Identificação de chaves e algoritmos criptográficos utilizando Algoritmo Genético e Teoria dos Grafos

Renato Hidaka Torres*, Gláucio A. de Oliveira*, José A. M. Xexéo*‡, William A. R. Souza† e Ricardo Linden‡
*Seção de Engenharia de Computação
Instituto Militar de Engenharia, Rio de Janeiro-RJ, CEP 22290-270
Emails:renatohidaka, glaucioaorj@gmail.com, xexeo@ime.eb.br
† COPPE/UFRJ - Universidade Federal do Rio de Janeiro - Rio de Janeiro - RJ, CEP 21945-970
Email: william@cos.ufrj.br
‡Faculdade Salesiana Maria Auxiliadora, Macaé - RJ - CEP 27910-970
Emails:rlinden@pobox.com, josexexeo@gmail.com

***Abstract*—This paper describes the use of genetic algorithms that use the *Calisnki-Harabasz* index as its evaluation function and graphs techniques to identify patterns in cryptograms generated by cryptografics algorithms certified by NIST (National Institute Standard Technology), namely AES, RC6, MARS, Twofish and Serpent.**

Evidence of patterns or "signatures" generated by the algorithms under test are detected, thus corroborating the results of other studies quoted here. The results obtained with these two techniques are compared with results reported in [4] and [5], showing superiority in the accuracy of class generation.

I. Introdução

Os algoritmos criptográficos utilizados comercialmente como o DES, o RSA e, recentemente, o AES passaram por intensivas avaliações tanto do meio acadêmico quanto de órgãos como o NIST (National Institute of Standards and Technology). O objetivo dessas avaliações, tanto de algoritmos quanto dos produtos criptográficos, é garantir um mínimo de segurança no uso desses artefatos e demanda dos grandes usuários da criptografia.

O processo de certificação utilizado pelo NIST, quando da concorrência que elegeu o AES como padrão, baseou-se em testes estatísticos. Os relatórios desses ensaios induzem à conclusão de que a métrica utilizada para qualificar o nível de segurança dos algoritmos testados é função dos níveis de aleatoriedade determinados nos diversos ensaios.

[1] ao relatar os ensaios realizados com os finalistas do AES (Mars, RC6, Rijndael, Serpent e Twofish), estabeleceram duas assertivas: a) "embora se acredite que os cinco algoritmos finalistas geram sequências aleatórias, os testes foram realizados para mostrar que há evidências empíricas para apoiar essa convicção"; b) "os resultados sugerem que, apesar de anomalia detectada no Serpent, é lícito afirmar que todos os algoritmos parecem não ter desvios da aleatoriedade detectáveis". Quase simultaneamente, [2] discutiu a metodologia utilizada nesses ensaios e, embora não questionasse a certificação, concluiu seu relatório com diversas restrições, entre elas: a) "as hipóteses de teste não são suficientemente claras, gerando interpretações diferentes para os resultados"; b) "testes estatísticos equivalentes, realizados com os mesmos dados, não geram, necessariamente, os mesmos resultados". Havia, portanto, controvérsias se não pela qualidade dos algoritmos, pelo menos pela completude e propriedade dos ensaios realizados. Mais recentemente, diversos pesquisadores levantaram outra questões em relação aos resultados apresentados por [1].

Diversos tipos de ensaio detectaram padrões nos criptogramas gerados pelos algoritmos submetidos aos testes do NIST, que permitiram distinguir criptogramas tanto por algoritmo quanto por chave de origem. Esses resultados são importantes porque explicitam indícios da transmissão de "assinaturas" dos algoritmos e das chaves aos criptogramas. [3] realizam o chamado "ataque de distinção", para criptogramas gerados pelo algoritmo RC6 com o auxílio da estatística do qui-quadrado. [4], [5] e [6] agruparam criptogramas gerados pelas cifras de blocos DES, AES e RSA, em função da chave, com diversos tamanhos de criptogramas e chaves. [7] usaram Máquinas de Vetor de Suporte (SVM) para identificar as cifras de bloco DES, Triple DES, Blowfish, AES e RC5, a partir de conjuntos de criptogramas gerados por eles, [8] propuseram um método para fortalecer o algoritmo RC6, como uma contramedida ao ataque com a estatística do qui-quadrado. [9] desenvolveu métodos de histograma e de predição de bloco, técnicas de expansão de dados e técnicas baseadas em ataques secundários para identificação das cifras de bloco DES, AES, Blowfish, Triple DES e RC5. O resultado desses trabalhos reforçou a hipótese da existência de "assinaturas" transmitidas para os criptogramas pelas chaves e pelos próprios algoritmos.

Este trabalho tem o mesmo objetivo de identificar algoritmos e chaves a partir de padrões encontrados nos criptogramas gerados por eles. Entretanto, serão utilizadas duas técnicas diferentes das anteriores. A primeira modela o conjunto de criptogramas num Grafo e, a partir daí, com um algoritmo de complexidade $O(n^2)$, determina o conjunto de subgrafos existentes. Os subgrafos correspondem aos algoritmos ou chaves empregados no processo de cifrar os criptogramas em tela. Na segunda técnica, dada uma matriz de similaridades entre os criptogramas, o número de classes entre esses documentos cifrados será obtido por meio de um Algoritmo Genético implementado que utilizará o indíce *Calinski-Harabasz* (CH). O ponto máximo dessa função corresponde ao número de algoritmos ou chaves diferentes em teste. Ao atingir esse

ponto, o Algoritmo Genético pára e agrupa os criptogramas na quantidade de grupos correspondente ao valor desse máximo. O algoritmo, portanto, detecta quantos grupos "similares" há no conjunto de criptogramas e distribui os criptogramas por esses grupos. Em cada grupo deverão ser alocados somente - e todos eles - criptogramas gerados pelo mesmo algoritmo ou chave.

Uma das principais contribuições das técnicas apresentadas neste artigo é propor algoritmos de particionamento que consigam encontrar padrões em criptogramas com o objetivo de reduzir o custo computacional em trabalhos criptoanalíticos. Outras contribuições deste trabalho foram melhorar a qualidade das classes encontradas em relação a sua homogeneidade, comparando com os trabalhos de [4] e [5], além do fato dessas técnicas não precisarem de conhecimento anterior da quantidade de algoritmos (quantidade de classes) sendo avaliados. As demais seções deste artigo estão divididas da seguinte forma: A seção 2 apresenta a fase de pré-processamento necessária para obter os dados de entrada dos Algoritmos de Grafos e Genéticos. Na seção 3 é apresentada a modelagem do Grafo e na seção 4 a modelagem do Algoritmo Genético. Os experimentos serão apresentados na seção 5, descrevendo os ensaios e resultados. Por fim, a seção 6 conterá a conclusão do trabalho.

II. *Pré-Processamento*

A fase de pré-processamento é necessária para modelar os criptogramas em uma estrutura de dados que possa ser utilizada como dado de entrada pelos algoritmos de classificação não-supervisionada. Desta forma foi utilizado o modelo apresentado no trabalho de [4] e [5], que considera os criptogramas que compõem uma coleção como um espaço de vetores de *n*-dimensões, onde *n* é o número de blocos binários no universo dos criptogramas. Deste modo, sejam dois criptogramas 1 e 2 em que a frequência $f_{n,1}$ é relacionada ao *n-ésimo* bloco do criptograma 1 assim como a frequência $f_{n,2}$ é relacionada ao *n-ésimo* bloco do criptograma 2 . Então, o vetor para o criptograma 2 é definido como $\vec{C}_2$ = ($f_{1,2}$, $f_{2,2}$, .., $f_{n,2}$) e, da mesma forma, o vetor para o criptograma 1 é representado por $\vec{C}_1$ = ($f_{1,1}$,$f_{2,1}$, ...,$f_{n,1}$). Mostra-se, na Figura 1, o processo de formação dos vetores dos criptogramas. Constrói-se um "dicionário" com *n* blocos binários, gerados pelo processo de contagem dos próprios blocos. O tamanho de cada bloco pode ser determinado por qualquer divisor do tamanho da chave.

Para avaliar o grau de associação (similaridade) entre o criptograma 1 e o criptograma 2, faz-se uma correlação entre os seus vetores $\vec{C}_1$ e $\vec{C}_2$. Esta correlação pode ser quantificada pelo *cosseno* do ângulo entre esses dois vetores na equação (1).

$$cos(\vec{C}_1, \vec{C}_2) = \frac{\sum_{i=0}^{n}(\vec{C}_1 * \vec{C}_2)}{\sqrt{\sum_{i=0}^{n}\vec{C}_1^2 * \sum_{i=0}^{n}\vec{C}_2^2}} \quad (1)$$

Assim, a modelagem de cada criptograma em um vetor de blocos binários quantifica a frequência com que estes blocos aparecem no criptograma e nos permite determinar o grau de

Blocos binários	Criptogramas					
	$\vec{C}_1$	$\vec{C}_2$	$\vec{C}_3$	$\vec{C}_4$	.....	$\vec{C}_i$
11101001	$f_{1,1}$	$f_{1,2}$	$f_{1,3}$	$f_{1,4}$	.....	$f_{1,i}$
10101010	$f_{2,1}$	$f_{2,2}$	$f_{2,3}$	$f_{2,4}$	.....	$f_{2,i}$
:	:	:	:	:	.....	:
:	:	:	:	:	.....	:
:	:	:	:	:	.....	:
11100010	$f_{n,1}$	$f_{n,2}$	$f_{n,3}$	$f_{n,4}$	.....	$f_{n,i}$

Figura 1. Dicionário de blocos. $f_{n,i}$ é a frequência do *n*-ésimo bloco do *i*-ésimo criptograma da coleção de criptogramas

associação entre os criptogramas. Este grau é uma medida de similaridade entre um par de vetores. Deste modo, calcula-se a similaridade entre todos os criptogramas de uma determinada coleção gerando uma da matriz de similaridades. A figura 2 ilustra a fase do *pré-processamento*.

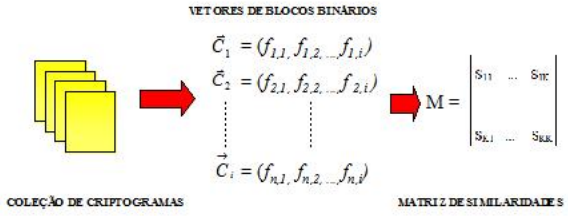

Figura 2. Fase de *Pré-processamento.*

III. Modelagem do Grafo

Dado um Grafo não direcionado G(V, E) em que $V = x_1, x_2, ..., x_l$ são os vértices e E são as arestas (x_i, x_j), cujo peso determina a similaridade entre os vértices aos quais ela está conectada. A estrutura de dados utilizada para a representação do Grafo foi a matriz de proximidade, na qual para cada célula foi discretizado o valor da similaridade entre os vértices extremos de cada aresta, esta similaridade conforme já mencionado na fase de *pré-processamento*, foi calculada utilizando a função *cosseno*.

Para a discretização, foi utilizado um parâmetro de corte T sugerido por [10]. Esse parâmetro determina a discretização da matriz de proximidade, se $S_{ij} \geq T$, então os documentos x_i e x_j possuirão uma aresta de conexão, caso contrário, se $S_{ij} \leq T$, então não existirá conexão entre os documentos x_i e x_j.

É importante ressaltar que o parâmetro de corte é dinâmico, podendo ser alterado dependendo a característica da matriz de similaridade. Seu principal objetivo é eliminar os ruídos, isto é, eliminar as pequenas similaridades que possam ocorrer entre documentos com chaves ou algoritmos diferentes, e reforçar a conexão entre documentos de mesma chave ou algoritmo.

Desta foma o pseudocódigo do algoritmo de classificação não-supervisionada possui os seguintes passos:

1) Construir a lista de adjacência para cada vértice.

2) Escolher uma lista e aplicar uma busca em largura para encontrar todos os vértices conexos ao vértice cabeça da lista e marcar estes vértices como já visitados.
3) Para cada vértice encontrado, repetir o passo 2 até que todos os vértices conexos sejam visitados.
4) Se todos os vértices já foram buscados fim, senão voltar ao passo 2 para encontrar uma nova partição.

Dado o algoritmo, pode-se perceber que o número de partições encontradas será igual ao número de criptogramas cifrados pela mesma chave ou algoritmo, e a complexidade do algoritmo é $O(n^2)$ ficando esta complexidade em função da iteração dos vértices e da busca em largura para cada vértice ainda não visitado.

IV. Modelagem do Algoritmo Genético

A. Representação Cromossomial

Inicialmente, o Algoritmo Genético gera aleatoriamente uma matriz binária. Esta matriz é ilustrada na figura 3. Cada linha da matriz representa uma classe e cada coluna um criptograma. Se um criptograma pertence a uma determinada classe, então o elemento da matriz que esta no cruzamento da sua coluna com a linha da classe terá valor igual a um. Caso contrário, o elemento será igual a zero. Como cada criptograma pertence a uma única classe, cada coluna da matriz tem um elemento com valor um, tendo o restante valor zero.

Classes	Criptogramas				
	C_1	C_2	C_3	.....	C_i
Classe 1	1	0	1	.....	0
Classe 2	0	0	0	.....	0
Classe 3	0	0	0	.....	0
Classe 4	0	1	0	.....	0
:	:	:	:	:	:
Classe k	0	0	0	.....	0

Figura 3. Modelo representativo do *cromossomo* do Algoritmo Genético

B. Função de Avaliação

De acordo com [11], as funções de avaliação mais usuais para problemas envolvendo a tarefa de classificação não-supervisionada são:

a) *Minimização do traço* (*W*);

$$W = \sum_{k=1}^{K} \sum_{i=1}^{n_k} \parallel x_i - Z_k \parallel^2 \tag{2}$$

, onde:

- k = Número de classes de um conjunto de dados;
- n_k = Número de dados em uma classe k;
- z_k = Centro geométrico (centroide) de uma classe de dados;
- x_i = *i- ésimo* dado do conjunto.

b) *Maximização do traço* ($\frac{B}{W}$);

$$\frac{B}{W} = \frac{\sum_{k=1}^{K} n_k \parallel Z_k - Z \parallel^2}{\sum_{k=1}^{K} \sum_{i=1}^{n_k} \parallel x_i - Z_k \parallel^2} \tag{3}$$

, onde:

- z = Centro geométrico (centroide) do conjunto de todos os dados;

Nessas funções, *W* representa a medida de dispersão dentro de uma classe de dados e *B* a medida de dispersão entre classes. Ambas as funções, incrementadas no Algoritmo Genético, realizam a classificação não-supervisionada em uma coleção de criptogramas. Entretanto, há uma restrição correlacionada a estas funções no que diz respeito a quantidade de classes geradas. Para a realização de uma classificação não-supervisionada, o valor do número de classes torna-se um parâmetro indispensável de entrada no algoritmo modelado. É compreensível esta limitação pelo fato de [12] nos mostrar que *W* é equivalente a função erro-quadrático utilizada no algoritmo *K-means*. Esta restrição de acordo com [13], obriga o usuário especificar o número de classes que se deve gerar em uma massa de dados. Para erradicar esta restrição, foi inserido no Algoritmo Genético o índice *Calinski−Harabasz* (CH) procedente de [14]. Assim, a classificação não-supervisionada foi executada sem a necessidade de informar o número exato de classes para o algoritmo modelado. [15], [16], utilizaram este índice em outros trabalhos de classificação não-supervisionada automática com o próprio *K-means* em variados conjuntos de dados não correlacionados a criptogramas.

c) Índice *Calinski−Harabasz* (CH)

$$\frac{B}{W} = \frac{\sum_{k=1}^{K} n_k \parallel Z_k - Z \parallel^2 (n-K)}{\sum_{k=1}^{K} \sum_{i=1}^{n_k} \parallel x_i - Z_k \parallel^2 (K-1)} \tag{4}$$

Este índice é a função de maximização de traço (*B*/*W*) multiplicada pelo fator (*n*-K)/(K-1) que é similar ao termo de Hartigan (*n*-K-1) procedente de [17]. Os termos multiplicadores tanto do índice CH quanto de Hartigan, de acordo com [18], são os graus de liberdade aplicados na função estatística. [19] consideram o termo de Hartigan como um fator de penalidade para a correção de um número grande de classes.

C. Parâmetros de entrada

A tabela 1 contém os parâmetros utilizados no Algoritmo Genético com o quais foram encontrados os melhores resultados.

Tabela 1. Parâmetros específicos de entrada do Algoritmo Genético no conjunto de dados analisados

Parâmetro	Valor
Tamanho da população	200
Número de gerações	2000
Taxa de *crossover*	0.95
Taxa de mutação	0.01

V. EXPERIMENTOS

A. Descrição dos ensaios

Para avaliar a eficiência dos algoritmos no que diz respeito a identificação de assinatura em criptogramas cifrados com a mesma chave ou algoritmo, foram realizados quatro experimentos. O modo de operação utilizado foi o ECB(Electronic Codebook). A justificativa para a utilização desse modo está no fato de que os algoritmos devem ter força suficiente para resistir a ataques sob a condição de "pior caso".

O primeiro ensaio utilizou um conjunto de textos aleatórios cifrados pelo AES, MARS, RC6, TWOFISH e SERPENT. O objetivo deste experimento é classificar os arquivos cifrados pelo tipo de algoritmo criptográfico em uso e confirmar que cada algoritmo provê uma assinatura própria. Todos os algoritmos utilizaram a mesma chave com um tamanho de 128 bits e os arquivos de texto em claro foram criptografados pelo software proveniente do [20] que também possui um gerador aleatório de chaves. Foram utilizados 21 criptogramas de 10 Kbytes para cada algoritmo criptográfico totalizando um conjunto de 105 criptogramas.

O segundo experimento teve como objetivo identificar a assinatura de criptogramas que utilizaram o mesmo tipo de algoritmo criptográfico e chaves diferentes. Neste caso, foi utilizado o algoritmo AES. As chaves foram geradas pela classe *KeyGenerator* oriundas da biblioteca *javax.crypto* disponível na linguagem Java. O tamanho dos arquivos também foi de 10 Kbytes. Foram utilizados 30 arquivos para cada uma das cinco chaves totalizando 150 criptogramas.

O terceiro experimento teve como objetivo verificar a eficiência dos algoritmos do primeiro ensaio para criptogramas de 4 Kbytes.

O quarto experimento teve como objetivo verificar a eficiência dos algoritmos do primeiro ensaio para criptogramas de 2 Kbytes.

B. Resultados

Para avaliar os resultados da classificação não-supervisionada, foram utilizada as métricas de revocação (recall) e precisão (precision). A revocação (r) é a razão entre a quantidade de criptogramas corretamente classificados pelo sistema (cs) e a quantidade esperada de criptogramas classificados (c). A precisão (p) é a proporção entre a quantidade de criptogramas corretamente classificados pelo sistema (cs) e o total de criptogramas classificados pelo sistema (na). Deste modo, considerando *n* classes existentes, os valores de revocação e precisão são dados por :

$$r = \frac{1}{n}\sum_{i=1}^{n}(cs_i/c_i) \text{ e } p = \frac{1}{n}\sum_{i=1}^{n}(cs_i/a_i), \text{ onde:}$$

- cs_i = Número de classes de um conjunto de dados;
- c_i = Número real de criptogramas existentes na classe *i*;
- a_i = Número de criptogramas classificados para a classe *i*.

Os ensaios 1, 2 e 3 apresentaram resultados satisfatórios tendo revocação e precisão igual a 1 para as duas técnicas aplicadas, Algoritmo Genético e Grafos.

A figura 4 ilustra o resultado dos classes encontradas no ensaio 1 pelo algoritmo que utiliza a teoria dos Grafos. Para os ensaios 2 e 3 a figura é similar, tendo como única diferença a conexão entre os vértices.

A figura 5 ilustra a curva do índice CH onde o ponto máximo da função de avaliação representa o número de classes corretas para o primeiro ensaio. Para os experimentos 2 e 3 a curva do índice CH apresentou um comportamento semelhante.

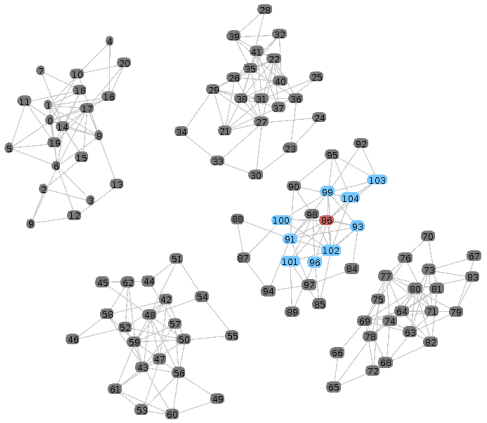

Figura 4. Partição com revocação = 1 e precisão = 1

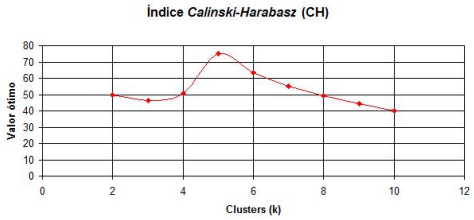

Figura 5. Partição com revocação = 1 e precisão = 1

Para criptogramas de 2 Kbytes os resultados do ensaio 4 tanto para a técnica dos Grafos como para os Algoritmos Genéticos não foram satisfatórios. A métrica de revocação, aplicando o Algoritmo de Grafos e Genéticos, tomou valores de 0.3492 e 0.64, respectivamente. Com relacão a métrica de precisão, os valores foram 1 e 0.717 para Grafos e Genéticos, respectivamente.

Em face do exposto, pode-se perceber que o reconhecimento de padrões em criptogramas que utilizam o modo de operação ECB possuem restrição quanto ao tamanho dos arquivos. Foram realizados ensaios bem sucedidos (valores máximos de revocação e precisão), para as duas técnicas utilizadas, até o limite inferior de arquivos de 4 Kbytes de tamanho.

Comparando os resultados, identifica-se a possibilidade do limite inferior do tamanho dos criptogramas estar próximo de 4 Kbytes; valor testado com o qual as duas técnicas detectaram assinaturas pela chave ou algoritmo utilizado no processo de cifragem. Outra característica que pode ser comparada é o tempo de processamento das duas técnicas. Os ensaios foram

realizados com uma máquina com processador Core 2 Duo 2.6 GHz. A técnica de Grafos mostrou-se bem mais rápida. O tempo de processamento do Algoritmo Genético para cada experimento foi de 3 minutos, enquanto que o Algoritmo de Grafos foi de 2 segundos.

VI. Conclusão

Os testes estatísticos propostos pelo NIST têm por objetivo a detecção de padrões nos criptogramas gerados pelos algoritmos ensaiados e, assim, testar se os mesmos são bons geradores de números pseudo-aleatórios. Entretanto, conforme visto ao longo deste trabalho, no caso dos algoritmos finalistas do processo para a escolha do AES, esses testes não foram suficientes para detectar os padrões que vários autores, posteriormente e com diferentes técnicas, foram capazes de detectar.

Neste artigo, foram propostas duas novas técnicas baseadas em Algoritmos Genéticos e Grafos, as quais foram capazes de separar corretamente os criptogramas gerados pelos algoritmos finalistas do concurso do AES: MARS, RC6, Rijndael, Serpent e Twofish; o que leva a identificação desses algoritmos a partir dos criptogramas gerados pelos mesmos. A identificação relatada demonstra a existência de assinaturas nos criptogramas, as quais são decorrentes das transformações realizadas pelos algoritmos criptográficos ou da mudança de transformação no algoritmo provocada pela chave utilizada na cifração.

As técnicas apresentadas contribuem principalmente na identificação correta de cifras modernas, apresentado melhores resultados na identificação de cifras do que os trabalhos relacionados relatados, e com um custo computacional muito inferior a qualquer outra técnica mencionada.

Os resultados deste trabalho sugerem como trabalhos futuros: a) a identificação e separação de classes de chaves para um mesmo algoritmo; b) estudos para modificar as transformações matemáticas nos algoritmos criptográficos testados neste artigo para que, mesmo em modo ECB, estes não propaguem informações dos textos claros para os criptogramas gerados.

Referência

[1] Soto, J. and Bassham, L, *Randomness Testing of the Advanced Encryption Standard Candidates Algorithms*, NIST Internal Report, 2000.

[2] Murphy, S, *The Power of NIST´s Statistical Testing of AES Candidates*, Information Security Group, Royal Holloway, University of London, 2000.

[3] Knudsen, L.R. and Meier, W, *Correlations in RC6 with a Reduced Number of Rounds*, Proceedings of the 7th International Workshop on Fast Software Encryption, 2000.

[4] Carvalho, C. A. B, *O uso de técnicas de recuperação de informações em criptoanálise*, Instituto Militar de Engenharia, 2006.

[5] Souza, W. A. R, *Identificação de padrões em criptogramas usando técnicas de classificação de textos*, Instituto Militar de Engenharia, 2006.

[6] Souza, W. A. R et al, *Método de Agrupamento de Criptogramas em Função das Chaves de Cifrar*, IV Workshop em Algoritmos e Aplicações de Mineração de Dados (SBBD/SBES), 2008.

[7] Dileep, A. D and Sekhar, C. C, *Identification of block ciphers using support vector machines*, International Joint Conference on Neural Networks, 2006.

[8] Ueda, T. K e Terada, R, *Uma Versão Mais Forte do Algoritmo RC6 contra a criptoanálise*, VII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, 2007.

[9] Nagireddy, *A Pattern Recognition Approach to Block Cipher Identification*, Indian Institute of Technology Madras, 2008.

[10] Ding, H and Zhang, Y, *An Automatic kernel of Graph Clustering Method in Conforming Clustering Number*, International Conference on Computational Intelligence and Security Workshops, 2007.

[11] Passos, E. and Goldschimidt, R., *Data Mining: um guia prático*, Elsevier Editora Ltda., Rio de Janeiro, Brasil, 2005.

[12] Jain, A. K.; Murty, M. N.; Flynn, P. J., *Data Clustering: A Review.*, ACM Computing Surveys, Vol. 31, No. 3, September , 1999.

[13] Halkidi, M, Batistakis, Y, Varzigiannis, M, *Clustering algorithms and validity measures*, Departament of Informatics, Athens University of Economics Business, 2001.

[14] Calinski, R.B., e Harabasz, J., *A Dendrite Method for Cluster Analysis.* Communications in Statistics, 3(1), 1-27, 1974.

[15] Maulik, U. e Bandyopadhyay, S., *Performance Evaluation of Some Clustering Algorithms and Validity Indices.*, IEEE,2002.

[16] Nunes, E. e Conci, A., *Clusterização Automática na Redução da Dimensionalidade dos Dados*, XI Simpósio de Pesquisa Operacional e Logística da Marinha, 5 e 6 de Agosto de 2008 na Escola de Guerra Naval-EGN (Livro de resumos p. 36-37).

[17] Hartigan, J. A., *Clustering Algorithms*, Wiley Series in Probability and Mathematical Statistics, John Wiley and Sons, Inc,1975.

[18] Bassab, W. O., Miazaki, E. S. e Andrade, D. F, *Associação Brasileira de Estatística*, ABE 9^o Simpósio Nacional de Probabilidade e Estatística, São Paulo, 1990.

[19] Li, Xiang., RAMACHANDRAN, Rahul., MOVVA, Sunil., e GRAVES Sara, *Storm Clustering for Data-driven Weather Forecasting*, 24th Conference on International Institute of Professional Studies (IIPS). University of Alabama in Huntsville,2008.

[20] Carvalho, J. L. N; Guedes, L. C. C; Amaral, J. C; Salles, D. V,.*Especificação do algoritmo GCC. Confidencial*, Centro de Análises de Sistemas Navais, DP/3903/11C, pp. 39, maio, 1999.

[21] Valdenay, S.*An Experiment on DES Statistical Cryptanalysis*,ACM Conference on Computer and Communication Security, pages 139-147, 1996.
.