Algoritmia Aplicada

Ano lectivo 2011-2012

Aula 4 – Processamento de cadeias de caracteres: Criptação

Sumário (1ª parte; última aula)

- Processamento de cadeias de caracteres
  - Compressão
    - Codificação segundo comprimentos de séries
      - Caso de ficheiros de texto
      - Caso de ficheiros binários
    - Codificação segundo comprimentos variáveis (método de Huffman)

Sumário (2ª parte)

- Processamento de cadeias de caracteres
  - Criptação
    - Introdução. Conceitos fundamentais
    - Cifra de *César*
    - Tabela de substituição
    - Cifra de *Vigenere*
    - Cifra de *Vernam*
    - Métodos de permutação
    - Método "RSA"

Processamento de cadeias de caracteres – Compressão vs. Criptação

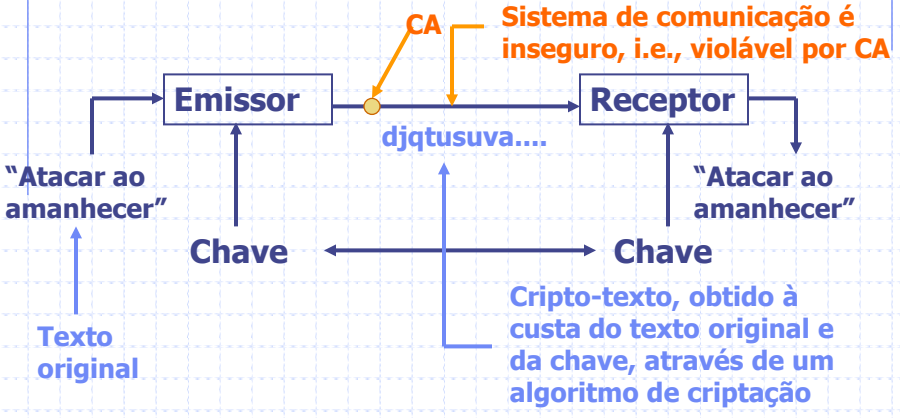
- **Compressão**
  - A codificação de cadeias de caracteres tem por objectivo **economizar espaço**
- **Criptação**
  - A codificação de cadeias de caracteres tem por objectivo **mantê-los secretos**

Processamento de cadeias de caracteres – Criptação

- **Definições**
  - **Criptação:** codificação da informação por forma a mantê-la secreta
  - **Criptografia:** Desenvolvimento de técnicas de criptação
  - **Cripto-análise:** Trabalho associado à tarefa de tentar descodificar a informação codificada (secreta)
  - **Cripto-sistema:** conjunto de elementos que assegura um meio secreto de comunicação entre duas entidades

Processamento de cadeias de caracteres – Criptação

- **Estrutura de um cripto-sistema típico**



CA (Cripto-analista) conhecerá o cripto-texto e o método de criptação, desconhecendo apenas a chave!

Processamento de cadeias de caracteres – Criptação

- **Aplicações**
  - Sistemas de comunicações militares e diplomáticas
  - Transferência de fundos entre bancos
  - Manutenção de ficheiros secretos por parte dos utilizadores,...

Processamento de cadeias de caracteres – Criptação

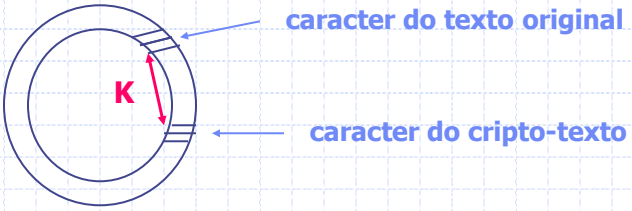
- **Métodos**
  - Métodos simples de criptação
  - Método mais complexo, mas extremamente seguro – Método RSA

Criptação – Cifra de César

- Cada caracter do texto original é substituído pelo caracter que aparece ***k*** posições mais à frente no alfabeto, sendo **k** um número inteiro fixo

  (César usava k=3)

- Para o efeito, o alfabeto é visto como se estivesse organizado em **círculo**, de modo a que o 1º caracter sucede ao último caracter:



Criptação – Cifra de César

- **Exemplo:**

K = 1

Texto original: **E □ F A C I L**

Cripto-texto: **F A G B D J M**

Nota: O espaço é o último caracter

- **Método fraco,** visto que o cripto-analista terá de analisar um **reduzido nº de casos**
- Por exemplo, se tivermos 27 caracteres (A, B, ..., Z, espaço), teremos apenas **26 hipóteses** diferentes,

1 <= k <= 26

pois o caso **k=27** corresponderia a uma mensagem criptada igual à mensagem original!

Criptação – Tabela de substituição

- Cada caracter do texto original é substituído por um outro caracter de acordo com uma **tabela de substituição**

- Exemplo de uma tabela de substituição:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
T	H	E		Q	U	I	C	K	B	R	O	W	N	F	X	J	M	P	D	V	R	L	A	Z	Y	G

Então teremos para o exemplo:

Texto original: E □ F A C I L

Cripto-texto: U T I H □ B W

Criptação – Tabela de substituição

- Este processo **é muito mais seguro**
  O cripto-analista terá de tentar **27! tabelas** para tentar ler a mensagem.
  (Note-se que: $27! > 10^{28}$)

- Note-se que havendo 27 caracteres, haverá **27 maneiras diferentes** de codificar cada caracter
  (são 27 e não 26, porque se admite que um caracter possa ser substituído por ele próprio, como acontece com o "Y" na tabela anterior)

Criptação – Tabela de substituição

- O elevadíssimo nº de hipóteses é **enganador**, no que se refere à segurança do método.

  É relativamente fácil de descobrir a tabela de substituição tendo em conta aspectos como sejam:

  - **Frequência das letras inerente a qualquer linguagem**

    Por exemplo, o *E* é a letra mais frequente em Inglês, pelo que o cripto-analista assumirá, à partida, e possivelmente com sucesso, que a letra mais frequente no cripto-texto substituirá o E!

  - **Combinação de letras**

    Por exemplo, em Inglês, certas combinações de 2 letras (como o QJ) nunca ocorrem, enquanto que outras (como ER) são muito comuns

Criptação – Cifra de Vigenere

- É uma **extensão** da **Cifra de César**. Em vez de se considerar um mesmo deslocamento (K) para todos os caracteres do texto original, teremos **diferentes deslocamentos**
- Isto consegue-se considerando uma **pequena chave**, em que cada caracter é usado para determinar o valor de k a usar em cada momento
- A **chave** será **repetida** as vezes necessárias para cobrir todos os caracteres do texto original

Criptação – Cifra de Vigenere

- **Exemplo:**

Texto original: E □ M U I T O □ F A C I L

Chave: A B D

K = 1, 2, 4

Codificação: E □ M U I T O □ F A C I L

A B D A B D A B D A B D A

Cripto-texto: F B Q V K X P B J B E M M

Criptação – Cifra de Vigenere

- **Curiosidades:**
  - Códigos diferentes para o mesmo caracter
    Por exemplo, o I aparece codificado como K e M
  - Códigos iguais para caracteres diferentes
    Por exemplo, o I e o L aparecem ambos codificados como M

Codificação: E □ M U I T O □ F A C I L

A B D A B D A B D A B D A

Cripto-texto: F B Q V K X P B J B E M M

- **Segurança do método:**
  - tanto maior, quanto mais comprida for a chave

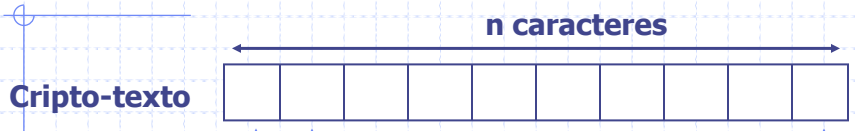
Criptação – Cifra de Vernam

- É uma **generalização** do método **Cifra de Vigenere**, em que a **chave** é tão comprida quanto o **texto** a criptar.
- Método **totalmente seguro,** visto que cada **caracter** da **chave** é usado apenas **uma vez**.

Assim, o cripto-analista terá de tentar, para cada posição do cripto-texto, todos os caracteres do alfabeto!

Isto corresponde a obter todas as mensagens possíveis com o comprimento do texto original!

Criptação – Cifra de Vernam



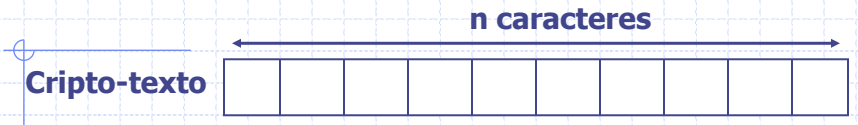
Nesta posição do texto original poderia estar um qualquer caracter do alfabeto (A, ..., Z, espaço)

Idem .... Idem

Raciocínio do cripto-analista

Há 27^n hipóteses diferentes!

Criptação – Cifra de Vernam



Raciocínio do cripto-analista

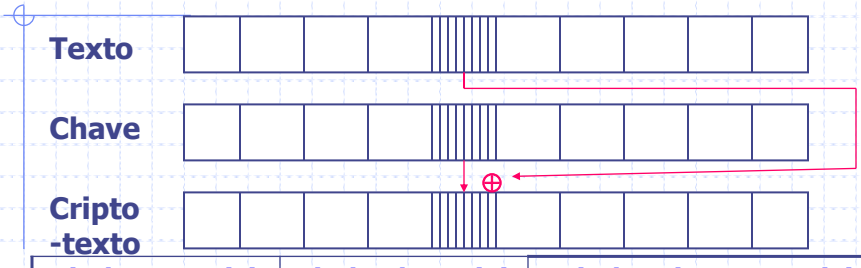
Há 27^n hipóteses diferentes!

Nota: Se o alfabeto tivesse 2 caracteres apenas (α, β) e fosse n=3, teríamos 2^3 =8 hipóteses para a mensagem transmitida

$\alpha\alpha\alpha$	$\alpha\alpha\beta$	$\alpha\beta\alpha$	$\alpha\beta\beta$
$\beta\alpha\alpha$	$\beta\alpha\beta$	$\beta\beta\alpha$	$\beta\beta\beta$

- **Embora o método seja totalmente seguro há o problema da transmissão da chave, pelo que o método é apenas usado para mensagens curtas (⇒ chaves curtas)**

Criptação – Variante para textos e chave em binário



Bit do Texto (T)	Bit da Chave (C)	Bit do Cripto-texto (Z)
0	0	0
0	1	1
1	0	1
1	1	0
		$Z = T \oplus C$

Criptação – Variante para textos e chave em binário

Bit do Texto (T)	Bit da Chave (C)	Bit do Cripto-texto (Z)
0	0	0
0	1	1
1	0	1
1	1	0
		$Z = T \oplus C$

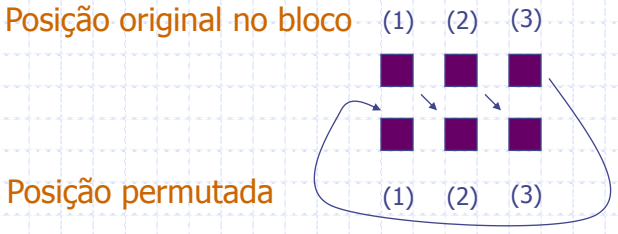
$T \oplus Z = C$ **(1)**

$Z \oplus C = T$ **(2)** ⟶

A operação de **descriptação** (obtenção do texto original à custa do cripto-texto e da chave) é idêntica à de criptação. Fazer o **ou exclusivo** de cada bit do cripto-texto com cada bit da chave!

Métodos de Permutação

- **Nestes métodos, os caracteres do texto original são rearranjados, obtendo-se dessa forma o cripto-texto**
- **Por exemplo, podemos dividir o texto original em sucessivos blocos de 3 caracteres e, em cada bloco, permutar os caracteres do modo seguinte:**



Métodos de Permutação

- **Exemplo:**

  Texto original: E FACIL

  Texto original artificialmente aumentado com dois caracteres (Y, Z): | **E F** | **ACI** | **LYZ** |
  seleccionados aleatoriamente

  Cripto-texto: FE IACZLY

- **Para descodificar o cripto-texto, o receptor terá de conhecer a chave de permutação e a sua inversa. No exemplo considerado:**

  1 2 3 ← **posição original**

  2 3 1 ← **posição permutada**

Método "RSA"

- **O nome é devido aos autores que em 1978 publicaram um artigo com o método:**

  ***"A Method for obtaining digital signatures and public key cryptosystems"*, R.L. Rivest, A. Shamir, L. Adelman, Communications of the ACM 21,2**

- **A grande importância deste método advém do facto de ser (praticamente) impossível de quebrar o código do cripto-texto e de não necessitar de chaves de grande dimensão (como no caso da Cifra de Vernam*)**

  ***problema de transmissão da chave**

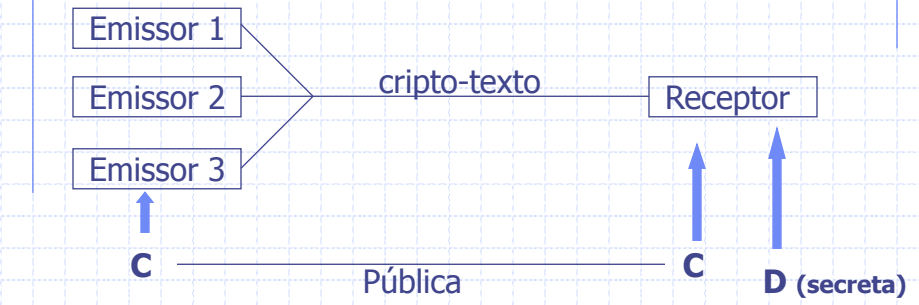
Método "RSA"

- **Propriedades gerais do método (comuns a outros métodos da mesma família - *"Public Key Encryption Schemes")***
  1. **São usadas duas chaves:**
     - **uma chave de criptação do texto original**
     - **uma chave de descodificação do cripto-texto**
  2. **Para cada chave de criptação, há exactamente uma chave de descodificação correspondente, diferente da chave de criptação**
  3. **Há muitos pares (chave de criptação, chave de descodificação respectiva)**

Método "RSA"

4. **A chave de criptação é tornada pública (não é secreta), pelo receptor, a todos aqueles que serão emissores de mensagens para ele**
5. **No entanto, a chave de descodificação é apenas conhecida pelo receptor (é secreta)**
6. **É quase impossível determinar a chave de descodificação se só se conhece a chave de criptação. Assim sendo, só mesmo, o receptor pode descodificar o cripto-texto, visto ser o único que conhece a chave de descodificação:**

Método "RSA"



C = chave de criptação
D = chave de descodificação

Método "RSA"

Os Emissores 1 e 2 não conseguem descodificar o cripto-texto relativo a uma mensagem enviada pelo Emissor 3, mesmo tendo conhecimento de C. Falta-lhes D!

Idem para mensagens enviadas pelo Emissor 1 ou pelo Emissor 2

Por maioria de razão, será praticamente impossível, a qualquer cripto-analista, descodificar qualquer cripto-texto com destino ao Receptor!

Método "RSA"

- **Criação das chaves C e D**

  a) **Gerar três números primos(1) "aleatórios" com cerca de 100 algarismos cada um (2):**

  x – nº primo
  y – nº primo
  d – nº primo, sendo o maior dos três

  (1) Nº primo é aquele que só é divisível por si próprio e por 1 (por exemplo, os nºs seguintes são primos: 2,3,5,7,11,13,17)

  (2) Os autores sugerem no mínimo 100 algarismos

Método "RSA"

b) **Calcular o número inteiro N (3), por:**

$$N = x \cdot y$$

(3) N terá cerca de 200 algarismos

c) **Determinar o número inteiro c através da equação**

$$(c \cdot d) \bmod ((x-1) \cdot (y-1)) = 1$$

d) **Prova-se (4) que, com N, d e c escolhidos desta maneira, se tem, para qualquer nº inteiro (M), menor do que N:**

$$\left(M^c \bmod N\right)^d \bmod N = M \quad \forall M \leq N-1$$

(4) A demonstração poderá encontrar-se em qualquer manual de Teoria dos Números

Método "RSA"

e) As chaves de criptação (C) e de descodificação (D) serão constituídas pelos pares de inteiros seguintes:

C = (N, c)
D = (N, d)

f) Um exemplo, em que os números são propositadamente pequenos (por isso, sem interesse prático!):

⇒ Números primos arbitrados:
x = 47
y = 59
d = 157

⇒ N = x.y = 47 * 59 = 2773

Método "RSA"

⇒ $(c \cdot d) \bmod ((x-1)\cdot(y-1)) = 1$

$(c \cdot 157) \bmod (46 * 58) = 1$

$(c \cdot 157) \bmod (2668) = 1$

$c * 157 = 2668$

$c = {}^{2668}\!/_{157} = 17$

⇒ Então teremos:

C = (N, c) = (2773, 17)
D = (N, d) = (2773, 157)

Método "RSA"

- **Porque é quase impossível determinar a chave D?**
  - **A chave C é pública, donde conhece-se N e c**
  - **Então para determinar D=(N, d), é necessário determinar d**
  - **Ora, de acordo com o que acabamos de ver, c e d relacionam-se através da equação**

$$(c \cdot d) \bmod ((x-1)\cdot(y-1)) = 1$$

que foi usada por quem criou as chaves!

  - **No entanto, o cripto-analista conhece N e sabe que há a seguinte relação N = x . y**

Método "RSA"

- **Então ele "só" terá de pegar no número N (com cerca de 200 algarismos!) e tentar decompô-lo no produto de dois factores primos!**
- **A tabela que se segue ilustra a dificuldade do problema (admite-se que cada operação demora 10^{-6}s):**

Nº de algarismos de N	Tempo
50	4h
75	104 dias
100	74 anos
200	4 biliões de anos
300	$5 * 10^{15}$ anos

!!!

Método "RSA"

- **Operação de criptação**
  - **Ilustremos a exposição que se vai seguir com um exemplo:**

N c

C = (2773, 17)

D = (2773, 157)

N d

} Chaves

Texto a criptar = IDESOFMARCH

Método "RSA"

1º Passo: Decompor o texto em blocos de caracteres. Converter cada bloco num nº inteiro entre 0 e N-1 (0…2772)

Para converter um bloco de caracteres num número inteiro, atribuir um código numérico a cada caracter e concatenar os códigos numéricos

No caso, consideremos os seguintes códigos para os vários caracteres do alfabeto:

A	00	G	06	M	12	S	18	Y	24
B	01	H	07	N	13	T	19	Z	25
C	02	I	08	O	14	U	20		
D	03	J	09	P	15	V	21		
E	04	K	10	Q	16	W	22		
F	05	L	11	R	17	X	23		

Método "RSA"

Por exemplo, o bloco ZU converte-se em 2520, podendo ser calculado assim:

$$25*10^2+20$$

O bloco ZITA converte-se em 25081900, podendo ser calculado assim:

$$25*10^6+8*10^4+19*10^2+0$$

Método "RSA"

1º Passo (continuação):

No nosso exemplo, uma vez que a gama é (0...2772), deveremos considerar blocos de 2 caracteres

ID	ES	OF	MA	RC	HX
0803	0418	1405	1200	1702	0723
(M_1)	(M_2)	(M_3)	(M_4)	(M_5)	(M_6)

caracter seleccionado aleatoriamente, por forma a que o último bloco fique com 2 caracteres

Este é o texto original sob forma numérica

Método "RSA"

2º Passo (criptação propriamente dita):

Transforma-se cada nº inteiro M_i (M_1.....M_6) num outro número inteiro T_i (T_1.....T_6) usando para o efeito a chave de criptação C (C=(N,c)):

$$T_i = M_i^c \bmod N$$

No nosso caso $T_i = M_i^{17} \bmod 2773$ **, ou seja,**

$$T_1 = M_1^{17} \bmod 2773 = 0803^{17} \bmod 2773 = 0779$$

Feitas as contas para os outros casos, teremos:

0779	1983	2641	1444	0052	0802
(T_1)	(T_2)	(T_3)	(T_4)	(T_5)	(T_6)

Este é o cripto-texto

Método "RSA"

- **Operação de descodificação do Cripto-texto**

  Usar a chave de descodificação D (D = (N, d)) para calcular:

  $$T_i^d \bmod N$$ **para todo o bloco i**

  Ora, prova-se que se obtém, com esta operação, o bloco original M_i, isto é

  $$M_i = T_i^d \bmod N$$

  Demonstração: Substituir T_i pela sua expressão de cálculo

  $$T_i^d \bmod N = \left(M_i^c \bmod N\right)^d \bmod N$$

Método "RSA"

Ora, referiu-se atrás que:

$$\left(M_i^c \bmod N\right)^d \bmod N = M_i \quad \text{c.q.d.}$$

No nosso caso $M_i = T_i^{157} \bmod 2773$**, ou seja,**

$$M_1 = T_1^{157} \bmod 2773 = 0779^{157} \bmod 2773 = 0803$$

Feitas as contas para os outros casos, teremos:

0803 0418 1405 1200 1702 0723

(M_1) (M_2) (M_3) (M_4) (M_5) (M_6)

Texto original sob forma numérica. A reconstrução do texto seria agora trivial à custa destes números!

Método "RSA"

➢ **Cálculo eficiente de T_i e de M_i**

- **Conforme vimos, na fase de criptação é necessário calcular:**

  $T_i = M_i^c \bmod N$; i= 1, 2, ...

  e na fase de descodificação

  $M_i = T_i^d \bmod N$; i=1, 2, ...

- **Sendo d e c números com cerca de 100 algarismos e N um número com cerca de 200 algarismos, é absolutamente necessário dispor de um algoritmo eficiente para aqueles cálculos**

Método "RSA"

Algoritmo para cálculo de $R = A^k \bmod N$, com A, K e N inteiros

```
Função R (A, K, N)
CALL Binario (K, B, T)
R ← 1
DO FOR I=T TO 0 STEP –1
    R ← (R*R) mod N
    IF B[I]=1
    THEN R ← (R*A) mod N
```

Binário é um procedimento que, dado K, devolve um vector B com a representação em binário do número, sendo T a última posição

Método "RSA"

Exemplo: Calcular $3^6 \bmod 500$

(O resultado é 229, visto que $3^6 = 729$)

Temos **$6 = 1\,1\,0_{(2)}$** então **B =**

0	1	1
0	**1**	**2**

Traçagem do algoritmo: e **T=2**

R ← 1 * 1 mod 500 = 1
R ← 1 * 3 mod 500 = 3

R ← 3 * 3 mod 500 = 9
R ← 9 * 3 mod 500 = 27

R ← 27 * 27 mod 500 = 229

Próxima aula

- Algoritmos Matemáticos
  - Polinómios