**Fortalecimento da família
de cifras FEAL**

**Uma versão mais resistente
à Criptoanálise Diferencial**

Paulo Geraldo Pinheiro

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO GRAU DE MESTRE
EM
MATEMÁTICA APLICADA

Área de Concentração: Ciência da Computação
Orientador: Prof. Dr. Routo Terada

*Durante a elaboração deste trabalho o autor recebeu apoio financeiro da CAPES
e FAPESP(proc. 93/0603-1).*

-São Paulo, 17 de fevereiro de 1994-

Fortalecimento da família de cifras FEAL

Uma versão mais resistente
à Criptoanálise Diferencial

Este exemplar corresponde à redação
final da dissertação devidamente corrigida
e defendida por Paulo Geraldo Pinheiro
e aprovada pela comissão julgadora.

São Paulo, 02 de maio de 1994.

Banca examinadora:

- Prof. Dr. Routo Terada (orientador) - MAC-IME-USP
- Prof. Dr. Siang Wun Song - MAC-IME-USP
- Prof. Dr. José Galvão Leite - MAE-IME-USP

*À Tânia,
à Maria Nanci e
aos meus pais.*

Resumo

Nesta dissertação, estudamos como o método da Criptoanálise Diferencial é aplicado à família de cifras FEAL. Essa criptoanálise é um ataque do tipo texto escolhido que pode ser convertido diretamente em um ataque do tipo texto conhecido, sob a hipótese de que uma quantidade suficiente de pares de texto legível e cifrado sejam fornecidos. Como uma forma de fortalecimento da cifra contra a Criptoanálise Diferencial, nós propomos a introdução de uma função de troca probabilística na metade direita do bloco de entrada de cada iteração de uma cifra FEAL. Chamamos a nova versão de FEAL-N/RP (FEAL-N com redução de pares). Mostramos que essa alteração anula a probabilidade das principais características dadas por Biham e Shamir e que a nova cifra é mais forte que a original com relação a um ataque diferencial baseado em qualquer característica do FEAL. Além disso, a nova versão ainda pode ser implementada eficientemente em software.

Abstract

In this dissertation we studied how the Differential Cryptanalysis method is applied to the FEAL cipher family. This cryptanalysis is a chosen plaintext attack that can be converted directly into a known plaintext attack under the assumption that enough pairs of plaintext and ciphertext are given. As a way to strengthen the cipher against Differential Cryptanalysis, we propose the introduction of a probabilistic swapping function on the right half of the input block in each iteration of a FEAL cipher. We call the new version FEAL-N/RP (FEAL-N with reduction of pairs). We show that this alteration vanishes the probability of the most important characteristics given by Biham and Shamir and that the new cipher is stronger than the original one with respect to a differential attack based on any FEAL characteristic. Furthermore, the new version can still be efficiently implemented by software.

Agradecimentos

Ao Professor Routo Terada pela ótima orientação.

À Nádia pelo constante interesse nesse trabalho.

À Vivian pelos divertidos finais de semana e por toda a força recebida na reta final.

Ao Flávio, Laís, Margarete, Maria de Fátima, Nivaldo e Andréa pelo apoio e incentivo.

À Francisca, Lilian e Walter por todos os atendimentos fora de horário.

A todos aqueles que de alguma forma contribuíram para esse trabalho.

Conteúdo

1	Introdução	1
1.1	Conceitos básicos	2
1.2	Cifras de Feistel	3
2	A família de cifras FEAL	5
2.1	Especificações do FEAL	6
2.2	Sobre a função f	13
3	Criptanálise Diferencial das cifras de Feistel	17
4	Criptanálise Diferencial da família de cifras FEAL	26
4.1	Considerações gerais	26
4.2	Um algoritmo para decidir se $(X; Y) \rightarrow Z$ ou $(X; Y) \rightsquigarrow Z$ pela função S_i .	28
4.3	Propriedades das funções S e f	40
4.4	Características para o FEAL-N(X)	44
4.5	Os ataques $1R$ e $2R$	52
4.6	Ataques diferenciais do tipo texto conhecido	55
4.7	Conclusões	56
4.8	Apêndice	57
5	Uma proposta de fortalecimento contra a Criptanálise Diferencial	60
5.1	FEAL-N(X)/RP: O FEAL-N(X) com redução de pares	60
5.2	Propriedades da função Tr	66
5.3	A utilização de características do FEAL-N(X) para o FEAL-N(X)/RP . . .	71
5.3.1	Sobre características iterativas simétricas	75
5.4	Conclusões	81
5.5	Apêndice	83
5.5.1	Descrição de X e Y tais que $p_f(X, Y) = 1$ e $Par(X) \wedge Par(Y)$. . .	83
5.5.2	Descrição de X e Y tais que $p_f(X, Y) = \frac{1}{2}$ e $Par(X) \wedge Par(Y)$. . .	83
5.5.3	Descrição de X e Y tais que $p_f(X, Y) \geq \frac{1}{4}$ e $Par(X) \wedge Par(Y) \wedge$ $Sim(Y)$	83

6 Resultados Experimentais para o FEAL-N/RP	84
6.1 Análise do número de trocas	84
6.2 Análise das características de Biham e Shamir	100
Conclusões Finais	102
Referências Bibliográficas	104

Capítulo 1

Introdução

A necessidade de manter em sigilo certas informações existe há milhares de anos. Os antigos espartanos, por exemplo, codificavam as suas mensagens militares. Para os chineses, o simples ato de escrever uma mensagem já a tornava secreta, pois poucos eram capazes de ler os seus caracteres. Naturalmente, as pessoas não demoraram muito para se darem conta das vantagens que poderiam ser obtidas pela interceptação de informações secretas, o que provocou uma batalha contínua e fascinante entre os “fabricantes” e os “quebradores” de mensagens codificadas. A arena onde ocorrem essas disputas são os meios de comunicação, os quais mudaram consideravelmente com o passar do tempo.

Os primeiros canais de comunicação foram os carteiros de confiança. A segurança de um tal canal depende tanto da confiabilidade do mensageiro quanto da sua capacidade de se safar de situações nas quais a abertura da mensagem pode ocorrer; isso sem levar em conta que a lentidão do processo bem como a pequena quantidade de informações que podem ser transmitidas dessa forma podem inviabilizar essa alternativa para muitas situações práticas.

Foi somente após a invenção do telégrafo que os meios de comunicação começaram a evoluir para a forma que atualmente conhecemos. Hoje em dia, a sociedade é fortemente dependente dos meios modernos, rápidos e precisos que existem para a transmissão de mensagens. Geralmente, o objetivo principal é meramente transmitir uma mensagem pelo meio mais rápido e barato que for possível. Existe, entretanto, um grande número de situações nas quais a informação é confidencial, e para as quais um interceptador poderia se beneficiar imensamente pelo conhecimento adquirido. Em tais situações, os comunicantes devem disfarçar o conteúdo da mensagem. A ciência que estuda o desenvolvimento de métodos para disfarçar o conteúdo de uma mensagem e analisa o seu nível de segurança é conhecida por Criptologia.

1.1 Conceitos básicos

¹Criptografia é a ciência e o estudo da escrita secreta. Uma cifra ou sistema criptográfico ou função criptográfica é um método secreto de escrita pelo qual um texto legível é transformado em texto cifrado. Esse processo de transformação é conhecido como ciframento, enquanto que o processo reverso é chamado de deciframento. Ambos os processos são controlados por uma ou mais chaves criptográficas. Dizemos que uma cifra é simétrica se a chave do processo de deciframento é a mesma utilizada no ciframento. O tamanho de bloco de uma cifra é o número de bits ou caracteres que ela pode cifrar. Quando o tamanho da mensagem é maior do que o tamanho de bloco da cifra, podemos fazer o ciframento através de aplicações do sistema criptográfico sobre trechos da mensagem. Os dois métodos mais utilizados são ECB - Electronic Code Book - e CBC - Cipher Block Chaining. O modo ECB consiste meramente em dividir a mensagem em trechos que possuem o tamanho de bloco da cifra e aplicar a função criptográfica a cada um dos trechos. Já o modo CBC utiliza o texto cifrado do trecho anterior da mensagem para obter o texto cifrado referente a um trecho. Para o primeiro trecho utiliza-se um valor inicial. A mensagem codificada é então obtida pela concatenação de cada um dos trechos cifrados.

Basicamente, existem dois tipos de cifras: transposições e substituições. Uma cifra de transposição rearranja os bits ou os caracteres do texto legível, ao passo que uma cifra de substituição troca bits, caracteres ou blocos de caracteres por outros valores. Um código é um tipo especial de cifra de substituição que utiliza um livro de código como sendo a chave. Algumas vezes, o termo código é utilizado para referenciar um tipo qualquer de cifra. Em aplicações computacionais, geralmente se utiliza uma combinação de transposições e substituições. Por exemplo, o DES[NBS77; BP82 pp. 267-285; Ko81 pp. 240-250] cifra blocos de 64 bits dessa forma.

Quebrar uma cifra significa descobrir um meio de obter o texto legível a partir de qualquer texto cifrado. Criptanálise é a ciência e o estudo de métodos para quebrar as cifras. Uma cifra é quebrável se é possível determinar o texto legível ou a chave secreta a partir do texto cifrado, ou ainda, determinar a chave a partir de pares de textos legível e cifrado. Basicamente, existem três tipos de ataque: texto cifrado conhecido, texto legível conhecido e texto legível escolhido.

Num ataque do tipo texto cifrado conhecido, o criptoanalista deve determinar a chave somente através de textos cifrados interceptados, muito embora o método de ciframento, o idioma no qual o texto legível foi escrito, o assunto e algumas palavras prováveis possam ser conhecidos. Em ataques do tipo texto legível conhecido, o criptoanalista conhece alguns pares de textos legível e cifrado e deve determinar a chave a partir deles. Hoje em dia, as cifras são consideradas aceitáveis somente se elas podem resistir a um ataque desse tipo sob a suposição de que o criptoanalista possui uma quantidade arbitrária de pares[De82].

¹As definições desta seção foram extraídas de [De82].

Num ataque do tipo texto legível escolhido, o criptoanalista é capaz de conseguir o texto cifrado correspondente ao texto legível selecionado. Esta é a situação mais favorável ao criptoanalista, a qual é pouco provável de ocorrer na prática, mas que permite comparar a segurança de duas cifras distintas.

Uma cifra é incondicionalmente segura se, independentemente da quantidade de texto cifrado interceptado, não existe informação suficiente para determinar o texto legível de forma única. Com uma única exceção, todas as cifras são quebráveis se considerarmos que os recursos são ilimitados.

Uma cifra é computacionalmente segura ou forte se ela não pode ser quebrada por análise sistemática com os recursos disponíveis.

O ramo do conhecimento que engloba a criptografia e a criptoanálise é conhecido como criptologia.

1.2 Cifras de Feistel

²Em uma cifra de Feistel o tamanho de bloco tem que ser par. Se temos um tamanho de bloco $2n$ então cada texto legível l é dividido em duas metades l_0 e l_1 . Cada chave k define um conjunto de subchaves $k_1, k_2, \dots, k_N$ para algum inteiro N fixado, e cada subchave k_i define uma transformação F_{k_i} do conjunto de blocos de tamanho n sobre ele mesmo. Qualquer texto legível l é cifrado em N iterações de acordo com a seguinte regra:

- Na i -ésima iteração, $1 \leq i < N$: $\lambda_{i-1} = (l_{i-1}, l_i) \rightarrow \lambda_i = (l_i, l_{i+1})$,
- Na última iteração: $\lambda_{N-1} = (l_{N-1}, l_N) \rightarrow \lambda_N = (l_{N+1}, l_N)$,

onde $l_{i+1} = l_{i-1} \oplus F_{k_i}(l_i)$, $1 \leq i \leq N$. O texto cifrado é então dado pelo bloco de $2n$ bits $l_{N+1}l_N$.

O deciframento de uma cifra de Feistel é muito simples, basta notarmos que $l_{i-1} = l_{i+1} \oplus F_{k_i}(l_i)$. Assim, se aplicarmos o mesmo processo utilizado no ciframento sobre o texto cifrado, mas utilizando as subchaves em ordem reversa, obteremos o texto legível.

É importante salientar que não é exigido que as funções F_{k_i} possuam qualquer propriedade especial, nem mesmo que sejam inversíveis. A figura 1.1 nos mostra a estrutura de uma cifra de Feistel.

Nos próximos capítulos, estaremos estudando a família de cifras FEAL- Fast Data Encipherment Algorithm, as quais são cifras do tipo Feistel. Veremos como o método de criptoanálise conhecido por Criptoanálise Diferencial (um método do tipo texto escolhido que possui a grande vantagem de poder ser convertido diretamente em um método do tipo texto conhecido) se aplica a essas cifras e faremos uma proposta de alteração para o FEAL com o intuito de anular as características dadas por Biham e Shamir e torná-lo mais resistente aos ataques da Criptoanálise Diferencial.

²Extraído parcialmente de [BP82]

Figura 1.1 *Estrutura de uma cifra de Feistel.*

Capítulo 2

A família de cifras FEAL

Durante o congresso Eurocrypt de 1987 foi apresentada a cifra FEAL-4 [ShMi87]. Os objetivos dos projetistas eram obter uma cifra que fosse eficientemente implementável em software e que não gerasse discussões sobre tabelas aleatórias, como as que ocorreram sobre o DES[NBS77].

O FEAL - Fast Data Encipherment Algorithm - é uma cifra do tipo Feistel que, ao contrário do DES[NBS77], não necessita fazer consultas a tabelas. É um tipo de cifra cujo procedimento de construção do texto cifrado é público e que utiliza a mesma chave para cifrar e decifrar os dados. A cifra é construída através de combinações de operações sobre bytes, tornando assim a implementação em software altamente eficiente. As operações utilizadas são:

- (i) soma módulo 256 de dois bytes;
- (ii) deslocamento circular sobre um byte e
- (iii) ou-exclusivo entre dois bytes.

Em 1988, a NTT - Nippon Telephone Telegraph - lançou uma nova versão da cifra com o dobro de iterações do FEAL-4, a qual foi denominada FEAL-8[MiShSh88]. Posteriormente, em 1990, a cifra FEAL-8 foi generalizada para FEAL-N, ou seja, FEAL com N iterações, onde N é um número maior ou igual a 4. Nessa generalização, os projetistas preocuparam-se em manter a compatibilidade entre os algoritmos. Assim, FEAL-4 coincide com FEAL-N para N=4 e FEAL-8 coincide com FEAL-N para N=8 [Mi90].

Acreditando que a busca exaustiva seja viável para o FEAL-N dentro de uma ou duas décadas devido aos avanços da tecnologia de LSI - Large Scale Integration (circuitos impressos que contém entre 10^2 e 10^5 portas)-, seus projetistas criaram o FEAL-NX, o qual é um FEAL-N que utiliza uma chave de 128 bits [Mi90].

O FEAL é um algoritmo que é eficiente tanto quando é implementado em hardware quanto em software. Por exemplo, a velocidade de processamento de um chip LSI de FEAL-8 é 96 Mbps em frequência de 12MHz. As velocidades dos programas para FEAL-N utilizando-se um microprocessador i80286 são:

- 1 Mbps para $N=4$;
- 670 Kbps para $N=8$;
- 440 Kbps para $N=16$;
- 220 Kbps para $N=32$ e
- 120 Kbps para $N=64$.

Neste capítulo, descreveremos detalhadamente essas cifras.

2.1 Especificações do FEAL

Nesta seção, vamos especificar os algoritmos da família de cifras FEAL. Para isto, nós utilizaremos as seguintes convenções:

- (i) $A, A_r, \dots$: seqüências de bits;
- (ii) $(A; B; \dots)$: concatenação de seqüências de bits nesta ordem;
- (iii) $A \oplus B$: operação ou-exclusivo¹ entre A e B ;
- (iv) ϕ : seqüência de 32 zeros;
- (v) $:=$: atribuição;

Inicialmente, vejamos como funcionam as funções utilizadas pelo FEAL.

Função S_i

É utilizada pelas funções f e f_K . Sua definição é a seguinte:

$$S_i(X_1, X_2) = \text{Rot2}((X_1 + X_2 + i) \bmod 256)$$

onde:

- X_1, X_2 são bytes;
- $i \in \{0, 1\}$;
- $\text{Rot2}(X)$ é o deslocamento circular de dois bits à esquerda sobre o byte X .

¹Se a e b são bits então $a \oplus b = 1$ se, e somente se, $a \neq b$. Se A e B são seqüências de n bits então $A \oplus B$ equivale à aplicação da operação ou-exclusivo bit a bit sobre A e B .

Função f

É utilizada na aleatorização dos dados. Sua definição é:

- Sejam $\alpha = (\alpha_0; \alpha_1; \alpha_2; \alpha_3)$ e $\beta = (\beta_0; \beta_1)$, onde os α_i , $i = 0, 1, 2, 3$, e os β_j , $j = 0, 1$, são bytes. Então $f(\alpha, \beta) = (f_0; f_1; f_2; f_3)$ é calculada seqüencialmente por:

$$\begin{aligned} f_1 &= \alpha_1 \oplus \beta_0 \\ f_2 &= \alpha_2 \oplus \beta_1 \\ f_1 &= f_1 \oplus \alpha_0 \\ f_2 &= f_2 \oplus \alpha_3 \\ f_1 &= S_1(f_1, f_2) \\ f_2 &= S_0(f_2, f_1) \\ f_0 &= S_0(\alpha_0, f_1) \\ f_3 &= S_1(\alpha_3, f_2) . \end{aligned}$$

A figura 2.1 nos mostra a estrutura da função f .

Figura 2.1 ([Mi90]) Função f .

Função f_K

É utilizada na geração das subchaves. Sua definição é:

- Sejam $\alpha = (\alpha_0; \alpha_1; \alpha_2; \alpha_3)$ e $\beta = (\beta_0; \beta_1; \beta_2; \beta_3)$, onde os α_i , $i = 0, 1, 2, 3$, e os β_j , $j = 0, 1, 2, 3$, são bytes. Então $f_K(\alpha, \beta) = (f_{K_0}; f_{K_1}; f_{K_2}; f_{K_3})$ é calculada sequencialmente por:

$$f_{K_1} = \alpha_1 \oplus \alpha_0$$

$$f_{K_2} = \alpha_2 \oplus \alpha_3$$

$$f_{K_1} = S_1(f_{K_1}, (f_{K_2} \oplus \beta_0))$$

$$f_{K_2} = S_0(f_{K_2}, (f_{K_1} \oplus \beta_1))$$

$$f_{K_0} = S_0(\alpha_0, (f_{K_1} \oplus \beta_2))$$

$$f_{K_3} = S_1(\alpha_3, (f_{K_2} \oplus \beta_3))$$

A figura 2.2 mostra a estrutura da função f_K .

Figura 2.2 ([Mi90]) Função f_K .

Geração das subchaves

Inicialmente, vamos descrever a geração das subchaves do FEAL-NX. Obteremos as chaves K_i , $0 \leq i \leq N + 7$ a partir de uma chave K de 128 bits. Fazemos:

```

( $K_E; K_D$ ) :=  $K$            /*  $K_E$  e  $K_D$  são seqüências de 64 bits */
( $K_{D_1}; K_{D_2}$ ) :=  $K_D$      /*  $K_{D_1}$  e  $K_{D_2}$  são seqüências de 32 bits */
( $A_0; B_0$ ) :=  $K_E$            /*  $A_0$  e  $B_0$  são seqüências de 32 bits */
 $D_0 := \phi$ 
 $XK_D := K_{D_1} \oplus K_{D_2}$ 
para  $r := 1$  até  $\lceil (N/2) + 4 \rceil$  faça
   $D_r := A_{r-1}$ ;
   $A_r := B_{r-1}$ ;
  resto :=  $r \bmod 3$ ;
  se resto = 0 então  $Q_r := K_{D_2}$ ;
  se resto = 1 então  $Q_r := XK_D$ ;
  se resto = 2 então  $Q_r := K_{D_1}$ ;
   $B_r := f_K(A_{r-1}, (B_{r-1} \oplus D_{r-1} \oplus Q_r));$   /*  $B_r = (B_{r_0}; B_{r_1}; B_{r_2}; B_{r_3})$  */
   $K_{2(r-1)} := (B_{r_0}; B_{r_1});$                   /* os  $B_{r_i}$  são bytes,  $i \in \{0, 1, 2, 3\}$  */
   $K_{2(r-1)+1} := (B_{r_2}; B_{r_3});$ 
fim-para

```

A figura 2.3 mostra o diagrama da geração de subchaves para o FEAL-NX.

A geração das subchaves para o FEAL-N é equivalente à geração das subchaves para o FEAL-NX quando K_E é a chave do FEAL-N e $K_D = (\phi, \phi)$. Ou seja, para obtermos as subchaves K_i , $0 \leq i \leq N + 7$, para o FEAL-N a partir de uma chave K de 64 bits, fazemos:

```

( $A_0; B_0$ ) :=  $K$ ;
 $D_0 := \phi$ ;
para  $r := 1$  até  $\lceil (N/2) + 4 \rceil$  faça
   $D_r := A_{r-1}$ ;
   $A_r := B_{r-1}$ ;
   $B_r := f_K(A_{r-1}, (B_{r-1} \oplus D_{r-1}));$   /*  $B_r = (B_{r_0}; B_{r_1}; B_{r_2}; B_{r_3})$  */
   $K_{2(r-1)} := (B_{r_0}; B_{r_1});$           /* os  $B_{r_i}$  são bytes,  $i \in \{0, 1, 2, 3\}$  */
   $K_{2(r-1)+1} := (B_{r_2}; B_{r_3});$ 
fim-para

```

A figura 2.4 mostra o diagrama da geração de subchaves para o FEAL-N.

Figura 2.3 ([Mi90]) Geração das subchaves no FEAL-NX.

Agora, já temos todas as informações necessárias para vermos como os dados são cifrados e decifrados utilizando-se o FEAL.

Algoritmo de ciframento

Sejam:

- L , um texto legível de 64 bits;
- K , a chave a ser utilizada no processo de ciframento e
- K_i , $0 \leq i \leq N + 7$, as subchaves geradas a partir de K .

Então, para cifrarmos L utilizando a chave K , fazemos:

```

( $E_0; D_0$ ) :=  $L$ ;          /*  $E_0$  e  $D_0$  são seqüências de 32 bits */
( $E_0; D_0$ ) := ( $E_0; D_0$ )  $\oplus$  ( $K_N; K_{N+1}; K_{N+2}; K_{N+3}$ );
( $E_0; D_0$ ) := ( $E_0; D_0$ )  $\oplus$  ( $\phi; E_0$ );
para  $r := 1$  até  $N$  faça
     $D_r := E_{r-1} \oplus f(D_{r-1}, K_{r-1})$ ;
     $E_r := D_{r-1}$ ;
fim-para;
( $E_N; D_N$ ) := ( $D_N; E_N$ );
( $E_N; D_N$ ) := ( $E_N; D_N$ )  $\oplus$  ( $\phi; D_N$ );
( $E_N; D_N$ ) := ( $E_N; D_N$ )  $\oplus$  ( $K_{N+4}; K_{N+5}; K_{N+6}; K_{N+7}$ )

```

O texto cifrado é dado por $(E_N; D_N)$. A figura 2.5 mostra o diagrama de ciframento do FEAL-N(X).

Algoritmo de deciframento

Sejam:

- C , um texto cifrado de 64 bits;
- K , a chave que foi utilizada na obtenção de C a partir de um texto legível L e
- K_i , $0 \leq i \leq N + 7$, as subchaves obtidas de K .

Então, para decifrarmos C utilizando a chave K , fazemos:

```

( $E_N; D_N$ ) :=  $C$ ;          /*  $E_N$  e  $D_N$  São seqüências de 32 bits */
( $E_N; D_N$ ) := ( $E_N; D_N$ )  $\oplus$  ( $K_{N+4}; K_{N+5}; K_{N+6}; K_{N+7}$ );
( $E_N; D_N$ ) := ( $E_N; D_N$ )  $\oplus$  ( $\phi; E_N$ );
para  $r := N$  até 1 faça
     $D_{r-1} := E_r \oplus f(D_r, K_{r-1})$ ;
     $E_{r-1} := D_r$ ;
fim-para;

```

Figura 2.4 ([Mi90]) Geração das subchaves no FEAL-N.

$$\begin{aligned}
(E_0; D_0) &:= (D_0; E_0); \\
(E_0; D_0) &:= (E_0; D_0) \oplus (\phi; E_0); \\
(E_0; D_0) &:= (E_0; D_0) \oplus (K_N; K_{N+1}; K_{N+2}; K_{N+3});
\end{aligned}$$

O texto legível L é dado por $(E_0; D_0)$. A figura 2.6 traz o diagrama de deciframento do FEAL-N(X).

Notemos que, a única diferença entre os algoritmos de ciframento e deciframento é a ordem de utilização das subchaves. Na próxima seção, veremos algumas propriedades da função f .

2.2 Sobre a função f

A força criptográfica de uma cifra de Feistel está fortemente ligada às propriedades da sua função F . Assim sendo, vamos estabelecer alguns conceitos sobre funções criptográficas e ver como a função f do FEAL se comporta em relação aos mesmos.

Definição 2.1 ([Fu87]) Dizemos que uma função criptográfica é completa se cada um dos seus bits de saída depende de todos os bits de entrada.

Para que uma cifra de Feistel seja completa, não é necessário que a sua função F seja completa. Além disso, uma cifra de Feistel completa possui no mínimo três iterações [Fu87].

Definição 2.2 ([Fu87]) Dizemos que uma função F exibe o Efeito Avalanche se, em média, metade dos seus bits de saída muda quando um único bit em qualquer posição da entrada é alterado.

Definição 2.3 ([Fu87]) Uma função F satisfaz o Critério de Avalanche Estrito se cada um dos seus bits de saída muda com probabilidade $\frac{1}{2}$ quando um bit em qualquer posição da entrada é alterado.

Devemos notar que o Critério de Avalanche Estrito inclui a propriedade " F é completa". Este critério é considerado essencial para se obter uma "boa" transformação criptográfica.

Definição 2.4 ([Fu87]) A matriz de dependência de uma função $F : \{0,1\}^n \mapsto \{0,1\}^m$ é uma matriz A $n \times m$ cuja entrada a_{ij} nos fornece a probabilidade de que o j -ésimo bit da saída de F mude quando o i -ésimo bit de entrada da função é complementado.

Figura 2.5 ([Mi90]) Algoritmo de ciframento do FEAL- $N(X)$.

Claramente, se F é completa então sua matriz de dependência não possui elementos nulos. Também, se F satisfaz o Critério de Avalanche Estrita então o valor de cada elemento da sua matriz de dependência é aproximadamente $\frac{1}{2}$. Vejamos agora, o comportamento da função f do FEAL em relação a esses conceitos.

O FEAL-N é uma cifra de Feistel de $N + 2$ iterações e a força criptográfica da função f depende da não-linearidade da função S_i . A função S_i não é completa. Somente um dos seus bits de saída depende de todos os bits de entrada. A sua matriz de dependência é dada na tabela 2.1.

$\frac{1}{64}$	$\frac{1}{128}$	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$
$\frac{1}{32}$	$\frac{1}{64}$	0	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
$\frac{1}{16}$	$\frac{1}{32}$	0	0	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$
$\frac{1}{8}$	$\frac{1}{16}$	0	0	0	1	$\frac{1}{16}$	$\frac{1}{4}$
$\frac{1}{4}$	$\frac{1}{8}$	0	0	0	0	1	$\frac{1}{2}$
$\frac{1}{2}$	$\frac{1}{4}$	0	0	0	0	0	1
1	$\frac{1}{2}$	0	0	0	0	0	0
0	1	0	0	0	0	0	0
$\frac{1}{64}$	$\frac{1}{128}$	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$
$\frac{1}{32}$	$\frac{1}{64}$	0	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$
$\frac{1}{16}$	$\frac{1}{32}$	0	0	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$
$\frac{1}{8}$	$\frac{1}{16}$	0	0	0	1	$\frac{1}{16}$	$\frac{1}{4}$
$\frac{1}{4}$	$\frac{1}{8}$	0	0	0	0	1	$\frac{1}{2}$
$\frac{1}{2}$	$\frac{1}{4}$	0	0	0	0	0	1
1	$\frac{1}{2}$	0	0	0	0	0	0
0	1	0	0	0	0	0	0

Tabela 2.1 (Fu87)] *Matriz de dependência da função S_i .*

Como a função S_i não é completa, segue que a função f do FEAL também não é. Segundo Walter Fumy[Fu87], uma análise das potências de f nos mostra que:

- f^i é completa para $i \geq 2$ e
- f^i satisfaz o Critério de Avalanche Estrita para $i \geq 3$.

Assim, o FEAL-N (FEAL-NX) é completo e satisfaz o Critério de Avalanche Estrita para $N \geq 4$.

Figura 2.6 ([Mi90]) Algoritmo de deciframento do FEAL-N(X).

Capítulo 3

Criptanálise Diferencial das cifras de Feistel

¹A Criptanálise Diferencial é um método do tipo texto escolhido que foi desenvolvido por Eli Biham e Adi Shamir em 1990. Este método analisa o efeito da diferença entre os elementos de um par de textos legíveis sobre a diferença entre os elementos do respectivo par de textos cifrados. Em geral, muitos pares de textos legíveis que possuem a mesma diferença são utilizados. Tal utilização é feita somente através dos respectivos pares de textos cifrados, do conhecimento da diferença que está sendo usada e do seu valor. Para as cifras de Feistel, a diferença é escolhida como sendo a operação ou-exclusivo. Essas diferenças são utilizadas para atribuir probabilidades às chaves possíveis e, assim, localizar a chave mais provável. Neste capítulo, veremos como analisar e explorar essas diferenças. Na descrição do método, em qualquer ponto intermediário durante o ciframento de um par de textos legíveis, X e X^* são os valores intermediários correspondentes das duas execuções do algoritmo e $X' = X \oplus X^*$.

Definição 3.1 ([BS91]) *Uma chave independente é uma lista de n subchaves que não necessariamente foram derivadas de alguma chave através de um algoritmo de geração de subchaves.*

Por simplicidade, supõe-se que todas as chaves são independentes. Os ataques às cifras de Feistel com chaves dependentes também são bem sucedidos na prática, mas a sua análise teórica é muito mais complicada.

Algumas operações realizadas durante o processo de ciframento permitem saber com exatidão qual o valor da diferença entre os elementos do par após a sua execução. Por exemplo:

¹Os conceitos e conclusões deste capítulo foram extraídos de [BS91].

(i) O valor de X' não se altera quando é feito um ou-exclusivo entre os elementos do par e uma mesma constante K , pois

$$\begin{aligned}
 (X \oplus K) \oplus (X^* \oplus K) &= (X \oplus K) \oplus (K \oplus X^*) \\
 &= X \oplus (K \oplus K) \oplus X^* \\
 &= X \oplus 0 \oplus X^* \\
 &= X \oplus X^* \\
 &= X'.
 \end{aligned}$$

(ii) Sejam (W, X) , o bloco de entrada de uma iteração da cifra, e $Y = F(X)$. Então, a operação ou-exclusivo em um par é linear em relação à operação ou-exclusivo que faz a conexão entre duas iterações consecutivas, pois

$$\begin{aligned}
 (W \oplus Y) \oplus (W^* \oplus Y^*) &= (Y \oplus W) \oplus (W^* \oplus Y^*) \\
 &= Y \oplus (W \oplus W^*) \oplus Y^* \\
 &= Y \oplus W' \oplus Y^* \\
 &= Y \oplus Y^* \oplus W' \\
 &= Y' \oplus W'.
 \end{aligned}$$

Já outras operações não são lineares em relação ao ou-exclusivo em um par, tais como as S-boxes do DES[NBS77] e a soma módulo 256 existente na função S_i do FEAL. Nesses casos, não se pode garantir exatamente qual o valor do ou-exclusivo em um par após a execução da operação. Em geral, muitos valores são possíveis. Um caso especial ocorre quando os elementos do par são idênticos. Nessa situação, os elementos do par após a aplicação de qualquer operação também são idênticos. Por conveniência, chamaremos uma operação não linear de S-box.

Embora não se possa dizer com certeza qual o valor do ou-exclusivo em um par após a aplicação de uma S-box, é crucial observar que nem todos os valores da diferença são possíveis, os valores que são possíveis não são distribuídos uniformemente e alguns desses valores ocorrem com uma frequência muito maior que outros. Sendo assim, é natural que se faça uma análise probabilística do comportamento de uma S-box.

Definição 3.2 ([BS91]) *Uma tabela que mostra a distribuição dos valores da diferença de entrada e dos valores da diferença de saída de todos os pares possíveis de uma S-box é dita uma tabela de distribuição dos valores da diferença nos pares da S-box. Nessa tabela, cada linha corresponde a um valor fixo da diferença de entrada, cada coluna corresponde a um valor fixo da diferença de saída e as entradas da tabela possuem o número de pares possíveis com tais valores das diferenças de entrada e saída.*

Definição 3.3 ([BS91]) *Sejam X , uma seqüência de m_1 bits, e Y , uma seqüência de m_2 bits, onde m_1 e m_2 são os tamanhos de entrada e saída de uma S -box. Dizemos que X pode causar Y por uma S -box se existe um par (X_1, X_2) em $\{0, 1\}^{m_1} \times \{0, 1\}^{m_1}$ tal que $X = X_1 \oplus X_2$ e $Y = S(X_1) \oplus S(X_2)$, onde $S(X_i)$ denota a aplicação da S -box a X_i , $i \in \{1, 2\}$. Se um tal par existe, indicamos isso por $X \rightarrow Y$. Senão, dizemos que X não pode causar Y pela S -box e escrevemos $X \not\rightarrow Y$.*

Definição 3.4 ([BS91]) *Dizemos que X pode causar Y com probabilidade p por uma S -box se para uma fração p dos pares cujo valor da diferença é X , o valor da diferença após a aplicação da S -box é Y .*

Generalizando para a função F , temos:

Definição 3.5 ([BS91]) *Sejam X e Y , seqüências de $m/2$ bits, onde m é o tamanho do bloco da cifra de Feistel. Dizemos que X pode causar Y com probabilidade p pela função F se para uma fração p de todos os pares de entrada possíveis, nos quais o valor da diferença é X , cifrados por todos os valores possíveis da subchave, o valor da diferença de saída da função F é Y . Nos casos onde $p > 0$, escrevemos $X \rightarrow Y$.*

Em algumas cifras de Feistel, tais como DES[NBS77] e FEAL, vale o seguinte lema:

Lema 3.1 ([BS91]) *Se $X \rightarrow Y$ com probabilidade p pela função F então todo par de entrada fixado (Z, Z^*) tal que $Z' = Z \oplus Z^* = X$ causa o valor da diferença de saída da função F ser igual a Y pela mesma fração p de todos os valores possíveis da subchave.*

Para as cifras de Feistel nas quais esse lema não é válido, supõe-se que essa fração é muito próxima a p , o que geralmente é o caso.

A subchave da função F é então descoberta pelo uso do seguinte método:

- (i) Escolhe-se um valor apropriado para a diferença entre os textos legíveis;
- (ii) Cria-se um número adequado de pares de textos legíveis cujos elementos possuem o valor da diferença escolhido;
- (iii) Esses pares são então cifrados e guarda-se apenas os pares de textos cifrados resultantes;
- (iv) Para cada par, deriva-se o valor esperado da diferença de saída de tantas S -boxes quanto possível na última iteração, através do uso do valor da diferença escolhido para o par de textos legíveis e do par de textos cifrados. Observemos que o par de entrada da função F na última iteração é conhecido, pois ele é parte do par de textos cifrados;
- (v) Para cada valor possível da subchave, conta-se o número de pares que possuem o valor da diferença de saída esperado na última iteração;
- (vi) O valor correto da subchave é o valor sugerido pelo maior número de pares, o qual nem sempre é único.

Entretanto, para a utilização deste método, devemos resolver o problema de se empurrar o conhecimento do valor da diferença no par de textos legíveis por tantas iterações quanto possível, sem que o valor da diferença de entrada da função F se torne nulo em duas iterações consecutivas. Tal cuidado deve ser tomado pois, quando o valor da diferença de entrada for zero, o valor da diferença de saída da função F também será zero, o que torna todas as subchaves igualmente prováveis.

Definição 3.6 ([BS91]) *Sejam Ω_L e Ω_C , os valores da diferença nos pares de textos legíveis e de textos cifrados, respectivamente. Uma característica de n iterações é uma tripla $\Omega = (\Omega_L, \Omega_\Lambda, \Omega_C)$, onde Ω_Λ é uma lista de n elementos $(\Lambda_1, \Lambda_2, \dots, \Lambda_n)$, cada um sendo um par da forma $\Lambda_i = (\lambda_{E_n}^i, \lambda_S^i)$, onde $\lambda_{E_n}^i$ e λ_S^i são seqüências de $m/2$ bits e m é o tamanho de bloco da cifra. Esta tripla deve satisfazer as seguintes condições:*

- (i) $\lambda_{E_n}^1$ é a metade direita de Ω_L ;
- (ii) $\lambda_{E_n}^2 = \lambda_S^1 \oplus (\text{metade esquerda de } \Omega_L)$;
- (iii) $\lambda_{E_n}^n$ é a metade direita de Ω_C ;
- (iv) $\lambda_{E_n}^{n-1} = \lambda_S^n \oplus (\text{metade esquerda de } \Omega_C)$ e
- (v) para todo i tal que $i \in \{2, 3, \dots, n-1\}$,

$$\lambda_S^i = \lambda_{E_n}^{i-1} \oplus \lambda_{E_n}^{i+1}$$

A figura 3.1 mostra uma característica genérica de n iterações.

Exemplo: Uma característica de uma iteração válida para toda cifra de Feistel com probabilidade 1.

Figura 3.1 Uma característica de n iterações genérica

Definição 3.7 ([BS91]) Um par certo relativo a uma característica de n iterações Ω e a uma chave independente K é um par (L, L^*) para o qual $L' = \Omega_L$ e, para as n primeiras iterações do ciframento do par utilizando-se a chave K , na i -ésima iteração o valor da diferença de entrada da função F é igual a λ_{En}^i e o valor da diferença de saída de F é igual a λ_S^i . Um par que não satisfaz essas condições é um par errado relativo à característica e a chave independente. Por simplicidade, diz-se apenas par certo e par errado.

Definição 3.8 ([BS91]) A concatenação de uma característica de n iterações $\Omega^1 = (\Omega_L^1, \Omega_\Lambda^1, \Omega_C^1)$ com uma característica de m iterações $\Omega^2 = (\Omega_L^2, \Omega_\Lambda^2, \Omega_C^2)$, onde Ω_C^1 é igual ao valor de Ω_L^2 com as metades trocadas, é a característica de $m+n$ iterações $\Omega = (\Omega_L^1, \Omega_\Lambda, \Omega_C^2)$, onde Ω_Λ é a concatenação das listas Ω_Λ^1 e Ω_Λ^2 .

Definição 3.9 ([BS91]) A i -ésima iteração de uma característica Ω , $1 \leq i \leq n$, tem probabilidade p_i^Ω se λ_{En}^i causa λ_S^i pela função F com probabilidade p_i^Ω .

Definição 3.10 ([BS91]) Uma característica de n iterações Ω tem probabilidade p^Ω se p^Ω é o produto das probabilidades das suas n iterações, ou seja:

$$p^\Omega = \prod_{i=1}^n p_i^\Omega$$

Pelas definições 3.8 e 3.10, a probabilidade de uma característica Ω que é a concatenação de Ω^1 com Ω^2 , é o produto das probabilidades p^{Ω^1} e p^{Ω^2} . Como consequência, toda característica de n iterações pode ser descrita como a concatenação de n características de uma iteração.

Teorema 3.1 ([BS91]) A probabilidade de uma característica de n iterações Ω é a probabilidade de que um par de textos legíveis fixado (L, L^*) , tal que $L' = \Omega_L$, seja um par certo quando chaves independentes e aleatórias são utilizadas.

Dem.: A probabilidade de que qualquer par de textos legíveis (L, L^*) que satisfaça $L' = \Omega_L$ seja um par certo é a probabilidade de que a cada iteração i , $1 \leq i \leq n$, λ_{En}^i cause λ_S^i pela função F . Pelo Lema 3.1, a probabilidade em cada iteração é independente da entrada exata e, como as chaves independentes e aleatórias randomizam as entradas de cada S-box, mantendo apenas o valor da diferença fixado, a i -ésima iteração independe das anteriores. Logo, a probabilidade de um par ser certo é o produto das probabilidades de λ_{En}^i causar λ_S^i pela função F , $1 \leq i \leq n$, o qual pela Definição 3.10 é a probabilidade da característica Ω . $\square$

Na prática, a probabilidade p^Ω é a probabilidade de que um par (L, L^*) , tal que $L' = \Omega_L$, seja um par certo, utilizando-se uma chave fixa, a qual se tenta encontrar. Esta probabilidade não é constante para todas as chaves, entretanto, supõe-se que p^Ω é uma aproximação muito boa.

Definição 3.11 ([BS91]) *Uma característica Ω é uma característica iterativa se Ω_L com as metades trocadas for igual a Ω_C .*

Pode-se concatenar uma característica iterativa a ela mesma um número arbitrário de vezes e obter características com um número arbitrário de iterações. A grande vantagem das características iterativas é que pode-se construir uma característica de n iterações com uma taxa de redução da probabilidade fixa para cada nova iteração. Por outro lado, em características não iterativas a taxa de redução da probabilidade geralmente cresce devido ao Efeito Avalanche.

O comportamento probabilístico da maioria das características não nos permite olhar para a intersecção de todas as chaves sugeridas pelos pares pois essa intersecção é geralmente vazia, dado que os pares errados não sugerem necessariamente a chave correta como um valor possível. Entretanto, a chave correta deve ser sugerida por todos os pares certos, os quais ocorrem, aproximadamente, com a probabilidade da característica. Todos os outros valores de chave possíveis são distribuídos uniformemente. Para se encontrar a chave, faz-se a contagem do número de ocorrências de cada uma das chaves sugeridas. A chave correta deve ser a que ocorre mais frequentemente.

Cada característica nos permite examinar um certo número de bits da subchave da última iteração. As características mais úteis são aquelas que possuem uma probabilidade máxima e um número máximo de bits da subchave cujas ocorrências possam ser contadas. Ainda assim, não é necessário fazer a contagem sobre todos os bits possíveis da subchave. As vantagens de se efetuar a contagem sobre todos os bits possíveis da subchave são a boa identificação do valor da chave correta e a pequena quantidade de dados necessários. Entretanto, contar o número de ocorrências de todos os valores possíveis de um grande número de bits exige muita memória, o que pode tornar o ataque inviável. Assim, pode-se fazer a contagem sobre um número menor de bits e utilizar os outros bits restantes apenas para identificar e descartar os pares errados nos quais o valor da diferença de entrada não pode causar o valor da diferença de saída.

A definição a seguir nos fornece um meio de avaliar a viabilidade de um método de contagem baseado em uma característica.

Definição 3.12 ([BS91]) *A proporção entre o número de pares certos e a contagem média em um método de contagem é chamada de proporção sinal-ruído do método e é denotada por S/R .*

Para encontrar a chave certa em um método de contagem, são necessários uma característica com probabilidade alta e pares de textos cifrados suficientes para garantir a existência de vários pares certos. A quantidade de pares necessária depende da probabilidade da característica, do número de bits da subchave que são contados e da porcentagem de pares errados que podem ser descartados antes de serem contados. Se estivermos fazendo a contagem sobre k bits da subchave então nós contamos o número de ocorrências de 2^k chaves possíveis em 2^k contadores. Os contadores contêm uma contagem média de

$m\alpha\beta/2^k$, onde m é o número de pares, α é a contagem média por par contado e β é a proporção entre os pares contados e todos os pares, i.é., contados e descartados.

A chave certa é contada aproximadamente mp vezes através dos pares certos, mais a contagem aleatória estimada acima para todas as chaves possíveis. A proporção sinal-ruído de um método de contagem baseado em uma característica é portanto:

$$S/R = \frac{mp}{m\alpha\beta/2^k} = \frac{2^k p}{\alpha\beta} \quad (3.1)$$

Como conseqüência da fórmula 3.1 temos que a proporção S/R de um método de contagem independe da quantidade de pares utilizada e que métodos de contagem distintos baseados em uma mesma característica, mas com um número diferente de bits da subchave que são contados, possuem valores de S/R distintos.

Em geral, relaciona-se o número de pares exigidos por um método de contagem com o número de pares certos que são necessários. Já o número de pares certos necessários é basicamente uma função da proporção S/R . Quando S/R é um valor suficientemente grande, poucos pares certos são necessários para identificar de forma única o valor dos bits que são contados. Biham e Shamir observaram experimentalmente que quando o valor de S/R é muito maior que 2, em geral, três ou quatro pares certos são suficientes. Quando o valor de S/R é muito menor que 1, um número muito grande de pares certos é necessário.

Normalmente os ataques utilizam várias características ao mesmo tempo. Assim, é natural buscar-se estruturas que armazenem os dados necessários de uma forma econômica.

Definição 3.13 ([BS91]) *Um quarteto é uma estrutura de quatro textos cifrados que contém simultaneamente dois pares de textos cifrados de uma característica Ω^1 e dois pares de textos cifrados de uma característica Ω^2*

Exemplo de um quarteto([BS91]): Seja L um texto qualquer de m bits. Então os textos:

- (1) L
- (2) $L \oplus \Omega_L^1$
- (3) $L \oplus \Omega_L^2$
- (4) $L \oplus \Omega_L^1 \oplus \Omega_L^2$,

após serem cifrados formam um quarteto, onde (1,2) e (3,4) são pares de Ω^1 e (1,3) e (2,4) são pares de Ω^2 .

Definição 3.14 ([BS91]) *Um octeto é uma estrutura de oito textos cifrados que contém simultaneamente quatro pares de textos cifrados para três características.*

Exemplo de um octeto: Seja L um texto qualquer de m bits. Então os textos:

- (1) L
- (2) $L \oplus \Omega_L^1$
- (3) $L \oplus \Omega_L^2$

- (4) $L \oplus \Omega_L^1 \oplus \Omega_L^2$
- (5) $L \oplus \Omega_L^3$
- (6) $L \oplus \Omega_L^3 \oplus \Omega_L^1$
- (7) $L \oplus \Omega_L^3 \oplus \Omega_L^2$
- (8) $L \oplus \Omega_L^3 \oplus \Omega_L^1 \oplus \Omega_L^2$,

após serem cifrados formam um octeto, onde:

- (1,2), (3,4), (5,6) e (7,8) são pares de Ω^1 ;
- (1,3), (2,4), (5,7) e (6,8) são pares de Ω^2 ;
- (1,5), (2,6), (3,7) e (4,8) são pares de Ω^3 .

Existe um tipo especial de octeto que contém quatro pares de quatro características Ω^1 , Ω^2 , Ω^3 e Ω^4 . Isso ocorre, por exemplo, quando $\Omega^4 = \Omega^1 \oplus \Omega^2 \oplus \Omega^3$ [BS91a]. Então, no nosso exemplo de octeto, os pares (1,8), (2,7), (3,6) e (4,5) são pares de Ω^4 .

A utilização de quartetos proporciona uma economia de metade dos dados, enquanto que a utilização de octetos economiza dois terços dos dados. O octeto especial nos dá uma economia de 75%.

Definição 3.15 *Seja n o número de iterações de uma cifra de Feistel. Se as características utilizadas pelo ataque diferencial possuírem $n-iR$ iterações, $0 \leq i < n$, então diz-se que o ataque é do tipo iR . Na prática, os valores de i são menores ou iguais a três.*

No próximo capítulo, veremos como esse método de criptoanálise pode ser aplicado à família de cifras FEAL.

Capítulo 4

Criptanálise Diferencial da família de cifras FEAL

O primeiro membro da família FEAL, denominado FEAL-4[ShMi87], foi quebrado pela primeira vez em 1988 por Bert Den Boer[BDB88] através de um ataque do tipo texto escolhido que utiliza uma quantidade de textos cifrados que varia entre 100 e 10.000. Como resposta, foi criado o FEAL-8[MiShSh88], o qual é quebrável pelo método da Criptanálise Diferencial[BS91a] através da utilização de apenas 2.000 textos cifrados. Novamente, os projetistas do FEAL reagiram criando o FEAL-N e o FEAL-NX[Mi90], o qual, como vimos no capítulo 2, é um FEAL-N que utiliza uma chave de 128 bits.

Veremos neste capítulo que tanto o FEAL-N quanto o FEAL-NX são quebráveis pela Criptanálise Diferencial mais rápido do que por busca exaustiva para qualquer N menor que 32. As outras criptanálises do FEAL publicadas até o presente momento, quer sejam do tipo texto escolhido quer sejam do tipo texto conhecido, somente conseguem sucesso para valores de N no máximo iguais a 8 [GC90; Mu90; TG91; MY92]. Muitos detalhes necessários a um bom entendimento do assunto que se encontram omissos na publicação original[BS91a] foram acrescentados neste capítulo.

4.1 Considerações gerais

Vamos inicialmente introduzir a notação a ser utilizada na descrição dos ataques, a qual é sempre válida a menos que se diga o contrário.

Notação:

- um número hexadecimal é denotado por um índice x (ex. $10_x = 16$),

- durante qualquer ponto intermediário no processo de ciframento dos pares de textos legíveis, X e X^* são os valores intermediários correspondentes das duas execuções do processo e $X' = X \oplus X^*$,

- L , um texto legível de 64 bits,
- C , um texto cifrado de 64 bits,
- $(e; d)$, as metades esquerda e direita de C ,
- $E(i)$, o bloco de entrada da função f na i -ésima iteração, $1 \leq i \leq N$,
- $SS(i)$, o bloco de saída da função f na i -ésima iteração, $1 \leq i \leq N$,
- X_i , o i -ésimo byte de X (a numeração é feita da esquerda para a direita),
- X_i^j , o j -ésimo bit de X_i , onde X_i^0 é o bit menos significativo de X_i ,
- $(X; Y)$, a concatenação de X e Y , onde X e Y são seqüências de bits.

Passaremos agora a analisar as operações existentes no FEAL sob o ponto de vista da Criptoanálise Diferencial.

No início e no final do processo de ciframento(deciframento), a metade direita do texto é misturada com a metade esquerda e o bloco completo é misturado com subchaves adicionais, o que é feito através do uso da operação ou-exclusivo. Graças a sua linearidade em relação à diferença no par, estas operações representam dificuldades menores aos ataques diferenciais[BS91a].

A operação ou-exclusivo entre subchaves e o texto cifrado é significativa quando tentamos encontrar as subchaves, pois o bloco de entrada da função f na última iteração é uma função do texto cifrado misturado com subchaves adicionais, ao invés de ser apenas uma função do texto cifrado como, por exemplo, no DES[NBS77]. Portanto, o método de contagem da subchave encontra uma combinação da subchave da última iteração com as subchaves adicionais[BS91a]. A seguir, essas subchaves “misturadas” são definidas formalmente.

Definição 4.1 ([BS91a]) *As combinações de subchaves que são encontradas pelo ataque são chamadas subchaves reais. A subchave real da $(i+1)$ -ésima iteração é denotada por AK_i , $0 \leq i \leq (N-1)$. As combinações de 16 bits $((AK_i)_0 \oplus (AK_i)_1; (AK_i)_2 \oplus (AK_i)_3)$ são chamadas subchaves reais de 16 bits. A subchave real da última iteração de uma cifra é chamada de última subchave real.*

O exemplo abaixo ilustra a definição 4.1 para o FEAL-8.

Exemplo([BS91a]) As subchaves reais do FEAL-8 nas iterações pares $(i+1)$ são os valores de 32 bits:

$$AK_i = (K_c; K_d) \oplus (K_e; K_f) \oplus am(K_i)$$

onde $am(K_i) = (0; (K_i)_0; (K_i)_1; 0)$. As subchaves reais nas iterações ímpares são os valores:

$$AK_i = (K_c; K_d) \oplus am(K_i).$$

Essa abordagem em relação às subchaves é feita em quase todos os ataques ao FEAL publicados até o presente momento.

As operações existentes na função f do FEAL são:

- (i) ou-exclusivo do bloco de entrada com a subchave,
- (ii) ou-exclusivo entre bytes do bloco de entrada,
- (iii) rotação de dois bits à esquerda em um byte e
- (iv) soma módulo 256 de dois bytes.

Como visto no capítulo 3, a diferença em um par é invariante em relação à operação que executa um ou-exclusivo entre uma constante e cada elemento do par. Conseqüentemente, o ou-exclusivo entre o bloco de entrada da função f e a subchave não altera o valor da diferença entre os elementos de um par. A operação ou-exclusivo entre bytes do bloco de entrada é linear em relação ao ou-exclusivo no par, assim como também o é a operação que executa uma rotação de dois bits à esquerda sobre um byte. Como a soma módulo 256 de dois bytes não é linear em relação à diferença no par, ela é a única desse tipo existente no FEAL e, portanto, a força criptográfica da cifra depende fortemente dela.

A operação soma módulo 256 é não linear, mas ainda assim os valores da diferença nos pares de entrada e saída da função S_i não são uniformemente distribuídos ($i = 0$ ou $i = 1$. Futuramente, teceremos considerações sobre a utilização de outros valores de i). No FEAL, não é necessário consultarmos a tabela de distribuição dos valores da diferença nos pares da função S_i (S-box) para decidirmos se dados três bytes X, Y e Z , $(X; Y) \rightarrow Z$ ou $(X; Y) \rightsquigarrow Z$ como no DES[NBS77][BS91a]. Na próxima seção, vamos obter um algoritmo que além de decidir se $(X; Y) \rightarrow Z$ ou $(X; Y) \rightsquigarrow Z$ pela função S_i , nos dá a probabilidade com a qual o evento ocorre.

4.2 Um algoritmo para decidir se $(X; Y) \rightarrow Z$ ou $(X; Y) \rightsquigarrow Z$ pela função S_i

Sejam X, Y, Z , bytes. Se $(X; Y)$ pode causar Z pela função S_i então existem bytes $X_1, X_2, Y_1, Y_2, Z_1, Z_2$ tais que:

- $X = X_1 \oplus X_2$,
- $Y = Y_1 \oplus Y_2$,
- $Z_1 = S_i(X_1, Y_1)$,
- $Z_2 = S_i(X_2, Y_2)$ e
- $Z = Z_1 \oplus Z_2$.

Utilizando a definição de S_i , o fato de $Rot2$ ser inversível e uma operação linear em relação à operação ou-exclusivo e, denotando $(Rot2)^{-1}(Z)$ por W , podemos escrever:

$$W^j = (X_1 + Y_1 + i)^j \oplus (X_2 + Y_2 + i)^j, \quad 0 \leq j \leq 7. \quad (4.1)$$

A fórmula (4.1) contém variáveis que desconhecemos. Como somente conhecemos valores da diferença entre algumas dessas variáveis, vamos reescrever (4.1) em função dessas diferenças conhecidas.

Lema 4.1 *Se A e B são bytes então o j -ésimo bit de $A+B$, $0 \leq j \leq 7$, é dado por:*

$$(A + B)^j = A^j \oplus B^j \oplus C^j,$$

onde,

$$C^j = \begin{cases} 0, & \text{se } j = 0 \\ (A^{j-1} + B^{j-1} + C^{j-1}) \text{ div } 2, & \text{c.c.} \end{cases}$$

é o valor de "carry" do bit $(j - 1)$ para $j > 0$.

Dem.: (Por indução em j)

Base: Se $j = 0$ então

$$\begin{aligned} (A + B)^j &= (A^j + B^j) \text{ mod } 2 \\ &= A^j \oplus B^j \\ &= A^j \oplus B^j \oplus 0 \\ &= A^j \oplus B^j \oplus C^j \end{aligned}$$

Se $j = 1$ então

$$\begin{aligned} (A + B)^j &= (A^j + B^j + ((A^{j-1} + B^{j-1}) \text{ div } 2)) \text{ mod } 2 \\ &= A^j \oplus B^j \oplus ((A^{j-1} + B^{j-1}) \text{ div } 2) \\ &= A^j \oplus B^j \oplus ((A^{j-1} + B^{j-1} + 0) \text{ div } 2) \\ &= A^j \oplus B^j \oplus ((A^{j-1} + B^{j-1} + C^{j-1}) \text{ div } 2) \\ &= A^j \oplus B^j \oplus C^j \end{aligned}$$

Hipótese: Suponhamos que o lema seja verdadeiro para j , $1 \leq j < 7$.

Passo: Seja $\text{carry}(k)$ o valor de "carry" do k -ésimo bit de $A + B$, $0 \leq k \leq j$. Então

$$\begin{aligned} (A + B)^{j+1} &= (A^{j+1} + B^{j+1} + \text{carry}(j)) \text{ mod } 2 \\ &= (A^{j+1} + B^{j+1} + ((A^j + B^j + \text{carry}(j-1)) \text{ div } 2)) \text{ mod } 2 \end{aligned}$$

Pela hipótese de indução, $\text{carry}(j-1) = C^j$. Assim,

$$\begin{aligned} (A + B)^{j+1} &= (A^{j+1} + B^{j+1} + ((A^j + B^j + C^j) \text{ div } 2)) \text{ mod } 2 \\ &= (A^{j+1} + B^{j+1} + C^{j+1}) \text{ mod } 2 \\ &= A^{j+1} \oplus B^{j+1} \oplus C^{j+1} \end{aligned}$$

□

Sabendo que $i \in \{0, 1\}$ e fazendo uso do lema 4.1, reescrevemos a fórmula (4.1) como:

$$W^j = X_1^j \oplus Y_1^j \oplus C_1^j \oplus X_2^j \oplus Y_2^j \oplus C_2^j, \quad 0 \leq j \leq 7 \quad (4.2)$$

onde

$$C_k^j = \begin{cases} i, & \text{se } j = 0 \\ (X_k^{j-1} + Y_k^{j-1} + C_k^{j-1}) \text{ div } 2, & \text{c.c.} \end{cases}$$

e $k \in \{1, 2\}$.

Podemos ainda escrever (4.2) como:

$$W^j = X^j \oplus Y^j \oplus C^j, \quad 0 \leq j \leq 7 \quad (4.3)$$

onde $C^j = C_1^j \oplus C_2^j$.

O sistema de equações (4.3) ainda contém variáveis não conhecidas, pois só conhecemos o valor de C^j para $j = 0$ ($C^0 = i \oplus i = 0$). Entretanto, podemos analisar o comportamento de C^{j+1} a partir de X^j, Y^j, C^j . A tabela 4.1 mostra um resumo dessa análise.

X^j	Y^j	C^j	W^j	C^{j+1}
0	0	0	0	0
0	0	1	1	∀
0	1	0	1	∀
0	1	1	0	∀
1	0	0	1	∀
1	0	1	0	∀
1	1	0	0	∀
1	1	1	1	1

“∀ indica que o valor do bit pode ser tanto 0 quanto 1”

Tabela 4.1 Comportamento de C^{j+1} a partir de X^j, Y^j, C^j .

Partindo da tabela 4.1, vamos enunciar alguns lemas que serão úteis na elaboração do nosso algoritmo.

Definição 4.2 Dizemos que C^j , $0 \leq j \leq 7$, está definido se $j=0$ ou se o seu valor é determinado de maneira única por X^{j-1}, Y^{j-1} e C^{j-1} .

Lema 4.2 C^{j+1} está definido para algum j , $0 \leq j \leq 7$ se, e somente se, $C^j = X^j = Y^j$. Além disso, nesse caso, $C^{j+1} = C^j$.

Dem.: Por inspeção da tabela 4.1

□

Lema 4.3 *Se para algum j , $0 \leq j \leq 7$, o valor de C^j está definido então podemos ter:*

$$(i) W^j = X^j \oplus Y^j \oplus C^j \text{ ou}$$

$$(ii) W^j \neq X^j \oplus Y^j \oplus C^j.$$

Dem.: Seja $j \in \{0, \dots, 7\}$ tal que o valor de C^j está definido. Notemos que $W^j \oplus X^j \oplus Y^j \in \{0, 1\}$. Se $C^j = W^j \oplus X^j \oplus Y^j$ então vale (i). Caso contrário, (ii) é válido. $\square$

Lema 4.4 *Se para algum j , $0 < j \leq 7$, o valor de C^j não está definido então sempre vale $W^j = X^j \oplus Y^j \oplus C^j$.*

Dem.: Como C^j pode ser tanto 0 quanto 1, definimos C^j por $W^j \oplus X^j \oplus Y^j$ e obtemos a igualdade entre W^j e $X^j \oplus Y^j \oplus C^j$. $\square$

Lema 4.5 *Se C^j está definido para algum j , $0 \leq j \leq 7$, então C^{j+1} também estará definido se, e somente se, $(X^j \wedge Y^j \wedge C^j = 1)$ ou $(\overline{X^j} \wedge \overline{Y^j} \wedge \overline{C^j} = 1)$.*

Dem.: $(\Rightarrow)$ Se C^{j+1} está definido para algum j , $0 \leq j \leq 7$, segue do lema 4.2 que $C^j = C^{j+1} = X^j = Y^j$. Assim, se $C^j = 0$ então $X^j = Y^j = 0$ e $\overline{X^j} \wedge \overline{Y^j} \wedge \overline{C^j} = 1$. Se $C^j = 1$ então $X^j = Y^j = 1$ e $X^j \wedge Y^j \wedge C^j = 1$.

$(\Leftarrow)$ Se $(X^j \wedge Y^j \wedge C^j = 1)$ ou $(\overline{X^j} \wedge \overline{Y^j} \wedge \overline{C^j} = 1)$ para algum j , $0 \leq j \leq 7$, então, claramente, $X^j = Y^j = C^j$. Pelo lema 4.2, segue que C^{j+1} é um valor definido e igual a C^j . $\square$

Lema 4.6 *Se C^j não está definido para algum j , $0 < j \leq 7$, então C^{j+1} estará definido se, e somente se, $(X^j \wedge Y^j \wedge W^j = 1)$ ou $(\overline{X^j} \wedge \overline{Y^j} \wedge \overline{W^j} = 1)$.*

Dem.: Se C^j não está definido para algum j , $0 < j \leq 7$, então pelo lema 4.4 sempre vale $W^j = X^j \oplus Y^j \oplus C^j$, donde $C^j = X^j \oplus Y^j \oplus W^j$, ou seja, C^j passa a ser um valor determinado. Como C^j agora é um valor definido, segue do lema 4.5 que C^{j+1} também estará definido se, e somente se, $X^j \wedge Y^j \wedge C^j = 1$ ou $\overline{X^j} \wedge \overline{Y^j} \wedge \overline{C^j} = 1$.

Assim, C^{j+1} estará definido se, e somente se:

$$X^j \wedge Y^j \wedge (X^j \oplus Y^j \oplus W^j) = 1 \text{ ou } \overline{X^j} \wedge \overline{Y^j} \wedge \overline{(X^j \oplus Y^j \oplus W^j)} = 1.$$

Agora,

$$\begin{aligned}
X^j \wedge Y^j \wedge (X^j \oplus Y^j \oplus W^j) &= (X^j \wedge Y^j \wedge X^j) \oplus (X^j \wedge Y^j \wedge Y^j) \oplus (X^j \wedge Y^j \wedge W^j) \\
&= (X^j \wedge Y^j) \oplus (X^j \wedge Y^j) \oplus (X^j \wedge Y^j \wedge W^j) \\
&= 0 \oplus (X^j \wedge Y^j \wedge W^j) \\
&= X^j \wedge Y^j \wedge W^j
\end{aligned}$$

e

$$\begin{aligned}
\overline{X^j} \wedge \overline{Y^j} \wedge \overline{(X^j \oplus Y^j \oplus W^j)} &= \overline{X^j} \wedge \overline{Y^j} \wedge (\overline{X^j} \odot \overline{Y^j} \odot \overline{W^j}) \\
&= (\overline{X^j} \wedge \overline{Y^j} \wedge \overline{X^j}) \odot (\overline{X^j} \wedge \overline{Y^j} \wedge \overline{Y^j}) \odot (\overline{X^j} \wedge \overline{Y^j} \wedge \overline{W^j}) \\
&= (\overline{X^j} \wedge \overline{Y^j}) \odot (\overline{X^j} \wedge \overline{Y^j}) \odot (\overline{X^j} \wedge \overline{Y^j} \wedge \overline{W^j}) \\
&= 1 \odot (\overline{X^j} \wedge \overline{Y^j} \wedge \overline{W^j}) \\
&= \overline{X^j} \wedge \overline{Y^j} \wedge \overline{W^j}
\end{aligned}$$

Logo, o lema está demonstrado. $\square$

Lema 4.7 *Se C^j não está definido, $0 < j \leq 7$, mas C^{j+1} está definido então*

$$C^{j+1} = W^j.$$

Dem.: Como C^j não está definido, mas C^{j+1} está, pelo lema 4.6 ou $(X^j \wedge Y^j \wedge W^j = 1)$ ou $(\overline{X^j} \wedge \overline{Y^j} \wedge \overline{W^j} = 1)$, ou seja, $X^j = Y^j = W^j$. Pelo lema 4.4, C^j passará a ser definido por $W^j \oplus X^j \oplus Y^j$. Logo, $C^j = W^j$ e, pelo lema 4.2, $C^{j+1} = C^j$. $\square$

Como já temos em mãos informações suficientes para elaborarmos o algoritmo, vamos a ele.

Algoritmo Zero

“Dados os bytes X , Y e Z , decide se $(X; Y) \rightarrow Z$ ou $(X; Y) \rightsquigarrow Z$ pela função S_i .”

início

```

01   $W := (Rot2)^{-1}(Z);$ 
02   $c := 0;$ 
03   $j := 0;$ 
04   $acabou := falso;$ 
05   $leva := verdadeiro;$ 
06   $cdefinido := verdadeiro;$ 

```

```

07  enquanto não acabou faça
08    se cdefinido então
09      se  $(W^j = X^j \oplus Y^j \oplus c)$  então
10        se  $(X^j \wedge Y^j \wedge c = 0 \text{ e } \overline{X^j} \wedge \overline{Y^j} \wedge \overline{c} = 0)$  então
11           $cdefinido := falso;$ 
12        fim-se
13      senão
14         $leva := falso;$ 
15         $acabou := verdadeiro;$ 
16      fim-se
17    senão
18      se  $(X^j \wedge Y^j \wedge W^j = 1 \text{ ou } \overline{X^j} \wedge \overline{Y^j} \wedge \overline{W^j} = 1)$  então
19         $cdefinido := verdadeiro;$ 
20         $c := W^j;$ 
21      fim-se
22    fim-se
23     $j := j + 1;$ 
24    se  $(j = 8)$  então
25       $acabou := verdadeiro;$ 
26    fim-se
27  fim-enquanto
28  se leva então
29     $(X; Y) \rightarrow Z;$ 
30  senão
31     $(X; Y) \rightsquigarrow Z;$ 
32  fim-se;
fim.

```

A seguir, justificamos o algoritmo Zero.

Lema 4.8 *Dados os bytes X, Y, Z , o algoritmo Zero decide corretamente se, pela função S_i , $(X; Y) \rightarrow Z$ ou $(X; Y) \rightsquigarrow Z$.*

Dem.: O algoritmo Zero utiliza a fórmula (4.3) e o fato de que $C^0 = 0$. Como os valores de X, Y e Z são dados e $W = (Rot2)^{-1}(Z)$, só nos resta verificar se os valores de C^j estão sendo definidos corretamente, $1 \leq j \leq 7$. Para isso, faremos indução sobre j . Observemos que a variável c contém o valor de C^j a cada iteração.

Base($j = 0$): $C^j = i \oplus i = 0$.

Hipótese: Suponhamos que o valor de C^j esteja sendo definido corretamente para j , $0 \leq j < 7$, e que j -ésima iteração não seja a última executada.

Passo: Se C^j é um valor definido então C^{j+1} também será um valor definido se, e somente se, $(W^j = X^j \oplus Y^j \oplus C^j) \wedge ((X^j \wedge Y^j \wedge C^j = 1) \vee (\overline{X^j} \wedge \overline{Y^j} \wedge \overline{C^j} = 1))$ (cf. linhas 08 - 12). Como não há motivo para definirmos C^{j+1} quando $(W^j \neq X^j \oplus Y^j \oplus C^j)$, segue do lema 4.5 que a lógica está correta. Ademais, C^{j+1} é igual a C^j , o que está de acordo com o lema 4.2.

Se C^j não é um valor definido então C^{j+1} será um valor definido se, e somente se, $((X^j \wedge Y^j \wedge W^j = 1) \vee (\overline{X^j} \wedge \overline{Y^j} \wedge \overline{W^j} = 1))$, (cf. linhas 17 - 22). Pelo lema 4.6, está correto. O valor de C^{j+1} é igual, neste caso, a W^j , o que confere com o lema 4.7.

Logo, C está sendo definido corretamente.

Falta mostrar que a decisão feita é a correta. O algoritmo Zero decide que $(X; Y) \rightsquigarrow Z$ se, e somente se, C^j é um valor definido e $W^j \neq X^j \oplus Y^j \oplus C^j$ (cf. linhas 08 - 16), o que confere com os lemas 4.3 e 4.4. $\square$

Lema 4.9 *O algoritmo Zero é um processo finito.*

Dem.: O algoritmo possui um único laço condicional (cf. linhas 07 - 27), cuja condição de controle é (não acabou). Inicialmente, *acabou* é falso e $j = 0$. A cada iteração, o valor de j é incrementado por uma unidade. Se para algum j a fórmula (4.3) não se verifica, *acabou* passa a ser verdadeiro. Senão, quando j atinge o valor 8 *acabou* passa a ser verdadeiro. Logo, o algoritmo Zero termina após no máximo oito iterações. $\square$

Lema 4.10 *O algoritmo Zero está correto e sua complexidade é $O(1)$.*

Dem.: A corretude do algoritmo decorre dos lemas 4.8 e 4.9. Agora, considerando que as operações de atribuição, comparação e lógicas sejam executadas em tempo constante, o que em geral é o caso, e, como ressaltado na demonstração do lema 4.9, o número de iterações é no máximo oito, concluímos que o tempo de execução do algoritmo é sempre menor do que uma certa constante M . Logo, a complexidade do algoritmo Zero é $O(1)$. $\square$

Procuraremos agora alterar o algoritmo Zero para que, além de decidir se $(X; Y) \rightarrow Z$ ou $(X; Y) \rightsquigarrow Z$, também nos informe, nos casos em que $(X; Y) \rightarrow Z$, a probabilidade com a qual isso ocorre. Inicialmente, consideraremos somente valores X, Y, Z , para os quais vale $(X; Y) \rightarrow Z$ pela função S_i .

Definição 4.3 *Dado o par de triplas de bits $((x_1, y_1, c_1), (x_2, y_2, c_2))$, dizemos que ele satisfaz $C^{j+1} \in (X^j, Y^j, C^j = W^j \oplus X^j \oplus Y^j)$ para algum j , $0 \leq j \leq 7$, se:*

- (i) $X^j = x_1 \oplus x_2$,
- (ii) $Y^j = y_1 \oplus y_2$,
- (iii) $C^j = c_1 \oplus c_2$ e
- (iv) $C^{j+1} = ((x_1 + y_1 + c_1) \text{ div } 2) \oplus ((x_2 + y_2 + c_2) \text{ div } 2)$.

Definição 4.4 *Sejam α, β, γ e δ , seqüências de n bits. Dizemos que os pares (α, β) e (γ, δ) são distintos se:*

- (i) $\alpha \neq \gamma$ ou $\beta \neq \delta$ e
- (ii) $\alpha \neq \delta$ ou $\beta \neq \gamma$.

Definição 4.5 *Seja $((X_1; Y_1), (X_2; Y_2))$ um par de seqüências de n bits tal que n é par e $\#X_1 = \#X_2 = \#Y_1 = \#Y_2$. Dizemos que os pares de triplas de bits $((x_1, y_1, c_1), (x_2, y_2, c_2))$ e $((x_1^*, y_1^*, c_1^*), (x_2^*, y_2^*, c_2^*))$ são distintos se:*

- (i) c_1 é o valor de carry do bit $n/2-1$ na soma $X_1 + Y_1$,
- (ii) c_2 é o valor de carry do bit $n/2-1$ na soma $X_2 + Y_2$,
- (iii) $c_1 = c_1^*$,
- (iv) $c_2 = c_2^*$ e
- (v) os pares de seqüências de $n+2$ bits

$$((x_1; X_1; y_1; Y_1), (x_2; X_2; y_2; Y_2))$$

e

$$((x_1^*; X_1; y_1^*; Y_1), (x_2^*; X_2; y_2^*; Y_2))$$

são distintos.

Baseados nestas definições, vamos determinar o número de pares distintos cuja diferença no par satisfaz $(X; Y)$ e para os quais $(X; Y) \rightarrow Z$ pela função S_i .

Lema 4.11 *Seja N_j o número de pares distintos de triplas de bits que satisfazem (X^j, Y^j, C^j) e C^{j+1} , $0 \leq j \leq 7$. Então, o número de pares distintos cuja diferença no par satisfaz $(X; Y)$ e para os quais $(X; Y) \rightarrow Z$ pela função S_i é dado por:*

$$\prod_{j=0}^7 N_j$$

Dem.: Decorre do algoritmo Zero que um par satisfaz $(X; Y)$ e causa Z pela função S_i se, e somente se, (X^j, Y^j, C^j) e C^{j+1} são satisfeitos para todo j , $0 \leq j \leq 7$. Logo, o número de pares distintos que satisfazem essas condições é igual ao número de instâncias distintas que satisfazem (X^j, Y^j, C^j) e C^{j+1} , $0 \leq j \leq 7$. Portanto, como N_j é o número de pares de triplas de bits distintos que satisfazem (X^j, Y^j, C^j) e C^{j+1} , $0 \leq j \leq 7$, segue que o número de pares distintos cuja diferença no par satisfaz $(X; Y)$ e para os quais $(X; Y) \rightarrow Z$ pela função S_i é dado por $\prod_{j=0}^7 N_j$. $\square$

Ao estudarmos o comportamento de C^{j+1} a partir de X^j, Y^j, C^j , notamos que o valor de N_j é muito bem definido dentro de cada um dos casos. A tabela 4.2 mostra os valores de N_j para cada um desses casos.

<i>caso</i>	X^j	Y^j	C^j	$N_j(0 \leq j \leq 6)$	N_7
1	0	0	0	4	4
2	0	0	1	2	4
3	0	1	0	$1^{(*)}, 2^{(**)}$	$2^{(*)}, 4^{(**)}$
4	0	1	1	2	4
5	1	0	0	$1^{(*)}, 2^{(**)}$	$2^{(*)}, 4^{(**)}$
6	1	0	1	2	4
7	1	1	0	$1^{(*)}, 2^{(**)}$	$2^{(*)}, 4^{(**)}$
8	1	1	1	4	4

“(*) na primeira ocorrência de um dos casos 3, 5 ou 7”

“(**) após ocorrer um dos casos 3, 5 ou 7 ”

Tabela 4.2 Valores de N_j .

Os valores de N_7 e $N_j, 0 \leq j \leq 6$, diferem em alguns casos devido ao fato de que C^8 não é utilizado, ou seja, seu valor é irrelevante. Vamos agora, utilizando a tabela 4.2, formular um algoritmo que nos forneça o valor de N_j , dados X^j, Y^j, C^j e j .

Algoritmo ParesTriplas(X^j, Y^j, C^j, j, dif)

“Fornece o valor de N_j . *dif* indica se um dos casos 3, 5 ou 7 já ocorreu ou não.”

início

01 se $(X^j \wedge Y^j \wedge C^j = 1)$ ou $(\overline{X^j} \wedge \overline{Y^j} \wedge \overline{C^j} = 1)$ então

02 $N_j := 4$;

03 senão

04 se $(C^j = 1)$ então

05 se $(j = 7)$ então

06 $N_j := 4$;

07 senão

08 $N_j := 2$;

09 fim-se

10 senão

11 se $(j = 7)$ então

12 se não *dif* então

13 $N_j := 2$;

14 senão

15 $N_j := 4$;

16 fim-se

17 senão

```

18      se não dif então
19           $N_j := 1;$ 
20           $diff := verdadeiro;$ 
21      senão
22           $N_j := 2;$ 
23      fim-se
24  fim-se
25  fim-se
26  retorna( $N_j$ );
fim.

```

Lema 4.12 *O algoritmo ParesTriplas define corretamente o valor de N_j e sua complexidade é $O(1)$.*

Dem.: A corretude decorre do fato de que o algoritmo ParesTriplas é uma reprodução em linguagem formal da tabela 4.2. Já a complexidade decorre do algoritmo só possuir comandos que são executados em tempo constante. $\square$

De posse do algoritmo ParesTriplas e do lema 4.11 já temos as informações necessárias para calcularmos o número de pares distintos cuja diferença satisfaz $(X; Y)$ e que causam Z pela função S_i . Mas, para sabermos $P[(X; Y) \rightarrow Z \text{ por } S_i]$ devemos saber também o número de pares distintos cuja diferença satisfaz $(X; Y)$. O lema abaixo nos fornece um meio de calcular esse valor.

Definição 4.6 *Sejam X, X_1, X_2 , seqüências de n bits. Então, $P_n(X)$ é o conjunto de todos os pares distintos (X_1, X_2) tais que $X = X_1 \oplus X_2$.*

Lema 4.13 ¹*Seja X uma seqüência de n bits. Então, valem as seguintes afirmações:*
(i) $\#P_n(X) \in \{2^n, 2^{n-1}\}$,
(ii) $\#P_n(X) = 2^n \Leftrightarrow X = 0$.

Dem.: (Ver o apêndice do capítulo.) $\square$

Como a probabilidade de $(X; Y)$ causar Z pela função S_i é dada pela fração dos pares distintos cuja diferença satisfaz $(X; Y)$ e que causam Z pela função S_i em relação a todos os pares distintos cuja diferença satisfaz $(X; Y)$, ou seja:

$$P[(X; Y) \rightarrow Z \text{ por } S_i] = \frac{\#\{A : A \in P_{16}((X; Y)) \wedge (A \in P_{S_i}(Z))\}}{\#P_{16}((X; Y))}$$

¹Se A é um conjunto então $\#A$ denota a cardinalidade de A .

onde

$$P_{S_i}(Z) = \{(A; B) \in \{0, 1\}^{16} \times \{0, 1\}^{16} : Z = S_i(A) \oplus S_i(B)\}.$$

Como quando $(X; Y) \rightarrow Z$ pela função S_i essa probabilidade é maior que zero, podemos assumir que sempre ocorre $(X; Y) \rightarrow Z$ e para os casos onde $(X; Y) \rightsquigarrow Z$, dizemos que $(X; Y) \rightarrow Z$ pela função S_i com probabilidade zero.

Vamos então ao algoritmo que calcula $P[(X; Y) \rightarrow Z \text{ por } S_i]$.

Algoritmo Um

“Dados os bytes X, Y, Z , calcula a probabilidade de $(X; Y)$ causar Z pela função S_i .”

início

```

01  “Determinação de  $\#P_{16}((X; Y))$ ”
02  se  $((X; Y) = 0)$  então
03       $tot pares := 2^{16}$ ;
04  senão
05       $tot pares := 2^{15}$ ;
06  fim-se
07  “Inicialização das variáveis”
08   $W := (Rot2)^{-1}(Z)$ ;
09   $diff := falso$ ;
10   $c := 0$ ;
11   $j := 0$ ;
12   $acabou := falso$ ;
13   $cdefinido := verdadeiro$ ;
14   $num pares := 1$ ;
15  “Decide se  $(X; Y) \rightarrow Z$  ou  $(X; Y) \rightsquigarrow Z$ ”
16  enquanto não  $acabou$  faça
17      se  $cdefinido$  então
18          se  $(W^j = X^j \oplus Y^j \oplus c)$  então
19               $num pares := num pares * ParesTriplas(X^j, Y^j, c, j, diff)$ ;
20              se  $((X^j \wedge Y^j \wedge c = 0) \wedge (\overline{X^j} \wedge \overline{Y^j} \wedge \overline{c} = 0))$  então
21                   $cdefinido := falso$ ;
22              fim-se
23              senão “ $(X; Y) \rightsquigarrow Z$ ”
24                   $num pares := 0$ ;
25                   $acabou := verdadeiro$ ;
26              fim-se
27      senão
28           $c := W^j \oplus X^j \oplus Y^j$ ;
29           $num pares := num pares * ParesTriplas(X^j, Y^j, c, j, diff)$ ;
30          se  $((X^j \wedge Y^j \wedge W^j = 1) \text{ ou } (\overline{X^j} \wedge \overline{Y^j} \wedge \overline{c} = 1))$  então

```

```

31      cdefinido := verdadeiro;
32      c :=  $W^j$ ;
33      fim-se
34      fim-se
35       $j := j + 1$ ;
36      se ( $j = 8$ ) então
37          acabou := verdadeiro;
38      fim-se
39      fim-enquanto
40      “Cálculo de  $P[(X;Y) \rightarrow Z \text{ por } S_i]$ ”
41      prob := numpares/totpares;
fim.

```

Lema 4.14 *Dados os bytes X, Y, Z , o algoritmo Um calcula corretamente a probabilidade de $(X;Y)$ causar Z pela função S_i . Além disso, o algoritmo Um é um processo finito e sua complexidade é $O(1)$.*

Dem.: Observemos que o algoritmo Um contém toda a lógica do algoritmo Zero mais o cálculo da probabilidade. Como o algoritmo Zero decide corretamente se $(X;Y) \rightarrow Z$ ou $(X;Y) \rightsquigarrow Z$ pela função S_i , segue que o algoritmo Um também toma essa decisão corretamente. Além disso, o algoritmo Um atribui probabilidade zero a $(X;Y) \rightarrow Z$ se, e somente se, $(X;Y) \rightsquigarrow Z$, o que está de acordo com nossas observações feitas antes da elaboração do algoritmo Um.

Verifiquemos então o cálculo da probabilidade para o caso $(X;Y) \rightarrow Z$ pela função S_i . Pelo lema 4.13, a determinação do número total de pares distintos cuja diferença satisfaz $(X;Y)$ está correta.

A determinação do número de pares distintos que satisfazem $(X;Y)$ e que causam Z pela função S_i é feita de acordo com o lema 4.11. Assim, devemos mostrar que o número de pares distintos de triplas de bits é calculado corretamente a cada iteração. Como isso é calculado pelo algoritmo ParesTriplas, o qual já provamos estar correto, só nos resta verificar se os parâmetros utilizados por ParesTriplas estão sendo fornecidos corretamente. Os parâmetros necessários são $X^j, Y^j, C^j, j, 0 \leq j \leq 7$ e *dif*. Se C^j está definido então c contém o valor de C^j . Senão, o parâmetro fornecido como C^j é $X^j \oplus Y^j \oplus W^j$, o que está de acordo com o lema 4.4. Como os valores de X^j, Y^j, j e *dif* são conhecidos em toda iteração, segue que os parâmetros são fornecidos corretamente ao algoritmo de cálculo do número de pares distintos de triplas de bits.

A probabilidade é então obtida dividindo-se o número de pares distintos que satisfazem $(X;Y)$ e que causam Z pela função S_i pelo número de pares distintos que satisfazem $(X;Y)$. Logo, a probabilidade é calculada corretamente pelo algoritmo Um.

O algoritmo Um é um processo finito pelos mesmos motivos que o algoritmo Zero o é, como estabelecido no lema 4.9.

Como os comandos que o algoritmo Um possui a mais que o algoritmo Zero podem ser realizados em tempo constante, segue que a complexidade do algoritmo Um é a mesma que a do algoritmo Zero, a qual pelo lema 4.10, é $O(1)$. $\square$

Observemos que tanto o algoritmo Zero quanto o algoritmo Um não fazem uso do valor de i . Logo, podemos concluir que:

Lema 4.15 *Sejam X, Y, Z , bytes. Então:*

(i) $(X; Y) \rightarrow Z$ pela função S_0 se, e somente se, $(X; Y) \rightarrow Z$ pela função S_1 .

(ii) $P[(X; Y) \rightarrow Z \text{ por } S_0] = P[(X; Y) \rightarrow Z \text{ por } S_1]$.

Com base nesse lema, daqui por diante nos referiremos à função S_i simplesmente por S . Na próxima seção, veremos algumas propriedades das funções S e f relacionadas com as diferenças nos pares.

4.3 Propriedades das funções S e f

Veremos nesta seção algumas propriedades das funções S e f relacionadas com a diferença entre os elementos de um par. Como a função S está embutida na função f , iniciaremos a exposição pela função S , pois assim poderemos utilizar algumas dessas propriedades para obtermos resultados para a função f . Se X, Y, Z são bytes então denotaremos a probabilidade de $(X; Y)$ causar Z pela função S por $p_S(X, Y, Z)$. Se X e Y são seqüências de 32 bits então denotaremos a probabilidade de X causar Y pela função f por $p_f(X, Y)$.

Lema 4.16 *Se X, Y, Z são bytes quaisquer então:*

$$p_S(X, Y, Z) = p_S(Y, X, Z).$$

Dem.: Decorre do algoritmo Um.

Lema 4.17 *Se X, Y, Z são bytes quaisquer então $p_S(X, Y, Z)$ é zero ou uma potência de $\frac{1}{2}$.*

Dem.: Se $(X; Y) \rightsquigarrow Z$ pela função S então $p_S(X, Y, Z) = 0$. Senão:

$$p_S(X, Y, Z) = \frac{\prod_{j=0}^7 N_j}{\#P_{16}((X; Y))}.$$

Como, pela tabela 4.2, $N_j \in \{1, 2, 4\}$ e, pelo lema 4.13, $\#P_{16} \in \{2^{15}, 2^{16}\}$, segue o resultado. $\square$

Lema 4.18 $p_S(\alpha, 0, 0) \neq 0 \Leftrightarrow \alpha = 0$.

Dem.: $(\Rightarrow)$ Se $p_S(\alpha, 0, 0) \neq 0$ então $(\alpha, 0) \rightarrow 0$ pela função S , ou seja:

$$C^j = \alpha^j$$

para todo $j, 0 \leq j \leq 7$. Como $C^0 = 0$ e $C^{j+1} = 0$ se $C^j = \alpha^j = 0$, segue que se $p_S(\alpha, 0, 0) \neq 0$ então $\alpha = 0$.

$(\Leftarrow)$ Se $\alpha = 0$ então $p_S(\alpha, 0, 0) = p_S(0, 0, 0) = 1$, pois, claramente, entradas iguais produzem sempre saídas iguais. $\square$

Lema 4.19 $p_S(X, Y, Z) = 1$ se, e somente se:

- (i) $X = Y = Z = 00_x$, ou
- (ii) $X = Y = 80_x$ e $Z = 00_x$, ou
- (iii) $X = 80_x, Y = 00_x$ e $Z = 02_x$, ou
- (iv) $X = 00_x, Y = 80_x$ e $Z = 02_x$.

Dem.: Claramente $p_S(0, 0, 0) = 1$, pois entradas iguais produzem saídas iguais. Pelo algoritmo Um, se na primeira iteração $j, 0 \leq j \leq 7$, onde não ocorre o caso 1, deve ocorrer um dos casos 3, 5 ou 7, pois $C^j = 0$. Agora, se um desses casos ocorre para j em $\{0, \dots, 6\}$, o valor de N_j é 1. Como $N_j \leq 4$ e $(X, Y) \neq 0$, segue que $p_S(X, Y, Z) \leq \frac{1}{2}$.

Portanto, para que tenhamos $p_S(X, Y, Z) = 1$, os casos 3, 5 ou 7 só podem ocorrer na última iteração do algoritmo Um, donde segue o lema. $\square$

Agora, vamos aplicar o algoritmo Um para obtermos algumas probabilidades $p_S(X, Y, Z)$ que nos serão úteis futuramente. Em todos os casos, estaremos calculando as probabilidades por:

$$p_S(X, Y, Z) = \frac{\prod_{j=0}^7 N_j}{\#P_{16}((X; Y))}$$

e $\oplus$ *carry* denota o valor do ou-exclusivo entre os "carry's".

Lema 4.20 $p_S(A0_x, 80_x, 80_x) = \frac{1}{2}$.

Dem.: $(Rot2)^{-1}(80_x) = 20_x$. Aplicando o algoritmo Um:

j	$A0_x$	80_x	$\oplus carry$	20_x	N_j
0	0	0	0	0	4
1	0	0	0	0	4
2	0	0	0	0	4
3	0	0	0	0	4
4	0	0	0	0	4
5	1	0	0	1	1
6	0	0	0	0	4
7	1	1	0	0	4

Assim, $p_S(A0_x, 80_x, 80_x) = \frac{2^{14}}{2^{15}} = \frac{1}{2}$. □

Lema 4.21 $p_S(E0_x, 80_x, 80_x) = \frac{1}{4}$.

Dem.: $(Rot2)^{-1}(80_x) = 20_x$. Aplicando o algoritmo Um:

j	$E0_x$	80_x	$\oplus carry$	20_x	N_j
0	0	0	0	0	4
1	0	0	0	0	4
2	0	0	0	0	4
3	0	0	0	0	4
4	0	0	0	0	4
5	1	0	0	1	1
6	1	0	1	0	2
7	1	1	0	0	4

Assim, $p_S(E0_x, 80_x, 80_x) = \frac{2^{13}}{2^{15}} = \frac{1}{4}$. □

Lema 4.22 $p_S(60_x, 80_x, 80_x) = \frac{1}{4}$.

Dem.: $(Rot2)^{-1}(80_x) = 20_x$. Aplicando o algoritmo Um:

j	60_x	80_x	$\oplus carry$	20_x	N_j
0	0	0	0	0	4
1	0	0	0	0	4
2	0	0	0	0	4
3	0	0	0	0	4
4	0	0	0	0	4
5	1	0	0	1	1
6	1	0	1	0	2
7	0	1	1	0	4

Assim, $p_S(60_x, 80_x, 80_x) = \frac{2^{13}}{2^{15}} = \frac{1}{4}$. □

Definição 4.7 *Sejam $\alpha = (\alpha_1, \alpha_2)$ e $\beta = (\beta_1, \beta_2)$, pares de seqüências de n bits. Dizemos que α e β são diferentes se $\alpha_1 \neq \beta_1$ ou $\alpha_2 \neq \beta_2$.*

Lema 4.23 *No FEAL se $X \rightarrow Y$ pela função f com probabilidade $p_f(X, Y)$ então todo par de entrada fixado Z, Z^* tal que $Z' = X$ causa o valor da diferença de saída da função f ser igual a Y pela mesma fração $p_f(X, Y)$ dos valores possíveis para a subchave real.*

Dem.: Seja f^P a função f com subchave nula. Para provarmos este lema, basta mostrarmos que se existem k pares diferentes (X_1, X_2) tais que $X_1 \oplus X_2 = X$ e $f^P(X_1) \oplus f^P(X_2) = Y$, então existem exatamente k valores diferentes para a subchave real, pois o número de pares diferentes cuja diferença satisfaz X é igual ao número total de subchaves reais.

Pela estrutura da função f , podemos escolher precisamente k valores para a subchave real AK_i , pois:

$$AK_i = Z \oplus X_1 = Z^* \oplus X_2.$$

Assim, a fração $p_f(X, Y)$ é mantida constante para todos os pares de entrada. □

Lema 4.24

$$p_f(X, Y) = p_S(X_0, Y_1, Y_0) \cdot p_S(X_0 \oplus X_1, X_2 \oplus X_3, Y_1) \cdot p_S(X_2 \oplus X_3, Y_1, Y_2) \cdot p_S(X_3, Y_2, Y_3).$$

Dem.: Decorre do lema 4.23 e da estrutura da função f . □

Uma consequência interessante dos lemas 4.24 e 4.18 é que $p_f(X, 0) \neq 0$ se, e somente se, $X = 0$, o que implica que não existe característica iterativa de duas iterações para o FEAL que não seja a trivial.

A seguir, enunciaremos uma série de resultados que são consequências do lema 4.24 e de propriedades da função S . Tais resultados serão úteis para a compreensão das probabilidades das características do FEAL. Os resultados são os seguintes:

- (i) $p_f(A0\ 00\ 80\ 00_x, 80\ 80\ 00\ 00_x) = \frac{1}{4}$,
 - (ii) $p_f(00\ 00\ 00\ 00_x, 00\ 00\ 00\ 00_x) = 1$,
 - (iii) $p_f(80\ 80\ 80\ 80_x, 02\ 00\ 00\ 02_x) = 1$,
 - (iv) $p_f(80\ 80\ 00\ 00_x, 02\ 00\ 00\ 00_x) = 1$,
 - (v) $p_f(00\ 00\ 80\ 80_x, 00\ 00\ 00\ 02_x) = 1$,
 - (vi) $p_f(80\ 60\ 80\ 00_x, 00\ 80\ 00\ 00_x) = \frac{1}{4}$,
 - (vii) $p_f(80\ E0\ 80\ 00_x, 00\ 80\ 00\ 00_x) = \frac{1}{4}$,
 - (viii) $p_f(A2\ 00\ 80\ 00_x, XY\ 88\ 20\ 8Z_x) = \frac{1}{8}$,
- onde $X \in \{5, 6, 7, 9, A, B, D, E, F\}$, $Y \in \{9, A, B\}$ e $Z \in \{0, 1, 3\}$.

Lema 4.25 *Se X e Y são seqüências de 32 bits fixadas então $p_f(X, Y)$ é zero ou uma potência de $\frac{1}{2}$.*

Dem.: Decorre dos lemas 4.17 e 4.24. □

Na próxima seção, veremos algumas características existentes para o FEAL.

4.4 Características para o FEAL-N(X)

São apresentadas aqui as principais características publicadas para os ataques diferenciais à família de cifras FEAL. Para facilitar nosso trabalho, daremos um nome a cada uma dessas características. Assim, a característica n-k é a k-ésima característica de n iterações apresentada nesta seção. A apresentação das características não leva em consideração o ou-exclusivo entre as metades do texto legível existente no FEAL para manter a compatibilidade com a definição de característica iterativa. As figuras 4.1, 4.2, 4.3 e 4.4 trazem as características de uma iteração que possuem probabilidade 1.

Das propriedades da função S e da estrutura da função f , segue que para um L' fixo estas quatro características são as únicas características de uma iteração com probabilidade 1 existentes para o FEAL. A partir destas quatro características, podemos formar as dezesseis características de duas iterações que possuem probabilidade 1 e as quatro características de três iterações com probabilidade 1 existentes para a cifra. A figura 4.5 exhibe a lei de formação dessas características de duas iterações e a lei de formação das características de três iterações cuja probabilidade é 1 aparece na figura 4.6.

Figura 4.1 ([BS91a]) *Característica 1-1.*

Figura 4.2 ([BS91a]) *Característica 1-2.*

Figura 4.3 *Característica 1-3.*

Figura 4.4 *Característica 1-4.*

onde R' e L' são elementos do conjunto:

$$\{0_x, 80800000_x, 00008080_x, 80808080_x\}$$

e $R' \rightarrow X'$ pela função f com probabilidade 1

e $L' \rightarrow Y'$ pela função f com probabilidade 1.

Figura 4.5 Lei de formação das 16 características de duas iterações que possuem probabilidade 1.

onde

$$R' \in \{0_x, 80800000_x, 00008080_x, 80808080_x\}$$

e $R' \rightarrow X'$ pela função f com probabilidade 1.

Figura 4.6 Lei de formação das 4 características de três iterações que possuem probabilidade 1.

onde, $X \in \{5, 6, 7, 9, A, B, D, E, F\}$, $Y \in \{9, A, B\}$, $Z \in \{0, 1, 3\}$ e $W = X \oplus 8$.

Figura 4.7 ([BS91a]) Característica 6-1.

Lema 4.26 *No FEAL, não existe uma característica de quatro iterações com probabilidade 1 que não seja a trivial.*

Dem.: Suponhamos que exista uma característica Ω de quatro iterações tal que $\Omega_L \neq 0$ e $p^\Omega = 1$. Como $\Omega_L \neq 0$, segue que as três primeiras iterações de Ω formam uma das três características de três iterações que possuem probabilidade 1 e não são triviais. Seja X um elemento qualquer de

$$\{02000000_x, 00000002_x, 02000002_x\},$$

o conjunto de valores possíveis de λ_S^3 . Como $\lambda_{En}^2 = 0_x$, segue que

$$\lambda_{En}^4 = \lambda_{En}^2 \oplus \lambda_S^3 = 0_x \oplus X = X.$$

Dado que $p^\Omega = 1$, temos $p_f(X, Y) = 1$, para algum Y em $\{0, 1\}^{32}$, o que, de acordo com o lemas 4.19 e 4.24, é absurdo. Portanto, a única característica de quatro iterações que possui probabilidade 1 é a trivial. $\square$

O FEAL tem também uma característica de cinco iterações com probabilidade $\frac{1}{16}$, a característica 5-1, e uma característica de 6 iterações com probabilidade $\frac{1}{128}$, característica 6-1, a qual aparece na figura 4.7. Notemos que a característica 5-1 é formada pelas cinco primeiras iterações da característica 6-1.

A característica 6-1 é utilizada por Biham e Shamir no ataque ao FEAL-8 utilizando apenas 1000 pares de textos cifrados. Este ataque, implementado em um COMPAQ, encontra a chave em menos de dois minutos com mais de 95% de sucesso. Com 2000 pares, a chave é encontrada com sucesso em quase 100% dos casos. O programa utiliza 280Kb de memória[BS91a].

Para um número arbitrário de iterações, o ataque ao FEAL utiliza uma característica iterativa que possui probabilidade $\frac{1}{4}$ por iteração. Assim, uma característica de n iterações com probabilidade 2^{-2n} sempre pode ser obtida, qualquer que seja o valor de n . A figura 4.8 mostra a característica iterativa.

Na próxima seção, veremos como os ataques dos tipos 1R e 2R podem ser aplicados ao FEAL-N(X) e que a criptoanálise diferencial é mais rápida do que a busca exaustiva para $N < 32$.

Figura 4.8 ([BS91a]) *Característica iterativa.*

4.5 Os ataques $1R$ e $2R$

Como a única diferença entre o FEAL-N e o FEAL-NX é o processamento da chave secreta e a Criptoanálise Diferencial ignora a geração das subchaves, encontrando as subchaves reais, conclui-se que um ataque diferencial quando aplicado ao FEAL-NX possui a mesma complexidade e performance desse ataque em relação ao FEAL-N. Por este motivo, estaremos sempre nos referindo ao FEAL-N ficando subentendido que o mesmo é válido para o FEAL-NX.

Quanto às características, os dois tipos de ataque utilizam quatro características, as quais são as quatro rotações possíveis da característica iterativa exibida na figura 4.8. Como o valor da diferença no par de textos legíveis em qualquer uma delas é igual ao ou-exclusivo entre os valores da diferença no par de textos legíveis das outras três, podemos fazer uso de octetos especiais para armazenar os pares relativos a essas quatro características. Assim, cada octeto nos fornece 16 pares de textos, o que reduz bastante o número necessário de textos legíveis escolhidos[BS91a].

Em ambos os ataques, existem dois bits indistinguíveis nas duas últimas subchaves reais. Esses dois bits são o bit mais significativo dos bytes 0 e 3 de cada uma dessas subchaves. Deste modo, o programa de ataque deve tentar todos os dezesseis valores possíveis desses bits quando for analisar as subchaves anteriores, o que pode ser feito em paralelo. Além disso, sempre existem dois valores para as subchaves reais de 16 bits porque se a chave $(K_0; K_1)$ é sugerida pelas diferenças, então a chave $(K_0 \oplus 80_x; K_1 \oplus 80_x)$ também é sugerida[BS91a].

O processo de contagem da última subchave real é feito em duas etapas. Na primeira etapa, encontra-se a última subchave real de 16 bits e, na segunda etapa, encontra-se os outros dois bytes da última subchave real.

Num ataque do tipo $1R$, conhecemos o valor da diferença no par de textos cifrados C' e, $E'(N)$ e $E'(N-1)$ pela característica. Podemos verificar se o valor de $d' \oplus e'$ é igual ao valor de $E'(N)$, e se $E'(N) \rightarrow SS'(N) = E'(N-1) \oplus e'$ pela função f . A triagem $E'(N) = d' \oplus e'$ deixa apenas 2^{-32} dos pares errados, enquanto que $E'(N) \rightarrow SS'(N)$ deixa 2^{-19} , pois quer seja $E'(N)$ igual a 80608000_x ou $80E08000_x$, existem aproximadamente 2^{13} valores possíveis para $SS'(N)$. Assim, apenas 2^{-51} pares errados conseguem passar por essas duas triagens[BS91a].

Os pares restantes são então utilizados no processo de contagem da última subchave real. Encontrada a última subchave real, os textos cifrados podem ser decifrados parcialmente, apenas uma iteração, reduzindo assim o problema à quebra do FEAL-(N-1). Para encontrar as outras subchaves, procede-se de maneira análoga, utilizando características com menos iterações, ou seja, com probabilidades mais altas[BS91a].

Algumas desvantagens desse tipo de ataque são o pequeno número de bits iguais a 1 em $E'(N)$, o qual é uma constante definida pela característica, e o fato de que muitos valores de $SS'(N)$ sugerem muitos valores iguais para a última subchave real[BS91a]. Em

seguida, mostraremos que um ataque do tipo $1R$ aplicado ao FEAL- N é mais rápido do que a busca exaustiva para qualquer $N < 32$.

Lema 4.27 *O ataque diferencial do tipo $1R$ aplicado ao FEAL- N é mais rápido do que a busca exaustiva para $N < 32$.*

Dem.: A proporção sinal-ruído para um ataque do tipo $1R$ ao FEAL, conforme a fórmula 3.1, é dada por:

$$\begin{aligned} (S/R)_{1R} &= \frac{p \cdot 2^k}{\alpha \cdot \beta} \\ &\approx \frac{2^{-2(N-1)} \cdot 2^{16}}{1 \cdot \left(\frac{1}{2^{2(N-1)}} + \frac{1}{2^{51}} \right)} \\ &= \frac{2^{67}}{2^{51} + 2^{2(N-1)}}. \end{aligned}$$

Para $N \leq 30$, $(S/R)_{1R} \geq 2^9$ o que implica que aproximadamente 8 pares certos são necessários ao processo de contagem[BS91a], ou seja, cada característica exige 2 pares certos. Com quatro características, são necessários 2^{2N+1} pares no total, ou seja, 2^{2N} textos legíveis devem ser criados em forma de octetos especiais. Assim, para $N \leq 30$, a complexidade do ataque é no máximo 2^{60} .

Para $N = 31$, $(S/R)_{1R} \approx 2^7$ o que implica que aproximadamente 2^4 pares certos são necessários para a contagem[BS91a]. Pelas observações acima, a complexidade do ataque é 2^{63} .

Para $N = 32$, $(S/R)_{1R} \approx 2^5$ e 2^5 são necessários[BS91a], o que nos dá uma complexidade de 2^{66} .

Como o valor da proporção sinal-ruído decresce quando o valor de N aumenta, segue que para $N \geq 32$ a complexidade desse ataque é maior do que 2^{64} . □

Em ataques do tipo $2R$, conhecemos o valor da diferença no par de textos cifrados C' , logo, conhecemos $E'(N)$, e $E'(N-1)$ e $E'(N-2)$ pela característica. Assim,

$$SS'(N-1) = E'(N) \oplus E'(N-2) \text{ e } SS'(N) = e' \oplus E'(N-1).$$

Verifica-se se cada par satisfaz $E'(N-1) \rightarrow SS'(N-1)$ e $E'(N) \rightarrow SS'(N)$ pela função f . Os pares que passam por essas triagens são então utilizados no processo de contagem. Os ataques do tipo $2R$ são mais rápidos do que a busca exaustiva para $N \leq 28$ [BS91a]. A tabela 4.3 mostra um resumo dos ataques diferenciais ao FEAL.

De posse dos valores das subchaves reais, pode-se derivar, caso se queira, o valor da chave secreta utilizada nos ciframentos pelo método dado por Bert Den Boer[BDB88].

N	ataques $2R$			ataques $1R$		
	Prob	Pares	Dados	Prob	Pares	Dados
8	2^{-12}	2^{14}	2^{13}	2^{-14}	2^{17}	2^{16}
9	2^{-14}	2^{16}	2^{15}	2^{-16}	2^{19}	2^{18}
10	2^{-16}	2^{18}	2^{17}	2^{-18}	2^{21}	2^{20}
11	2^{-18}	2^{20}	2^{19}	2^{-20}	2^{23}	2^{22}
12	2^{-20}	2^{22}	2^{21}	2^{-22}	2^{25}	2^{24}
13	2^{-22}	2^{24}	2^{23}	2^{-24}	2^{27}	2^{26}
14	2^{-24}	2^{26}	2^{25}	2^{-26}	2^{29}	2^{28}
15	2^{-26}	2^{28}	2^{27}	2^{-28}	2^{31}	2^{30}
16	2^{-28}	2^{30}	2^{29}	2^{-30}	2^{33}	2^{32}
17	2^{-30}	2^{32}	2^{31}	2^{-32}	2^{35}	2^{34}
18	2^{-32}	2^{34}	2^{33}	2^{-34}	2^{37}	2^{36}
19	2^{-34}	2^{36}	2^{35}	2^{-36}	2^{39}	2^{38}
20	2^{-36}	2^{38}	2^{37}	2^{-38}	2^{41}	2^{40}
21	2^{-38}	2^{40}	2^{39}	2^{-40}	2^{43}	2^{42}
22	2^{-40}	2^{42}	2^{41}	2^{-42}	2^{45}	2^{44}
23	2^{-42}	2^{44}	2^{43}	2^{-44}	2^{47}	2^{46}
24	2^{-44}	2^{46}	2^{45}	2^{-46}	2^{49}	2^{48}
25	2^{-46}	2^{49}	2^{48}	2^{-48}	2^{51}	2^{50}
26	2^{-48}	2^{52}	2^{51}	2^{-50}	2^{53}	2^{52}
27	2^{-50}	2^{55}	2^{54}	2^{-52}	2^{55}	2^{54}
28	2^{-52}	2^{58}	2^{57}	2^{-54}	2^{57}	2^{56}
29	2^{-54}			2^{-56}	2^{59}	2^{58}
30	2^{-56}			2^{-58}	2^{61}	2^{60}
31	2^{-58}			2^{-60}	2^{64}	2^{63}
32	2^{-60}			2^{-62}	2^{67}	2^{66}

Tabela 4.3 ([BS91a]) Ataques diferenciais ao FEAL- N .

4.6 Ataques diferenciais do tipo texto conhecido

Os ataques da Criptoanálise Diferencial são ataques do tipo texto escolhido nos quais os pares de texto legível podem ser escolhidos aleatoriamente, contanto que satisfaçam o valor da diferença no par. Ao contrário de outros ataques do tipo texto escolhido, os ataques diferenciais podem ser facilmente convertidos em ataques do tipo texto conhecido pelo seguinte motivo. Suponhamos que o ataque do tipo texto escolhido necessite de m pares e que nos sejam dados $2^{32}\sqrt{2m}$ textos legíveis conhecidos aleatórios e seus respectivos textos cifrados. Consideremos todos os pares possíveis de textos legíveis que esses textos podem formar. Como o tamanho do bloco do FEAL é 64 bits, existem apenas 2^{64} valores possíveis para a diferença nos pares e, assim, existem aproximadamente:

$$\frac{1}{2^{64}} \cdot \frac{(2^{32}\sqrt{2m})^2}{2} = \frac{2^{64} \cdot m}{2^{64}} = m.$$

pares criando cada valor da diferença. Em particular, existem aproximadamente m pares para cada um dos valores da diferença necessários para a Criptoanálise Diferencial com alta probabilidade[BS91a].

Os ataques do tipo texto conhecido não são limitados ao modo de operação ECB - Electronic Code Book. Em particular, o modo CBC - Cipher Block Chaining - também pode ser quebrado, pois quando os textos legível e cifrado são conhecidos é fácil calcular o valor real da entrada da função de ciframento. A tabela 4.4 sintetiza os resultados dos ataques diferenciais do tipo texto conhecido ao FEAL[BS91a].

Cifra	Número de pares para uma carac.	Número de textos escolhidos	Número de textos conhecidos
FEAL-4	4	8	$2^{33.5}$
FEAL-8	1000	2000	$2^{37.5}$
FEAL-16	2^{28}	2^{29}	$2^{46.5}$
FEAL-24	2^{44}	2^{45}	$2^{54.5}$
FEAL-30	2^{59}	2^{60}	2^{62}
FEAL-31	2^{62}	2^{63}	$2^{63.5}$

Tabela 4.4 ([BS91a]) Ataques diferenciais tipo texto conhecido ao FEAL.

4.7 Conclusões

Neste capítulo procuramos apresentar a Criptoanálise Diferencial do FEAL de uma forma mais detalhada do que a da publicação original[BS91a] com o intuito de proporcionarmos um melhor entendimento do assunto ao leitor. Assim, apresentamos os algoritmos Zero e Um, os quais, além de serem muito interessantes por si só, nos permitiram derivar propriedades para as funções S e f que serão muito úteis no próximo capítulo. Biham e Shamir apenas mencionam em sua publicação que para o FEAL não é necessário fazer consultas a tabelas para decidir se uma dada diferença pode causar outra pela função S . Também a partir do algoritmo Zero, ficou claro porque para a Criptoanálise Diferencial os dois tipos de S-box existentes na cifra se comportam como se fossem um único. Algumas pessoas podem se questionar sobre o que aconteceria se o valor de i na função S_i fosse diferente de zero ou um. Afinal de contas, como a única restrição imposta pelos projetistas é que a cifra somente possua operações sobre bytes, por que não utilizar um outro valor qualquer de i no intervalo $[2,255]$. Através de simulações em computador utilizando as características apresentadas neste capítulo e outros valores de i para as S-boxes do FEAL, não encontramos nenhum valor de i que reduzisse ou aumentasse a probabilidade das mesmas, o que nos leva a conjecturar que para a Criptoanálise Diferencial $S_i = S_j$ quaisquer que sejam os valores de i e j em $[0,255]$.

As leis de formação das características que possuem probabilidade um foi estabelecida e mostramos que para o FEAL não existe característica não trivial que possui essa probabilidade para quatro iterações ou mais. Tentamos explicar também, através do conceito de sinal-ruído, porque um ataque diferencial do tipo 1R baseado na característica iterativa de Biham e Shamir consegue ser mais rápido do que a busca exaustiva quando o número de iterações da cifra é menor ou igual a 31.

Esperamos que, com isso, tenhamos ajudado um pouco as pessoas que pretendem estudar o assunto.

4.8 Apêndice

Neste apêndice, demonstramos o lema 4.13 de uma maneira bastante detalhada, pois acreditamos que ele nos possibilita entender melhor a formação dos pares distintos que satisfazem uma certa diferença e, conseqüentemente, porque os valores de N_j mudam de acordo com os valores de X^j, Y^j e C^j .

Definição 4.8 *Sejam X, X_1, X_2 , seqüências de n bits. Então, $P_n(X)$ é o conjunto de todos os pares distintos (X_1, X_2) tais que $X = X_1 \oplus X_2$.*

Lema 4.28 *Sejam X, X_1 , seqüências de n bits. Então, os elementos de $P_n(X)$ são da forma (X_1, X_1) se, e somente se, $X = 0$.*

Dem.: ($\Rightarrow$) Se os elementos de $P_n(X)$ são da forma (X_1, X_1) então $X = X_1 \oplus X_1 = 0$.
($\Leftarrow$) Seja (X_1, X_2) um elemento de $P_n(X)$. Se $X = 0$ então $X_1 \oplus X_2 = 0$, ou seja, $X_1 = X_2$. □

Lema 4.29 *Sejam X, X_1, X_2 , seqüências de n bits. Então, todos os elementos de $P_n(X)$ ou são da forma (X_1, X_1) ou são da forma (X_1, X_2) com $X_1 \neq X_2$.*

Dem.: Decorre do lema 4.28. □

Lema 4.30 *Seja X uma seqüência de n bits. Então:*

$$\#P_{n+1}((0; X)) = 2\#P_n(X).$$

Dem.: Seja $(X_1, X_2) \in P_n(X)$. Claramente, $((0; X_1), (0; X_2))$ e $((1; X_1), (1; X_2))$ são os únicos elementos de $\#P_{n+1}((0; X))$ obtidos a partir de (X_1, X_2) . Como, por definição, todos os elementos de $P_n(X)$ são distintos, segue que $P_{n+1}((0; X))$ tem o dobro de elementos de $P_n(X)$, ou seja,

$$\#P_{n+1}((0; X)) = 2\#P_n(X). \quad \square$$

Notemos que o lema 4.30 independe da forma dos elementos de $P_n(X)$.

Lema 4.31 *Se $X = 0$ é uma seqüência de n bits então:*

$$\#P_{n+1}((1; X)) = \#P_n(X).$$

Dem.: Como $X = 0$, segue do lema 4.28 que todos os elementos de $P_n(X)$ são da forma (X_1, X_1) . Assim, seja (X_1, X_1) um elemento de $P_n(X)$. Então, $((0; X_1), (1; X_1))$ e $((1; X_1), (0; X_1))$ são os únicos pares de seqüências de $n+1$ bits gerados a partir de (X_1, X_1) e que satisfazem $(1; X)$. Como esses pares não são distintos, segue que de um elemento de $P_n(X)$ obtemos um único elemento de $P_{n+1}((1; X))$. Logo,

$$\#P_{n+1}((1; X)) = \#P_n(X).$$

□

Lema 4.32 *Seja X uma seqüência de n bits. Se $X \neq 0$ então:*

$$\#P_{n+1}((1; X)) = 2\#P_n(X).$$

Dem.: Como $X \neq 0$, segue dos lemas 4.28 e 4.29 que todos os elementos de $P_n(X)$ são da forma (X_1, X_2) , $X_1 \neq X_2$. Claramente, $((0; X_1), (1; X_2))$ e $((1; X_1), (0; X_2))$ são os únicos elementos de $P_{n+1}((1; X))$ gerados a partir de um elemento de $P_n(X)$. Portanto,

$$\#P_{n+1}((1; X)) = 2\#P_n(X).$$

□

Lema 4.33 *Seja X uma seqüência de n bits. Se $X = 0$ então $\#P_n(X) = 2^n$.*

Dem.: (Por indução em n)

Base($n=1$): $P_1(0) = \{(0, 0), (1, 1)\} \Rightarrow \#P_1(0) = 2 = 2^1$.

Hipótese: Suponhamos que o lema seja válido para $n \geq 1$.

Passo:

$$\begin{aligned} \#P_{n+1}(0) &= \#P_{n+1}((0; X)) \\ &= 2\#P_n(X), \quad \text{pelo lema 4.30} \\ &= 2 \cdot 2^n, \quad \text{pela hipótese de indução} \\ &= 2^{n+1}. \end{aligned}$$

□

Lema 4.34 *Seja X uma seqüência de n bits. Se $X \neq 0$ então $\#P_n(X) = 2^{n-1}$.*

Dem.: Como $X \neq 0$, segue que existe $j \in \{0, 1, \dots, n-1\}$ tal que $X^j = 1$. Assim, seja k tal que:

$$k = \min \{j : (X^j = 1) \wedge (j \in \{0, 1, \dots, n-1\})\}.$$

Se $k = 0$ então $P_{k+1}(1) = \{(0, 1)\}$ e $\#P_{k+1} = 2^k$. Senão, seja Y a seqüência formada pelos k bits menos significativos de X . Claramente, $Y = 0$. Pelo lema 4.31, $\#P_{k+1}((1; Y)) = \#P_k(Y)$, e, pelo lema 4.33, $\#P_{k+1}((1, Y)) = 2^k$.

Se $k = n - 1$ então o lema está demonstrado. Senão, suponhamos que o lema seja válido para $P_{k+j}(X[k+j])$, $1 \leq j$. onde $X[k+j]$ é a sequência formada pelos $k+j+1$ bits menos significativos de X .

Então:

$$\#P_{k+j+1}(X[k+j+1]) = \#P_{k+j+1}((X^{k+j}; X[k+j])).$$

Se $X^{k+j} = 0$ então, pelo lema 4.30, temos:

$$\#P_{k+j+1}((X^{k+j}; X[k+j])) = 2\#P_{k+j}(X[k+j]).$$

Senão, como $X[k+j] \neq 0$, pelo lema 4.32, temos:

$$\#P_{k+j+1}((X^{k+j}; X[k+j])) = 2\#P_{k+j}(X[k+j]).$$

Pela hipótese de indução, $\#P_{k+j}(X[k+j]) = 2^{k+j-1}$.

Portanto,

$$\#P_{k+j+1}(X[k+j+1]) = 2^{k+j}.$$

□

Lema 4.35 Se $\#P_n(X) = 2^n$ então $X = 0$.

Dem.: Suponhamos que $\#P_n(X) = 2^n$ e $X \neq 0$. Como $X \neq 0$, pelo lema 4.34, segue que $\#P_n(X) = 2^{n-1}$, o que é uma contradição.

□

Sintetizando os resultados obtidos nos lemas 4.33, 4.34 e 4.35, temos:

Lema 4.36 Seja X uma sequência de n bits. Então, valem as seguintes afirmações:

- (i) $\#P_n(X) \in \{2^n, 2^{n-1}\}$,
- (ii) $\#P_n(X) = 2^n \Leftrightarrow X = 0$.

Dem.: (i) Segue dos lemas 4.33 e 4.34.

(ii)($\Rightarrow$) Pelo lema 4.35. ($\Leftarrow$) Pelo lema 4.33.

□

Capítulo 5

Uma proposta de fortalecimento contra a Criptoanálise Diferencial

O modo mais simples de se fortalecer uma cifra de Feistel contra a Criptoanálise Diferencial consiste em aumentar o seu número de iterações. Essa modificação, entretanto, provoca um aumento considerável do tempo de ciframento(deciframento) do sistema criptográfico, o que a torna não muito atraente. Por este motivo, ultimamente têm surgido propostas de fortalecimento visando cifras específicas, tais como PES[LM91], uma nova versão de LOKI[BKPS91], RDES[KT93] e SWDES[TG93]. Neste capítulo, apresentaremos uma nova versão do FEAL-N(X)¹, o FEAL-N(X)/RP, que possui velocidade de ciframento muito próxima à velocidade do modelo original e que oferece maior resistência contra a criptoanálise de Biham e Shamir.

5.1 FEAL-N(X)/RP: O FEAL-N(X) com redução de pares

Como visto no capítulo 3, dados uma característica $\Omega = (\Omega_L, \Omega_A, \Omega_C)$ com probabilidade p^Ω e um par de textos legíveis (L, L^*) tal que $\Omega_L = L \oplus L^*$ então a probabilidade desse par ser um par certo em relação a Ω é p^Ω . Assim, é natural tentarmos diminuir o número de pares certos em relação a uma característica para fortalecermos uma cifra contra a Criptoanálise Diferencial, pois fazendo isso estaremos diminuindo a probabilidade da característica. A seguir, veremos que isso pode ser conseguido pela introdução de uma função de troca na metade direita do bloco de entrada de cada iteração do FEAL-N(X).

¹A notação FEAL-N(X) indica que a cifra pode ser tanto FEAL-N quanto FEAL-NX.

Definição 5.1 *Seja X uma seqüência de 32 bits. Então X_E é a seqüência formada pelos 16 bits mais significativos de X e X_D é a seqüência formada pelos 16 bits menos significativos de X . Se $X_E = X_D$ então diremos que X é simétrica.*

Definição 5.2 *Seja $X = (X_E; X_D)$ uma seqüência de 32 bits. Definimos a função troca Tr por:*

$$Tr(X) = \begin{cases} (X_E; X_D), & \text{se o número de bits iguais a 1 em } X \text{ é par} \\ (X_D; X_E), & \text{c.c.} \end{cases}$$

Definição 5.3 *Sejam X , uma seqüência de 32 bits, e κ , um bit. Definimos a função T como:*

$$T(X, \kappa) = \begin{cases} (Tr(X), Tr(X)), & \text{se } \kappa = 0 \\ (X, Tr(X)), & \text{se } \kappa = 1. \end{cases}$$

Se $(E_i; D_i)$ e $(e_i; d_i)$ são os blocos de entrada e saída, respectivamente, da $(i+1)$ -ésima iteração, $0 \leq i \leq (N-1)$, então a $(i+1)$ -ésima iteração do FEAL-N(X)/RP é definida por:

$$\begin{aligned} (Y_1, Y_2) &= T(D_i, \kappa); \\ d_i &= Y_2 \text{ e} \\ e_i &= E_i \oplus f(Y_1, K_i), \end{aligned}$$

onde

$$\begin{aligned} K_i &\text{ é a subchave a ser utilizada na iteração } (i+1) \text{ e} \\ \kappa &= \begin{cases} 0, & \text{no ciframento} \\ 1, & \text{no deciframento.} \end{cases} \end{aligned}$$

Se $i < N-1$ então o bloco de entrada da $(i+2)$ -ésima iteração é $(E_{i+1}; D_{i+1}) = (d_i; e_i)$. A figura 5.1 mostra a estrutura do FEAL-N(X)/RP.

A seguir, mostraremos que o FEAL com essa alteração é inversível e conserva o Efeito Avalanche.

Definição 5.4 *Seja I um conjunto qualquer e g uma função em I . Dizemos que g é uma involução se para todo $i \in I$, $g(g(i))=i$.*

Figura 5.1 Estrutura do FEAL- $N(X)/RP$

Lema 5.1 *Cada iteração do FEAL-N(X)/RP é inversível.*

Dem.: Sejam $(E_i; D_i)$ e $(e_i; d_i)$, os blocos de entrada e saída, respectivamente, da $(i + 1)$ -ésima iteração do FEAL-N(X)/RP, $0 \leq i \leq (N - 1)$. Por definição,

$$d_i = Y_2 \quad e \quad e_i = E_i \oplus f(Y_1, K_i)$$

onde $(Y_1, Y_2) = T(D_i, \kappa)$.

Caso 1: $\kappa = 0$ (ciframento).

Temos $Y_1 = Tr(D_i)$ e $Y_2 = Tr(D_i)$.

Então:

$$d_i = Tr(D_i) \quad e \quad e_i = E_i \oplus f(d_i, K_i).$$

Pelas propriedades da operação ou-exclusivo

$$E_i = e_i \oplus f(d_i, K_i)$$

e como a função Tr é uma involução, segue que

$$Tr(d_i) = Tr(Tr(D_i)) = D_i.$$

Caso 2: $\kappa = 1$ (deciframento).

Temos $Y_1 = D_i$ e $Y_2 = Tr(D_i)$.

Então:

$$d_i = Tr(D_i) \quad e \quad e_i = E_i \oplus f(D_i, K_i).$$

Como a função Tr é uma involução, segue que

$$Tr(d_i) = Tr(Tr(D_i)) = D_i$$

e pelas propriedades da operação ou-exclusivo

$$E_i = e_i \oplus f(D_i, K_i).$$

Logo, $(E_i; D_i)$ pode ser obtido a partir de $(e_i; d_i)$ qualquer que seja o valor de κ , ou seja, cada iteração do FEAL-N(X)/RP é inversível. $\square$

Notemos que as operações utilizadas para obter $(E_i; D_i)$ a partir de $(e_i; d_i)$ na demonstração do lema 5.1 são as mesmas operações que são executadas em uma iteração do FEAL-N(X)/RP quando o valor de κ é complementado.

Lema 5.2 *O FEAL-N(X)/RP é inversível.*

Dem.: O FEAL-N(X)/RP só difere do FEAL-N(X) dentro de cada iteração. Como o FEAL-N(X) é inversível e, pelo lema 5.1, cada iteração do FEAL-N(X)/RP é inversível, segue que o FEAL-N(X)/RP é inversível. $\square$

Lema 5.3 *O FEAL-N(X)/RP conserva o Efeito Avalanche.*

Dem.: Uma função possui o Efeito Avalanche se, em média, metade dos seus bits de saída se alteram quando um único bit em qualquer posição da entrada é alterado. Como a única diferença entre o FEAL-N(X)/RP e o FEAL-N(X) é a existência da função Tr em cada iteração e a função Tr apenas altera a posição dos bits e não o seu valor, segue que o FEAL-N(X)/RP possui o Efeito Avalanche se, e somente se, o FEAL-N(X) também o possui. Dado que o FEAL-N(X) possui Efeito Avalanche para $N \geq 4$, segue o lema. $\square$

A seguir, são dados os algoritmos de ciframento e deciframento do FEAL-N(X)/RP. Os valores $K_i, 0 \leq i \leq N + 7$, são as subchaves geradas a partir de uma chave K fixada. Os algoritmos de geração de subchaves são iguais aos das cifras originais (FEAL-N(X)) e ϕ denota uma sequência nula de 32 bits.

Algoritmo de Ciframento do FEAL-N(X)/RP

“Cifra um texto legível L de 64 bits pela chave K .”

início

$(E_0; D_0) := L;$

$(E_0; D_0) := (E_0; D_0) \oplus (K_N; K_{N+1}; K_{N+2}; K_{N+3});$

$(E_0; D_0) := (E_0; D_0) \oplus (\phi; E_0);$

para $r := 1$ até N faça

$(Y_1, Y_2) := T(D_{r-1}, 0);$

$E_r := Y_2;$

$D_r := E_{r-1} \oplus f(Y_1, K_{r-1});$

fim-para

$(E_N; D_N) := (D_N; E_N);$

$(E_N; D_N) := (E_N; D_N) \oplus (\phi; E_N);$

$(E_N; D_N) := (E_N; D_N) \oplus (K_{N+4}; K_{N+5}; K_{N+6}; K_{N+7});$

$C := (E_N; D_N);$

“ C é o texto cifrado.”

fim.

Algoritmo de Deciframento do FEAL-N(X)/RP

“Decifra um texto C de 64 bits pela chave K .”

início

$(E_N; D_N) := C;$
 $(E_N; D_N) := (E_N; D_N) \oplus (K_{N+4}; K_{N+5}; K_{N+6}; K_{N+7});$
 $(E_N; D_N) := (E_N; D_N) \oplus (\phi; E_N);$

para $r := N$ até 1 faça

$(Y_1, Y_2) := T(D_r, 1);$
 $E_{r-1} := Y_2;$
 $D_{r-1} := E_r \oplus f(Y_1, K_{r-1});$

fim-para

$(E_0; D_0) := (D_0; E_0);$
 $(E_0; D_0) := (E_0; D_0) \oplus (\phi; E_0);$
 $(E_0; D_0) := (E_0; D_0) \oplus (K_N; K_{N+1}; K_{N+2}; K_{N+3});$
 $L := (E_0; D_0);$

“ L é texto decifrado.”

fim.

Como podemos observar, a única diferença entre os algoritmos de ciframento e deciframento do FEAL-N(X)/RP são a ordem de utilização das subchaves e o valor do parâmetro κ na função T ($\kappa = 0$, no ciframento, e $\kappa = 1$, no deciframento). Assim, podemos ainda manter a simetria da cifra após a introdução da função Tr .

Uma das principais características do FEAL é o fato dele ser eficientemente implementável em software. A função Tr apesar de ser definida em relação a bits, pode ser facilmente implementada sobre bytes da seguinte forma:

- Constrói-se inicialmente uma tabela de paridade, *tabpar*, de 256 posições, tal que:

$$tabpar[i] = \begin{cases} 0, & \text{se o número de bits iguais a 1 em } i \text{ é par} \\ 1, & \text{c.c.} \end{cases},$$

$$0 \leq i \leq 255.$$

A construção de tal tabela é bastante simple e rápida. O trecho abaixo, em linguagem C, resolve este problema:

```
/* Declaração das variáveis */  
  
int quoc, resto, i;  
unsigned char tabpar[256];  
  
/* Construção da tabela de paridade */  
  
tabpar[0]=0;
```

```

for(i=1;i<256;i++)
{
    quoc=(int)i/2;
    resto=i - 2*quoc;
    tabpar[i]=(resto!=tabpar[quoc])?1:0;
};

/* Fim. */

```

Com a utilização dessa tabela, a função $Tr(X)$ pode ser implementada como:

```

se ((tabpar[X0]+tabpar[X1]+tabpar[X2]+tabpar[X3]) mod 2 = 1) então
    X := (XD; XE);
fim-se

```

Portanto, o FEAL-N(X)/RP é simétrico e eficientemente implementável em software. Simulações em computador mostraram que o tempo gasto pelo FEAL-N(X)/RP é aproximadamente 10% maior que o tempo gasto pelo FEAL-N(X). Na próxima seção, estabeleceremos algumas propriedades da função Tr que nos serão úteis para analisar a resistência do FEAL-N(X)/RP contra a Criptoanálise Diferencial.

5.2 Propriedades da função Tr

A partir desta seção, supomos que, dada uma seqüência de bits X :

- $Imp(X)$ é o evento “número de bits iguais a 1 em X é ímpar”,
- $Par(X)$ é o evento “número de bits iguais a 1 em X é par” e
- $Sim(X)$ é o evento “ X é simétrica”.

Lema 5.4 *Se $X = (X_E; X_D)$ é uma seqüência de 32 bits escolhida ao acaso com distribuição uniforme, então:*

$$P[Tr(X) = X] = \frac{1}{2}$$

e

$$P[Tr(X) \neq X] = \frac{1}{2}.$$

Dem.: Vamos mostrar que $Tr(X) = X$ se, e somente se, vale $Par(X)$.

($\Leftarrow$) Se vale $Par(X)$ então, pela definição da função Tr , $Tr(X) = X$.

($\Rightarrow$) Suponhamos que exista um X tal que valha $Tr(X) = X$ e $Imp(X)$. Como vale $Imp(X)$, segue que, $Tr(X) = (X_D; X_E)$. Dado que $Tr(X) = X$, temos $X_E = X_D$, ou

seja, X é uma sequência de bits simétrica. Mas, se X é simétrica, então vale $Par(X)$ (absurdo!).

Logo, $Tr(X) = X$ se, e somente se, vale $Par(X)$.

Como X é escolhida aleatoriamente de uma distribuição uniforme, $P[Par(X)] = \frac{1}{2}$.

Logo:

$$P[Tr(X) = X] = \frac{1}{2}.$$

Conseqüentemente:

$$\begin{aligned} P[Tr(X) \neq X] &= 1 - P[Tr(X) = X]. \\ &= \frac{1}{2} \end{aligned}$$

□

Para sabermos qual a influência dos ataques diferenciais sobre o FEAL-N(X)/RP, devemos estudar o comportamento da diferença num par após a aplicação de Tr . A Criptoanálise Diferencial trabalha com pares de textos legíveis (X_1, X_2) que possuem um certo valor da diferença fixado, onde um desses elementos é escolhido aleatoriamente. Nesse sentido, estaremos considerando seqüências de 32 bits X, X_1, X_2 , onde:

- X é um valor fixado,
- X_1 é escolhida aleatoriamente com distribuição uniforme, e
- $X_2 = X \oplus X_1$.

Lema 5.5 *Sejam X, X_1, X_2 , seqüências de 32 bits, tais que $X = X_1 \oplus X_2$. Então:*

$$P[Tr(X_1) \oplus Tr(X_2) = X] = \begin{cases} P[Tr(X_1) = X_1, Tr(X_2) = X_2] + \\ P[Tr(X_1) \neq X_1, Tr(X_2) \neq X_2], & \text{se } Sim(X) \\ P[Tr(X_1) = X_1, Tr(X_2) = X_2], & c.c. \end{cases}$$

Dem.: Vamos analisar os quatro casos possíveis que podem ocorrer com $Tr(X_1)$ e $Tr(X_2)$, e verificar sob quais condições a diferença é X .

Caso 1 ($Tr(X_1) = X_1, Tr(X_2) = X_2$): Claramente, neste caso, a diferença X continua válida.

Caso 2 ($Tr(X_1) \neq X_1, Tr(X_2) = X_2$): Suponhamos que a diferença X seja válida. Então temos:

$$((X_1)_E; (X_1)_D) \oplus ((X_2)_E; (X_2)_D) = (X_E; X_D)$$

e

$$((X_1)_D; (X_1)_E) \oplus ((X_2)_E; (X_2)_D) = (X_E; X_D)$$

donde

$$(X_1)_D \oplus (X_2)_E = X_E \quad e \quad (X_1)_E \oplus (X_2)_E = X_E.$$

Logo, $(X_1)_D = (X_1)_E$, ou seja, X_1 é simétrica. Mas, se X_1 é simétrica, então $Tr(X_1) = X_1$ (absurdo!).

Portanto, neste caso, a diferença X não é válida.

Caso 3 ($Tr(X_1) = X_1, Tr(X_2) \neq X_2$): Analogamente ao caso 2, a diferença X não é válida.

Caso 4 ($Tr(X_1) \neq X_1, Tr(X_2) \neq X_2$): Claramente, a diferença X só permanece válida se X for uma seqüência de bits simétrica.

Assim, como os quatro casos são disjuntos dois a dois, segue o resultado. $\square$

Os próximos três lemas sobre paridade de seqüências de bits, nos serão úteis para o cálculo de $P[Tr(X_1) \oplus Tr(X_2) = Y]$, onde $Y \in \{0, 1\}^{32}$.

Lema 5.6 *Sejam X, X_1, X_2 , seqüências de n bits, tais que $X = X_1 \oplus X_2$. Se vale $Par(X_1) \wedge Par(X_2)$ então vale $Par(X)$.*

Dem.: Sejam I e J , os conjuntos de índices definidos por:

$$I := \{i \in \{0, 1, \dots, n-1\} : X_1^i = X_2^i\}$$

e

$$J := \{0, 1, \dots, n-1\} \setminus I.$$

Se Y é uma seqüência de bits então denotaremos por $Y[I]$ a seqüência formada pelos bits $Y^i, i \in I$. Claramente, todos os bits de $X[I]$ são nulos.

Se vale $Par(X_1[I])$ então vale $Par(X_1[J])$, pois $Par(X_1)$ é válido. Como $X_1[I] = X_2[I]$ e vale $Par(X_2)$, segue que $Par(X_2[J])$ também é válido. Logo, o número de bits iguais a 1 em $X[J]$ é par.

Se vale $Imp(X_1[I])$ então vale $Imp(X_1[J])$, pois $Par(X_1)$ é válido. Como $X_1[I] = X_2[I]$ e vale $Par(X_2)$, segue que $Imp(X_2[J])$ também é válido. Logo, o número de bits iguais a 1 em $X[J]$ é par.

Portanto, como vale $Par(X[I]) \wedge Par(X[J])$, vale $Par(X)$. $\square$

Lema 5.7 *Sejam X, X_1, X_2 , seqüências de n bits, tais que $X = X_1 \oplus X_2$. Se vale $Imp(X_1) \wedge Par(X_2)$ então vale $Imp(X)$.*

Dem.: Análoga ao lema 5.6. $\square$

Lema 5.8 *Sejam X, X_1, X_2 , seqüências de n bits, tais que $X = X_1 \oplus X_2$. Se vale $Imp(X_1) \wedge Imp(X_2)$ então vale $Par(X)$.*

Dem.: Análoga ao lema 5.6. □

Voltemos ao problema de calcular $P[Tr(X_1) \oplus Tr(X_2) = X]$.

Lema 5.9 *Sejam X, X_1, X_2 , seqüências de 32 bits, tais que X_1 é escolhida aleatoriamente de uma distribuição uniforme, X é um valor fixado e $X = X_1 \oplus X_2$. Então:*

$$P[Tr(X_1) = X_1, Tr(X_2) = X_2] = \begin{cases} \frac{1}{2}, & \text{se } Par(X) \\ 0, & \text{c.c.} \end{cases}$$

Dem.: Observemos que o evento $\{(Tr(X_1) = X_1) \wedge (Tr(X_2) = X_2)\}$ ocorre se, e somente se, vale $Par(X_1) \wedge Par(X_2)$.

Pelo lema 5.6, isso ocorre somente se vale $Par(X)$.

Assim, utilizando os lemas 5.6, 5.7 e 5.8, temos:

$$\begin{aligned} P[Tr(X_1) = X_1, Tr(X_2) = X_2] &= P[Par(X_1) \wedge Par(X_2)] \\ &= P[Par(X_2) | Par(X_1)] \cdot P[Par(X_1)] \\ &= \begin{cases} P[Par(X_1)], & \text{se } Par(X) \\ 0, & \text{c.c.} \end{cases} \\ &= \begin{cases} \frac{1}{2}, & \text{se } Par(X) \\ 0, & \text{c.c.} \end{cases} \end{aligned} \quad \square$$

Lema 5.10 *Sejam X, X_1, X_2 , seqüências de 32 bits, tais que X_1 é escolhida aleatoriamente de uma distribuição uniforme, X é um valor fixado e $X = X_1 \oplus X_2$. Então:*

$$P[Tr(X_1) \neq X_1, Tr(X_2) \neq X_2] = \begin{cases} \frac{1}{2}, & \text{se } Par(X) \\ 0, & \text{c.c.} \end{cases}$$

Dem.: Como

$$P[(Tr(X_1) \neq X_1) \cup (Tr(X_2) \neq X_2)] = 1 - P[Tr(X_1) = X_1, Tr(X_2) = X_2],$$

segue do lema 5.9 que:

$$P[(Tr(X_1) \neq X_1) \cup (Tr(X_2) \neq X_2)] = \begin{cases} \frac{1}{2}, & \text{se } Par(X) \\ 1, & \text{c.c.} \end{cases}$$

Agora,

$$P[(Tr(X_1) \neq X_1) \cup (Tr(X_2) \neq X_2)] = P[Tr(X_1) \neq X_1] + P[Tr(X_2) \neq X_2] - P[Tr(X_1) \neq X_1, Tr(X_2) \neq X_2],$$

donde

$$P[Tr(X_1) \neq X_1, Tr(X_2) \neq X_2] = \begin{cases} \frac{1}{2}, & \text{se } Par(X) \\ 0, & \text{c.c.} \end{cases} \quad \square$$

Lema 5.11 *Sejam X, X_1, X_2 , seqüências de 32 bits, tais que X_1 é escolhida aleatoriamente de uma distribuição uniforme, X é um valor fixado e $X = X_1 \oplus X_2$. Então:*

$$P[Tr(X_1) \oplus Tr(X_2) = X] = \begin{cases} 1, & \text{se } Par(X) \wedge Sim(X) \\ \frac{1}{2}, & \text{se } Par(X) \wedge \overline{Sim(X)} \\ 0, & \text{c.c.} \end{cases}$$

Dem.: Decorre dos lemas 5.5, 5.9 e 5.10. $\square$

Definição 5.5 *Sejam X e Y seqüências de 32 bits. Dizemos que X causa Y pela função Tr com probabilidade p se para uma fração p de todos os pares distintos cuja diferença é X , o valor da diferença no par após a aplicação de Tr é Y .*

Com esses resultados, podemos concluir que:

- (i) $X \rightarrow X$ pela função Tr se, e somente se, vale $Par(X)$;
- (ii) $X \rightarrow (X_D; X_E)$ pela função Tr se, e somente se, vale $Par(X)$ e
- (iii) $P[X \rightarrow X \text{ por } Tr] = P[X \rightarrow (X_D; X_E) \text{ por } Tr]$.

Resta saber o que acontece com X quando vale $Imp(X)$. O lema a seguir trata dessa situação.

Lema 5.12 *Seja X uma seqüência de 32 bits. Se vale $Imp(X)$ então, para qualquer Y em $\{0, 1\}^{32}$ a probabilidade de X causar Y pela função Tr é menor ou igual a 2^{-15} .*

Dem.: Vamos mostrar que o número de pares distintos que satisfazem X e após a aplicação de Tr satisfazem Y é menor ou igual a 2^{16} .

Sejam (X_1, X_2) e (X_1^*, X_2^*) , dois pares distintos cuja diferença no par antes da aplicação da função Tr é X e após a aplicação da função Tr a diferença passa a ser Y . Como vale $Imp(X)$, segue que um dos elementos de cada par possui um número ímpar de bits iguais

a 1 e o outro possui um número par. Sem perda de generalidade, suponhamos que valha $Imp(X_1) \wedge Imp(X_1^*)$. Então temos:

$$((X_1)_E; (X_1)_D) \oplus X_2 = ((X_1^*)_E; (X_1^*)_D) \oplus X_2^*$$

e

$$((X_1)_D; (X_1)_E) \oplus X_2 = ((X_1^*)_D; (X_1^*)_E) \oplus X_2^*$$

donde

$$(X_1)_E \oplus (X_1^*)_E = (X_1)_D \oplus (X_1^*)_D$$

Assim, $X_1 \oplus X_1^*$ é simétrica. Dado que o número de seqüências de 32 bits distintas que são simétricas é 2^{16} , segue que existem no máximo 2^{16} pares distintos nessas condições. Como o número de pares distintos que satisfazem X é 2^{31} , segue o resultado. $\square$

Através desse lema, podemos perceber que a utilização de valores da diferença cujo número de bits iguais a 1 é ímpar não é de muita valia em um ataque ao FEAL-N(X)/RP. Na próxima seção, consideraremos o efeito das características do FEAL sobre o FEAL/RP.

5.3 A utilização de características do FEAL-N(X) para o FEAL-N(X)/RP

Vamos considerar nesta seção que se $\Omega = (\Omega_L, \Omega_\Lambda, \Omega_C)$ é uma característica de n iterações com probabilidade p^Ω para o FEAL-N(X) então Ω possui probabilidade q^Ω em relação ao FEAL-N(X)/RP. q_i^Ω denotará a probabilidade de $\lambda_{E_n}^i$ causar λ_S^i na i -ésima iteração da característica e r_i^Ω denotará a probabilidade de $\lambda_{E_n}^i$ causar $\lambda_{E_n}^i$ pela função Tr . A notação para as características e suas probabilidades é a mesma utilizada no capítulo 3.

Lema 5.13 $q_i^\Omega = p_i^\Omega \cdot r_i^\Omega, 1 \leq i \leq n$.

Dem.: Pela estrutura das características, temos:

$$\begin{aligned} q_i^\Omega &= P[\lambda_{E_n}^i \rightarrow \lambda_S^i \text{ por } f \circ Tr, \lambda_{E_n}^i \rightarrow \lambda_{E_n}^i \text{ por } Tr] \\ &= P[\lambda_{E_n}^i \rightarrow \lambda_S^i \text{ por } f, \lambda_{E_n}^i \rightarrow \lambda_{E_n}^i \text{ por } Tr] \\ &= P[\lambda_{E_n}^i \rightarrow \lambda_S^i \text{ por } f] \cdot P[\lambda_{E_n}^i \rightarrow \lambda_{E_n}^i \text{ por } Tr] \\ &= p_i^\Omega \cdot r_i^\Omega \end{aligned}$$

$\square$

Como, pelo lema 5.13, já temos um meio de determinar q_i^Ω a partir de p_i^Ω , vamos analisar o comportamento das características apresentadas para o FEAL-N(X) no capítulo 4 em relação ao FEAL-N(X)/RP. Estaremos utilizando os resultados do lema 5.11.

Definição 5.6 Dizemos que uma característica Ω é válida se não existe $i, 1 \leq i \leq n$, tal que q_i^Ω seja igual a zero.

Característica 1-1: Permanece válida e com mesma probabilidade, pois:

$$Par(00000000_x) \wedge Sim(00000000_x)$$

é válido.

Característica 1-2: Permanece válida e com mesma probabilidade, pois:

$$Par(80808080_x) \wedge Sim(80808080_x)$$

é válido.

Característica 1-3: Permanece válida, mas a sua probabilidade passa a ser $\frac{1}{2}$, pois vale:

$$Par(80800000_x) \wedge \overline{Sim(80800000_x)}.$$

Característica 1-4: Permanece válida, mas a sua probabilidade passa a ser $\frac{1}{2}$, pois vale:

$$Par(00008080_x) \wedge \overline{Sim(00008080_x)}.$$

Decorre dos decréscimos de probabilidades ocorridos para essas características que das dezesseis características de duas iterações com probabilidade 1 existentes para o FEAL-N(X), todas continuam válidas para o FEAL-N(X)/RP, mas apenas quatro delas conservam a probabilidade 1. Para as doze características restantes, oito passam a ter probabilidade $\frac{1}{2}$ e quatro passam a ter probabilidade $\frac{1}{4}$. Das quatro características de três iterações que possuem probabilidade 1 existentes para o FEAL-N(X), duas conservam a probabilidade 1 e duas passam a ter probabilidade $\frac{1}{4}$.

Característica 5-1: Esta característica não é válida para o FEAL-N(X)/RP pois:

$$\lambda_{En}^2 = \lambda_{En}^4 = A0008000_x \Rightarrow r_2^{5-1} = r_4^{5-1} = 0,$$

ou seja, $q^{5-1} = 0$.

Característica 6-1: Também deixa de ser válida, pois suas cinco primeiras iterações formam a característica 5-1, a qual possui probabilidade zero em relação ao FEAL-N(X)/RP.

Característica iterativa: Deixa de ser válida, pois sua probabilidade passa a ser 2^{-3} para $n = 1$ e zero para $n > 1$ em relação ao FEAL-N(X)/RP dado que:

$$\lambda_{En}^1 = 80608000_x \Rightarrow r_1^{iter} = \frac{1}{2} \Rightarrow q_1^{iter} = 2^{-3} \text{ e}$$

$$\lambda_{En}^2 = 80E08000_x \Rightarrow r_2^{iter} = 0 \Rightarrow q_2^{iter} = 0 \Rightarrow q^{iter} = 0.$$

A tabela 5.1 mostra os valores de p e q para cada uma das características citadas acima. Os próximos lemas nos fornecem resultados mais genéricos sobre a utilização de características do FEAL-N(X) para atacar o FEAL-N(X)/RP.

característica	p^Ω	q^Ω
1-1	1	1
1-2	1	1
1-3	1	$\frac{1}{2}$
1-4	1	$\frac{1}{2}$
5-1	$\frac{1}{16}$	0
6-1	$\frac{1}{128}$	0
iterativa	2^{-2n}	0, p/ $n > 1$

Tabela 5.1 Probabilidade de algumas características do FEAL-N(X) em relação ao FEAL-N(X)/RP.

Lema 5.14 $q^\Omega \leq p^\Omega$.

Dem.:

$$\begin{aligned} q^\Omega &= \prod_{i=1}^n q_i^\Omega \\ &= \prod_{i=1}^n p_i^\Omega \cdot r_i^\Omega, \quad \text{pelo lema 5.13} \\ &\leq \prod_{i=1}^n p_i^\Omega, \quad \text{pois } 0 \leq r_i^\Omega \leq 1, \quad 1 \leq i \leq n \\ &= p^\Omega. \end{aligned}$$

□

Lema 5.15 $q^\Omega = p^\Omega$ se, e somente se, os valores de λ_{En}^i , $1 \leq i \leq n$, são tais que valha:

$$Par(\lambda_{En}^i) \wedge Sim(\lambda_{En}^i). \quad (5.1)$$

Dem.: Decorre do lema 5.13 que $q^\Omega = p^\Omega$ se, e somente se, $r_i^\Omega = 1$ para todo $i, 1 \leq i \leq n$. Como r_i^Ω é a probabilidade de $\lambda_{En}^i \rightarrow \lambda_{En}^i$ pela função Tr , segue do lema 5.11 que $r_i^\Omega = 1$ se, e somente se, vale 5.1. $\square$

Lema 5.16 *Uma característica iterativa Ω válida para o FEAL-N(X) é também válida com mesma probabilidade para o FEAL-N(X)/RP se, e somente se, para todo $i, 1 \leq i \leq n$, valem:*

- (i) $Sim(\lambda_{En}^i)$ e
- (ii) $Sim(\lambda_S^i)$.

Dem.: A condição (i) é o lema 5.15 e a condição (ii) é uma decorrência de (i) e do fato de Ω ser uma característica iterativa. $\square$

Lema 5.17 *Uma característica Ω do FEAL-N(X) é válida para o FEAL-N(X)/RP se, e somente se, não existe $i, 1 \leq i \leq n$, tal que:*

$$Imp(\lambda_{En}^i) \quad (5.2)$$

seja válido.

Dem.: Pelo lema 5.13, $q_i^\Omega = p_i^\Omega \cdot r_i^\Omega$. Suponhamos que exista um $i, 1 \leq i \leq n$, tal que (5.2) seja válido. Pelo lema 5.11, segue que nessas condições o valor de r_i^Ω é zero, ou seja, $q_i^\Omega = 0$.

Por outro lado, se $q_i^\Omega = 0$ então existe $i, 1 \leq i \leq n$, tal que $r_i^\Omega = 0$, ou seja, $\lambda_{En}^i \rightsquigarrow \lambda_{En}^i$ pela função Tr . Logo, pelo lema 5.11 vale (5.2). $\square$

Definição 5.7 *Dizemos que uma característica Ω de n iterações é simétrica se para todo $i, 1 \leq i \leq n$, λ_{En}^i e λ_S^i são simétricas.*

Definição 5.8 *Dizemos que uma característica Ω de n iterações possui probabilidade média p_m^Ω se $\prod_{i=1}^n p_m^\Omega = p^\Omega$.*

Pelo lema 5.16, as características iterativas simétricas são as únicas características iterativas do FEAL-N(X) que não sofrem nenhuma redução de probabilidade quando aplicadas ao FEAL-N(X)/RP. Porém, na próxima seção, mostraremos que, na prática, não existe característica iterativa simétrica para o FEAL-N(X), fora as triviais, com probabilidade média maior que 2^{-4} .

5.3.1 Sobre características iterativas simétricas

Durante toda essa seção, não estaremos considerando as características triviais, pois elas não possuem nenhuma utilidade para os ataques diferenciais. Assim, sempre que o único tipo de característica existente dentro de determinadas condições for trivial, estaremos dizendo que uma tal característica não existe.

Vamos inicialmente determinar um limitante para $p_f(X, Y)$ quando X e Y são seqüências de bits simétricas e não nulas.

Lema 5.18 *Sejam X e Y , seqüências simétricas de 32 bits. Se X causa Y pela função f e $X \neq 0$ então $p_f(X, Y) \leq 2^{-4}$.*

Dem.: Sejam X e Y , seqüências simétricas de 32 bits. Vamos verificar o que ocorre quando a probabilidade de alguma S-box da função f é igual a 1. Para isso, vamos analisar separadamente cada S-box. Estaremos utilizando arbitrariamente o lema 4.19, o algoritmo Um e a estrutura da função f .

Caso 1: $p_S(X_0, Y_1, Y_0) = 1$.

Caso 1.1: $X_0 = Y_0 = Y_1 = 00_x$.

Como $Y = 0$, segue que $X = 0$.

Caso 1.2: $X_0 = Y_1 = 80_x$ e $Y_0 = 00_x$.

Devemos ter $(80_x; 80_x \oplus X_1) \rightarrow 00_x$ e $(00_x; X_1) \rightarrow 80_x$ pela função S . Agora, $(80_x; 80_x \oplus X_1) \rightarrow 00_x$ se, e somente se, $X_1 = 00_x$ e $(00_x; 00_x) \leadsto 80_x$. Logo, X_0, Y_0, Y_1 não podem assumir esses valores.

Caso 1.3: $X_0 = 80_x, Y_1 = 00_x$ e $Y_0 = 02_x$.

Devemos ter $(00_x; 80_x \oplus X_1) \rightarrow 02_x$ e $(02_x; X_1) \rightarrow 00_x$ pela função S . Agora, $(00_x; 80_x \oplus X_1) \rightarrow 02_x$ se, e somente se, $X_1 = 00_x$ e $(02_x; 00_x) \leadsto 00_x$. Logo, X_0, Y_0, Y_1 não podem assumir esses valores.

Caso 1.4: $X_0 = 00_x, Y_1 = 80_x$ e $Y_0 = 02_x$.

Devemos ter $(80_x; X_1) \rightarrow 02_x$ e $(02_x; X_1) \rightarrow 80_x$ pela função S . Agora, $(80_x; X_1) \rightarrow 02_x$ se, e somente se, $X_1 = 00_x$ e $(02_x; 00_x) \leadsto 80_x$. Logo, X_0, Y_0, Y_1 não podem assumir esses valores.

Caso 2: $p_S(X_0 \oplus X_1, X_0 \oplus X_1, Y_1) = 1$.

Caso 2.1: $X_0 \oplus X_1 = 00_x$ e $Y_1 = 00_x$.

Devemos ter $(00_x; 00_x) \rightarrow Y_0$, o que ocorre se, e somente se, $Y_0 = 00_x$. Logo, $Y = 0$ e, conseqüentemente, $X = 0$.

Caso 2.2: $X_0 \oplus X_1 = 80_x$ e $Y_1 = 00_x$.

Devemos ter $(80_x \oplus X_1; 00_x) \rightarrow Y_0$, $(00_x; 80_x) \rightarrow Y_0$ e $(Y_0; X_1) \rightarrow 00_x$ pela função S . Agora, $(00_x; 80_x) \rightarrow Y_0$ se, e somente se, $Y_0 = 02_x$ e $(80_x \oplus X_1; 00_x) \rightarrow 02_x$ se, e somente se, $X_1 = 00_x$. Mas, $(02_x; 00_x) \rightsquigarrow 00_x$. Logo, $X_0 \oplus X_1$ e Y_1 não podem assumir esses valores.

Caso 3: $p_S(Y_1, X_0 \oplus X_1, Y_0) = 1$.

Caso 3.1: $Y_1 = Y_0 = X_0 \oplus X_1 = 00_x$.

Como $Y = 0$, segue que $X = 0$.

Caso 3.2: $Y_1 = X_0 \oplus X_1 = 80_x$ e $Y_0 = 00_x$.

$X_0 \oplus X_1, Y_0$ e Y_1 não podem assumir esses valores, pois $(80_x; 80_x) \rightsquigarrow 80_x$ pela função S .

Caso 3.3: $Y_1 = 80_x, X_0 \oplus X_1 = 00_x$ e $Y_0 = 02_x$.

$X_0 \oplus X_1, Y_0$ e Y_1 não podem assumir esses valores, pois $(00_x; 00_x) \rightsquigarrow 80_x$ pela função S .

Caso 3.4: $X_0 \oplus X_1 = 80_x, Y_1 = 00_x$ e $Y_0 = 02_x$.

Devemos ter $(80_x \oplus X_1; 00_x) \rightarrow 02_x$ e $(X_1; 02_x) \rightarrow 00_x$ pela função S . Agora, $(80_x \oplus X_1; 00_x) \rightarrow 02_x$ se, e somente se, $X_1 = 00_x$ e $(00_x; 02_x) \rightsquigarrow 00_x$. Logo, $X_0 \oplus X_1, Y_0$ e Y_1 não podem assumir esses valores.

Caso 4: $p_S(Y_0, X_1, Y_1) = 1$.

Caso 4.1: $Y_1 = Y_0 = X_1 = 00_x$.

Como $Y = 0$, segue que $X = 0$.

Caso 4.2: $X_1 = Y_1 = 80_x$ e $Y_0 = 00_x$.

Devemos ter $(80_x \oplus X_0; 80_x) \rightarrow 00_x$ e $(X_0; 80_x) \rightarrow 00_x$ pela função S . Agora, $(80_x \oplus X_0; 80_x) \rightarrow 00_x$ se, e somente se, $X_0 = 00_x$ e $(00_x; 80_x) \rightsquigarrow 00_x$. Logo, X_1, Y_0 e Y_1 não podem assumir esses valores.

Caso 4.3: $X_1 = 00_x, Y_0 = 80_x$ e $Y_1 = 02_x$.

Devemos ter $(X_0; 02_x) \rightarrow 80_x$ e $(X_0; X_0) \rightarrow 02_x$. Agora, $(X_0; X_0) \rightarrow 02_x$ somente se $X_0^6 = 1$ e $(X_0; 02_x) \rightarrow 80_x$ somente se $X_0^1 = 1$, donde $p_S(X_0; 02_x; 80_x) \leq \frac{1}{4}$ e $p_S(X_0; X_0; 02_x) \leq \frac{1}{4}$. Logo, pelo lema 4.24, $p_f(X, Y) \leq 2^{-6}$.

Caso 4.4: $X_1 = 80_x, Y_0 = 00_x$ e $Y_1 = 02_x$.

Devemos ter $(80_x \oplus X_0; 80_x \oplus X_0) \rightarrow 02_x$ e $(80_x \oplus X_0; 02_x) \rightarrow 00_x$. Agora, $(80_x \oplus X_0; 80_x \oplus X_0) \rightarrow 02_x$ somente se $X_0^6 = 1$ e $(80_x \oplus X_0; 02_x) \rightarrow 00_x$ somente se $X_0^1 = 1$. Analogamente ao caso 4.3, $p_f(X, Y) \leq 2^{-6}$.

Assim, quando uma das S-boxes da função f possui probabilidade 1, $X \neq 0$ e $Y \neq 0$, temos $p_f(X, Y) \leq 2^{-6}$. Pelo lema 4.17, a probabilidade de qualquer S-box da função f é zero ou uma potência de $\frac{1}{2}$. Pelo lema 4.24, $p_f(X, Y)$ é o produto das probabilidades de cada S-box. Portanto, o lema está demonstrado. $\square$

Definição 5.9 *Seja Ω uma característica. Dizemos que a i -ésima iteração de Ω é uma iteração nula se $\lambda_{E_n}^i$ é nula.*

De acordo com o lema 5.18, se Ω é uma característica iterativa simétrica então as únicas iterações de Ω que possuem probabilidade maior que $\frac{1}{16}$ são as iterações nulas. Desta forma, vamos determinar o número máximo de iterações nulas que uma característica iterativa simétrica de n iterações Ω pode possuir e, assim, chegarmos a um limitante para a probabilidade média de Ω .

Definição 5.10 *Sejam X, Y, Z e C , bytes. Dizemos que C é a diferença de carry de $((X; Y), Z)$ se $((Rot2)^{-1}(Z))^i = X^i \oplus Y^i \oplus C^i$, para todo i em $\{0, 1, 2, \dots, 7\}$, $C^0 = 0$ e $C^{i+1} = C^i$ quando $C^i = X^i = Y^i$.*

Definição 5.11 *Sejam X e Y , seqüências de 32 bits, tais que $p_f(X, Y) > 0$. Então:*

- (i) C_0 é a diferença de carry de $((X_0; Y_1), Y_0)$,
- (ii) C_1 é a diferença de carry de $((X_0 \oplus X_1; X_2 \oplus X_3), Y_1)$,
- (iii) C_2 é a diferença de carry de $((Y_1; X_2 \oplus X_3), Y_2)$ e
- (iv) C_3 é a diferença de carry de $((Y_2; X_3), Y_3)$.

Lema 5.19 *Se X e Y são seqüências simétricas de 32 bits, X causa Y pela função f e existe j em $\{0, 1, 2, \dots, 7\}$ tal que $C_0^j = C_1^j = C_2^j = C_3^j = 0$ então valem:*

- (i) $X_1^j = Y_0^j = 0$ e
- (ii) $Y_0^{(j+2) \bmod 8} = Y_1^j \oplus X_0^j$.

Dem.: Se X e Y são seqüências simétricas de 32 bits e X causa Y pela função f então:

- $(X_0; Y_1) \rightarrow Y_0$ pela função S ,
 - $(X_0 \oplus X_1; X_0 \oplus X_1) \rightarrow Y_1$ pela função S ,
 - $(Y_1; X_0 \oplus X_1) \rightarrow Y_0$ pela função S e
 - $(Y_0; X_1) \rightarrow Y_1$ pela função S .
- Pelo algoritmo Zero, para $i \in \{0, 1, \dots, 7\}$, valem:

$$Y_0^{(i+2) \bmod 8} = X_0^i \oplus Y_1^i \oplus C_0^i$$

$$Y_1^{(i+2) \bmod 8} = C_1^i$$

$$Y_0^{(i+2) \bmod 8} = Y_1^i \oplus X_0^i \oplus X_1^i \oplus C_2^i$$

$$Y_1^{(i+2) \bmod 8} = Y_0^i \oplus X_1^i \oplus C_3^i$$

Como $C_0^j = C_1^j = C_2^j = C_3^j = 0$, temos:

$$X_0^j \oplus Y_1^j = Y_1^j \oplus X_0^j \oplus X_1^j$$

$$Y_0^j \oplus X_1^j = 0$$

donde, segue o resultado. □

Definição 5.12 Uma característica iterativa Ω possui tamanho n se n é o menor número inteiro tal que:

- (i) $\lambda_{En}^n = \lambda_S^1 \oplus \lambda_{En}^2$ e
- (ii) $\lambda_{En}^1 = \lambda_{En}^{n-1} \oplus \lambda_S^n$.

Definição 5.13 $C_i(j)$ é o valor da diferença de carry C_i na j -ésima iteração de uma característica Ω , $i \in \{0, 1, 2, 3\}$.

Lema 5.20 Se existe uma característica iterativa simétrica Ω de tamanho n , $n \geq 4$ então $p_m^\Omega \leq 2^{-3}$.

Dem.: Vamos mostrar que se a i -ésima iteração de Ω é nula, $i \leq n - 3$, então as iterações $(i + 1)$, $(i + 2)$ e $(i + 3)$ não podem ser nulas.

Caso 1: As iterações i e $(i + 1)$ são nulas.

Claramente, todas as iterações da característica são nulas.

Caso 2: As iterações i e $(i + 2)$ são nulas.

Pela definição de característica, temos $\lambda_S^{i+1} = \lambda_{En}^i \oplus \lambda_{En}^{i+2}$. Como $\lambda_{En}^i = \lambda_{En}^{i+2} = 0$, segue que $\lambda_S^{i+1} = 0$. Dado que no FEAL $p_f(X, 0) \neq 0$ se, e somente se, $X = 0$, temos que λ_{En}^{i+1} também é zero, ou seja, as iterações i e $(i + 1)$ da característica são nulas. Portanto, recaímos no caso 1, e, conseqüentemente, todas as iterações de Ω são nulas.

Caso 3: As iterações i e $(i + 3)$ são nulas.

Suponhamos que exista um k em $\{0, \dots, 7\}$ tal que $C_j(l)^k = 0$ para todo j em $\{0, 1, 2, 3\}$ e l em $\{1, \dots, n\}$. Pelo lema 5.19, $\lambda_{En}^{i+1}, \lambda_S^{i+1}, \lambda_{En}^{i+2}, \lambda_S^{i+2}$ são tais que:

$$(\lambda_{En}^{i+1})_1^k = (\lambda_{En}^{i+2})_1^k = (\lambda_S^{i+1})_0^k = (\lambda_S^{i+2})_0^k = 0.$$

Pela definição de característica, $\lambda_{En}^{i+1} = \lambda_S^{i+2}$ e $\lambda_{En}^{i+2} = \lambda_S^{i+1}$, ou seja:

$$(\lambda_{En}^{i+1})_0^k = (\lambda_{En}^{i+1})_1^k = (\lambda_{En}^{i+2})_0^k = (\lambda_{En}^{i+2})_1^k = 0$$

e

$$(\lambda_S^{i+1})_0^k = (\lambda_S^{i+1})_1^k = (\lambda_S^{i+2})_0^k = (\lambda_S^{i+2})_1^k = 0,$$

donde, pelo algoritmo Zero, $C_j(l)^{k+1} = 0$ para todo j em $\{0, 1, 2, 3\}$ e l em $\{1, \dots, n\}$.

Como $C_j(l)^0 = 0$ para todo j em $\{0, 1, 2, 3\}$ e l em $\{1, \dots, n\}$, segue que Ω é uma característica trivial, ou seja, não existe uma característica iterativa simétrica Ω nessas condições.

Assim, em uma característica iterativa simétrica devem existir pelo menos 3 iterações não nulas para cada iteração nula. Uma iteração nula ocorre com probabilidade 1 e, pelo lema 5.18, uma iteração simétrica não nula possui probabilidade menor ou igual a 2^{-4} . Logo, $p_m^\Omega \leq 2^{-3}$. □

Não sabemos se o caso em que as iterações i e $(i + 4)$ de Ω são nulas pode ou não ocorrer. Caso esta situação também não seja possível então o limitante para p_m^Ω passa a ser $2^{-\frac{10}{3}}$, pois as iterações i e $(i + 5)$ de Ω também não podem ser ambas nulas (a demonstração deste fato é análoga ao caso 3 do lema 5.20). Procuraremos agora melhorar esse resultado para valores particulares de n e também limitar p_m^Ω para os valores de n não cobertos pelo lema 5.20.

Lema 5.21 *Se existe uma característica iterativa simétrica Ω cujo tamanho n , $n \geq 3$, é um número ímpar então Ω não possui nenhuma iteração nula.*

Dem.: Vamos mostrar que se Ω possui uma iteração nula, então Ω é uma característica trivial. Assim, suponhamos que exista um k em $\{0, \dots, 7\}$ tal que $C_j(i)^k = 0$ para todo j em $\{0, 1, 2, 3\}$ e i em $\{1, \dots, n\}$.

Como λ_{En}^i e λ_S^i são simétricas para todo $i \in \{1, \dots, n\}$, pelo lema 5.19, segue que:

$$(\lambda_{En}^i)_1^k = (\lambda_S^i)_0^k = 0.$$

Dado que Ω é uma característica iterativa, então são válidas as seguintes afirmações:

- (i) $\lambda_S^1 = \lambda_{En}^n \oplus \lambda_{En}^2$,
- (ii) $\lambda_S^n = \lambda_{En}^{n-1} \oplus \lambda_{En}^1$ e
- (iii) para todo i em $\{2, \dots, n-1\}$ vale:

$$\lambda_S^i = \lambda_{En}^{i-1} \oplus \lambda_{En}^{i+1}.$$

Logo, valem:

- (i) $(\lambda_S^i)_1^k = 0$ para todo $i \in \{1, \dots, n\}$,
- (ii) $(\lambda_{En}^i)_0^k = (\lambda_{En}^{i+2})_0^k$ para todo $i \in \{1, \dots, n-2\}$,
- (iii) $(\lambda_{En}^{n-1})_0^k = (\lambda_{En}^1)_0^k$ e
- (iv) $(\lambda_{En}^n)_0^k = (\lambda_{En}^2)_0^k$.

Como n é um número ímpar, segue que:

$$(\lambda_{En}^i)_0^k = (\lambda_{En}^j)_0^k$$

para qualquer (i, j) em $\{1, \dots, n\}^2$.

Assim, se Ω possui uma iteração nula então

$$(\lambda_{En}^i)_0^k = 0$$

para todo i em $\{1, \dots, n\}$, donde, pelo algoritmo Zero, $C_j(i)^{k+1} = 0$ para todo j em $\{0, 1, 2, 3\}$ e i em $\{1, \dots, n\}$.

Como $C_j(i)^0 = 0$ para todo j em $\{0, 1, 2, 3\}$ e i em $\{1, \dots, n\}$, segue que Ω é uma característica trivial, ou seja, não existe uma característica iterativa simétrica Ω cujo tamanho é um número ímpar e que possui alguma iteração nula. $\square$

Uma forma alternativa para o enunciado do lema 5.21 é “Se existe uma característica iterativa simétrica Ω cujo tamanho é um número ímpar então $p_m^\Omega \leq 2^{-4}$ ”.

Lema 5.22 *Não existe característica iterativa de tamanho 2.*

Dem.: Seja Ω uma característica iterativa de tamanho 2. Como Ω é uma característica iterativa, valem:

- (i) $\lambda_{En}^1 \oplus \lambda_S^2 = \lambda_{En}^1$ e
- (ii) $\lambda_{En}^2 \oplus \lambda_S^1 = \lambda_{En}^2$.

Logo, $\lambda_S^1 = \lambda_S^2 = 0$. Como no FEAL $p_f(X, 0) \neq 0$ se, e somente se, $X = 0$, segue que λ_{En}^1 e λ_{En}^2 são nulas, ou seja, Ω é uma característica trivial. Portanto, no FEAL não existe característica iterativa de tamanho 2. $\square$

Lema 5.23 *Se Ω é uma característica iterativa de tamanho 4 então Ω não possui nenhuma iteração nula.*

Dem.: Suponhamos que Ω possua uma iteração nula. Sem perda de generalidade, suponhamos que essa iteração seja a primeira. Como Ω é iterativa, valem $\lambda_S^1 = \lambda_S^3$ e $\lambda_S^2 = \lambda_S^4$. Como $\lambda_S^1 = 0$ e, no FEAL, $p_f(X, 0) \neq 0$ se, e somente se, $X = 0$, segue que $\lambda_{En}^3 = 0$. Agora, $\lambda_S^2 = \lambda_{En}^1 \oplus \lambda_{En}^3 = 0$, ou seja, λ_{En}^2 e λ_{En}^4 também são nulas. Logo, Ω é uma característica trivial.

Portanto, se Ω é uma característica iterativa de tamanho 4 então Ω não possui nenhuma iteração nula. □

Lema 5.24 *Se Ω é uma característica iterativa simétrica de tamanho n , $n \leq 5$, então $p_m^\Omega \leq 2^{-4}$.*

Dem.: Claramente, não faz sentido considerarmos característica de tamanho zero e, para o FEAL-N(X), a única característica iterativa de tamanho 1 é a trivial. Pelo lema 5.22, não existe característica iterativa Ω de tamanho 2 e, pelos lemas 5.18, 5.21 e 5.23, $p_m^\Omega \leq 2^{-4}$ se Ω é uma característica iterativa simétrica de tamanho 3, 4 ou 5. □

Como o FEAL-N(X) possui Efeito Avalanche para N maior ou igual a quatro e pelo lema 5.24 não existe característica iterativa simétrica de tamanho menor ou igual a cinco com probabilidade média maior que 2^{-4} , podemos concluir que a chance de existir uma característica iterativa simétrica que possua probabilidade média maior que 2^{-4} é nula.

5.4 Conclusões

Neste capítulo propusemos a introdução da função Tr na família de cifras FEAL como uma forma de fortalecimento contra a Criptoanálise Diferencial. Ademais, mostramos que qualquer característica do FEAL-N(X) nunca é mais eficaz quando aplicada ao FEAL-N(X)/RP do que era para o FEAL-N(X). As únicas características iterativas que conseguem atacar o mesmo número de iterações para as duas versões são as simétricas, as quais, como vimos, possuem a probabilidade média limitada por 2^{-4} , ou seja, em relação a uma característica desse tipo, o FEAL-N(X)/RP é pelo menos tão forte quanto o DES[NBS77].

Mostramos também que uma característica iterativa Ω para a cifra original só é aplicável para a nova versão se as diferenças de entrada e saída possuírem um número par de bits iguais a 1. As únicas iterações que não sofrem nenhuma redução de probabilidade

são as simétricas na entrada. Como uma iteração não nula que é simétrica nas diferenças de entrada e saída possui probabilidade não muito atraente (2^{-4} é apenas um limitante para essa probabilidade. A melhor probabilidade de uma tal iteração que conhecemos é 2^{-9}) e levando-se em conta a dificuldade de utilizar iterações que não são totalmente simétricas, possuem probabilidade alta e cuja diferença de entrada seja simétrica, podemos assumir que em geral a probabilidade de uma característica para o FEAL-N(X) quando aplicada ao FEAL-N(X)/RP sofre um decréscimo de $\frac{1}{2}$ por iteração.

A nova versão permite que se explore um novo tipo de característica, cuja definição é a seguinte:

Definição 5.14 *Seja X^* a seqüência de bits obtida a partir da troca das metades de uma seqüência de bits X . Sejam Ω_L e Ω_C , os valores da diferença nos pares de textos legíveis e textos cifrados, respectivamente. Uma característica de n iterações para o FEAL-N(X)/RP é uma tripla $\Omega = (\Omega_L, \Omega_\Lambda, \Omega_C)$, onde Ω_Λ é uma lista de n elementos $(\Lambda_1, \Lambda_2, \dots, \Lambda_n)$, cada um sendo um par da forma $\Lambda_i = (\lambda_{En}^i, \lambda_S^i)$, onde os λ_{En}^i são seqüências de 32 bits cujo número de bits iguais a 1 é par e os λ_S^i são seqüências de 32 bits. Esta tripla deve ainda satisfazer as seguintes condições:*

- (i) λ_{En}^1 é a metade direita de Ω_L ou essa metade com as suas metades trocadas;
- (ii) $\lambda_{En}^2 = \lambda_S^1 \oplus (\text{metade esquerda de } \Omega_L)$ ou $\lambda_{En}^2 = (\lambda_S^1 \oplus (\text{metade esquerda de } \Omega_L))^*$;
- (iii) λ_{En}^n é a metade direita de Ω_C ;
- (iv) $\lambda_{En}^{n-1} = \lambda_S^n \oplus (\text{metade esquerda de } \Omega_C)$ e
- (v) para todo i tal que $i \in \{2, 3, \dots, n-1\}$,

$$\lambda_S^i = \lambda_{En}^{i-1} \oplus \lambda_{En}^{i+1}$$

ou

$$\lambda_S^i = \lambda_{En}^{i-1} \oplus (\lambda_{En}^{i+1})^*$$

Até o presente momento conhecemos poucos resultados para essas características, cujas demonstrações não serão apresentadas aqui por brevidade. Alguns são:

- Não existe característica iterativa de tamanho dois para o FEAL-N(X)/RP cuja probabilidade média seja maior que 2^{-4} ;

- Se uma característica iterativa de tamanho três possuir uma iteração nula então ela também é uma característica para o FEAL-N(X). Uma característica desse tipo é baseada em um ponto fixo (um ponto fixo é uma seqüência de bits que pode causar ela mesma pela função f [Kn92]). Não sabemos se o FEAL possui ou não algum ponto fixo diferente do zero, mas, caso exista, temos fortes indicações de que essa iteração deve ter uma probabilidade bastante baixa.

- Uma característica iterativa de tamanho quatro para o FEAL-N(X)/RP não possui nenhuma iteração nula.

Esperamos, em breve, conseguirmos obter resultados mais significativos para a nova versão do FEAL. No apêndice do capítulo, descrevemos algumas das melhores iterações que uma característica desse tipo poderia conter.

5.5 Apêndice

Fornecemos nesse apêndice a forma de algumas das iterações mais significativas que uma característica iterativa para o FEAL-N(X)/RP pode conter. Esperamos com isso proporcionar algumas informações úteis para que seja feita uma melhor análise da segurança do novo sistema criptográfico.

5.5.1 Descrição de X e Y tais que $p_f(X, Y) = 1$ e $Par(X) \wedge Par(Y)$.

$$X = 00_x 00_x 00_x 00_x \text{ e } Y = 00_x 00_x 00_x 00_x.$$

$$X = 80_x 80_x 80_x 80_x \text{ e } Y = 02_x 00_x 00_x 02_x.$$

5.5.2 Descrição de X e Y tais que $p_f(X, Y) = \frac{1}{2}$ e $Par(X) \wedge Par(Y)$.

$$X = 00_x 00_x X_3 X_3 \text{ e } Y = 00_x 00_x 00_x Y_3, \text{ onde } X_3 = 2^7 + 2^j \text{ e } Y_3 = 2 + 2^{(j+2) \bmod 8}, \quad 0 \leq j \leq 6, \text{ ou } X_3 = 2^6 \text{ e } Y_3 = 3.$$

$$X = 80_x 80_x X_3 X_3 \text{ e } Y = 02_x 00_x 00_x Y_3, \text{ onde } X_3 = 2^j \text{ e } Y_3 = 2^{(j+2) \bmod 8}, \quad 0 \leq j \leq 6, \text{ ou } X_3 = 2^7 + 2^6 \text{ e } Y_3 = 1.$$

$$X = 00_x 80_x X_3 \oplus 80_x X_3 \text{ e } Y = 00_x 00_x 02_x Y_3, \text{ onde } X_3 = 00_x \text{ e } Y_3 = 08_x, \text{ ou } X_3 = 82_x \text{ e } Y_3 = 02_x.$$

$$X = 80_x 00_x X_3 \oplus 80_x X_3 \text{ e } Y = 02_x 00_x 02_x Y_3, \text{ onde } X_3 = 02_x \text{ e } Y_3 = 00_x, \text{ ou } X_3 = 80_x \text{ e } Y_3 = 0A_x.$$

$$X = X_0 X_0 00_x 00_x \text{ e } Y = Y_0 00_x 00_x 00_x, \text{ onde } X_0 = 2^7 + 2^j \text{ e } Y_0 = 2 + 2^{(j+2) \bmod 8}, \quad 0 \leq j \leq 6, \text{ ou } X_0 = 2^6 \text{ e } Y_0 = 3.$$

$$X = X_0 X_0 80_x 80_x \text{ e } Y = Y_0 00_x 00_x 02_x, \text{ onde } X_0 = 2^j \text{ e } Y_0 = 2^{(j+2) \bmod 8}, \quad 0 \leq j \leq 6, \text{ ou } X_0 = 2^7 + 2^6 \text{ e } Y_0 = 1.$$

5.5.3 Descrição de X e Y tais que $p_f(X, Y) \geq \frac{1}{4}$ e $Par(X) \wedge Par(Y) \wedge Sim(Y)$.

$$X = 80 \ 00 \ 82 \ 02_x, Y = 02 \ 00 \ 02 \ 00_x, p_f(X, Y) = \frac{1}{2}.$$

$$X = 80 \ 00 \ 86 \ 06_x, Y = 02 \ 00 \ 02 \ 00_x, p_f(X, Y) = \frac{1}{4}.$$

Capítulo 6

Resultados Experimentais para o FEAL-N/RP

Neste capítulo, apresentaremos os resultados dos testes feitos com o FEAL-N/RP. Todos os programas foram escritos em linguagem C e executados em estações de trabalho SUN. Para a geração de dados aleatórios foi utilizada a função `drand48()` a qual gera números pseudo-aleatórios uniformemente distribuídos no intervalo $[0,1)$ através do método congruencial linear com aritmética de inteiros de 48 bits. Os valores das constantes multiplicativa e aditiva utilizados foram, respectivamente, $5DEECE66D_x$ e B_x . A semente inicial foi determinada pela função `srand48()`.

6.1 Análise do número de trocas

O objetivo de tal análise é comprovar através de simulações em computador que o FEAL-N(X) é um bom gerador de dados aleatórios. O comportamento da função Tr foi analisado isoladamente e dentro da cifra FEAL. Para a função Tr foram geradas 10 amostras de tamanho 1.000.000 e calculada a fração dos textos que sofreram troca entre as metades. Os valores obtidos para cada amostra foram: 0.49958, 0.50074, 0.50003, 0.49988, 0.50143, 0.50029, 0.49975, 0.50093, 0.49904, 0.49953, cuja média aritmética é 0.500119799.

O FEAL-N/RP foi analisado para os valores de N iguais a 4, 8, 16 e 32. Foram feitos três tipos de análise: textos aleatórios com chave secreta fixada, texto fixados com chaves secretas aleatórias e textos aleatórios com chaves secretas aleatórias. Os valores que são fixos também foram obtidos aleatoriamente. Em cada uma das análises, foram utilizadas amostras de tamanho 100.000 e contados o número de trocas em cada iteração e o número de textos que sofreram i trocas, $0 \leq i \leq N$ durante os ciframentos. Os resultados obtidos foram:

Análise do Número de Trocas para o FEAL-4/RP
Tipo: Textos e chaves secretas aleatórias
Tamanho da amostra: 100.000

Iteração	Número de Trocas
01	49868
02	49994
03	50004
04	50137

Trocas	Número de Ciframentos
00	6248
01	25050
02	37481
03	24893
04	6328

Análise do Número de Trocas para o FEAL-8/RP
 Tipo: Textos e chaves secretas aleatórias
 Tamanho da amostra: 100.000

Iteração	Número de Trocas
01	50080
02	49767
03	50258
04	50068
05	49813
06	50009
07	49821
08	50157

Trocas	Número de Ciframentos
00	392
01	3131
02	10900
03	21947
04	27363
05	21783
06	10946
07	3146
08	392

Análise do Número de Trocas para o FEAL-16/RP

Tipo: Textos e chaves secretas aleatórias

Tamanho da amostra: 100.000

Iteração	Número de Trocas
01	50180
02	49974
03	50103
04	49955
05	49832
06	50112
07	50365
08	49679
09	50115
10	49933
11	50285
12	50206
13	50160
14	49749
15	49727
16	50122

Trocas	Número de Ciframentos
00	0
01	20
02	182
03	861
04	2708
05	6640
06	12246
07	17448
08	19569
09	17503
10	12305
11	6732
12	2746
13	831
14	186
15	22
16	1

Análise do Número de Trocas para o FEAL-32/RP
Tipo: Textos e chaves secretas aleatórias
Tamanho da amostra: 100.000

Iteração	Número de Trocas
01	50296
02	50201
03	49977
04	49845
05	50185
06	49945
07	49978
08	49983
09	50287
10	49900
11	50137
12	50318
13	50185
14	50116
15	50122
16	49933
17	49990
18	49872
19	50009
20	49837
21	49834
22	50247
23	50023
24	50034
25	49962
26	50233
27	49799
28	49718
29	50144
30	50168
31	50402
32	50101

Trocas	Número de Ciframentos
00	0
01	0
02	0
03	0
04	1
05	4
06	23
07	83
08	216
09	657
10	1499
11	2945
12	5278
13	7989
14	10923
15	13115
16	14044
17	13229
18	10966
19	8053
20	5375
21	3016
22	1556
23	684
24	244
25	78
26	20
27	2
28	0
29	0
30	0
31	0
32	0

Análise do Número de Trocas para o FEAL-4/RP

Tipo: Textos aleatórios com chave secreta fixa

Tamanho da amostra: 100.000

Chave: 4F 96 18 9D 0F 4B 45 55

Iteração	Número de Trocas
01	49972
02	50169
03	49936
04	49928

Trocas	Número de Ciframentos
00	6162
01	24995
02	37690
03	24982
04	6171

Análise do Número de Trocas para o FEAL-8/RP

Tipo: Textos aleatórios com chave secreta fixa

Tamanho da amostra: 100.000

Chave: AD D8 72 F1 A1 6C 37 8E

Iteração	Número de Trocas
01	50064
02	49822
03	49897
04	50054
05	49857
06	50108
07	49826
08	50230

Trocas	Número de Ciframentos
00	398
01	3163
02	10918
03	21812
04	27357
05	22025
06	10802
07	3142
08	383

Análise do Número de Trocas para o FEAL-16/RP

Tipo: Textos aleatórios com chave secreta fixa

Tamanho da amostra: 100.000

Chave: C0 D2 F6 15 82 4B D4 94

Iteração	Número de Trocas
01	50051
02	49811
03	49822
04	49951
05	50085
06	49861
07	49809
08	49952
09	49986
10	49942
11	50014
12	50023
13	49999
14	49813
15	49718
16	49831

Trocas	Número de Ciframentos
00	3
01	16
02	169
03	876
04	2887
05	6661
06	12339
07	17595
08	19447
09	17458
10	12118
11	6665
12	2739
13	826
14	181
15	19
16	1

Análise do Número de Trocas para o FEAL-32/RP

Tipo: Textos aleatórios com chave secreta fixa

Tamanho da amostra: 100.000

Chave: 5A E1 4C A2 45 78 91 EB

Iteração	Número de Trocas
01	50148
02	50059
03	50018
04	50365
05	49999
06	49725
07	50184
08	49951
09	50102
10	49682
11	50212
12	50006
13	50222
14	49905
15	50051
16	49940
17	50005
18	49988
19	50064
20	49841
21	50141
22	50029
23	49986
24	50298
25	49940
26	49858
27	50069
28	49855
29	50053
30	49877
31	50095
32	49926

Iteração	Número de Trocas
00	0
01	0
02	0
03	0
04	0
05	8
06	20
07	80
08	274
09	644
10	1444
11	3069
12	5268
13	8034
14	10816
15	13237
16	13984
17	13163
18	11015
19	8183
20	5251
21	3023
22	1451
23	666
24	245
25	92
26	28
27	3
28	2
29	0
30	0
31	0
32	0

Análise do Número de Trocas para o FEAL-4/RP
Tipo: Texto fixado com chaves secretas aleatórias
Tamanho da amostra: 100.000
Texto: 32 70 B8 55 19 32 FA C1

Iteração	Número de Trocas
01	50198
02	50166
03	50083
04	50024

Trocas	Número de Ciframentos
00	6013
01	24994
02	37752
03	24991
04	6250

Análise do Número de Trocas para o FEAL-8/RP

Tipo: Texto fixado com chaves secretas aleatórias

Tamanho da amostra: 100.000

Texto: AD 73 1F F0 8B EA C0 AE

Iteração	Número de Trocas
01	50228
02	49964
03	50060
04	50160
05	50023
06	50304
07	49928
08	49981

Trocas	Número de Ciframentos
00	378
01	3187
02	10713
03	21800
04	27404
05	22038
06	10908
07	3195
08	377

Análise do Número de Trocas para o FEAL-16/RP

Tipo: Texto fixado com chaves secretas aleatórias

Tamanho da amostra: 100.000

Texto: EC BA 6E DF D5 DF 7A FB

Iteração	Número de Trocas
01	49984
02	50035
03	50133
04	50286
05	49920
06	50033
07	50117
08	49927
09	50015
10	49986
11	50111
12	50104
13	50146
14	49948
15	49872
16	49935

Trocas	Número de Ciframentos
00	1
01	28
02	178
03	885
04	2753
05	6511
06	12162
07	17718
08	19458
09	17492
10	12277
11	6679
12	2787
13	855
14	184
15	30
16	2

Análise do Número de Trocas para o FEAL-32/RP

Tipo: Texto fixado com chaves secretas aleatórias

Tamanho da amostra: 100.000

Texto: 00 72 9C B7 2F 9B 8D 14

Iteração	Número de Trocas
01	49859
02	49816
03	49576
04	50106
05	49851
06	50059
07	49888
08	49950
09	49967
10	50070
11	50132
12	49987
13	50274
14	50104
15	49942
16	49823
17	49923
18	49802
19	49946
20	49915
21	50009
22	50160
23	49757
24	49878
25	49853
26	49758
27	50196
28	50169
29	49945
30	49979
31	49921
32	49892

Trocas	Número de Ciframentos
00	0
01	0
02	0
03	0
04	0
05	3
06	16
07	91
08	229
09	716
10	1511
11	3043
12	5294
13	8135
14	10988
15	13097
16	14095
17	13109
18	10945
19	7932
20	5306
21	3047
22	1475
23	657
24	235
25	52
26	18
27	5
28	1
29	0
30	0
31	0
32	0

Como podemos observar para todas as cifras, em cada iteração aproximadamente metade dos textos possuem as suas metades trocadas, a média aritmética aproximada do número de trocas por ciframentos é $N/2$ e a probabilidade de que um texto sofra i trocas durante o processo de ciframento é, aproximadamente, $\binom{N}{i} 2^{-N}$.

6.2 Análise das características de Biham e Shamir

Para cada característica apresentada por Biham e Shamir em [BS91a] foram gerados 100.000 pares de textos aleatórios e feita a contagem por iteração do número de pares que satisfazem o valor da diferença da característica na entrada da função f . Os resultados obtidos são dados a seguir. Observemos que esses resultados estão de acordo com a análise teórica feita no capítulo 5.

Característica 1-1

Iteração	FEAL-N	FEAL-N/RP
1	100000	100000

Característica 1-2

Iteração	FEAL-N	FEAL-N/RP
1	100000	100000

Característica 1-3

Iteração	FEAL-N	FEAL-N/RP
1	100000	50224

Característica 1-4

Iteração	FEAL-N	FEAL-N/RP
1	100000	50143

Característica 5-1

Iteração	FEAL-N	FEAL-N/RP
1	100000	49579
2	100000	0
3	25142	0
4	25142	0
5	6298	0

Característica 6-1

Iteração	FEAL-N	FEAL-N/RP
1	100000	50052
2	100000	0
3	24994	0
4	24994	0
5	6315	0
6	6315	0

Característica iterativa

Iteração	FEAL-N	FEAL-N/RP
1	100000	49996
2	25109	0
3	6260	0
4	1529	0

Conclusões Finais

Neste capítulo, vamos fazer um resumo geral do que foi feito nesta dissertação e citar alguns dentre os vários estudos que ainda podem ser feitos. No capítulo 1, apresentamos os conceitos básicos da Criptologia. No capítulo 2, foram descritos detalhadamente os algoritmos de ciframento, deciframento e de geração de subchaves da família de cifras FEAL. No terceiro capítulo, foram apresentados os conceitos da Criptoanálise Diferencial, os quais são válidos para todas as cifras de Feistel.

No quarto capítulo, vimos como o método da Criptoanálise Diferencial se aplica à família de cifras FEAL. Em nossa descrição do método, procuramos completar algumas lacunas existentes na publicação original, tentando assim tornar o método mais fácil de ser entendido e mais coerente para o leitor.

No quinto capítulo, propusemos a introdução da função de troca Tr na metade direita do bloco de entrada de cada iteração do FEAL-N(X) com o objetivo de reduzir a probabilidade das características de Biham e Shamir, criando assim uma nova versão da cifra a qual denominamos FEAL-N(X)/RP. Mostramos que essa nova cifra é mais forte do que o FEAL para qualquer ataque baseado em uma característica para a cifra original e citamos alguns resultados que conhecemos para um outro tipo de característica que pode ser explorado para a nova versão. No capítulo seguinte, apresentamos algumas simulações que comprovam a aleatoriedade na função Tr inserida no FEAL e a análise teórica feita para as características.

Cabe notar que nos estudos feitos para a função Tr quer seja como função isolada quer como parte de uma cifra não levamos em consideração a definição da função f . Assim, essa alteração pode ser introduzida em qualquer cifra de Feistel e para a qual valerá a mesma análise feita para as características. Como exemplo, temos:

- Se essa função de troca for introduzida no DES[NBS77] então duas das três características iterativas propostas em [BS91] são anuladas e a outra sofrerá uma redução de probabilidade por um fator de $\frac{1}{2}$ a cada duas iterações;
- Se Tr for introduzida em LOKI[BPS90] então as duas características iterativas apresentadas em [BS91b] deixarão de ser válidas.

Dentre os estudos que podem ser feitos para o FEAL-N(X)/RP podemos destacar:

- Encontrar um limitante para a probabilidade média de uma característica iterativa ou encontrar uma característica iterativa cuja probabilidade média seja razoável (algo

maior que 2^{-4});

- Alterar a função S para uma função que se aproxime mais do conceito de função completa e que também seja eficientemente implementável em software. Analisar o comportamento em relação a um ataque diferencial;

- Analisar a resistência da nova cifra para um ataque diferencial que utilize características condicionais como descrito em [AB93];

- Analisar a resistência da nova cifra para um ataque baseado em uma melhor aproximação linear do algoritmo de ciframento (Esse tipo de criptoanálise é conhecido como Criptoanálise Linear e é mais forte do que a Criptoanálise Diferencial. A melhor criptoanálise desse tipo publicada é feita para o DES[NBS77] em [Ma93]. Uma boa aproximação linear para a função S_i do FEAL é dada em [TG91]).

No apêndice do capítulo 5 foram apresentadas algumas das melhores iterações que uma característica para a família de cifras FEAL pode possuir. Esperamos que com isso estejamos facilitando o trabalho daqueles que pretendem analisar mais profundamente a segurança do FEAL-N(X)/RP.

Referências Bibliográficas

Existem dois congressos internacionais de Criptologia que são realizados anualmente, Crypto, nos Estados Unidos, e Eurocrypt, na Europa. Os trabalhos apresentados nesses congressos são publicados na série Lecture Notes in Computer Science.

[AB93] Ben-Aroya, I. e Biham, E., "Differential Cryptanalysis of Lucifer", Proceedings of Crypto '93, 1993.

[BDB88] Boer, B. D., "Cryptanalysis of F.E.A.L.", Proceedings of Eurocrypt '88, pp. 293-299, 1988.

[BKPS91] Brown, L., Kwan, M., Pierprzyk, J. e Seberry, J., "Improving resistance to differential cryptanalysis and the redesign of LOKI", Proceedings of Asiacrypt '91, pp. 25-30, 1991.

[BP82] Becker, H., Piper, F., "Cipher Systems", John Wiley & Sons, 1982.

[BPS90] Brown, L., Pierprzyk, J. and Seberry, J., "LOKI - A cryptographic primitive for authentication and secrecy applications", Proceedings of Auscrypt '90, pp. 229-236, 1990.

[BS91] Biham, E. e Shamir, A., "Differential Cryptanalysis of DES-like Cryptosystems", Journal of Cryptology, pp. 3-72, 1991.

[BS91a] Biham, E. e Shamir, A., "Differential Cryptanalysis of Feal and N-Hash", Proceedings of Eurocrypt '91, pp. 1-16, 1991.

[BS91b] Biham, E. e Shamir, A., "Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer", Proceedings of Crypto '91, 1991.

[De82] Denning, D. E. R., "Cryptography and Data Security", Addison-Wesley Publishing Company, 1982.

- [Fu87] Fumy, W., "On the F-function of FEAL", Proceedings of Crypto '87, pp. 434-437, 1987.
- [GC90] Gilbert, H. e Chassé, G., "A statistical attack of the FEAL-8 Cryptosystem", Proceedings of Crypto '90, pp. 22-33, 1990.
- [Kn92] Knudsen, L. R., "Iterative Characteristics of DES and S^2 -DES", Proceedings of Crypto '92, pp. 1-11, 1992.
- [Ko81] Konheim, A. G., "Cryptography, A Primer", John Wiley & Sons, 1981.
- [KT93] Koyama, K. e Terada, R. "How to strengthen DES-like Cryptosystems against Differential Cryptanalysis", IEICE Trans. Fundamentals, pp. 63-69, 1993.
- [LM91] Lai, X. e Massey, J. L., "Markov ciphers and differential cryptanalysis", Proceedings of Eurocrypt '91, pp. 17-31, 1991.
- [Ma93] Matsui, M., "Linear Cryptanalysis Method for DES Cipher", Pre-proceedings of Eurocrypt '93, pp. 112-123, 1993.
- [Mu90] Murphy, S., "The Cryptanalysis of FEAL-4 with 20 Chosen Plaintexts", Journal of Cryptology, pp. 145-154, 1990.
- [Mi90] Miyaguchi, S., "The FEAL Cipher Family", Proceedings of Crypto '90, pp. 627-638, 1990.
- [MY92] Matsui, M. e Yamagishi, A., "A new method for known-plaintext attack of FEAL cipher", Proceedings of Eurocrypt '92, pp. 77-87, 1992.
- [NBS77] National Bureau of Standards, "Data Encryption Standards", FIPS Publication 46, U. S. Dept. of Commerce, 1977.
- [ShMi87] Shimizu, A., Miyaguchi, S., "Fast Data Encipherment Algorithm FEAL", Advances in Cryptology - Eurocrypt'87, Lecture Notes in Computer Science 304.
- [MiShSh88] Miyaguchi, S., Shiraishi, A., Shimizu, A., "Fast Data Encipherment Algorithm FEAL-8", Review of the Electrical Communication Laboratories, vol. 36-4, 1988.
- [TG91] Tardy-Corffdir, A. e Gilbert H., "A known plaintext attack of FEAL-4 and FEAL-6", Proceedings of Crypto '91, pp. 172-182, 1991.
- [TG93] Terada, R., Guimarães, J. C. F., "Proposta de Fortalecimento do Sistema Criptográfico DES contra a Criptoanálise Diferencial" (em preparação).