FEAL

A cifra «**F**ast Data **E**ncipherment **Al**gorithm (FEAL)» é uma família de algoritmos que têm tido um papel importante no desenvolvimento e refinamento de vários avanços nas técnicas de criptoanálise, incluindo a criptoanálise linear e diferencial.

FEAL-N aplica um texto claro de 64 bits num texto cifrado de 64 bits através de uma chave secreta de 64 bits. É uma cifra Feistel com N rodadas similar à cifra DES, mas com uma função f bastante mais simples, e aumentado por estágios iniciais e finais os quais fazem o XOR das duas metades da informação assim como o XOR das sub-chaves directamente com as metades da informação.

Cifra FEAL

FEAL foi concebido tendo em vista a velocidade e simplicidade, em especial para uma implementação de software em processadores de 8 bits (por exemplo, «chips» de cartões).

Usa operações «byte-oriented» (adições 8 bit módulo 256, rotações de dois bits para a esquerda, e XOR), evitando permutações de bits e procuras em tabelas, permitindo implementações com uma dimensão do código pequena.

A versão comercial inicialmente proposta com 4 rodadas (FEAL-4), posicionada como uma alternativa rápida ao DES, verificou-se no entanto ser consideravelmente menos segura do que se esperava.

FEAL

Verificou-se também que FEAL-8 oferecia muito menos segurança do que se pensava.

FEAL-16 ou FEAL-32 podem ainda oferecer uma segurança comparável ao DES, mas o débito da cifra reduz-se com o aumento do número de rodadas.

Além disso, enquanto a velocidade das implementações do DES pode ser melhorada através da utilização de tabelas de procura de grande dimensão, isso é mais difícil de obter com o FEAL.

Funções f

O algoritmo abaixo especifica a cifra FEAL-8. A função $f(A, Y)$ aplica um par de entrada de (32,16) bits numa saída de 32 bits. Na função f, duas substituições orientadas ao «byte» (caixas S) S_0 e S_1 são usadas, cada uma, duas vezes; sendo que cada uma delas aplica um par de entrada de 8 bits a uma saída de 8 bits.

S_0 e S_1 adicionam um único bit $d \in \{0, 1\}$ aos argumentos x e y de 8 bits, ignoram o transporte no último bit, rodam o resultado de dois bits para a esquerda (ROT2):

$$S_d(x, y) = \mathrm{ROT2}(x + y + d \bmod 256)$$

Sub-Chaves

A criação e ordenação das sub-chaves usa a função $f_K(A, B)$ similar à função f, aplicando duas entradas de 32 bits numa saída de 32 bits.

	$U \leftarrow f(A, Y)$	$U \leftarrow f_k(A, B)$
$t_1 =$	$(A_0 \oplus A_1) \oplus Y_0)$	$A_0 \oplus A_1$
$t_2 =$	$(A_2 \oplus A_3) \oplus Y_1)$	$A_2 \oplus A_3$
$U_1 =$	$S_1(t_1, t_2)$	$S_1(t_1, t_2 \oplus B_0)$
$U_2 =$	$S_0(t_2, U_1)$	$S_0(t_2, U_1 \oplus B_1)$
$U_0 =$	$S_0(A_0, U_1)$	$S_0(A_0, U_1 \oplus B_2)$
$U_3 =$	$S_1(A_3, U_2)$	$S_1(A_3, U_2 \oplus B_3)$

Figura: Saída: $U = (U_0U_1U_2U_3)$ para as funções FEAL f e f_K

A_i, B_i, Y_i, t_i, e U_i são variáveis de 8 bits.

Algoritmo FEAL

Algoritmo FEAL-8

ENTRADA: texto claro de 64 bits $M = m_1 \dots m_{64}$; chave de 64 bits $K = k_1 \dots k_{64}$
SAÍDA: texto cifrado de 64 bits $C = c_1 \dots c_{64}$.

1. Calcular dezasseis sub-chaves de 16 bits K_i a partir de K.
2. Definir $M_L = m_1 \dots m_{32}$, $M_R = m_{33} \dots m_{64}$.
3. $(L_0R_0) \leftarrow (M_LM_R) \oplus ((K_8K_9)(K_{10}K_{11}))$, XOR com o penúltimo grupo de quatro sub-chaves (8 a 11).
4. $R_0 \leftarrow R_0 \oplus L_0$.

Algoritmo FEAL (continuação)

Algoritmo FEAL-8 (continuação)

5. Para i de 1 a 8 faz:
   1. $L_i \leftarrow R_{i-1}$
   2. $R_i \leftarrow L_{i-1} \oplus f(R_{i-1}, K_{i-1})$.

   Usando a tabela apropriada para a obtenção de $f(A, Y)$ com $A = R_{i-1} = (A_0A_1A_2A_3)$ e $Y = K_{i-1} = (Y_0Y_1)$.
6. $L_8 \leftarrow L_8 \oplus R_8$.
7. $(R_8L_8) \leftarrow (R_8L_8) \oplus ((K_{12}K_{13})(K_{14}K_{15}))$. XOR com o último grupo de quatro sub-chaves (12 a 15).
8. $C \leftarrow (R_8L_8)$. A ordem final dos blocos é trocada.

Determinação das Sub-chaves

FEAL-8: Determinação das Sub-chaves

ENTRADA: Chave de 64 bits, $K = k_1 \dots k_{64}$.
SAÍDA: Chave estendida de 256 bits (16 sub-chaves de 16 bits K_i, $0 \le i \le 15$).

1. Inicialização:
   $U^{(-2)} \leftarrow 0$, $U^{(-1)} \leftarrow k_1 \dots k_{32}$, $U^{(0)} \leftarrow k_{33} \dots k_{64}$.
2. Calcular $K_0, \dots, K_{15}$ com i de 1 a 8:
   1. $U \leftarrow f_K(U^{(i-2)}, U^{(i-1)} \oplus U^{(i-3)})$.
   2. $K_{2i-2} = (U_0U_1)$, $K_{2i-1} = (U_2U_3)$, $U^{(i)} \leftarrow U$.

Com f_K definida pela tabela apropriada, aonde A e B denotam vectores de 4 «bytes»: $A = U^{(i-2)} = (A_0A_1A_2A_3)$;
$B = U^{(i-1)} \oplus U^{(i-3)} = (B_0B_1B_2B_3)$.
Com $U \stackrel{\text{def}}{=} (U_0U_1U_2U_3)$ para U_i com 8 bits.

Desencriptação & Generalizações

Nota (FEAL, Desencriptação)

A desencriptação da cifra FEAL pode se obtida usando o mesmo algoritmo, com a mesma chave e texto cifrado $C = (R_8, L_8)$ como texto claro de entrada M, mas com a ordem das sub-chaves trocada. Mais especificamente, as sub-chaves $((K_{12}K_{13})(K_{14}K_{15}))$ são usadas no XOR inicial, as sub-chaves $((K_8K_9)(K_{10}K_{11}))$ para o XOR final, e as chaves de rodada são dadas por K_7 até K_0. Isto é análogo à desencriptação da cifra DES.

Nota (FEAL-N)

A cifra FEAL com chave de 64 bits pode ser generalizado para N rodadas, com N par. recomenda-se a utilização de um $N = 2^x$, para $x = 3$ tem-se FEAL-8. FEAL-N usa $N + 8$ sub-chaves de 16 bits: $K_0, \ldots, K_{N-1}$, nas rodadas $0 \leq 1 \leq n - 1$; $K_N, \ldots, K_{N+3}$ para o XOR inicial; e $K_{N+4}, \ldots, K_{N+7}$ para o XOR final.
O algoritmo de determinação das sub-chaves é generalizado para calcular as sub-chaves K_0 até K_{N+7}, com $1 \leq i \leq (N/2) + 4$.

Generalizações

Nota (FEAL-NX)

A extensão da cifra FEAL-N para a utilização de chaves com 128 bits é designada por FEAL-NX. A extensão é feita da seguinte forma:

- *A chave é dividida em duas metades de 64 bits $(K_L K_R)$.*
- *K_R é dividida em duas metades de 32 bits $(K_{R_1} K_{R_2})$.*
  - *para $1 \leq 1 \leq (N/2) + 4$, define-se $Q_i = K_{R_1} \oplus K_{R_2}$ para $i \equiv 1 \mod 3$; $Q_i = K_{R_1}$ para $i \equiv 2 \mod 3$; e $Q_i = K_{R_2}$ para $i \equiv 0 \mod 3$.*
  - *O segundo argumento $(U^{(i-1)} \oplus U^{(i-3)})$ de f_K no passo 2.1 do algoritmo de determinação das sub-chaves é substituído por $U^{(i-1)} \oplus U^{(i-3)} \oplus Q_i$.*

Para $K_R = 0$, FEAL-NX é igual a igual a FEAL-N com K_L a chave de 64 bits K.

Projecto Prático 1

Implemente a Cifra FEAL-8.

- Alfabeto Português incompleto $\mathcal{A} = \{\text{a-z}\}$.
- Preenchimento não âmbiguo.
- Modo de operação ECB.
- Interface Entrada/Saída: ficheiros e linha de comando.
- Algoritmo em *C++*.

Projecto Prático 2 — Resolução

Ficheiros

Makefile	Automatização dos procedimentos de compilação

Exemplo

Exemplo

Para (em hexadecimal) um texto claro[4]

$M = 00000000\ 00000000$ e chave $K = 01234567\ 89ABCDEF$,

o algoritmo de geração das sub-chaves gera as chaves
$(K_0, \ldots, K_7) = DF3BCA36\ F17C1AEC\ 45A5B9C7\ 26EBAD25$,
$(K_8, \ldots, K_{15}) = 8B2AECB7\ AC509D4C\ 22CD479B\ A8D50CB5$.

O algoritmo FEAL-8 gera o texto cifrado
$C = CEEF2C86\ F2490752$.

Para FEAL-16, o correspondente texto cifrado é
$C = 3ADE0D2A\ D84D0B6F$.

Para FEAL-32
$C = 69B0FAE6\ DDED6B0B$.

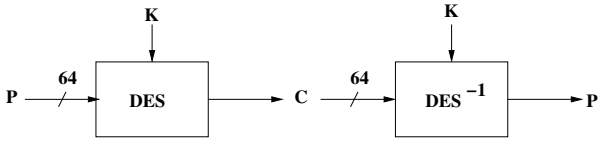
Para uma chave de 128 bits (K_L, K_R) com $K_L = K_R = K$ como se viu acima, M tem com texto cifrado FEAL-$8X$ correspondente
$C = 92BEB65D\ 0E9382FB$.

[4] $0_{16} = 0000_2, F_{16} = 1111_2$, 16 hexadecimais correspondem a 64 bits.

Data Encryption Standard

Definição (DES)

A cifra DES é uma cifra Feistel que processa blocos de texto claro de comprimento $n = 64$ bits, produzindo blocos de texto cifrado de igual comprimento. O tamanho efectivo da chave secreta K é de $k = 56$ bits; mais precisamente a chave de entrada K é especificada como uma chave de 64 bits, dos quais 8 bits (os bits, $8, 16, \ldots, 64$) podem ser usados como bits de paridade. As 2^{56} chaves implementam (no máximo) 2^{56} das 2^{64}! bijecções possíveis num bloco de comprimento 64-bits.



DES

No algoritmo DES a encriptação é efectuada em 16 estádios ou rodadas.

- Da chave de entrada K, dezasseis sub-chaves de comprimento 48-bits são geradas, uma para cada rodada.
- Em cada uma das rodadas, seleccionam-se cuidadosamente 8 substituições 6-para-4 (caixas-S) S_i, designadas no seu conjunto por S.
- O texto claro de 64 bits de comprimento é dividido em duas metades de 32 bits cada L_0 e R_0.

DES

Cada rodada é funcionalmente equivalente, obtendo da rodada anterior blocos de 32 bits L_{i-1} e R_{i-1} e produzindo blocos de 32 bits de saída L_i e R_i, para $1 \leq i \leq 16$, da seguinte forma:

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1}, K_i), \\ &\quad \text{aonde } f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i)) \end{aligned}$$

- E é uma permutação fixa com expansão transformando R_{i-1} de 32 bits para 48 bits (todos os bits são usados pelo menos uma vez).
- P é outra permutação fixa em 32 bits.
- Uma permutação inicial IP precede a primeira rodada;
- Após a última rodada, as metades esquerda e direita são trocadas;
- Finalmente a sequência resultante e permutada pelo inverso da permutação inicial.

DES

Para a desencriptação usa-se a mesma chave e o mesmo algoritmo, mas com as sub-chaves aplicadas às rodadas internas pela ordem inversa.

Uma visão simplificada é a de que a metade direita de cada rodada (após a expansão dos 32 bits para 8 caracteres de 6 bits cada) aplica uma substituição dependente da chave em cada um dos 8 caracteres, depois disso usa uma transposição fixa para redistribuir os bits dos caracteres resultantes para produzir a saída de 32 bits.

Criptoanálise

Criptoanálise das Cifras Fiestel.

M. Matsui, *Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology - EUROCRYPT'93 (LNCS no. 765), Springer-Verlag, pp. 386-397, 1994.

E. Biham and A. Shamir, *Differential Cryptanalysis of DES-like Cryptosystems*, Journal of Cryptology, vol. 4, no. 1, pp. 3-72, 1991.

Howard M. Heys, *A Tutorial on Linear and Differential Cryptanalysis*, Memorial University of Newfoundland, Canada, (`https://ioactive.com/wp-content/uploads/2015/07/ldc_tutorial.pdfInternet`)

Criptoanálise Linear

A criptoanálise linear explora a alta probabilidade de ocorrência de expressões lineares envolvendo bits do texto claro, do texto cifrado, e das sub-chaves.

É um **ataque texto claro conhecido**:

- o atacante tem à sua disposição um conjunto de textos claros e os correspondentes textos cifrados.
- o atacante não tem forma de seleccionar os textos claros (e os correspondentes textos cifrados) a que tem acesso.

Criptoanálise Linear

A ideia base é a de aproximar uma porção da cifra com uma expressão linear, sendo que a linearidade se refere a uma operação de bits, módulo 2.
Um tal expressão é da forma:

$$X_{i_1} \oplus X_{i_2} \oplus \cdots \oplus X_{i_u} \oplus Y_{j_1} \oplus Y_{j_2} \oplus \cdots \oplus Y_{j_v} = 0$$

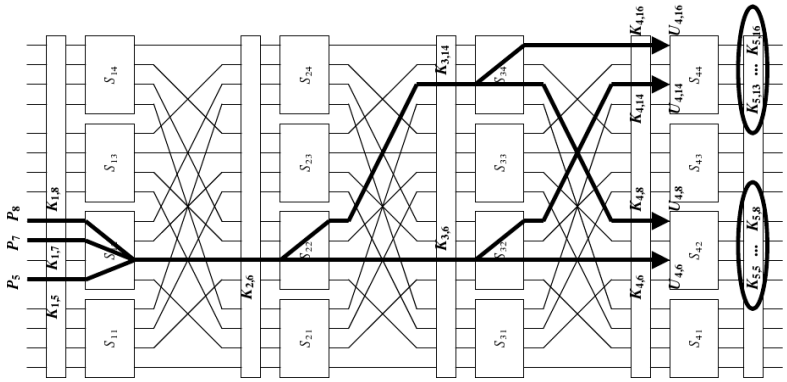
aonde X_i representa o bit de ordem i da entrada $X = [X_1, X_2, \ldots]$ e Y_j representa o bit de ordem j da saída $Y = [Y_1, Y_2, \ldots]$.

A aproximação na criptoanálise linear é a de determinar expressões da forma referida acima que tenham uma alta, ou baixa, probabilidade de ocorrência.

Considerando que num caso em que os valores escolhidos são aleatórios, a probabilidade da ocorrência de uma tal expressão é de, exactamente, $1/2$.

É o desvio, em relação ao valor de $1/2$, para a ocorrência de uma tal expressão, que é explorado na criptoanálise linear.

Criptoanálise Linear



Resistência à Criptoanálise Linear

cifra	complexidade na informação textos claros conhecidos	espaço complexidade	processamento complexidade
FEAL-4	5	30KB	6min
FEAL-6	100	100KB	40min
FEAL-8	2^{24}	280KB	10min

Criptoanálise Diferencial

A criptoanálise diferencial explora a alta probabilidade de ocorrência de relações entre diferenças entre textos claros e diferenças entre os correspondentes textos cifrados.

- É um **ataque texto claro escolhido**
  - O atacante é capaz de seleccionar textos claros e os correspondentes textos cifrados.
  - O atacante seleccionará pares de textos claros, X' e X'', que satisfaçam um dado ΔX, sabendo que para esse valor de ΔX, um dado valor de ΔY ocorre com uma probabilidade alta.

Criptoanálise Diferencial

Por exemplo, considere-se um sistema com entrada $X = [X_1 X_2 \dots X_n]$ e saída $Y = [Y_1 Y_2 \dots Y_n]$.
Sejam X' e X'' duas entradas no sistema com as correspondentes saídas Y' e Y''. A diferença nas entradas é dado por $\Delta X = X' \oplus X''$, consequentemente:

$$\Delta X = [\Delta X_1 \Delta X_2 \dots \Delta X_n], \qquad \text{com } \Delta X_i = X'_i \oplus X''_i.$$

De forma semelhante, $\Delta Y = Y' \oplus Y''$ é a diferença na saída, e

$$\Delta Y = [\Delta Y_1 \Delta Y_2 \dots \Delta Y_n] \qquad \text{com } \Delta Y_i = Y'_i \oplus Y''_i.$$

Numa cifra aleatória, a probabilidade que uma dada diferença na saída, ΔY ocorra, dado uma diferença na entrada particular ΔX é de $1/2^n$ aonde n é o número de bits de X.
A criptoanálise diferencial tenta explorar o cenário no qual uma dada diferença ΔY ocorre, dado uma diferença na entrada ΔX particular, com uma alta probabilidade p_D (i.e., bastante maior do que $1/2^n$). O par $(\Delta X, \Delta Y)$ é referido como o diferencial.