# Cifragem de Imagens usando Cifras de Bloco e Sequências Caóticas

José A. P. Artiles, Daniel P. B. Chaves, Cecilio Pimentel

***Resumo*— A cifragem de dados com alta correlação, como imagens, é um desafio para cifras de bloco, já que padrões característicos dos dados poderão transparecer após cifrados. Isto decorre do mapeamento determinístico realizado pela cifra. Considerando a cifra AES como referência, neste trabalho é apresentada uma técnica de randomização desta cifra, empregando mapas caóticos, que alcança bons indicadores de segurança e robustez com menos iterações do algoritmo. Também é proposto um procedimento para reduzir a complexidade da cifragem.**

***Palavras-Chave*— Cifra de Bloco, cifragem de imagens, mapas caóticos, padrão avançado de criptografia.**

***Abstract*— Encryption of data with high correlation, such as images, is a challenge for block ciphers, since patterns of the original image may remain after encryption. This is due to the deterministic mapping performed by the cipher. Considering the AES cipher as reference, it is presented in this work a randomization technique of this cipher, employing chaotic maps, that achieves good security and robustness indicators with fewer iterations of the algorithm. A procedure is also proposed to reduce the complexity of encryption.**

***Keywords*— Block chipher, imagem encription, chaotic maps, advanced encryption standard.**

## I. Introdução

O AES (*Advanced Encryption Standard*) é o atual algoritmo de cifra de bloco recomendado pelo NIST (*National Institute of Standards and Technology*), sendo inicialmente indicado para proteção de informação digital não classificada de agentes governamentais, ou para a proteção de informação digital de agentes privados [1]. A partir de 2003 a NSA (US *National Security Agency*) passou a permitir o emprego do AES para proteção de informação classificada de agentes governamentais para comprimentos de chave de 192 e 256 *bits*. Uma das etapas mais importantes deste algoritmo é a substituição, que proporciona a confusão do texto cifrado [2]. Esta etapa é realizada por uma unidade denominada de S-box, que realiza um mapeamento determinístico entre *bytes* definido por uma sequência de operações em $GF(2^8)$. Além disso, para cifras de bloco em geral, as S-boxes não são suficientemente seguras contra ataques de criptoanálise diferencial devido a sua arquitetura rígida [3]. Portanto, técnicas para melhorar a segurança dessa unidade têm impacto preminente na segurança da cifra de bloco como um todo.

Neste trabalho, o AES é aplicado diretamente como unidade de cifragem, o equivalente a utilizá-lo no modo de operação EBC (*electronic code book*). Existem algumas fraquezas associadas a este modo de operação, visto que a cifragem é realizada de forma determinística [4]. Isto significa que blocos de texto claros idênticos são cifrados em blocos de texto cifrado idênticos, quando a mesma chave é utilizada. Esse característica pode ser contornada com o uso de cifragem probabilística ou cifras randomizadas [4]; quando cifragens de um mesmo bloco de texto claro geram possivelmente blocos de texto cifrado distintos, para a mesma chave. Essa característica pode ser gerada com o emprego de outros modos operação, tais como: CBC-*cipher block chaining*, CFB-*cipher feedback* e OFB-*output feedback*. Contudo, técnicas de criptoanálise com texto claro escolhido mais sofisticadas comprometem a segurança do sistema [5]. Como alternativa, pode-se empregar fontes de entropia descorrelacionadas do texto claro e da chave, *e.g.*, mapas caóticos; estratégia adotada neste trabalho.

O objetivo deste trabalho é projetar uma S-box randomizada para algoritmo AES empregando mapas caóticos. Nesse contexto, há três contribuições principais neste artigo: uma técnica para agregar uma fonte de entropia ao AES de forma compatível com o algoritmo original, ou seja, permitindo o seu uso em modo randomizado ou não randomizado; pela aplicação de uma série de testes de aleatoriedade e robustez contra criptoanálise, demonstra-se que a proposta atinge com menos iterações que o AES original níveis satisfatórios de segurança; quantifica-se a possível redução de complexidade no algoritmo AES ao empregá-lo em modo randomizado. Para a análise de desempenho do sistema proposto, são utilizadas imagens como fonte de informação.

O restante deste trabalho está organizado em seis seções. Na Seção II são descritos o algoritmo AES original e o mapa caótico utilizado neste trabalho. Na Seção III a modificação proposta é apresenta. A Seção IV descreve os testes realizados e discorre sobre os respectivos resultados. A Seção V faz uma análise do algoritmo modificado, em termos de difusão da chave e da criptoanálise diferencial. A eliminação de uma etapa do algoritmo AES original é proposta na Seção VI. As conclusões deste trabalho são apresentadas na Seção VII.

## II. Preliminares

### A. Resumo do Algoritmo AES

A estrutura do Algoritmo de Rijndael adotado pelo AES utiliza blocos de informação de 128 *bits* e chaves de cifragem de 128, 192, 256 *bits*, variando a quantidade de iterações dependendo do comprimento da chave. Neste trabalho utiliza-se o AES com chave $k_0$ de 128 *bits*, portanto, empregando 10 iterações [4], operando no modo ECB. O processo de cifragem é descrito convertendo-se o bloco de 128 *bits* de entrada em uma matriz de estado $4 \times 4$ em que cada elemento é um *byte*. De forma similar, a chave também é representada por uma

Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco, Recife-PE, Brasil, E-mails: joseantonio.artiles@ufpe.br, daniel.chaves@ufpe.br, cecilio@ufpe.br. Este trabalho foi parcialmente financiado pelo CNPq e FACEPE.

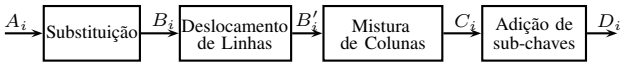



Fig. 1. Diagrama em blocos de uma iteração típica do AES.

matriz $4\times4$. O algoritmo apresenta quatro blocos, como ilustra a Fig. 1. O conjunto de operações realizadas em cada iteração é descrito a seguir.

*1)* **Substituição de *byte***: Cada *byte* da matriz de estado, que é a entrada de uma S-box, é substituído por um *byte* obtido por operações realizadas em $GF(2^8)$, um corpo gerado pelo polinômio primitivo $P(x) = x^8 + x^4 + x^3 + x + 1$. Para a $i$-ésima iteração, a saída de uma S-box $B_i$ está relacionada com a entrada $A_i$ via

$$B_i = S(A_i) = M \cdot A_i^{-1} + N \tag{1}$$

em que $M$ é uma matriz $8 \times 8$ binária, $A_i^{-1}$ é o inverso multiplicativo de $A_i$ no corpo e $N$ é um vetor $8 \times 1$ binário não nulo. A operação produto em (1) é realizada bit a bit, de modo que, $A_i^{-1}$ é empregado como um vetor $8 \times 1$. Em cada iteração existem 16 S-boxes idênticas no caminho de dados que operam de forma paralela, cada uma com 1 *byte* de entrada e de saída.

*2)* **Deslocamento das linhas**: Após a etapa de substituição, a $i$-ésima linha da matriz de estados, $i = 0, 1, 2, 3$, é deslocada ciclicamente de $i$ *bytes* para a direita.

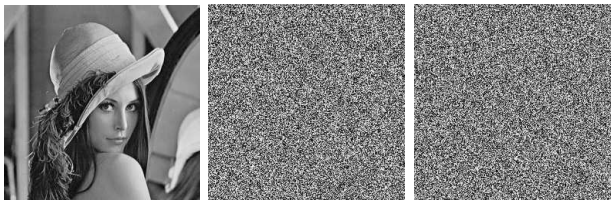
*3)* **Mistura de colunas**: Este processo faz uma transformação linear nas colunas da matriz de estado. Os 4 *bytes* de cada coluna desta matriz são considerados um vetor coluna que é multiplicado em $GF(2^8)$ por uma matriz $4 \times 4$ como indicado a seguir

$$\begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} B'_0 \\ B'_1 \\ B'_2 \\ B'_3 \end{bmatrix}. \tag{2}$$

*4)* **Adição de sub-chave**: Nesta operação, a matriz de estado é somada em $GF(2^8)$ com a matriz de sub-chave, formando-se uma nova matriz de estado usada na próxima iteração. A sub-chave $k_i$, $i = 1, 2, \cdots, 10$, usada em cada iteração é obtida de um processo realizado a partir da chave $k_0$. Neste processo, realiza-se várias operações, nas quais encontra-se a utilização de quatro S-boxes.

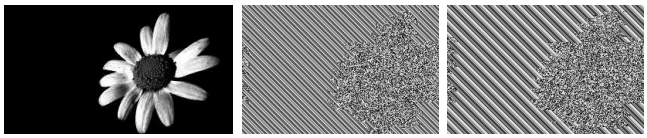
Em cada iteração realiza-se os mesmos conjunto de operações, exceto na primeira em que a chave $k_0$ é somada com os dados de informação antes do bloco de substituição, e na décima em que o bloco mistura de colunas não é utilizado.

A Fig. 2 ilustra o resultado da cifragem da imagem da Lena pelo algoritmo AES. Observa-se que a imagem depois de uma iteração apresenta uma saída aparentemente aleatória. Por outro lado, quando a imagem tem alta correlação, o algoritmo AES mantém padrões da imagem original, mesmo após a décima iteração, conforme descrito em [4, p. 141] e ilustrado na Fig. 3 para a imagem da flor.



(a) (b) (c)

Fig. 2. Imagem da Lena cifrada pelo algoritmo AES, (a) imagem original, (b) imagem cifrada após a primeira iteração, (c) imagem cifrada após a décima iteração.



(a) (b) (c)

Fig. 3. Imagem da flor cifrada pelo algoritmo AES, (a) imagem original, (b) imagem cifrada após a primeira iteração, (c) imagem cifrada após a décima iteração.

### *B. Mapas Caóticos*

O comportamento dos mapas caóticos unidimensionais é observado mediante uma série temporal discreta $\{x_i\}_{i=0}^{\infty}$, obtida pela iteração de uma função não-linear $f(x)$, sob uma condição inicial $x_0$, da seguinte forma:

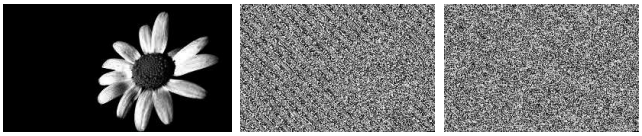
$$x_n = f(x_{n-1}), n = 1, 2, 3, \ldots \tag{3}$$

Denomina-se $\{x_n\}_{i=0}^{\infty} = \{x_0, f(x_1), f(x_2), \ldots\}$ uma órbita de $f$ iniciando em $x_0$. O mapa $f : [-1, 1] \rightarrow [-1, 1]$ usado neste trabalho é o mapa cúbico, para o qual $f(x) = 4x^3 - 3x$ [6]. O valor de $x_0$ é dado pelos 128 *bits* da chave original $k_0$. Esta é dividida em blocos de 16 *bytes*, $v_1 v_2 \cdots v_{16}$ [7], e calcula-se $v'_0 = \sum_{i=1}^{16} \frac{v_i}{256}$. Então, $x_0 = v'_0 - \lfloor v'_0 \rfloor$, em que $v_i$ e $\lfloor \cdot \rfloor$ são, respectivamente, o equivalente decimal do $i$-ésimo bloco e a função piso. Gera-se a partir de $x_0$ uma órbita finita usando (3), em que as primeiras 200 amostras são eliminadas devido ao transiente da órbita, e usando-se uma quantização em 2 níveis, gera-se uma sequência binária caótica $\{z_k\}$.

## III. AES Modificado com o Mapa Caótico

Nesta seção o algoritmo AES1 é apresentado, que é idêntico ao AES exceto pela soma à saída das S-boxes de um *byte* gerado a partir de um mapa caótico. Apresentamos esta proposta com 3 *bits* caóticos por S-box, podendo ser verificada uma melhoria nos indicadores de aleatoriedade e robustez contra criptoanálise com o aumento deste número de *bits*.

Para cada bloco de informação, a sequência caótica $\{z_k\}$ é seccionada em 60 *bits*, em que os primeiros 48 *bits* são usados nas 16 S-boxes da etapa de substituição, enquanto os 12 *bits* restantes são usados nas 4 S-boxes da unidade de geração de sub-chaves. Para cada S-box, utiliza-se três *bits* caóticos $(z_j, z_{j+1}, z_{j+2})$, com representação polinomial dada por $c(x) = z_j x^2 + z_{j+1} x + z_{j+2}$. Como as operações de cada S-box são definidas em $GF(2^8)$, multiplica-se $c(x)$ por um polinômio primitivo em $GF(2^5)$, obtendo assim o polinômio $h(x) = c(x)p(x) \mathrm{mod} P(x)$, em que $P(x)$ é apresentado na



(a) (b) (c)

Fig. 4. Imagem da flor cifrada pelo algoritmo AES1, (a) imagem original, (b) imagem cifrada após a segunda iteração, (c) imagem crifrada após a terceira iteração.

Subeção II-A. Os coeficientes deste polinômio formam um *byte* denominado $\mathbf{h}$. A escolha de $p(x)$ é feita para que $\mathbf{h}$ seja o mais balanceado possível entre zeros e uns. Neste trabalho, o polinômio escolhido via teste exaustivo é $p(x) = x^5 + x^4 + x^3 + x + 1$. Este *byte* é somado (XOR bit a bit) com o *byte* da saída da S-box na etapa de substituição de *bytes*. Na segunda iteração, os 48 *bits* caóticos obtidos na primeira iteração são utilizados novamente fazendo-se um deslocamento cíclico à direita dado pelo valor na base dez dos últimos três bits utilizados na última S-box. Por exemplo, se o polinômio da última S-box é $c(x) = x^2 + x + 1$, faz-se um deslocamento cíclico de 7 *bits* na sequência de 48 *bits* obtida do mapa caótico. Este procedimento é repetido em cada iteração. O deslocamento é feito para evitar a rigidez das S-boxes, que levam a um comportamento determinístico, mantendo os mesmos 48 *bits* caóticos por bloco de informação. Na obtenção das sub-chaves, um procedimento similar é realizado nas S-boxes usadas para obtenção das sub-chaves $k_i$.

A Fig. 4 mostra o resultado da cifragem da imagem da flor utilizando o sistema proposto. Observa-se que com três iterações a saída apresenta uma aparência aleatória, o que é esperado de um bom algoritmo criptográfico. As próximas seções trazem as análises quantitativas desta proposta, entre as quais são medidas: aleatoriedade do texto cifrado; resistência a ataques; sensibilidade a chave; correlação do texto cifrado.

## IV. Resultados

Esta seção inicia com a descrição de alguns quantificadores comumente usados para avaliar tanto a aleatoriedade da imagem cifrada como a capacidade de resistência à ataques estatísticos [8]–[10].

*1)* **Entropia** (H): Uma medida de aleatoriedade é dada pela entropia da fonte, definida por:

$$H = -\sum_{i=1}^{L} \Pr(m_i) \log \Pr(m_i) \tag{4}$$

em que $\Pr(m_i)$ é a probabilidade do símbolo $m_i$ e $L$ é o número total de símbolos. Como neste trabalho utilizamos imagens com 256 níveis de cinza, o valor máximo da entropia da fonte é 8 *bits*.

*2)* **Ataque diferencial** (A-D): Para analisar o efeito na imagem cifrada com a mudança de um *byte* na imagem original, utiliza-se o número da taxa de mudança de pixels (NPCR, *number of pixeis change rate*) e a intensidade de alteração média unificada (UACI, *unified average changing intensity*) [10]. Sejam $C_1$ e $C_2$ duas imagens cifradas de dimensão $W \times H$, tal que suas imagens originais diferem em um *byte*. O valor do $(i, j)$-th pixel (variando entre 0 e 255) de $C_1$ e de $C_2$ é denotado por $C_1(i, j)$ e $C_2(i, j)$, respetivamente. O NPCR e o UACI entre as imagens $C_1$ e $C_2$ são definidos por:

$$\text{NPCR} = \frac{\sum_{ij} D(i,j)}{W \times H} \times 100 \tag{5}$$

$$\text{UACI} = \frac{1}{W \times H} \sum_{ij} \frac{|C_1(i,j) - C_2(i,j)|}{255} \times 100 \tag{6}$$

em que $D(i, j)$ é igual a 1 (um) se $C_1(i, j) = C_2(i, j)$, ou igual a 0 (zero) se $C_1(i, j) \neq C_2(i, j)$. A cifra apresenta uma boa capacidade de suportar um ataque diferencial se, em média, os valores de NPCR e UACI estão acima de $99\%$ e $33\%$, respetivamente [10].

*3)* **Sensibilidade a chave** (S-C): Um bom algoritmo criptográfico deve ter alta sensibilidade a chave $k_0$. Neste caso, uma pequena diferença na chave original conduz a uma grande mudança nas imagens decifradas. Este teste é realizado da seguinte forma:

- Etapa 1. Cifrar a imagem com uma chave $k_0$.
- Etapa 2. Alterar a chave em um *bit* (a chave alterada é denominada de $k_0'$). Existem 128 possíveis $k_0'$ para cada chave original $k_0$.
- Etapa 3. Calcular a fração de bits média que mudam entre a imagem cifrada com a chave $k_0$ e a mesma imagem cifrada com cada uma de suas 128 variações (cifradas com $k_0'$).

A sensibilidade a chave é calculada como a média do valor obtido na Etapa 3 para 400 valores distintos de $k_0$.

*4)* **NIST**: Empregamos a bateria de teste NIST (versão 800-22) [11] para testar se uma sequência é adequada para aplicações criptográficas. Os testes são utilizados para determinar a aceitação ou rejeição da hipótese de aleatoriedade ideal com nível de significância $\alpha$. Neste trabalho adotamos $\alpha = 0,01$, ou seja, uma sequência é aprovada com nível de significância de $99\%$. Uma sequência binária que representa uma imagem cifrada é a entrada da bateria de teste NIST.

*5)* **Análise da correlação** (A-C): O coeficiente de correlação $\rho$ entre dois pixels adjacentes $a$ e $b$ é definido por:

$$\rho = \frac{\text{cov}(a, b)}{D(a)D(b)} \tag{7}$$

em que $\text{cov}(a, b)$ denota a covariância entre os pixels $a$ e $b$ e $D(i)$ denota a variância do pixel $i$. Este quantificador é calculado para três disposições dos pixels: na horizontal (Hrt), na vertical (Vrt) e na diagonal (Dgn).

A Tabela I mostra os resultados dos quantificadores apresentados nesta seção para a imagem da flor cifrada pelo algoritmo AES para valores selecionados do número de iterações (Itr na tabela). Observa-se que a entropia não alcança um valor próximo de 8 *bits*, mesmo na décima iteração, bem como não ocorre aprovação nos testes da bateria NIST (representado por um símbolo X na tabela). Observa-se ainda uma correlação significativa entre os pixels adjacentes (indicada pelos valores de A-C na tabela). Convém destacar que os quantificadores A-D e S-C produzem resultados satisfatórios após a terceira iteração.

TABELA I

DESEMPENHO DO ALGORITMO AES PARA IMAGEM DA FLOR.

| Itr | H | A-D | | S-C | NIST | A-C | | |
|---|---|---|---|---|---|---|---|---|
| | | NCPR | UACI | | | Hrt | Vrt | Dgn |
| 1 | 6,13 | 47,8 | 13,4 | 16,9 | X | 0,54 | 0,52 | 0,51 |
| 3 | 6,21 | 99,5 | 33,2 | 49,7 | X | -0,45 | -0,47 | 0,42 |
| 6 | 6,22 | 99,6 | 33,2 | 49,9 | X | 0,4 | -0,38 | -0,39 |
| 10 | 6,23 | 99,8 | 33,3 | 49,9 | X | -0,35 | 0,31 | -0,32 |

TABELA II

DESEMPENHO DO ALGORITMO AES1 PARA IMAGEM DA FLOR.

| Itr | H | A-D | | S-C | NIST | A-C | | |
|---|---|---|---|---|---|---|---|---|
| | | NCPR | UACI | | | Hrt | Vrt | Dgn |
| 1 | 7,94 | 99,7 | 33,4 | 49,7 | S | -0,19 | 0,18 | 0,2 |
| 2 | 7,94 | 99,7 | 33,4 | 49,7 | S | 0,12 | 0,1 | 0,14 |
| 3 | 7,96 | 99,8 | 33,4 | 49,7 | S | -0,057 | 0,041 | 0,054 |
| 4 | 7,97 | 99,8 | 33,4 | 49,8 | S | 0,04 | -0,039 | -0,041 |
| 6 | 7,99 | 99,8 | 33,4 | 49,9 | S | 0,025 | 0,022 | -0,024 |
| 10 | 7,99 | 99,8 | 33,4 | 49,9 | S | -0,015 | -0,012 | 0,011 |

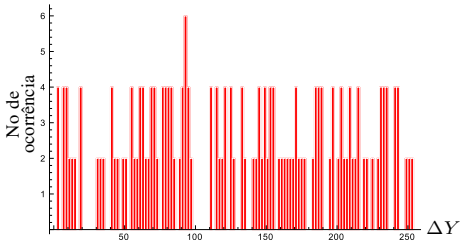



Fig. 5. Criptoanálise diferencial da S-box do AES.

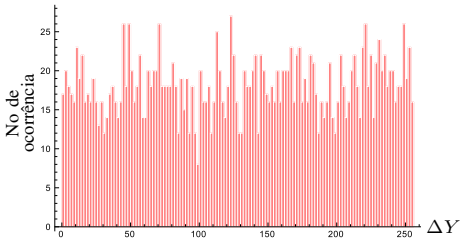



Fig. 6. Criptoanálise diferencial da S-box do AES1.

O desempenho do algoritmo AES1 para a imagem da flor é mostrado na Tabela II. Observa-se uma melhoria significativa em relação aos resultado do AES mostrado na Tabela I, sendo que após a primeira iteração o valor da entropia encontra-se próxima ao valor máximo, bem como o sistema é aprovado na bateria de teste NIST (representada pelo símbolo S na tabela), mantendo bons indicadores A-D e S-C. Após a segunda iteração observa-se ainda valores significativos de correlação entre os pixels, o que também pode ser observado na Fig. 4-(b). Após a terceira iteração há uma redução significativa da correlação para as três disposições consideradas, o que é observado qualitativamente na Fig. 4-(c).

A partir da comparação entre as Tabelas I e II, conclui-se que a entropia adicionada pelos *bits* caóticos através das alterações propostas para as S-boxes (tanto aquelas no caminho de dados quanto as contidas na unidade de geração de sub-chaves), permite alcançar bons indicadores de aleatoriedade e robustez contra criptoanálise com um menor número de iterações.

## V. CRIPTOANÁLISE DIFERENCIAL E DIFUSÃO DA CHAVE

A redução do número de iterações acarreta uma fraqueza da cifra de bloco para ataques com texto claro escolhido, ou mesmo texto claro conhecido, em decorrência do processo deficitário de difusão e confusão. Este comportamento pode ser verificado pela criptoanálise diferencial e pela análise da difusão na obtenção das sub-chaves. Esta seção é dedicada à condução dessas análises.

### A. Criptoanálise Diferencial

A ideia básica utilizada na criptoanálise diferencial é tabular diferenças específicas na entrada de uma S-box que levam a diferenças específicas na sua saída com probabilidade maior do que seria esperado para uma permutação aleatória, em que ocorrem todas as possíveis saídas com a mesma probabilidade. Define-se a diferença $\Delta X$ entre duas entradas de uma S-Box $A_i$ e $A_i'$ (sequências de 8 *bits*), como $\Delta X = A_i \oplus A_i'$, em que $\oplus$ denota adição em $GF(2)$. Com o uso da chave $k_0$, se obtêm uma diferença na saída $\Delta Y = S(A_i) \oplus S(A_i')$, em que $S(\cdot)$ é a função que define a permutação de entrada e saída da S-Box, com uma probabilidade $p(\Delta Y)$. Para uma cifra de bloco fraca, podem haver diferenças nas saídas que ocorrem com alta probabilidade enquanto algumas diferenças podem não ocorrer. Este comportamento pode ser explorado para diminuir o espaço de chaves do sistema [4].

Fixando um valor de $A_i$ e variando todos os possíveis 256 valores de $A_i'$ calcula-se os possíveis $\Delta Y$, como mostra a Fig. 5. Observa-se que só ocorrem aproximadamente a metade dos possíveis valores de $\Delta Y$. Este comportamento é similar para qualquer valor de $A_i$ fixado. Para um valor de entrada $A_i$ fixado, a partir das probabilidades $p(\Delta Y)$ obtidas a partir da Fig. 5 usa-se (4) e obtém-se uma entropia igual a 6,86 *bits* (valor similar para qualquer $A_i$ fixado).

No algoritmo AES1, introduz-se o mapa caótico nas S-boxes. Nesse caso, para um determinado par de entrada $A_i$ e $A_i'$, oito possíveis valores de $\Delta Y$ são possíveis, com uma probabilidade de ocorrência similar, devido aos oito possíveis polinômios caóticos $h(x)$. Agora, para cada valor de $A_i$, deve-se calcular todos os possíveis valores de $\Delta Y$ para todos os valores de $A_i'$ e $h(x)$. A Fig. 6 mostra que todos os possíveis valores de $\Delta Y$ ocorrem e têm probabilidades mais equânimes do que o mostrado na Fig. 5. Para um valor de entrada $A_i$, a partir das probabilidades $p(\Delta Y)$, obtém-se uma entropia igual a 7,86 *bits*. O emprego do mapa caótico leva a um aumento da entropia, visto que para cada entrada das S-boxes existem oito possíveis saídas, aumentando o número de possíveis diferenças $\Delta Y$ e menor diferença nos valores de $p(\Delta Y)$.

### B. Difusão da Chave

Para medir a difusão na geração de sub-chaves, gera-se uma chave $k_0$ e determinam-se as sub-chaves $k_1, k_2, \cdots, k_{10}$, usadas em cada iteração do AES, a partir do sistema de geração de sub-chaves. Para cada $k_0$, derivam-se outras 128 chaves que diferem de $k_0$ por um *bit* e para cada uma destas são determinadas as respectivas sub-chaves. Para medir a difusão da chave, para cada $k_0'$, uma chave derivada de $k_0$, com respectivas sub-chaves $k_1', \cdots, k_{10}'$, calcula-se o número

TABELA III
DIFUSÃO NA GERAÇÃO DE CHAVES

| Itr | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| AES | 8,5 | 22,7 | 26,4 | 34,7 | 37,5 | 49,7 | 49,8 | 49,9 | 49,9 |
| AES1 | 49,7 | 49,8 | 49,8 | 49,8 | 49,9 | 49,9 | 49,9 | 49,9 | 49,9 |

TABELA IV
DESEMPENHO DO ALGORITMO AES2 PARA IMAGEM DA FLOR.

| Itr | H | A-D | | S-C | NIST | A-C | | |
|---|---|---|---|---|---|---|---|---|
| | | NCPR | UACI | | | Hrt | Vrt | Dgn |
| 1 | 7,3 | 99,7 | 33,1 | 49,7 | X | 0,21 | 0,203 | -0,22 |
| 3 | 7,94 | 99,8 | 33,2 | 49,8 | S | -0,11 | -0,14 | 0,13 |
| 5 | 7,95 | 99,8 | 33,3 | 49,8 | S | -0,096 | 0,11 | -0,103 |
| 6 | 7,98 | 99,8 | 33,3 | 49,8 | S | -0,041 | 0,033 | 0,032 |
| 7 | 7,98 | 99,8 | 33,3 | 49,9 | S | 0,029 | -0,028 | -0,027 |
| 10 | 7,99 | 99,8 | 33,3 | 49,9 | S | -0,021 | -0,019 | 0,015 |

médio de *bits* que diferem entre $k_i$ e $k_i'$. Com uma boa difusão, o número médio de *bits* distintos em cada iteração, para qualquer $k_0'$, deve ser 50%.

Neste trabalho são empregadas 200 diferentes chaves $k_0$ e calcula-se em média quantos *bits* estão mudando entre $k_i$ e $k_i'$, como mostra a Tabela III para os algoritmos AES e AES1. Observa-se que no algoritmo AES são necessárias seis iterações para se obter uma boa difusão (fração superior a 0,49 dos bits de $k_i'$ foram alterados para $i$ igual ou maior que 6). Com a utilização do algoritmo AES1, este nível de difusão é obtido na primeira iteração. A análise das Tabelas II e III indica que o AES1 atinge níveis satisfatórios de aleatoriedade e robustez contra criptoanálise com apenas três iterações, para a imagem da flor.

## VI. ALGORITMO AES2

A introdução de incerteza no sistema pelos *bits* caóticos permite simplificar o algoritmo AES, visando reduzir sua complexidade. A seguir, as consequências para a qualidade da cifragem será analisada quando o bloco mistura de colunas é eliminado. Esse bloco é responsável pela difusão de *bytes*, e exige um grande número de operações de soma módulo dois. Denominaremos por AES2 esta variação do AES1. A Tabela IV apresenta os resultados dos quantificadores em função do número de iterações, quando aplicados ao AES2. Observa-se que com seis iterações os quantificadores apresentam valores similares ao do algoritmo AES1 com três iterações, indicando a viabilidade deste algoritmo.

A seguir faremos uma comparação do número de somas (XOR) realizadas pelos algoritmos AES1 e AES2 por iteração. Para este cálculo, consideramos que os valores da saída das S-boxes $B_i$ são obtidos de uma tabela existente em uma memória. Os possíveis polinômios caóticos $h(x)$, obtidos do mapa caótico, também são guardados em uma memória. Desta forma, não são contabilizadas as operações feitas para obtenção dos polinômios $h(x)$ e das saídas das S-boxes.

Os algoritmos AES1 e AES2 fazem oito somas (XOR) dos *bits* caóticos com $B_i$ em cada S-box. No bloco de substituição de *byte* existem 16 S-boxes, então em uma iteração realizam-se 128 somas (XOR) dadas pela adição dos *bits* caóticos.

Como foi mostrado em (2), o bloco de mistura de colunas realiza diferentes operações em $GF(2^8)$. A multiplicação por $(01)$, resulta na saída o mesmo valor da entrada. A multiplicação por $(02)$ é similar a fazer um deslocamento na sequência de entrada, e se o *bit* mais significativo de $B_i'$ é 1, faz-se uma soma (XOR) com a sequência $\{00011011\}$. A multiplicação por $(03)$ corresponde a fazer uma soma (XOR) dos valores obtidos nas duas multiplicações anteriores. Na obtenção dos valores de $C_i$, $i = 1 \cdots 4$, no bloco mistura de colunas, são feitas 160 somas (XOR). Em uma iteração existem quatro operações de mistura de colunas, então existem 640 somas (XOR) para a implementação deste bloco por iteração. No processo de geração de sub-chaves, realizam-se em cada iteração 168 somas (XOR) devido às operações nas S-boxes. Em cada iteração, o algoritmo AES1 realiza 936 somas (XOR) enquanto o AES2 faz 296 somas (XOR). No caso da imagem da flor, o algoritmo AES2 com 6 iterações realiza 1032 somas (XOR) a menos que o algoritmo AES1 com 3 iterações.

## VII. CONCLUSÕES

Um esquema de cifragem probabilística foi proposto com a introdução de bits caóticos na etapa de substituição de *bytes* e na geração de sub-chaves do algoritmo AES. Foi demonstrado que um algoritmo consagrado de cifra de bloco, quando associado adequadamente com mapas caóticos agindo como fonte de entropia, apresenta resultados que apontam para dois caminhos: redução do número de iterações e simplificação do algoritmo de cifragem. Como consequência, essa estratégia pode conduzir a menor custo energético e simplificação de hardware. Como trabalhos futuros, esses resultados serão corroborados com mais testes de robustez e a extensão para outras cifras de bloco.

## REFERÊNCIAS


[1] Federal Information Processing Standards (FIPS), "Advanced Encryption Standard (AES)," NIST, Tech. Rep. FIPS 197, November 2001. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

[2] H. Li, "Efficient and flexible architecture for aes," *IEE P-Circ Dev Syst*, vol. 153, no. 6, pp. 533–538, Dec. 2006.

[3] A. Bechtsoudis and N. Sklavos, "Side channel attacks cryptanalysis against block ciphers based on FPGA devices," in *2010 IEEE Computer Society Annual Symposium on VLSI*, July 2010, pp. 460–461.

[4] C. Paar and J. Pelzl, *Understanding Cryptography, A Textbook for Students and Practitioners*. Springer, 2010.

[5] E. Biham and A. Shamir, *Differential cryptanalysis of the data encryption standard*. Springer-Verlag, 1993.

[6] F. Lau and C. Tse, *Chaos-Based Digital Communication Systems*, ser. Engineering online library. Springer, 2010.

[7] N. Pareek, V. Patidar, and K. Sud, "Cryptography using multiple one-dimensional chaotic maps," *Commun Nonlinear Sci Numer Simul*, vol. 10, no. 7, pp. 715–723, 2005.

[8] B. Wang *et al.*, "Evaluating the permutation and diffusion operations used in image encryption based on chaotic maps," *Optik*, vol. 127, no. 7, pp. 3541–3545, 2016.

[9] Z. Zhenjun *et al.*, "Multiple-image encryption with bit-plane decomposition and chaotic maps," *Opt Laser Eng*, vol. 80, pp. 1–11, 2016.

[10] L. Wenhao, S. Kehui, and Z. Congxu, "A fast image encryption algorithm based on chaotic map," *Opt Laser Eng*, vol. 84, pp. 26–36, 2016.

[11] L. E. Bassham *et al.*, "Statistical test suite for random and pseudo random number generators for cryptographic applications," NIST, Tech. Rep. 800-22 Rev 1a, September 2010. [Online]. Available: http://ws680.nist.gov/publication/get_pdf.cfm?pub_id=906762