Estudo sobre a Aplicação de Autômatos Celulares Caóticos em Criptografia

Mauro Tardivo Filho , Marco Aurélio Amaral Henriques (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

{maurotf,marco}@dca.fee.unicamp.br

Abstract – This paper deals with the study of Cellular Automata (CA) applied in cryptography. Motivated by CA properties, it summarizes CA construction, CA with chaotic behavior and Margolus neighborhood partitioning method for reversible CA. After this, it reviews CA applications in cryptography, discussing some articles that used CA as a source of random numbers, stream ciphers, encrypting and hash algorithms. Finally, it mentions future work that could be done, such as: performance improvement using the parallelism property of CA, evaluation of hardware implementation and cryptanalysis methods that can evaluate the security of algorithms which uses CA.

Keywords – Cellular Automata, Cryptography, Chaotic System

1. Introdução

O objetivo deste estudo é avaliar novas abordagens para criptografia baseadas em autômatos celulares. Autômato celular é um modelo de sistemas discretos com regras simples, mas que apresentam um comportamento complexo. O "jogo da vida"[3], um autômato celular de duas dimensões e dois estados, criado por John Conway foi alvo de grande atenção na década de 70. Em 1983, Stephen Wolfram publicou o primeiro de uma série de artigos onde analisava o comportamento e a complexidade de autômatos celulares induzidos por regras elementares e em 2002 reuniu todos seus estudos no livro "A New Kind of Science"[10].

Com esta motivação, este trabalho explica o funcionamento dos autômatos celulares que são utilizados em criptografia e apresenta algumas abordagens já realizadas por outros pesquisadores.

2. Autômatos Celulares

2.1. Autômatos Celulares Convencionais

Autômatos celulares (CA) unidimensionais têm o comportamento de um vetor de *N* células, cada uma com valor binário e evoluem a partir de uma regra Φ definida. Um autômato celular é chamado uniforme quando todas as células evoluem a partir de uma mesma regra e é chamado híbrido quando diferentes células evoluem a partir de regras diferentes.

Cada célula possui um raio de interação *r* e, desta forma, existem $k = (2r + 1)$ parâmetros de entrada para a função Φ, o que permite 2^k diferentes combinações binárias de vizinhos. Com isso, há 2^{2^k} possíveis combinações binárias para o próximo estado. Pode-se definir o próximo estado da *i*-ésima célula como $x_i^{t+1} = \Phi(x_{i-r}^t, ..., x_i^t, ..., x_{i+r}^t)$.

As condições de fronteira possuem duas abordagens: autômato nulo considera as células adjacentes aos extremos com valor nulo; autômato periódico considera as células dos extremos adjacentes entre si.

Como o estado de uma célula depende apenas de estados anteriores da mesma e de suas vizinhas, o processamento do estado de cada célula em uma geração pode ser realizado de maneira independente, ou seja, os estados das células são calculados em paralelo.

2.2. Caos Determinístico

Caos determinístico é o comportamento instável de um sistema determinístico influenciado por um número (mesmo que) pequeno de fatores. Devido à grande dependência das condições iniciais, pequenos erros são amplificados pelo número elevado de operações que ocorrem, tornando o comportamento pseudo-aleatório. As características que determinam o caos determinístico são:

- sequência dos estados é obtida algoritmicamente;
- dois estados similares se distanciam exponencialmente com relação ao tempo;
- a dinâmica do sistema após um número suficiente de iterações torna-se impossível de prever de forma analítica (apenas com a própria iteração do algoritmo);

- a dinâmica do sistema não possui padrões reconhecíveis.

Esta propriedade caótica de grande dependência das condições iniciais se assemelha ao efeito avalanche, característica básica de algoritmos criptográficos e que consiste em se obter grandes variações na saída dos algoritmos quando pequenas alterações são feitas na entrada.

2.3. Autômatos Celulares Caóticos

Os CA são modelos computacionais cuja evolução é processada a partir de suas condições iniciais usando uma função que determina o estado da próxima geração. Do ponto de vista criptográfico, deseja-se obter um CA que é capaz de prescrever uma evolução caótica de forma garantir as propriedades listadas na Seção 2.2. CA de Classe 3 [8] possuem uma dinâmica temporal que é computacionalmente irreversível.

Entre as regras dessa classe, a regra 30 é uma das que melhor produz sequências aleatórias [9] que, para raio de vizinhança $r = 1$, são definidas por:

$$x_i^{t+1} = x_{i-1}^t \oplus (x_i^t \vee x_{i+1}^t)$$

A Tabela 1 apresenta a regra que governa o próximo estado de uma célula a partir do estado atual da célula e de suas vizinhas.

Tabela 1. Estado atual e o próximo estado gerado a partir da regra 30

atual	111	110	101	100	011	010	001	000
próximo	x0x	x0x	x0x	x1x	x1x	x1x	x1x	x0x

A Figura 1 apresenta a dinâmica espaço-tempo caótica do CA utilizando a regra 30.

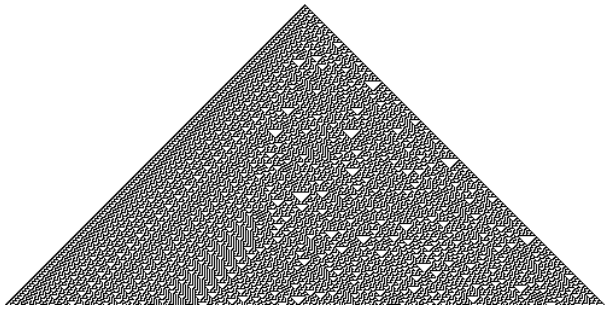

Figura 1. Autômato Celular utilizando Regra 30

Desta forma, os CA caóticos são aproveitáveis para criptografia devido também a sua irredutibilidade e irreversibilidade computacional. Irredutibilidade trata da incapacidade de predizer qualquer estado futuro, devido ao elevado número de células que afetarão tal estado após diversas gerações. Irreversibilidade trata da impossibilidade de obter estados anteriores, por exemplo, o estado inicial, devido as características caóticas explicadas na Seção 2.2.

2.4. Vizinhança de Margolus

Existem algumas técnicas para gerar autômatos celulares reversíveis, sendo o autômato celular particionado em grupos, onde as regras atuarão apenas localmente. Uma das técnicas mais simples é o particionamento por vizinhança de Margolus [7].

Autômatos celulares convencionais normalmente não são inversíveis porque o passo inverso não é determinístico, sendo que uma das principais razões é a perda de informação durante a evolução do mesmo.

Como pode ser visto na Figura 2, o próximo estado de uma célula do CA convencional unidimensional depende dela mesma e de suas duas vizinhas. A vizinhança de Margolus utiliza duas células como entrada para gerar o próximo estado de duas células, conservando a quantidade de informação.

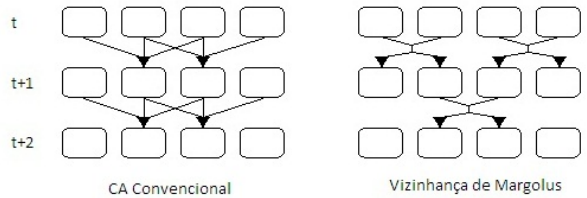

Figura 2. Vizinhança de Margolus

Para que a informação seja propagada entre as partições, proporcionando o efeito avalanche, a vizinhança de Margolus alternadamente modifica a grade de particionamento entre as iterações, como pode ser visto para um CA bidimensional na Figura 3.

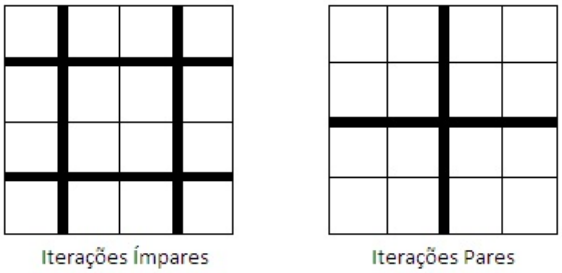

Figura 3. Particionamento da vizinhança de Margolus nas iterações ímpares e pares

A vizinhança de Margolus proporciona a possibilidade do passo reverso para CA de forma mais simples, visto que não há necessidade de escolha de autômatos inversíveis específicos. Além disso, o efeito avalanche ocorre devido a alternância da grade de particionamento para difusão dos dados.

3. Aplicações

Há na literatura, artigos que utilizam CA para algoritmos de cifragem, algoritmos de hash e algoritmos para gerar números aleatórios.

Anghelescu, Ionita e Bostan [1] apresentam um algoritmo de cifragem e decifragem utilizando CA híbrido unidimensional programável em circuitos lógicos. O interessante desta referência é a utilização de regras diferentes a cada iteração da cifragem. Porém, existem algumas lacunas no artigo com relação a performance do mesmo em comparação com outros algoritmos já consolidados, implementação em hardware, exploração do paralelismo existente em CA e resistência contra métodos de criptoanálise mais comuns.

Das e Ray [2] apresentam também um algoritmo de cifragem e decifragem utilizando CA híbrido unidimensional programável em circuitos lógicos, mas o mapeamento do texto em claro no autômato celular não é clara, deixando dúvidas sobre como implementar o mesmo. Os autores do artigo implementaram o algoritmo aproveitando o paralelismo, o que trouxe um desempenho superior ao AES implementado no mesmo computador, mas o ponto fraco está no custo de implementação maior do que os algoritmos comparados. Apesar do desempenho melhor, não há nenhuma criptoanálise do algoritmo proposto, portanto não se conhece a robustez do mesmo.

Nandi, Kar e Chaudhuri [6] aplicam CA híbrido unidimensional programável em circuitos lógicos para cifragem e decifragem em estruturas de bloco e fluxo. No algoritmo de cifra de fluxo proposto, CA são utilizados como geradores de chaves aleatórias, o que representa um importante aspecto para a criptografia,visto que protocolos criptográficos necessitam de números aleatórios para chaves de sessão. Foi realizada criptoanálise para as propostas de cifras de bloco e fluxo, porém ficou faltando analisar o desempenho das mesmas com relação a outros algoritmos conhecidos.Além disso, implementações em hardware e/ou utilizando paralelismo não foram mostradas.

Kumar, Kesava e Salivahanan [5] apresentam CA híbrido bidimensional para cifragem e CA híbrido unidimensional como gerador de sub-chaves de rodada. Este algoritmo tem o objetivo de ser utilizado em dispositivos restritos (redes de sensores sem-fio e cartões inteligentes, por exemplo) devido a sua baixa complexidade de hardware e baixo custo de energia. O algoritmo foi implementado em MATLAB e em hardware,tendo este último explorado o paralelismo de CA, que garantiu um desempenho superior ao AES. Porém faltou explicar como é o mapeamento do texto em claro para o espaço bidimensional, além de verificar a resistência do algoritmo contra criptoanálise.

Hecht [4] apresenta CA unidimensional para algoritmo de hash. Este algoritmo divide a mensagem em blocos para servir de entrada para o CA. A entrada do CA é um ou-exclusivo de um bloco da mensagem juntamente com a saída gerada pelo CA anterior. Este processo é repetido para todos os blocos da mensagem e o hash é a saída do CA para o último bloco. Não existem detalhes sobre o qual tipo de CA utilizar, pois este algoritmo pode ser projetado de acordo com os recursos computacionais disponíveis. No entanto, estão faltando análises de desempenho, criptoanálise e implementações do algoritmo para se obter um melhor conhecimento de sua aplicação.

Como pôde ser visto, existem alguns pontos não bem explorados na implementação de CA em criptografia, principalmente com relação ao paralelismo. Por causa da evolução dos processadores, que hoje apresentam vários núcleos, os CA podem ter um desempenho superior aos algoritmos tradicionais. Entretanto, há dúvidas com relação a robus-

tez de CA em criptografia, por não ser uma abordagem tão difundida e por falta de criptoanálises mais rigorosas

4. Conclusões

Este trabalho fez uma revisão sobre o estudo de autômatos celulares para possíveis aplicações em criptografia. Como trabalhos futuros, podem ser citados: proposta de melhorias de desempenho de CA aproveitando o paralelismo existente na estrutura do mesmo, avaliação de implementação em hardware e métodos de criptoanálise que permitam avaliar mais precisamente a segurança dos algoritmos que utilizam CA.

Referências

[1] Petre Anghelescu, Silviu Ionita, and Ionel Bostan. Design of programmable cellular automata based cipher scheme. *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pages 187–192, 2009.

[2] Debasis Das and Abhishek Ray. A Parallel Encryption Algorithm for Block Ciphers Based on Reversible Programmable Cellular Automata. *Journal of Computer Science*, 1(1):82–90, 2010.

[3] Martin Gardner. The fantastic combinations of john conway's new solitaire game "life". *Scientific American*, (223):120–123, 1970.

[4] Juan Pedro Hecht. Autómatas celulares caóticos en la generación de funciones. *IV Congreso Iberoamericano de Seguridad Informática CIBSI'07*, pages 157–170, 2007.

[5] K J Jegadish Kumar, K Chenna Kesava, and S Salivahanan. Novel and Efficient Cellular Automata based Symmetric Key Encryption Algorithm for Wireless Sensor Networks. *International Journal*, 13(4), 2011.

[6] S. Nandi, B.K. Kar, and P. Pal Chaudhuri. Theory and applications of cellular automata in cryptography. *IEEE Transactions on Computers*, 43(12):1346–1357, 1994.

[7] Tommaso Toffoli and Norman Margolus. Invertible cellular automata: A review. *Physica D 45*, 1990.

[8] Stephen Wolfram. Universality and complexity in cellular automata. *Physica D10*, 1984.

[9] Stephen Wolfram. Random sequence generation by cellular automata. *Adv Appl Math*, 7:123, 1986.

[10] Stephen Wolfram. *A New Kind of Science*. Wolfram Media, Inc, 2002.