Esquemas simétricos de cifra

Notas para a UC de “Segurança Informática”

Inverno de 11/12

Pedro Félix (pedrofelix *em* cc.isel.ipl.pt)

Instituto Superior de Engenharia de Lisboa

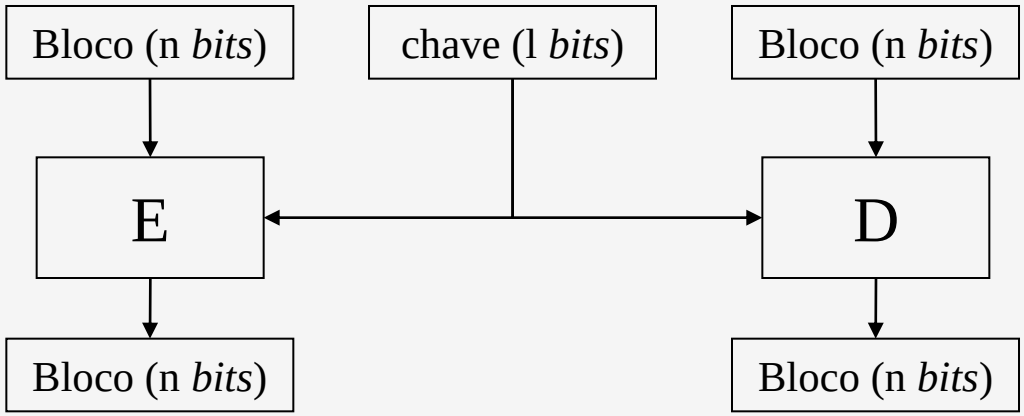
Sumário

- Primitivas de cifra em bloco
- Primitivas iteradas
- Cifra múltipla
- Modos de operação
- Formas de *padding*

Primitivas de cifra em bloco

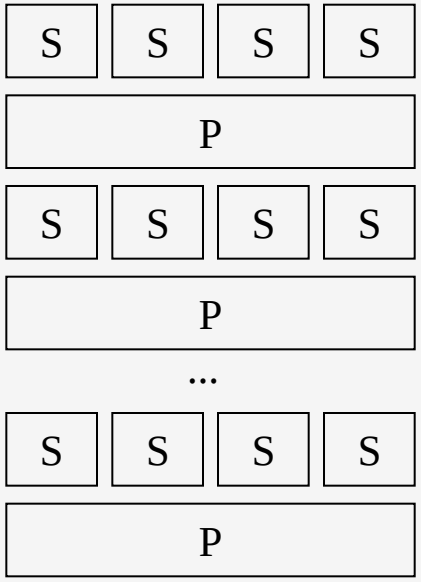
- Primitiva de cifra em bloco
  - Função **E: $\{0,1\}^l \rightarrow \{0,1\}^n \rightarrow \{0,1\}^n$**

    tal que $\forall$ **k** $\in$ **$\{0,1\}^l$** a função **E(k)** é uma permutação
  - Designa-se por **D: $\{0,1\}^l \rightarrow \{0,1\}^n \rightarrow \{0,1\}^n$** a função que verifica $\forall$ **k** $\in$ **$\{0,1\}^l$** e $\forall$ **m** $\in$ **$\{0,1\}^n$: D(k)(E(k)(m)) = m**
- A dimensão do bloco é **n** (ex. 64 *bit*, 128 *bit*)
- A dimensão da chaves é **l** (ex. 56 *bit*, 128 *bit*, 256 *bit*)



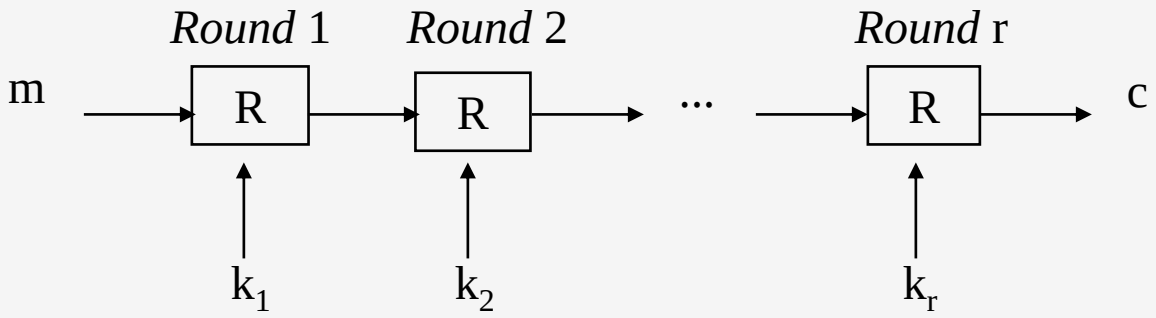
Notas

- A dimensão **n** do bloco deve ser suficientemente elevada para impossibilitar ataques baseados na estatística do texto em claro
- A dimensão da chave **l** deve ser suficientemente elevada para impossibilitar ataques de pesquisa exaustiva
- Elementos construtores
  - Substituições
  - Transposições
- Redes SP (*Substitution-Permutation*)



Primitivas iteradas

- Dada uma função **R: $\{0,1\}^n \rightarrow \{0,1\}^n$**, pode ser criado um sistema de cifra por composições sucessivas desta função:
  - **$E(k) = (R(k_r)$ o $R(k_{r-1})$ o … o $R(k_1))$**
- Um sistema assim obtido diz-se *iterado*. A função **R** é designada por função de *round* e cada aplicação da função constitui um *round*
- Para cada *round* é utilizada uma *sub-chave* $\mathbf{k_i}$ derivada da chave **k** fornecida ao sistema
- A obtenção das sub-chaves é designada por escalonamento de chaves (*key scheduling*)



Cifra múltipla: cifra dupla

- Resolver a baixa dimensão das chaves na primitiva DES (l = 56)
- Primeira solução:
  - Cifrar um bloco usando uma chave e cifrar o resultado com outra chave:

    c = $\mathbf{E(k_2)(E(k_1)(m))}$

    m = $\mathbf{D(k_1)(D(k_2)(c))}$
- Se a primitiva constitui um grupo em relação à composição, então existe uma chave $\mathbf{k_3}$ tal que

  $\mathbf{c = E(k_2)(E(k_1)(m)) = E(k_3)(m)}$

  qualquer que seja **m**

- Prova-se que a primitiva DES não constitui um grupo
- Espaço de chaves 2^{2n}, o que implica uma pesquisa exaustiva com 2^{2n} operações. No entanto, um ataque *meet-in-the-middle* reduz o número de chaves a testar para 2^{n+1}.

Ataque *meet-in-the-middle*

- Neste ataque, o adversário tem dois pares (m_1, c_1) e (m_2, c_2) tal que

  $c_1 = E(k_2)E(k_1)(m_1))$

  $c_2 = E(k_2)E(k_1)\ (m_2))$
- Para todo o **k** possível, o adversário calcula $E(k)(m_1)$ e guarda o resultado
- Para todo o **k** possível, o adversário calcula $D(k)(c_1)$ e compara o resultado com os resultado do ponto anterior. Se coincidir com algum, é possível que o par de chaves obtido seja o par (k_1, k_2) (k_1 - chave usada na cifra, k_2 – chave usada na decifra)
- Se se verificar que $c_2 = E(k'_2)(E(k'_1)(m_2))$, onde (k'_2, k'_1) é o par de chaves encontrado no passo anterior, então, com bastante probabilidade, o par obtido é o par (k_1, k_2) procurado
- A probabilidade pode ser aumentada verificando com outros pares claro-cifra (m_i, c_i)
- O ataque utiliza o máximo de $2.2^n = 2^{n+1}$ cifras.

Cifra múltipla: cifra tripla

- Outra solução: (E-D-E)
  - Operar três vezes sobre um bloco usando duas chaves:
    $c = E(k_1)(D(k_2)(E(k_1)(m)))$
- As chaves k_1 e k_2 alternam para evitar um ataque *meet-in-the-middle*.

- Outra solução:
  - Operar três vezes sobre um bloco usando três chaves:
    $c = E(k_3)(D(k_2)(E(k_1)(m)))$

- Norma: FIPS 46-3 e ANSI X9.52

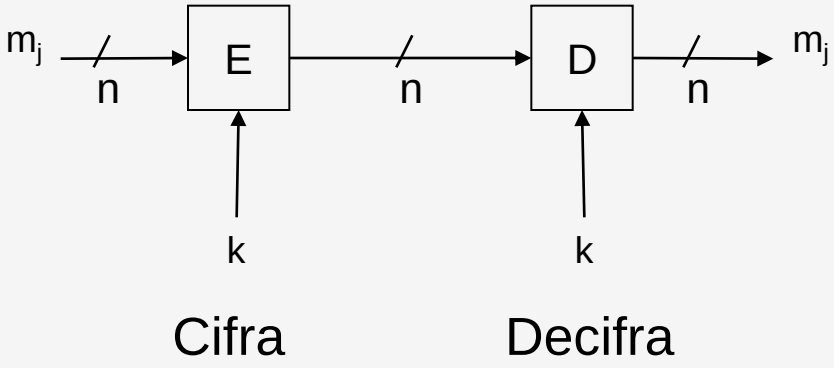
- Porquê EDE e não EEE? Se **$k_1 = k_2$** o modo triplo transforma-se no modo simples.

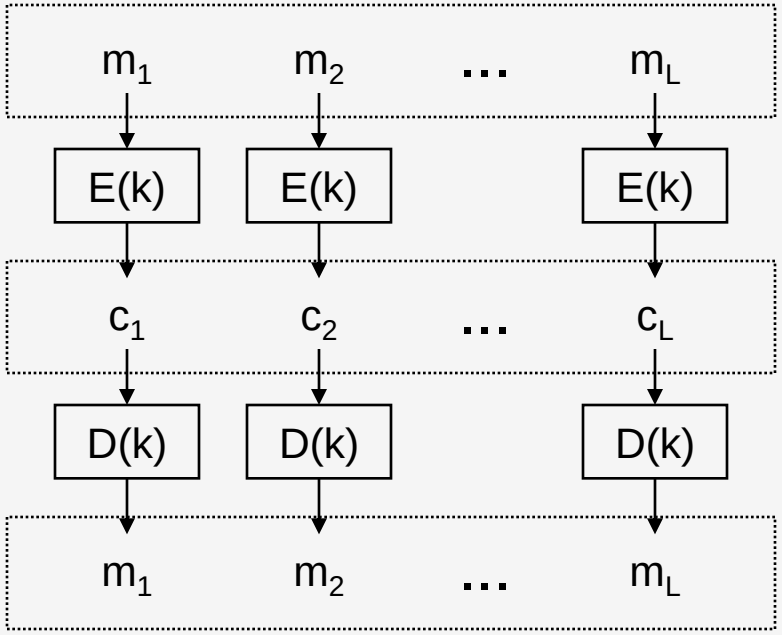
Modos de operação

- Problema: Como efectuar a cifra de mensagens com dimensão superior à de um bloco?

- Considerações:
  - Padrões no texto em claro não deverão ser evidentes no texto cifrado
  - A eficiência do método usado não deverá ser muito inferior à eficiência da primitiva de cifra em bloco usada
  - A dimensão do texto cifrado deve ser aproximadamente igual à dimensão do texto em claro
  - Em algumas aplicações é importante que a decifra seja capaz de recuperar de erros, adições e remoções de *bits* ocorridos no texto cifrado
  - Acesso aleatório – capacidade de decifrar e alterar apenas parte do criptograma

Modo *Electronic-CodeBook* (ECB)

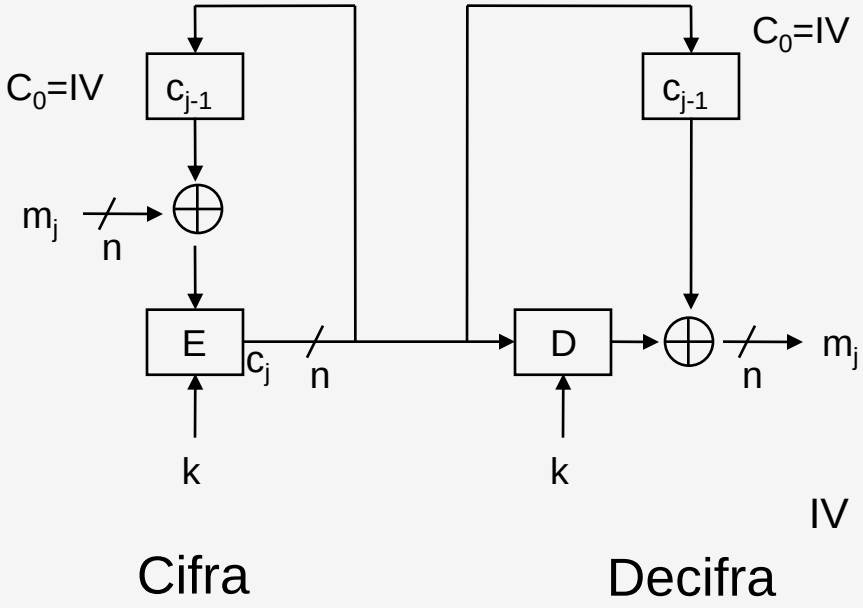


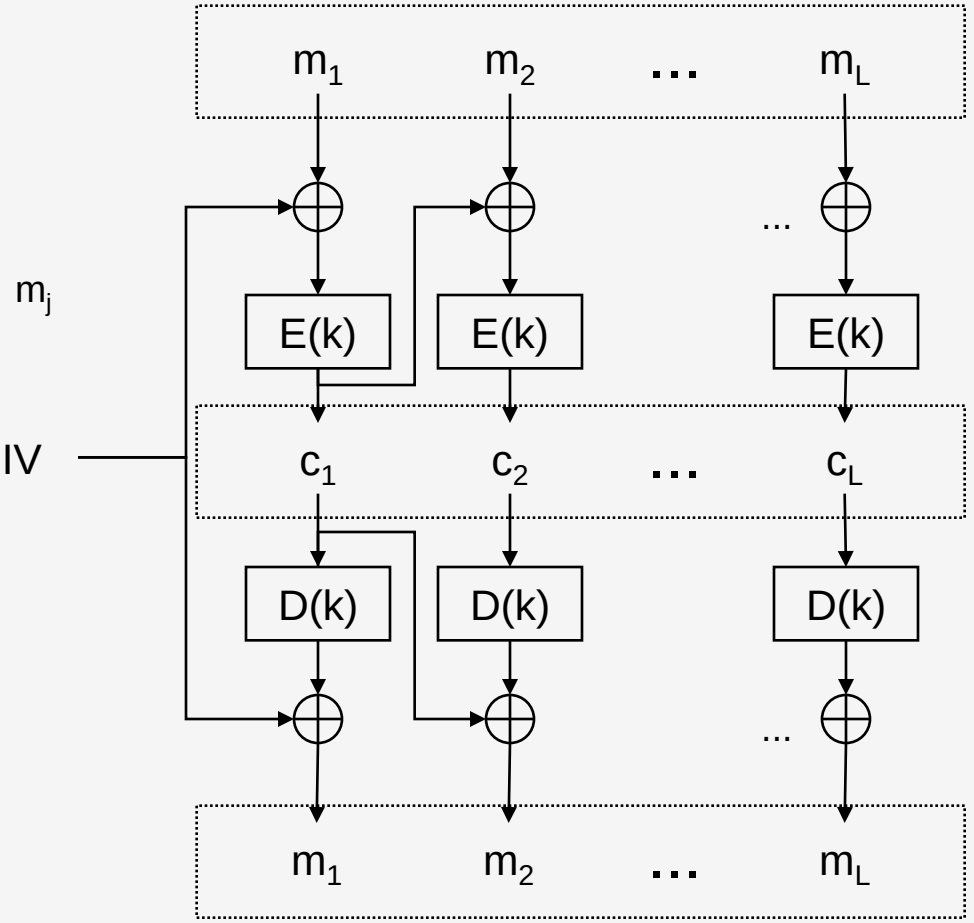


Modo *electronic-codebook* (ECB)

- Blocos de texto em claro iguais:
  - Blocos de texto em claro iguais, cifrados com a mesma chave, implicam blocos de texto cifrado iguais
- Interdependência na cifra:
  - A cifra é realizada de forma independente de bloco para bloco
- Propagação de erros:
  - A ocorrência de erros num bloco de texto cifrado afecta apenas a decifra desse bloco
- Acesso aleatório:
  - Permite acesso aleatório para decifra e “recifra” de múltiplos de blocos.

Modo *cipher block chaining* (CBC)



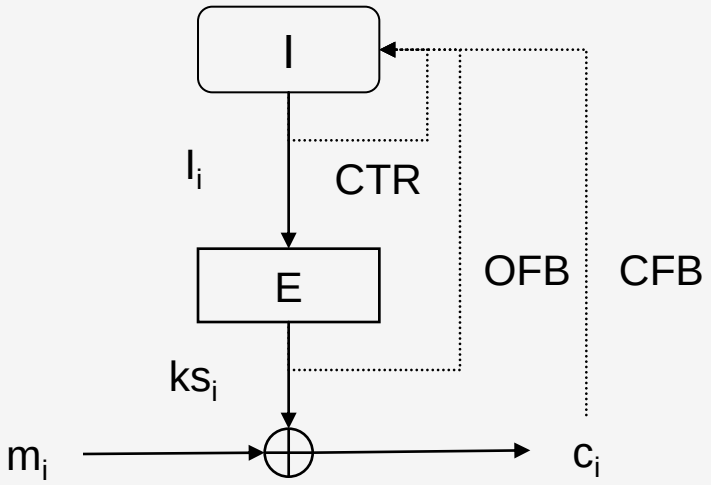


Modo *cipher block chaining* (CBC)

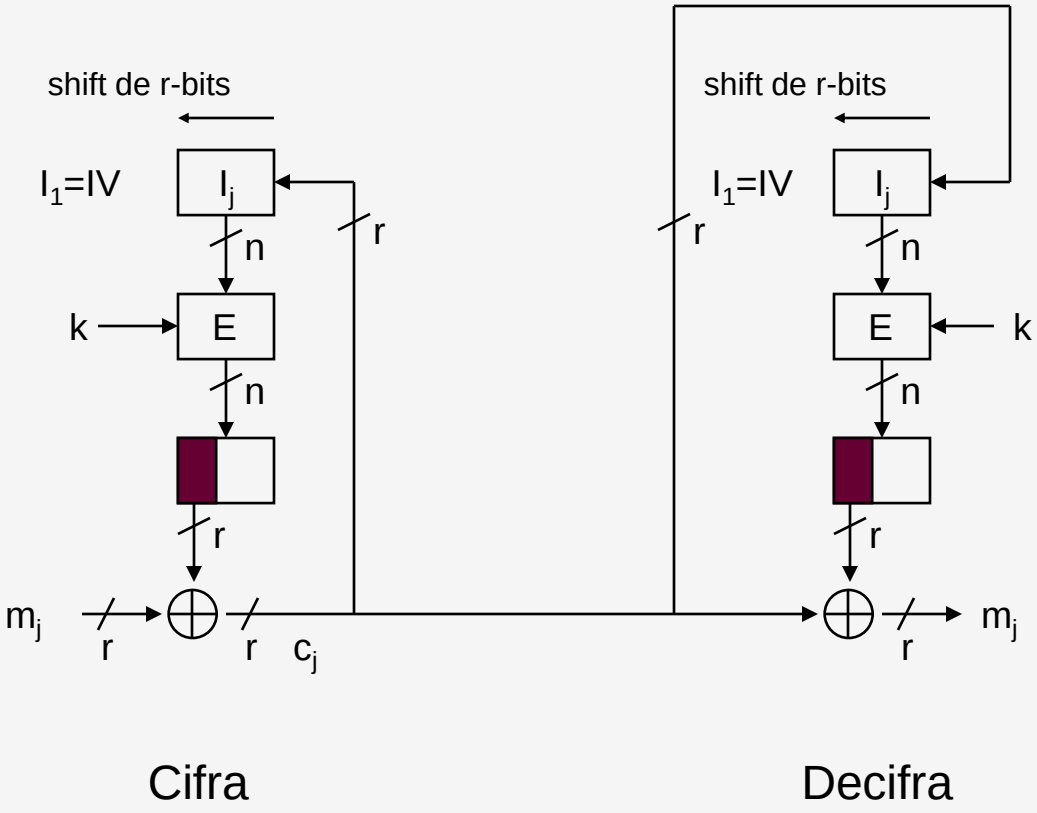
- Blocos de texto em claro iguais:
  - Sob a mesma chave e sob o mesmo vector de iniciação, duas mensagens iguais implicam criptogramas iguais
- Interdependência na cifra:
  - A cifra de um bloco de texto em claro afecta a cifra dos blocos seguintes
- Propagação e recuperação de erros:
  - A ocorrência de erros num bloco $\mathbf{c}_j$ de texto cifrado afecta a decifra do próprio bloco e a do bloco seguinte $\mathbf{c}_{j+1}$. A decifra do bloco $\mathbf{c}_{j+1}$ terá erros nas mesmas posições que $\mathbf{c}_j$
- Observações:
  - A reordenação dos blocos de texto cifrado afecta a decifra
  - É relativamente fácil manipular um determinado bloco de texto em claro

Modos de operação em *stream*

- Modo *Stream*
  - Estado **I**
  - *Key stream* **ks**
  - $\mathbf{ks_i = E(k)(I_i)}$
  - $\mathbf{c_i = m_i \oplus ks_i}$
- *Cipher FeedBack* (CFB)
  - $\mathbf{I_i = c_{i-1}}$
- *Output FeedBack* (OFB)
  - $\mathbf{I_i = ks_{i-1}}$
- *Counter* (CTR)
  - $\mathbf{I_i = f(I_{i-1})}$
- Problema:
  - se $\mathbf{ks_i = ks_j}$ então $\mathbf{m_i \oplus m_j = c_i \oplus c_j}$



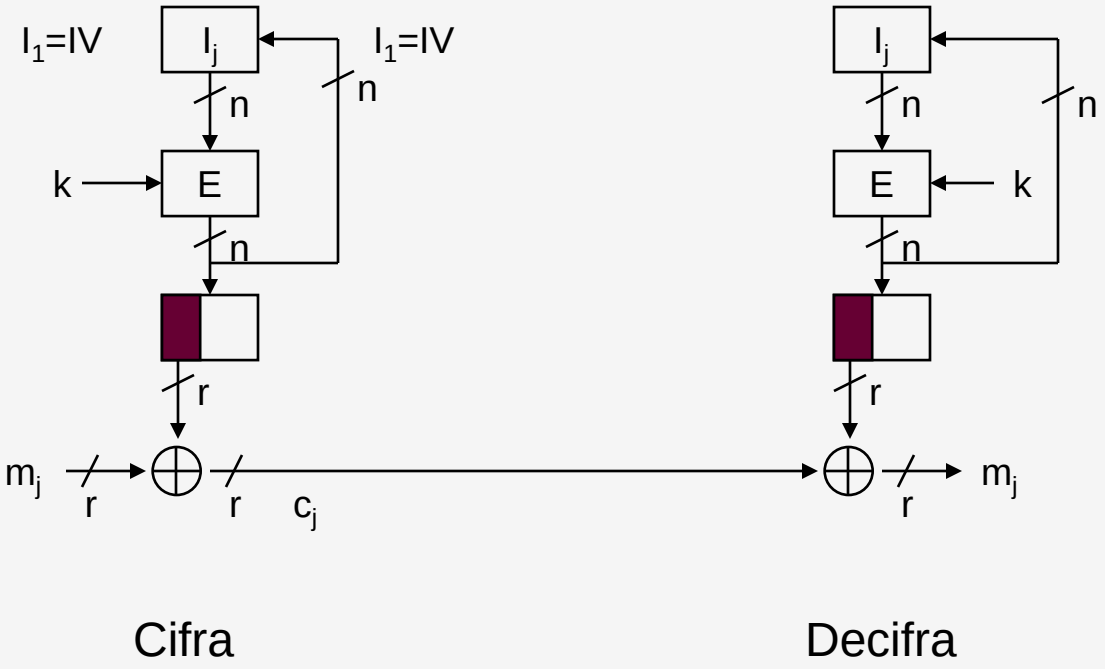
Modo *Cipher feedback* (CFB)



Modo *Cipher feedback* (CFB)

- Blocos de texto em claro iguais:
  - Sob a mesma chave e sob o mesmo vector de iniciação, duas mensagens iguais implicam criptogramas iguais
- Interdependência na cifra:
  - A cifra de um bloco de texto em claro afecta a cifra dos blocos seguintes
- Propagação e recuperação de erros:
  - A ocorrência de erros num bloco $\mathbf{c}_j$ de texto cifrado afecta a decifra do próprio bloco e a dos **n/r** blocos seguintes. O bloco $\mathbf{m}_j$ resultante da decifra do bloco $\mathbf{c}_j$ terá erros nas mesmas posições que $\mathbf{c}_j$
- Observações:
  - A reordenação dos blocos de texto cifrado afecta a decifra
  - É relativamente fácil manipular um determinado bloco de texto em claro

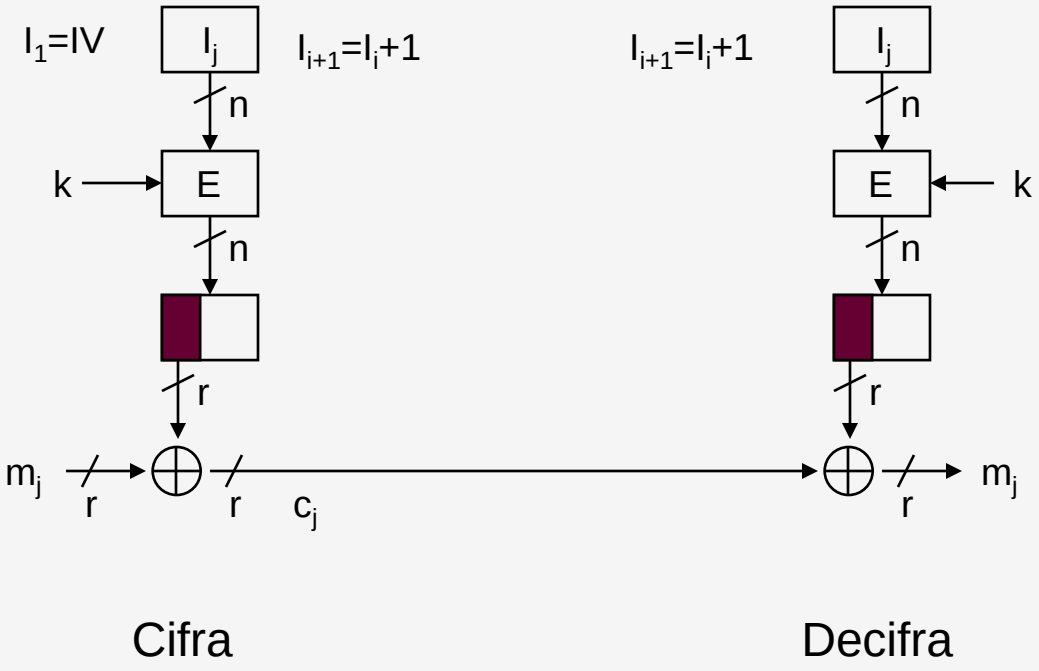
Modo *Output feedback* (OFB)



Modo *Output feedback* (OFB)

- Blocos de texto em claro iguais:
  - Sob a mesma chave e sob o mesmo vector de iniciação, duas mensagens iguais implicam criptogramas iguais
- Propagação e recuperação de erros:
  - A ocorrência de erros num bloco de texto cifrado $\mathbf{c}_j$ afecta apenas a decifra desse bloco. O bloco $\mathbf{m}_j$ resultante da decifra do bloco $\mathbf{c}_j$ terá erros nas mesmas posições que $\mathbf{c}_j$
- Observações:
  - É relativamente fácil manipular um determinado bloco de texto em claro

Modo *Counter* (CTR)



Modo *Counter* (CTR)

- Blocos de texto em claro iguais:
  - Sob a mesma chave e sob o mesmo vector de iniciação, duas mensagens iguais implicam criptogramas iguais
- Propagação e recuperação de erros:
  - A ocorrência de erros num bloco de texto cifrado $\mathbf{c}_j$ afecta apenas a decifra desse bloco. O bloco $\mathbf{m}_j$ resultante da decifra do bloco $\mathbf{c}_j$ terá erros nas mesmas posições que $\mathbf{c}_j$
- Acesso aleatório:
  - Permite acesso aleatório para decifra e “recifra” de *bits*
- Observações:
  - É relativamente fácil manipular um determinado bloco de texto em claro

Vectores iniciais

- Nunca repetir o IV
  - CBC – problema do ECB no primeiro bloco
  - Modos CFB, OFB e CTR – repetição do *key stream*
- Os IV não têm de ser confidenciais
- Geração do IV
  - Contador
    - Problema: pequena distância de *hamming*
  - Previsível
    - Problema: ataques activos (ex. SSH)
  - Aleatório
    - Prefixar à mensagem
  - Baseado em *nounce* (*number used once*)
    - CBC: **IV = E_k(*nounce*)**
    - CTR: **S = *nounce* | i**

Qual escolher

- CBC
  - Muito usado na prática: SSL, IPSEC, ...
  - Seguro no modelo CPA (*Chosen Plaintext Atack*)

- CTR
  - Pouco usado na prática: não faz parte dos modos de operação normalizados para utilização com o DES (FIPS PUB 81)
  - Seguro no modelo CPA
  - Operação em paralelo
  - Acesso aleatório
  - Não amplifica erros
  - Não necessita de *padding*
  - Modo de operação em *stream*

Padding

- Seja X o número de bytes a acrescentar para que a dimensão da mensagem seja múltipla da dimensão do bloco
- PKCS# 5 (CBC-PAD):
  - Acrescentar X *bytes* com o valor X
  - Utilizações PKCS# 7, CMS, SSL
- ESP-PAD:
  - Acrescentar X *bytes* com os valores ’01’ ’02’ ... X
  - Utilizações: IPSEC
- SSH-PAD:
  - Acrescentar X-1 bytes aleatórios seguido do byte com valor X

- A segurança do esquema depende da forma de *padding*?

- Ataque proposto por S. Vaudenay: *chosen ciphertext attack* utilizando o destinatário como oráculo que recebe criptogramas e retorna 1 ou 0 conforme o *padding* esteja correcto ou não.