Tipos e Operações

Uma variável do tipo (*unsigned*) *int* pode ser manipulada na sua forma binária.

Operação (em bits)	Significado
&	Conjunção
|	Disjunção
^	Disjunção exclusiva (XOR)
$>> n$	Deslocamento de n bits para a direita
$<< n$	Deslocamento de n bits para a esquerda

Para lidar com inteiros com um número bem definido de "bytes" temos a biblioteca *stdint* (`#include <stdint.h>`) a qual define os seguintes tipos (com as respectivas operações).

```
int8_t  int16_t  int32_t  int64_t
uint8_t  uint16_t  uint32_t  uint64_t
```


FEAL8 - Caixas S

Na implementação das caixas S temos:

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

isto é, temos:

- Rotação de 2 bits para a esquerda;
- adição módulo 256.

Soluções (para o caso de um bloco de 8bits):

- $\mathtt{ROT}n(\mathtt{x}) = (\mathtt{x} << n) \mid (\mathtt{x} >> 8 - n)$

```
ROT2(11101001) = (11101001 << 2) | (11101001 >> 6)
                 (10100100)      |   (00000011)
                                  (10100111)
```

- Basta fazer a operação de adição em variáveis de 8bits (ignorando o transporte).

```
uint8_t d, uint8_t x, uint8_t y, uint8_t;
```


Separar "Bytes"

- Na função f_k temos duas variáveis de entrada de 32bits que têm de ser divididas em secções de 8bits.
- Na função f temos duas variáveis de entrada, uma de 32bits e outra de 16bits, ambas têm de ser divididas em secções de 8bits.

Soluções:

- "Máscaras" — designa-se por máscara ("mask") um padrão de bits que serve como forma de realçar um dado conjunto de "bits" através da conjunção binária.
  Por exemplo: 32bits quer-se realçar o segundo "byte".
  - Máscara:
    $65280_{10} = FF00_{16} = 00000000\,00000000\,11111111\,00000000_2$
  - Conjunção com a máscara: $A \leftarrow A\ \&\ Mascara$

```
  11100101 11010101 11110010 10101010
& 00000000 00000000 11111111 00000000
-------------------------------------
  00000000 00000000 11110010 00000000
```


Separar "Bytes" (continuação)

Temos então que, para dividir uma variável de 32bits em secções de 8bits, temos de fazer:

- Aplicar uma máscara apropriada e retirar o primeiro "byte";
- Deslocar a variável um "byte" para a direita;
- repetir o processo com os restantes "bytes".

```
A3 = A & 255; A >>= 8;
...
```

```
A & 255
  11100101 11010101 11110010 10101010
& 00000000 00000000 00000000 11111111
-------------------------------------
  00000000 00000000 00000000 10101010
A >>= 8
A = 00000000 11100101 11010101 11110010
```

Juntar "Bytes"

Tanto na função f_k como na função f é necessário, no fim, "juntar" secções de 8bits numa só sequência de 32bits.
Solução:

- Definir uma variável de tamanho apropriado;
- Deslocar cada uma das secções para a sua posição correcta;
- Combinar todas as secções numa só sequência.

```
(X0 << 24) | (X1 << 16) | (X2 << 8) | X3;
```

```
X0 = 11100101; X1 = 11010101; X2 = 11110010; X3 = 10101010
X0 << 24 11100101 00000000 00000000 00000000
X1 << 16 00000000 11010101 00000000 00000000
X2 <<  8 00000000 00000000 11110010 00000000
X3       00000000 00000000 00000000 10101010
|        11100101 11010101 11110010 10101010
```


Introdução às Cifras por Blocos

Definição (Cifras por Blocos)

Uma n-bit cifra por blocos é uma função $E : V_n \times \mathcal{K} \to V_n$, tal que para cada k-bit $K \in \mathcal{K}$, $E(P, K)$ é uma função invertível (a função de encriptação *para K) de V_n para V_n, denotada por, $E_K(P)$. A função inversa é a* função de desencriptação, *denotada por, $D_K(C)$.*

A expressão $C = E_K(P)$ denota o facto de que o texto cifrado C resulta da encriptação do texto claro («plaintext») P, usando-se para tal a chave K.

- n é designado o *comprimento do bloco*.
- A utilização de blocos de igual comprimento tanto para o texto claro como para o texto cifrado evita a expansão da informação.

Modos de Preenchimento

Uma cifra por blocos encripta os textos claros em blocos de comprimento fixo com n-bits (usualmente $n = 64$), para mensagens que excedam esse comprimento a mensagem é particionada em blocos de comprimento n-bits, sendo cada um dos blocos encriptado de forma separada.

Se o comprimento da mensagem não for um múltiplo de n é necessário preencher de alguma forma a mensagem de forma a que tenhamos um múltiplo de n.

Modos de Preenchimento

Método de Preenchimento 1

ENTRADA: x, texto claro; n, comprimento (em bits) do bloco.
SAÍDA: x_0, texto claro preenchido de forma a ter um comprimento múltiplo de n.

1. Concatenar a x o menor número de (possivelmente zero) 0-bits necessários para obter um texto cujo comprimento seja um múltiplo de n.

Método de Preenchimento 2

ENTRADA: x, texto claro; n, comprimento (em bits) do bloco.
SAÍDA: x_0, texto claro preenchido de forma a ter um comprimento múltiplo de n.

1. Concatenar a x um único 1-bit.
2. Concatenar de seguida o menor número de (possivelmente zero) 0-bits necessários para obter um texto cujo comprimento seja um múltiplo de n.

Prós e Contras dos Métodos de Preenchimento

Método 1

+ nunca acrescenta novos blocos à mensagem.
– é ambíguo — os eventuais 0-bits no fim do texto original não se conseguem distinguir daquelas que foram acrescentados no processo de preenchimento. Um tal método é aceitável se o comprimento do texto claro (antes do preenchimento) é sabido, por outros meios, pelo destinatário da mensagem.

Método 2

+ não é ambíguo.
– quando o comprimento do texto original é já um múltiplo do comprimento do bloco, *resulta do método a criação de um bloco extra*.

Modos de Operação

Os métodos mais usuais de particionar uma mensagem em blocos são:

ECB «Electronic CodeBook».

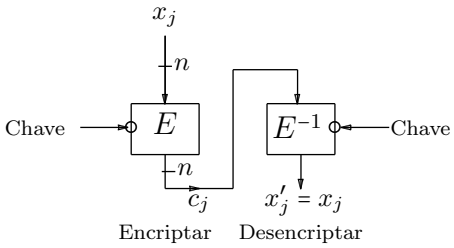
CBC «Cipher-Block Chaining».

CFB «Cipher FeedBack».

Existem outros modos: Output FeedBack (**OBF**); Counter Mode (**CTR**); Offset-Codebook Mode (**OCM**).

Modo de Operação ECB

Electronic Codebook (ECB) — uma mensagem é particionada em blocos de comprimento n-bit e estes são encriptados separadamente.



Algoritmo ECB

ENTRADA: chave de comprimento k (bits), blocos de comprimento n (em bits) de texto claro, $x_1, \ldots x_t$.
SAÍDA: produz blocos de texto cifrados $c_1, \ldots, c_t$

1. Cifrar: para $1 \leq j \leq t$, $c_j \leftarrow E_K(x_j)$.
2. Decifrar: para $1 \leq j \leq t$, $x_j \leftarrow E_K^{-1}(c_j)$.

Propriedades do Modo de Operação ECB

Propriedades do modo de operação ECB:

1. Blocos de texto claro idênticos: sob a mesma chave resultam em blocos cifrados idênticos.
2. Dependências entre blocos: os blocos são cifrados de forma independente entre si. A re-ordenação dos blocos de texto cifrado resulta no re-ordenamento dos blocos de texto claro.
3. Propagação de Erros: um ou mais erros em bits num único bloco de texto cifrado afecta somente esse bloco. Para uma cifra típica E, o desencriptar de um tal bloco é então aleatório (com cerca de 50% de recuperação de texto claro).
4. Perdas de informação: a recuperação de bits «perdidos» nas fronteiras dos blocos não é possível.

Propriedades do Modo de Operação ECB

Nota (Utilização do modo ECB)

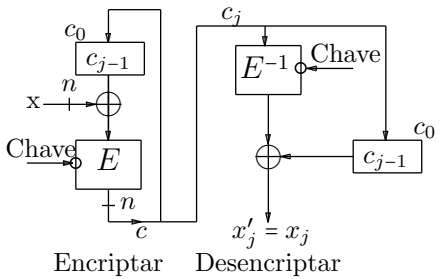
Dado que os blocos de texto cifrado são independentes uns dos outros a substituição maliciosa de blocos no modo ECB (por exemplo a inserção de um bloco muito frequente) não afecta a desencriptação dos blocos adjacentes. Mais, os blocos cifrados não escondem os padrões, isto é blocos cifrados idênticos implicam blocos de texto claro idênticos. Por esta razão o modo ECB não é recomendável para mensagens de comprimento maior do que um bloco, ou nos casos em que a chave é usada mais do que uma vez.

Nota (re-sincronização vs. erros nas fronteiras)

Por re-sincronização entende-se a recuperação por erros (sem perda de informação) nos blocos. Por erros nas fronteiras entende-se a «perda» de bits nas fronteiras dos blocos.

Modo de Operação CBC

Cipher-Block Chaining — utiliza um vector de inicialização de n-bits.



Algoritmo CBC

ENTRADA: K, chave de comprimento k (bits), n_0 bloco inicial de comprimento n-bits, blocos de comprimento n (em bits) de texto claro, $x_1, \ldots x_t$.
SAÍDA: produz blocos de texto cifrados $c_1, \ldots, c_t$.

1. Cifrar: $c_0 \leftarrow n_0$. Para $1 \leq j \leq t$, $c_j \leftarrow E_K(c_{j-1} \oplus x_j)$.
2. Decifrar: $c_0 \leftarrow n_0$. Para $1 \leq j \leq t$, $x_j \leftarrow c_{j-1} \oplus E_K^{-1}(c_j)$.

Propriedades do Modo de Operação CBC

Propriedades do Modo de Operação CBC:

1. Blocos de texto claro idênticos: sob a mesma chave e bloco inicial resultam em blocos cifrados idênticos. Mudando o bloco inicial, a chave, ou um primeiro bloco de texto claro (por exemplo, com um entrada aleatória) resulta num texto cifrado diferente.
2. Dependências entre blocos: a dependência entre blocos faz com que o texto cifrado c_j dependa de x_j e de todos os blocos de texto claro precedentes. O re-ordenar dos blocos de texto cifrado afecta a correcta decifração do texto global.
3. Propagação de erros: um erro num único bit no texto cifrado c_j afecta o decifrar dos blocos c_j e c_{j-1}, isto dado que x_j depende de c_j e de c_{j-1}. Nessas condições o bloco x_j, recuperado de c_j, é tipicamente, aleatório (50% errados), e o bloco x_{j+1} tem um bit errado precisamente aonde o bloco c_j tinha. O adversário pode então planear uma alteração no bloco x_{j+1}, alterando para tal o bloco c_j.
4. Recuperação de erros: um erro (incluindo a perda de um ou mais blocos) num bloco c_j afecta somente o bloco que se lhe segue de imediato, o bloco c_{j+2} não é afectado pelo erro ocorrido em c_j.

Propriedades do Modo de Operação CBC

Nota (Propagação de Erros na Encriptação)

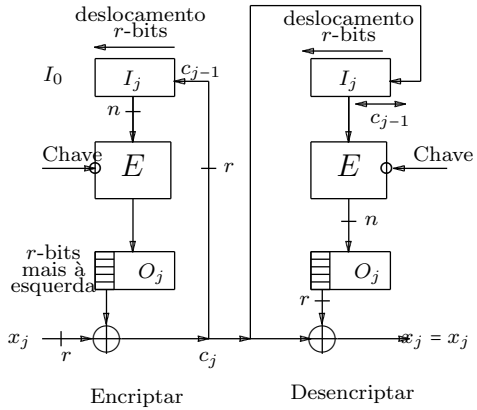
Embora o modo CBC consiga recuperar de erros nos blocos cifrados, modificações no bloco de texto claro x_j durante a encriptação alteram os blocos cifrados subsequentes. Isto tem efeitos na usabilidade dos modos com encadeamento para aplicações que requeiram acessos aleatórios de leitura/escrita à informação encriptada. O modo ECB é uma alternativa nesses casos.

Nota (Integridade do Vector de Inicialização em CBC)

Enquanto que o vector de inicialização no modo CBC não necessita de permanecer secreto, a sua integridade tem de ser protegida, isto dado que modificações maliciosas ao mesmo permitem ao adversário fazer modificações (em bits) previsíveis ao primeiro bloco de texto claro recuperado. Usar um vector de inicialização secreto é uma forma de impedir isso, no entanto se a integridade da mensagem é um requerimento é necessário usar um outro tipo de mecanismo.

Modo de Operação CFB

Cipher Feedback (CFB) — r-bit caracteres/r-bit re-alimentação. Enquanto o modo CBC processa o texto claro n bits de cada vez (usando uma cifra de blocos de comprimento n), algumas aplicações podem requerer que um bloco de r-bits seja encriptado e transmitido sem demoras, para um dado $r < n$ (usualmente $r = 1$ ou $r = 8$).



Modo de Operação CFB

Algoritmo CFB

ENTRADA: K, chave de comprimento k (bits), I_0 bloco inicial de comprimento n-bits, blocos de comprimento r-bits de texto claro, $x_1, \ldots x_t$, $1 \leq r \leq n$.
SAÍDA: produz blocos, de comprimento r-bits, de texto cifrados $c_1, \ldots, c_t$.

1. Cifrar: $I_1 \leftarrow I_0$ (I_1 é o valor no registo de deslocamento, para $1 \leq j \leq t$).
   1. $O_j \leftarrow E_k(I_j)$ (cálculo do resultado da cifra por blocos).
   2. $o_j \leftarrow$ os r-bits mais à esquerda de O_j
   3. $c_j \leftarrow x_j \oplus o_j$ (transmite o bloco cifrado, de comprimento r-bits, c_j).
   4. $I_{j+1} \leftarrow 2^r \cdot I_j + c_j \mod 2^n$ (desloca c_j para o lado esquerdo do registo de deslocamento).
2. Decifrar: $I_1 \leftarrow I_0$. Para $1 \leq j \leq t$, $x_j \leftarrow c_j \oplus o_j$, aonde o_j, O_j, e I_j são calculados da forma já descrita.

Propriedades do Modo de Operação CFB

Propriedades do modo de operação CFB.

1. Blocos de texto claro idênticos: assim como para o modo de operação CBC a mudança do bloco inicial resulta num bloco cifrado diferente. O bloco inicial não necessita de ser secreto.
2. Dependências entre blocos: similar ao modo de operação CBC, o bloco c_j depende dos blocos x_j e x_{j-1}. Consequentemente o re-ordenar dos blocos cifrados afecta a decifração. Para uma correcta decifração é necessário que os $\lceil n/r \rceil$-blocos precedentes estejam correctos (de forma a que o registo de deslocamento contenha um valor correcto).

Propriedades do Modo de Operação CFB

Propriedades do modo de operação CFB (continuação).

3. Propagação de erros: um ou mais bits num único r-bit bloco cifrado c_j afecta a desencriptação desse e dos próximos $\lceil n/r \rceil$ blocos cifrados, ou seja até que n bits de texto cifrado sejam processados, após o que o bloco com erros c_j foi deslocado para fora do registo de deslocamento. O texto claro recuperado x_j vai deferir do bloco original x_j precisamente na posição (em bits) na qual c_j contém o erro; os outros blocos de texto claro incorrectamente recuperados são tipicamente vectores aleatórios, i.e. têm 50% de bits em erro.
   Consequentemente um adversário podem causar modificações previsíveis num dado bit de x_j, por alteração do bit correspondente em c_j.

Propriedades do Modo de Operação CFB

Propriedades do modo de operação CFB (continuação).

4. Recuperação de erros: o modo CFB é auto-sincronizável, de forma similar ao CBC, mas requer $\lceil n/r \rceil$ de blocos cifrados para recuperar.
5. Débito: para $r < n$, o débito decresce por um factor de n/r (vs. CBC) dado que cada execução de E só dá origem a r bits de texto cifrado.

Nota (CFB e a Função de Encriptação)

Dado que a função de encriptação E é usada, no modo CFB, tanto para a encriptação como para a desencriptação, este modo não pode ser usado em conjunção com cifras de chave pública. Nesses casos deve-se usar o modo CBC.

Encriptação Múltipla

Se uma cifra por blocos é susceptível a um ataque por procura exaustiva da chave (devido a um comprimento da chave inadequado), então a encriptação do mesmo bloco mais do que uma vez pode aumentar a segurança.

As técnicas para a encriptação múltipla podem ser usadas em conjunção com os modos de operação já estudados, o E passa a denotar a encriptação múltipla em vez de simples.

Cifras em Cascata

Definição (Cifras em Cascata)

Uma cifra em cascata *é a concatenação de $L \geq 2$ cifras por blocos (designados por estágios) cada uma com uma chave independente. O texto claro é a entrada para o primeiro estágio, sendo a saída deste estágio a entrada do seguinte. A saída do estágio L é a saída da cifra em cascata.*

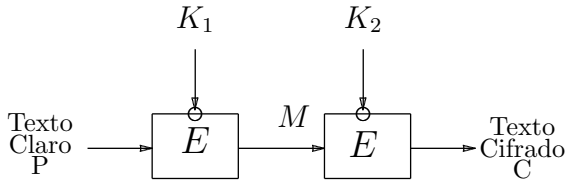
No caso mais simples, todos os estágios numa cifra em cascata têm chaves de comprimento k, e blocos de comprimento n. As cifras de cada estágio podem ser diferentes (cifra em cascata genérica), ou serem todas a mesma (cascata de cifras idênticas).

Encriptação Múltipla

Definição (Encriptação Múltipla)

A encriptação múltipla *é similar à cascata de L cifras idênticas, mas os estágios podem não ser independentes, e para um dado estágio podemos ter a função de encriptação E ou a sua correspondente inversa $D = E^{-1}$.*

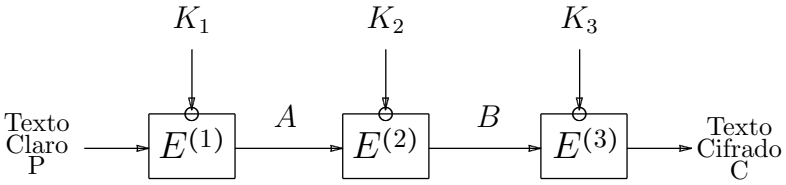
Dois casos importantes de múltipla encriptação são a dupla e tripla encriptação.



Definição (Dupla Encriptação)

A dupla encriptação *é definida por $E(x) = E_{K_2}(E_{K_1}(x))$, aonde E_k denota uma cifra por blocos E com chave K.*

Encriptação Múltipla



Definição (Tripla Encriptação)

A tripla encriptação *é definida por* $E(x) = E^{(3)}_{K_3}(E^{(2)}_{K_2}(E^{(1)}_{K_1}(x)))$, *onde* $E^{(j)}_k$ *denota ou* E_k, *ou* $D_K = E^{-1}_k$. *O caso* $E(x) = E_{K_3}(D_{K_2}(E_{K_1}(x)))$, *é usualmente designado por* E-D-E *tripla encriptação. O sub-caso* $K_1 = K_3$ *é usualmente designado por tripla encriptação de duas chaves.*

Encriptação Múltipla

Na dupla encriptação é usual usar dois estágios independentes K_1 e K_2.

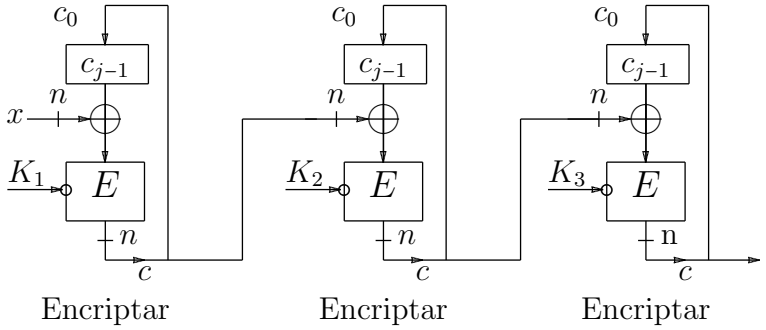
Na tripla encriptação, de forma a poupar recursos no que se refere à gestão de chaves, é usual usar chaves dependentes dos estágios.

A E-D-E tripla encriptação com $K_1 = K_2 = K_3$ é equivalente a uma encriptação simples.

Modos de Operação em Encriptação Múltipla

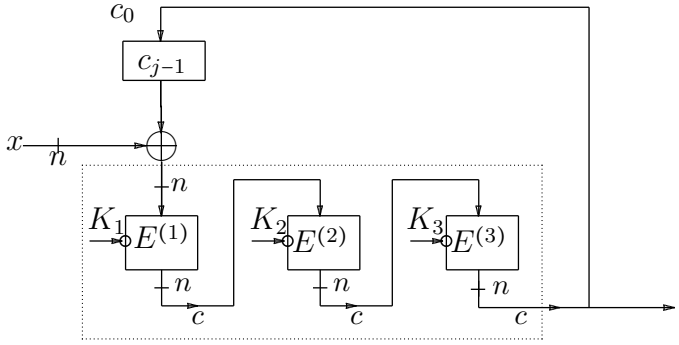
Em contraste ao modo de operação das cifras simples, os modos múltiplos são variantes de encriptações múltiplas construídas por concatenação de modos simples seleccionados.
Por exemplo, a combinação de três CBCs em modo de operação simples é designado por *CBC-triplo-interno*.

CBC-triplo-interno



Modos de Operação em Encriptação Múltipla

Uma alternativa é o modo designado por *CBC-triplo-externo*, em que temos a composição de uma encriptação tripla com um modo de operação simples, ou seja uma aplicação (externa) do modo de operação CBC, após a aplicação sequencial de três ECB.



Modos de Operação em Encriptação Múltipla

Com «hardware» específico (replicado), os modos múltiplos tais como o CBC-triplo-interno podem ser sequenciados («pipelined»), permitindo deste modo uma eficiência próxima da encriptação simples, sendo por isso mais vantajoso que o CBC-triplo-externo.

Nota (Segurança do modo CBC-triplo-interno)

Muitos dos modos múltiplos de operação são mais fracos que o correspondente modo ECB múltiplo (isto é, encriptação múltipla a funcionar como uma só cifra para o modo de operação simples externo). Em alguns casos (por exemplo, ECB-CBC-CBC) não são significativamente mais fortes que uma encriptação simples.
Em particular, sob alguns tipos de ataques, o CBC-triplo-interno é significativamente mais fraco do que o CBC-triplo-externo.

Encriptação Múltipla

Embora pareça contra-intuitivo, é possível construir exemplos aonde uma cascata de cifras reduz a segurança. No entanto, em geral tem-se que:

Facto (Segurança de Cifras em Cascata)

Uma cascata de n (com chaves independentes) cifras é pelo menos tão segura como a primeira cifra componente. Uma cascata de cifras de permutações (por exemplo cifras aditivas) é tão segura como a componente mais segura.

Composição de Cifras

Uma forma de tentar aumentar a dificuldade de cripto-análise de uma cifra é dada pela composição de cifras.

A composição de involuções não é necessariamente uma involução. No entanto as involuções podem ser facilmente compostas de forma a se obter uma função de certa forma mais complicada e que é fácil de inverter.

Por exemplo, se $E_{k_1}, E_{k_2}, \ldots, E_{k_t}$ são involuções, então a inversa de $E_k = E_{k_1}, E_{k_2}, \ldots, E_{k_t}$ é $E_k^{-1} = E_{k_t}, E_{k_{t-1}}, \ldots, E_{k_1}$, isto é a composição das involuções pela ordem inversa.

Cifras Produto

As cifras simples de substituição e de transposição não são muito seguras. No entanto a combinação delas pode criar cifras bastante mais seguras do que as cifras de partida.

Cifra Produto

Sejam $\mathcal{M} = \mathcal{C} = \mathcal{K}$ o conjunto de todas as sequências binárias de comprimento 6. O número de elementos de $\mathcal{M}$ é $2^6 = 64$. Seja $m = (m_1 m_2 \ldots m_6)$ e sejam:

$$\begin{aligned} E_k^{(1)}(m) &= m \oplus k, \text{aonde } k \in \mathcal{K}, \\ E^{(2)}(m) &= (m_4 m_5 m_6 m_1 m_2 m_3). \end{aligned}$$

A operação denotada $\oplus$ é o *ou-exclusivo (XOR)*. $E_k^{(1)}$ é uma cifra de substituição poli-alfabética, $E^{(2)}$ é uma cifra de transposição (sem chave). A cifra produto é dada por $E_k^{(1)} E^{(2)}$.

Na terminologia inglesa designa-se este tipo de cifras um «round».

Confusão e Difusão

Definição (Confusão)

Uma cifra é dita adicionar confusão *sempre que aumentar a complexidade da relação entre a chave e o texto cifrado.* Uma cifra de substituição adiciona confusão a uma cifra produto.

Definição (Difusão)

A difusão *refere-se ao espalhar dos «bits» numa mensagem de forma que qualquer redundância no texto claro seja espalhada ao longo do texto cifrado.* Uma cifra de transposição adiciona difusão a uma cifra produto.

Temos então que uma cifra produto composta de uma substituição e de uma transposição adiciona confusão e difusão ao processo de encriptação.

Cifras de Chaves Simétricas

Definição (Cifra de Chaves Simétricas)

Considere uma cifra constituída por um conjunto de funções de encriptação e desencriptação, respectivamente $\{E_e : e \in \mathcal{K}\}$ *e* $\{D_d : d \in \mathcal{K}\}$*, onde* $\mathcal{K}$ *é o espaço das chaves.*

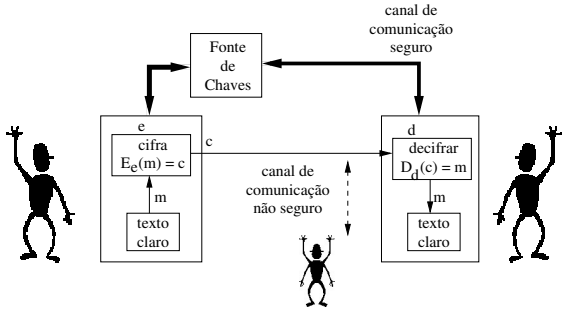
A cifra é dita uma cifra de chaves simétricas *se para cada par de chaves* (e, d)*, é computacionalmente «fácil» determinar* d *sabendo só o valor de* e *e, de igual modo, determinar* e *de* d*.*

Em muitas aplicações as chaves são iguais, $e = d$. Outros termos usados são *chave única*, *chave secreta*.

Cifras de Chaves Simétricas

Num sistema de cifra de chaves simétricas é necessário que os intervenientes tenham conhecimento das chaves simétricas (eventualmente uma só chave secreta).

A chave tem de ser enviada por um canal seguro.



Cifras de Chaves Simétricas por Blocos

As *cifras de chaves simétricas por blocos* são dos elementos mais proeminentes e importantes em muitos sistemas criptográficos.

Individualmente providenciam confidencialidade.

Como um bloco de um sistema maior a sua enorme versatilidade torna possível a construção de geradores pseudo-aleatórios, cifras fieira, MACs, e funções de dispersão.

Podem ser ainda uma das principais componentes em sistemas de autenticação de mensagens, integridade dos dados, autenticação de entidades, e esquemas de assinaturas digitais.

Cifras de Chaves Simétricas por Blocos

Não existem cifras de chaves simétricas por blocos que sejam aplicáveis a todas as situações, tal facto deve-se a:

- restrições impostas na velocidade de processamento e/ou na memória usada.
- restrições impostas pelo tipo de plataforma a ser usada (hardware e/ou software).
- diferentes níveis de tolerância das cifras a diferentes modos de utilização.

De uma forma geral às preocupações com a eficiência contrapõem-se as questões de segurança.

Cifras de Chaves Simétricas por Blocos

De entre as cifras deste tipo mais importantes temos:

- *DES* «**D**ata **E**ncryption **S**tandard», é (era!?) um das cifras deste tipo mais importantes. Estabeleceu o precedente em meados de 1970 como a primeira cifra de nível comercial com uma completa e aberta especificação dos detalhes de implementação.
- *FEAL* «**F**ast Data **E**ncipherment **Al**gorithm», é uma família de algoritmos que tiveram muita importância no desenvolvimento e melhoramento de várias técnicas de criptoanálise, nomeadamente a criptoanálise linear e diferencial.
- *AES (Rijndael)* «**A**dvanced **E**ncryption **S**tandard», foi o resultado de um esforço americano (1999) para a substituição do DES que entretanto começou a mostrar-se insuficientemente seguro.

Outras cifras deste tipo também importantes são: **IDEA** (International Data Encryption Algorithm) e **RC6**.

Cifras de Chaves Simétricas por Blocos

As cifras de chaves simétricas por blocos estão relacionadas com dois princípios gerais: cifras produto e cifras Feistel. Cada um destes princípios envolve a iteração de uma sequência comum (ou rodada) de operações.

A ideia básica de uma cifra produto é a de construir uma função de encriptação complexa por composição de várias operações simples, que embora de forma individualmente não oferecem protecção suficiente, são complementares quando combinadas podendo providenciar o grau de protecção que se deseja.
As operações básicas incluem: transposições, translações (i.e. XOR), transformações lineares ($ax + b$), multiplicação modular, e substituições simples.

Cifra Produto & Rede de Substituição-Permutação

Definição (Cifra Produto)

Uma cifra produto *combina duas ou mais transformações de tal forma que a cifra resultante seja mais segura que as componentes individuais.*

Definição (Rede de Substituição-Permutação)

Uma Rede de Substituição-Permutação *é uma cifra produto composta de um número de estágios, cada um envolvendo substituições e permutações.*

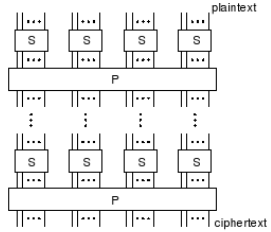


Figure 7.7: *Substitution-permutation (SP) network.*

Cifra de Blocos Iterada

Definição (Cifra de Blocos Iterada)

Uma cifra de blocos iterada *é uma cifra de blocos envolvendo a repetição sequencial de uma função interna, designada por função rodada.*

Os parâmetros de configuração da cifra incluem: o número de rodadas r, o comprimento em bits do bloco n, o comprimento em bits k, da chave de entrada K da qual r sub-chaves K_i (chaves das rodadas) são derivadas.

Por razões de invertibilidade (para permitir uma decifração única), para cada um dos valores K_i a função de rodada é uma bijecção nos valores de entrada da rodada.

Cifra Feistel

Definição (Cifra Feistel)

Uma cifra Feistel *é uma cifra de blocos iterada que aplica um texto claro de $2t$-bits (L_0R_0), sendo que L_0 e R_0 são blocos com t bits, num texto cifrado (R_rL_r), através de um processo com r rodadas, aonde $r \geq 1$.*

Para $1 \leq i \leq r$, a rodada i aplica $(L_{i-1}R_{i-1}) \xrightarrow{K_i} (L_iR_i)$ da seguinte forma: $L_i = R_{i-1}$, $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$, aonde cada uma das sub-chaves K_i é derivada da chave da cifra K.

Cifra Feistel

Usualmente uma cifra Feistel tem $r \geq 3$, sendo r, em geral, par.

A estrutura de uma cifra Feistel especifica a ordenação dos textos cifrados de saída com (R_rL_r) e não (L_rR_r); os blocos são trocados da sua ordenação após a última rodada.

A desencriptação é feita através do mesmo processo com r rodadas mas com a sub-chaves usadas por ordem inversa; por exemplo o último passo é desfeito através de uma simples repetição do mesmo.

A função f da cifra de Feistel pode ser uma cifra produto, no entanto a função f não necessita de ser invertível para que se possa inverter a cifra de Feistel.

Cifra Feistel