



IDENTIFICAÇÃO DE CONTEXTOS LINGÜÍSTICOS EM LINGUAGENS  
DESCONHECIDAS GERADAS POR CIFRAS DE BLOCOS

William Augusto Rodrigues de Souza

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientador: Luis Alfredo Vidal de Carvalho

Rio de Janeiro  
Outubro de 2011

IDENTIFICAÇÃO DE CONTEXTOS LINGÜÍSTICOS EM LINGUAGENS  
DESCONHECIDAS GERADAS POR CIFRAS DE BLOCOS

William Augusto Rodrigues de Souza

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ  
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA  
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS  
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM  
ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

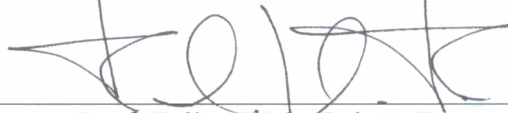
Examinada por:



Prof. Luis Alfredo Vidal de Carvalho, D.Sc



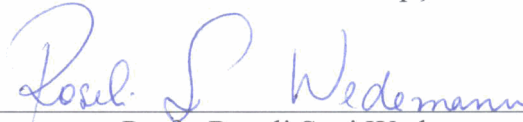
Prof. Antonio Alberto Fernandes de Oliveira, D.Sc



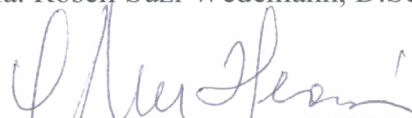
Prof. Felipe Maia Galvão França, Ph.D.



Profa. Lúcia Maria de Assumpção Drummond, D.Sc.



Profa. Roseli Suzi Wedemann, D.Sc.



Prof. José Antônio Moreira Xexéo, D.Sc.



Prof. Geraldo Bonorino Xexéo, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

OUTUBRO DE 2011

Souza, William Augusto Rodrigues de

Identificação de Contextos Lingüísticos em Linguagens Desconhecidas Geradas por Cifras de Blocos / William Augusto Rodrigues de Souza. – Rio de Janeiro: UFRJ/COPPE, 2011.

XIX, 105 p.: il.; 29,7 cm.

Orientador: Luís Alfredo Vidal de Carvalho

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2011.

Referencias Bibliográficas: p. 89 – 94.

1. Criptografia. 2. Identificação de Cifras. 3. Cifras de Blocos. 4. Lingüística Computacional. 5. Criptoanálise. I. Carvalho, Luís Alfredo Vidal de. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

A memória desses seres maravilhosos:

Ao meu Padrinho Tio Hélio Chagas Silva: *Hélio, você  
não é apenas, você é o máximo!*

Ao meu cachorro King.

## **AGRADECIMENTOS**

A toda a minha família, incluindo os meus animais, pelo amor, carinho, compreensão e apoio. Sempre recebo muito energia de vocês!

Ao Professor Luís Alfredo Vidal de Carvalho, pela orientação objetiva e oportuna, pela confiança e por me conceder a oportunidade de realizar o doutorado.

Ao Disnei Vieira Salles, pelas muitas horas de discussão sobre este trabalho o que contribuiu sensivelmente para o desenvolvimento do mesmo.

Ao meu mestre Professor José Antônio Moreira Xexéo, pelas discussões, sugestões e parcerias.

A todos os integrantes da Divisão de Criptologia do Centro de Análises de Sistemas Navais, os quais “sustentaram o fogo” durante o período do meu doutorado, permitindo que todas as missões fossem cumpridas com êxito. Tem sido uma honra comandá-los!

Ao Centro de Análises de Sistemas Navais pelo apoio para a realização do curso.

Aos meus superiores no Centro de Análises de Sistemas Navais em especial aos Almirantes: Gâmbua, Liseo e Pontes Lima; e aos Comandantes: Queiroz, Ungaretti e Benits; pelo apoio e incentivo.

Às Professoras doutoras Lúcia Maria de Assumpção Drummond e Roseli Suzi Wedemann e aos Professores doutores Antonio Alberto Fernandes de Oliveira, Felipe Maia Galvão França, José Antônio Moreira Xexéo e Geraldo Bonorino Xexéo, pela valorosa e prestimosa participação na banca examinadora.

Ao Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia (COPPE) da Universidade Federal do Rio de Janeiro e em especial aos integrantes (professores, alunos e funcionários) do Programa de Engenharia de Sistemas e Computação (PESC) pela acolhida, pela convivência harmoniosa e pelo ambiente adequado ao desenvolvimento de pesquisas.

Ao Grupo de Segurança de Informação do Instituto Militar de Engenharia pelas discussões no campo de Identificação de Cifras de Blocos.

A todos aqueles que contribuíram direta ou indiretamente para o sucesso deste trabalho.

*“Sapientia longe preestat divitiis.”*

Anônimo

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

## IDENTIFICAÇÃO DE CONTEXTOS LINGÜÍSTICOS EM LINGUAGENS DESCONHECIDAS GERADAS POR CIFRAS DE BLOCOS

William Augusto Rodrigues de Souza

Outubro/2011

Orientador: Luís Alfredo Vidal de Carvalho

Programa: Engenharia de Sistemas e Computação

Este trabalho define e demonstra uma metodologia para identificar  $n$  – *cifras* de blocos. O trabalho demonstra também que tal metodologia pode ser utilizada para classificar e identificar criptogramas a partir do modo de operação e da chave criptográfica. A metodologia é baseada em técnicas de Recuperação de Informações e demonstra que o sucesso na identificação se deve à correta separação dos criptogramas em grupos, a qual é possível devido à existência de propriedades intrínsecas nos modelos matemáticos dessas cifras, que criam contexto lingüístico (ou assinatura) nos criptogramas. Tal contexto lingüístico pode ser evidenciado a partir de um identificador de contexto. Logo, os criptogramas podem ser tratados como textos claros escritos em uma linguagem desconhecida, onde cada linguagem é determinada por uma combinação qualquer de parâmetros criptográficos. Assim, qualquer técnica capaz de classificar textos claros, levando em conta a distribuição estatística dos elementos do conjunto léxico (ou conjuntos léxicos) usado por tais textos, pode também classificar criptogramas.

São formulados resultados teóricos de que as cifras de blocos geram linguagens desconhecidas e que se pode identificar estas cifras (ou uma combinação de parâmetros criptográficos) a partir de identificadores de contexto lingüístico contidos nestas linguagens. Os resultados teóricos apresentados são válidos para qualquer linguagem desconhecida. Tal metodologia é considerada, por analogia, como um “ataque de distinção”.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

IDENTIFICATION OF LINGUISTIC CONTEXTS IN UNKNOWN LANGUAGES  
GENERATED BY BLOCK CIPHERS

William Augusto Rodrigues de Souza

October/2011

Advisor: Luís Alfredo Vidal de Carvalho

Department: Computing and System Engineering

This thesis defines and demonstrates a methodology to identify N blocks ciphers. The work also demonstrates that this methodology can be used to classify and identify cryptograms from the operation mode and the encryption key. The methodology is based on techniques of Information Retrieval and demonstrates that success in identifying is due to the correct clustering of cryptograms in groups, which is possible due to the existence of intrinsic properties in mathematical models of these ciphers that create linguistic context (or signature) in cryptograms. Such linguistic context may be evidenced from a context identifier. Therefore, the cryptogram can be treated as plain texts written in an unknown language, where each language is determined by any combination of cryptographic parameters. Thus, any technique able to categorize plain texts, taking into account the statistical distribution of elements in the lexical set (lexical sets) used by such texts, can also categorize cryptograms.

Theoretical results are formulated that block ciphers generate unknown languages and which can be identified these ciphers (or a combination of cryptographic parameters) from the linguistic context identifiers contained in these languages. The theoretical results presented are valid for any unknown language. This methodology is considered, by analogy, as a "distinguishing attack."



## SUMÁRIO

LISTA DE FIGURAS .....	xiv	
LISTA DE TABELAS.....	xv	
LISTA DE DEFINIÇÕES E EXEMPLOS .....	xvii	
LISTA DE PROPOSIÇÕES, COROLÁRIOS E PROVAS .....	xviii	
LISTA DE FÓRMULAS .....	xix	
<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>1</b>
<b>1.1</b>	<b>Criptografia: Visão Geral.....</b>	<b>1</b>
<b>1.2</b>	<b>Criptanálise: Breve Histórico .....</b>	<b>3</b>
<b>1.3</b>	<b>Contexto da Tese .....</b>	<b>6</b>
<b>1.4</b>	<b>Cenários para Identificação de Cifras de Blocos .....</b>	<b>7</b>
1.4.1	Combinanção de Parâmetros Criptográficos.....	8
1.4.2	Identificação pelo Número de Rodadas .....	8
1.4.3	Identificação de Composição de Cifra de Bloco Fixa.....	8
1.4.4	Identificação de Composição de Cifra de Bloco Distinta .....	9
<b>1.5</b>	<b>Objetivos da Tese .....</b>	<b>9</b>
<b>1.6</b>	<b>Motivação .....</b>	<b>10</b>
<b>1.7</b>	<b>Caracterização do Problema .....</b>	<b>11</b>
<b>1.8</b>	<b>Organização da Tese .....</b>	<b>12</b>
<b>2</b>	<b>FUNDAMENTOS TEÓRICOS PARA IDENTIFICAÇÃO DE CONTEXTOS LINGÜÍSTICOS E DE <math>N - CIFRAS</math> .....</b>	<b>14</b>
<b>2.1</b>	<b>Sistema Criptográfico .....</b>	<b>14</b>
<b>2.2</b>	<b>Substituição, Permutação, Confusão e Difusão.....</b>	<b>14</b>
<b>2.3</b>	<b>Rede de Substituições e Permutações.....</b>	<b>15</b>
<b>2.4</b>	<b>Cifras de Blocos.....</b>	<b>15</b>
<b>2.5</b>	<b>Composição de Cifras de Blocos.....</b>	<b>16</b>
<b>2.6</b>	<b>Modos de Operação: ECB e CBC.....</b>	<b>16</b>
<b>2.7</b>	<b>Rede de Feistel Tradicional: <math>i - \acute{e}sima</math> Rodada.....</b>	<b>17</b>

<b>2.8</b>	<b>Lingüística Computacional e Recuperação de Informações.....</b>	<b>19</b>
<b>2.9</b>	<b>Estrutura Criptográfica dos Finalistas ao AES .....</b>	<b>22</b>
2.9.1	MARS .....	23
2.9.2	RC6 .....	24
2.9.3	Rijndael .....	27
2.9.4	Serpent.....	29
2.9.5	Twofish.....	31
2.9.6	Resumo das Transformações Criptográficas dos Finalistas ao AES .....	32
<b>3</b>	<b>RESULTADOS SOBRE IDENTIFICAÇÃO DE CIFRAS DE BLOCOS.....</b>	<b>34</b>
<b>3.1</b>	<b>Método Baseado no Texto do <math>\chi^2</math> .....</b>	<b>34</b>
<b>3.2</b>	<b>Métodos com Testes Estatísticos e as Funções XOR e Limiar .....</b>	<b>35</b>
3.2.1	Testes Estatísticos e a Função e XOR .....	35
3.2.2	Função Limiar.....	36
<b>3.3</b>	<b>Máquinas de Vetor de Suporte .....</b>	<b>38</b>
<b>3.4</b>	<b>Recuperação de Informação e Lingüística Computacional.....</b>	<b>40</b>
<b>3.5</b>	<b>Métodos Baseado em Histograma e em Predição de Blocos.....</b>	<b>41</b>
<b>4</b>	<b>METODOLOGIA PARA IDENTIFICAR CIFRAS DE BLOCOS.....</b>	<b>42</b>
<b>4.1</b>	<b>Recuperação de Informação na Identificação de Cifras.....</b>	<b>42</b>
<b>4.2</b>	<b>Unidade Básica para o Tratamento de Criptogramas .....</b>	<b>42</b>
<b>4.3</b>	<b>Modelagem de Criptogramas sobre um Espaço Vetorial .....</b>	<b>43</b>
<b>4.4</b>	<b>Medida de Similaridade entre Criptogramas .....</b>	<b>45</b>
<b>4.5</b>	<b>Agrupamento de Criptogramas.....</b>	<b>45</b>
<b>4.6</b>	<b>Avaliação do Agrupamento de Criptogramas .....</b>	<b>46</b>
<b>4.7</b>	<b>Descrição da Metodologia para 5 – cifras de Blocos .....</b>	<b>47</b>
<b>5</b>	<b>EXPERIMENTOS, RESULTADOS E AVALIAÇÕES DE 5 – CIFRAS .....</b>	<b>50</b>
<b>5.1</b>	<b>Introdução aos Experimentos Computacionais .....</b>	<b>50</b>
<b>5.2</b>	<b>Bases de Criptogramas .....</b>	<b>50</b>
<b>5.3</b>	<b>Descrição das Bases de Criptogramas.....</b>	<b>51</b>
5.3.1	Base 1 de Criptogramas.....	51
5.3.2	Base 2 de Criptogramas.....	51

5.3.3	Base 3 de Criptogramas.....	51
5.3.4	Base 4 de Criptogramas.....	52
5.3.5	Base 5 de Criptogramas.....	52
5.3.6	Base 6 de Criptogramas.....	53
5.3.7	Base 7 de Criptogramas.....	53
5.3.8	Base 8 de Criptogramas.....	54
5.3.9	Base 9 de Criptogramas.....	54
5.3.10	Base 10 de Criptogramas.....	55
5.3.11	Base 11 de Criptogramas.....	56
5.3.12	Base 12 de Criptogramas.....	56
5.3.13	Base 13 de Criptogramas.....	56
5.3.14	Base 14 de Criptogramas.....	57
5.4	Experimento 1: Identificação de 5 – <i>cifras</i> .....	57
5.4.1	Separação dos Grupos.....	57
5.4.2	Classificação das Cifras .....	58
5.5	Experimento 2: Identificação do Modo de Operação de 5 – <i>cifras</i> .....	59
5.5.1	Separação dos Grupos.....	59
5.5.2	Classificação dos Modos de Operação: ECB e CBC.....	60
5.6	Experimento 3: Agrupamento em Função de Parâmetros.....	62
5.6.1	Fixando o Modo de Operação e Variando a Cifra e a Chave.....	62
5.6.2	Variando a Cifra, o Modo de Operação e a Chave .....	63
5.6.3	Separação Usando um Número Inferior de Rodadas.....	64
5.7	Experimento 4: Composição de 5 – <i>cifras</i> Iguais no Modo ECB.....	64
5.7.1	Separação em Grupos Fixando uma Chave.....	65
5.7.2	Separação em Grupos Usando Chaves Diferentes.....	65
5.8	Experimento 5: Composição de 5 – <i>cifras</i> Diferentes no Modo ECB.....	66
5.8.1	Separação dos Grupos.....	66
5.9	Experimento 6: Composição de 5 – <i>cifras</i> nos Modos ECB e CBC.....	67
5.10	Experimento 7: Separação de Mensagens e Criptogramas .....	68
5.10.1	Separação de Mensagens.....	69
5.10.2	Separação dos Criptogramas .....	70
5.10.3	Separação das Mensagens e Criptogramas.....	71

5.10.4	Separação por Alfabeto.....	72
5.11	ANÁLISE ESTATÍSTICA DAS BASES DE CRIPTOGRAMAS.....	72
<b>6</b>	<b>RESULTADOS TEÓRICOS.....</b>	<b>74</b>
<b>6.1</b>	<b>Generalização da Metodologia para <math>n</math>-cifras de Bloco.....</b>	<b>74</b>
<b>6.2</b>	<b>Qualquer Método, Baseado no Conjunto Léxico de uma Linguagem, Usado para Agrupar e Classificar Textos Claros Pode Agrupar e Classificar Criptogramas.....</b>	<b>76</b>
<b>6.3</b>	<b>Cada Cifra Gera um Identificador de Contexto.....</b>	<b>78</b>
<b>6.4</b>	<b>Cada Chave Gera um Identificador de Contexto.....</b>	<b>80</b>
<b>6.5</b>	<b>A Variação de Parâmetros Criptográficos nas Cifras de Blocos Cria um Número Variado Linguagens Desconhecidas.....</b>	<b>81</b>
<b>7</b>	<b>DISCUSSÃO.....</b>	<b>83</b>
<b>7.1</b>	<b>Trabalhos Futuros.....</b>	<b>85</b>
<b>8</b>	<b>CONCLUSÕES.....</b>	<b>87</b>
<b>9</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>89</b>
<b>10</b>	<b>APÊNDICES.....</b>	<b>95</b>
<b>10.1</b>	<b>Apêndice 1: Formatos das Chaves Criptográficas.....</b>	<b>96</b>
10.1.1	Conjunto de Chaves 1: Formato Hexadecimal.....	96
10.1.2	Conjunto de Chaves 1: Formato Binário.....	96
10.1.3	Conjunto de Chaves 2: Formato Hexadecimal.....	97
10.1.4	Conjunto de Chaves 2: Formato Binário.....	98
<b>10.2</b>	<b>Apêndice 2: Agrupamento de Criptogramas por Relações Fuzzy Equivalentes.....</b>	<b>101</b>
10.2.1	Introdução.....	101
10.2.2	Descrição do Procedimento.....	102
10.2.3	Cálculo da Pertinência.....	102
10.2.4	Teste das Propriedades da Matriz de Relações.....	103
10.2.5	Defuzzificação.....	103

10.2.6	Agrupamento .....	104
10.2.7	Experimento, Resultado e Avaliação.....	104
10.2.8	Conclusão .....	105

## LISTA DE FIGURAS

Figura 1.1:	Modelo para Segurança das Comunicações (adaptada de STALLINGS, 2010).....	2
Figura 2.1:	Cifração na Estrutura de Feistel.....	18
Figura 2.2:	Decifração na Estrutura Feistel.....	18
Figura 2.3:	Estrutura Criptográfica do MARS (Adaptada de BURWICK, 1999).....	23
Figura 2.4:	Estrutura de Cifração do RC6 (Adaptada de RIVEST, 1998).....	26
Figura 2.5:	Estrutura de Cifração do Rijndael (LAMBERT, 2004).....	28
Figura 2.6:	Estrutura da Cifração do Serpent .....	30
Figura 2.7:	Estrutura de Cifração do Twofish (adapatada de SCHNEIER, 1998) .....	32
Figura 4.1:	Modelo de Espaço de Vetores de Criptogramas .....	44
Figura 4.2:	Dendrograma.....	46
Figura 4.3:	Revocação e Precisão .....	46
Figura 4.4:	Esquema do mecanismo com o caso particular de $n = 5$ cifras .....	48
Figura 4.5:	Modelo para Identificação de Cifra.....	48
Figura 10.2.1:	Esquema do agrupamento por relações fuzzy equivalentes .....	101

## LISTA DE TABELAS

Tabela 2.1:	Terminologia alternativa para as operações do Rijndael.....	27
Tabela 2.2:	Operações dos Algoritmos Finalistas do AES .....	33
Tabela 5.1:	Configuração da base 1 de criptogramas.....	51
Tabela 5.2:	Configuração da base 2 de criptogramas.....	51
Tabela 5.3:	Configuração da base 3 de criptogramas.....	52
Tabela 5.4:	Configuração da base 4 de criptogramas.....	52
Tabela 5.5:	Configuração da base 5 de criptogramas.....	52
Tabela 5.6:	Configuração da base 6 de criptogramas.....	53
Tabela 5.7:	Configuração da base 7 de criptogramas.....	54
Tabela 5.8:	Configuração da base 8 de criptogramas.....	54
Tabela 5.9:	Configuração da base 9 de criptogramas.....	55
Tabela 5.10:	Configuração da base 10 de criptogramas.....	56
Tabela 5.11:	Configuração da base 11 de criptogramas.....	56
Tabela 5.12:	Configuração da base 12 de criptogramas.....	56
Tabela 5.13:	Resultado da separação em grupos.....	58
Tabela 5.14:	Resultado da identificação das cifras de blocos.....	58
Tabela 5.15:	Resultado da separação em grupos nos modos: ECB e CBC .....	60
Tabela 5.16:	Resultado da Separação em Grupos no modo ECB .....	60
Tabela 5.17:	Identificação dos modos de operação com grupos ECB e CBC.....	61
Tabela 5.18:	Identificação para grupos no modo ECB.....	61
Tabela 5.19:	Resultado da separação em grupos.....	62
Tabela 5.20:	Resultado da separação em grupos com diferentes combinações.....	63
Tabela 5.21:	Resultado da Separação em Grupos por Rodadas.....	64
Tabela 5.22:	Separação em grupos com cinco composições e uma chave.....	65
Tabela 5.23:	Separação em grupos com cinco composições e cinco chaves.....	66
Tabela 5.24:	Separação em grupos usando composição de 5 – <i>cifras</i> de bloco.....	67
Tabela 5.25:	Quatro composições no modo CBC e a última no modo ECB .....	68
Tabela 5.26:	Quatro composições no modo ECB e a última no modo CBC.....	68
Tabela 5.27:	Separação em grupos somente das mensagens .....	69
Tabela 5.28:	Separação em grupos somente dos criptogramas.....	70
Tabela 5.29:	Separação em grupos das mensagens e criptogramas .....	71

Tabela 5.30: Separação em grupos por alfabeto .....	72
Tabela 5.31: Contagem de blocos e repetição de blocos nos textos claros .....	73
Tabela 10.2.1: Resultado da separação em grupos.....	104



## LISTA DE DEFINIÇÕES E EXEMPLOS

Definição 2.1: Sistema Criptográfico.....	14
Definição 2.2: Cifra de Bloco .....	16
Definição 2.3: Composição de Cifras de Blocos .....	16
Definição 2.4: Modo de Operação ECB.....	17
Definição 2.5: Modo de Operação CBC.....	17
Definição 2.6: Cifração na Estrutura de Feistel.....	18
Definição 2.7: Decifração na Estrutura de Feistel .....	19
Definição 2.8: Alfabeto .....	19
Exemplo 2.1: A língua portuguesa .....	19
Exemplo 2.2: O alfabeto ASCII .....	19
Definição 2.9: Palavra .....	19
Exemplo 2.3: Uma palavra $n$ o código ASCII.....	19
Definição 2.10: Comprimento de uma Palavra.....	19
Definição 2.11: Concatenação de duas Palavras.....	19
Definição 2.12: Conjunto de todas as Palavras sobre um Alfabeto .....	19
Definição 2.13: Conjunto de Palavras de Comprimento $n$ sobre um Alfabeto .....	20
Exemplo 2.4: Um conjunto de palavras de comprimento $n$ .....	20
Definição 2.14: Linguagem sobre um Alfabeto.....	20
Exemplo 2.5: A determinação de uma linguagem sobre um alfabeto .....	20
Definição 2.15: Similaridade entre dois vetores .....	20
Definição 2.16: Procedimento de agrupamento.....	21
Exemplo 2.6: Um procedimento de agrupamento .....	21
Definição 2.17: Identificador de Contexto .....	22
Exemplo 2.7: Um Identificador de Contexto em Economia .....	22
Definição 3.1: Função Limiar .....	36

## LISTA DE PROPOSIÇÕES, COROLÁRIOS E PROVAS

Proposição 6.1: Generalização para $n - cifras$ no modo ECB.....	74
Prova 6.1.....	74
Proposição 6.2: Generalização para $n - cifras$ no modo CBC .....	75
Prova 6.2.....	75
Proposição 6.3: Métodos para Classificar Textos Claros, Classificam Criptogramas.....	76
Prova 6.3.....	76
Proposição 6.4: Cada Cifra Gera um Identificador de Contexto .....	78
Prova 6.4.....	78
Proposição 6.5: Cada Chave Gera um Identificador de Contexto .....	80
Prova 6.5.....	80
Proposição 6.6: Criação de um Número Variado Linguagens Desconhecidas.....	81
Prova 6.6.....	82

Corolário 6.1: Equivalência de métodos de classificação com $IV$ fixo no modo CBC .....	77
---	----

## LISTA DE FÓRMULAS

Fórmula 4.1:  $s_{co-seno}(\bar{c}_i, \bar{c}_j) = \frac{\sum_{k=1}^n (c_{i,k} \times c_{j,k})}{\sqrt{\sum_{k=1}^n (c_{i,k})^2 \times \sum_{k=1}^n (c_{j,k})^2}} \dots\dots\dots 45$

Fórmula 4.2:  $R = \frac{n}{|k|} \dots\dots\dots 47$

Fórmula 4.3:  $P = \frac{n}{|g|} \dots\dots\dots 47$

Fórmula 4.4:  $R = \frac{1}{m} \sum_{i=1}^m \frac{n_i}{|k_i|} \dots\dots\dots 47$

Fórmula 4.5:  $P = \frac{1}{m} \sum_{i=1}^m \frac{n_i}{|g_i|} \dots\dots\dots 47$

# 1 INTRODUÇÃO

Este capítulo será dividido nas seguintes seções: na seção 1.1 será apresentada, de forma sucinta, uma visão geral sobre Criptografia; na seção 1.2 um breve histórico das principais técnicas de Criptoanálise utilizada na Criptografia Contemporânea; a seção 1.3 apresenta o contexto em que se desenvolve a presente tese; na seção 1.4 todos os cenários (hipóteses de trabalho) que serão empregados para a metodologia para identificação de cifras de blocos; na seção 1.5 os objetivos da tese; na seção 1.6 a motivação que resultou no presente trabalho; na seção 1.7 a caracterização do problema a ser resolvido e na seção 1.8 a organização de toda a Tese.

## 1.1 CRIPTOGRAFIA: VISÃO GERAL

Desde o início da sua história a mais de 4000 anos (KAHN, 1967), o desenvolvimento da criptografia vem sendo estimulado pelo risco de interceptação, por pessoas não autorizadas, de mensagens produzidas por governos e chefes militares. Costuma-se contextualizar a criptografia em quatro fases distintas: manual (ou de lápis e papel), mecânica, eletromecânica e computacional; considerando-se criptografia clássica até a fase da criptografia eletromecânica. As cifras produzidas durante as fases da criptografia clássica são chamadas de cifras clássicas.

A partir da década de 1970, o NBS<sup>1</sup> estabeleceu o primeiro algoritmo criptográfico padrão com base em cifras<sup>2</sup> simétricas de blocos, o Data Encryption Standard – DES (NIST, 1999). Paralelamente, desenvolveu-se a criptografia assimétrica ou de chave pública, da qual o algoritmo RSA<sup>3</sup> (RIVEST *et al*, 1978) (RIVEST *et al*, 1979) (RIVEST *et al*, 1983) é o representante mais relevante.

O objetivo principal da criptografia é manter o sigilo de mensagens que trafegam em um canal de comunicações inseguro, ou seja, sujeito a interceptação por usuários não autorizados. Assim, em um canal sujeito a interceptação, usar criptografia é fundamental para a preservação do sigilo, de tal maneira que somente as partes que possuem a chave

---

<sup>1</sup> National Bureau of Standards (atual NIST).

<sup>2</sup> Nesta tese, os termos algoritmo, algoritmo criptográfico e cifra, são usados indistintamente.

<sup>3</sup> Batizado em homenagem aos seus autores: Rivest, Shamir e Adleman.

criptográfica (utilizada para cifrar e/ou decifrar as mensagens) tenham acesso ao conteúdo de uma mensagem criptografada.

Pode-se classificar o tipo de criptografia em simétrica e assimétrica. A criptografia é simétrica ou de chave secreta se a chave usada para cifrar é igual à chave usada para decifrar ou se uma chave é facilmente deduzida a partir da outra. Diferentemente, a criptografia é assimétrica ou de chave pública quando a chave usada para cifrar é diferente da chave usada para decifrar e não se pode deduzir, em tempo computacional viável, uma chave a partir da outra.

A figura 1.1 ilustra um modelo de segurança baseado em criptografia simétrica, o qual é muito usado na proteção de mensagens sigilosas que trafegam em algum canal de comunicação, como, por exemplo, a Internet.

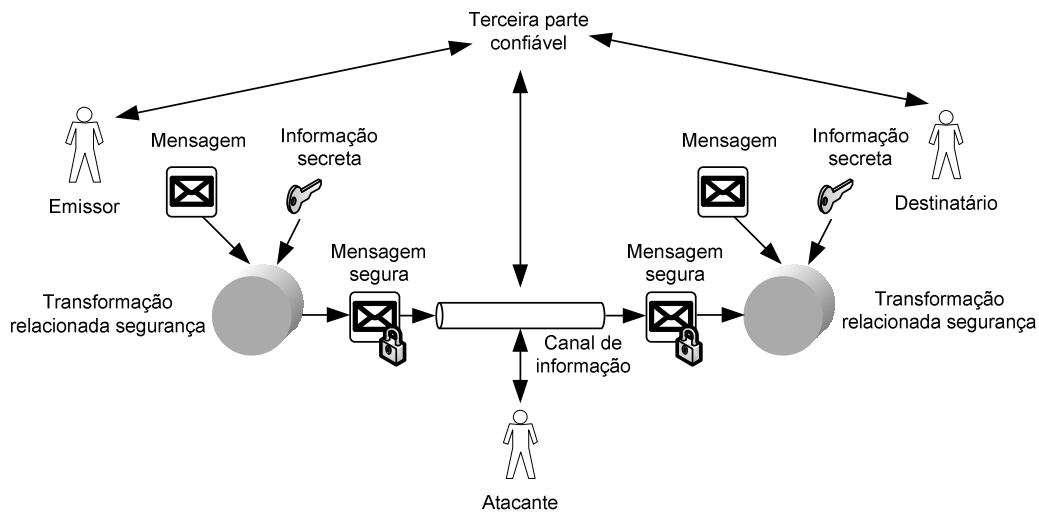


Figura 1.1: Modelo para Segurança das Comunicações (adaptada de STALLINGS, 2010)

As cifras podem, ainda, serem classificadas em: cifras de blocos e cifras de fluxo (ou de cadeia). As cifras de blocos, como o DES citado anteriormente, transformam mensagens em criptogramas operando sobre blocos de *bits* ou *bytes*. O tamanho dos blocos depende do algoritmo e, muitas vezes, do tamanho da chave utilizada. Já as cifras de fluxo transformam mensagens em criptogramas *bit a bit* ou *byte a byte*. As cifras de fluxo não serão objeto de estudo nesta tese.

Com relação às cifras de bloco, o modo de operação é outro parâmetro que influencia o seu funcionamento. Modo de operação é uma técnica para aprimorar o efeito do algoritmo

criptográfico ou adaptá-lo para uma aplicação (STALLINGS, 2010). O NIST<sup>4</sup> definiu os modos de operação a seguir, para possibilitar o emprego de cifras de blocos em diferentes tipos de aplicações: *Electronic Codebook* (ECB), *Cipher Block Chaining* (CBC), *Cipher Feedback* (CFB), *Output Feedback* (OFB), *Counter* (CTR) (NIST 1980) e (NIST 2001), e *Counter com Cipher Block Chaining* (CCM) (STINSON, 2006). A metodologia para se identificar cifras de bloco apresentada neste trabalho, utilizará, além de outros parâmetros criptográficos, os modos de operação ECB e CBC.

Pode-se dizer que os principais parâmetros criptográficos são as configurações iniciais do processo de cifração. Assim, são parâmetros criptográficos: o algoritmo criptográfico, o tamanho da chave criptográfica, a própria chave criptográfica, o modo de operação utilizado e a mensagem que se deseja cifrar.

Nesta tese, consideram-se os seguintes parâmetros criptográficos: o algoritmo criptográfico, a chave criptográfica e o modo de operação utilizado.

## 1.2 CRIPTOANÁLISE: BREVE HISTÓRICO

O outro ramo da criptologia<sup>5</sup>, a criptoanálise, busca obter o conhecimento do texto legível ou da chave utilizada para criptografar, sem o conhecimento desta chave. Busca também obter os parâmetros criptográficos. A criptoanálise clássica era fortemente baseada nas características lingüísticas do idioma de origem do texto legível (SINGH, 2003). Desta maneira, a criptoanálise explorava as propriedades intrínsecas do idioma refletidas nos criptogramas. O interesse na lingüística permanece nos dias de hoje, pois embora as técnicas atuais de criptografia envolvam problemas matemáticos difíceis, a sua matéria-prima continua sendo o texto<sup>6</sup> (SOUZA, 2007).

A atividade de criptoanálise é também conhecida como ataque, o qual é geralmente classificado de acordo com a quantidade de informações disponíveis (STINSON, 2006) e (STALLINGS, 2010) e com os objetivos e necessidades do criptoanalista (atacante) (MENEZES, 1996). Os ataques são listados abaixo, em ordem decrescente de dificuldade:

**a. ataque somente com texto cifrado:** o criptoanalista possui apenas o texto cifrado.

---

<sup>4</sup> National Institute of Standard and Technology.

<sup>5</sup> Área que se dedica ao estudo da criptografia e da criptoanálise.

<sup>6</sup> Outros tipos de dados podem ser cifrados, como: sons, imagens e vídeos.

**b. ataque com texto claro conhecido:** o criptoanalista tem a quantidade que desejar de textos em claro e seus respectivos criptogramas.

**c. ataque com texto claro escolhido:** o criptoanalista tem acesso ao algoritmo criptográfico e assim escolhe os textos a serem cifrados.

**d. ataque com texto claro escolhido adaptado:** o criptoanalista pode modificar os textos em claro escolhidos de acordo com os resultados de testes anteriores.

Dois dos mais bem sucedidos ataques às cifras de blocos são a criptoanálise diferencial (BIHAM e SHAMIR, 1991) e a criptoanálise linear (MATSUI, 1993). A criptoanálise diferencial é um ataque com texto em claro escolhido que compara a diferença, por meio da operação *XOR* (ou exclusivo), de um par de textos em claro (entrada) com a diferença dos criptogramas correspondentes (saída), requerendo  $2^{47}$  (aproximadamente 140 trilhões) pares de textos claros escolhidos para ser bem sucedido. A criptoanálise linear é um ataque com textos em claro conhecidos que usa aproximações lineares para descrever as transformações realizadas no DES, requerendo  $2^{43}$  (aproximadamente nove trilhões) pares textos claro conhecidos para ser bem sucedido. Quebrar<sup>7</sup> o DES com complexidade  $2^{47}$  ou  $2^{43}$  é um esforço muito menor do que o requerido pela força bruta que é de  $2^{56}$ . Porém, a necessidade de  $2^{47}$  ou  $2^{43}$  pares de textos exige um elevado custo computacional (MENEZES, 1996), (STINSON, 2006) e (STALLINGS, 2010).

Em tese, todas as cifras de blocos são suscetíveis às criptoanálises linear e diferencial. Entretanto, uma cifra só tem a sua segurança comprometida se a complexidade para executar tais ataques for menor do que a requerida pela força bruta no espaço de chaves.

Assim, após mais de 20 anos de uso, ataques de criptoanálise linear e diferencial, e a quebra por força bruta do DES (EFF, 1998), o NIST lançou um concurso para escolher o novo padrão criptográfico, também baseado em cifras simétricas de blocos, denominado Advanced Encryption Standard – AES. Cinco algoritmos foram finalistas do concurso: MARS, RC6, Rijndael, Serpent e Twofish. O algoritmo Rijndael foi o vencedor, em 2001 (NIST, 2001).

A verificação da robustez desses e de outros algoritmos criptográficos exige o teste de requisitos de segurança. Estes requisitos se baseiam na resistência contra diferentes tipos de ataques, alguns disponíveis na literatura científica. Dentre eles, estão os já citados ataques de

---

<sup>7</sup> Descobrir uma fraqueza no algoritmo que permita, a um atacante, desfazer a transformação criptográfica produzida por este algoritmo.

Criptanálise Linear e Diferencial. Outros ataques podem ser vistos em (MENEZES, 1996) e (CANNIÈRE, 2006). Apesar da maioria dos ataques não serem práticos em função dos atuais recursos computacionais, eles têm sido úteis para definir a robustez criptográfica de criptosistemas simétricos, como, por exemplo, as cifras de blocos contemporâneas.

Na essência, todos esses ataques, mesmo não sendo práticos, visam obter propriedades estatísticas, usando pares de texto claro e texto cifrado (criptograma), independente da chave criptográfica, que apoiem a verificação da robustez criptográfica das cifras de blocos. Assim, “quebrar” uma cifra não significa, necessariamente, obter o texto claro a partir de um criptograma sem o conhecimento da chave. “Quebrar” pode significar explorar uma fraqueza na cifra que permita um esforço menor que a força bruta para realizar a criptanálise de uma cifra. Em suma, quebrar uma cifra pode significar encontrar uma evidência de que a cifra não funciona como anunciado (SCHNEIER, 2000).

A robustez criptográfica está relacionada ao fato de que o algoritmo criptográfico precisa gerar textos cifrados (criptogramas), a partir de textos claros, com seqüências aleatórias ou pseudo-aleatórias de *bits*. No processo de escolha do AES, os ataques apresentados contra os algoritmos candidatos tinham apenas relevância acadêmica, possuindo complexidade um pouco menor do que a busca exaustiva, os quais em sua maioria utilizavam versões da cifra com menor número de rodadas do que o proposto (DAEMEN e RIJMEN, 2002).

No intuito de verificar se um determinado algoritmo criptográfico pode ser utilizado como gerador de números pseudo-aleatórios, o NIST propõe uma bateria de testes estatísticos (NIST, 2008). A aprovação nestes testes estatísticos é uma condição necessária, mas não suficiente, para se caracterizar a robustez criptográfica desses algoritmos. MURPHY (2000), por exemplo, contestou sobre a adequabilidade desses testes, sugerindo que os mesmo precisariam de testes complementares.

Visando, ainda, a busca de fraquezas em um algoritmo criptográfico pelo emprego de técnicas estatísticas, pode ser citado o “Ataque de Distinção” (KNUDSEN e MEIER, 2000), que aplica o teste  $\chi^2$  em um conjunto de criptogramas gerados pelo RC6. A conclusão desse trabalho indica que o RC6 é vulnerável a tal ataque com 15 iterações e complexidade da ordem de  $2^{125}$ . Com relação ao RC6, uma contramedida a esse tipo de ataque foi proposta por UEDA e TERADA (2007) com intuito de fortalecer este algoritmo.

Métodos de Inteligência Computacional também podem ser utilizados para verificação inicial da segurança de criptosistemas, buscando revelar padrões em seus criptogramas (LASKARI, 2007). Os trabalhos de ALBASSAL e WAHDAN (2004), DELEEP *et al* (2008)



e RAO *et al* (2009), por exemplo, apresentam o uso de métodos de Inteligência Computacional na criptoanálise de cifras de blocos baseadas na estrutura de Feistel. A Estrutura de Feistel será apresentada nos fundamentos teóricos.

Técnicas de recuperação de informações foram usadas com sucesso para agrupar criptogramas, no modo ECB, em função da chave, aplicadas aos algoritmos DES, AES e RSA (CARVALHO, 2006), (SOUZA, 2007) e (SOUZA *et al*, 2008). Outras técnicas foram usadas para identificar cifras, no modo ECB, com máquinas de vetor de suporte aplicadas aos algoritmos DES, Triple DES, Blowfish, AES e RC5 (DILLEP e SEKHAR, 2006); com testes de aleatoriedade, operações XOR e funções limiar, aplicadas aos algoritmos DES, IDEA, Blowfish, RC4, Camellia e RSA (MAHESHWARI, 2001), (CHANDRA, 2002), (RAO, 2003) e (SAXENA, 2008); e com métodos de histograma e de predição de bloco aplicados aos algoritmos DES, Triple DES, Blowfish, RC5 e AES (NAGIREDDY, 2008). Cabe citar o trabalho de KANT *et al* (2005), o qual propõe um método para identificação de cifras de fluxo baseado em fusão por regra de votação por maioria (Majority Voting).

Esses trabalhos citados, entretanto, estão limitados a identificar e distinguir cifras com mecanismos especializados em um único algoritmo, o qual se pretende identificar. Assim, nesses trabalhos, dado um conjunto de criptogramas, o identificador pode separar do conjunto de criptogramas apenas os criptogramas gerados pelo algoritmo no qual é especializado. Além disso, há poucos relatos utilizando o modo CBC.

### 1.3 CONTEXTO DA TESE

Este trabalho contorna as limitações acima, ao definir e demonstrar uma metodologia para identificar  $n$  – cifras de blocos. A metodologia é aplicada a um caso particular de  $n = 5$  onde são utilizadas as cifras de blocos finalistas do concurso para a escolha do AES: MARS, RC6, Rijndael, Serpent e Twofish; nos modos de operação ECB e CBC. Após a aplicação ao caso particular, são apresentadas as proposições e as respectivas provas de que a metodologia vale para um  $n$  qualquer.

O trabalho também demonstra que tal metodologia pode ser utilizada para classificar e identificar criptogramas a partir de outros parâmetros criptográficos além das cifras, como: modo de operação e chave.

A metodologia é baseada em técnicas de Recuperação de Informações e demonstra que o sucesso na identificação se deve à correta separação dos criptogramas em grupos, a qual é

possível devido à existência de propriedades intrínsecas nos modelos matemáticos dessas cifras, as quais criam contexto lingüístico (ou assinatura) nos criptogramas. Tais contextos lingüísticos poder ser evidenciados a partir de um identificador de contexto. As definições de identificador de contexto e contexto lingüístico serão formuladas nos fundamentos teóricos.

O trabalho demonstra, ainda, que os criptogramas podem ser tratados como textos claros escritos em uma linguagem<sup>8</sup> desconhecida e utilizando um alfabeto binário, onde cada linguagem é determinada por uma combinação qualquer de parâmetros criptográficos. Desta forma, cada combinação de parâmetros criptográficos determina um conjunto léxico para esta linguagem desconhecida. Assim, qualquer técnica de classificação capaz de classificar textos claros, levando em conta a distribuição estatística dos elementos do conjunto léxico (ou conjuntos léxicos) usado por tais textos, pode também classificar criptogramas.

Neste contexto, são formulados os resultados teóricos, onde são apresentadas as proposições e suas respectivas provas de que as cifras de blocos geram linguagens desconhecidas e que se pode identificar estas cifras (ou uma combinação de parâmetros criptográficos) a partir de identificadores de contexto lingüístico contidos nestas linguagens. Os resultados teóricos apresentados são válidos para qualquer linguagem desconhecida.

Na taxinomia proposta anteriormente para os ataques, pode-se ver que o item “a” – ataque somente com texto cifrado – é o ataque em que o criptoanalista dispõe de menos informações. Assim, é necessário se ter pelo menos os criptogramas e o algoritmo que os cifrou. Logo, a identificação da cifra pode ser o primeiro passo nesse tipo de ataque.

Na figura 1.1, nota-se que o canal de informação é vulnerável a ataques passivos, como análise de tráfego, mesmo quando protegido por criptografia. Assim, a metodologia apresentada pode obter a identificação do algoritmo criptográfico, a detectar a mudança do algoritmo utilizado na cifração, ou ainda, detectar a mudança na chave criptográfica utilizada para cifrar a mensagem que trafega na rede. Logo, tal metodologia é considerada, por analogia, como um ataque de distinção.

#### 1.4 CENÁRIOS PARA IDENTIFICAR CIFRAS DE BLOCO

Neste item serão apresentados os cenários utilizados nos processos e mecanismos para a construção da metodologia para identificar cifras de blocos.

---

<sup>8</sup> Neste trabalho, os termos idioma, língua e linguagem serão usados de maneira indistinta.

#### 1.4.1 COMBINAÇÃO DE PARÂMETROS CRIPTOGRÁFICOS

Neste cenário, se supõem que uma combinação qualquer de parâmetros criptográficos determina um conjunto léxico próprio e único. Desta forma, os criptogramas podem ser agrupados e classificados pelos mesmos processos utilizados para agrupar textos legíveis, usando como critério uma combinação qualquer de parâmetros criptográficos.

Em termos de resultados da metodologia apresentada neste trabalho para identificar cifras de bloco, tal cenário indica que a combinação de parâmetros criptográficos deixa uma assinatura nos criptogramas, o que pode permitir a identificação de tal combinação.

#### 1.4.2 IDENTIFICAÇÃO PELO NÚMERO DE RODADAS

Este cenário foi criado a partir dos resultados dos Testes Estatísticos propostos pelo NIST, para os algoritmos finalistas do AES. Estes testes comprovam que os algoritmos finalistas se estabilizam estatisticamente até a quarta rodada (SOTO e BASSHAM, 2000). A partir deste ponto, supõe-se que os mesmos já geraram transformações particulares suficientes para permitir a sua identificação.

#### 1.4.3 IDENTIFICAÇÃO DE COMPOSIÇÃO DE CIFRA DE BLOCO FIXA

Uma ou mais cifrações de um mesmo criptograma com a mesma cifra não impede e nem enfraquece a identificação. Dessa forma, o objetivo desse cenário é fazer cifrações sucessivas de um criptograma com o mesmo algoritmo para que o processo de cifração torne o criptograma mais forte, em processo semelhante ao que acontece com o algoritmo Triple DES, no esquema EEE<sup>9</sup> (STALLINGS, 2010). Da mesma forma que no cenário 1.4.2, se a partir do número mínimo de rodadas recomendadas para os algoritmos, os mesmos geraram transformações particulares suficientes para permitir a sua identificação, mais rodadas não dificultarão a identificação do algoritmo, embora teoricamente o processo de cifração possa se tornar mais forte (seguro).

---

<sup>9</sup> Encryption, Encryption, Encryption – Cifração, Cifração, Cifração.

#### 1.4.4 IDENTIFICAÇÃO DE COMPOSIÇÃO DE CIFRA DE BLOCO DISTINTA

Uma ou mais cifrações de um mesmo criptograma com algoritmos diferentes resulta na identificação do último algoritmo utilizado no processo de cifração. Por exemplo, considerando o cenário 1.4.1, fixando o modo de operação e a chave a cada composição, tem-se que um algoritmo criptográfico determina um conjunto léxico. Logo, uma cifração com uma composição de cifras vai resultar na identificação do último algoritmo utilizado no processo de cifração.

Por exemplo, cifrando um texto com a composição:

$$MARS_{nr,k_i}^{MOp} (RC6_{nr,k_i}^{MOp} (Twofish_{nr,k_i}^{MOp} (texto_1)))$$

Onde:

1.  $MOp$  denota o modo de operação da cifra de bloco;
2.  $nr$  é o número de rodadas; e
3.  $k_i$  é a  $i$ -ésima chave secreta.

O criptograma gerado deste processo, quando submetido à identificação, vai revelar que o mesmo foi cifrado pelo algoritmo MARS.

#### 1.5 OBJETIVOS DA TESE

Neste item, serão descritos os objetivos do presente trabalho, os quais resultarão no desenvolvimento de uma metodologia de identificação de  $n$ -cifras de bloco e na formulação de teoria associadas às linguagens desconhecidas e seus respectivos contextos lingüísticos que permitem a identificação das cifras de blocos.

Assim, serão considerados os seguintes objetivos específicos:

1. Estruturar a metodologia de identificação para  $n$ -cifras de blocos, a partir de técnicas de agrupamento e classificação, para identificar um caso particular de  $n=5$  onde são utilizadas as cifras de blocos: MARS, RC6, Rijndael, Serpent e Twofish; nos modos de operação ECB e CBC;
2. Identificar as cifras MARS, RC6, Rijndael, Serpent e Twofish a partir dos criptogramas gerados por estas cifras;

3. Demonstrar que a identificação pode ser feita para qualquer parâmetro criptográfico variável, desde que os demais permaneçam fixos;
4. Demonstrar que a identificação pode ser feita a partir da primeira rodada de uma Rede de Substituição e Permutação;
5. Demonstrar que uma ou mais cifrações de um mesmo criptograma com a mesma cifra não impede e nem enfraquece a identificação;
6. Demonstrar que uma ou mais cifrações de um mesmo criptograma com algoritmos diferentes resulta na identificação do último algoritmo utilizado no processo de cifração.
7. Demonstrar a existência de identificadores de contextos nos criptogramas;
8. Demonstrar que qualquer técnica de classificação capaz de classificar textos levando em conta a distribuição estatística dos elementos do conjunto léxico (ou conjuntos léxicos) usado por tais textos, pode também classificar criptogramas;
9. Generalizar o mecanismo identificador de  $n$ -cifras de blocos, estruturado no objetivo “1”, para um  $n$  qualquer; e
10. Apresentar resultados teóricos envolvendo a metodologia.

## 1.6 MOTIVAÇÃO

Observando os trabalhos de CARVALHO (2006), SOUZA (2007) e SOUZA *et al* (2008), nota-se que mesmo o AES, que foi aprovado nos testes estatísticos do NIST, foi suscetível ao processo de agrupamento de seus criptogramas em função da chave de cifrar. Este fato indica a possibilidade de agrupamento de criptogramas em função do algoritmo, o qual poderia levar à classificação e consequente identificação do algoritmo criptográfico. A partir disso, é possível a estruturação de um ataque mais forte de análise de tráfego de mensagens que transitam em canal de comunicação protegido por criptografia, permitindo a obtenção de informações sobre os parâmetros criptográficos utilizados para proteger as mensagens.

Outra motivação para este trabalho está nos ataques onde o criptoanalista dispõe de menos informações, por exemplo, “ataque somente com texto cifrado” onde é necessário se ter pelo menos os criptogramas e o algoritmo que os cifrou. Assim, o primeiro passo nesse tipo de ataque seria a identificação da cifra. Alguns trabalhos, como os de MAHESHWARI (2001) e SAXENA (2008), têm como principal motivação este fato.

Uma busca na literatura relativa à criptografia não encontrou trabalhos relevantes relacionados ao proposto neste trabalho, além da bibliografia citada. Uma boa fonte de informação sobre a ciência da criptologia é o CryptoDB (IACR 2009), banco de dados mantido pela Associação Internacional de Pesquisa em Criptologia (IACR) e que possui os artigos das principais conferências internacionais sobre criptologia: Crypto, Eurocrypt, Asiacrypt, Workshop on Cryptographic Hardware and Embedded Systems (CHES), Public Key Cryptography Conference (PKC), Theory of Cryptography Conference (TCC), Workshop on Fast Software Encryption (FSE) e também da Revista de Criptologia do IACR (Journal of Cryptology).

## 1.7 CARACTERIZAÇÃO DO PROBLEMA

A criptografia é tratada oficialmente pelo Brasil como assunto de interesse para o desenvolvimento e segurança nacionais. A estratégia Nacional de Defesa (DEFESA, 2009) prevê, em seu capítulo sobre Segurança Nacional, “o aperfeiçoamento dos dispositivos e procedimentos de segurança que reduzam a vulnerabilidade dos sistemas relacionados à Defesa Nacional contra ataques cibernéticos”. Tal tarefa inclui o uso intensivo de criptografia.

O uso da criptografia agregado aos sistemas computacionais de uma organização pode se concretizar de três maneiras:

a. **Padrão internacional:** A organização pode utilizar um algoritmo criptográfico que seja listado como padrão, por exemplo, o DES ou o AES.

Vantagens: Os padrões são intensamente “atacados” pela comunidade mundial de criptologia e caso tenha vida longa a sua confiabilidade é presumível.

Desvantagem: É possível que alguma agência ou órgão qualquer tenha descoberto fraquezas no algoritmo, mas não as tenha divulgado.

b. **Sistemas criptográficos de terceiros:** A organização usa produtos criptográficos desenvolvidos, codificados, implementados ou embutidos em sistemas por terceiros.

Vantagens: Baixo impacto na operação e desenvolvimento do sistema.

Desvantagem: Delegação de segurança. Neste caso, a segurança é medida não pela força da solução, mas pela confiança que se tem no fornecedor.

c. **Desenvolvimento dos próprios algoritmos criptográficos:** A organização desenvolve os seus próprios produtos criptográficos.

Vantagens: Conhecimento completo sobre o algoritmo criptográfico.

Desvantagem: A organização precisa garantir a confiabilidade do algoritmo criptográfico.

A escolha do tipo de uso depende do ativo de informação que se deseja proteger. No caso de agências governamentais que tratam de informações que requerem alta proteção e no caso de informações operativas de Forças Armadas, não é seguro a utilização de produtos criptográficos desenvolvidos para o público em geral. Tais organizações devem optar pelo item “c” acima.

O projeto e a construção de algoritmos criptográficos requerem um processo que garanta a confiabilidade desses algoritmos. O NIST propõe um conjunto com 16 testes para verificar a aleatoriedade dos criptogramas (NIST 2008). O NESSIE<sup>10</sup> propõe outro conjunto de testes estatísticos, que tem como subconjunto o conjunto de testes do NIST e que também tem a finalidade de verificar a aleatoriedade dos criptogramas (NESSIE 1999). Existem outras propostas no campo dos testes estatísticos, como os testes Diehard (DIEHARD, 2009). Contudo, mesmo o AES, aprovado nos testes estatísticos do NIST, foi suscetível ao processo de agrupamento de seus criptogramas em função da chave de cifrar, como pode ser visto nos trabalhos exploratórios de CARVALHO (2006), SOUZA (2007) e SOUZA *et al* (2008).

Pelo exposto, pode-se notar que outros testes complementares aos testes estatísticos são necessários para o aumento da confiabilidade de uma cifra, corroborando MURPHY (2000) sobre a necessidade de testes complementares aos propostos pelo NIST (2008). Logo, cabe verificar a extensão da possibilidade de classificação de algoritmos criptográficos em função de parâmetros criptográficos, os quais permitam identificar tais parâmetros, como o algoritmo criptográfico e o modo de operação.

## 1.8 ORGANIZAÇÃO DA DISSERTAÇÃO

No Capítulo 2 é realizada a fundamentação teórica necessária para o desenvolvimento deste trabalho. No Capítulo 3 será apresentada uma revisão que descreve trabalhos na área de identificação de cifras de blocos, assim como os seus respectivos resultados. No Capítulo 4 é apresentada a metodologia que será utilizada para identificar  $n - cifras$  de bloco. O Capítulo 5 apresenta as bases de dados empregados nos experimentos, assim como os experimentos

---

<sup>10</sup> New European Schemes for Signatures, Integrity, and Encryption.

computacionais, seus resultados e avaliações para o caso particular de 5 – *cifras* . O Capítulo 6 apresenta os resultados teóricos obtidos no presente trabalho. Uma discussão sobre os resultados experimentais e teóricos, considerando os cenários e os objetivos, e os trabalhos futuros é realizada no Capítulo 7. A conclusão é apresentada no capítulo 8.



## 2 FUNDAMENTOS TEÓRICOS PARA IDENTIFICAÇÃO DE CONTEXTOS LINGÜÍSTICOS E DE *N-CIFRAS*

Neste capítulo serão apresentados os conceitos básicos de criptografia, lingüística computacional e recuperação de informações necessários para o desenvolvimento da metodologia para Identificação de Cifras de Bloco.

### 2.1 SISTEMA CRIPTOGRÁFICO

**Definição 2.1:** Um sistema criptográfico pode ser definido como uma quintupla  $(T, C, K, E, D)$ , onde as seguintes condições são satisfeitas (STINSON, 2006):

1.  $T = \{t_1, t_2, \dots, t_n\}$  é um conjunto finito de possíveis textos claros;
2.  $C = \{c_1, c_2, \dots, c_n\}$  é um conjunto finito de possíveis criptogramas;
3.  $K = \{k_1, k_2, \dots, k_n\}$ , o espaço de chaves, é um conjunto finito de possíveis chaves;
4. Para cada  $k \in K$ , existe uma regra de cifração  $e_k \in E$  e uma correspondente regra de decifração  $d_k \in D$ , onde  $e_k : T \rightarrow C$  e  $d_k : C \rightarrow T$  são funções, tais que  $d_k(e_k(t)) = t$  para todo texto claro  $t \in T$ .

### 2.2 SUBSTITUIÇÃO, PERMUTAÇÃO, CONFUSÃO E DIFUSÃO

A criptografia clássica se utiliza de duas transformações básicas: a transposição e a substituição. Na transposição é feita uma permutação dos caracteres da mensagem, de acordo com algum critério. A ordem original dos caracteres é modificada, de maneira que o texto fica embaralhado (KAHN, 1967). É importante notar que o conjunto de caracteres da mensagem não se altera, apenas as suas posições são modificadas (SINGH, 2001). Dada a sua forma de operação, a transposição é também conhecida como permutação. Na substituição, cada caractere da mensagem é substituído por outro caractere, presente ou não na mensagem original.

SHANNON (1949) sugeriu dois métodos para frustrar a análise estatística dos criptogramas: a confusão e a difusão. A confusão visa tornar obscura e complexa a relação entre o texto claro, a chave e o criptograma e pode ser obtida por meio de substituição. A

difusão visa dissipar os padrões estatísticos e as redundâncias do texto claro, distribuindo-os caracteres (ou bits) ao longo do criptograma e pode ser obtida por meio de permutação.

Embora o artigo de SHANNON (1949) se refira às cifras clássicas, as modernas cifras de blocos utilizam tanto a confusão quanto a difusão em suas transformações criptográficas (SCHNEIER, 1996). Como se vê, as primeiras técnicas da história da criptografia, a substituição e a permutação, são utilizadas até hoje, embora o seu uso seja potencializado pelo poder computacional existente.

### 2.3 REDE DE SUBSTITUIÇÕES E PERMUTAÇÕES

Uma Rede (ou cadeia) de Substituições e Permutações (RSP) é um cifrador produto (ou composição de cifras) composto por mais de um estágio, cada um deles envolvendo substituições e permutações (MENEZES, 1996). Cada um desses estágios é também chamado de rodada. A quantidade de rodadas é um parâmetro importante para a segurança da cifras de blocos.

Os princípios de projeto de cifras de blocos indicam que quanto maior o número de rodadas da cifra, mais difícil é realizar um ataque contra esta cifra (STALLINGS, 2010). De fato, pode-se ver no projeto do algoritmo RC6, que versões deste algoritmo com poucas rodadas são suscetíveis aos ataques de criptoanálise diferencial e linear (RIVEST, 1998). Neste ataques, no caso do RC6, a quantidade de textos necessários para realizar a criptoanálise diferencial é de  $2^{56}$  com oito rodadas e de  $2^{238}$  com 20 rodadas e para realizar a criptoanálise linear é de  $2^{47}$  com oito rodadas e de  $2^{155}$  com 20 rodadas (CONTINI, 1998).

Um dado importante é que todos os algoritmos finalistas do AES se estabilizam estatisticamente até a quarta rodada (SOTO e BASSHAM, 2000).

### 2.4 CIFRAS DE BLOCO

Uma cifra de bloco consiste de um processo criptográfico (cifração/decifração), que transforma blocos de bits de uma mensagem (texto claro) em blocos de bits de um criptograma (texto cifrado). O tamanho de cada bloco (bits) pode ser definido em função de parâmetros criptográficos, como: o algoritmo, o tamanho da chave ou o número de rodadas. Neste trabalho usaremos blocos de tamanho  $n$ , com  $n = 128$ .

**Definição 2.2:** Seja  $c$  um criptograma,  $t$  um texto claro,  $k \in K$  e  $e_k$  e  $d_k$  funções de cifração e decifração, respectivamente. Diz-se que a quintupla  $(c, t, k, e_k, d_k)$  é uma cifra de bloco se  $e_k(t) = c$ , divide  $t$  em  $t_1, t_2, \dots, t_n$ , obtendo  $c_1, c_2, \dots, c_n$  e  $d_k(c) = t$ , divide  $c$  em  $c_1, c_2, \dots, c_n$  obtendo  $t_1, t_2, \dots, t_n$ , onde:

1.  $|t_1| = |t_2| = \dots = |t_n|$  e  $|t_1| + |t_2| + \dots + |t_n| = |t|$ ;
2.  $|c_1| = |c_2| = \dots = |c_n|$ ,  $|c_1| + |c_2| + \dots + |c_n| = |c|$ ; e
3.  $|c| = |t|$ .

Da definição acima, observe que os blocos do texto claro têm o mesmo tamanho dos blocos do texto cifrado.

## 2.5 COMPOSIÇÃO DE CIFRAS DE BLOCO

Uma composição é uma maneira de construir funções mais complexas a partir de outras funções mais simples (MENEZES, 1996).

**Definição 2.3:** Seja  $T$  um conjunto finito de possíveis textos claros, seja  $C_1$  e  $C_2$  dois conjuntos finitos de possíveis criptogramas e seja  $e_k^1 : T \rightarrow C_1$  e  $e_k^2 : C_1 \rightarrow C_2$  duas funções. A composição de  $e_k^2$  com  $e_k^1$ , denotada por  $e_k^2 \circ e_k^1$  é uma função de  $T$  para  $C_2$ , tal que  $e_k^2 \circ e_k^1 : T \rightarrow C_2$ , e definida por  $(e_k^2 \circ e_k^1)(t) = e_k^2(e_k^1(t))$ ,  $\forall t \in T$ .

A definição acima pode ser estendida para três ou mais funções. Assim, para as funções  $e_k^1, e_k^2, \dots, e_k^m$ , pode-se definir  $e_k^m \circ \dots \circ e_k^2 \circ e_k^1$ , com tanto que o domínio de  $e_k^m$  seja igual ao contradomínio de  $e_k^{m-1}$  e assim por diante (MENEZES, 1996).

## 2.6 MODOS DE OPERAÇÃO: ECB e CBC

Os modos de operação têm a finalidade de aperfeiçoar o resultado de uma cifra ou de preparar uma cifra para um uso particular ou prover funcionalidades para uma aplicação (STALLINGS, 2010), (NIST, 1980) e (NIST<sup>B</sup>, 2001). Neste trabalho, vamos considerar os modos ECB e CBC, conforme as definições a seguir.

**Definição 2.4.** Sejam:  $k \in K$ ,  $e_k \in E$ ,  $c_i \in c$  e  $t_i \in t$ , onde  $i = 1, \dots, n$ . Definimos como ECB, o modo de operação onde  $\forall i = 1, \dots, n$ ,  $c_i = e_k(t_i)$ , onde  $t_i$  é um bloco de texto claro.

Alguns autores não recomendam o uso deste modo de operação, como FERGUSON *et al* (2010). Entretanto, este é o modo utilizado nos testes de aleatoriedade do NIST (SOTO, 1999), por ser o que permite que a cifração seja a mais semelhante com a transformação original proposta pela cifra, já que as cifras de blocos são projetadas para cifrar a mensagem por bloco de bits ou bytes (DAEMEN, 2002). Uma particularidade na metodologia empregada pelo NIST é o uso de textos claros aleatórios nos testes estatísticos. Como será visto mais adiante, o sucesso dos resultados desta tese para o modo ECB se baseiam no seguinte: Se  $i \neq j$  e  $t_i = t_j$ , então temos que  $e_k(t_i) = e_k(t_j)$ . Em outras palavras,  $c_i = c_j$ .

**Definição 2.5.** Sejam:  $k \in K$ ,  $e_k \in E$ ,  $c_i \in c$  e  $t_i \in t$ , onde  $i = 1, \dots, n$  e  $c_0 = VI$  (Vetor de Inicialização). Definimos como CBC, o modo de operação onde  $\forall i = 1, \dots, k$ ,  $c_i = e_k(t_i \oplus c_{i-1})$ , onde  $t_i$  é um bloco de texto claro e  $\oplus$  denota a operação “ou” exclusivo. Nota-se que tal modo faz um encadeamento entre os blocos, o que confere maior aleatoriedade ao criptograma gerado. O modo CBC é amplamente utilizado em criptossistemas (FERGUSON *et al*, 2010).

No que se segue,  $e_k^{ECB}$  e  $e_k^{CBC}$  denotarão regras de cifração nos modos ECB e CBC, respectivamente.

## 2.7 REDE DE FEISTEL TRADICIONAL: *I-ÉSIMA* RODADA

Neste item, será apresentado o processo criptográfico (cifração/decifração), da *i-ésima* rodada de uma Rede de Feistel Tradicional. Nesse tipo de rede, um bloco de comprimento  $n$  de uma mensagem é dividido em duas partes (subblocos) de comprimento  $n/2$ , portanto  $n$  deve ser par. Tais subblocos sofrem iterações (rodadas), durante o processo criptográfico.

Para ilustrar o processo, vamos considerar as seguintes notações:

- i.  $B_i$  é o bloco do texto claro ou cifrado de  $n$  bits;
- ii.  $L_i$  e  $R_i$  são subblocos de  $B_i$  de  $n/2$  bits, onde  $L_i$  é a parte esquerda e  $R_i$  a parte a direita;
- iii.  $k_i$  é o bloco de chave parcial;
- iv.  $nr$  é o número de rodadas;

- v.  $\oplus$  é a operação “ou exclusivo”; e
- vi.  $f$  é a função que define o processo de confusão da cifra de bloco.

Baseado no exposto acima e considerando uma Rede de Feistel Tradicional, temos para a  $i$ -ésima rodada o seguinte:

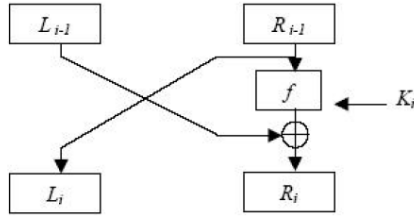


Figura 2.1 – Cifração na Estrutura de Feistel

**Definição 2.6.** Seja  $i$  a  $i$ -ésima iteração. Sejam  $B_i$  o bloco que será cifrado com a chave  $k_i$ ,  $f$  a função que será utilizada na cifração e  $L_i$  a parte esquerda do bloco  $B_i$  e  $R_i$  a parte direita do bloco  $B_i$ . Assim, a cifração é dada por:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

Onde  $1 \leq i \leq nr$ .

A decifração se baseia na propriedade de simetria da operação lógica XOR (ou exclusivo). Esta operação equivale à “soma módulo dois”, conforme a figura 2.2.

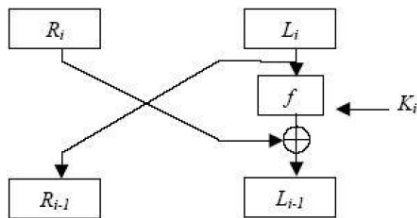


Figura 2.2: Decifração na Estrutura Feistel

**Definição 2.7.** Seja  $i$  a  $i$ -ésima iteração. Sejam  $B_i$  o bloco que será decifrado com a chave  $k_i$ ,  $f$  a função que será utilizada na cifração e  $L_i$  a parte esquerda do bloco  $B_i$  e  $R_i$  a parte direita do bloco  $B_i$ . Assim, a decifração é dada por:

$$R_{i-1} = L_i$$

$$L_{i-1} = R_i \oplus f(R_{i-1}, K_i)$$

Onde  $1 \leq i \leq nr$ .

## 2.8 LINGÜÍSTICA COMPUTACIONAL E RECUPERAÇÃO DE INFORMAÇÕES

O Objetivo desta seção é apresentar os conceitos básicos usados nos Sistemas de Recuperação de Informações, especialmente no agrupamento e classificação de informações e também alguns conceitos utilizados em lingüística computacional.

**Definição 2.8:** Um alfabeto  $\Sigma$  é um conjunto finito e não-vazio de símbolos  $\zeta$ , utilizados em uma determinada linguagem. Neste trabalho os termos linguagem e língua são usados indistintamente.

**Exemplo 2.1:** A língua portuguesa utiliza o alfabeto  $\Sigma = \{a, b, c, \dots, w, x, y, z\}$ .

**Exemplo 2.2:** O alfabeto ASCII é muito utilizado em computadores.

**Definição 2.9:** Uma palavra é uma seqüência finita e não-vazia de símbolos  $\zeta$  de um alfabeto  $\Sigma$ , que será denotada por  $\psi$ .

**Exemplo 2.3:** Se  $\Sigma$  é o código ASCII, então  $\psi = (a, B, 0, 1, @, M, 10)$  é uma palavra de  $\Sigma$ .

**Definição 2.10:** O comprimento (tamanho) de uma palavra é a quantidade de símbolos  $\zeta$  de  $\Sigma$ , repetidos ou não, utilizados para compor a palavra. Seja  $\psi$  uma palavra, denota-se por  $|\psi|$  o comprimento desta palavra.

**Definição 2.11:** Sejam  $\psi_i = (\zeta_{1i}, \zeta_{2i}, \dots, \zeta_{mi})$  e  $\psi_j = (\zeta_{1j}, \zeta_{2j}, \dots, \zeta_{pj})$  duas palavras de comprimento  $m$  e  $p$ . Define-se  $\psi_i \psi_j = (\zeta_{1i}, \zeta_{2i}, \dots, \zeta_{mi}, \zeta_{1j}, \zeta_{2j}, \dots, \zeta_{pj})$  como a concatenação de  $\psi_i$  e  $\psi_j$ .

**Definição 2.12:** Seja  $\Sigma$  um alfabeto. Define-se  $\Sigma^*$  como o conjunto de todas as palavras  $\psi$  sobre  $\Sigma$ .

**Definição 2.13:** Sejam  $\Sigma$  um alfabeto de símbolos  $\zeta$  e  $\psi$  uma palavra de comprimento  $n$ . Define-se:

$$\Sigma^n = \{\psi \in \Sigma^* \mid |\psi| = n\}, \text{ onde } \Sigma^n \subseteq \Sigma^*.$$

Em outras palavras,  $\Sigma^n$  é o conjunto de palavras de comprimento  $n$  sobre  $\Sigma$ .

**Exemplo 2.4:** Sejam  $\Sigma = \{a, e, i, o, u\}$  e  $|\psi| = 5$ . Determine  $\Sigma^n$ .

Solução:

$$\Sigma^5 = \{\psi \in \Sigma^* \mid |\psi| = 5\}. \text{ Daí vem:}$$

$$\psi_1 = (a, e, i, o, u), \quad \psi_2 = (e, a, i, o, u), \quad \psi_3 = (e, i, o, a, u), \dots, \psi_{3125} = (i, i, i, i, i).$$

$$\text{Finalizando, tem-se: } \Sigma^5 = (\psi_1, \psi_2, \dots, \psi_{3125}).$$

**Definição 2.14:** Seja  $\Sigma$  um alfabeto e  $\Sigma^n \subseteq \Sigma^*$ , onde  $\Sigma^n = \{\psi_1^n, \dots, \psi_m^n\}$ ,  $|\psi_m^n| = n$ . A  $n$ -upla  $L = \{\Sigma^1, \Sigma^2, \dots, \Sigma^n\}$  ou  $L = \{\{\psi_1^1, \dots, \psi_m^1\}, \{\psi_1^2, \dots, \psi_p^2\}, \dots, \{\psi_1^n, \dots, \psi_q^n\}\}$ , onde  $n$  denota o comprimento e  $m, o, p$  denotam a ordem, com  $n, m, p, q \in \aleph$ , define uma linguagem  $L$  sobre  $\Sigma$ , se  $L \subseteq \Sigma^*$ .

**Exemplo 2.5:** Se  $\Sigma = \{a, e, i, o, u\}$ , determine  $L$ .

Solução:

$$\Sigma^1 = \{\psi_1^1 = (a), \psi_2^1 = (e), \psi_3^1 = (i), \psi_4^1 = (o), \psi_5^1 = (u)\}$$

$$\Sigma^2 = \{\psi_1^2 = (a, a), \psi_2^2 = (a, e), \psi_3^2 = (a, i), \dots, \psi_{25}^2 = (uu)\}$$

...

$$\Sigma^5 = \{\psi_1^5 = (a, e, i, o, u), \psi_2^5 = (e, a, i, o, u), \dots, \psi_{3125}^5 = (i, i, i, i, i)\}$$

Pelos exemplos 2.4 e 2.5, pode-se ver que existem 3125 possibilidades de se construir palavras desse alfabeto de comprimento igual a cinco. Disto,  $\Sigma^* = \{\Sigma^1, \Sigma^2, \Sigma^3, \Sigma^4, \Sigma^5\}$ .

Considerando agora outros aspectos de uma linguagem, como por exemplo, semântica, pode-se construir  $L \subset \Sigma^*$ .

Cabe ressaltar que considerando os criptogramas, tem-se  $L = \Sigma^n$ , se  $|\psi| = n$  e  $\psi$  é uma palavra de  $L$ .

**Definição 2.15:** Sejam  $\vec{d}_i = (d_{1,i}, d_{2,i}, \dots, d_{n,i})$  e  $\vec{d}_j = (d_{1,j}, d_{2,j}, \dots, d_{n,j})$ , representação vetorial de dois documentos em  $\aleph^n$ , para os quais se deseja obter a similaridade. O valor

relacionado à  $d_{l,i}$  é a frequência do  $l$ -ésimo termo (ou palavra) no documento  $d_i$ . O valor relacionado à  $d_{l,j}$  é a frequência do  $l$ -ésimo termo no documento  $d_j$ . A similaridade (ou grau de associação) entre  $d_i$  e  $d_j$  pode ser obtida aplicando a  $\vec{d}_i$  e  $\vec{d}_j$  uma função que mede a similaridade entre dois vetores, como, por exemplo,  $s_{medida}(\vec{d}_i, \vec{d}_j)$ . De modo análogo, se pode obter a similaridade entre dois criptogramas  $c_i$  e  $c_j$  e dois textos claros  $t_i$  e  $t_j$ .

**Definição 2.16:** Sejam  $D = \{d_1, d_2, \dots, d_n\}$  uma coleção de documentos,  $G = (g_1, g_2, \dots, g_m)$  um coleção de grupos,  $s_{medida}$  uma medida do grau de associação entre os elementos de  $D$  e  $M$  um método de agrupamento qualquer usado na Teoria de Recuperação de Informações. Diz-se que a quádrupla  $(D, G, s_{medida}, M)$  é um procedimento de agrupamento se  $M(D, s_{medida}) = G$ , onde:

- i.  $\forall g_i \in G, g_i$  contém um ou mais elementos de  $D$ ;
- ii.  $D = \bigcup_{i=1}^m \{g_i\}$ ;
- iii.  $g_i \neq \emptyset$ , onde o símbolo  $\emptyset$  denota o conjunto vazio; e
- iv.  $g_i \cap g_j = \emptyset, i \neq j$  e  $i, j = 1, \dots, m$ .

**Exemplo 2.6:** Sejam  $D = \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8\}$  e  $s_{medida}$  uma medida qualquer de similaridade, por exemplo,  $s_{medida}(\vec{d}_i, \vec{d}_j) = s_{co-seno}(\vec{d}_i, \vec{d}_j)$ . Aplicando-se o método de agrupamento  $M(D, s_{medida})$ , obtemos, por exemplo,  $G = \{g_1, g_2, g_3, g_4\}$ , onde:  $g_1 = \{d_1, d_3, d_5\}$ ,  $g_2 = \{d_2, d_4, d_7\}$ ,  $g_3 = \{d_6\}$  e  $g_4 = \{d_8\}$ .

No que se segue, serão adotadas as seguintes convenções:

- i. As palavras dos documentos, textos claros ou textos cifrados (criptogramas) serão também palavras de tamanho fixo obtidos de uma dada linguagem  $L$ . Por exemplo, a Constituição Federal do Brasil poderá ser considerada como um documento ou um texto claro, se não for cifrada por algum algoritmo criptográfico, cujas palavras pertencem também à língua portuguesa. Na forma cifrada ela poderá ser considerada como um texto cifrado, entretanto as palavras serão diferentes e também pertencerão a uma língua que, a priori, é desconhecida.

- ii. Devido aos aspectos computacionais, o alfabeto  $\Sigma$  será o código ASCII com suas respectivas representações numéricas;



- iii.  $\mathfrak{R}^n = \mathfrak{R} \times \mathfrak{R} \times \dots \times \mathfrak{R}$ , denotará o espaço vetorial de dimensão  $n$ ; e
- iv.  $\vec{D} = (\vec{d}_1, \vec{d}_2, \dots, \vec{d}_n)$ , denotará uma representação vetorial da coleção de documentos  $D$  em  $\mathfrak{R}^n$ .

**Definição 2.17:** Sejam  $L = (\psi_1, \psi_2, \dots, \psi_m)$  uma linguagem sobre  $\Sigma$ , e  $D$  uma coleção de documentos de contexto  $\psi_r$ , tal que  $\psi_r \in L$ . Define-se um Identificador de Contexto  $IdC_{\psi_r}$  por uma seqüência de palavras de  $L$ , repetidas ou não, que identifica o contexto  $\psi_r \in L$ .

**Exemplo 2.7:** Sejam  $L$  a língua portuguesa,  $\Sigma =$  código ASCII, e  $\psi_r =$  economia. Então,  $IdC_{economia} = (\text{débito, crédito, oferta, demanda, custo, preço, } \dots)$ , define um Identificador em economia.

## 2.9 ESTRUTURA CRIPTOGRÁFICA DOS FINALISTAS DO AES

Nesta seção vamos descrever de forma sucinta as principais características e transformações criptográficas das cifras que foram finalistas do concurso AES (*Advanced Encryption Standard*).

O concurso AES foi lançado pelo NIST em janeiro de 1997, com o objetivo de escolher um novo padrão de algoritmo criptográfico para substituir o DES (*Data Encryption Standard*) e a sua principal variante conhecida por Triple DES. O avanço da tecnologia dos computadores permitiu que fossem exploradas fraquezas no DES, como o próprio tamanho da chave, tornando necessário o estabelecimento de um novo padrão criptográfico. A proposta principal do NIST era fazer um algoritmo tão seguro quanto o Triple DES, entretanto mais eficiente (DAEMEN, 2002).

Ao final da primeira etapa do concurso, cinco cifras foram escolhidas: MARS (IBM), RC6 (RSA), Rijndael (John Daemen e Vincent Rijmen), Serpent (Ross Anderson, Eli Biham e Lars Knudsen) e TwoFish (Bruce Schneier, John kelsey e outros), sendo o vencedor do concurso o Rijndael, tornando-se conhecido a partir de então como AES (NIST, 2001).

A seguir, são apresentadas as descrições das principais características e transformações criptográficas das cifras finalistas.

## 2.9.1 MARS

O MARS é uma cifra de bloco simétrica proposta por Carolynn Burwick, Don Coppersmith *et al* da IBM<sup>11</sup> (BURWICK *et al*, 1999), a qual trabalha com blocos de 128 *bits* e chaves com tamanhos variando entre 128 e 400 *bits*.

Com a finalidade de oferecer proteção contra os principais ataques criptoanalíticos, a cifra apresenta uma Estrutura Criptográfica constituída de três fases, as quais utilizam uma rede de Feistel tipo três (type-3 Feistel): a primeira fase com oito rodadas, a segunda com 16 rodadas, oito em modo *forward* e oito em modo *backward*, e a terceira com oito rodadas (figura 2.3). A cifra usa duas *S-box*<sup>12</sup> que implementam a operação de substituição, responsável pelo processo de confusão durante o processo criptográfico.

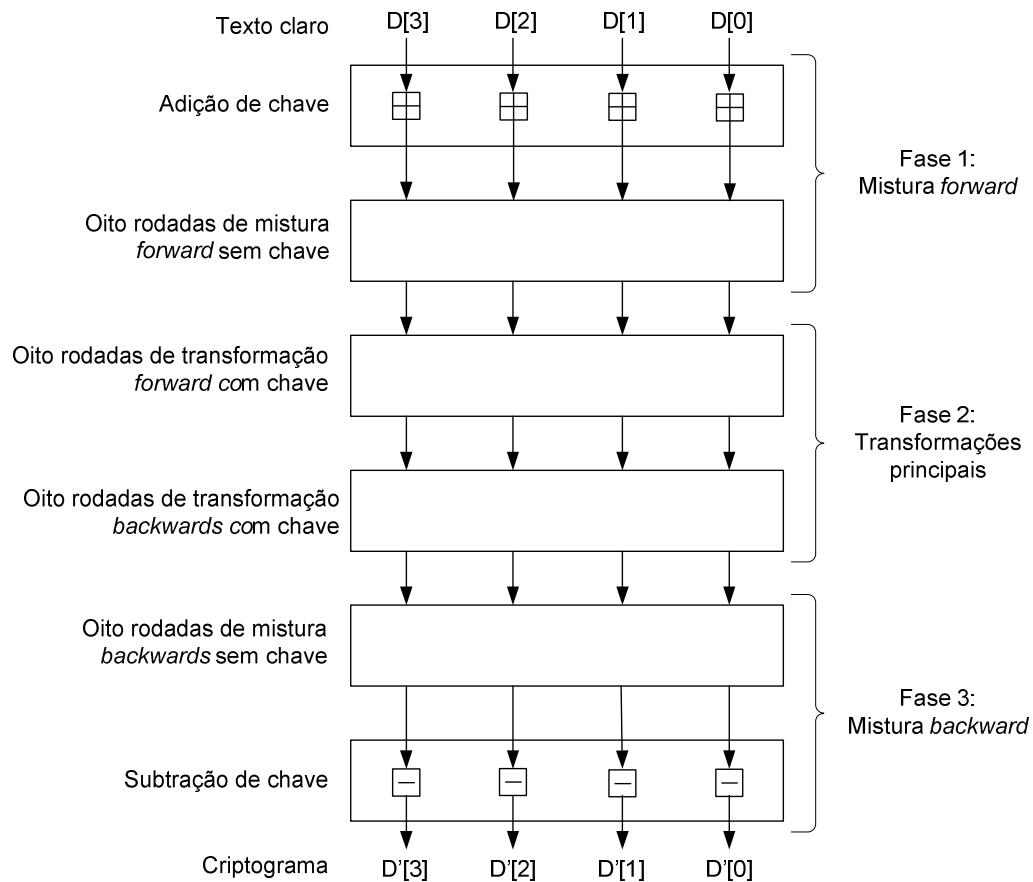


Figura 2.3: Estrutura Criptográfica do MARS (Adaptada de BURWICK, 1999).

<sup>11</sup> A descrição do MARS nesta seção é baseada em (BURWICK *et al*, 1999).

<sup>12</sup> Caixa de substituição ou caixa-S.

Uma rede de Feistel tipo três é composta de diversas rodadas, onde a cada rodada uma palavra da mensagem e algumas palavras da chave são utilizadas para modificar todas as outras palavras da mensagem.

O processo de cifração do MARS é realizado por meio das operações ou transformações criptográficas: Substituição, *XOR*, *ADD*, Rotação, Expansão, Multiplicação e Subtração.

Na primeira fase, quatro palavras das chaves parciais, obtidas pelo escalonador de chaves com auxílio da chave criptográfica, são adicionadas a cada palavra do bloco de entrada e então são executadas oito rodadas de transformações criptográficas sem chaves parciais (misturas *forward*) combinadas com outras transformações adicionais: *XOR*, *ADD* e Rotação; de forma que em cada rodada uma palavra do bloco é utilizada para modificar as outras três. Assim, a primeira palavra do bloco é o índice para as duas *S-box*. Operações *XOR* e *ADD* são realizadas sobre as outras três palavras do bloco. Ao final da rodada, uma rotação de 24 *bits* à direita é realizada sobre a primeira palavra do bloco e, finalmente, uma rotação sobre todas as palavras é realizada.

Na segunda fase são realizadas as principais transformações criptográficas, em um total de 16 rodadas, sendo oito em modo *forward* e oito no modo *backwards*, onde a cada rodada uma função *E* é executada. A função *E* consiste de uma combinação de multiplicações, rotações e uma consulta a uma *S-box*, recebendo uma palavra com entrada e fornecendo três palavras de saída. Estas três palavras são misturadas com outras três palavras por meio de operações *XOR* ou *ADD* e a palavra de entrada sofre uma rotação de 13 posições à esquerda.

A terceira fase consiste de oito rodadas, utiliza misturas *backwards* e é equivalente a decifração da primeira fase (mistura *forward*), exceto pelo fato de que os dados de entrada da terceira fase são processados na ordem inversa dos dados de saída da primeira fase. Por exemplo, os dados de saída da primeira fase  $D[3]$ ,  $D[2]$ ,  $D[1]$  e  $D[0]$  são colocados na entrada da terceira fase na ordem  $D[0]$ ,  $D[1]$ ,  $D[2]$  e  $D[3]$ . Assim, uma fase cancelaria a outra se fossem aplicadas em seqüência.

No processo de decifração, são utilizadas operações inversas das operações de cifração.

## 2.9.2 RC6

O RC6 é uma cifra de bloco simétrica, cujo projeto se baseia em modificações no algoritmo RC5, visando atender aos requisitos de segurança propostos pelo AES. Nesse sentido, no projeto do RC6 seus autores Ronald Rivest, Robshaw *et al*, da RSA

Laboratories<sup>13</sup> (RIVEST, 1998) e do MIT, consideraram todos os ataques contra o RC5 apresentados na literatura científica e propuseram mudanças para frustrar tais ataques quando empregados contra o RC6.

A cifra opera blocos de 128 *bits* e chaves com tamanhos variando entre 16 a 255 *bytes*. A execução normal do algoritmo é realizada em 20 rodadas, o que atende aos requisitos de segurança propostos no AES; embora ele seja parametrizado para executar  $r$  rodadas, onde  $8 \leq r \leq 32$ . Por ser bastante parametrizado ele é especificado como  $RC6-w/r/b$ , onde  $w$  representa o tamanho da palavra em *bits*,  $r$  o número de rodadas e  $b$  representa o tamanho da chave em *bytes*. Diferente dos demais finalistas, o RC6 não usa *S-box* apesar de implementar uma Rede de Feistel não tradicional, mas bastante adequada à tecnologia usada nos atuais computadores.

O processo de cifração do RC6 é realizado por meio das operações ou transformações criptográficas: *XOR*, *ADD*, Rotação à esquerda, Rotação à direita, multiplicação, permutação e Subtração.

Basicamente, o processo criptográfico é realizado em três fases descritas abaixo de forma sucinta. As seguintes notações serão consideradas:

1.  $A$ ,  $B$ ,  $C$ ,  $D$  são registradores (vetores) de 32 *bits* usados para armazenar cada bloco de entrada, durante o processo de cifração/decifração;
2.  $S$  é o vetor que armazena  $2r + 4$  chaves parciais de 32 *bits* geradas pelo escalonador de chaves do RC6, a partir da chave criptográfica (chave secreta ou chave de sessão de sessão) de 128 *bits*;
3.  $f$  é a permutação usada para modificar palavras de cada bloco com auxílio do vetor  $S$  e outras operações aritméticas como, por exemplo, rotações, adições, etc; e
4.  $t, u$  são variáveis inteiras auxiliares, usadas durante as rodadas da cifra.

Assim, as fases do processo de cifração do RC6 podem ser resumidas da seguinte maneira:

1. Fase 1 (Pré-Rodada): uma operação inicial de adição é realizada entre os conteúdos dos registradores  $B$  e  $D$  com as duas primeiras chaves parciais de  $S$ .
2. Fase 2 (Rodada): esta fase consiste de  $r$ -rodadas, onde todas as operações e transformações criptográficas da cifra RC6 são executadas da seguinte maneira: A função  $f$  é executada e o seu resultado é armazenado nas variáveis  $t$  e  $u$ . Continuando, uma operação

---

<sup>13</sup> A descrição do RC6 nesta seção é baseada em (RIVEST *et al*, 1998).



### 3.9.3 RIJNDAEL

O Rijndael é uma cifra de bloco simétrica, que opera blocos de 128 *bits* e chaves com tamanhos entre 128, 192 e 256 *bits*, embora no padrão estabelecido pelo NIST (2001), quando da vitória do Rijndael, seja adotado somente o bloco de 128 *bits*. A cifra foi projetada e apresentada ao AES por Joan Daemen e Vincent Rijmen da Banksys e da Universidade Católica de Leuven, respectivamente (DAEMEM e RIJMEN, 1999).

Em termos de Estrutura Criptográfica, o Rijndael não utiliza Rede de Feistel e seu processo de criptografia pode ocorrer em 10, 12 ou 14 rodadas, para as chaves de 128, 192 e 256 *bits*, respectivamente.

O processo de cifração do Rijndael é realizado por meio das operações ou transformações criptográficas: *ADD*, Substituição, Permutação, Multiplicação, Expansão e Rotação, representadas pelas funções *ADDRoundKey*, *SubBytes*, *ShiftRows*, *MixColumns*, *KeyExpansion* e *RotWord*, respectivamente. Neste trabalho, será utilizada a terminologia sugerida em LAMBERT (2004), conforme a tabela 2.1.

No processo de decifração, são utilizadas operações inversas das operações de cifração (NIST, 2001).

Tabela 2.1 – Terminologia alternativa para as operações do Rijndael

<b>Sugerida</b>	<b>FIPS 197</b>
SOMASUBCHAVE	<i>ADDRoundKey</i>
SUBSTITUIÇÃO	<i>SubBytes</i>
PERMUTAÇÃO	<i>ShiftRows</i>
MULTIPLICAÇÃO	<i>MixColumns</i>
EXPANSÃO	<i>KeyExpansion</i>
ROTAÇÃO	<i>RotWord</i>

A cifração é iniciada pela aplicação da transformação SOMASUBCHAVE que implementa uma soma módulo dois, i.e., uma operação lógica “OU EXCLUSIVO” realizada *bit a bit*, entre a chave e o bloco da mensagem.

A SUBSTITUIÇÃO é uma operação realizada *byte a byte*, na qual se utiliza uma matriz (*S-box*), onde é feito o cruzamento entre essa matriz e os *bytes* do bloco da seguinte maneira: a metade esquerda do *byte* de entrada na SUBSTITUIÇÃO indica a linha da matriz e a

metade direita indica a coluna da matriz. O byte de saída da SUBSTITUIÇÃO será obtido pela interseção da linha com a coluna dessa matriz.

Na PERMUTAÇÃO é simplesmente realizada uma troca na posição dos bytes de um bloco. Por exemplo, sejam  $B = (b_0, b_1, \dots, b_{15})$  um bloco de 16 bytes e  $P$  a operação de permutação. Então, após a operação, teríamos o seguinte arranjo:

$$P(B) = (b_0, b_5, b_{10}, b_{15}, b_4, b_9, b_{14}, b_3, b_8, b_{13}, b_2, b_7, b_{12}, b_1, b_6, b_{11}).$$

A operação de MULTIPLICAÇÃO é realizada por meio da multiplicação de uma matriz constante de  $32 \times 32$  bits. Assim, um bloco de 128 bits de entrada é dividido em quatro palavras (subblocos) de 32 bits. Cada palavra é então multiplicada pela matriz. As palavras resultantes da multiplicação são concatenadas, formando a saída da operação.

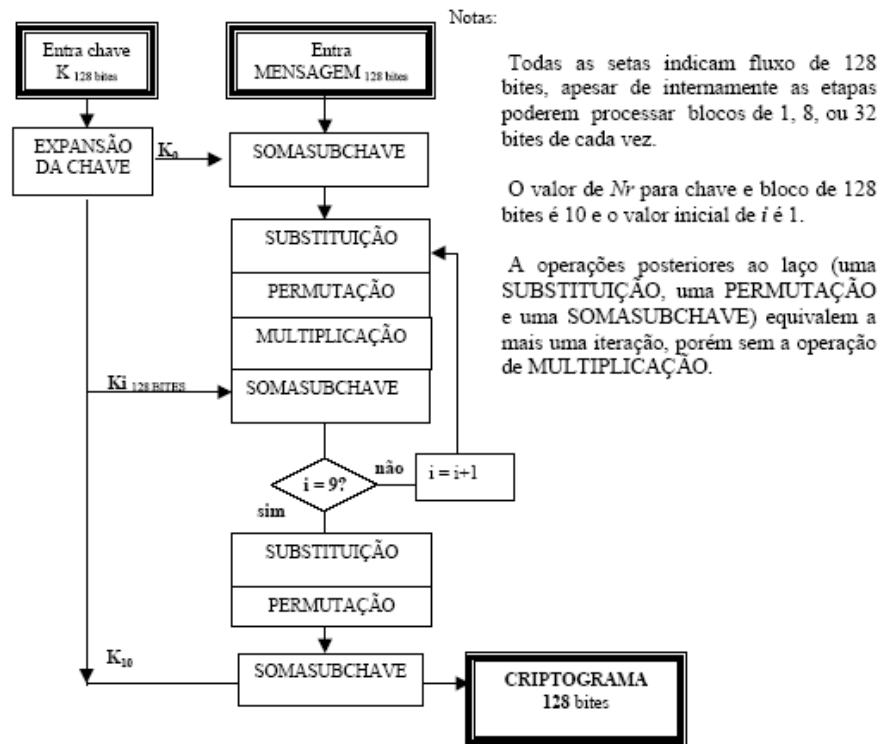


Figura 2.5 – Estrutura de Cifração do Rijndael (LAMBERT, 2004)

A operação de MULTIPLICAÇÃO pode ser encontrada de maneira detalhada em LAMBERT (2003), onde também encontramos uma proposta de simplificação para a mesma.

A EXPANSÃO consiste em gerar  $nr + 1$  subchaves, onde  $nr$  é o número de rodadas, que serão utilizadas pelas operações de SOMASUBCHAVE, ou seja, uma subchave para cada vez que a operação SOMASUBCHAVE for executada. A primeira subchave é a própria chave fornecida na entrada.

A ROTAÇÃO faz parte da EXPANSÃO e consiste na realização de um deslocamento cíclico de *bytes* à esquerda. Assim, seja  $B$  um bloco de 16 *bytes*, o qual foi dividido em quatro palavras de 32 *bits*. Seja  $R$  a operação de ROTAÇÃO. Então, temos:

$$B_{128} = (p_0, p_1, p_2, p_3), R(p_0, p_1, p_2, p_3) = (p_1, p_2, p_3, p_0).$$

O fluxo de execução do Rijndael é descrito na figura 2.5. O algoritmo recebe uma chave a qual é submetida a uma operação de EXPANSÃO. A finalidade desta operação é gerar um total de  $nr + 1$  subchaves que serão utilizadas ao longo do processo de cifração. Em seguida, é realizada a operação SOMASUBCHAVE, a qual realiza uma soma módulo dois entre o resultado da EXPANSÃO e a mensagem original. Nesta primeira etapa, a chave é a própria chave original recebida na entrada.

Na próxima etapa, inicia-se o laço, onde serão realizadas  $nr - 1$  operações de SUBSTITUIÇÃO, PERMUTAÇÃO, MULTIPLICAÇÃO e SOMASUBCHAVE.

Após  $nr - 1$  rodadas, o algoritmo realiza uma operação de SUBSTITUIÇÃO, PERMUTAÇÃO e SOMASUBCHAVE, sendo finalmente gerado o criptograma.

#### 2.9.4 SERPENT

O Serpent é uma cifra de bloco simétrica, proposta por Ross Anderson da Universidade de Cambridge, Eli Biham da Technion e Lars Knudsen da Universidade de Bergen<sup>14</sup> (ANDERSON *et al*, 1999). Ela opera blocos de 128 *bits* e chaves com tamanhos de 128, 192 e 256 *bits*. No processo criptográfico, a cifra implementa uma Rede de Feistel com 32 rodadas onde, os blocos de entrada (texto claro/texto cifrado) e chaves parciais(subchaves) são divididos em subblocos de 32 *bits*.

A cifra usa oito *S-box*, as quais foram inicialmente cópias das *S-box* do DES, e depois foram trocadas por outras oito *S-box* “mais fortes” que implementam a operação de Substituição. Outras operações como *XOR*, Rotação, Permutação e Deslocamento são também utilizadas no processo criptográfico. O processo de cifração do Serpent pode ser resumido nas seguintes fases:

1. Fase 1 (Pré-Rodada): Nesta fase uma permutação inicial é aplicada no bloco da mensagem (texto claro).

---

<sup>14</sup> A descrição do Serpent nesta seção é baseada em (ANDERSON *et al*, 1999).



2. Fase 2 (Rodada): Esta fase consiste de 32 rodadas onde, em cada rodada, são realizadas operações de: inserção de chave, considerando as 33 subchaves geradas pelo escalonador de chaves a partir da chave criptográfica; de substituição por meio das *S-box*, as quais são usadas por quatro vezes, cada uma mapeando quatro bits de entrada para quatro bits de saída; e uma transformação linear, na qual são executadas operações de Rotação, deslocamento e *XOR* sobre os blocos.

3. Fase 3 (Pós-Rodada): Uma permutação final é aplicada no bloco retornado após as 32 rodadas.

A figura 2.6 abaixo ilustra a Estrutura Criptográfica do processo de cifração da Serpent. Quanto à decifração, utilizam-se as operações inversas das empregadas na cifração.

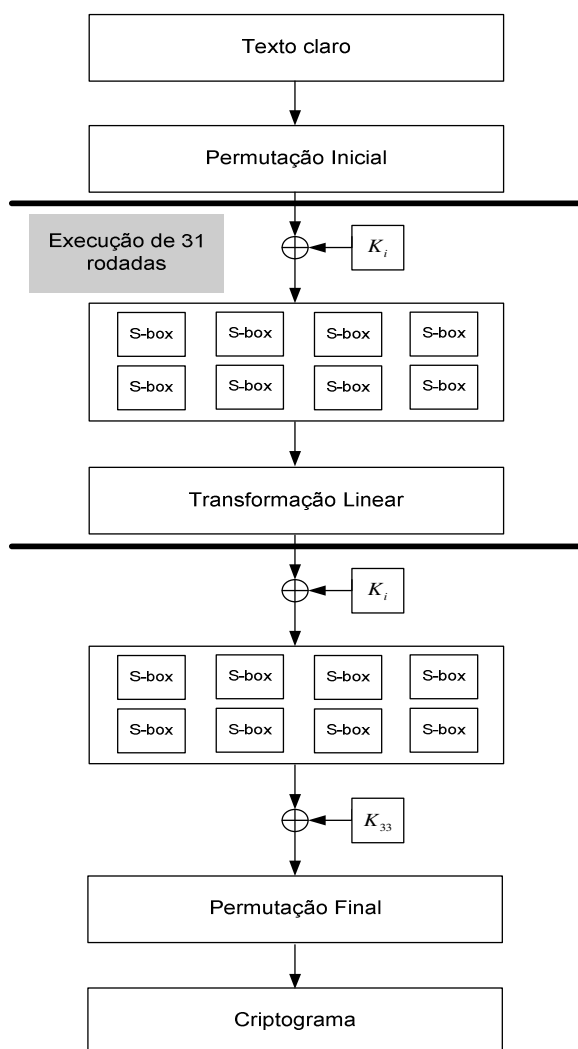


Figura 2.6 – Estrutura da Cifração do Serpent

### 3.9.5 TWOFISH

O Twofish é uma cifra de bloco simétrica proposta por Bruce Schneier, John Kelsey, Doug Whiting, Chris Hall e Neils Ferguson, da Counterpane Systems, e David Wagner da Universidade da Califórnia em Berkeley<sup>15</sup> (SCHNEIER *et al*, 1998). Ela opera com blocos de 128 *bits* e chaves criptográficas com tamanhos de 128, 192 e 256 *bits*. No processo criptográfico, blocos da mensagem e das chaves parciais (subchaves) são divididos em subblocos de 32 *bits*. A cifra utiliza a estrutura de Feistel com 16 rodadas e quatro *S-boxes*, as quais são dependentes da chave.

O processo de cifração do Twofish é realizado por meio das operações ou transformações criptográficas: Substituição (*S-boxes*), *XOR*, *ADD*, Rotação à esquerda, Rotação à direita, multiplicação, funções *F*, *g* e uma permutação. Além disso, a cifra utiliza também algumas estruturas particulares como *Pseudo-Hadamard Transforms* (PHT), Matrizes MDS, e *Whitening*. Para maiores detalhes sobre essas estruturas consultar SCHNEIER *et al* (1998).

No início da cifração as quatro palavras do texto claro passam por um *XOR* com as quatro palavras da chave. Esta operação é denominada *Whitening* de entrada (Pré-Rodada). Seguem-se então as 16 rodadas da estrutura de Feistel onde são executadas operações *XOR*, Rotações e uma função *F*. Na função *F* é executada a função *g*, a qual se constitui de Substituições com as *S-boxes* e as multiplicações com a matriz MDS. Ainda na função *F* são realizadas operações *ADD*, uma rotação e as adições da estrutura PHT, com a saída da função *g*. Ao final das 16 rodadas, uma permutação desfaz a permutação final realizada pela estrutura de Feistel e uma operação de *Whitening* de saída (Pós-Rodada) completa a cifração.

Basicamente, o processo de cifração do Twofish pode ser resumido nas seguintes fases:

1. Fase 1 (Pré-Rodada): nesta fase é feito um *XOR* entre os quatro subblocos da mensagem com quatro subblocos iniciais das chaves parciais.

2. Fase 2 (Rodada): consiste de 16 rodadas onde, em cada rodada, são executadas as operações aritméticas e as transformações criptográficas descritas acima. Antes de se executar a função *F*, executa-se a função *g*, constituída por substituições (*S-boxes*), multiplicações com a matriz MDS, adição, *XOR* e uma rotação. O valor retornado dessa

---

<sup>15</sup> A descrição do Twofish nesta seção é baseada em (SCHNEIER *et al*, 1998).

função é adicionado com a estrutura PHT, constituindo, assim, a saída da função  $F$ . Finalizando, a saída de  $F$  é adicionada com subchaves.

3. Fase 3 (Pós-Rodada): nesta fase, uma permutação desfaz a permutação final realizada pela Rede de Feistel e um XOR entre os quatro subblocos após as 16 rodadas, é realizado com quatro subblocos de chaves parciais.

A figura 2.7 abaixo ilustra a Estrutura de Cifração do Twofish. No processo de decifração, utilizam-se operações inversas aquelas empregadas na cifração.

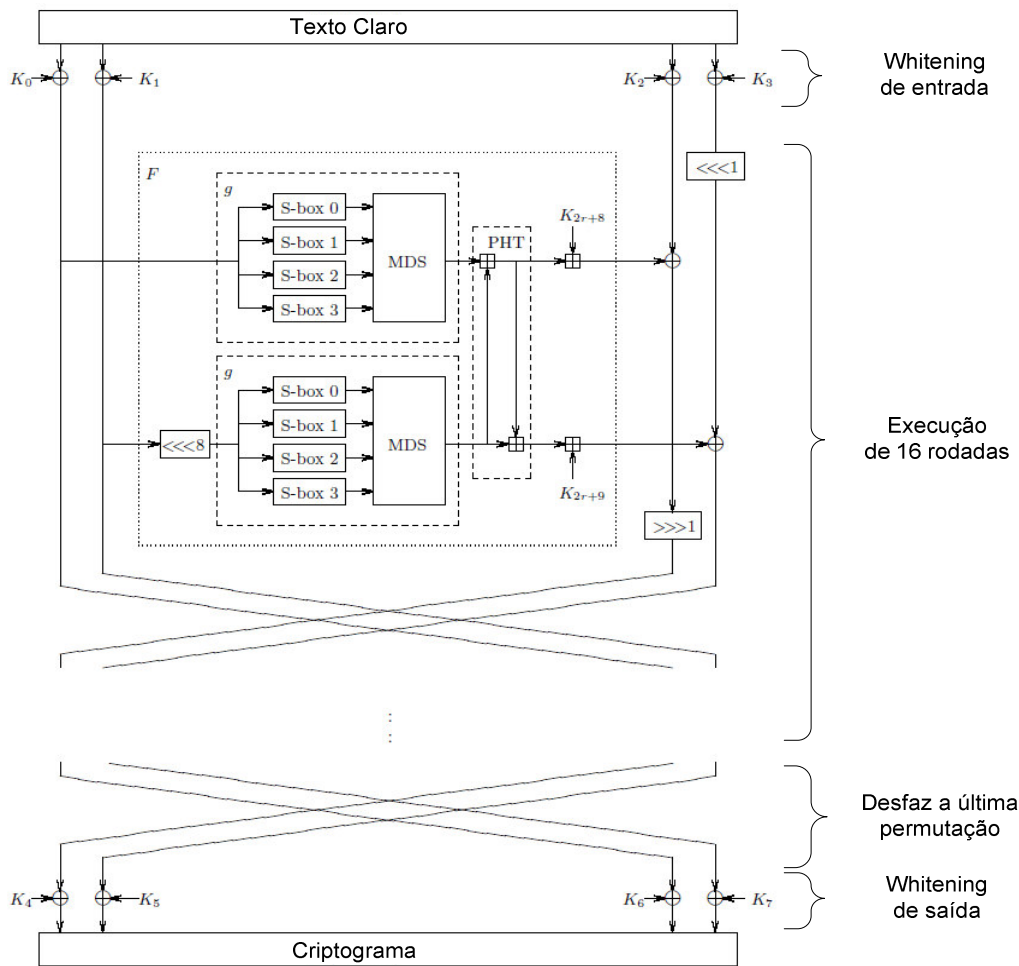


Figura 2.7 – Estrutura de Cifração do Twofish (adaptada de SCHNEIER, 1998)

## 2.9.6 RESUMO DAS TRANSFORMAÇÕES CRIPTOGRÁFICAS DOS FINALISTAS

Os algoritmos criptográficos transformam os textos claros em criptogramas por meio de operações ou transformações criptográficas, que podem ser lineares ou não-lineares. Cada um

dos algoritmos finalistas do concurso do AES possui essas operações projetadas e interagindo entre si de maneira particular, embora, na essência, esses algoritmos utilizem operações bem semelhantes.

As descrições desses algoritmos, descritas anteriormente, detalham as operações dos mesmos. Na tabela 2.2, vê-se uma comparação entre os algoritmos no que se refere às suas operações.

Tabela 2.2. Operações dos Algoritmos Finalistas do AES

Cifras	Operações						
	Linear					Não-linear	
	<i>XOR</i>	Adição	Subtração	Permutação	Rotação	Multiplicação	<i>S-box</i>
<b>MARS</b>	●	●	●		●	●	●
<b>RC6</b>	●	●	●	●	●	●	
<b>Rijndael</b>	●	●	●	●	●	●	●
<b>Serpent</b>	●			●	●		●
<b>Twofish</b>	●	●	●	●	●	●	●

A Estrutura de Feistel é utilizada pelos os algoritmos MARS, RC6, Serpent e Twofish.

O Twofish utiliza uma estrutura PHT que equivale a operações de Adição (*ADD*), operações de *Whitening* que equivale a operações *XOR* e matrizes MDS que equivalem a operações de Multiplicação.

As operações descritas na tabela 2.2 são baseadas na aritmética modular e os seus resultados são intrínsecos ao *modus operandi* de cada transformação. Assim, a cada entrada em uma operação corresponderá uma saída particular. Da mesma forma, a cada entrada submetida a um conjunto de operações que interagem entre si corresponderá uma saída particular. Esta última afirmação serve para definir os algoritmos criptográficos como um conjunto de operações, funções ou transformações matemáticas que recebem um texto claro na entrada e o apresenta, na saída, transformado em um criptograma.

Neste contexto, pode-se dizer que as propriedades intrínsecas dos modelos matemáticos dos algoritmos criptográficos transformam os textos claros em criptogramas de maneira particular. Esta particularidade contribui para a identificação do algoritmo por meio de técnicas de classificação.

### 3 RESULTADOS SOBRE IDENTIFICAÇÃO DE CIFRAS DE BLOCO

A metodologia que será apresentada neste trabalho, tem como objetivo identificar uma cifra de bloco a partir somente do criptograma gerado por tal cifra. Assim, a aplicação dessa metodologia representa um “ataque somente com texto cifrado”. Desta forma, ela considera que os criptogramas submetidos ao ataque tenham sido gerados pelo modelo completo das cifras<sup>16</sup> e que não se tenham informações sobre a mesma. Tal detalhe é importante ser ressaltado, já que muitos dos ataques propostos na literatura consideram um modelo reduzido da cifra.

Ressalta-se, também, que neste trabalho são realizados alguns experimentos que consideram ora o enfraquecimento, ora o fortalecimento do algoritmo criptográfico, com a finalidade de testar o processo de identificação de cifras de bloco proposto neste trabalho.

Assim, foram estudados modelos e métodos presentes na literatura que contemplam classificação, agrupamento, distinção e outros métodos para identificar as cifras de blocos, os quais são apresentados a seguir.

#### 3.1 MÉTODO BASEADO NO TESTE $\chi^2$

KNUDSEN e MEIER (2000) utilizam a estatística do  $\chi^2$  para, a partir de um conjunto de criptogramas, indicar se os mesmos foram gerados pelo algoritmo RC6 ou não. Tal trabalho teve com resultado 95% de acerto. A conclusão do trabalho indica que o algoritmo RC6 é vulnerável à estatística do  $\chi^2$ , com 15 iterações e com complexidade  $2^{125}$ . Os autores consideram que a distinção com 16 iterações é um problema em aberto.

Esta criptoanálise é uma das mais bem sucedidas técnicas contra o algoritmo RC6 sendo intensamente explorada (SANTOS, 2009), principalmente com modelos reduzidos do algoritmo, mas, ainda assim, a complexidade computacional e a complexidade de se montar o ataque são consideráveis. A técnica foi originalmente demonstrada para o algoritmo DES (VAUDINAY, 1999).

Tal técnica estatística pode ser usada em um ataque de recuperação de chaves (ISOGAI, 2003).

---

<sup>16</sup> Sem diminuição do número de rodadas e sem qualquer espécie de enfraquecimento da cifra.

## 3.2 MÉTODOS COM TESTES ESTATÍSTICOS E AS FUNÇÕES XOR E LIMAR

### 3.2.1 TESTES ESTATÍSTICOS E A FUNÇÃO XOR

No trabalho de MAHESHWARI (2001) é realizada a tentativa de identificação das cifras DES e IDEA, por meio de um processo de classificação, a partir de uma base de criptogramas cifrados por esses dois algoritmos.

A partir de uma base de 80 criptogramas: 40 cifrados pelo DES e 40 cifrados pelo IDEA; são aplicados os seguintes testes estatísticos para medir a aleatoriedade dos criptogramas: teste do  $\chi^2$ , de frequência, de corrida, de colisão, de intervalo, serial, poker e de permutação<sup>17</sup>; para identificar se, dado um criptograma, ele foi cifrado pelo DES ou pelo IDEA. Os experimentos consistiam em verificar se um determinado teste estatístico gerava resultados diferentes para criptogramas cifrados pelo DES ou pelo IDEA, de tal forma que seria possível identificar tais algoritmos. Os testes não foram capazes de identificar nenhum algoritmo corretamente.

Ainda, usando a mesma base de criptogramas acima, foram realizadas operações *XOR* nos criptogramas para identificar se, dado um criptograma, ele foi cifrado pelo DES ou pelo IDEA. Os experimentos consistiam em verificar se determinados arranjos de blocos submetidos a operações *XOR* seguidos da aplicação dos mesmos testes estatísticos geravam resultados diferentes para criptogramas cifrados pelo DES ou pelo IDEA, de tal forma que seria possível identificar tais algoritmos.

Os experimentos foram realizados tratando os criptogramas de três maneiras diferentes:

- i. O primeiro bloco do criptograma passou por um *XOR* com todo o criptograma, de maneira que o criptograma resultante tinha um bloco a menos que o criptograma original;
- ii. O criptograma passou por um *XOR* com uma cadeia de 64 *bits* aleatórios; e
- iii. Cada par de blocos do criptograma passou por um *XOR* resultando em um novo bloco, que substituiu o par de blocos utilizados na operação *XOR*. Assim, o novo criptograma possuía quase a metade<sup>18</sup> dos blocos que o criptograma original possuía.

Nenhum dos três experimentos foi capaz de identificar as cifras.

---

<sup>17</sup> Para maiores detalhes sobre os testes estatísticos, queira consultar MAHESHWARI (2001).

<sup>18</sup> Foi mantida a expressão “quase a metade”, embora imprecisa, para preservar a coerência com o texto original.

### 3.2.2 FUNÇÃO LIMIAR

**Definição 3.1:** Sejam  $f : \{0,1\}^m \rightarrow R$  uma função real e  $e_1^k$  e  $e_2^k$  cifras de bloco com uma chave fixa  $k$ . Diz-se que  $f$  é limiar se:

- i.  $f(x) = 1$ , se ela identifica o algoritmo  $e_1^k$ ; e
- ii.  $f(x) = 0$ , se ela identifica o algoritmo  $e_2^k$ .

O exemplo que se segue, ilustra uma aplicação desse tipo de função nos algoritmos DES e IDEA. Neste caso, será considerado que  $y = f(x) = 1$  identifica o DES e  $y = f(x) = 0$  identifica o IDEA.

Considerando  $x = (b_0, b_1, \dots, b_{319})$  os primeiros 320 bits de um criptograma cifrado pelo DES ou pelo IDEA. Considerando, ainda, que a função  $f$  tem a forma:

- i.  $f = \sum_{i=0}^{319} c_i b_i \geq T = 1$ , para o DES; e
- ii.  $f = \sum_{i=0}^{319} c_i b_i < T = 0$  para o IDEA.

Onde:  $c_i$  e  $T$  são números reais.

A função objetivo a ser maximizada  $\sum_{i=0}^{319} c_i - T$  foi utilizada, sujeita as seguintes restrições:

1.  $-\sum_{i=0}^{319} c_i b_i + T = 0$  para o DES; e
2.  $\sum_{i=0}^{319} c_i b_i - T = 0$  para o IDEA.

Daí, usando técnicas da programação linear baseadas em combinações das restrições acima, valores de  $(c_0, c_1, \dots, c_{319})$  e  $T$  foram encontrados. Assim, cada criptograma é dividido em  $m$  segmentos (ou vetores) de 320 bits. São contados os  $n$  segmentos cujo resultado da função fique acima de  $T$  e calculada a taxa  $R$  atinente ao valor de  $n$  e ao total de  $m$  segmentos. A função proposta é construída e testada em três níveis.

A taxa  $R$  é a razão entre o número  $n$  de segmentos com valores maiores que  $T$  e o número total  $m$  de segmentos.

Os experimentos falharam no processo de identificação ficando os valores de  $R$  tanto para o DES como para o IDEA bem próximos um do outro, em quase todos os experimentos para todos os níveis propostos.

Ainda no trabalho de MAHESHWARI (2001), há relatos de sucesso na classificação de cifras clássicas, usando técnicas de distribuição de frequência. Mesmo assim, nem todos os experimentos alcançaram 100% de acerto no processo de identificação.

CHANDRA (2002) estendeu a função limiar proposta por MAHESHWARI (2001) na tentativa de identificar as cifras DES e IDEA, com  $f = \sum_{i=0}^{319} c_i b_i \geq T$ , para o DES, e  $f = \sum_{i=0}^{319} c_i b_i < T$  para o IDEA. A partir da função limiar de nível 1 (um) foram definidos quatro modelos: *good-test* estático, *good-test* dinâmico, *good-test* dinâmico estendido e índice de subarquivo<sup>19</sup>. Os experimentos obtiveram bons resultados.

O *good-test* estático identificou corretamente 74,9% dos criptogramas gerados pelo DES e 77,7% dos criptogramas gerados pelo IDEA, no melhor caso, com criptogramas de 2 a 8 *Kbytes*.

O *good-test* dinâmico identificou corretamente 96% dos criptogramas gerados pelo DES e 93% dos criptogramas gerados pelo IDEA, no melhor caso, com criptogramas de 200 *Kbytes*. Em uma variação deste teste, para trabalhar com arquivos de 40 *Kbytes* foram identificados corretamente 78% dos criptogramas gerados pelo DES e 97% dos criptogramas gerados pelo IDEA, no melhor caso.

O *good-test* dinâmico estendido identificou corretamente 67% dos criptogramas gerados pelo DES e 92% dos criptogramas gerados pelo IDEA, no melhor caso, com criptogramas de 40 *Kbytes*.

O índice de subarquivo identificou corretamente 71% dos criptogramas gerados pelo DES e 99% dos criptogramas gerados pelo IDEA, no melhor caso, com criptogramas de 40 *Kbytes*.

RAO (2003) utilizou a função limiar proposta por MAHESHWARI (2001) na tentativa de identificar as cifras RSA e IDEA, a partir de 1000 criptogramas: 500 gerados pelo RSA e 500 gerados pelo IDEA, com  $f = \sum_{i=0}^{319} c_i b_i \geq T$ , para o RSA, e  $f = \sum_{i=0}^{319} c_i b_i < T$ , para o IDEA; e identificou corretamente 54% dos criptogramas gerados pelo RSA e 58% dos criptogramas

---

<sup>19</sup> Para maiores detalhes sobre o que foi mudado na estrutura das funções, queira consultar CHANDRA (2002).



gerados pelo IDEA, no melhor caso, com criptogramas de 4 *Kbytes*. Uma modificação

$f = \sum_{i=0}^{319} c_i b_i \geq xT$  foi realizada para maximizar a diferença entre os valores relativos ao RSA

e ao IDEA, porém não apresentou um rendimento significativo. Outros dois modelos foram construídos, com base na função limiar: Cifração Repetida e Decifração com Chave Aleatória.

Na Cifração Repetida os 500 criptogramas cifrados pelo RSA são cifrados mais  $n$  vezes pelo RSA e os 500 criptogramas cifrados pelo IDEA são também cifrados mais  $n$  vezes pelo IDEA. Três variações no formato da cifração foram realizadas e a classificação alcançou no melhor caso 65% para o RSA e 70% para o IDEA.

Decifração com Chave Aleatória testa a quantidade de *bits* alterados após a decifração do criptograma por duas chaves diferentes. A taxa de acerto na identificação ficou em 85% tanto para o RSA quanto para o IDEA.

SAXENA (2008) estendeu os trabalhos de MAHESHWARI (2001), CHANDRA (2002) e RAO (2003) na tentativa de identificar os algoritmos Blowfish, Camellia e RC4.

Os experimentos consistiam em, dado um criptograma, identificar se:

- i. O criptograma foi cifrado pelo Blowfish ou pelo RC4; e
- ii. O criptograma foi cifrado pelo Camellia ou pelo RC4.

Com  $f = \sum_{i=0}^{319} c_i b_i > T$ , para o Blowfish ou Camellia, e  $f = \sum_{i=0}^{319} c_i b_i \leq T$ , para o RC4.

O trabalho buscou bons vetores de testes  $(c_0, c_1, \dots, c_{319})$ , os quais eram selecionados a partir de um valor limite, onde diversos desses valores limites foram testados. Estes bons vetores de testes foram usados para classificar os algoritmos com máquinas de vetor de suporte, por meio da resolução de um problema de programação linear. O melhor resultado na classificação entre as cifras Blowfish e RC4 foi de 99,8%, com o valor limite de 0,05 e na classificação entre as cifras Camellia e RC4 foi de 99,7%, com o valor limite de 0,05.

### 3.3 MÁQUINAS DE VETOR DE SUPORTE

No trabalho de DILLEP e SEKHAR (2006) foram usadas máquinas de vetor de suporte (Support Vector Machine – SVM) para identificar as cifras de bloco DES, Triple DES, Blowfish, AES e RC5, usando os modos de operação ECB e CBC. O trabalho usou uma abordagem baseada em classificação, considerando os criptogramas como documentos e

representando os mesmos como vetores, semelhante ao que foi feito nos trabalhos de CARVALHO (2006) e SOUZA (2007). Entretanto, trataram os termos desses documentos como seqüências fixas de 16 *bits* que se repetem ao longo do criptograma ou como seqüências variáveis de *bits* delimitadas pelas três seqüências de 16 *bits* que mais se repetem em 50 criptogramas. Estes termos, então, constituíram um dicionário para apoiar o processo de classificação. Já em CARVALHO (2006) e SOUZA (2007) os termos tinham tamanho igual ao tamanho dos blocos, variando de acordo com a cifra ou com o tamanho das chaves o que permitiu a classificação adequada dos criptogramas.

Em DILLEP e SEKHAR (2006) foram usadas duas abordagens relacionadas à criação dos dicionários. Na primeira, um dicionário comum a todas as cifras foi gerado a partir dos criptogramas produzidos pelas cifras DES, Triple DES, Blowfish, AES e RC5. Na segunda, foi criado um dicionário especializado para cada uma das cifras relacionadas.

Nos experimentos foram utilizadas como núcleo das SVM as funções: linear, sigmoid, polinomial e gaussiana.

Em uma primeira fase, as SVM foram treinadas e depois foi realizado o processo de classificação.

O melhor resultado do trabalho foi obtido quando se utiliza uma SVM para cada cifra, dicionários específicos, tamanho fixo do termo e a função gaussiana como núcleo. Neste caso, o experimento conseguiu identificar com 97,78% de precisão as cifras DES, Triple DES e Blowfish, modo de operação ECB. A mesma chave foi usada para cifrar todos os criptogramas gerados por essas cifras.

Outro bom resultado utilizando uma SVM para cada cifra, dicionários específicos, tamanho fixo do termo e a função gaussiana como núcleo, conseguiu identificar com 95% de precisão as cifras DES, Triple DES, Blowfish, AES e RC5, no modo de operação ECB. Neste caso, foi utilizada uma chave diferente para cada uma das cifras na fase de treino. Na fase de classificação as mesmas chaves foram usadas para cada uma dessas cinco cifras descritas. Entretanto, quando na fase de classificação foram usadas chaves diferentes daquelas utilizadas na fase de treino, a precisão caiu para 41%.

O trabalho relata que um experimento, onde foi utilizada a seguinte configuração: uma SVM para cada cifra, dicionários específicos, tamanho fixo do termo e a função gaussiana como núcleo; foi possível identificar com 87,5% de precisão as cifras DES no modo CBC, e as cifras Triple DES, Blowfish, AES e RC5, no modo ECB. Neste caso, foi utilizada uma chave diferente para cada uma das cifras na fase de treino. Na fase de classificação as

mesmas chaves foram mantidas para cada uma dessas cinco cifras descritas. Entretanto, quando na fase de classificação foram usadas chaves diferentes daquelas utilizadas na fase de treino, a precisão caiu para 37,25%.

Em SOUZA (2007) existe uma tentativa de classificação de criptogramas em função do algoritmo criptográfico usando o mapa de Kohonen (KOHONEN, 2001), que opera de maneira similar à SVM. Esse trabalho obteve resultados que motivam trabalhos futuros com o Mapa de Kohonen na classificação de criptogramas.

### 3.4 RECUPERAÇÃO DE INFORMAÇÃO E LINGÜÍSTICA COMPUTACIONAL

Os trabalhos de CARVALHO (2006) e SOUZA (2007) e SOUZA *et al* (2008) realizaram agrupamento de criptogramas gerados pelas cifras de blocos DES, AES e RSA, em função da chave de cifrar, com diversos tamanhos de criptogramas e com diversas chaves diferentes e aleatórias. Esses trabalhos se baseiam no fato de que uma chave criptográfica determina um conjunto léxico próprio e único. Considerando um contexto lingüístico, uma chave criptográfica dá origem a um idioma desconhecido utilizando um alfabeto binário, onde os termos deste idioma são os blocos dos criptogramas. Assim, os criptogramas podem ser agrupados pelos mesmos processos utilizados para agrupar textos legíveis.

Os métodos utilizados nesses trabalhos resultaram na formação de grupos de criptogramas, os quais podem ser avaliados pelas medidas revocação e precisão (FUNG, 2003), onde revocação indica a capacidade do método de recuperar todos os criptogramas relevantes. Precisão, por sua vez, indica a capacidade do método de recuperar apenas criptogramas relevantes. Criptogramas relevantes são aqueles que possuem características intrínsecas a um determinado grupo. Por exemplo, em um grupo formado por criptogramas cifrados pelo MARS, os criptogramas relevantes são os cifrados pelo MARS. Os demais criptogramas cifrados por outros algoritmos, que eventualmente estejam nesse grupo, não são relevantes. Os trabalhos obtiveram revocação e precisão máxima na maioria dos experimentos.

Ainda em SOUZA (2007) foram exploradas medidas de similaridades, distância, métodos de agrupamento e classificação; na tentativa de se buscar melhores valores nas medidas de revocação e precisão.

Em CARVALHO (2006) e OLIVEIRA (2006), há relatos de 100% de acerto na classificação de cifras clássicas polialfabéticas.

### 3.5 MÉTODOS BASEADOS EM HISTOGRAMA E EM PREDIÇÃO DE BLOCOS

No trabalho de NAGIREDDY (2008) foram desenvolvidos os métodos de histograma e de predição de bloco. Tais métodos utilizam técnicas de expansão de dados e de ataques secundários (*side channel attack*) para identificação das cifras de bloco DES, AES, Blowfish, Triple DES e RC5, nos modos de operação ECB e CBC. Buscou-se também identificar o modo de operação utilizado no algoritmo.

O método de histograma utiliza a frequência de várias seqüências de *bits*, onde cada seqüência diferente representa um símbolo, testando tamanhos de seqüências entre 4 e 16 *bits*. Para cada uma das cifras descritas criam-se histogramas em uma fase inicial do processo. Na classificação, a cada novo criptograma, um novo histograma é gerado e por meio do cálculo da distância Euclidiana entre o novo histograma e os histogramas já criados, rotulados e armazenados durante a fase inicial do processo, a menor distância indica a cifra correspondente ao novo histograma. O processo foi bem sucedido ao identificar as cifras DES, Blowfish, Triple DES e RC5 com símbolos de 16 *bits*, obtendo 100% de precisão com o modo ECB.

O método de histograma também obteve sucesso em determinar o modo de operação ECB baseado apenas nos criptogramas, com 100% de precisão. Ressalta-se que nesse método ora se identifica a cifra ora o modo de operação, ou seja, método não identifica ambos conjuntamente. No modo CBC, obteve 100% de precisão considerando apenas o primeiro bloco do criptograma desde que se mantenha constante o vetor de inicialização, que é parte fundamental do modo CBC.

Os demais métodos de NAGIREDDY (2008), embora apresentem resultados importantes, não são expressivos quando comparados com o método que utiliza histograma.

Considerando o escopo da presente tese, tem-se que o trabalho de NAGIREDDY (2008) apresenta boas sugestões, entretanto, tem a complexidade da geração de múltiplos classificadores (múltiplos histogramas), além de usar tamanhos de mensagens de 10 *Kbytes* o que é considerando grande para a metodologia usada na presente tese.

## 4 METODOLOGIA PARA IDENTIFICAR CIFRAS DE BLOCO

### 4.1 RECUPERAÇÃO DE INFORMAÇÃO NA IDENTIFICAÇÃO DE CIFRAS

A utilização de informações é fundamental para as organizações. Para tais organizações, tem sido um problema tratar o grande volume de informações que são necessárias no dia-a-dia, principalmente as textuais (MANNING, 2003). No caso da internet, pelo menos dois fatores de complexidade foram adicionados ao tratamento da informação: a divulgação em larga escala e a distribuição ao longo da grande rede mundial.

Uma maneira de abordamos estes problemas é por meio do uso dos Sistemas de Recuperação de Informações (sistemas RI). Estes sistemas comparam uma declaração formal de necessidade de informação, chamada consulta, com as informações disponíveis em uma base de dados (FRAKES, 1992). No contexto da Recuperação de Informações (RI) esta base de dados é composta por um conjunto de objetos, que representam documentos. Os documentos são arquivos digitais, os quais são geralmente compostos de informações textuais, mas podem conter também, imagens, gráficos, fórmulas matemáticas e outras formas de representar informações (FRAKES, 1992).

Considerando apenas as informações textuais, as tarefas de RI utilizam uma abordagem baseada na distribuição estatística das palavras pelos documentos (FRAKES, 1992). Pode ser acrescentado às palavras algum tipo de peso, como a frequência com que estas ocorrem no documento ou na base de dados (HARMAN, 1992).

A abordagem estatística da RI é útil na identificação de cifras, como pode ser visto em (CARVALHO, 2006), (SOUZA, 2007) e (SOUZA *et al*, 2008), os quais utilizaram esta abordagem para agrupar criptogramas em função da chave criptográfica.

### 4.2 UNIDADE BÁSICA PARA O TRATAMENTO DE CRIPTOGRAMAS

Um sistema RI, aplicado as informações textuais, considera a palavra como unidade básica para a recuperação de informações. No caso dos criptogramas, não é possível identificar claramente tal unidade básica. Entretanto, a regra de transformação das cifras de blocos indica como determinar a unidade básica para o tratamento dos criptogramas por um sistema RI, como se ver a seguir.

As cifras de bloco transformam mensagens em criptogramas operando sobre blocos de *bits* ou *bytes*. O tamanho dos blocos depende do algoritmo e, muitas vezes, do tamanho da chave utilizada. Por exemplo, o AES utiliza blocos de 128, 192 ou 256 *bits*, dependendo do tamanho da chave<sup>20, 21</sup>. Desta forma, pode-se dizer que o alfabeto de um criptograma é um alfabeto binário e, no caso das cifras de bloco, a unidade básica de recuperação de criptogramas é um bloco.

Durante o seu processamento, um sistema RI busca identificar a quantidade de palavras sem repetição da coleção de documentos e a distribuição dessas palavras pelos documentos, para depois compará-los e verificar o grau de similaridade entre estes documentos, isto é, o quanto os documentos são semelhantes. Assim, quanto mais similares dois documentos são, maior a chance de pertencerem ao mesmo grupo. A justificativa para a escolha de um bloco como unidade básica para a recuperação dos criptogramas é a mesma: quanto mais blocos em comum dois criptogramas possuem, maior a chance de pertencerem ao mesmo grupo, onde cada grupo classifica ou identifica uma cifra.

Com cifras de blocos, no modo ECB, se um mesmo bloco de texto em claro é cifrado duas vezes com a mesma chave, os dois blocos de criptogramas resultantes deste processo serão iguais. Considerando que uma chave qualquer submetida com um texto claro a um algoritmo criptográfico determina um idioma particular e desconhecido, este processo dará origem a criptogramas escritos neste idioma (SOUZA *et al*, 2008). Assim, fixando-se o texto claro e a chave e variando o algoritmo, tem-se que criptogramas produzidos com um mesmo algoritmo serão mais similares uns com os outros, uma vez que compartilharão o alfabeto (binário) e o mesmo idioma (determinado pelo algoritmo), logo devem possuir o mesmo conjunto de elementos léxicos.

#### 4.3 MODELAGEM DE CRIPTOGRAMAS SOBRE UM ESPAÇO VETORIAL

A coleção de criptogramas é modelada sobre um espaço de vetores. Desta forma, para representar os criptogramas são utilizados vetores de *dimensão-n*, onde *n* é o número de blocos distintos em todo o conjunto de criptogramas, onde cada bloco representa um eixo no

---

<sup>20</sup> A descrição do AES só contempla o bloco de 128 *bits*. Porém, o algoritmo Rijndael pode tratar blocos de 128, 192, e 256 *bits* (NIST, 2001).

<sup>21</sup> Neste trabalho, os tamanhos de chaves são iguais aos tamanhos de blocos, respectivamente.

espaço vetorial e, conseqüentemente, um ponto no espaço determinará um criptograma (figura 4.1).

Aplicando a Definição 2.15 especificamente aos criptogramas, tem-se: Sejam os vetores  $\vec{c}_i = (c_{1,i}, c_{2,i}, \dots, c_{n,i})$  e  $\vec{c}_j = (c_{1,j}, c_{2,j}, \dots, c_{n,j})$ , representação vetorial de dois criptogramas em  $\mathfrak{R}^n$ , para os quais se deseja obter a similaridade. O valor relacionado à  $c_{l,i}$  é a freqüência do  $l$ -ésimo termo (ou palavra) no documento  $c_i$ . O valor relacionado à  $c_{l,j}$  é a freqüência do  $l$ -ésimo termo no documento  $c_j$ . A similaridade (ou grau de associação) entre  $c_i$  e  $c_j$  pode ser obtida aplicando a  $\vec{c}_i$  e  $\vec{c}_j$  uma função que mede a similaridade entre dois vetores, como, por exemplo,  $s_{medida}(\vec{c}_i, \vec{c}_j)$ .

Na figura 4.1 pode-se notar que os criptogramas 1, 2 e 3 são pontos no espaço, representados pelas coordenadas (X, Y, Z), (X, 0, Z) e (0, Y, Z), respectivamente.

No exemplo, a quantidade de blocos sem repetição é igual a três (010101000101, 110101100111, 011111000110), ou seja,  $n = 3$  e, assim,  $\mathfrak{R}^3$ . Temos então um espaço vetorial de três dimensões e os seguintes vetores:

Criptograma 1 (010101000101, 110101100111, 011111000110) = vetor 1 (1, 1, 1).

Criptograma 2 (010101000101, 0, 011111000110) = vetor 2 (1, 0, 1).

Criptograma 3 (0, 110101100111, 011111000110) = vetor 3 (0, 1, 1).

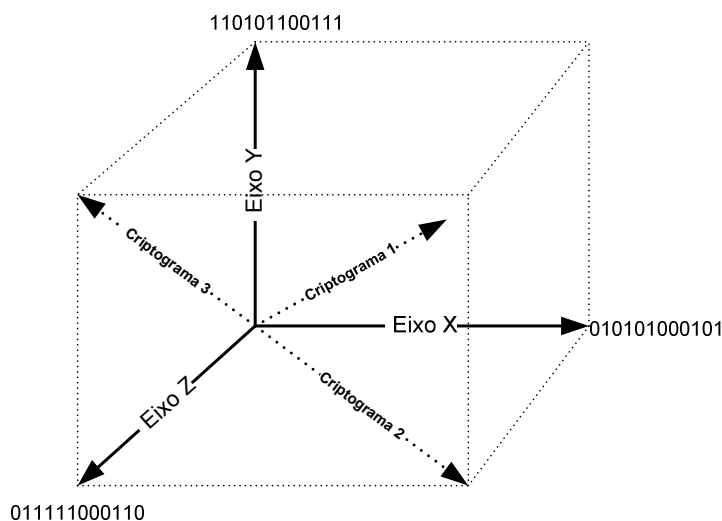


Figura 4.1 – Modelo de Espaço de Vetores de Criptogramas

#### 4.4 MEDIDA DE SIMILARIDADE ENTRE CRIPTOGRAMAS

A similaridade entre dois criptogramas  $c_i$  e  $c_j$  foi associada ao co-seno do ângulo,  $s_{co-seno}(\vec{c}_i, \vec{c}_j)$ , e calculada pela fórmula 4.1, tal que  $0 \leq s \leq 1$ . Quanto maior o valor de  $s$ , maior a similaridade entre os criptogramas. O co-seno do ângulo é amplamente utilizado em tarefas de Recuperação de Informações (HARMAN, 1992). Então, cria-se uma matriz de similaridades, armazenando, em suas células, os valores de similaridade dos pares de criptogramas da coleção.

$$s_{co-seno}(\vec{c}_i, \vec{c}_j) = \frac{\sum_{k=1}^n (c_{i,k} \times c_{j,k})}{\sqrt{\sum_{k=1}^n (c_{i,k})^2 \times \sum_{k=1}^n (c_{j,k})^2}} \quad \text{Fórmula 4.1}$$

#### 4.5 AGRUPAMENTO DE CRIPTOGRAMAS

O método para agrupar criptogramas neste trabalho, utiliza a técnica hierárquica aglomerativa da ligação simples (RASMUSSEN, 1992) formando grupos dispersos (JAIN, 1999), o que permite que dois criptogramas quaisquer que estejam em um grupo, possuam valor de similaridade mais baixo que a similaridade do próprio grupo. Assim, com a conclusão do procedimento,  $n$  criptogramas são categorizados em  $m$  grupos. O valor de  $m$  é desconhecido no início do procedimento.

Inicialmente, cada criptograma pertencerá a um único grupo. A seguir, identifica-se, na matriz de similaridades, o par de criptogramas com o maior valor de similaridade para formar o primeiro grupo. Atribui-se ao grupo um valor de “similaridade de grupo” igual ao maior valor de similaridade existente entre os pares de criptogramas pertencentes ao grupo. A matriz de similaridade é atualizada pela substituição da similaridade do par pela similaridade do grupo. Esse procedimento é repetido até que um critério de parada seja alcançado.

A estrutura final do agrupamento, representando a ordem em que as inclusões e junções dos grupos ocorrem é representada por um dendrograma (RASMUSSEN, 1992) (figura 4.2). Observando essa figura, pode-se notar que os grupos foram formados a partir de um determinado valor de similaridade, o qual pode ser utilizado como o critério de parada citado anteriormente.



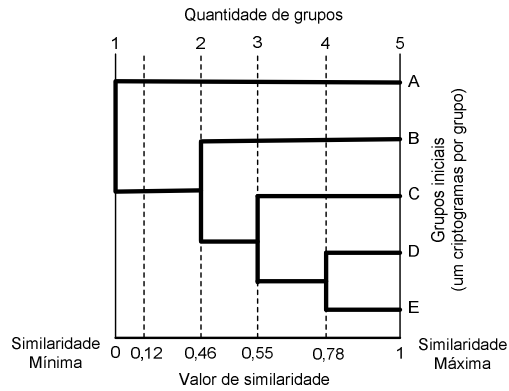


Figura 4.2: Dendrograma

Neste trabalho, utiliza-se um valor de similaridade próximo de zero como critério de parada, pois é pouco provável a repetição de blocos ao longo dos criptogramas

#### 4.6 AVALIAÇÃO DO AGRUPAMENTO DE CRIPTOGRAMAS

Na avaliação da qualidade do agrupamento de criptogramas foram utilizadas as medidas revocação e precisão (YATES, 1999) e (FUNG, 2003) (Figura 4.3).

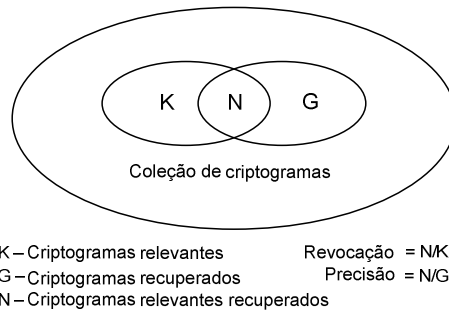


Figura 4.3: Revocação e Precisão

Os valores de revocação e precisão são definidos como segue. Suponha que  $K$  seja o conjunto formado pelos criptogramas cifrados com um determinado algoritmo  $\Delta$ . Seja  $G$  o agrupamento construído pelo método proposto e que supostamente contém os criptogramas gerados pelo algoritmo  $\Delta$ . Seja  $|k|$  o número de elementos no conjunto  $K$ ,  $|g|$  o número de elementos de  $G$  e  $n$  o número de elementos do conjunto  $K$  presentes no grupo  $G$ . Então, os

valores de Revocação e Precisão são obtidos pelas Fórmulas 4.2 e 4.3, respectivamente, tal que  $0 \leq R \leq 1$  e  $0 \leq P \leq 1$ .

Revocação, portanto, indica a capacidade do método de recuperar todos os criptogramas relevantes. Precisão, por sua vez, indica a capacidade do método de recuperar apenas criptogramas relevantes.

$$R = \frac{n}{|k|} \quad \text{Fórmula 4.2}$$

$$P = \frac{n}{|g|} \quad \text{Fórmula 4.3}$$

Deve-se notar que as fórmulas acima determinam a revocação e a precisão para um grupo apenas. Para consolidar os valores de  $m$  grupos pode-se utilizar a macro-média (YANG, 1999), onde a revocação e a precisão são calculadas localmente, para cada um dos  $m$  grupos, sendo obtida a média ao final (fórmulas 4.4 e 4.5).

$$R = \frac{1}{m} \sum_{i=1}^m \frac{n_i}{|k_i|} \quad \text{Fórmula 4.4}$$

$$P = \frac{1}{m} \sum_{i=1}^m \frac{n_i}{|g_i|} \quad \text{Fórmula 4.5}$$

#### 4.7 DESCRIÇÃO DA METODOLOGIA PARA 5 – CIFRAS DE BLOCO

A metodologia proposta para identificar as cifras considera os blocos dos criptogramas como palavras de um texto de idioma desconhecido (CARVALHO, 2006) (SOUZA, 2007) e agrupa os criptogramas com base na similaridade existente entre eles, similaridade essa medida com base na frequência com que os blocos ocorrem em cada um dos criptogramas. O procedimento é não supervisionado, ou seja, os processos são executados sem conhecimento sobre: os textos claros, os algoritmos criptográficos, a chave utilizada no processo de cifração ou a quantidade de grupos que serão formados.

Desta forma, uma coleção de textos claros é cifrada por cada um dos algoritmos criptográficos propostos, gerando uma coleção de criptogramas composta por  $n$  elementos, onde  $n = \text{quantidade de textos claros} \times \text{quantidade algoritmos criptográficos utilizados}$ .

A coleção de criptogramas é representada em um espaço de vetores de maneira que seja possível determinar a similaridade entre cada par de criptogramas por meio do cálculo de similaridades com a medida do co-seno do ângulo  $\theta$ , e, assim, construir uma matriz de similaridades. A partir dessa matriz é realizado o agrupamento pelo método da ligação simples. Na fase de análise de grupos pode-se escolher um valor de similaridades no intervalo  $[0,1]$  para se determinar a quantidade de grupos que serão formados. Por fim, a qualidade do agrupamento é avaliada pelas medidas revocação e precisão (Figura 4.4).

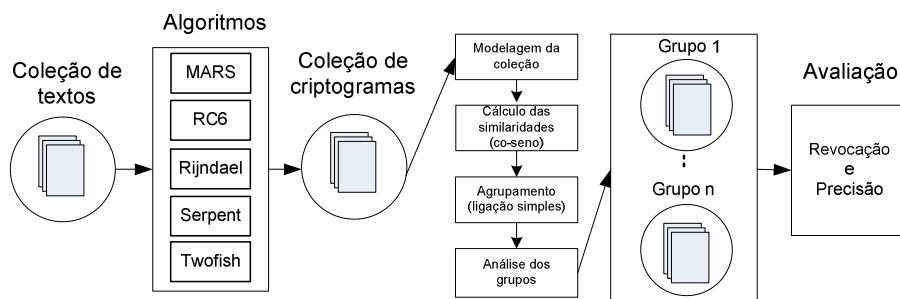


Figura 4.4: Esquema do mecanismo com o caso particular de  $n = 5$  cifras

A seguir, a pertinência de cada novo criptograma a um dos grupos anteriormente formados (classificação) indica o algoritmo que cifrou o mesmo ou indica que uma nova cifra, diferente das anteriormente classificadas, foi detectada (figura 4.5). Tal procedimento pode ser considerado, por analogia, como um ataque de distinção.

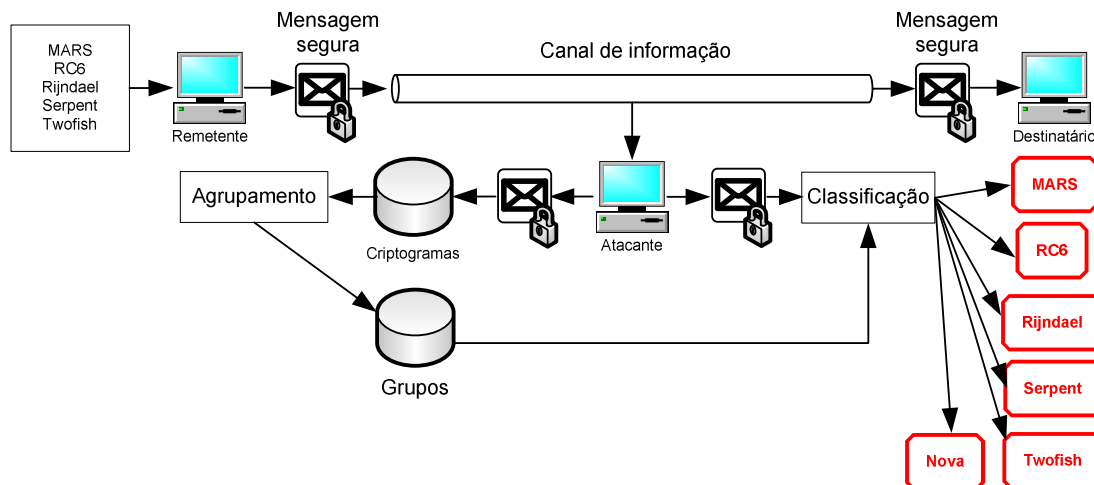


Figura 4.5: Modelo para Identificação de Cifra

Na figura 4.5, nota-se que o canal de informação é vulnerável a ataques passivos, como análise de tráfego, mesmo quando protegido por criptografia. Realizando a análise de tráfego, um atacante poderia obter informações sobre alguns padrões de tráfego relativos às mensagens, como frequência e tamanho da mensagem. Com a metodologia proposta, pode-se obter informação mais útil como a identificação do algoritmo criptográfico ou a detecção de mudança do algoritmo utilizado na cifração.

A metodologia pode ser empregada também para detectar a mudança na chave criptográfica utilizada para cifrar a mensagem que trafega na rede.

## 5 EXPERIMENTOS, RESULTADOS E AVALIAÇÕES DE 5-CIFRAS

### 5.1 INTRODUÇÃO AOS EXPERIMENTOS COMPUTACIONAIS

Os experimentos têm a finalidade de contribuir para a realização dos objetivos propostos na presente tese.

De maneira geral, este trabalho indica que se variarmos um parâmetro criptográfico e fixamos os demais será possível identificar o parâmetro variante. Assim, os experimentos propostos visam explorar estas características para identificar tais parâmetros e fazer uma proposição geral para a identificação de criptogramas.

A ferramenta WARS Text, desenvolvida em Souza (2007 e 2008), viabiliza os métodos propostos para a metodologia, os quais foram descritos no capítulo 4.

Todos os experimentos foram realizados com blocos de 128 *bits*.

As chaves criptográficas utilizadas nos experimentos foram criadas a partir de gerador definido em CARVALHO *et al* (1999) e estão detalhadas no apêndice 9.1.

### 5.2 BASES DE CRIPTOGRAMAS

Neste trabalho são utilizados 30 textos claros para cada um de nove tamanhos: 1024, 1536, 2048, 2560, 3072, 4096, 6144, 8192 e 10240 *bytes*; totalizando 270 textos claros. Os textos claros foram obtidos em (BIBLE, 2005) e foram cifrados pelos cinco algoritmos finalistas do AES: MARS, RC6, Rijndael, Serpent e Twofish, utilizando-se uma, cinco ou 30 chaves de 128 *bits*.

Na seção 5.4.7, os experimentos com idiomas foram obtidos em (BIBLE, 2009) e os criptogramas foram obtidos em SOUZA (2007). Tais criptogramas foram criados a partir de textos claros obtidos em BIBLE (2005) e foram cifrados pelos algoritmos: AES e DES, utilizando-se duas chaves de 128 *bits* e duas chaves de 64 *bits*, respectivamente. As chaves utilizadas não estão detalhadas, uma vez que não era o escopo do trabalho de SOUZA (2007).

### 5.3 DESCRIÇÃO DAS BASES DE CRIPTOGRAMAS

#### 5.3.1 BASE 1 DE CRIPTOGRAMAS

Esta base possui para cada modo de operação: ECB e CBC; nove coleções, cada uma com determinado tamanho. Cada coleção é composta por 150 criptogramas, gerados pelos cinco algoritmos: MARS, RC6, Rijndael, Serpent e Twofish, a partir dos mesmos 30 textos em claro e uma única chave de 128 *bits* para cada modo (tabela 5.1).

Tabela 5.1. Configuração da base 1 de criptogramas

Modo	Algoritmos	Textos claros	Chaves	Número de Coleções	Criptogramas por coleção	Total de criptogramas
ECB	5	30	1	9	150	1.350
CBC	5	30	1	9	150	1.350
Total de criptogramas na base 1						2.700

#### 5.3.2 BASE 2 DE CRIPTOGRAMAS

Para cada algoritmo: MARS, RC6, Rijndael, Serpent e Twofish; são definidas nove coleções, cada uma com determinado tamanho. Cada coleção é composta por 60 criptogramas, gerados nos modos de operação: ECB e CBC; sendo 30 criptogramas por modo, a partir dos mesmos 30 textos em claro e uma única chave aleatória  $k$  de 128 *bits* (tabela 5.2).

Tabela 5.2. Configuração da base 2 de criptogramas

Algoritmos	Modo	Textos claros	Chaves	Número de Coleções	Criptogramas por coleção	Total de criptogramas
MARS	2	30	1	9	60	540
RC6	2	30	1	9	60	540
Rijndael	2	30	1	9	60	540
Serpent	2	30	1	9	60	540
Twofish	2	30	1	9	60	540
Total de criptogramas na base 2						2.700

#### 5.3.3 BASE 3 DE CRIPTOGRAMAS

São definidas nove coleções, cada uma com determinado tamanho. Cada coleção com 4500 criptogramas, onde cada uma das cifras: MARS, RC6, Rijndael, Serpent e Twofish;

cifrou 30 criptogramas com cada uma de trinta chaves diferentes e aleatórias de 128 *bits*, de  $k_1$  a  $k_{30}$ , no modo de operação ECB, totalizando 900 criptogramas cifrados por cada cifra em uma coleção (tabela 5.3).

Tabela 5.3. Configuração da base 3 de criptogramas

Algoritmos	Modo	Textos claros	Chaves	Número de Coleções	Criptogramas por coleção	Total de criptogramas
5	1	30	30	9	4.500	40.500
Total de criptogramas na base 3						40.500

#### 5.3.4 BASE 4 DE CRIPTOGRAMAS

São definidas nove coleções, cada uma com determinado tamanho. Cada coleção com 13.500 criptogramas, onde cada uma das cifras: MARS, RC6, Rijndael, Serpent e Twofish; cifrou 30 criptogramas com cada uma de trinta chaves diferentes e aleatórias de 128 *bits*, de  $k_1$  a  $k_{30}$ , para cada um dos modos de operação: ECB e CBC, totalizando 1.800 criptogramas cifrados por cifra em uma coleção (tabela 5.4).

Tabela 5.4. Configuração da base 4 de criptogramas

Algoritmos	Modo	Textos claros	Chaves	Número de coleções	Criptogramas por coleção	Total de criptogramas
5	2	30	30	9	9.000	81.000
Total de criptogramas na base 4						81.000

#### 5.3.5 BASE 5 DE CRIPTOGRAMAS

São definidas nove coleções para uma, duas, três e quatro rodadas dos algoritmos, cada coleção com determinado tamanho. Cada coleção é composta por 150 criptogramas, gerados pelos algoritmos: MARS, RC6, Rijndael, Serpent e Twofish, a partir dos mesmos 30 textos claros e uma única chave aleatória  $k$  de 128 *bits* no modo de operação ECB (tabela 5.5).

Tabela 5.5. Configuração da base 5 de criptogramas

Algoritmos	Rodadas	Textos claros	Chaves	Número de coleções	Criptogramas por coleção	Total de criptogramas
5	1	30	1	9	150	1.350
5	2	30	1	9	150	1.350
5	3	30	1	9	150	1.350
5	4	30	1	9	150	1.350
Total de criptogramas na base 5						5.400

### 5.3.6 BASE 6 DE CRIPTOGRAMAS

São definidas nove coleções, cada uma com determinado tamanho. Cada coleção com 150 criptogramas, onde cada uma das cifras: MARS, RC6, Rijndael, Serpent e Twofish; cifrou por cinco vezes consecutivas 30 criptogramas com uma única chave aleatória  $k$  de 128 *bits*, no modo de operação ECB (tabela 5.6), de acordo com as composições abaixo:

$$\begin{aligned}
 &MARS_k^{ECB} (MARS_k^{ECB} (MARS_k^{ECB} (MARS_k^{ECB} (MARS_k^{ECB} (texto_1, \dots, texto_{30}))))); \\
 &RC6_k^{ECB} (RC6_k^{ECB} (RC6_k^{ECB} (RC6_k^{ECB} (RC6_k^{ECB} (texto_1, \dots, texto_{30}))))); \\
 &Rijndael_k^{ECB} (Rijndael_k^{ECB} (Rijndael_k^{ECB} (Rijndael_k^{ECB} (Rijndael_k^{ECB} (texto_1, \dots, texto_{30}))))); \\
 &Serpent_k^{ECB} (Serpent_k^{ECB} (Serpent_k^{ECB} (Serpent_k^{ECB} (Serpent_k^{ECB} (texto_1, \dots, texto_{30}))))); \\
 &Twofish_k^{ECB} (Twofish_k^{ECB} (Twofish_k^{ECB} (Twofish_k^{ECB} (Twofish_k^{ECB} (Twofish_k^{ECB} (texto_1, \dots, texto_{30}))))).
 \end{aligned}$$

Tabela 5.6. Configuração da base 6 de criptogramas

Algoritmos	Composições	Textos claros	Chaves	Número de coleções	Criptogramas por coleção	Total de criptogramas
5	5	30	1	9	150	1.350
Total de criptogramas na base 6						1.350

### 5.3.7 BASE 7 DE CRIPTOGRAMAS

São definidas nove coleções, cada uma com determinado tamanho. Cada coleção com 150 criptogramas, onde cada uma das cifras: MARS, RC6, Rijndael, Serpent e Twofish; cifrou por cinco vezes consecutivas 30 criptogramas com cinco chaves diferentes e aleatórias de 128 *bits*, de  $k_1$  a  $k_5$ , no modo de operação ECB (tabela 5.7), de acordo com as composições abaixo:

$$\begin{aligned}
 &MARS_{k_1}^{ECB} (MARS_{k_2}^{ECB} (MARS_{k_3}^{ECB} (MARS_{k_4}^{ECB} (MARS_{k_5}^{ECB} (texto_1, \dots, texto_{30}))))); \\
 &RC6_{k_1}^{ECB} (RC6_{k_2}^{ECB} (RC6_{k_3}^{ECB} (RC6_{k_4}^{ECB} (RC6_{k_5}^{ECB} (texto_1, \dots, texto_{30}))))); \\
 &Rijndael_{k_1}^{ECB} (Rijndael_{k_2}^{ECB} (Rijndael_{k_3}^{ECB} (Rijndael_{k_4}^{ECB} (Rijndael_{k_5}^{ECB} (texto_1, \dots, texto_{30}))))); \\
 &Serpent_{k_1}^{ECB} (Serpent_{k_2}^{ECB} (Serpent_{k_3}^{ECB} (Serpent_{k_4}^{ECB} (Serpent_{k_5}^{ECB} (texto_1, \dots, texto_{30}))))); \\
 &Twofish_{k_1}^{ECB} (Twofish_{k_2}^{ECB} (Twofish_{k_3}^{ECB} (Twofish_{k_4}^{ECB} (Twofish_{k_5}^{ECB} (texto_1, \dots, texto_{30}))))).
 \end{aligned}$$



Tabela 5.7. Configuração da base 7 de criptogramas

Algoritmos	Composições	Textos claros	Chaves	Número de coleções	Criptogramas por coleção	Total de criptogramas
5	5	30	5	9	150	1.350
Total de criptogramas na base 7						1.350

### 5.3.8 BASE 8 DE CRIPTOGRAMAS

São definidas nove coleções, cada uma com determinado tamanho. Cada coleção com 150 criptogramas, onde cada uma das cifras: MARS, RC6, Rijndael, Serpent e Twofish; cifrou com uma única chave aleatória  $k$  de 128 bits, no modo de operação ECB, os mesmos 30 textos em claro (tabela 5.8), de acordo com as composições abaixo:

$$\begin{aligned}
 &MARS_k^{ECB} (RC6_k^{ECB} (Rijndael_k^{ECB} (Serpent_k^{ECB} (Twofish_k^{ECB} (texto_1, \dots, texto_{30})))))); \\
 &RC6_k^{ECB} (Rijndael_k^{ECB} (Serpent_k^{ECB} (Twofish_k^{ECB} (MARS_k^{ECB} (texto_1, \dots, texto_{30})))))); \\
 &Rijndael_k^{ECB} (Serpent_k^{ECB} (Twofish_k^{ECB} (MARS_k^{ECB} (RC6_k^{ECB} (texto_1, \dots, texto_{30})))))); \\
 &Serpent_k^{ECB} (Twofish_k^{ECB} (MARS_k^{ECB} (RC6_k^{ECB} (Rijndael_k^{ECB} (texto_1, \dots, texto_{30})))))); \\
 &Twofish_k^{ECB} (MARS_k^{ECB} (RC6_k^{ECB} (Rijndael_k^{ECB} (Serpent_k^{ECB} (texto_1, \dots, texto_{30}))))).
 \end{aligned}$$

Tabela 5.8. Configuração da base 8 de criptogramas

Algoritmos	Composições	Textos claros	Chaves	Número de coleções	Criptogramas por coleção	Total de criptogramas
5	5	30	1	9	150	1.350
Total de criptogramas na base 8						1.350

### 5.3.9 BASE 9 DE CRIPTOGRAMAS

São definidas nove coleções, cada uma com determinado tamanho. Cada coleção com 150 criptogramas, onde cada uma das cifras: MARS, RC6, Rijndael, Serpent e Twofish; cifrou com uma única chave aleatória  $k$  de 128 bits, os mesmos 30 textos em claro, onde as

quatro primeiras cifras da composição usam o modo CBC e a última usa o modo ECB (tabela 5.9), de acordo com as composições abaixo:

$$\begin{aligned}
 &MARS_k^{ECB} (RC6_k^{CBC} (Rijndael_k^{CBC} (Serpent_k^{CBC} (Twofish_k^{CBC} (texto_1, \dots, texto_{30})))))); \\
 &RC6_k^{ECB} (Rijndael_k^{CBC} (Serpent_k^{CBC} (Twofish_k^{CBC} (MARS_k^{CBC} (texto_1, \dots, texto_{30})))))); \\
 &Rijndael_k^{ECB} (Serpent_k^{CBC} (Twofish_k^{CBC} (MARS_k^{CBC} (RC6_k^{CBC} (texto_1, \dots, texto_{30})))))); \\
 &Serpent_k^{ECB} (Twofish_k^{CBC} (MARS_k^{CBC} (RC6_k^{CBC} (Rijndael_k^{CBC} (texto_1, \dots, texto_{30})))))); \\
 &Twofish_k^{ECB} (MARS_k^{CBC} (RC6_k^{CBC} (Rijndael_k^{CBC} (Serpent_k^{CBC} (texto_1, \dots, texto_{30}))))).
 \end{aligned}$$

Tabela 5.9. Configuração da base 9 de criptogramas

Algoritmos	Composições	Textos claros	Chaves	Número de coleções	Criptogramas por coleção	Total de criptogramas
5	5	30	1	9	150	1.350
Total de criptogramas na base 9						1.350

### 5.3.10 BASE 10 DE CRIPTOGRAMAS

São definidas nove coleções, cada uma com determinado tamanho. Cada coleção com 150 criptogramas, onde cada uma das cifras: MARS, RC6, Rijndael, Serpent e Twofish; cifrou com uma única chave aleatória  $k$  de 128 bits, os mesmos 30 textos em claro, onde as quatro primeiras cifras da composição usam o modo ECB e a última usa o modo CBC (tabela 5.10), de acordo com as composições abaixo:

$$\begin{aligned}
 &MARS_k^{CBC} (RC6_k^{ECB} (Rijndael_k^{ECB} (Serpent_k^{ECB} (Twofish_k^{ECB} (texto_1, \dots, texto_{30})))))); \\
 &RC6_k^{CBC} (Rijndael_k^{ECB} (Serpent_k^{ECB} (Twofish_k^{ECB} (MARS_k^{ECB} (texto_1, \dots, texto_{30})))))); \\
 &Rijndael_k^{CBC} (Serpent_k^{ECB} (Twofish_k^{ECB} (MARS_k^{ECB} (RC6_k^{ECB} (texto_1, \dots, texto_{30})))))); \\
 &Serpent_k^{CBC} (Twofish_k^{ECB} (MARS_k^{ECB} (RC6_k^{ECB} (Rijndael_k^{ECB} (texto_1, \dots, texto_{30})))))); \\
 &Twofish_k^{CBC} (MARS_k^{ECB} (RC6_k^{ECB} (Rijndael_k^{ECB} (Serpent_k^{ECB} (texto_1, \dots, texto_{30}))))).
 \end{aligned}$$

Tabela 5.10. Configuração da base 10 de criptogramas

Algoritmos	Composições	Textos claros	Chaves	Número de coleções	Criptogramas por coleção	Total de criptogramas
5	5	30	1	9	150	1.350
Total de criptogramas na base 10						1.350

### 5.3.11 BASE 11 DE CRIPTOGRAMAS

É definida uma coleção com 240 textos claros, com 30 textos de aproximadamente 6144 *bytes* para cada um dos idiomas a seguir: alemão, dinamarquês, holandês, espanhol, francês, grego, hebreu e português (tabela 5.11).

Tabela 5.11. Configuração da base 11

Textos claros	Idiomas	Número de coleções	Textos claros por coleção	Total de criptogramas
30	8	1	240	240
Total de textos na base 11				240

### 5.3.12 BASE 12 DE CRIPTOGRAMAS

É definida uma coleção com criptogramas de 6144 bytes, com 120 criptogramas, onde o AES cifrou com duas chaves diferentes e aleatórias de 128 *bits*, os mesmos 30 textos claros, totalizando 60 criptogramas; e o DES cifrou com duas chaves diferentes e aleatórias de 64 *bits*, os mesmos 30 textos em claro, totalizando 60 criptogramas. Todas as cifrações foram realizadas no modo ECB (tabela 5.12).

Tabela 5.12. Configuração da base 12 de criptogramas

Algoritmos	Modo	Textos claros	Chaves	Número de Coleções	Criptogramas por cifra	Total de criptogramas
AES	1	30	2	1	60	60
DES	1	30	2	1	60	60
Total de criptogramas na base 12						120

### 5.3.13 BASE 13 DE CRIPTOGRAMAS

Esta base é composta pelo conteúdo da base 11 (tabela 5.11) e pelo conteúdo da base 12 (tabela 5.12), com 120 criptogramas e 240 textos claros, totalizando 360 objetos.

### 5.3.14 BASE 14 DE CRIPTOGRAMAS

Esta base é composta pelo conteúdo da base 12 (tabela 5.12) e por 90 textos claros, com 30 textos de aproximadamente 6144 *bytes* para cada um dos idiomas a seguir: grego, hebreu e português, com 120 criptogramas e 90 textos claros, totalizando 210 objetos.

## 5.4 EXPERIMENTO 1: IDENTIFICAÇÃO DE 5-CIFRAS

O experimento é dividido em duas fases:

1. Fase 1: separar os criptogramas em cinco grupos de maneira que cada grupo contenha criptogramas gerados pela mesma cifra e somente por ela;
2. Fase 2: Classificar cada criptograma por cifra de bloco, de acordo com os grupos criados na fase 1. O resultado das duas fases identifica a cifra de bloco.

Este experimento utiliza a base 1 de criptogramas. Para cada modo de operação: ECB e CBC; são definidas nove coleções, cada uma com determinado tamanho. Cada coleção é composta por 150 criptogramas, gerados por cinco algoritmos: MARS, RC6, Rijndael, Serpent e Twofish, a partir dos mesmos 30 textos em claro e uma única chave aleatória  $k$  de 128 *bits* para cada modo, portanto, o experimento utiliza 1350 criptogramas (150 criptogramas x 9 coleções = 1350). Os primeiros 70% dos criptogramas de cada coleção foram usados na fase 1 e os 30% restantes foram usados na fase 2.

### 5.4.1 SEPARAÇÃO DOS GRUPOS

Na tabela 5.13, observa-se que o procedimento separou corretamente os criptogramas com tamanho a partir de 1024 *bytes*, de maneira que cada grupo continha criptogramas gerados pelo mesmo algoritmo e somente por ele, obtendo valor máximo de precisão nos dois modos de operação. A discriminação é perfeita e mesmo no modo CBC, o procedimento não mistura no mesmo grupo, criptogramas oriundos de algoritmos diferentes.

Os resultados com a métrica revocação não são tão bons. O modo ECB, para alguns tamanhos de criptogramas, não alcançou o valor máximo, o que é de se esperar visto que textos com menos de 4096 *bytes* ainda geram dicionário muito pequeno. No modo CBC, os valores de revocação mostram que o procedimento terminou com a quantidade grupos inicial, onde cada grupo continha apenas um criptograma, indicando que não foram achadas

similaridades entre os criptogramas, o que confirma o melhor nível de aleatoriedade dado pelo modo CBC.

Tabela 5.13. Resultado da separação em grupos

Tamanho dos criptogramas	ECB		CBC	
	P	R	P	R
1024	1	0,17	1	0,03
1536	1	0,20	1	0,03
2048	1	0,33	1	0,03
2560	1	0,50	1	0,03
3072	1	0,87	1	0,03
4096	1	0,97	1	0,03
6144	1	1	1	0,03
8192	1	1	1	0,03
10240	1	1	1	0,03

#### 5.4.2 CLASSIFICAÇÃO DAS CIFRAS

Nesta fase o experimento utiliza, para cada modo, 405 criptogramas (30% da coleção), submetidos um a um ao processo de classificação, com a finalidade de alocar cada criptograma a um dos grupos anteriormente formados e, assim, identificar a cifra.

Observa-se na tabela 5.14 que no modo ECB o experimento obteve sucesso identificando corretamente os algoritmos geradores de 303 (75%) criptogramas submetidos ao processo. Nota-se que para os experimentos cujo total de acertos foi menor do que 100%, um novo grupo foi criado, o que garante a precisão máxima.

Tabela 5.14. Resultado da identificação das cifras de bloco

Tamanho dos criptogramas (em bytes)	ECB		CBC	
	Acertos	Novo grupo	Acertos	Novo grupo
1024	22 %	Sim	0 %	Sim
1536	24 %	Sim	0 %	Sim
2048	69 %	Sim	0 %	Sim
2560	91 %	Sim	0 %	Sim
3072	89 %	Sim	0 %	Sim
4096	78 %	Sim	0 %	Sim
6144	100 %	Não	0 %	Sim
8192	100 %	Não	0 %	Sim
10240	100 %	Não	0 %	Sim

Já no modo CBC, o experimento não foi capaz de classificar os criptogramas, dado que não se pôde obter similaridade entre os mesmos. Contudo, foram criados novos grupos, o que indica que criptogramas cifrados por algoritmos diferentes não se misturam em um mesmo grupo, garantindo a máxima precisão.

## 5.5 EXPERIMENTO 2: IDENTIFICAÇÃO DO MODO DE OPERAÇÃO DE 5-CIFRAS

O experimento é dividido em duas fases:

1. Fase 1: separar os criptogramas em dois grupos de maneira que cada grupo contenha criptogramas gerados no mesmo modo de operação e somente eles;
2. Fase 2: Classificar cada criptograma por modo de operação, de acordo com os grupos criados na fase 1. O resultado das duas fases identifica o modo de operação.

Este experimento utiliza a base 2 de criptogramas. Para cada algoritmo: MARS, RC6, Rijndael, Serpent e Twofish; são definidas nove coleções, cada uma com determinado tamanho. Cada coleção é composta por 60 criptogramas, gerados nos modos de operação: ECB e CBC; sendo 30 criptogramas por modo, a partir dos mesmos 30 textos em claro e uma única chave aleatória  $k$  de 128 *bits*. Os primeiros 70% dos criptogramas de cada coleção foram usados na fase 1 e os 30% restantes foram usados na fase 2.

### 5.5.1 SEPARAÇÃO DOS GRUPOS

Nas tabelas 5.15 e 5.16, observa-se que o procedimento separou corretamente os criptogramas com tamanho a partir de 1024 *bytes*, de maneira que cada grupo continha criptogramas gerados no mesmo modo de operação e somente por ele, obtendo valor máximo de precisão em todos os modos de operação. O procedimento não mistura no mesmo grupo, criptogramas oriundos de modos de operação diferentes.

Os resultados com a métrica revocação tiveram valores pequenos. Individualmente, o modo ECB teve mais de um criptograma agrupado. Nos grupos CBC, o procedimento terminou com a quantidade grupos inicial, onde cada grupo continha apenas um criptograma, indicando que não foram achadas similaridades entre os criptogramas, o que diminui o valor de revocação.

Tabela 5.15. Resultado da separação em grupos nos modos: ECB e CBC

Tamanho dos criptogramas	MARS		RC6		Rijndael		Serpent		Twofish	
	P	R	P	R	P	R	P	R	P	R
1024	1	0,10	1	0,10	1	0,10	1	0,10	1	0,10
1536	1	0,12	1	0,12	1	0,12	1	0,12	1	0,12
2048	1	0,18	1	0,18	1	0,18	1	0,18	1	0,18
2560	1	0,27	1	0,27	1	0,27	1	0,27	1	0,27
3072	1	0,45	1	0,45	1	0,45	1	0,45	1	0,45
4096	1	0,50	1	0,50	1	0,50	1	0,50	1	0,50
6144	1	0,52	1	0,52	1	0,52	1	0,52	1	0,52
8192	1	0,52	1	0,52	1	0,52	1	0,52	1	0,52
10240	1	0,52	1	0,52	1	0,52	1	0,52	1	0,52

Observando a tabela 5.16, onde se considera apenas os grupos ECB, observa-se que os valores de revocação e precisão estão idênticos aos do resultado relatados na tabela 5.13 do experimento 1, o que indica consistência dos resultados.

Tabela 5.16. Resultado da Separação em Grupos no modo ECB

Tamanho dos criptogramas	MARS		RC6		Rijndael		Serpent		Twofish	
	P	R	P	R	P	R	P	R	P	R
1024	1	0,17	1	0,17	1	0,17	1	0,17	1	0,17
1536	1	0,20	1	0,20	1	0,20	1	0,20	1	0,20
2048	1	0,33	1	0,33	1	0,33	1	0,33	1	0,33
2560	1	0,50	1	0,50	1	0,50	1	0,50	1	0,50
3072	1	0,87	1	0,87	1	0,87	1	0,87	1	0,87
4096	1	0,97	1	0,97	1	0,97	1	0,97	1	0,97
6144	1	1	1	1	1	1	1	1	1	1
8192	1	1	1	1	1	1	1	1	1	1
10240	1	1	1	1	1	1	1	1	1	1

A partir dos resultados, percebe-se que as transformações realizadas pelos algoritmos (parâmetros fixos) não influenciam nos resultados para os modos de operação (parâmetros variáveis), tendo os valores de revocação e precisão se mantido idênticos para todos os algoritmos.

### 5.5.2 CLASSIFICAÇÃO DOS MODOS DE OPERAÇÃO: ECB E CBC

Nesta fase o experimento utiliza, para cada modo, 162 criptogramas (30% da coleção), submetidos um a um ao processo de classificação, com a finalidade de alocar cada criptograma a um dos grupos anteriormente formados e, assim, identificar a cifra.

Observa-se na tabela 5.17 que os resultados foram iguais para todos os algoritmos, indicando que o algoritmo utilizado não influenciou nos resultados. O experimento identificou corretamente os modos de operação de 59 (37%) criptogramas submetidos ao processo. O mesmo efeito produzido pelos criptogramas que usaram o modo CBC na fase 1, ocorreu na fase 2, diminuindo o número de acertos.

Os criptogramas identificados foram os cifrados no modo de operação ECB. No modo CBC, o experimento não foi capaz de classificar os criptogramas, dado que não se pôde obter similaridade entre os mesmos. Mesmo assim, foram criados novos grupos sempre que o acerto foi menor do que 100%, o que indica que criptogramas cifrados com modos de operações diferentes não se misturam em um mesmo grupo, garantindo a máxima precisão.

Tabela 5.17. Identificação dos modos de operação com grupos ECB e CBC

Tamanho dos criptogramas	MARS		RC6		Rijndael		Serpent		Twofish	
	Acerto	Novo grupo	Acerto	Novo grupo	Acerto	Novo grupo	Acerto	Novo grupo	Acerto	Novo grupo
1024	11%	Sim	11%	Sim	11%	Sim	11%	Sim	11%	Sim
1536	12%	Sim	12%	Sim	12%	Sim	12%	Sim	12%	Sim
2048	35%	Sim	35%	Sim	35%	Sim	35%	Sim	35%	Sim
2560	46%	Sim	46%	Sim	46%	Sim	46%	Sim	46%	Sim
3072	45%	Sim	45%	Sim	45%	Sim	45%	Sim	45%	Sim
4096	39%	Sim	39%	Sim	39%	Sim	39%	Sim	39%	Sim
6144	50%	Não	50%	Não	50%	Não	50%	Não	50%	Não
8192	50%	Não	50%	Não	50%	Não	50%	Não	50%	Não
10240	50%	Não	50%	Não	50%	Não	50%	Não	50%	Não

Observando a tabela 5.18, onde se considera apenas os criptogramas que usaram o modo ECB, observa-se que a identificação alcançou resultados idênticos ao experimento 1. Neste resultado, percebe-se que as transformações realizadas pelos algoritmos não influenciam nos resultados.

Tabela 5.18. Identificação para grupos no modo ECB

Tamanho dos criptogramas	MARS		RC6		Rijndael		Serpent		Twofish	
	Acerto	Novo grupo	Acerto	Novo grupo	Acerto	Novo grupo	Acerto	Novo grupo	Acerto	Novo grupo
1024	22 %	Sim	22 %	Sim	22 %	Sim	22 %	Sim	22 %	Sim
1536	24 %	Sim	24 %	Sim	24 %	Sim	24 %	Sim	24 %	Sim
2048	69 %	Sim	69 %	Sim	69 %	Sim	69 %	Sim	69 %	Sim
2560	91 %	Sim	91 %	Sim	91 %	Sim	91 %	Sim	91 %	Sim
3072	89 %	Sim	89 %	Sim	89 %	Sim	89 %	Sim	89 %	Sim
4096	78 %	Sim	78 %	Sim	78 %	Sim	78 %	Sim	78 %	Sim
6144	100 %	Não	100 %	Não	100 %	Não	100 %	Não	100 %	Não
8192	100 %	Não	100 %	Não	100 %	Não	100 %	Não	100 %	Não
10240	100 %	Não	100 %	Não	100 %	Não	100 %	Não	100 %	Não



## 5.6 EXPERIMENTO 3: AGRUPAMENTO EM FUNÇÃO DE PARÂMETROS

Nos experimentos anteriores nota-se que: fixando a chave e o modo de operação e variando a cifra, pode-se identificar a cifra; e fixando a chave e a cifra e variando o modo de operação, pode-se identificar o modo de operação. Em SOUZA (2007) nota-se que fixando a cifra e o modo de operação e variando as chaves, podem-se agrupar os criptogramas por chaves. Assim, o objetivo deste experimento é verificar o resultado do agrupamento quando:

1. Fixado apenas o modo de operação e variando a cifra e as chaves;
2. Variando a cifra, o modo de operação e as chaves.

### 5.6.1 FIXANDO O MODO DE OPERAÇÃO E VARIANDO A CIFRA E A CHAVE

Para o objetivo “1”, este experimento utiliza a base 3 de criptogramas, onde são definidas nove coleções, cada uma com determinado tamanho. Cada coleção com 4.500 criptogramas, onde cada uma das cifras: MARS, RC6, Rijndael, Serpent e Twofish; cifrou 30 criptogramas com cada uma de trinta chaves diferentes e aleatórias de 128 *bits*, de  $k_1$  a  $k_{30}$ , no modo de operação ECB. Logo, o experimento visa medir a separação em grupos de uma combinação *algoritmo + chave*.

Tabela 5.19. Resultado da separação em grupos

Tamanho dos criptogramas	Algoritmo + chave	
	P	R
1024	1	0,17
1536	1	0,20
2048	1	0,33
2560	1	0,50
3072	1	0,87
4096	1	0,97
6144	1	1
8192	1	1
10240	1	1

Neste contexto, o experimento foi realizado e separou corretamente os criptogramas com tamanho a partir de 1024 *bytes*, de maneira que cada grupo continha criptogramas gerados com o mesmo algoritmo e com a mesma chave e somente com eles, obtendo valor máximo de precisão. O procedimento não mistura no mesmo grupo criptogramas oriundos de algoritmos e chaves diferentes.

Na tabela 5.19 estão relatados os valores de revocação e precisão onde se observa que os mesmos são coerentes com os demais experimentos. A tabela relata *algoritmo + chave* de forma geral dado que os grupos formados ficaram iguais para todas as combinações, não importando qual o algoritmo e nem qual a chave utilizada.

### 5.6.2 VARIANDO A CIFRA, O MODO DE OPERAÇÃO E A CHAVE

Para o objetivo “2”, este experimento utiliza a base 4 de criptogramas, onde são definidas nove coleções, cada uma com determinado tamanho. Cada coleção com 9.000 criptogramas, onde cada uma das cifras: MARS, RC6, Rijndael, Serpent e Twofish; cifrou 30 criptogramas com cada uma de trinta chaves diferentes e aleatórias de 128 *bits*, de  $k_1$  a  $k_{30}$ , para cada um dos modos de operação: ECB e CBC. Logo, o experimento visa medir a separação em grupos de uma combinação *algoritmo + chave + modo de operação*.

Neste contexto, o experimento foi realizado e separou corretamente os criptogramas com tamanho a partir de 1024 *bytes*, de maneira que cada grupo continha criptogramas gerados com o mesmo algoritmo, com a mesma chave e com o mesmo modo e somente com eles, obtendo valor máximo de precisão. O procedimento não mistura no mesmo grupo, criptogramas oriundos de algoritmos, chaves e modos de operação diferentes.

Tabela 5.20. Resultado da separação em grupos com diferentes combinações

Tamanho dos criptogramas	Algoritmo + chave + ECB		Algoritmo + chave + CBC		Algoritmo + chave + ECB + CBC	
	P	R	P	R	P	R
1024	1	0,17	1	0,03	1	0,10
1536	1	0,20	1	0,03	1	0,12
2048	1	0,33	1	0,03	1	0,18
2560	1	0,50	1	0,03	1	0,27
3072	1	0,87	1	0,03	1	0,45
4096	1	0,97	1	0,03	1	0,50
6144	1	1	1	0,03	1	0,52
8192	1	1	1	0,03	1	0,52
10240	1	1	1	0,03	1	0,52

Na tabela 5.20 estão relatados os valores de revocação e precisão onde se observa que os mesmos são coerentes com os demais experimentos. Os grupos formados ficaram iguais para todas as combinações de *algoritmo + chave*. Entretanto, no caso da revocação existe

diferença dependendo do modo de operação utilizado. Nota-se que os valores foram iguais aos relatados na tabela 5.13 do experimento 1.

O resultado deste experimento mostra que cada combinação de parâmetros criptográficos gera uma única assinatura no criptograma, a qual permite identificar tais parâmetros por técnicas de classificação.

### 5.6.3 SEPARAÇÃO USANDO UM NÚMERO INFERIOR DE RODADAS

O objetivo do experimento é verificar a influência de um pequeno número de rodadas no processo de separação dos grupos.

Este experimento utiliza a base 5 de criptogramas, onde são definidas nove coleções para uma, duas, três e quatro rodadas dos algoritmos, cada coleção com determinado tamanho. Cada coleção é composta por 150 criptogramas, gerados pelos algoritmos: MARS, RC6, Rijndael, Serpent e Twofish, a partir dos mesmos 30 textos em claro e uma única chave aleatória  $k$  de 128 *bits* no modo de operação ECB.

Observa-se na tabela 5.21 que os resultados foram iguais aos da tabela 5.11 do experimento 1, demonstrando que processo de separação dos grupos é bem sucedido a partir do menor número de rodadas possível.

Tabela 5.21. Resultado da Separação em Grupos por Rodadas

Tamanho dos criptogramas	1 rodada		2 rodadas		3 rodadas		4 rodadas	
	P	R	P	R	P	R	P	R
1024	1	0,17	1	0,17	1	0,17	1	0,17
1536	1	0,20	1	0,20	1	0,20	1	0,20
2048	1	0,33	1	0,33	1	0,33	1	0,33
2560	1	0,50	1	0,50	1	0,50	1	0,50
3072	1	0,87	1	0,87	1	0,87	1	0,87
4096	1	0,97	1	0,97	1	0,97	1	0,97
6144	1	1	1	1	1	1	1	1
8192	1	1	1	1	1	1	1	1
10240	1	1	1	1	1	1	1	1

### 5.7 EXPERIMENTO 4: COMPOSIÇÃO DE 5 CIFRAS IGUAIS NO MODO ECB

O objetivo deste experimento é verificar o resultado do agrupamento com a composição de cinco cifras iguais, quando utilizada:

1. Uma única chave para todas as composições; e

2. Uma chave diferente para cada uma das composições.

### 5.7.1 SEPARAÇÃO EM GRUPOS FIXANDO UMA CHAVE

Para o objetivo “1”, este experimento utiliza a base 6 de criptogramas, onde são definidas nove coleções, cada uma com determinado tamanho. Cada coleção com 150 criptogramas, onde cada uma das cifras: MARS, RC6, Rijndael, Serpent e Twofish; cifrou por cinco vezes consecutivas 30 criptogramas com uma única chave aleatória  $k$  de 128 *bits*, no modo de operação ECB.

Observa-se na tabela 5.22 que os resultados foram iguais ao da tabela 5.13 do experimento 1, demonstrando que as composições usando a mesma cifra e a mesma chave não produzem nenhum efeito no processo de separação dos grupos.

Tabela 5.22. Separação em grupos com cinco composições e uma chave

Tamanho dos criptogramas	P	R
1024	1	0,17
1536	1	0,20
2048	1	0,33
2560	1	0,50
3072	1	0,87
4096	1	0,97
6144	1	1
8192	1	1
10240	1	1

### 5.7.2 SEPARAÇÃO EM GRUPOS USANDO CHAVES DIFERENTES

Para o objetivo “2”, este experimento utiliza a base 7 de criptogramas, onde são definidas nove coleções, cada uma com determinado tamanho. Cada coleção com 150 criptogramas, onde cada uma das cifras: MARS, RC6, Rijndael, Serpent e Twofish; cifrou por cinco vezes consecutivas 30 criptogramas com cinco chaves diferentes e aleatórias de 128 *bits*, de  $k_1$  a  $k_5$ , no modo de operação ECB.

Observa-se na tabela 5.23 que os resultados foram iguais aos da tabela 5.13 do experimento 1 e iguais aos da tabela 5.22 da seção anterior, demonstrando que as composições usando a mesma cifra com chaves diferentes não produzem nenhum efeito no processo de separação dos grupos.

Tabela 5.23. Separação em grupos com cinco composições e cinco chaves

Tamanho dos criptogramas	P	R
1024	1	0,17
1536	1	0,20
2048	1	0,33
2560	1	0,50
3072	1	0,87
4096	1	0,97
6144	1	1
8192	1	1
10240	1	1

## 5.8 EXPERIMENTO 5: COMPOSIÇÃO DE 5 CIFRAS DIFERENTES MODO ECB

### 5.8.1 SEPARAÇÃO DOS GRUPOS

Dada a evidência de que as propriedades intrínsecas dos modelos matemáticos de cada cifra geram assinaturas nos criptogramas, o que foi parcialmente demonstrado pelos experimentos anteriores, o objetivo deste experimento é verificar se a composição de cifras diferentes permite gerar assinaturas nos criptogramas, de tal forma que a última cifra utilizada na composição seja corretamente agrupada.

Cada coleção é composta por 150 criptogramas cifrados com uma única chave aleatória  $k$  de 128 *bits*, por composições com cinco cifras de blocos, conforme especificado para a base 8 de criptogramas:

$$\begin{aligned}
 &MARS_k^{ECB} (RC6_k^{ECB} (Rijndael_k^{ECB} (Serpent_k^{ECB} (Twofish_k^{ECB} (texto_1, \dots, texto_{30})))))); \\
 &RC6_k^{ECB} (Rijndael_k^{ECB} (Serpent_k^{ECB} (Twofish_k^{ECB} (MARS_k^{ECB} (texto_1, \dots, texto_{30})))))); \\
 &Rijndael_k^{ECB} (Serpent_k^{ECB} (Twofish_k^{ECB} (MARS_k^{ECB} (RC6_k^{ECB} (texto_1, \dots, texto_{30})))))); \\
 &Serpent_k^{ECB} (Twofish_k^{ECB} (MARS_k^{ECB} (RC6_k^{ECB} (Rijndael_k^{ECB} (texto_1, \dots, texto_{30})))))); \\
 &Twofish_k^{ECB} (MARS_k^{ECB} (RC6_k^{ECB} (Rijndael_k^{ECB} (Serpent_k^{ECB} (texto_1, \dots, texto_{30}))))).
 \end{aligned}$$

Observa-se na tabela 5.24 que os resultados foram iguais aos da tabela 5.13 do experimento 1 e iguais aos das tabelas 5.22 e 5.23 do experimento 4, demonstrando que as

composições usando a cifras diferentes não produzem nenhum efeito no processo de separação dos grupos.

Tabela 5.24. Separação em grupos usando composição de 5 cifras de bloco

Tamanho dos criptogramas	P	R
1024	1	0,17
1536	1	0,20
2048	1	0,33
2560	1	0,50
3072	1	0,87
4096	1	0,97
6144	1	1
8192	1	1
10240	1	1

## 5.9 EXPERIMENTO 6: COMPOSIÇÃO DE 5 CIFRAS NOS MODOS ECB E CBC

Observa-se no experimento 1 que embora a precisão tenha sido máxima, tanto a separação quanto a classificação não tiveram sucesso no modo CBC. Considerando os resultados dos experimentos anteriores, neste experimento objetiva-se testar:

1. Dada uma composição com cinco cifras de blocos, se a mistura gerada pelas quatro primeiras cifrações usando o modo CBC pode ser comprometida por uma última cifração no modo ECB; e

2. Dada uma composição com cinco cifras de blocos, se a repetição de blocos gerada por quatro cifrações no modo ECB pode ser desfeita por uma última cifração no modo CBC.

Para cada um dos objetivos “1” e “2” acima, são definidas nove coleções, cada uma com determinado tamanho. Cada coleção é composta por 150 criptogramas, cifrados com uma única chave aleatória  $k$  de 128 *bits*, por composições com cinco cifras de blocos, onde para o objetivo “a” as quatro primeiras cifras usam o modo CBC e a última usa o modo ECB, conforme especificado para a base 9 de criptogramas. Para o objetivo “b” as quatro primeiras cifras usam o modo ECB e a última usa o modo CBC, conforme especificado para a base 10 de criptogramas.

Na tabela 5.25, pode-se ver que, como esperado, o resultado até a quarta cifração não gerou repetição de nenhum padrão e a quinta cifração no modo ECB não comprometeu o efeito das cifrações anteriores.

Tabela 5.25. Quatro composições no modo CBC e a última no modo ECB

Tamanho dos criptogramas	Quatro cifrações no modo CBC		Última cifração no modo ECB	
	Precisão	Revocação	Precisão	Revocação
1024	1	0,03	1	0,03
1536	1	0,03	1	0,03
2048	1	0,03	1	0,03
2560	1	0,03	1	0,03
3072	1	0,03	1	0,03
4096	1	0,03	1	0,03
6144	1	0,03	1	0,03
8192	1	0,03	1	0,03
10240	1	0,03	1	0,03

Os resultados na tabela 5.26 mostram que uma última cifração no modo CBC destrói todos os padrões repetidos que se propagaram ao longo das quatro cifrações anteriores no modo ECB.

Os valores de precisão e revocação nas tabelas 5.25 e 5.26 se mantiveram coerentes com os experimentos anteriores.

Tabela 5.26. Quatro composições no modo ECB e a última no modo CBC

Tamanho dos criptogramas	Quatro cifrações no modo ECB		Última cifração no modo CBC	
	Precisão	Revocação	Precisão	Revocação
1024	1	0,17	1	0,03
1536	1	0,20	1	0,03
2048	1	0,33	1	0,03
2560	1	0,50	1	0,03
3072	1	0,87	1	0,03
4096	1	0,97	1	0,03
6144	1	1	1	0,03
8192	1	1	1	0,03
10240	1	1	1	0,03

## 5.10 EXPERIMENTO 7: SEPARAÇÃO DE MENSAGENS E CRIPTOGRAMAS

O objetivo dos experimentos é verificar se a metodologia é capaz de separar tanto as mensagens quanto os criptogramas em diversos conjuntos, de tal forma que os criptogramas gerados pelo mesmo algoritmo e chave – e somente eles – fiquem agrupados no mesmo conjunto, da mesma forma que os textos legíveis de um mesmo idioma e alfabeto. Com a finalidade de mostrar a influência dos elementos léxicos (JURAFSKY e MARTIN, 2009) no

processo de agrupamento, foram selecionados textos legíveis escritos em idiomas e alfabetos diferentes, além dos conjuntos de textos cifrados.

### 5.10.1 SEPARAÇÃO DE MENSAGENS

O objetivo deste experimento é verificar a separação em grupos somente das mensagens. Foram utilizados 30 textos de cada um dos idiomas a seguir: alemão, dinamarquês, holandês, espanhol, francês, grego, hebreu e português, conforme especificado para a base 11. Para identificar o valor máximo de precisão e revocação, foram testados três valores de corte (tabela 5.27).

Tabela 5.27. Separação em grupos somente das mensagens

Oito idiomas diferentes, sendo dois com alfabetos distinto dos demais.		
Corte	Precisão	Revocação
0,001	0,375	1
0,455	1	1
0,550	1	1

Observando a tabela 5.27, pode-se concluir que o agrupamento ocorreu com sucesso, obtendo o valor máximo de precisão e revocação a partir da similaridade de corte 0,455. Os idiomas utilizados no experimento indicam a existência de nove grupos naturais, cada grupo representado por um idioma. Uma inspeção no conteúdo dos textos pode sugerir a existência de quatro grupos representados por um valor de similaridade qualquer no  $[0,1]$ , com a configuração abaixo:

1. Grupo 1: alemão, dinamarquês, holandês;
2. Grupo 2: espanhol, francês e português;
3. Grupo 3: grego; e
4. Grupo 4: hebreu.

A formação acima se deve a maior ou menor interseção de termos entre os textos, o que indica que à medida que o valor de corte tende a zero a precisão é reduzida uma vez que uma menor quantidade de interseções é requerida para gerar a pertinência a um grupo, degenerando a um grupo quando do valor zero.

Por outro lado, quando o valor de corte tende a um, a precisão aumenta, já que uma interseção cada vez maior de termos é exigida para a pertinência a um grupo, levando ao caso trivial onde cada grupo possui apenas um texto.



Observando-se os grupos formados com o valor de corte 0,001, nota-se que embora o processo tenha gerado grupos imprecisos, os textos em grego e em hebreu não se misturaram em grupos de outros idiomas, o que reduz o processo de agrupamento aqui descrito a um fato estatístico (RASMUSSEN, 1992) (JAIN, 1999).

### 5.10.2 SEPARAÇÃO DOS CRIPTOGRAMAS

O objetivo deste experimento é verificar a separação em grupos somente dos criptogramas. Foram utilizados dois conjuntos de 30 criptogramas cada, gerados pelo algoritmo AES, cada conjunto cifrado com chave diferente e aleatória de 128 *bits*, e dois conjuntos de 30 criptogramas cada, gerados pelo algoritmo DES, cada conjunto cifrado com chave diferente e aleatória de 64 *bits*, conforme especificado para a base 12. Para identificar o valor máximo de precisão e revocação, foram testados três valores de corte (tabela 5.28).

Tabela 5.28. Separação em grupos somente dos criptogramas

Dois conjuntos de criptogramas cifrados com o algoritmo AES e dois cifrados com o DES.		
Corte	Precisão	Revocação
0,001	1	1
0,455	1	0,03
0,550	1	0,03

Observando a tabela 5.28, pode-se concluir que o agrupamento ocorreu com sucesso, obtendo o valor máximo de precisão e revocação com a similaridade de corte 0,001.

Considerando o princípio de que uma chave determina um idioma, o qual possui o seu próprio conjunto léxico, pode-se deduzir a existência de quatro grupos naturais, cada grupo representado por uma chave criptográfica utilizada no processo de cifrar, duas para o algoritmo AES e duas para o algoritmo DES.

O experimento demonstra que os quatros grupos são formados a partir do valor de corte 0,001. Considerado o fato de que poucos termos se repetem ao longo dos criptogramas, é esperado que mesmo criptogramas com baixa interseção de termos fiquem no mesmo grupo. Este fato, somado ao efeito de dispersão criado pelo método da ligação simples (seção 4.5), justificam o sucesso do agrupamento com um baixo valor de corte.

No caso dos criptogramas, a baixa interseção entre estes leva o resultado do processo rapidamente ao caso trivial onde cada grupo possui apenas um criptograma, quando o valor de corte tende a um.

### 5.10.3 SEPARAÇÃO DAS MENSAGENS E CRIPTOGRAMAS

O objetivo deste experimento é verificar a separação em grupos a partir de uma coleção composta por mensagem e criptogramas. Foram utilizados dois conjuntos de 30 criptogramas cada, gerados pelo algoritmo AES, cada conjunto cifrado com chave diferente e aleatória de 128 *bits*, e dois conjuntos de 30 criptogramas cada, gerados pelo algoritmo DES, cada conjunto cifrado com chave diferente e aleatória de 64 *bits*, e 30 textos de cada um dos idiomas a seguir: alemão, dinamarquês, holandês, espanhol, francês, grego, hebreu e português, conforme especificado para a base 13. Para identificar o valor máximo de precisão e revocação, foram testados três valores de corte (tabela 5.29).

Tabela 5.29. Separação em grupos das mensagens e criptogramas

Coleção de criptogramas cifrados com o AES e DES e mensagens de oito idiomas diferentes		
Corte	Precisão	Revocação
0,001	0,583	1
0,455	1	1
0,550	1	1

Considerando a hipótese de que os criptogramas podem ser considerados textos de um idioma desconhecido e determinado por uma combinação qualquer de parâmetros criptográficos e observando-se os resultados dos experimentos das seções 5.10.1 e 5.10.2, tem-se que juntar os dados destes experimentos equivale a ter 12 grupos naturais, onde cada grupo possui documentos escritos em idiomas diferentes, e onde alguns desses idiomas utilizam alfabetos diferentes (binário, grego e hebraico).

Assim, reduz-se o problema do agrupamento desta coleção ao número de interseções de termos ao longo dos documentos.

A separação em grupos ocorreu com sucesso (tabela 5.29), obtendo o valor máximo de precisão e revocação a partir da similaridade de corte 0,455.

#### 5.10.4 SEPARAÇÃO POR ALFABETO

O objetivo deste experimento é verificar a separação em grupos a partir de mensagens escritas com alfabetos diferentes, ou seja, sem repetir idiomas que utilizam o mesmo alfabeto, e mais os conjuntos de criptogramas. Foram utilizados dois conjuntos de 30 criptogramas cada, gerados pelo algoritmo AES, cada conjunto cifrado com chave diferente e aleatória de 128 *bits*, e dois conjuntos de 30 criptogramas cada, gerados pelo algoritmo DES, cada conjunto cifrado com chave diferente e aleatória de 64 *bits*, e 30 textos de cada um dos idiomas a seguir: grego, hebreu e português, conforme especificado para a base 14. Para identificar o valor máximo de precisão e revocação, foram testados três valores de corte (tabela 5.30).

Tabela 5.30. Separação em grupos por alfabeto

Coleção de criptogramas cifrados com o AES e DES mais os idiomas português, grego, hebreu.		
Corte	Precisão	Revocação
0,001	1	1
0,455	1	1
0,550	1	1

Na tabela 5.30, observa-se que a separação em grupos ocorreu com sucesso, com precisão e revocação máxima para todas as similaridades de corte. Mais uma vez, reduz-se o problema da separação em grupos ao número de interseções de termos ao longo dos objetos. Neste caso, a interseção entre os sete grupos naturais, é zero, dado que os termos estão todos escritos em alfabetos diferentes.

#### 5.11 ANÁLISE ESTATÍSTICA DAS BASES DE CRIPTOGRAMAS

Na contagem de blocos repetidos ou não, constata-se que quanto mais blocos uma coleção de textos possui, maior a chance de repetição. Na tabela 5.31, nota-se que os mesmos blocos que se repetem nos textos claros, se repetem nos criptogramas cifrados no modo ECB. Quando a cifração é realizada no modo CBC, não há repetições. Tais resultados são esperados.

Tabela 5.31. Contagem de blocos e repetição de blocos nos textos claros

Tamanho (em <i>bytes</i> )	30 Textos Claros		30 criptogramas (ECB)	
	Blocos	Repete	Blocos	Repete
1024	1920	25	1920	25
1536	2880	29	2880	29
2048	3840	38	3840	38
2560	4800	58	4800	58
3072	5760	72	5760	72
4096	7680	105	7680	105
6144	11520	260	11520	260
8192	15360	409	15360	409
10240	19200	565	19200	565

## 6 RESULTADOS TEÓRICOS

### 6.1 GENERALIZAÇÃO DA METODOLOGIA PARA $N - CIFRAS$ DE BLOCO

Comparando os modos de operação, o processo de encadeamento utilizado pelo CBC foi o responsável pela não obtenção de resultados semelhantes nos experimentos no modo ECB. Em outras palavras, considerando as definições 2.1 (sistema criptográfico), 2.2 (cifra de blocos), 2.3 (composição de cifras de blocos), 2.4 (modo de operação ECB) e 2.5 (modo de operação CBC), o modo CBC pode ser reescrito da seguinte maneira: Se  $t_i$  e  $t_j \in t$ ,  $i, j = 1, \dots, n$ , então  $e_k(t_i \oplus c_{i-1}) \neq e_k(t_j \oplus c_{j-1})$ . Isto é  $c_i \neq c_j$ . Esse fato acrescentou complexidade aos processos de separação e classificação de criptogramas.

**Proposição 6.1:** Sejam  $k \in K$  e  $\bigcup_{\lambda=1}^m e_k^\lambda = \{e_k^1, e_k^2, \dots, e_k^m\}$  uma família de regras de cifração

distintas no modo ECB, de modo que o domínio de  $e_k^m$  seja igual ao contradomínio de  $e_k^{m-1}$  e assim por diante.

Se  $t_i = t_j$ ,  $i \neq j$  onde  $i, j = 1, \dots, n$ ,

Então  $c_i^m = e_k^m \circ \dots \circ e_k^2 \circ e_k^1(t_i) = e_k^m \circ \dots \circ e_k^2 \circ e_k^1(t_j) = c_j^m$ , onde  $c_i^m$  e  $c_j^m$  são blocos de algum criptograma  $c \in C$ , gerados pela regra de cifração  $e_k^m$ .

**Prova 6.1:** Sejam  $t_i, t_j \in t$ , tal que  $t_i = t_j$  com  $i \neq j$ , para  $i, j = 1, \dots, n$  e  $P(m) : e_k^m \circ \dots \circ e_k^2 \circ e_k^1$ . Daí, pela Indução Matemática no número de regras de cifração vem:

i. Verificar se  $P(1)$  é verdadeira.

Para  $m = 1$ , tem-se que  $P(1) : e_k^1$ . Aplicando  $e_k^1$  a  $t_i$  e  $t_j$  tem-se que:  $e_k^1(t_i) = c_i^1$  e  $e_k^1(t_j) = c_j^1$ . Por hipótese  $e_k^1$  é uma cifra de bloco no modo ECB e  $t_i = t_j$ , logo pela definição 2.3 (modo ECB) tem-se que  $c_i^1 = c_j^1$ , onde  $c_i^1$  e  $c_j^1$  são blocos de algum criptograma  $c \in C$ .

ii. Hipótese de Indução

Neste caso, supõem-se que para  $m = u$ ,  $P(u) = e_k^u \circ \dots \circ e_k^2 \circ e_k^1$  seja verdadeira. Isto equivale a dizer que:

$$c_i^u = e_k^u \circ \dots \circ e_k^2 \circ e_k^1(t_i) = e_k^u \circ \dots \circ e_k^2 \circ e_k^1(t_j) = c_j^u$$

iii. Será provado agora para  $m = u + 1$ , ou seja,  $P(u + 1) = e_k^v \circ e_k^u \circ \dots \circ e_k^2 \circ e_k^1$ .

Tem-se que provar o seguinte:

$$c_i^v = e_k^v \circ e_k^u \circ \dots \circ e_k^2 \circ e_k^1(t_i) = e_k^v \circ e_k^u \circ \dots \circ e_k^2 \circ e_k^1(t_j) = c_j^v$$

A idéia da prova é mostrar que  $c_i^v = c_j^v$ , onde  $c_i^v$  e  $c_j^v$  são blocos de algum criptograma  $c \in C$  gerados pela regra de cifração  $e_k^v$ . Pela hipótese de indução tem-se que:

$c_i^u = e_k^u \circ \dots \circ e_k^2 \circ e_k^1(t_i) = e_k^u \circ \dots \circ e_k^2 \circ e_k^1(t_j) = c_j^u$ . Assim, tem-se que  $c_i^u = c_j^u$  obtidos pela aplicação da regra de cifração  $e_k^u$ . Tais blocos são elementos de algum criptograma  $c \in C$ . A obtenção destes criptogramas é feita recursivamente pela aplicação da definição 2.3 (modo ECB). Como  $c_i^u = c_j^u$  e a regra de cifração  $e_k^v$  esta no modo ECB, então, pode-se aplicar novamente a definição 2.3 (modo ECB). Daí, vem:

$$e_k^v(c_i^u) = e_k^v(c_j^u), \text{ ou seja, } c_i^v = c_j^v.$$

cqd.

#### Observações:

a) Se as regras de cifração da família  $\bigcup_{\lambda=1}^m e_k^\lambda$  forem iguais, ou seja,  $e_k^m = e_k^{m-1}, \dots, e_k^2 = e_k^1$ , a prova seria semelhante ao que foi feito na Proposição 6.1; e

b) Novamente, se as regras de cifração da família  $\bigcup_{\lambda=1}^m e_k^\lambda$  forem aplicadas a um conjunto de textos claros  $t_1, t_2, \dots, t_n$ ; a prova da Proposição 6.1 seria semelhante. A diferença é que deveríamos considerar para cada texto  $t_s$ ,  $s = 1, \dots, n$ , blocos da forma  $t_{1,i}$  e  $t_{1,j}$ ;  $t_{2,i}$  e  $t_{2,j}$  até  $t_{n,i}$  e  $t_{n,j}$  e aplicar o mesmo raciocínio utilizado na demonstração da Proposição 6.1.

**Proposição 6.2:** Sejam  $k \in K$  e  $\bigcup_{\lambda=1}^m e_k^\lambda = \{e_k^1, e_k^2, \dots, e_k^m\}$  uma família de regras de cifração distintas no modo CBC, de maneira que o domínio de  $e_k^m$  seja igual ao contradomínio de  $e_k^{m-1}$  e assim por diante.

Se  $t_i, t_j \in t$ , distintos ou não, onde  $i, j = 1, \dots, n$ .

$$\text{Então } c_i^m = e_k^m \circ \dots \circ e_k^2 \circ e_k^1(t_i) \neq c_j^m = e_k^m \circ \dots \circ e_k^2 \circ e_k^1(t_j).$$

**Prova 6.2:** Basta usar a definição 2.4 (modo CBC) e repetir o mesmo raciocínio utilizado na demonstração da Proposição 6.1. cqd.

**Observação:** como foi visto nos resultados dos experimentos computacionais relativos ao modo de operação CBC, os valores baixos de revocação indicaram desempenho ruim do processo de agrupamento e classificação utilizado nesta tese. Mas, visto que o espaço de criptogramas é finito, acredita-se que a metodologia pode ser aplicada com sucesso no modo CBC e a outros modos de operação, desde que se aumente o espaço amostral de criptogramas, semelhante ao que ocorre nas técnicas de criptoanálise linear e diferencial. O aumento deste espaço amostral pode se dar à medida que se aumente também o poder computacional.

Baseado nos resultados das Proposições 6.1 e 6.2, verifica-se que as regras de cifração no modo ECB serão adequadas aos processos de separação e classificação de criptogramas de modo a permitir a identificação de cifras. Entretanto no modo CBC a identificação não é possível. Isto se deve ao processo de encadeamento utilizado no modo CBC.

## 6.2 QUALQUER MÉTODO, BASEADO NO CONJUNTO LÉXICO DE UMA LINGUAGEM, USADO PARA AGRUPAR E CLASSIFICAR TEXTOS CLAROS PODE AGRUPAR E CLASSIFICAR CRIPTOGRAMAS

**Proposição 6.3:** Sejam  $(D, G, s_{medida}, M)$  um método de agrupamento qualquer usado na Teoria de Recuperação de Informações e  $(C, \overline{G}, s_{medida}, \overline{M})$  o método de agrupamento usado nesta tese, onde  $e_k(D) = C$ ,  $k \in K$  e  $\bigcup_{\lambda=1}^m e_k^\lambda = \{e_k^1, e_k^2, \dots, e_k^m\}$  uma família de regras de cifração distintas no modo ECB,. Então,

$$G = \overline{G}.$$

**Prova 6.3:** Sejam  $\psi_i$  e  $\psi_j$ ,  $i \neq j$ , palavras distintas de  $D$  e  $C$ , respectivamente. Da convenção acima, vamos supor que  $|\psi_i| = |\psi_j| = l$ ,  $l > 0$  e  $l \in \aleph$ .

Da definição 2.16, tem-se que  $M(D, s_{medida}) = G$  e  $\overline{M}(C, s_{medida}) = \overline{G}$ . Como  $e_k(D) = C$ , tem-se da definição 2.1 que  $d_k(e_k(D)) = d_k(C) = D$ . Como  $|\psi_i| = |\psi_j|$ , isto significa que as palavras de  $D$  e  $C$  são agrupadas da mesma forma.

Como  $s_{medida}$  de  $M$  é igual a  $s_{medida}$  de  $\overline{M}$ ,  $M$  e  $\overline{M}$  só consideram os elementos léxicos, isto é, as palavras e suas freqüências, os grupos de  $G$  são idênticos aos de  $\overline{G}$  e pela

definição 2.16, se aplicarmos  $\forall i, d_k(\bar{g}_i) = g_i$  com  $\bar{g}_i \in \bar{G}$  e  $g_i \in G$ , então temos que  $G = \bar{G}$ .  
 cqd.

A Proposição acima estabelece que técnicas para agrupar documentos em sistemas de Recuperação de Informação em uma linguagem  $L$ , podem ser usadas para agrupar criptogramas gerados por  $e_k$  e vice-versa. Ressalta-se, a título de ilustração, que este agrupamento só será válido se  $M$  e  $\bar{M}$  utilizem a mesma função  $s_{medida}$ ,  $M$  e  $\bar{M}$  só consideram os elementos léxicos, isto é, as palavras e suas frequências e  $e_k(D) = C$ .

Como o processo de classificação será bem sucedido se o processo de agrupamento também o for, então pela Proposição 6.3 técnicas para classificar documentos em sistemas de Recuperação de Informação em uma linguagem  $L$ , podem ser usadas para classificar criptogramas gerados por  $e_k$  e vice-versa.

**Corolário 6.1:** Sejam  $D$  uma coleção de documentos e  $IV$  uma palavra fixa de  $L$ , tal que  $|IV| = |\psi|, \psi \in D$ ; e  $D' = \{\psi \oplus IV \mid \psi \in D\}$ , onde  $\oplus$  denota a operação “ou exclusivo”. Se  $e_k(D') = C'$ , então

$$M(D', s_{medida}) = \bar{M}(C', s_{medida}).$$

Baseado na definição 2.17, observou-se que uma regra de cifração define um contexto sobre um conjunto de documentos cifrados por essa regra. Cabe ressaltar que se  $e_k^\lambda \in \{e_k^1, e_k^2, \dots, e_k^m\}$ ,  $\lambda = 1, \dots, m$ , opera no modo ECB, haverá uma quantidade maior de palavras repetidas no conjunto de documentos cifrados, oriundas das repetições das palavras que ocorrem nos documentos em claro de  $L$ . Por outro lado, se a regra de cifração opera no modo CBC não haverá, com grande probabilidade, repetições de padrões. Para que a repetição de padrões ocorra, seria necessária uma quantidade de documentos muito grande, os quais não seriam viáveis processar com a atual tecnologia dos computadores, considerando aspectos de memória e capacidade de processamento. Essa característica intrínseca do modo CBC de uma regra de cifração, é responsável pela aleatorização dos documentos cifrados, destruindo, dessa forma, a repetição de palavras que normalmente ocorrem na mesma linguagem de acordo com algum contexto.

A seguir, serão apresentados alguns resultados envolvendo o conceito de  $IdC_{\psi_r}$ , juntamente com a combinação de parâmetros utilizados em uma família de regras de cifração distintas ou não.



### 6.3 CADA CIFRA GERA UM IDENTIFICADOR DE CONTEXTO

**Proposição 6.4:** Sejam  $k \in K$ , uma chave criptográfica e  $\bigcup_{\lambda=1}^m e_k^\lambda = \{e_k^1, e_k^2, \dots, e_k^m\}$  uma família de regras de cifração, no modo ECB. Então,

$$e_k^i \in \bigcup_{\lambda=1}^m e_k^\lambda, e_k^i, i = 1, \dots, m; \text{ gera identificadores de contextos distintos.}$$

**Prova 6.4:** Sejam  $k \in K$  e  $e_k^\lambda, \lambda = 1, \dots, m$ , uma regra de cifração da família  $\bigcup_{\lambda=1}^m e_k^\lambda$ . Se  $t_i = (t_{1,i}, t_{2,i}, \dots, t_{n,i}), t_i \in T$ , é um texto claro formado pelos blocos  $t_{j,i}, j = 1, \dots, n$ . Então, dado algum contexto  $\psi_r$  de  $t_i \in L$ , devemos considerar as seguintes possibilidades:

i. Existe  $IdC_{\psi_r} = (t_{1,i}, t_{2,i}, \dots, t_{l,i})$ , onde  $l \leq n$ , formado por blocos repetidos de  $t_i$ , tais que  $|t_{1,i}| = |t_{2,i}| = \dots = |t_{n,i}|$  e  $\sum_{j=1}^n |t_{j,i}| = |t_i|$ ; e

ii. Existe  $IdC_{\psi_r} = (t_{1,i}, t_{2,i}, \dots, t_{l,i})$ , onde  $l = n$ , formado por blocos não repetidos de  $t_i$ , tais que  $|t_{1,i}| = |t_{2,i}| = \dots = |t_{n,i}|$  e  $\sum_{j=1}^n |t_{j,i}| = |t_i|$ .

Considerando o item “i” acima, ou seja, que haja repetições de blocos de  $t_i$ .

Seja  $e_k^s$  uma regra de cifração qualquer da família  $\bigcup_{\lambda=1}^m e_k^\lambda$ . Supondo que  $\exists l \leq n$ , tal que  $t_{j,i} = t_{l,i}$ , tem-se:

$$e_k^s(t_{j,i}) = c_{j,i}^1 = e_k^s(t_{l,i}) = c_{l,i}^1, \text{ com } |c_{j,i}^1| = |c_{l,i}^1|;$$

$$e_k^s(c_{j,i}^1) = c_{j,i}^2 = e_k^s(c_{l,i}^1) = c_{l,i}^2, \text{ com } |c_{j,i}^2| = |c_{l,i}^2|.$$

Aplicando esse raciocínio  $u$  – vezes, obtém-se o Identificador  $IdC_{e_k^s} = (c_{1,i}^u, c_{2,i}^u, \dots, c_{l,i}^u)$ .

Supondo que existam pelo menos dois blocos distintos, por exemplo,  $t_{j,i} \neq t_{l,i}$ , ou seja  $j \neq l$ . Como  $e_k^s$  é injetora e aplicando esta função  $u$  – vezes, logo se tem que  $e_k^s(t_{j,i})^u \neq e_k^s(t_{l,i})^u$ .

Supondo-se agora  $e_k^r \in \bigcup_{\lambda=1}^m e_k^\lambda$  uma regra de cifração. Se  $r = s$ , então  $e_k^r = e_k^s$ , logo  $IdC_{e_k^s} = IdC_{e_k^r}$ . Então, nada a demonstrar.

Supondo-se, então  $e_k^r \neq e_k^s$ , logo  $r \neq s$ . Será calculado o  $IdC_{e_k^r}$  e comparado com os valores de  $IdC_{e_k^s}$ . Assim:

Se  $t_{j,i} = t_{l,i}$  e aplicando a regra de cifração  $e_k^r$ , tem-se:

Se  $e_k^r(t_{j,i}) = r_{j,i}^1$  e como  $e_k^s(t_{j,i}) = c_{j,i}^1$ , tem-se que  $r_{j,i}^1 \neq c_{j,i}^1$ , com  $|r_{j,i}^1| = |c_{j,i}^1|$ . Continuando, se  $e_k^r(r_{j,i}^1) = r_{j,i}^2$  e  $e_k^s(c_{j,i}^1) = c_{j,i}^2$ , tem-se de modo análogo que  $r_{j,i}^2 \neq c_{j,i}^2$ . Aplicando esse raciocínio  $u$  - vezes, será obtido o Identificador  $IdC_{e_k^r} = (r_{1,i}^u, r_{2,i}^u, \dots, r_{l,i}^u) \neq (c_{1,i}^u, c_{2,i}^u, \dots, c_{l,i}^u) = IdC_{e_k^s}$ .

Considerando agora a possibilidade do item “ii” acima, ou seja,  $IdC_{\psi_r} = (t_{1,i}, t_{2,i}, \dots, t_{n,i})$ , sem repetição de palavras. A demonstração é semelhante, ou seja, basta considerar que se  $t_{j,i} \neq t_{l,i}$ , então  $e_k^s(t_{j,i}) \neq e_k^s(t_{l,i})$ , pois,  $e_k^s$  é injetora.

Aplicando esse raciocínio  $u$  - vezes, como foi feito acima será obtido  $IdC_{e_k^s} = (c_{1,i}^u, c_{2,i}^u, \dots, c_{n,i}^u)$ , com blocos  $c_{j,i}^u \neq c_{l,i}^u$ ,  $l \neq j$ .

Se for considerado  $e_k^r \neq e_k^s$ , com  $t_{j,i} \neq t_{l,i}$ , a demonstração será análoga e será obtido  $IdC_{e_k^r} = (r_{1,i}^u, r_{2,i}^u, \dots, r_{n,i}^u)$ , com blocos  $r_{j,i}^u \neq r_{l,i}^u$ ,  $l \neq j$ . Da mesma forma,  $IdC_{e_k^s} \neq IdC_{e_k^r}$ . Cabe observar, que se pode, matematicamente, obter  $e_k^r(t_{j,i}) = e_k^s(t_{l,i})$ . Mas, tal resultado não ocorre na presente metodologia dado que uma mesma função é sempre aplicada a pontos distintos ou não. cqd.

A proposição acima mostra que criptogramas obtidos por regras de cifração no modo ECB podem ser classificados por qualquer método utilizado para classificar textos na Teoria de Recuperação de Informação, se utilizado o conceito de Identificador de Contexto. Isto porque dada uma linguagem  $L$  não cifrada, é sempre possível construir em  $L$  um Identificador de Contexto devido às características intrínsecas de  $L$ , como, por exemplo, a redundância.

Outra situação que cabe ressaltar é que uma regra de cifração no modo ECB propaga as palavras repetidas de um identificador de contexto, desde que não se leve em conta, por exemplo, aspectos sintáticos ou semânticos de  $L$ .

## 6.4 CADA CHAVE GERA UM IDENTIFICADOR DE CONTEXTO

**Proposição 6.5:** Sejam  $k \in K = \{k_1, k_2, \dots, k_n\}$ , um conjunto de chaves criptográficas e  $e_k$ , uma regra de cifração da família  $\bigcup_{\lambda=1}^m e_k^\lambda$ , no modo ECB,  $t_i$  um texto claro,  $t_i \in T$ , e  $c_j$  um criptograma,  $c_j \in C$ , cifrados por  $e_k$ . Então,

$$e_{k_r} \text{ e } e_{k_s} \in \bigcup_{\lambda=1}^m e_k^\lambda \text{ geram Identificadores de Contextos distintos, onde } r, s = 1, \dots, m$$

**Prova 6.5:** Sejam  $k_r, k_s \in K$ ,  $e_k$  uma regra de cifração,  $t_i = (t_{1,i}, t_{2,i}, \dots, t_{n,i})$  um texto claro formado pelos blocos  $t_{j,i}, j = 1, \dots, n$ , e  $c_j = (c_{1,j}, c_{2,j}, \dots, c_{n,j})$  um criptograma formado pelos blocos  $c_{l,i}, l = 1, \dots, n$ , onde tais blocos satisfazem as condições da Proposição 6.4. Então, consideram-se as seguintes possibilidades:

i.  $k_r = k_s$  e  $e_k^\lambda$  fixa para algum  $\lambda$ . Disto e da Proposição 6.4 item “i”, tem-se

$$IdC_{e_{k_r}}^i = IdC_{e_{k_s}}^j ;$$

ii.  $k_r = k_s$  e  $e_k^i \neq e_k^j, i \neq j$ . Disto e da Proposição 6.4 item “i”, tem-se agora

$$IdC_{e_{k_r}}^i \neq IdC_{e_{k_s}}^j ;$$

iii.  $k_r \neq k_s$  e  $e_k^\lambda$  fixa para algum  $\lambda$ . Seja  $t_{j,i}$  um bloco de texto claro. Como por hipótese  $k_r \neq k_s$ , então  $(t_{j,i}, k_r) \neq (t_{j,i}, k_s)$ , pela definição de igualdade de pares ordenados. Como  $e_k^\lambda$  é injetora, então  $e_{k_r}^\lambda(t_{j,i}) \neq e_{k_s}^\lambda(t_{j,i})$ .

Aplicando  $u - vezes$  a regra de cifração com as chaves  $k_s$  e  $k_r$  será obtido:

Se  $e_{k_r}^\lambda(t_{j,i}) = c_{j,r,i}^1$  e  $e_{k_s}^\lambda(t_{j,i}) = c_{j,s,i}^1$  então  $c_{j,r,i}^1 \neq c_{j,s,i}^1$ , pois  $(t_{j,i}, k_r) \neq (t_{j,i}, k_s)$  e  $e_k^\lambda$  é injetora.

Continuando, tem-se:

Se  $e_{k_r}^\lambda(c_{j,r,i}^1) = c_{j,r,i}^2$  e  $e_{k_s}^\lambda(c_{j,s,i}^1) = c_{j,s,i}^2$ , então  $c_{j,r,i}^2 \neq c_{j,s,i}^2$ , pelo mesmo motivo acima.

Continuando esse processo  $u - vezes$ , tem-se que:

$$IdC_{e_{k_r}}^\lambda = (c_{1,r,i}^u, c_{2,r,i}^u, \dots, c_{l,r,i}^u) \neq (c_{1,s,i}^u, c_{2,s,i}^u, \dots, c_{l,s,i}^u) = IdC_{e_{k_s}}^\lambda, \text{ onde } l \leq n,$$

Finalmente, será considerada a última possibilidade.

iv.  $k_r \neq k_s$  e  $e_k^v \neq e_k^q, v \neq q$

Seja  $t_{j,i}$  um bloco qualquer de um texto claro. Por hipótese tem-se que  $k_r \neq k_s$  e  $e_k^v \neq e_k^q$ , então será mostrado que  $IdC_{e_{k_r}^v} \neq IdC_{e_{k_s}^q}$ . Assim:

Como  $k_r \neq k_s$  então, pelo mesmo argumento mostrado em “iii”, tem-se que:

$(t_{j,i}, k_r) \neq (t_{j,i}, k_s)$ . Disto, como  $e_k^v \neq e_k^q$ , então, tem-se que:  $e_{k_r}^v(t_{j,i}) \neq e_{k_r}^q(t_{j,i})$  e  $e_{k_s}^v(t_{j,i}) \neq e_{k_s}^q(t_{j,i})$ .

Disto e da Proposição 6.4, tem-se que fixada uma chave  $k$  usando regras de cifração distintas, prova-se que  $IdC_{e_{k_r}^v} \neq IdC_{e_{k_r}^q}$  e  $IdC_{e_{k_s}^v} \neq IdC_{e_{k_s}^q}$ . Disto e do item “iii” acima, tem-se que para  $k_r \neq k_s$  e  $e_k^v$  fixa, tem-se que:  $IdC_{e_{k_r}^v} \neq IdC_{e_{k_r}^q}$  e  $IdC_{e_{k_s}^v} \neq IdC_{e_{k_s}^q}$ . Logo,  $e_{k_r}$  e  $e_{k_s}$  geram Identificadores de Contexto distintos para qualquer  $k_r \neq k_s$ , cq.d.

## 6.5 A VARIAÇÃO DE PARÂMETROS CRIPTOGRÁFICOS NAS CIFRAS DE BLOCO CRIA UM NÚMERO VARIADO LINGUAGENS DESCONHECIDAS

Como pode ser visto anteriormente, a metodologia desenvolvida neste trabalho para identificar cifras de bloco, utiliza regras de cifração como Identificador de Contexto de uma dada linguagem desconhecida. Baseado neste fato e nos resultados demonstrados nesta seção será apresentado, a seguir, uma proposição que estabelece uma relação entre essa linguagem desconhecida com outras, cujos aspectos léxicos, sintáticos e semânticos são conhecidos.

**Proposição 6.6:** Sejam  $K = \{k_1, k_2, \dots, k_n\}$  um conjunto de chaves criptográficas,  $T = \{t_1, t_2, \dots, t_n\}$  uma coleção de textos claros, tais que  $t_i = (t_{1,i}, t_{2,i}, \dots, t_{n,i})$  são blocos do texto claro  $t_i$ , e  $C = \{c_1, c_2, \dots, c_n\}$  uma coleção de criptogramas, tais que  $c_i = (c_{1,i}, c_{2,i}, \dots, c_{n,i})$  são blocos do criptograma  $c_i$ , onde:

$$\text{i. } |t_{1,i}| = |t_{2,i}| = \dots = |t_{n,i}| \text{ e } \sum_{j=1}^n |t_{j,i}| = |t_i|; \text{ e}$$

$$\text{ii. } |c_{1,i}| = |c_{2,i}| = \dots = |c_{n,i}| \text{ e } \sum_{l=1}^n |c_{l,i}| = |c_i|.$$

Se  $L = (\psi_1, \psi_2, \dots, \psi_n)$  é uma linguagem conhecida sobre  $\Sigma = \{0,1\}^n$  tais que  $|\psi_i| = |t_{j,i}|$ ,

$\psi_i \in \Sigma^n$  e  $e_k^s \in \bigcup_{\lambda=1}^m e_k^\lambda = \{e_k^1, e_k^2, \dots, e_k^m\}$ , uma regra de cifração no modo ECB. Então,

$e_k^s : T \rightarrow C$  gera uma linguagem desconhecida  $\bar{L} = (c_{1,i}, c_{2,i}, \dots, c_{n,i})$ ,  $|\psi_i| = |c_{j,i}|$ , sobre  $\Sigma = \{0,1\}^n$ , tais que  $c_{j,i} \in \Sigma^n$ .

**Prova 6.6:** Sejam  $L = (\psi_1, \psi_2, \dots, \psi_n)$  uma linguagem conhecida sobre  $\Sigma = \{0,1\}^n$  tais que  $|\psi_i| = |t_{j,i}|$ ,  $k \in K$ , e  $e_k^s \in \bigcup_{\lambda=1}^m e_k^\lambda = \{e_k^1, e_k^2, \dots, e_k^m\}$ ,  $s = 1, \dots, m$ .

Baseado na combinação de parâmetros utilizada nas provas das Proposições 6.4 e 6.5, supõem-se, sem perda de generalidade, as seguintes possibilidades:

i.  $k_i = k_j$ ,  $i \neq j$

Sejam  $k_1, k_2 \in K$ , tal que  $k_1 = k_2$  e  $t_{j,i} \in t_i$ . Como  $L$  é uma linguagem provida de léxico, sintaxe e semântica conhecidos, então pelas Proposições 6.4 e 6.5, existem palavras repetidas ou não de  $L$  que definem Identificadores de Contexto. Disto, tem-se da Proposição 6.5 item “i” que  $IdC_{e_{k_1}} = IdC_{e_{k_2}}$ . Disto  $e_{k_1}^s(t_{j,i}) = e_{k_2}^s(t_{j,i}) = c_{j,i}$ ,  $j = 1, \dots, n, \forall t_{j,i} \in t_i$ . Daí  $e_k^s(t_i) = c_i = (c_{1,i}, c_{2,i}, \dots, c_{n,i})$ . Então, tem-se que existe  $\bar{L} = (c_{1,i}, c_{2,i}, \dots, c_{n,i})$ , tal que  $|c_{j,i}| = |\psi_i|$  sobre  $\Sigma = \{0,1\}^n$ , de modo que  $c_{j,i} \in \Sigma^n$ , onde  $\bar{L}$  é uma linguagem desconhecida.

Considerando agora chaves  $k_i \neq k_j$ . Disto e da Proposição 6.5, item “iii”, dado  $e_k^s$  fixa para algum  $s = 1, \dots, m$ ; tem-se que  $IdC_{e_{k_1}^s} \neq IdC_{e_{k_2}^s}$ . Daí,  $e_{k_1}^s(t_{j,i}) \neq e_{k_2}^s(t_{j,i}), \forall t_{j,i} \in t_i$ . Logo,  $e_{k_1}^s(t_{j,i}) = c_{r,i} \neq e_{k_2}^s(t_{j,i}) = \bar{c}_{u,i}$ , onde  $r, u = 1, \dots, n$ . Assim, obtém-se  $\bar{L}_{k_1} = (c_{1,i}, c_{2,i}, \dots, c_{r,i}) \neq (c_{1,i}, c_{2,i}, \dots, c_{u,i}) = \bar{L}_{k_2}$ . Isto é verdade porque se  $k_1 \neq k_2$ , então  $(t_{j,i}, k_1) = (t_{j,i}, k_2), \forall t_{j,i} \in t_i$ . Como  $e_{k_i}^s$  é injetora, logo  $e_{k_1}^s(t_{j,i}) \neq e_{k_2}^s(t_{j,i})$ . Isto significa que  $\bar{L}_{k_1} \neq \bar{L}_{k_2}$  são linguagens desconhecidas e distintas, geradas por  $e_{k_1}^s$  e  $e_{k_2}^s$ , cqfd.

Baseado nos resultados desta seção verifica-se que a variação de parâmetros, como: regras de cifração, chaves criptográficas e iterações repetidas das regras de cifração aplicadas a uma coleção  $T$  de textos claros de  $L$ ; permite-nos obter um número variado de linguagens desconhecidas  $\bar{L}$ .

## 7 DISCUSSÃO

Analisando os resultados dos experimentos relatados no capítulo 5, tanto no modo ECB como no modo CBC, nota-se que os valores de revocação e precisão se mantiveram constantes, exceto na parte 1 do experimento “2” e parte 2 do experimento “3”, quando os dois modos foram considerados em uma mesma coleção e então, o valor de revocação caiu pela metade.

O experimento “1” foi bem sucedido na identificação correta das cifras atingindo os objetivos específicos “1” e “2”, propostos nesta tese, alcançando 100% de identificação no modo ECB, com criptogramas a partir de 6144 bytes.

Os experimentos “2” e “3” foram bem sucedidos em separar os grupos e em realizar a classificação, embora com valores de revocação menores do que a identificação das cifras, atingindo os objetivos específicos “3” e “7” proposto nesta tese e confirmando o cenário “1”, uma vez que pode ser visto pelos resultados, que uma combinação de parâmetros criptográficos forma um conjunto léxico próprio e único, o qual contribui para a correta identificação dos parâmetros criptográficos. Assim, a combinação gera um contexto lingüístico (assinatura) no criptograma, dado que a transformação do criptograma é única e particular para tal combinação o que permite identificar tais parâmetros por técnicas de classificação.

Ainda sobre os experimentos “2” e “3”, dado que outros modos de operação, como o CFB e o OFB, usam encadeamento de blocos nos seus processos, semelhante ao que ocorre com o modo CBC, deduz-se que a separação em grupos e a classificação nesses modos se comportaram de forma semelhante ao modo CBC. Pode-se usar a generalização apresentada na seção 6.1 de maneira análoga para demonstrar que existe uma família de modos de operação que usam encadeamento e assim, provar o comportamento análogo.

O experimento “3” (item 5.6.3) foi bem sucedido ao separar corretamente os criptogramas no modo ECB, a partir de 1, 2, 3 e 4 rodadas, atingindo o objetivo específico “4”, contribuindo com os objetivos “1”, “2” e “3” e confirmando o cenário “2”, o que induz que o mesmo padrão de repetição com o número total de rodadas também acontece com qualquer número de rodadas. Entretanto, não se fez um estudo para saber o real impacto nas transformações realizadas pelos algoritmos dada a diminuição do número de rodadas. Apesar disso, em SOTO e BASSHAM (2000) existe uma abordagem do problema da diminuição das

rodadas, sugerindo que não há impactos nas transformações dos algoritmos, exceto as já esperadas diminuição da difusão e confusão.

O experimento “4” foi bem sucedido ao separar corretamente os criptogramas no modo ECB, quando utilizando uma composição de cinco cifras iguais, com uma mesma chave ou variando com cinco chaves diferentes, uma para cada composição, atingindo o objetivo específico “5” e contribuindo com os objetivos “1”, “2”, “3”, “4” e “7” proposto na tese e confirmando, assim, o cenário “3”, demonstrando que uma ou mais composições de cifra não fortalece o processo de cifração a ponto de impedir a separação correta da cifra em grupos e a sua conseqüente identificação.

O experimento “5” foi bem sucedido ao separar corretamente os criptogramas no modo ECB, quando utilizando uma composição de cinco cifras diferentes, variando a cifra a cada composição e utilizando uma mesma chave, atingindo o objetivo específico “6” e contribuindo para validar o objetivo “7” proposto na tese e confirmando, assim, o cenário “4”, demonstrando que uma ou mais composições de cifra diferentes não fortalece o processo de cifração a ponto de impedir a separação correta da cifra em grupos e a sua conseqüente identificação e demonstrando, por exclusão, a existência de contextos lingüísticos (assinaturas) nos criptogramas, já que se os mesmos não existissem, as composições subseqüentes a primeira não permitiriam a correta separação em grupos das cifras utilizadas nessas composições subseqüentes. Outra demonstração deste experimento é que as propriedades intrínsecas dos modelos matemáticos das cifras influenciam na identificação correta das mesmas, conforme discutido na seção 3.9.6 e no capítulo 6 (Resultados Teóricos). Assim, a cada composição, uma cifra sobrepõe o contexto lingüístico da cifra utilizada na composição anterior, pelas transformações intrínsecas de suas operações matemáticas.

O experimento “6” demonstra que em uma composição de cifras de blocos com quatro cifrações no modo CBC não deixam assinaturas que possam ser usadas por uma última cifração no modo ECB; e que em uma composição de cifras de blocos com quatro cifrações no modo ECB deixam assinaturas, mas estas são destruídas por uma última cifração no modo CBC.

O experimento “7” foi bem sucedido ao verificar que é possível ter em um mesmo conjunto, criptogramas e textos claros e submetê-los ao um processo único de agrupamento e classificação de tal forma que os criptogramas gerados pela mesma chave – e somente eles – fiquem agrupados no mesmo conjunto, da mesma forma que os textos claros de um mesmo idioma e alfabeto, contribuindo para validar o objetivo “8” proposto na presente tese. Por

esse processo, verifica-se a equivalência entre os processos de agrupamento de textos e criptogramas. Assim, as mesmas técnicas para agrupar e classificar textos podem ser utilizadas para agrupar e classificar criptogramas. Desta forma, pode-se considerar estas técnicas para identificar cifras, por meio de agrupamento e classificação de criptogramas.

Dos experimentos, percebe-se que a identificação das cifras é possível por meio de um processo de classificação, o qual tem seu percentual de acerto relacionado ao processo de separação em grupos: quanto melhor a separação, maior a taxa de acerto na identificação. As melhores separações ocorrem para os maiores tamanhos de criptogramas, uma vez que os blocos se repetem mais nos tamanhos de textos maiores, como se observa na tabela 5.33.

A constância dos valores de revocação e precisão se deve ao padrão de repetição dos blocos ao longo da coleção de criptogramas, como se observa na tabela 5.33, onde é mostrado que o padrão de repetição de blocos dos textos claros permanece idêntico nos criptogramas.

As Proposições, e suas respectivas provas, apresentadas no capítulo 6 (Resultados Teóricos), concretizam os objetivos “7”, “8”, “9” e “10” propostos nesta tese e contribuem na demonstração dos demais objetivos.

## 7.1 TRABALHOS FUTUROS

Como sugestões para trabalhos futuros, os resultados apresentados indicaram que os algoritmos estudados nesta tese não são estruturas algébricas isomórficas. Contudo cabe um estudo formal deste fato e a conseqüente prova. Recomenda-se, então como trabalho futuro esse estudo com a finalidade de provar o não isomorfismo.

Outro trabalho no mesmo campo sugerido acima é realizar experimentos semelhantes aos deste trabalho, para cifras de bloco que apresentam estruturas algébricas já identificadas, como: RSA<sup>22</sup> (Anel), Curvas Elípticas (Grupo Aditivo) e ElGamal (Grupo Multiplicativo), visando validar a metodologia proposta para cifras que apresentam modelos matemáticos isomorfos. Como exemplo, um dos objetivos nesse caso seria identificar algoritmos a partir de criptogramas gerados por duas ou mais Curvas Elípticas isomórficas.

Da mesma forma que a composição de uma ou mais cifras de blocos não impediu a identificação das mesmas, é possível que o aumento no número de rodadas de uma cifra não

---

<sup>22</sup> Em SOUZA (2007) e SOUZA *et al* (2008) foram realizados experimentos com o RSA, os quais incluíram os algoritmos DES e AES. Os experimentos tiveram bons resultados de Revocação e Precisão.



impeça e nem enfraqueça a identificação. Contudo, recomenda-se que se façam experimentos em trabalhos futuros dado que os princípios de projeto de cifras de blocos indicam que quanto maior o número de rodadas da cifra, mais difícil fica para se realizar um ataque contra esta cifra (STALLINGS, 2010). De fato, pode-se ver no projeto do algoritmo RC6, que versões deste algoritmo com poucas rodadas são suscetíveis aos ataques de criptoanálise diferencial e linear (RIVEST *et al*, 1998). Neste ataques, no caso do RC6, a quantidade de textos necessários para realizar a criptoanálise diferencial é de  $2^{56}$  com oito rodadas e de  $2^{238}$  com 20 rodadas e para realizar a criptoanálise linear é de  $2^{47}$  com oito rodadas e de  $2^{155}$  com 20 rodadas (CONTINI *et al*, 1998). Alerta-se que antes de realizar o aumento arbitrário do número de rodadas de qualquer cifra é necessário realizar estudo sobre o projeto de cada cifra considerada para tal aumento, já que em algumas cifras o número de rodadas está ligado ao tamanho da chave ou é dependente da mesma.

## 8 CONCLUSÕES

O trabalho propõe uma metodologia para identificar  $n$ -cifras de blocos e comprova o seu sucesso com um caso particular de  $n = 5$ , identificando corretamente as cifras de blocos MARS, RC6, Rijndael, Serpent e Twofish a partir de contextos lingüísticos detectados sobre um conjunto de criptogramas cifrados por elas e apresentando a prova matemática de que a metodologia é válida para qualquer valor de  $n$ .

Os experimentos demonstram que a correta identificação dos algoritmos depende da correta separação dos criptogramas em grupos de algoritmos, onde o sucesso da separação é dado pelas medidas precisão e revocação. Demonstram, também, que a identificação correta das cifras é influenciada pelas propriedades intrínsecas dos modelos matemáticos das cifras, dado que cada uma realiza as suas transformações de uma maneira particular.

Demonstrou-se no presente trabalho que qualquer método, baseado no conjunto léxico de uma linguagem, usado para agrupar e classificar textos claros pode agrupar e classificar criptogramas, que cada cifra e chave criptográfica geram um identificador de contexto lingüístico, o qual permite identificar essa cifra ou chave e que a variação de parâmetros criptográficos nas cifras de bloco cria um número variado linguagens desconhecidas, com seus respectivos contextos lingüísticos. Assim, cada combinação de parâmetros criptográficos gera um contexto lingüístico, que pode ser considerado uma assinatura no criptograma, o qual permite identificar tais parâmetros por técnicas de classificação.

Os resultados teóricos provam formalmente os experimentos e concretizam os objetivos propostos na presente tese.

Concluí-se que uma coleção de criptogramas no modo ECB, pode ser corretamente identificada com 100% de acerto, desde que tais criptogramas tenham pelo menos 6144 bytes. Além disso, concluí-se que variando um parâmetro criptográfico e fixando os demais, é possível classificar ou identificar o parâmetro variante.

As principais contribuições deste trabalho são:

- 1) Demonstração de uma metodologia de identificação para  $n$ -cifras de bloco;
- 2) Identificação de cifras de blocos para um caso particular de  $n = 5$ ;
- 3) Demonstração da existência de contextos lingüísticos (assinatura) nos criptogramas;

- 4) Demonstração de cada combinação de parâmetros criptográficos gera um contexto lingüístico no criptograma, a qual permite identificar tais parâmetros por técnicas de classificação;
- 5) Demonstração de que variando um parâmetro criptográfico e fixando os demais, é possível classificar ou identificar o parâmetro variante;
- 6) Demonstração da influência de propriedades intrínsecas dos modelos matemáticos das cifras na identificação correta das mesmas;
- 7) Demonstração de qualquer método, baseado no conjunto léxico de uma linguagem, usado para agrupar e classificar textos claros pode agrupar e classificar criptogramas;
- 8) Demonstração de que cada cifra gera um identificador de contexto lingüístico;
- 9) Demonstração de que cada chave criptográfica gera um identificador de contexto lingüístico; e
- 10) Estabelecimentos de fundamentos, proposições e provas formais para o campo de pesquisa de identificação de cifras de blocos.

## 9 REFERÊNCIAS BIBLIOGRÁFICAS

- ALBASSAL, A.M.B and WAHDAN, A.M. **Neural network based cryptanalysis of a feistel type block cipher**. In: Proceedings 2004 International Conference on Electrical, Electronic and Computer Engineering, 2004, pp. 231 – 237.
- ANDERSON, R; BIHAM, E; e KNUDSEN, L. **Serpent**: a proposal for the advanced encryption Standard. 1999.
- BARROS NETO, B; SCARMINIO, I. S; BRUNS, R. E. **Como fazer experimentos: pesquisa e desenvolvimento na ciência e na indústria**. 4 ed. Porto Alegre: Bookman, 2010.
- BIBLE. **Bible in basic english**. Disponível em: <http://www.o-bible.com/bbe.html> [capturado 13 dez. 2005].
- BIBLE. **The unbound bible**. Disponível em: <http://unbound.biola.edu> [capturado 13 abr. 2009].
- BIHAM, E; SHAMIR A. **Differential cryptanalysis of DES-like cryptosystems**. Journal of Cryptology, 1991, 4(1): 3 – 72.
- BURWICK, C, COPPERSMITH, D. *et al.* **MARS – a candidate cipher for AES**. IBM corporation. 1999.
- CARVALHO, Carlos A. B. de. **O uso de técnicas de recuperação de informações em criptoanálise**. 2006. 75 f. Dissertação (Mestrado em Sistemas e Computação) – Instituto Militar de Engenharia, Rio de Janeiro. Disponível em: <http://www.des.ime.br/dissertacoes.htm> [capturado 05 ago. 2006].
- 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.
- CANNIÈRE, C; BIRYUKOV, A; PRENEEL, B. **An Introduction to Block Cipher Cryptanalysis**. Proceedings of IEEE, vol. 94, nº 2, pp. 346 – 356, February, 2006.
- CHANDRA, G. **Classification of Modern Ciphers**. 2002. 49 f. Thesis (Master of Technology) – Indian Institute of Technology Kanpur, Kanpur, Índia. Disponível em: <http://www.security.iitk.ac.in/contents/projects/cryptanalysis/repository/girish.pdf> [capturado 05 out. 2009].
- CONTINI, S; RIVEST, R; ROBSHAW, M.J.B.; and YIN, Y. L. (1998). **The Security of the RC6 Block Cipher**. Disponível em: <http://www.rsa.com/rsalabs> [capturado 01 fev. 2010].
- DAEMEN, J. and RIJMEN V. **AES proposal: Rijndael**. 1999.

- DAEMEN, J. and RIJMEN V. **The design of Rijndael: AES – the Advanced Encryption Standard**. New York: Springer, 2002.
- DEFESA. **Estratégia Nacional de Defesa**. Disponível em: <http://www.defesa.gov.br> [capturado 1 out. 2009].
- DIEHARD. **The Marsaglia random number CDROM including the Diehard battery of tests of randomness**. Disponível: <http://www.stat.fsu.edu/pub/diehard> [capturado 10 abr. 2009].
- DILEEP, A. D. and SEKHAR, C. C. (2006). **Identification of block ciphers using support vector machines**. In International Joint Conference on Neural Networks (IJCNN 2006), Vancouver, BC, Canada, July 2006, pp. 2696-2701.
- DILEEP, A. D; SWAPNA, S; SEKHAR, C. C; KANT, S; SAXENA, P.K. **Decryption of Feistel Type Block Ciphers using Hetero-Association Model**. In: Proceedings XIV National Conference on Communications, Mumbai, India, 2008.
- EFF (ELECTRONIC FRONTIER FOUNDATION). **Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design**. 1998.
- FERGUSON, N; SCHNEIER, B; and KOHNO, T. **Cryptography engineering: design principles and practical applications**. Indianapolis: Wiley, 2010.
- FRAKES, William B. Introduction to information storage and retrieval system. In: FRAKES, William B, YATES, Ricardo B. **Information retrieval: data structures and algorithms**. Upper Saddle River: Prentice Hall, 1992. p. 1-12.
- FUNG, B. C. M; Wang, K; Ester M. **Hierarchical document clustering using frequent itemsets**. Proceedings of the SIAM International Conference on Data Mining, (**SDM 2003**), May 2003, San Francisco, CA.
- GERSTING, J. L. **Fundamentos matemáticos para a ciência da computação**. 4 ed. Rio de Janeiro: LTC, 2001.
- HARMAN, Donna. Ranking algorithms. In: FRAKES, William B, YATES, Ricardo B. **Information retrieval: data structures and algorithms**. Upper Saddle River: Prentice Hall, 1992. p. 363-392.
- HOPCROFT, J. E; ULLMAN, J. D. e MOTWANI, R. **Introdução à teoria de autômatos, linguagens e computação**. 2 ed. Rio de Janeiro: Campus, 2001.
- ISOGAI, N.; MATSUNAKA, T. and MIYAJI, A. (2003). **Optimized  $\chi^2$ -Cryptanalysis against RC6**. Proceedings of the 7th International Workshop on Fast Software Encryption.
- JAIN, A. K; MURTY, M. N; FLYNN, P. J. **Data Clustering: A Review**. ACM Computing Surveys, Vol. 31, No 3, Setember 1999. p. 264-323.

- JURAFSKY, D. and MARTIN, J. H. **Speech and language processing**: An introduction to natural language processing, computational linguistics, and speech recognition. 2 ed. Upper Saddle River: Pearson, 2009.
- KAHN, D. **The codebreakers**: the story of secret writing. New York: Macmillan Publishing, 1967.
- KANT, S; SHARMA, V; and DASS, B.K. **On Recognition of Cipher Bit Stream from Different Sources Using Majority Voting Fusion Rule**. In: Ratio Mathematica, number 15, 2005, 90 – 111. Disponível: [http://www.apav.it/sito\\_ratio/indice\\_ratio\\_15.htm](http://www.apav.it/sito_ratio/indice_ratio_15.htm) [capturado 05 abr. 2009].
- KNUDSEN, L.R. and MEIER, W. (2000). **Correlations in RC6 with a Reduced Number of Rounds**. Proceedings of the 7th International Workshop on Fast Software Encryption.
- KOHONEN, T. **Self-organizing maps**. 3 ed. New York: Springer, 2001.
- LAMBERT, J. A. **Cifrador simétrico de blocos**: projeto e avaliação. 2004. 353 f. Dissertação (Mestrado em Sistemas e Computação) – Instituto Militar de Engenharia, Rio de Janeiro.
- LASKARI, E.C; MELETIOU, G.C; STAMATIOU, Y.C; VRAHATIS, M.N. **Cryptography and Cryptanalysis through Computational Intelligence**. Computational Intelligence in Information Assurance and Security. Studies in Computational Intelligence, 2007, Volume 57, 1-49.
- LEWIS, H. R. e PAPADIMITRIOU, C. H. **Elementos de teoria da computação**. 2 ed. Porto Alegre: Bookman, 2000.
- LIPSCHUTZ, S. e LIPSON, M. **Matemática discreta**. 2 ed. Porto Alegre: Bookman, 2004.
- MAHESHWARI, P. **Classification of Ciphers**. 2001. 28 f. Thesis (Master of Technology) – Indian Institute of Technology Kanpur, Kanpur, Índia. Disponível em: <http://www.security.iitk.ac.in/contents/projects/cryptanalysis/repository/pooja.ps> [Capturado 05 out. 2009].
- MANNING, Christopher D; SCHÜTZE, Hinrich. **Foundations of statistical natural language processing**. Massachusetts: MIT Press, 2003.
- MATSUI, M. **Linear cryptanalysis method for DES cipher**. In: Advances in Cryptology – Eurocrypt 1993, volume 765, LNCS, Journal of Cryptology, 1993, 386 – 397. Springer-Verlag.
- MENEZES, Alfred J; OORSCHOT, Paul C. Van; VANSTONE, Scott A. **Handbook of applied cryptography**. Boca Raton: CRC Press, 1996.
- MENEZES, P. F. B. **Linguagens formais e autômatos**. 4 ed. Porto Alegre: Instituto de Informática da UFRGS: Sagra Luzzatto: 2002.

- NAGIREDDY, Sreenivasulu. **A Pattern Recognition Approach to Block Cipher Identification**. 2008. 85 f. Thesis (Master of Science – By research) – Indian Institute of Technology Madras, Madras, Índia. Disponível em: [http://lantana.tenet.res.in/website\\_files/thesis/MS/sreenivasuluNR\\_thesis.pdf](http://lantana.tenet.res.in/website_files/thesis/MS/sreenivasuluNR_thesis.pdf) [capturado 05 out. 2009].
- NESSIE (NEW EUROPEAN SCHEMES FOR SIGNATURES, INTEGRITY, AND ENCRYPTION). **List of General NESSIE Test Tools**. 1999.
- NIST (NATIONAL INSTITUTE OF STANDARD AND TECNOLOGY). **Federal Information Processing Standard, publication 81 (FIPS 81): DES Modes of Operations**. Washington D.C., 1980.
- NIST (NATIONAL INSTITUTE OF STANDARD AND TECNOLOGY). **Federal Information Processing Standard, publication 46-3 (FIPS 46-3): Data Encryption Standard (DES)**. Washington D.C., 1999.
- NIST<sup>A</sup> (NATIONAL INSTITUTE OF STANDARD AND TECNOLOGY). **Federal Information Processing Standard, publication 197 (FIPS 197): Announcing the advanced encryption standard (AES)**. Washington D.C., 2001.
- NIST<sup>B</sup> (NATIONAL INSTITUTE OF STANDARD AND TECNOLOGY). **Recommendation for Block Cipher Modes of Operation: Methods and Techniques**. NIST Special Publication 800-38A. Revision 1. Washington D.C., 2001.
- NIST (NATIONAL INSTITUTE OF STANDARD AND TECNOLOGY). **A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications**. NIST Special Publication 800-22. Revision 1. Washington D.C., 2008.
- OLIVEIRA, C. M. G. M; XEXEO, J. A. M; CARVALHO, C. A. B. **Clustering and Categorization Applied to Cryptanalysis**. *Cryptologia*, v. 30, p. 266-280, 2006.
- RAO, M. B. **Classification of RSA and IDEA Ciphers**. 2003. 31 f. Thesis (Master of Technology) – Indian Institute of Technology Kanpur, Kanpur, Índia. Disponível em: <http://www.security.iitk.ac.in/contents/projects/cryptanalysis/repository/anoopjain.pdf> [capturado 05 out. 2009].
- RAO, K.V.S; KRISHNA, M.R; and BABU, D.B. **Cryptanalysis of a Feistel Type Block Cipher by Feed Forward Neural Network Using Right Sigmoidal Signals**. In: *International Journal of Soft Computing*, volume 4, issue 3, 2009, 131 – 135.
- RASMUSSEN, Edie. Clustering algorithms. In: FRAKES, William B, YATES, Ricardo B. **Information retrieval: data structures and algorithms**. Upper Saddle River: Prentice Hall, 1992. p. 419-442.
- RIVEST, R; SHAMIR, A; ADLEMAN, L. M. (1978). **A Method for Obtaining Digital Signatures and Public-Key Cryptosystems**. *Communications of ACM*, v. 21, n. 2, Feb 1978, pp. 120-126.

- RIVEST, R; SHAMIR, A; ADLEMAN, L. M. (1979). **On Digital Signatures and Public-Key Cryptosystems**. MIT Laboratory for Computer Science, Technical Report MIT/LCS/TR-212, Jan 1979.
- RIVEST, R; SHAMIR, A; ADLEMAN, L. M. (1983). **Cryptographic Communications System and Method**. U.S. Patent #4.405.829, 20 Sep 1983.
- RIVEST, R; ROBSHAW, M.J.B.; SIDNEY, R. e YIN, Y. L. (1998). **The RC6 Block Cipher**. Disponível em: <http://www.rsa.com/rsalabs> [capturado 01 fev. 2010].
- SANTOS, K. C. (2009). **Introdução ao Ataque de Distinção usando a Técnica  $\chi^2$  de Criptoanálise**. 17 f. Monografias em Sistemas e Computação – Instituto Militar de Engenharia, Rio de Janeiro. Nº 01/2009. ISSN 1982-9035. Disponível em: <http://www.comp.ime.eb.br/techreports/index.html>
- SAXENA, G. **Classification of Ciphers Using Machine Learning**. 2008. 50 f. Thesis (Master of Technology) – Indian Institute of Technology Kanpur, Índia. Disponível em: [http://www.security.iitk.ac.in/contents/publications/more/ciphers\\_machine\\_learning.pdf](http://www.security.iitk.ac.in/contents/publications/more/ciphers_machine_learning.pdf) [capturado 05 out. 2009].
- SCHEINERMAN, E. R. **Matemática discreta: uma introdução**. São Paulo: Pioneira Thomson Learning, 2003.
- SCHNEIER, B. **Applied cryptography: protocols, algorithms and source code in C**. 2 ed. Massachussets: Addison-Wesley Publishing Company Reading, 1996.
- SCHNEIER, B; KELSEY, J; WHITING, D. *et al.* **Twofish: a 128-bit block cipher**. 1998.
- SCHNEIER, B. **A Self-Study Course in Block-Cipher Cryptanalysis**. Cryptologia, ISSN 0161-1194, v. 24, issue 1, p. 18-33, 2000.
- SHANNON, C. E. **A Mathematical Theory of Communication**. Bell System Technical Journal, v. 27, n.4, 1948, p. 379-423.
- SHANNON, C. E. **Communication Theory of Secrecy Systems**. Bell System Technical Journal, v. 28, n.4, 1949, p. 656-715.
- SINGH, S. **O livro dos códigos**. 3 ed. Rio de Janeiro: Record, 2003.
- SOTO, J. (1999). **Randomness Testing of the Advanced Encryption Standard Candidates Algorithms**. NIST Internal Report. NIST IR 6390. Disponível em: <http://csrc.nist.gov/publications/nistir/ir6390.pdf> [capturado 10 ago. 2008].
- SOTO, J. and BASSHAM, L. (2000). **Randomness Testing of the Advanced Encryption Standard Finalist Candidates**. NIST Internal Report. NIST IR 6483. Disponível em: <http://csrc.nist.gov/publications/nistir/ir6483.pdf> [capturado 10 ago. 2008].
- SOUZA, W. A. R. (2007). **Identificação de padrões em criptogramas usando técnicas de classificação de textos**. 2007. 252 f. Dissertação (Mestrado em Sistemas e Computação) –



Instituto Militar de Engenharia, Rio de Janeiro. Disponível em:  
<http://www.des.ime.eb.br/dissertacoes.htm>.

SOUZA, W. A. R; XEXÉO, J. A. M; OLIVEIRA, C. M. G. M. (2008). **Método de Agrupamento de Criptogramas em Função das Chaves de Cifrar**. In: IV Workshop em Algoritmos e Aplicações de Mineração de Dados (SBBD/SBES), Campinas.

STALLINGS, W. **Cryptography and network security: principles and practice**. 5 ed. Upper Saddle River: Prantice Hall, 2010.

STINSON, D. R. **Cryptography: theory and practice**. 3th ed. Boca Raton: CRC, 2006.

VAUDENAY, S. (1996). **An Experiment on DES Statistical Cryptanalysis**. ACM Conference on Computer and Communications Security, pages 139-147.

LAMBERT, J. A; XEXÉO, J. A. M; PAZ DE LIMA, A. **Rijndael: um algoritmo no estilo DES**. 2003. Relatório Técnico nº RT084/DE-9/FEV03 (Mestrado em Sistemas e Computação) – Instituto Militar de Engenharia, Rio de Janeiro.

YANG, Y; LIU, X. **A re-examination of text categorization methods**. In Proceedings of SIGIR'99, p. 42-49, 1999.

YATES, R. B; RIBEIRO NETO, B. **Modern information retrieval**. New York: addison Wesley, 1999.

**10**      **APÊNDICES**

## 10.1 APÊNDICE 1: FORMATOS DAS CHAVES CRIPTOGRÁFICAS

Os conjuntos de chaves abaixo foram utilizados nos experimentos e estão representados na base hexadecimal e binária. As chaves criptográficas foram criadas a partir de gerador definido em CARVALHO (1999).

### 10.1.1 CONJUNTO DE CHAVES 1: FORMATO HEXADECIMAL

As chaves de 1 a 5 foram utilizadas nos experimentos cuja representação de chaves se dá por  $k_i$ , conforme abaixo:

$$k_1 = 5D6F695C8ECAD0FA7BF0D5C9B975C9A2;$$

$$k_2 = 7B9AB9DD1AD5FC6E961271CB8D47C340;$$

$$k_3 = 2D1D52D74C90CA351AA9341D3521B332;$$

$$k_4 = 4DADE42FAB9181B736C5A745CE7A4B4A; e$$

$$k_5 = A31BFBB1FA916B2C6BDCECC1A4843851.$$

A chave  $k$  citada na presente e igual à chave  $k_i$  descrita nesta seção.

### 10.1.2 CONJUNTO DE CHAVES 1: FORMATO BINÁRIO

As chaves de 1 a 5 foram utilizadas nos experimentos cuja representação de chaves se dá por  $k_i$ , conforme abaixo:

$$k_1 = 0101110101101111011010010101110010001110110010101101000011111010011110111110000110101011100100110111001011101011100100110100010;$$

$$k_2 = 01111011100110101011100111011101000110101101010111111100011011101001011000010010011100011100101110001101010001111100001101000000;$$

$$k_3 = 00101101000111010101001011010111010011001001000011001010001101010001101010001101010001001101000001110100110101001000011011001100110010;$$

$$k_4 = 01001101101011011110010000101111101010111001000110000001101101110011011011000101101001110100010111001110011110100100101101001010;$$
 e

$k_5 = 1010001100011011111101110110001111110101001000101101011001011000110110111001110011001100000110100100100001000011100001010001.$

A chave  $k$  citada na presente e igual à chave  $k_1$  descrita nesta seção.

### 10.1.3 CONJUNTO DE CHAVES 2: FORMATO HEXADECIMAL

As chaves de 1 a 30 foram utilizadas nos experimentos cuja representação de chaves se dá por  $k_{i..j}$ , conforme abaixo:

$k_1 = 5356A3ECEF00744E900CC05BD72AC666;$   
 $k_2 = DC2CC20E22A7F511CFFF4E6534235F75;$   
 $k_3 = 4897F8C61744768011A427B169D24130;$   
 $k_4 = 4CA01A95A2444BA874A8C5CF3AA999B3;$   
 $k_5 = 28CD21C796DB6C50FDE88C41322A1A60;$   
 $k_6 = 61F5261C0ABFE8B3ED5953EFFF8315A3;$   
 $k_7 = 6E3CC3787522A604D8E261CACDD56EBB;$   
 $k_8 = 8A61B880EC007CEF7EC954468A49F915;$   
 $k_9 = 04522A65AC7CA096C7911DF3B6D9992F;$   
 $k_{10} = 17A6C9D5FBCCD6986908C889D6339102;$   
 $k_{11} = 4E12AC52B335F5936F91A6C3FFF0682A;$   
 $k_{12} = 4988C6874C7C2F474A8B6FBD51A78408;$   
 $k_{13} = D7305A27F96301A8BCA2F436C6FFEDCF;$   
 $k_{14} = 8576A5A498D84B199B8A0C299469222D;$   
 $k_{15} = A8B67ED5A00847890AC8FD5607B3004C;$   
 $k_{16} = D25B315072A54CCD90CFC2505AB7B049;$   
 $k_{17} = 58E4076F94D8F34E530EB5DE6414A9E2;$   
 $k_{18} = 6A9045B3FA26F5D4D1F4D1F21E06DB03;$   
 $k_{19} = CB3CD0451707787594311194E560C4D5;$   
 $k_{20} = CBA0DD9797C7D89257245352DF289976;$

$k_{21} = 1D50AEB17957FAD3A2678C0427ADA74A$   
 $k_{22} = DD83A9C189BFABA870E63D6B8D06BA22;$   
 $k_{23} = 313146055F9E34877D6124BF50625837;$   
 $k_{24} = 0D063743E18309D9FA4CF9C54D3E7CA2;$   
 $k_{25} = 80D5CFE46F7123234314C27279CE1CBC;$   
 $k_{26} = 9A4B4898F04F631B0E878B7E59C014A7;$   
 $k_{27} = 11072FDF36DAB888F21E2968C2E5EBB7;$   
 $k_{28} = EE4BBB0AD921F23CCD181480462A11AD;$   
 $k_{29} = CCB3E7CCEEE68E4DB12581DBBB827A74; e$   
 $k_{30} = 08034A27257DCEA6400D99CE17F6C7B0.$

#### 10.1.4 CONJUNTO DE CHAVES 2: FORMATO BINÁRIO

As chaves de 1 a 30 foram utilizadas nos experimentos cuja representação de chaves se dá por  $k_{i..j}$ , conforme abaixo:

$k_1 = 0101001101010110101000111110110011101111000000001110100010011101001$   
 $000000001100110000000101101111010111001010101100011001100110;$

$k_2 = 11011100001011001100001000001110001000101010011111110101000100011100$   
 $11111111111010011100110010100110100001000110101111101110101;$

$k_3 = 0100100010010111111100011000110000101110100010001110110100000000001$   
 $000110100100001001111011000101101001110100100100000100110000;$

$k_4 = 01001100101000000001101010010101101000100100010001001011101010000111$   
 $010010101000110001011100111100111010101010011001100110110011;$

$k_5 = 00101000110011010010000111000111100101101101101101101100010100001111$   
 $110111101000100011000100000100110010001010100001101001100000;$

$k_6 = 01100001111101010010011000011100000010101011111111101000101100111110$   
 $11010101100101010011111011111111111100000110001010110100011;$

$k_7 = 0110111000111100110000110111000011101010010001010100110000001001101$   
 $100011100010011000011100101011001101110101010110111010111011;$

$k_8=1000101001100001101110001000000011101100000000001111100111011110111$   
 $111011001001010101000100011010001010010010011111100100010101;$

$k_9=00000100010100100010101001100101101011000111110010100000100101101100$   
 $011110010001000111011111001110110110110110011001100100101111;$

$k_{10}=0001011110100110110010011101010111111011110011001101011010011000011$   
 $0100100001000110010001000100111010110001100111001000100000010;$

$k_{11}=0100111000010010101011000101001010110011001101011111010110010011011$   
 $011111001000110100110110000111111111111100000110100000101010;$

$k_{12}=0100100110001000110001101000011101001100011111000010111101000111010$   
 $0101010001011011011111011110101010001101001111000010000001000;$

$k_{13}=1101011100110000010110100010011111111001011000110000000110101000101$   
 $1110010100010111101000011011011000110111111111101101110011111;$

$k_{14}=1000010101110110101001011010010010011000110110000100101100011001100$   
 $1101110001010000011000010100110010100011010010010001000101101;$

$k_{15}=101010001011011001111110110101011010000000010000100011110001001000$   
 $0101011001000111111010101011000000111101100110000000001001100;$

$k_{16}=1101001001011011001100010101000001110010101001010100110011001101100$   
 $1000011001111110000100101000001011010101101111011000001001001;$

$k_{17}=0101100011100100000001110110111110010100110110001111001101001110010$   
 $1001100001110101101011101111001100100000101001010100111100010;$

$k_{18}=0110101010010000010001011011001111111010001001101111010111010100110$   
 $1000111110100110100011111001000011110000001101101101100000011;$

$k_{19}=1100101100111100110100000100010100010111000001110111100001110101100$   
 $1010000110001000100011001010011100101011000001100010011010101;$

$k_{20}=1100101110100000110111011001011110010111110001111101100010010010010$   
 $1011100100100010100110101001011011111001010001001100101110110;$

$k_{21}=0001110101010000101011101011000101111001010101111111101011010011101$   
 $0001001100111100011000000010000100111101011011010011101001010;$

$k_{22}=1101110110000011101010011100000110001001101111111010101110101000011$   
 $1000011100110001111010110101110001101000001101011101000100010;$

$k_{23}$ =0011000100110001010001100000010101011111100111100011010010000111011  
1110101100001001001001011111101010000011000100101100000110111;

$k_{24}$ =0000110100000110001101110100001111100001100000110000100111011001111  
1101001001100111110011100010101001101001111100111110010100010;

$k_{25}$ =1000000011010101110011111110010001101111011100010010001100100011010  
00011000101001100001001110010011110011100111000011100101111100;

$k_{26}$ =1001101001001011010010001001100011110000010011110110001100011011000  
0111010000111100010110111111001011001110000000001010010100111;

$k_{27}$ =0001000100000111001011111101111100110110110110101011100010001000111  
1001000011110001010010110100011000010111001011110101110110111;

$k_{28}$ =1110111001001011101110110000101011011001001000011111001000111100110  
0110100011000000101001000000001000110001010100001000110101101;

$k_{29}$ =1100110010110011111001111100110011101110111001101000111001001101101  
1000100100101100000011101101110111011100000100111101001110100;

$k_{30}$ =0000100000000011010010100010011100100101011111011100111010100110010  
0000000001101100110011100111000010111111101101100011110110000.

## 10.2 APÊNDICE 2: AGRUPAMENTO DE CRIPTOGRAMAS POR RELAÇÕES FUZZY EQUIVALENTES

Neste apêndice será apresentado um método de agrupamento de criptogramas baseado relações fuzzy equivalentes, o qual utiliza os mesmos procedimentos descritos nesta tese (figura 4.4), exceto a fase de Agrupamento (ligação simples) e de Análise de Grupos, as quais são substituídas por operações fuzzy: Teste de Matriz, Defuzzificação, Composição e Agrupamento (fuzzy).

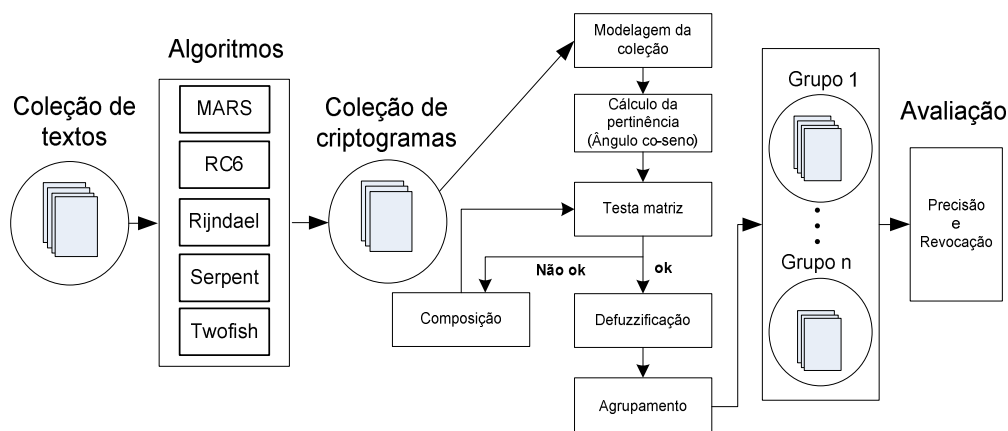


Figura 10.2.1: Esquema do agrupamento por relações fuzzy equivalentes

### 10.2.1 INTRODUÇÃO

As Relações Fuzzy mapeiam elementos de um conjunto universo  $X$  para elementos de outro conjunto universo  $Y$ , por meio do produto cartesiano  $X \times Y$ , onde a força da relação dos pares ordenados destes conjuntos é dada por uma função de pertinência, a qual pode expressar vários graus de pertinência no intervalo  $[0,1]$  (ROSS, 2004). Os graus de pertinência associados às relações são armazenados em uma matriz de relação. Uma relação Fuzzy de  $X$  para  $X$  é uma relação fuzzy equivalente se a matriz de relação respectiva atender a todas as seguintes propriedades: reflexividade, simetria e transitividade. Exemplos de agrupamento por relação fuzzy equivalente podem ser vistos em (ROSS, 2004) e (LIANG, 2005).

Neste contexto, este apêndice apresenta um novo método de agrupamento de criptogramas, baseado em relações fuzzy equivalentes, capaz de reunir, num mesmo conjunto, sem conhecimento das mensagens ou das chaves utilizadas, criptogramas gerados



pela mesma chave. Isto é, dado um conjunto de criptogramas cifrados com várias chaves, o procedimento os separa em conjuntos diferentes, de tal forma que criptogramas originados da mesma chave agrupam-se num mesmo conjunto e cada conjunto só contém criptogramas gerados com a mesma chave. No experimento, foi utilizada a base 1 de criptogramas.

### 10.2.2 DESCRIÇÃO DO PROCEDIMENTO

Assim, como nos demais experimentos desta tese, o procedimento fuzzy utilizado (Figura 10.2.1) considera os blocos dos criptogramas como palavras de um texto de idioma desconhecido e categoriza os criptogramas com base na similaridade existente entre eles, a qual é medida com base na frequência com que palavras ocorrem nos documentos.

Assim, dada uma coleção de criptogramas  $X$ , esta coleção é mapeada em uma relação de  $X$  para  $X$  onde uma função de pertinência calcula os valores relacionados aos graus de associação entre os pares ordenados e os armazena em uma matriz de relações. Tal matriz é testada para o atendimento das propriedades de reflexividade, simetria e transitividade (a ser comentada a frente). Em caso negativo, uma ou mais operações de composição podem ser realizadas sobre a matriz para que a mesma adquira as propriedades citadas. Após a determinação de um corte- $\lambda$  apropriado é gerada uma matriz defuzzificada, a partir da matriz de relações, onde o valor “1” determina a pertinência de um par ordenado a um grupo. Por fim, a qualidade da categorização é avaliada pelas medidas de revocação e precisão (YATES e RIBEIRO NETO, 1999) (FUNG, 2003).

### 10.2.3 CÁLCULO DA PERTINÊNCIA

Nas relações fuzzy, a força da relação entre os pares ordenados é dada por uma função de pertinência  $\mu_{\tilde{R}}(x, y)$ , limitada a valores no intervalo  $[0,1]$ . Assim, podemos associar a função  $\mu_{\tilde{R}}(x, y)$  ao co-seno do ângulo (HARMAN, 1992) (ROSS, 2005) de dois vetores representantes dos criptogramas  $c_i$  e  $c_j$ , calculado pela fórmula (4.1). Quanto maior o valor de  $\mu$ , maior o grau de pertinência do par  $c_i$  e  $c_j$  a um conjunto. Desta forma, constrói-se uma matriz de relações (semelhante à matriz de similaridades) armazenando, em suas células, os valores de pertinência dos pares de criptogramas contidos na coleção de criptogramas. Os números que rotulam as linhas e as colunas representam os criptogramas da coleção.

#### 10.2.4 TESTE DAS PROPRIEDADES DA MATRIZ DE RELAÇÕES

Para realizar o agrupamento como descrito neste método, é necessário que a matriz de relações seja testada antes do processo de defuzzificação para verificar se a mesma atende a determinadas propriedades.

Assim, a relação Fuzzy de  $X$  para  $X$  é uma relação equivalente se a matriz de relação respectiva a tal relação atender a todas as seguintes propriedades:

i. Reflexividade:  $\mu_{\tilde{R}}(x_i, x_i) = 1$ ;

ii. Simetria:  $\mu_{\tilde{R}}(x_i, x_j) = \mu_{\tilde{R}}(x_j, x_i)$ ;

iii. Transitividade:  $\mu_{\tilde{R}}(x_i, x_j) = \lambda_1$  e  $\mu_{\tilde{R}}(x_j, x_k) = \lambda_2 \rightarrow \mu_{\tilde{R}}(x_i, x_k) = \lambda$ , onde

$$\lambda \geq \min(\lambda_1, \lambda_2).$$

A matriz que atende somente as propriedades “i” e “ii” não é transitiva e representa uma relação de proximidade e não de equivalência. Neste caso, podem-se aplicar operações de composição na tentativa de torná-la transitiva.

A composição é descrita como: Dada uma relação  $\tilde{R} = (x, y)$  e uma relação  $\tilde{S} = (y, z)$ , encontrar uma relação  $\tilde{T} = (x, z)$ , a partir de  $\tilde{T} = \tilde{R} \circ \tilde{S}$ . Para esta tarefa, pode-se utilizar uma operação de composição do tipo max–min, a qual é dada por

$$\chi_{\tilde{T}}(x, z) = \bigvee_{y \in Y} (x_{\tilde{R}}(x, y) \wedge x_{\tilde{S}}(y, z)).$$

O número máximo de composições é dado por  $\tilde{R}^{n-1} = \tilde{R} \circ \tilde{R} \circ \dots \circ \tilde{R} = \tilde{R}$ , onde  $n$  é o número de criptogramas em uma coleção.

#### 10.2.5 DEFUZZIFICAÇÃO

A defuzzificação é obtida por meio do parâmetro corte- $\lambda$ , no qual se utiliza um valor próximo de zero, já que é pouco provável a repetição de blocos ao longo dos criptogramas. Desta forma, para um corte- $\lambda = 0,001$  toda e qualquer célula da matriz de relação com valor igual ou maior que 0,001 será sobreposta com o valor 1. As demais serão sobrepostas com o valor 0. A defuzzificação gera uma matriz defuzzificada.

## 10.2.6 AGRUPAMENTO

O agrupamento consiste em buscar cada uma das células da matriz de relações com valor igual a “1”, linha a linha, de forma que cada linha forma um grupo. Como os números que rotulam as linhas e as colunas representam os criptogramas da coleção, a cada linha, obtém-se o número da linha e o número da coluna da célula da matriz que contém valor igual a “1” e atribuem-se os criptogramas, que são representados por estas linha e coluna, ao grupo da linha sendo processada.

## 10.2.7 EXPERIMENTO, RESULTADO E AVALIAÇÃO

O objetivo do experimento é separar os criptogramas em cinco grupos de maneira que cada grupo contenha criptogramas gerados pela mesma cifra e somente por ela. Este experimento utiliza a base 1 de criptogramas. Para cada modo de operação: ECB e CBC; são definidas nove coleções, cada uma com determinado tamanho. Cada coleção é composta por 150 criptogramas, gerados por cinco algoritmos: MARS, RC6, Rijndael, Serpent e Twofish, a partir dos mesmos 30 textos em claro e uma única chave aleatória  $k$  de 128 *bits* para cada modo, portanto, o experimento utiliza 1350 criptogramas.

Tabela 10.2.1 Resultado da separação em grupos

Tamanho dos criptogramas	Agrupamento Hierárquico Aglomerativo				Agrupamento Fuzzy por relações equivalentes			
	ECB		CBC		ECB		CBC	
	P	R	P	R	P	R	P	R
1024	1	0,17	1	0,03	1	0,17	1	0,03
1536	1	0,20	1	0,03	1	0,20	1	0,03
2048	1	0,33	1	0,03	1	0,23	1	0,03
2560	1	0,50	1	0,03	1	0,37	1	0,03
3072	1	0,87	1	0,03	1	0,30	1	0,03
4096	1	0,97	1	0,03	1	0,30	1	0,03
6144	1	1	1	0,03	1	0,77	1	0,03
8192	1	1	1	0,03	1	1	1	0,03
10240	1	1	1	0,03	1	0,70	1	0,03

Observando a tabela 10.2.1, pode-se concluir que o agrupamento ocorreu com sucesso, obtendo o valor máximo de precisão com todos os tamanhos de criptograma, ou seja, o

procedimento utilizado não agrupou criptogramas originados de cifras diferentes em um mesmo grupo em nenhum caso.

Quanto aos valores de revocação, observa-se que para alguns tamanhos de criptograma, os valores não alcançaram o valor máximo, ou seja, em alguns casos, o procedimento utilizado separou criptogramas gerados pela mesma cifra em grupos diferentes. Mas, novamente, nunca em grupos de criptogramas gerados por outra cifra.

Comparando os resultados obtidos com o método de agrupamento hierárquico aglomerativo da ligação simples com o método de agrupamento fuzzy por relações equivalentes, percebe-se que o método fuzzy teve um desempenho inferior ao método da ligação simples.

Uma possível explicação para o melhor desempenho do agrupamento hierárquico aglomerativo da ligação simples pode ser a seguinte: uma vez que o critério de parada no método da ligação simples é um valor de similaridade próximo de zero, já que é pouco provável a repetição de blocos ao longo dos criptogramas, a ligação simples é adequada, pois a dispersão gerada pelo método permite que dois criptogramas quaisquer que estejam em um mesmo grupo, possuam valor de similaridade mais baixo que a similaridade do próprio grupo. Já o método fuzzy não possui a característica da dispersão uma vez que a defuzzificação, que determinará em última instância a pertinência aos grupos, é feita par a par, antes do processo de agrupamento.

#### 10.2.8 CONCLUSÃO

Este apêndice propõe um procedimento baseado em relações fuzzy equivalentes e demonstra a sua capacidade para agrupar criptogramas, de tal forma que somente criptogramas cifrados com uma mesma chave sejam categorizados no mesmo grupo. Os experimentos demonstraram que os grupos são formados com precisão máxima, independente do algoritmo criptográfico ou do tamanho da chave utilizada. Porém, foi demonstrado que as relações fuzzy equivalentes, embora mantenham os valores de precisão iguais ao método de agrupamento hierárquico aglomerativo da ligação simples, apresenta desempenho inferior nos valores de revocação. Tal desempenho pode ser explicado pela dispersão gerada pelo método da ligação simples.