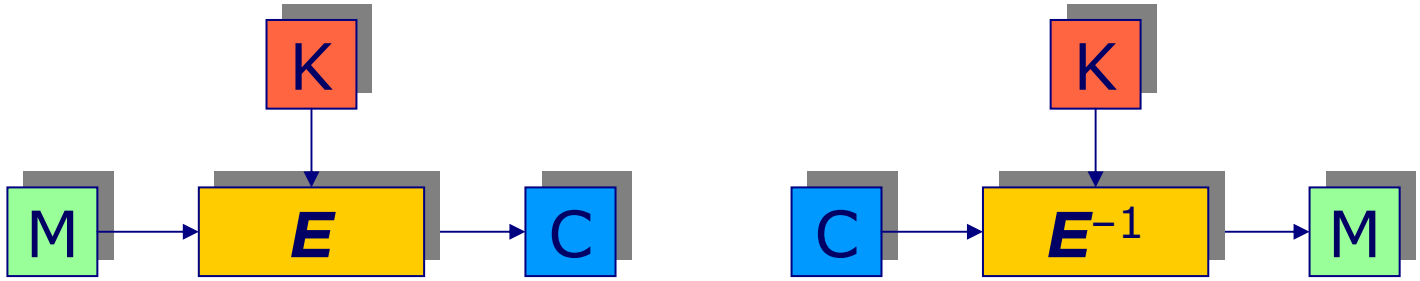
Segurança da Informação

Cifras de Fluxo e de Bloco
Evolução Histórica
Data Encryption Standard (DES)
Cifração Múltipla

Cifra Simétrica – Definição

- Uma *cifra simétrica* é uma transformação matemática inversível cujo cálculo depende, no sentido direto e no sentido inverso, de uma *mesma* informação secreta (a chave).



- Duas famílias:
  - cifras de fluxo.
  - cifras de bloco.

Cifras de Fluxo

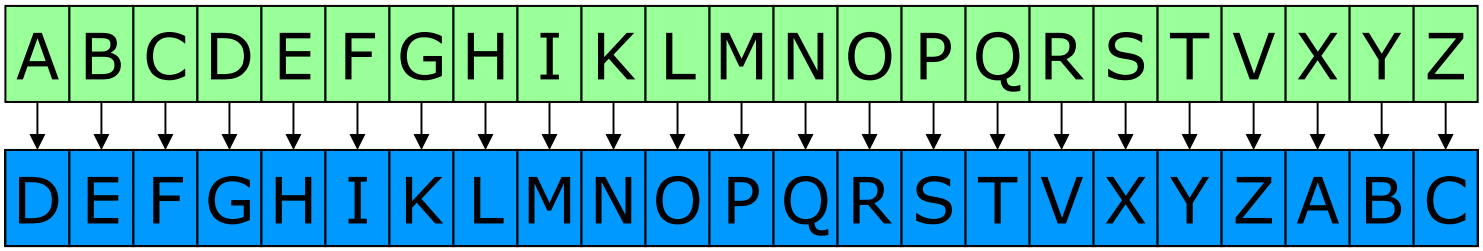
De César ao RC4™

- Evolução histórica da idéia de *cifra de fluxo* (*stream cipher*), desde o Império Romano até o advento da Internet.

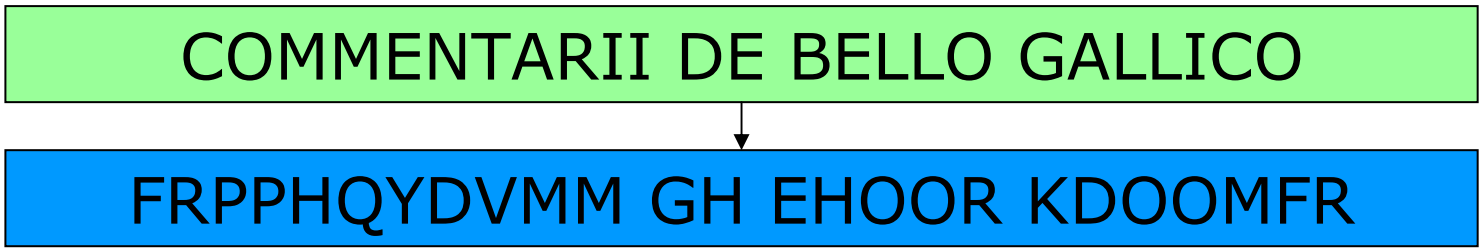
- Um algoritmo forte, inadequadamente integrado, pode conduzir a um sistema criptográfico fraco.

Cifra de César

- Técnica empregada por Júlio César para intercâmbio de mensagens com seus generais.
- Consiste em substituir cada letra da mensagem pela terceira letra correspondente no alfabeto latino.



- Exemplo:



Análise

- Definem-se:
  - M: mensagem clara (M_i = *i*-ésima letra de M)
  - C: mensagem cifrada (C_i = *i*-ésima letra de C)
  - *k*: deslocamento das letras no alfabeto (*k* = 3)

- O alfabeto latino clássico possuía *p* = 23 letras. Numerando-as de 0 a 22, a cifra de César pode ser descrita pelas fórmulas:
  - $C_i \leftarrow (M_i + k) \bmod p$
  - $M_i \leftarrow (C_i - k) \bmod p$

Generalização

- A *chave* da cifra de César é o deslocamento da posição das letras no alfabeto, e pode ser representada por uma letra, segundo a numeração (na cifra de César, $k = 3 = D$).
- Obviamente, não há nada especial no valor $k = 3$ (por exemplo, Augusto utilizava $k = 4$).
- Não precisa ser fixo: pode ser um segredo comum entre cada remetente e cada destinatário de cada mensagem.
- Vantagem: aumenta o esforço de decifração para um adversário.

Exercício

- Decifrar a seguinte mensagem:

- RH EINR HNBHN SIXH FCANA HXOX PNIGN IXQRBQX

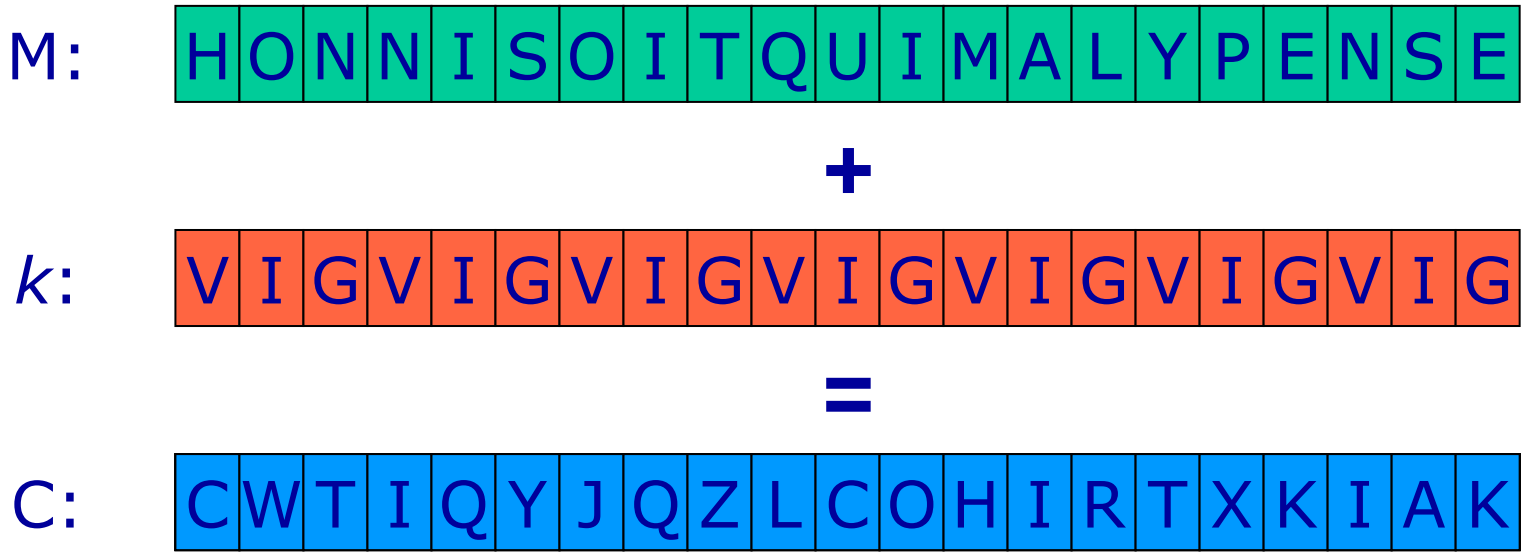
Cifra de Vigenère

- A chave pode consistir de vários caracteres: $k = k_0\, k_1 \ldots k_{m-1}$ (repetidos ciclicamente).

- *Cifra de Vigenère*:
  - $C_i \leftarrow (M_i + k_{i \bmod m}) \bmod p$
  - $M_i \leftarrow (C_i - k_{i \bmod m}) \bmod p$

- Vantagem: aumenta o esforço de decifração para um adversário (exponencial: p^m chaves possíveis).

Cifra de Vigenère



Cifra de Vernam

- Utilização de alfabetos arbitrários, contendo *p* caracteres.

- Num caso extremo, $p = 2$ (alfabeto binário): *Cifra de Vernam*:
  - $C_i \leftarrow (M_i + k_{i \bmod m}) \bmod 2 \leftarrow M_i \oplus k_{i \bmod m}$
  - $M_i \leftarrow (C_i - k_{i \bmod m}) \bmod 2 \leftarrow C_i \oplus k_{i \bmod m}$

- Vantagem: operação auto-inversa (ou-exclusivo).

One-Time Pad

- Utilização de chave inteiramente aleatória, com o mesmo comprimento da mensagem.

- *One-time pad*:
  - $C_i \leftarrow M_i \oplus k_i$
  - $M_i \leftarrow C_i \oplus k_i$

- Desvantagem: o problema da proteção da mensagem "reduz-se" ao da proteção de uma chave de igual tamanho.

Cifra de Fluxo

- Expansão de uma chave curta K para uma seqüência de bits k_i (*máscara*) do tamanho da mensagem.

- *Cifra de Fluxo*:
  - $k_i \leftarrow f(K, i)$ *// derivação de chave*
  - $C_i \leftarrow M_i \oplus k_i$ *// encriptação*
  - $M_i \leftarrow C_i \oplus k_i$ *// decriptação*

- Na prática, agrupam-se os bits em blocos (por exemplo, em bytes).

Cifra de Fluxo

- Transformação opera em mensagens de qualquer tamanho.
- O tamanho do texto cifrado é o mesmo do texto claro.

$$E\colon \{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^*$$

- A cifração de cada bit da mensagem altera o estado interno do algoritmo (memória).

RC4™

- Rivest Cipher #4.
- Algoritmo proprietário jamais divulgado.
- "Reconstituído" por engenharia reversa.
- Padrão em SSL.
- Chave de tamanho variável (múltiplo de 8 bits, tamanho máximo de 2048 bits).
- Pequeno número de operações simples, orientadas a processadores de 8 bits.

RC4™: Inicialização

for $i \leftarrow 0, \ldots, 255$ {
    $S[i] \leftarrow i$
}
$p \leftarrow 0$
for $i \leftarrow 0, \ldots, 255$ {
    $p \leftarrow (p + S[i] + K[i \bmod m]) \bmod 256$
    $S[i] \leftrightarrow S[p]$
}

RC4™: Operação

$i \leftarrow p \leftarrow 0$

for $t \leftarrow 0, \ldots, m-1$ {

$i \leftarrow t \mod 256$

$p \leftarrow (p + S[i]) \mod 256$

$z \leftarrow (S[i] + S[p]) \mod 256$

$S[i] \leftrightarrow S[p]$

$C_t \leftarrow M_t \oplus z$ (ou $M_t \leftarrow C_t \oplus z$)

}

Discussão

- Um sistema utiliza RC4™ para cifrar mensagens a partir de uma senha conhecida pelo remetente e pelo destinatário.

- A senha é registrada uma única vez (por exemplo, numa operação de cadastramento de usuário) e utilizada para cifrar todas as mensagens enviadas pelo remetente para o destinatário.

Discussão

- Que tipo de vulnerabilidade existe no sistema proposto?

- Como aproveitar essa vulnerabilidade para quebrar o sistema?

Vulnerabilidade do Sistema

- A segurança de *qualquer* cifra de fluxo baseia-se na hipótese de que a chave é utilizada *uma única vez*.

- Motivo: a diferença binária ($\oplus$) entre mensagens cifradas é igual à diferença binária entre as mensagens claras correspondentes.

Vulnerabilidade do Sistema

$$\left.\begin{aligned} C_i &= M_i \oplus k_i \\ C_i^* &= M_i^* \oplus k_i \end{aligned}\right\} \Rightarrow C_i \oplus C_i^* = M_i \oplus M_i^*$$

$\therefore$ Se alguma mensagem clara *M* for comprometida, *qualquer* outra mensagem *M** pode ser recuperada como $M^* = M \oplus C \oplus C^*$.

Conseqüências

- Embora este exemplo utilize o RC4™, *qualquer* cifra de fluxo sofreria o mesmo problema (inclusive *one-time pad*).

- A vulnerabilidade *não* está no algoritmo, mas em sua *utilização imprópria*.

Outras cifras de fluxo

- A5 (GSM $\rightarrow$ quebrável em tempo real).
- eStream: ECrypt Stream Cipher Project <www.ecrypt.eu.org/stream/>.
- Resultados recentes indicam ser extremamente difícil projetar uma cifra de fluxo segura sistematicamente ☹.

Cifras de Bloco

Cifra de bloco

- A transformação opera em mensagens de tamanho fixo (característico do algoritmo).

$$E\colon \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$$

- O estado interno do algoritmo não é afetado de uma cifração para outra.
- É possível converter uma cifra de bloco em uma cifra de fluxo.
- Normalmente é um algoritmo iterativo (várias aplicações sucessivas de uma transformação simples, dependente da chave).

Algoritmos Principais

- DES (Data Encryption Standard)
- 3DES (DES triplo)
- AES (Advanced Encryption Standard)

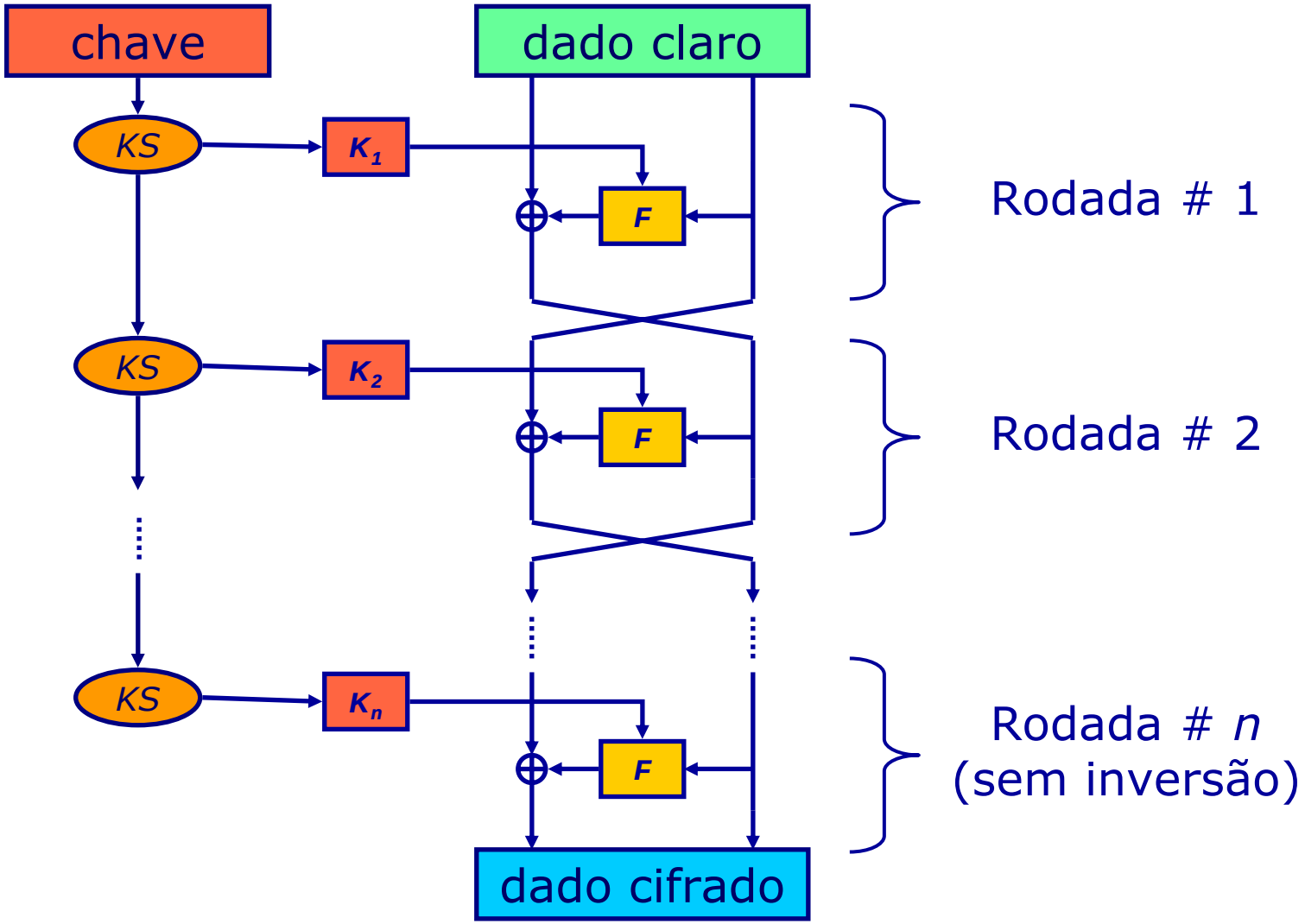
DES

- Concurso semi-público promovido em 1975 pelo governo americano.
- Algoritmo original proposto pela IBM, alterado pela NSA segundo critérios não divulgados na época.
- Estabelecido como padrão governamental em 1977 (primeiro padrão criptográfico com especificação pública).

DES

- Estrutura voltada a hardware, mas razoavelmente eficiente em software.
- Bloco de 64 bits, chave de 56 bits (*estrutura de Feistel*).
- Escalonamento de chaves linear (subchaves são simples subconjuntos de 48 bits dentre os 56 bits da chave).

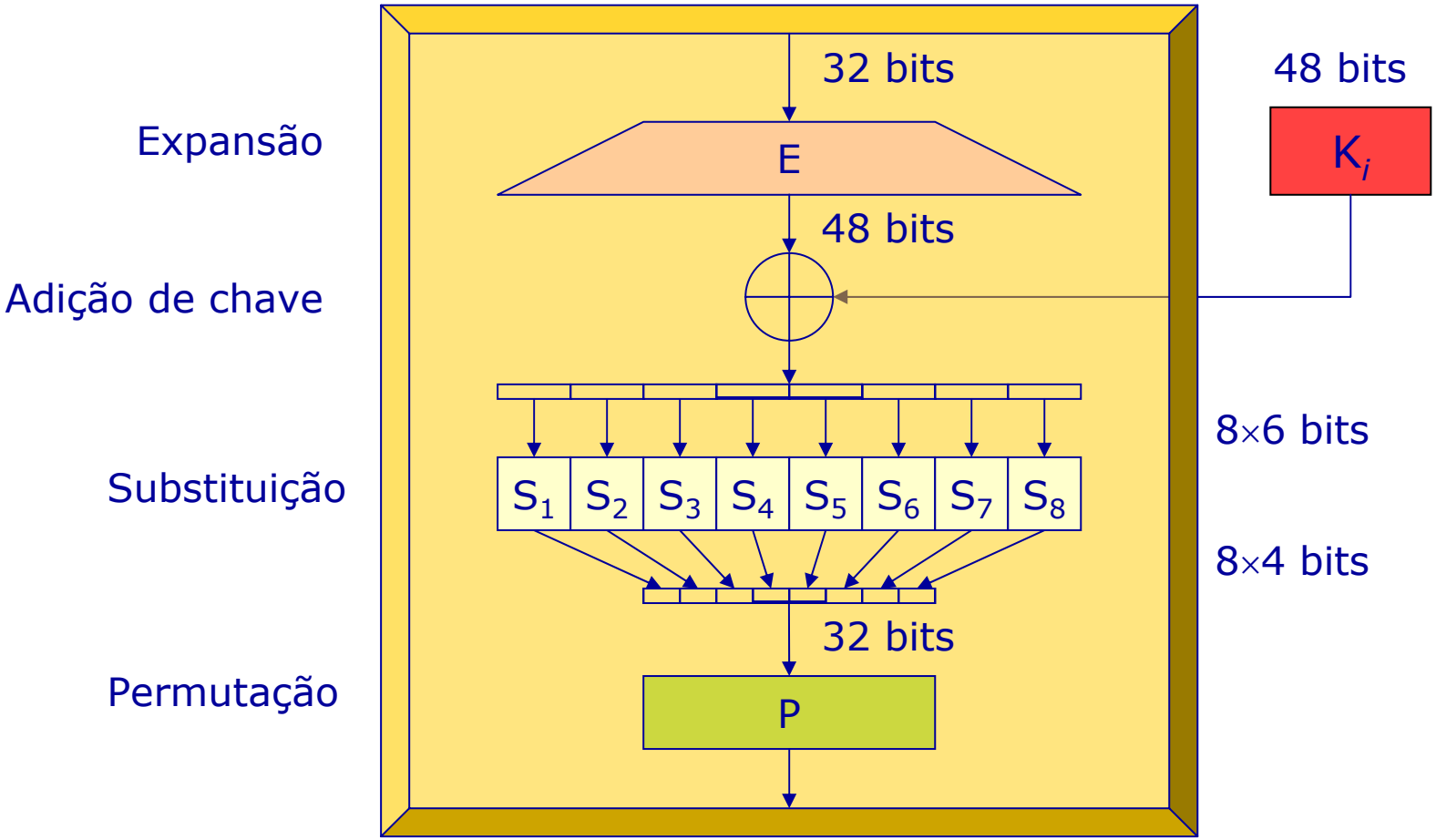
Estrutura de Feistel



DES - A Função *F*

- Expandir linearmente os 32 bits de entrada para 48 bits.
- Adicionar (XOR) a subchave K_i.
- Particionar o resultado em 8 grupos de 6 bits.
- Substituir o valor de cada por outro valor através de consulta a 8 tabelas distintas, cada uma mapeando 6 bits em 4 bits.
- Permutar os bits dos 8 valores obtidos das tabelas, produzindo a saída de 32 bits.

DES - A Função *F*



Obsolescência do DES

- Chave de 56 bits insuficiente para resistir à capacidade computacional disponível atualmente.
- Tamanho de bloco pode ser pequeno demais para evitar ataques futuros.
- Definitivamente aposentado pelo NIST em 2004.

3DES

- Paliativo para aproveitar a ampla base existente de implementações do DES em hardware e software.
- Construção tripla: cifrar cada bloco três vezes, com três chaves distintas K_1, K_2, K_3 de 56 bits.
- Variante popular: decifrar com K_2 (EDE).

$$M \xrightarrow{E_{K_1}} X \xrightarrow{E^{-1}_{K_2}} Y \xrightarrow{E_{K_3}} C$$

- Segurança equivalente a de uma cifra com chave de ≈112 bits (*não* é 168 bits!).

2DES ?

- Idéia: cifrar cada bloco de dados duas vezes, usando duas chaves distintas K_1 e K_2 de 56 bits.

$$M \xrightarrow{E_{K_1}} X \xrightarrow{E_{K_2}} C$$

- Funciona?

Falha da Construção Dupla

- Hipótese: conhecem-se uma mensagem clara *M* e a mensagem cifrada *C* correspondente.

- Quebrar o sistema por força bruta exige testar todas as 2^{56} chaves K_1 e, para cada uma delas, todas as 2^{56} chaves K_2, totalizando 2^{112} operações.

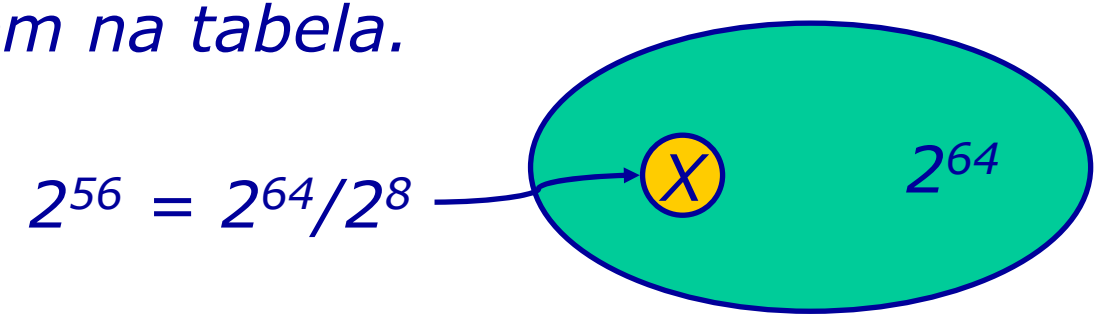
- *Ataque "meet-in-the-middle": muito mais rápido.*

Meet-in-the-middle – 1ª fase

$$M \xrightarrow{E_{K_1}} X$$

- Cifra-se M com todas as 2^{56} chaves K_1 possíveis, obtendo uma tabela T com 2^{56} entradas (todos os valores X obtidos, associados às chaves K_1 que os produzem).

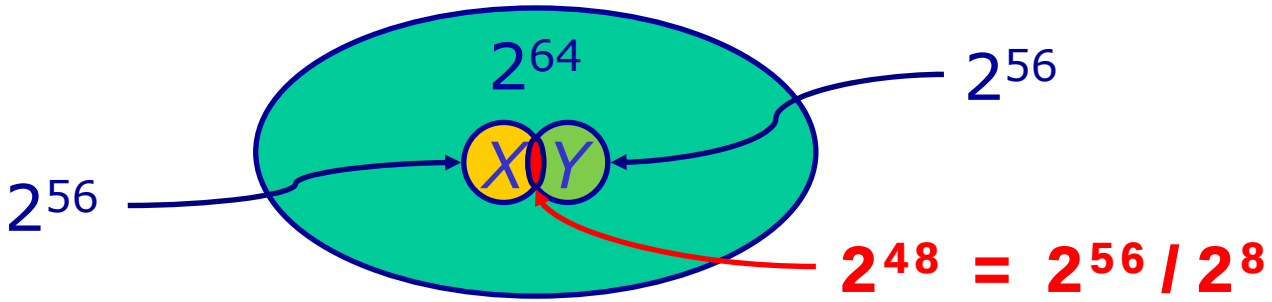
- *Apenas* $2^{56}/2^{64} = 1/2^8$ *de todos os valores de 64 bits ocorrem na tabela.*



Meet-in-the-middle – 2ª fase

$$Y \xleftarrow{E_{K_2}^{-1}} C$$

- Decifra-se C com todas as 2^{56} chaves K_2 possíveis.
- Hipótese de cifra forte: valores decifrados estão *uniformemente distribuídos* entre os 2^{64} valores possíveis.
- *Apenas uma fração* $\approx 1/2^8$ *dentre os* 2^{56} *(i.e.* 2^{48}*) valores obtidos da decifração de C ocorrem na tabela T previamente obtida.*



Meet-in-the-middle – 3ª fase

- Portanto, dos 2^{112} pares (K_1, K_2) sobram apenas 2^{48} pares capazes de transformar M em C, e para cada K_1 resta uma única chave K_2 possível.

- Escolhe-se um segundo par conhecido (M', C'), e testam-se todos os 2^{48} pares (K_1, K_2) que sobraram na fase anterior.

$$M' \xrightarrow{E_{K_1}} X \xrightarrow{E_{K_2}} C'$$

- O par correto (K_1, K_2) é isolado com probabilidade 99,99847% (i.e. a probabilidade de colisão é apenas $2^{48}/2^{64}$).

Esforço do ataque

- Esforço da 1ª fase: 2^{56} operações.
- Esforço da 2ª fase: 2^{56} operações.
- Esforço da 3ª fase: 2^{48} operações.
- Esforço total: $2^{56} + 2^{56} + 2^{48} \sim 2^{57}$ operações.
- Ganho de segurança sobre a cifra simples: 1 bit!

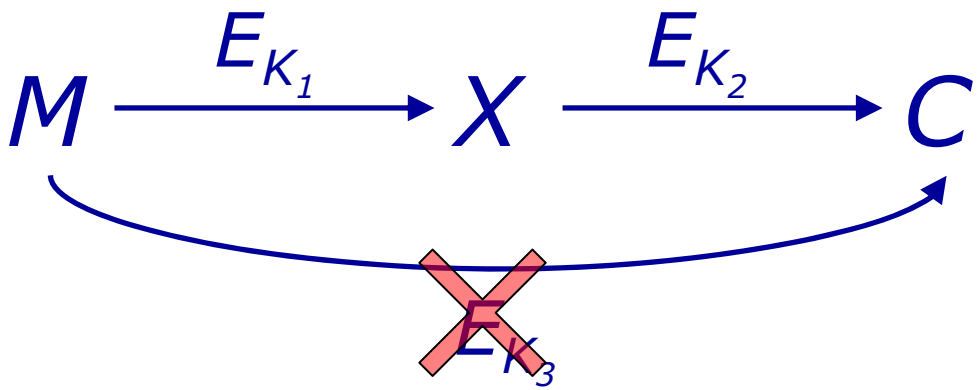
Exercício

- O ataque descrito contra o 2DES baseia-se no fato de que o tamanho da chave é menor que o tamanho do bloco (as encriptações de um bloco fixo sob todas as chaves possíveis não esgotam os blocos cifrados possíveis).

- *Como atacar a construção dupla com um algoritmo em que o tamanho da chave seja igual ou maior que o tamanho do bloco?*

Exercício

- A segurança das construções múltiplas baseia-se implicitamente na hipótese de que a cifra *não* é transitiva.



- O que aconteceria se a cifra *fosse* transitiva?