Geradores de Números Pseudo-Aleatórios baseados em Recorrências Lineares sobre GF(2^e)

Afonso S. D. Neto, Carlos E. C. Souza†, Davi Moreno, Matheus H. S. Sousa, Daniel P. B. Chaves e Cecilio Pimentel

***Resumo*— Este trabalho apresenta novos PRNGs baseados em recorrências lineares sobre corpos de extensão GF(2^e). É realizada uma análise do período da recorrência linear e são obtidas as condições para que os PRNGs propostos alcancem o período máximo de $2^e - 1$. Também é proposto um diagrama lógico do circuito utilizando estruturas de árvore binária balanceada, com análise de complexidade do circuito. As propriedades estatísticas dos PRNGs propostos são avaliadas por meio das baterias de testes estatísticos NIST e TestU01. Os resultados mostram que, para configurações específicas, os PRNGs são aprovados em todos os testes aplicados.**

***Palavras-Chave*— Geradores de números pseudo-aleatórios, corpos finitos, recorrências lineares, diagrama lógico do circuito.**

***Abstract*— In this work we introduce new PRNGs based on linear recurrences over the Galois field GF(2^e). We analyze the period properties of the linear recurrence and identify the conditions under which the proposed PRNGs achieve the maximum period of $2^e - 1$. A logical circuit diagram using balanced binary tree structures is proposed and its complexity is analyzed. The statistical quality of the proposed PRNGs is evaluated using the NIST and TestU01 statistical test suites. The results show that, for specific configurations, the PRNGs pass all the applied tests.**

***Keywords*— Pseudo-random number generators, finite fields, linear recurrences, circuit logical diagram.**

I. Introdução

Geradores de números pseudo-aleatórios (PRNGs) são componentes fundamentais no projeto de sistemas de comunicação e esquemas criptográficos. Idealmente, PRNGs devem gerar sequências binárias com estatísticas semelhantes às de uma sequência verdadeiramente aleatória, além de demandar baixos recursos de hardware para implementação em circuito. Trabalhos recentes propõem novas famílias de PRNGs baseados em sistemas dinâmicos caóticos [1]–[4]. Alternativamente, novas propostas vêm utilizando mapas caóticos definidos sobre estruturas algébricas discretas, conhecidos na literatura como mapas caóticos discretos [5]. Por exemplo, em [6]–[8] é utilizado o mapa de Arnold discreto sobre os anéis de inteiros $\mathbb{Z}_{2^m}$ e $\mathbb{Z}_{3^m}$.

PRNGs baseados em mapas caóticos discretos apresentam a vantagem de utilizar exclusivamente aritmética modular. A natureza discreta das operações modulares evita problemas de arredondamento e truncamento, frequentemente observados em implementações de mapas caóticos definidos sobre os reais [5], [9]. Neste caso, essas operações podem levar à degradação da dinâmica caótica após algumas iterações. Além disso, operações de ponto flutuante empregadas nestes sistemas requerem mais recursos de hardware para implementação em circuito. Outra vantagem apresentada por mapas discretos é a replicação exata da dinâmica.

A dinâmica gerada por mapas definidos sobre estruturas discretas é, em geral, descrita por relações de recorrência. Uma nova abordagem para projetos de PRNGs sobre estruturas discretas consiste em explorar recorrências definidas sobre estruturas alternativas aos tradicionais anéis de inteiros, como corpos de extensão GF(p^e), em que p é um número primo e o expoente $e \geq 2$ é um inteiro. Neste contexto, o corpo GF(2^e) é conveniente para a implementação em hardware devido ao corpo base consistir de aritmética binária. Até onde chega o conhecimento dos autores, o uso de corpos de extensão no projeto de PRNGs ainda não é explorado na literatura.

Neste trabalho, é proposto um PRNG baseado em uma recorrência linear sobre o corpo de extensão GF(2^e). É apresentada uma análise teórica do período da recorrência, e as condições para alcançar o período máximo de $2^e - 1$ são obtidas. A complexidade do circuito é avaliada em termos do número de portas lógicas (XORs e multiplexadores) e do comprimento do caminho crítico, que determina a frequência máxima de operação. O diagrama lógico proposto utiliza estruturas de árvore binária balanceada para otimizar o compromisso entre área ocupada e atraso. A performance dos PRNGs propostos é analisada com uma extensa bateria de testes estatísticos, incluindo National Institute of Standards and Technology (NIST) SP 800-22 e as baterias do TestU01: Federal Information Processing Standard (FIPS), Alphabit, BlockAlphabit, PseudoDIEHARD, Rabbit, SmallCrush, Crush e BigCrush. Convém ressaltar que subconjuntos dessas baterias são usados em [1]–[4], [6]–[8]. Por exemplo, a bateria NIST é empregada em [1], [4], [6]–[8], enquanto as baterias NIST, Alphabit e Rabbit são utilizadas em [2], [3]. Os resultados mostram que configurações específicas do PRNG proposto são aprovadas em todos os testes aplicados.

O restante deste trabalho está organizado da seguinte forma. Na Seção II são apresentados conceitos preliminares sobre

Os autores são do Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco, Recife-PE; E-mails: {afonso.delgado, carlos.ecsouza, davi.moreno, matheus.henriquesousa, daniel.chaves, cecilio.pimentel}@ufpe.br.

†Agora com o Wireless Information and Network Sciences Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

corpos finitos necessários para o desenvolvimento do trabalho. A Seção III apresenta a recorrência linear proposta sobre GF(2^e) e a análise do seu período. Na Seção IV é proposto o diagrama lógico do circuito e sua análise de complexidade. A Seção V descreve as baterias de testes estatísticos utilizadas. Os resultados experimentais são apresentados na Seção VI. Por fim, as conclusões são discutidas na Seção VII.

II. Preliminares

Dado um polinômio não nulo $p(x)$, o grau de $p(x)$, denotado $\deg(p(x))$, é o maior expoente de x com coeficiente não nulo. Seja GF(2) o conjunto das classes de equivalência dos inteiros módulo dois. Seja GF(2)$[X]$ o conjunto dos polinômios com coeficientes em GF(2) e $h(x) \in$ GF(2)$[X]$ um polinômio primitivo com $\deg(h(x)) = e$. O corpo de extensão GF(2^e) é definido pelo conjunto das classes de equivalência módulo $h(x)$ dos polinômios em GF(2)$[X]$ com grau menor que e. O corpo GF(2) é denominado corpo base de GF(2^e). Seja $\alpha \notin$ GF(2) uma raiz de $h(x)$. Um elemento de GF(2^e) pode ser escrito como uma combinação linear das potências de α. Portanto, GF(2^e) é um espaço vetorial e-dimensional sobre GF(2^e) e seus elementos são gerados pela base $\{1, \alpha, \alpha^2, \ldots, \alpha^{k-1}\}$. O peso de Hamming de um polinômio é definido como o número de coeficientes não nulos. Dado um polinômio não nulo $p(x) \in$ GF(2^e), sua ordem, denotada por ord$(p(x))$, é o menor inteiro positivo k tal que $p(x)^k = 1$.

III. Recorrência Linear sobre GF(2^e)

Considere a recorrência linear sobre GF(2^e) definida por

$$x_{n+1} = a(x)x_n + c(x) \pmod{h(x)} \tag{1}$$

em que $n \geq 0$, $a(x), c(x) \in$ GF(2^e) e $h(x)$ é um polinômio primitivo sobre GF(2). A aplicação recursiva de (1) sobre uma condição inicial x_0 gera uma sequência de polinômios $S = \{x_0, x_1, x_2, \ldots\}$, em que cada $x_n \in S$ é um polinômio com grau menor que e em GF(2)$[X]$. Cada $x_n \in S$ pode ser representado por e coeficientes binários, portanto a concatenação destes coeficientes produz sequência binárias de comprimento e a cada iteração.

A. Análise do Período

O período de um mapa iterativo é o menor $m > n$ tal que $x_m = x_n$. Expandindo (1) e expressando o termo geral x_n em função de x_0, obtemos

$$x_n = x_0 a(x)^n + c(x)\sum_{i=0}^{n-1} a(x)^i = x_0 a(x)^n + c(x)\frac{1-a(x)^n}{1-a(x)}. \tag{2}$$

Igualando os termos gerais x_m e x_n em (2), obtemos:

$$x_0(a(x)^m - a(x)^n) = c(x)(1-a(x))^{-1}(a(x)^m - a(x)^n). \tag{3}$$

Esta equação divide a análise em dois casos:

1) Se $x_0 = c(x)(1-a(x))^{-1}$, a equação é satisfeita para quaisquer m e n. O menor $m > n$ é $m = n+1$, resultando em um período 1. Este representa o ponto fixo da relação linear e deve ser evitado.
2) Se $x_0 \neq c(x)(1-a(x))^{-1}$, devemos ter $a(x)^m = a(x)^n$, o que implica $a(x)^{m-n} = 1$. Isto significa que $(m-n)|\text{ord}(a(x))$. O menor $m > n$ que satisfaz esta condição é $m = \text{ord}(a(x)) + n$.

Concluímos que o período da recorrência linear é dado por:

$$\text{Período} = \begin{cases} 1, & \text{se } x_0 = c(x)(1-a(x))^{-1} \\ \text{ord}(a(x)), & \text{se } x_0 \neq c(x)(1-a(x))^{-1}. \end{cases} \tag{4}$$

O período máximo é alcançado quando ord$(a(x))$ é máximo. Em GF(2^e), esse valor é $2^e - 1$, e é obtido pelos elementos primitivos do corpo. Portanto, para alcançar o período máximo de $2^e - 1$, deve-se garantir que $a(x)$ seja um elemento primitivo de GF(2^e) e $x_0 \neq c(x)(1-a(x))^{-1}$.

É importante observar que se $x_0 \neq c(x)(1-a(x))^{-1}$ a sequência S gerada não possui ponto fixo. De fato, se existir algum $i > 0$ tal que $x_i = c(x)(1-a(x))^{-1}$, então $x_{i-1} = c(x)(1-a(x))^{-1}$. Por indução, todos os elementos são iguais x_i, incluindo x_0, contradizendo a hipótese inicial.

IV. Diagrama Lógico e Análise de Complexidade do Circuito

A proposta de diagrama lógico para a recorrência linear apresentada na Seção III requer uma análise do compromisso entre área ocupada e comprimento do caminho crítico do circuito. A área é determinada principalmente pela quantidade de componentes lógicos necessários (portas XOR e multiplexadores), enquanto o caminho crítico representa o maior atraso entre a entrada e a saída do circuito, afetando a frequência máxima de operação do circuito.

O circuito implementa duas operações principais: a relação de recorrência $y_n = a(x)x_n + c(x)$ e a redução modular $x_{n+1} = y_n \bmod h(x)$, em que $h(x)$ um polinômio de grau e e $a(x)$ um polinômio de grau d. No projeto do circuito, busca-se minimizar o número de componentes e o comprimento do caminho crítico utilizando uma estrutura paralela e balanceada. As Figuras 1 e 2 ilustram as duas etapas do circuito.

Seja $a(x) = \sum_{i=1}^{w} x^{t_i}$, em que w é o número de coeficientes não nulos de $a(x)$ e $t_1 > t_2 > \cdots > t_w$. Para a implementação da relação de recorrência, ilustrada na Figura 1, o circuito organiza portas XOR em uma estrutura de árvore binária balanceada para paralelizar as operações. Cada nível da árvore combina pares de entradas usando portas XOR até que um resultado final seja obtido na raiz.

Para um número de entradas k, esta estrutura requer exatamente $k-1$ portas XOR e tem uma profundidade de $\lceil \log_2(k) \rceil$. No nosso caso, com w termos não nulos em $a(x)$ e o fator $c(x)$, cada coeficiente de y_n requer w portas XOR de $(e+d)$ bits organizadas em $\lceil \log_2(w+1) \rceil$ níveis. O número total de portas XOR de 1 bit para esta etapa é então $(e+d)(w-1)$, com um caminho crítico de comprimento $\lceil \log_2(w+1) \rceil \tau_{\text{XOR}}$, onde τ_{XOR} representa o atraso de uma porta XOR de 1 bit.

O circuito de redução modular, ilustrado na Figura 2, segue uma estrutura similar de árvore binária ao implementar a redução módulo $h(x)$. Como há $d+1$ entradas, o esquema requer d portas XOR de e bits e o mesmo número de elementos MUX de e bits, organizadas em uma árvore de profundidade $\lceil \log_2(d+1) \rceil$. Isto cria um caminho crítico de comprimento

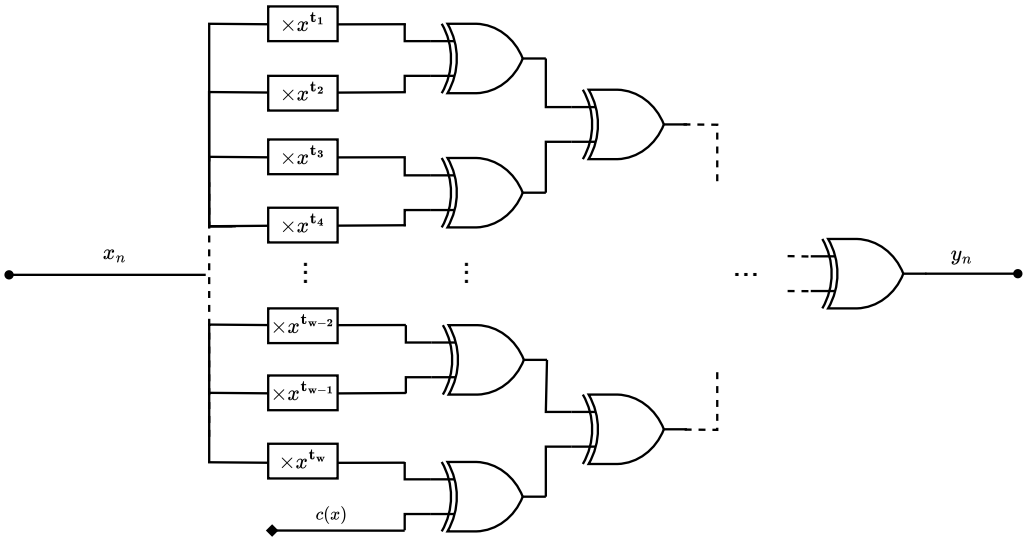

Fig. 1
DIAGRAMA LÓGICO DA RELAÇÃO DE RECORRÊNCIA $y_n = a(x)x_n + c(x)$.

$\tau_{\text{MUX}} + \lceil \log_2(d+1) \rceil \tau_{\text{XOR}}$, onde τ_{MUX} é o atraso de um elemento MUX. A contagem total de portas para esta etapa é ed tanto para portas XOR quanto para MUX de 1 bit.

Combinando estes componentes, o circuito requer no máximo $(e+d)(w-1) + ed$ portas XOR de 1 bit e ed elementos MUX de 1 bit. O comprimento total do caminho crítico é $\left(\lceil \log_2(w+1) \rceil + \lceil \log_2(d+1) \rceil\right) \tau_{\text{XOR}} + \tau_{\text{MUX}}$. Esta implementação revela várias propriedades:

- O comprimento do caminho crítico é independente da escolha do polinômio primitivo $h(x)$.
- O caminho crítico escala logaritmicamente com d e w.
- A contagem de portas aumenta linearmente com os graus d e e de $a(x)$ e $h(x)$, respectivamente.

V. TESTES DE ALEATORIEDADE

Para avaliar a qualidade das sequências geradas pelo PRNG proposto, utilizamos duas suítes de testes estatísticos: NIST SP 800-22 [10] e TestU01 [11]. A bateria NIST SP 800-22 é um padrão estabelecido pelo National Institute of Standards and Technology, consistindo em 15 testes estatísticos que avaliam diferentes aspectos de aleatoriedade. Os testes incluem análises de frequência em nível de bit e bloco, corridas de zeros e uns, rank de matrizes binárias, complexidade linear, entropia aproximada, testes universais de Maurer, e testes de autocorrelação. Cada teste produz dois indicadores principais: a taxa de aprovação, que representa o percentual de sequências que passam em uma série de testes, e o p-valor, que é utilizado para avaliar a concordância entre os dados e a hipótese de aleatoriedade. O teste NIST é realizado com um nível de confiança $\alpha = 0{,}01$, sendo este o valor recomendado em [10]. Os testes são realizados com um conjunto de 100 subsequências binárias, cada uma com comprimento de 10.000 bits.

O TestU01 é uma biblioteca mais extensa, oferecendo um conjunto abrangente de testes estatísticos organizados em diferentes baterias:

- FIPS: 16 testes que implementam os critérios definidos pelo Federal Information Processing Standard, requerendo uma sequência de 19,53 Kb.
- Alphabit: 17 testes específicos para sequências binárias, analisando uma sequência de 576 Mb, com, no máximo, 2^{26} bits por teste.
- BlockAlphabit: Versão estendida do Alphabit com 102 testes, processando uma sequência de 3,37 Gb em blocos, também com, no máximo, 2^{26} bits por teste.
- Pseudo-DIEHARD: 126 testes baseados na bateria DIEHARD, necessitando de uma sequência de 5,88 Gb.
- Rabbit: 40 testes, utilizando uma sequência de 1,49 Gb, com $n = 2^{26}$ bits por teste.
- SmallCrush: Versão reduzida da bateria Crush com 15 testes, requerendo uma sequência de 6,76 Gb.
- Crush: Bateria com 144 testes, analisando uma sequência de 1008,62 Gb.
- BigCrush: Bateria mais rigorosa da família Crush com 160 testes, necessitando uma sequência de aproximadamente 10,6 Tb.

Para o TestU01, cada teste estatístico produz um p-valor específico que determina a aprovação da sequência. Uma sequência é considerada aprovada em um teste quando seu p-valor está no intervalo [0,001; 0,999], indicando que a sequência possui boas propriedades de aleatoriedade.

VI. RESULTADOS EXPERIMENTAIS

A metodologia experimental adotada seguiu uma estratégia sistemática de busca por configurações efetivas do PRNG. Realizamos uma busca por polinômios primitivos $h(x)$ de grau e fixando o peso de Hamming, o coeficientes de maior

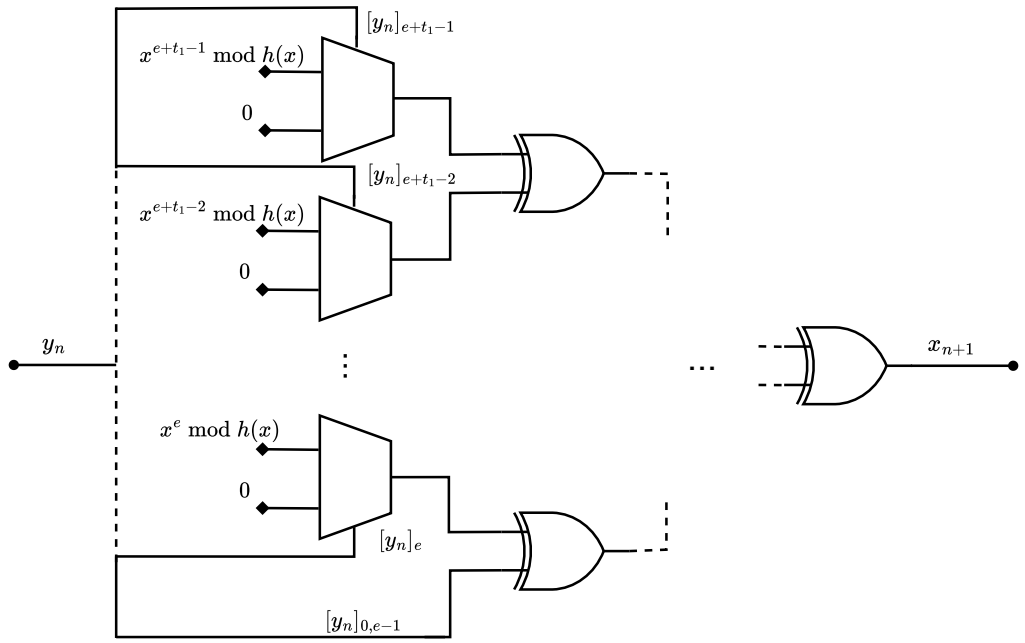

Fig. 2
DIAGRAMA LÓGICO DA REDUÇÃO MODULAR $x_{n+1} = y_n \bmod h(x)$, EM QUE $[y_n]_j$ DENOTA O COEFICIENTE DE x^j EM y_n.

TABELA I
RESULTADOS DAS BATERIAS DO TESTE TESTU01 PARA DIFERENTES CONFIGURAÇÕES DO PRNG. $c(x) = 1$.

Parâmetros de Entrada		Resultados das Baterias							
$h(x)$	$a(x)$	FIPS	Alpha	BlkAlpha	PsDH	Rabbit	SmCrsh	Crsh	BgCrsh
$x^{31}+x^{13}+x^8+x^3+1$	x^4+1	✓	×	×	×	×	×	×	×
$x^{31}+x^{22}+x^{15}+x^4+1$	$x^{30}+x^{18}+x^7+x^1+1$	✓	✓	✓	✓	×	×	×	×
$x^{128}+x^{109}+x^{99}+x^{44}+1$	$x^{112}+x^{101}+x^{60}+x^9+1$	✓	✓	✓	✓	×	✓	×	×
$x^{256}+x^{165}+x^{89}+x^{35}+1$	$x^{245}+x^{106}+x^{98}+x^{11}+1$	✓	✓	✓	✓	×	✓	×	×
$x^{113}+x^{83}+x^{72}+x^{59}+1$	$x^{94}+x^{90}+x^{25}+x^1+1$	✓	✓	✓	✓	✓	✓	×	×
$x^{243}+x^{116}+x^{115}+x^{58}+1$	$x^{179}+x^{160}+x^{75}+x^{33}+1$	✓	✓	✓	✓	✓	✓	✓	×
$x^{291}+x^{240}+x^{202}+x^{178}+1$	$x^{245}+x^{99}+x^{64}+x^4+1$	✓	✓	✓	✓	✓	✓	✓	×
$x^{451}+x^{402}+x^{292}+x^9+1$	$x^{406}+x^{360}+x^{168}+x^7+1$	✓	✓	✓	✓	✓	✓	✓	✓

TABELA II
COMPLEXIDADE DO CIRCUITO PARA DIFERENTES CONFIGURAÇÕES DO PRNG. $c(x) = 1$.

Parâmetros de Entrada		Complexidade do Circuito		
$h(x)$	$a(x)$	Portas XOR	Elementos MUX	Caminho Crítico
$x^{31}+x^{13}+x^8+x^3+1$	x^4+1	159	124	$5\tau_{\text{XOR}}+\tau_{\text{MUX}}$
$x^{31}+x^{22}+x^{15}+x^4+1$	$x^{30}+x^{18}+x^7+x^1+1$	1.174	930	$8\tau_{\text{XOR}}+\tau_{\text{MUX}}$
$x^{113}+x^{83}+x^{72}+x^{59}+1$	$x^{94}+x^{90}+x^{25}+x^1+1$	11.450	10.622	$10\tau_{\text{XOR}}+\tau_{\text{MUX}}$
$x^{128}+x^{109}+x^{99}+x^{44}+1$	$x^{112}+x^{101}+x^{60}+x^9+1$	15.296	14.336	$10\tau_{\text{XOR}}+\tau_{\text{MUX}}$
$x^{243}+x^{116}+x^{115}+x^{58}+1$	$x^{179}+x^{160}+x^{75}+x^{33}+1$	45.185	43.497	$11\tau_{\text{XOR}}+\tau_{\text{MUX}}$
$x^{256}+x^{165}+x^{89}+x^{35}+1$	$x^{245}+x^{106}+x^{98}+x^{11}+1$	64.724	62.720	$11\tau_{\text{XOR}}+\tau_{\text{MUX}}$
$x^{291}+x^{240}+x^{202}+x^{178}+1$	$x^{245}+x^{99}+x^{64}+x^4+1$	73.439	71.295	$11\tau_{\text{XOR}}+\tau_{\text{MUX}}$
$x^{451}+x^{402}+x^{292}+x^9+1$	$x^{406}+x^{360}+x^{168}+x^7+1$	204.755	203.401	$12\tau_{\text{XOR}}+\tau_{\text{MUX}}$

grau e o termo constante. Para limitar a complexidade das buscas computacionais, fixamos o peso de Hamming em 5. Uma vez definido $h(x)$, buscamos um polinômio $a(x)$ que seja uma raiz primitiva do corpo. Análises iniciais indicam que

diferentes polinômios $c(x)$ não oferecem vantagens estatísticas significativas, portanto escolhemos $c(x) = 1$.

A Tabela I mostra os resultados das baterias do TestU01 para diferentes escolhas de $h(x)$ e $a(x)$. Para cada configuração apresentada na tabela, múltiplas condições iniciais foram testadas para garantir a consistência dos resultados. No caso de $h(x)$ com grau $e = 31$, observa-se na primeira linha da tabela que para $a(x)$ com grau 4 e peso 2 todos os testes falham exceto o FIPS. Este comportamento também é observado para graus maiores de $h(x)$, não apresentado na tabela. A segunda linha da tabela indica uma melhoria significativa ao se utilizar um polinômio $a(x)$ de maior grau e peso 5, obtendo a aprovação em quatro baterias de testes.

Consideramos então polinômios $h(x)$ com graus dados por potências de 2 ($e = 128$ e $e = 256$), uma escolha natural para implementações computacionais. Observa-se que, embora estas configurações tenham demonstrado melhor desempenho, superando o SmallCrush, ainda falham na bateria Rabbit. Investigações adicionais, não apresentadas na tabela, com polinômios $a(x)$ com peso de Hamming maior que 5 não resultam em melhorias.

Empiricamente, observamos que polinômios de grau ímpar igual ou superior a 113 consistentemente superam a bateria Rabbit, como exemplificado na quinta linha da tabela. Extensivos testes com diversos graus ímpares entre 113 e 243 (não mostrados na tabela) confirmam esta tendência. O grau 243 emerge como o menor valor encontrado capaz de gerar sequências que aprovam em todos os testes, incluindo a rigorosa bateria Crush. Adicionalmente, o grau 291 é o primeiro a superar a bateria de testes Crush. Por fim, o grau 451 representa o maior grau investigado em nossa análise, também superando todas as baterias de testes e confirmando a robustez estatística das configurações com graus elevados.

Esta dependência das propriedades estatísticas das sequências geradas com os graus dos polinômios estabelece um compromisso entre complexidade do circuito e robustez estatística. A Tabela II apresenta as métricas de complexidade consideradas para as configurações do PRNG mostradas na Tabela I. O aumento da complexidade torna-se evidente com o crescimento dos graus dos polinômios.

Por fim, é importante ressaltar que todas as configurações apresentadas na Tabela I também foram submetidas à bateria de testes NIST SP 800-22, sendo aprovadas em todos os testes estatísticos desta suíte.

VII. Conclusões

Este trabalho apresentou uma nova estrutura de PRNGs baseada em recorrências lineares sobre corpos de extensão GF(2^e), com três contribuições principais: análise teórica do período, proposta de implementação em hardware e validação estatística.

A análise teórica demonstrou que o período máximo de $2^e - 1$ é alcançado quando $a(x)$ é um elemento primitivo do corpo e a condição inicial x_0 é diferente do único ponto fixo da recorrência. O diagrama lógico proposto utiliza estruturas de árvore binária balanceada, resultando em um circuito com crescimento quadrático no número de portas lógicas, mas com caminho crítico crescendo apenas forma logarítmica com os graus dos polinômios.

Os resultados experimentais revelaram uma relação importante entre o grau dos polinômios e a qualidade estatística das sequências geradas. Apresentamos uma seleção de configurações que demonstram diferentes compromissos entre complexidade de implementação e robustez estatística, desde implementações mais simples com grau 31 até configurações mais robustas com grau 451.

A estrutura proposta é especialmente adequada para implementações em hardware dedicado, onde o crescimento logarítmico do caminho crítico permite alta frequência de operação mesmo com polinômios de graus elevados. Trabalhos futuros podem explorar otimizações adicionais no circuito, como técnicas de pipeline e paralelização, além de uma implementação em Field-Programmable Gate Array (FPGA) para validação experimental das frequências máximas de operação previstas pela análise teórica do caminho crítico.

Referências

[1] A. A. Rezk, A. H. Madian, A. G. Radwan, and A. M. Soliman, "Multiplierless chaotic pseudo random number generators," *AEÜ Int. J. Electron. Commun.*, vol. 113, p. 152947, Jan. 2020.

[2] N. T. Nguyen, T. Bui, G. Gagnon, P. Giard, and G. Kaddoum, "Designing a pseudorandom bit generator with a novel five-dimensional-hyperchaotic system," *IEEE Trans. Ind. Electron.*, vol. 69, no. 6, pp. 6101–6110, Jun. 2022.

[3] S. Li, Y. Liu, F. Ren, and Z. Yang, "Design of a high throughput pseudorandom number generator based on discrete hyper-chaotic system," *IEEE Trans. Circuits and Syst. II: Exp. Briefs*, vol. 70, no. 2, pp. 806–810, Feb. 2023.

[4] Y. Luo, C. Fan, C. Xu, and X. Li, "Design and FPGA implementation of a high-speed PRNG based on an n-D non-degenerate chaotic system," *Chaos, Solitons & Fractals*, vol. 183, p. 114951, Jun. 2024.

[5] L. Kocarev, J. Szczepanski, J. M. Amigo, and I. Tomovski, "Discrete chaos-I: Theory," *IEEE Trans. Circuits and Syst. I: Regular Papers*, vol. 53, no. 6, pp. 1300–1309, June 2006.

[6] C. E. C. Souza, D. P. B. Chaves, and C. Pimentel, "One-dimensional pseudo-chaotic sequences based on the discrete Arnold's cat map over $\mathbb{Z}_{3^m}$," *IEEE Trans. Circuits and Syst. II: Exp. Briefs*, vol. 68, no. 1, pp. 491–495, Jan. 2021.

[7] C. E. C. Souza, D. Moreno, D. P. B. Chaves, and C. Pimentel, "Pseudo-chaotic sequences generated by the discrete Arnold's map over $\mathbb{Z}_{2^m}$: Period analysis and FPGA implementation," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.

[8] V. B. E. Mebenga, V. R. Kopparthi, H. D. Nzeuga, J. A. E. Fouda, G. M. D. Dagoumguei, G. B. Bitjoka, P. Rangababu, and S. L. Sabat, "An 8-bit integer true periodic orbit PRNG based on delayed Arnold's cat map," *AEÜ - Int. J. Electron. Commun.*, vol. 162, p. 154575, Apr. 2023.

[9] B. Chirikov and F. Vivaldi, "An algorithmic view of pseudochaos," *Physica D: Nonlinear Phenomena*, vol. 129, no. 3, pp. 223–235, May 1999.

[10] L. E. Bassham III *et al.*, "SP 800-22 Rev. 1a. A statistical test suite for random and pseudorandom number generators for cryptographic applications," Nat. Inst. Std. & Technol., Gaithersburg, MD, United States, Tech. Rep., 2010.

[11] P. L'ecuyer and R. Simard, "TestU01: A C library for empirical testing of random number generators," *ACM Trans. Math. Softw.*, vol. 33, no. 4, pp. 1–40, 2007.