra que a maioria das imagens menores ou iguais a 256×256 levaram menos que 10 segundos no processo de cifragem, empregando 64 passos de tempo. Porém, a figura que possui dimensão 512×512 gastou mais que 30 segundos para finalizar o processo. Posteriormente, os novos testes visam computar o tempo total de processamento realizado na decifragem. Inicialmente a Figura 17 (b), demonstra a decifragem dos textos cifrados gerados na Figura 17 (a). Os resultados não diferem muito do tempo total de cifragem. No caso das imagens coloridas, o tempo final de processamento entre as operações de cifragem e decifragem, foram praticamente idênticas. Contudo, o modelo

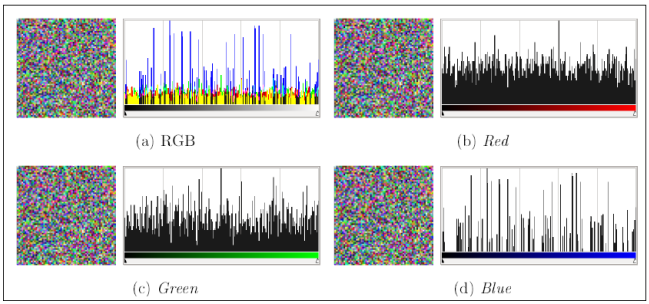

Figura 16. Histogramas de cor e análise de propagação da perturbação em $T = 64$.

apresentou um tempo elevado quando lida com imagens maiores numa abordagem sequencial. Consequentemente, isto reforça que o modelo deve ser implementado numa arquitetura paralela, tornando a criptografia mais rápida aos arquivos maiores.

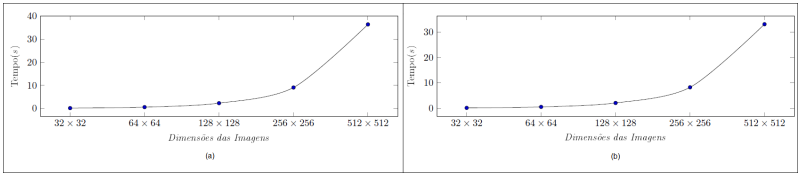

Figura 17. (a) Imagens cifradas em $T = 64$ passos de tempo. (b) Imagens decifradas em $T = 64$ passos de tempo.

6 Discussão dos Resultados

Os experimentos realizados com o modelo de criptografia BCCA tiveram como objetivo avaliar a qualidade da cifragem. O estudo de caso apresentado neste trabalho para a avaliação da cifragem foi a cifragem de imagens digitais coloridas. O modelo BCCA é capaz de cifrar qualquer sequência de bits, incluindo-se textos ou imagens preto e branco, portanto um tratamento de pré-processamento de imagens coloridas foi necessário. Esse pré-processamento foi o responsável por misturar os bits dos canais de cores e também por evitar zonas de texturas Prasad [19] nas imagens cifradas. Além disso, foram realizados experimentos para analisar a qualidade de imagens preto e branco que são representadas por matrizes onde cada pixel é representado por um bit. As maiores conclusões e observações dos experimentos são: (a) o tempo de processamento do método BCCA, se implementado numa arquitetura paralela, em relação ao modelo criptográfico AES é superior, tanto na cifragem

quanto na decifragem; (b) para uma boa cifragem do modelo BCCA - média de 0.5 0s a partir de análise da perturbação - devem ser utilizadas chaves sensíveis com alta entropia $s > 0.7$; (c) a propagação da perturbação pode ser vista ao longo de todo o reticulado, diferentemente do modelo de Gutowitz [9]; (d) o modelo de criptografia BCCA reduz o crescimento acelerado de bits observados nos modelos de Gutowitz [9] e Oliveira [17]; (e) toda sequência de bits como entrada pode ser cifrada utilizando-se o modelo BCCA, diferentemente do modelo Oliveira [18]; (f) o método aqui proposto pode permitir alto nível de paralelismo quando empregado num hardware paralelo, diferentemente de Oliveira [15]; (g) a qualidade de cifragem não é alterada na implementação paralela ou sequencial; (h) o pré-embaralhamento das matrizes de bits para cada canal de cor reduz a criação de zonas de texturas Prasad [19] e portanto foi empregado; (i) o histograma de cores indica que o método BCCA é melhor quando o método é aplicado por mais passos de tempo T; (j) testes de processamento de tempo sequencial foram empregados para mostrarem que à medida que a se aumenta o tamanho da imagem a ser cifrada, aumenta-se também o tempo de processamento tanto no implementação sequencial quanto na paralela, já que ambas são sensíveis ao tamanho da imagem. No entanto, a abordagem paralela do BCCA teria um tempo muito inferior, se comparado a abordagem sequencial, justificando a importância do emprego dos ACs na criptografia.

7 Definição formal do modelo de criptografia BCCA

A criptografia do modelo BCCA pode ser definida por uma 6-tupla $(\mathcal{P}, \mathcal{K}, T, \mathcal{C}, \varepsilon, \mathcal{D})$ que satisfaz as seguintes condições:

1. $\mathcal{P}$ é um conjunto de reticulados iniciais de tamanho $m \times (n + p)$ bits, também denominados blocos originais;

2. T representa a quantidade de passos necessários à cifragem, com $T = \max(m, n, p)$;

3. $\mathcal{K}$ é o conjunto das chaves possíveis $k \in \mathcal{K}$ é composta por uma palavra de 64 bits, sendo o tamanho da regra (chave). A partir do núcleo de uma regra principal sensível a um dos extremos (esquerda ou direita) e com entropia recomendada superior a 0.7;

4. $\mathcal{C}$ é um conjunto de textos cifrados;

5. Para cada $k \in \mathcal{K}$ tem uma regra de criptografia $e_k \in \varepsilon$, dada pelo cálculo de pré-imagem do modelo de AC 1D com borda fixa, e uma correspondente regra de descriptografia $e_k \in \mathcal{D}$, dada pela evolução para frente do modelo de AC 1D com borda fixa. Cada $e_k : \mathcal{P} \to \mathcal{C}$ e $d_k : \mathcal{C} \to \mathcal{P}$ são funções tais que $d_k(e_k(\mathcal{P})) = \mathcal{P}$.

8 Conclusões e trabalhos futuros

Este trabalho apresenta e investiga um novo modelo de criptografia paralelo baseado no cálculo de pré-imagens de autômatos celulares unidimensionais com chaves caóticas e borda fixa, denominado BCCA. Nosso modelo tem como caso de estudo as imagens coloridas digitais, o qual é muito relevante ao contexto de criptografia Prasad [19] (i) o estudo de cifragem de imagens digitais tem ganhado espaço devido ao advento dos equipamentos de captura dessas imagens e constante transmissão desses dados; (ii) criptografia de imagens é um problema canônico à criptografia, pois sua complexidade é superior que a cifragem de um texto e a metodologia se aproxima da cifragem de vídeos, mas com complexidade inferior.

Utilizando-se do cálculo de pré-imagens dos ACs unidimensionais, o modelo de criptografia obteve êxito em cifrar e decifrar imagens digitais. O grau de confusão de bits contidos nas chaves criptográficas de caráter caótico proporciona o efeito de embaralhamento contemplado pelas cifras, seja pelo aspecto visual ou através da análise de propagação da perturbação. O método consegue alcançar um dos principais requisitos relativos à segurança digital atribuído a confidencialidade, onde os textos cifrados originados no processo permitem a recuperação de suas informações somente por entidades autorizadas.

O modelo BCCA emprega como base a criptografia de Gutowitz [9] e Oliveira [17] baseada em ACs. O modelo elaborado diminui a quantidade de bits enviados ao texto cifrado através de uma otimização, representada pela técnica da borda. Além da redução dos bits enviados, o procedimento garante bloquear o crescimento do texto cifrado [9] e [17], aumentando a viabilidade do modelo criptográfico e eliminando problemas, tais como maior gasto de espaço em disco e maior custo no transporte das cifras. Todavia, o modelo ainda contempla uma taxa de bits extras, que deve ser salva para possibilitar a posterior recuperação do texto claro. Esses bits extras são necessários para que todo texto plano dado como entrada possa ser cifrado por qualquer chave (propriedade de sensitividade), diferentemente do algoritmo implementado em [18] que não permite a cifragem de qualquer sequencia de bits de entrada. Além disso, o modelo aqui empregado é altamente paralelizável, diferentemente do modelo de [15], que embora aumente o reticulado apenas quando é necessário, o mecanismo de pilha empregado neste algoritmo, para a recuperação de pré-imagens, reduz o paralelismo do mesmo. Um importante diferencial do modelo em comparação a outros algoritmos conhecidos na literatura, tais como AES e DES Stinson [23] é a capacidade de empregar a computação paralela. Mesmo que alguns métodos tenham sido investigados na tentativa de paralelizar os procedimentos redundantes dos modelos AES e DES Le [10], sabe-se que os mesmos apresentam vários trechos que precisam ser processados de modo sequencial.

Como continuidade deste trabalho, testes de criptoanálise para a verificação se a borda facilita o trabalho de criptoanálise podem ser investigados. Para a melhoria do modelo, caso o texto cifrado seja corrompido, uma análise de perda de dados poderá ser estudada. Uma mudança da atual borda utilizada para uma sequência pseudo-aleatória proposta por Wolfram [26], que partir de um reticulado e um índice gera seus valores aleatoriamente e em seguida

emprega uma execução *forward* em $T = 64$ passos de tempo, é uma outra proposta que pode apresentar ao modelo resultados interessantes. Em resumo, a estrutura da borda é flexível e pode proporcionar diversos experimentos com o objetivo de obter melhores cifragens.

Outra sugestão de alteração no processo de cifragem pode ser a rotação do núcleo das regras de transição (isto é, a geração de novas regras ao processo a partir de modificações do núcleo de uma chave principal). Sabe-se que toda a motivação do emprego do cálculo de pré-imagens baseado em ACs reside na implementação paralela, e que esse procedimento não altera o resultado final da cifragem. Portanto, espera-se aplicar melhorias em relação ao tempo de processamento a partir da implantação na plataforma FPGA (Field Programmable Gate Array) Anghelescu [1], Rajagopalan [20], Raut [21] através da exploração da computação paralela, justificando toda a motivação do emprego de ACs na criptografia.

Contribuição dos autores:

- Eduardo Cassiano da Silva: implementação do modelo proposto BCCA, estudo e validação do modelo, escrita do artigo, implementação e validação dos experimentos, adaptação do banco de imagens para os experimentos.

- Jaqueline Aparecida Jorge Papini Soares: Auxílio e tomada de decisões em relação aos testes e experimentos conduzidos, auxílio na implementação do modelo proposto BCCA, escrita do artigo, orientação do aluno Eduardo Cassiano da Silva.

- Danielli Araújo Lima: Proposição do modelo BCCA, auxílio e tomada de decisões em relação aos testes e experimentos conduzidos, auxílio na implementação do modelo proposto BCCA, escrita do artigo, orientação do aluno Eduardo Cassiano da Silva.

Referências

[1] P. Anghelescu, S. Ionita, and E. Sofron. Fpga implementation of hybrid additive programmable cellular automata encryption algorithm. In *Hybrid Intelligent Systems, 2008. HIS'08. Eighth International Conference on*, pages 96–101. IEEE, 2008.

[2] H. Barros de Macedo, G. M. Barbosa de Oliveira, and C. H. Costa Ribeiro. Dynamic behaviour of network cellular automata with non-chaotic standard rules. In *Complex Systems (WCCS), 2014 Second World Conference on*, pages 451–456. IEEE, 2014.

[3] A. P. Castro and D. A. Lima. Autômatos celulares aplicados a modelagem de dinâmica populacional em situação de risco. *Workshop of Applied Computing for the Management of the Environment and Natural Resources*, 2013.

[4] J. Daemen and V. Rijmen. Rijndael/aes. In *Encyclopedia of Cryptography and Security*, pages 520–524. Springer, 2005.

[5] D. Das and A. Ray. A parallel encryption algorithm for block ciphers based on reversible programmable cellular automata. *arXiv preprint arXiv:1006.2822*, 2010.

[6] C. Feliciani and K. Nishinari. An improved cellular automata model to simulate the behavior of high density crowd and validation by experimental data. *Physica A: Statistical Mechanics and its Applications*, 451:135–148, 2016.

[7] G. B. Ferreira, P. A. Vargas, and G. M. Oliveira. An improved cellular automata-based model for robot path-planning. In *Advances in Autonomous Robotics Systems*, pages 25–36. Springer, 2014.

[8] G. Gimp. Image manipulation program. *User Manual, Edge-Detect Filters, Sobel, The GIMP Documentation Team*, 8(2):8–7, 2008.

[9] H. Gutowitz. Cryptography with dynamical systems. *Kluwer Academic Press*, 1995.

[10] D. Le, J. Chang, X. Gou, A. Zhang, and C. Lu. Parallel aes algorithm for fast data encryption on gpu. In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, volume 6, pages V6–1. IEEE, 2010.

[11] H. A. Lima and D. A. Lima. Autômatos celulares estocásticos bidimensionais aplicados a simulação de propagação de incêndios em florestas homogêneas. *Workshop of Applied Computing for the Management of the Environment and Natural Resources*, 2014.

[12] M. J. L. Lima. Criptografia baseada no calculo generico de pre-imagens de autômatos celulares. Master's thesis, Universidade Presbiteriana Mackenzie, 2005.

[13] H. B. Macêdo, G. Oliveira, and C. H. Ribeiro. A comparative study between the dynamic behaviours of standard cellular automata and network cellular automata applied to cryptography. *International Journal of Intelligent Systems*, 31(2):189–207, 2016.

[14] J. Machicao, A. G. Marco, and O. M. Bruno. Chaotic encryption method based on life-like cellular automata. *Expert Systems with Applications*, 39(16):12626–12635, 2012.

[15] G. M. Oliveira, L. G. Martins, L. S. Alt, and G. B. Ferreira. Exhaustive evaluation of radius 2 toggle rules for a variable-length cryptographic cellular automata-based model. In *Cellular Automata*, pages 275–286. Springer, 2010.

[16] G. M. B. Oliveira. Autômatos celulares: aspectos dinâmicos e computacionais. *III Jornada de Mini-cursos em Inteligência Artificial (MCIA) - Sociedade Brasileira de Computação*, 8:297 – 345, 2003.

[17] G. M. B. Oliveira, A. Coelho, and L. Monteiro. Cellular automata cryptographic model based on bi-directional toggle rules. *Int. J. of Modern Physics C*, 2004.

[18] G. M. B. Oliveira, M. Lima, H. Macedo, and A. Branquinho. A cryptographic modelo based on the pre-image computation of cellular automata. *Theory and Applications of Cellular Automata*, pages 139 – 155, 2008.

[19] V. C. Prasad and S. Maheswari. Robust watermarking of aes encrypted images for drm systems. In *Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), 2013 International Conference on*, pages 189–193. IEEE, 2013.

[20] S. Rajagopalan, H. N. Upadhyay, J. B. B. Rayappan, and R. Amirtharajan. Dual cellular automata on fpga: An image encryptors chip. *Res. J. Inform. Technol*, 6:223–236, 2014.

[21] L. Raut and D. H. Hoe. Stream cipher design using cellular automata implemented on fpgas. In *System Theory (SSST), 2013 45th Southeastern Symposium on*, pages 146–149. IEEE, 2013.

[22] S. Sen, C. Shaw, D. R. Chowdhuri, N. Ganguly, and P. P. Chaudhuri. Cellular automata based cryptosystem (cac). In *Information and Communications Security*, pages 303–314. Springer, 2002.

[23] D. R. Stinson. *Cryptography: theory and practice*. CRC press, 2005.

[24] S. Vasantha, N. Shivakumar, and D. S. Rao. A new encryption and decryption algorithm for block cipher using cellular automata rules. *International Journal*, 130, 2015.

[25] S. Wolfram. *Cellular Automata.* Los Alamos Science., 1986.

[26] S. Wolfram. *A New Kind of Science*. Wolfram Media - (1st edition): 1197 - 2006-09-19T07:35:05.000+0200, January 2002.

[27] A. Wuensche and M. Lesser. *The global dynamics of cellular automata: An atlas of basin of attraction fields of one-dimensional cellular automata.* Number 1. Andrew Wuensche, 1992.

[28] L. Yu, Y. Li, and X. Xia. Image encryption algorithm based on self-adaptive symmetrical-coupled toggle cellular automata. In *Image and Signal Processing, 2008. CISP'08. Congress on*, volume 3, pages 32–36. IEEE, 2008.

[29] M. Zeghid, M. Machhout, L. Khriji, A. Baganne, and R. Tourki. A modified aes based algorithm for image encryption. *International Journal of Computer Science and Engineering*, 1(1):70–75, 2007.

[30] Y. Zhang, J. Qiao, B. Wu, W. Jiang, X. Xu, and G. Hu. Simulation of oil spill using ann and ca models. In *Geoinformatics, 2015 23rd International Conference on*, pages 1–5. IEEE, 2015.