UFRJ - Universidade Federal do Rio de Janeiro

Rio de Janeiro, 04 de Março de 2011

NA AULA PASSADA

↳ CIFRA DE CÉSAR

↳ CHAVE: VALOR DE K

-> MÁQUINA ENIGMA

↳ CHAVE: POSIÇÃO INICIAL DAS 3 RODAS DENTRE AS 5

A Cifra de César e a Máquina Enigma são exemplos de criptografia de chave privada.

Temos dois modelos de criptografia

1) Criptografia de chave privada (ou chave simétrica)

↳ A mesma chave é utilizada para encriptar e decriptar mensagens. Esta chave deve ser mantida privada entre o remetente e o destinatário.

2) Criptografia de chave pública (ou chave assimétrica)

↳ Existem duas chaves. Uma é utilizada apenas para encriptação e a outra é utilizada apenas para decriptação. A chave de encriptação é pública. Qualquer um pode utilizá-la para enviar mensagens criptografadas para um dado destinatário. Só a chave de decriptação é privada, sendo conhecida apenas pelo destinatário

↳ Requisito de Segurança:

-> Para que um método deste modelo seja seguro, é necessário que o processo de calcular a chave privada a partir do valor conhecido da chave pública seja computacionalmente muito complexo e custoso.

↳ Exemplos de Criptografia de Chave Pública:

/ /

→ RSA (1979)

→ EL-GAMAL (1985)

- O RSA EM LINHAS MUITO GERAIS:

1) Construção das duas chaves:

→ Seleciono dois primos muito grandes (150 algarismos, cada) p e q

→ Calculo $n = p \cdot q$

2) Chave pública de encriptação: n

3) Chave privada de decriptação: par de primos (p, q)

Para calcular a chave pública a partir da chave privada → multiplicar dois números

↓

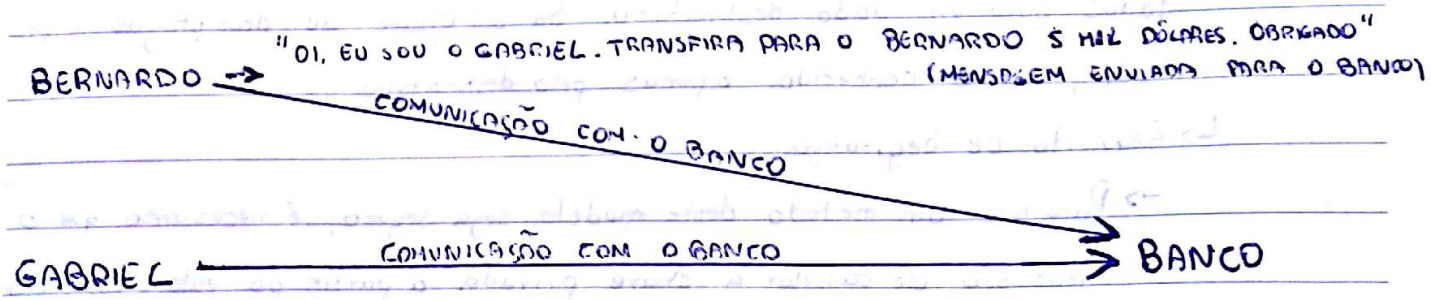
tarefa fácil computacionalmente

Para calcular a chave privada a partir da chave pública → fatoração de n

↓

tarefa computacionalmente muito custosa

OBS: É possível testar se um número é primo ou composto sem precisar fatorá-lo, Desta forma, a dificuldade da fatoração não impede a seleção inicial dos dois primos.



Só criptografia não resolve esse problema particular. Precisamos de alguma forma de certificação da origem da mensagem.

↳ Uma maneira de conseguir isso é utilizar um método de assinatura digital.

- ALGORITMOS

↳ Duas questões são fundamentais na análise de qualquer algoritmo:

1) O algoritmo é correto? (Ele produz a saída desejada?)

2) O algoritmo sempre termina?

- NOSSO PRIMEIRO ALGORITMO: ALGORITMO DA DIVISÃO (INTEIRO):

ENTRADA: Dois números inteiros positivos a e b

SAÍDA: O quociente q e o resto r da divisão de a por b.

a, b, q e r satisfazem as seguintes relações: $a = bq + r$

$0 \leq r < b$

- INSTRUÇÕES:

1) $R \leftarrow a$, $Q \leftarrow 0$

2) Enquanto $R \geq b$ faça:

2.1) $R \leftarrow R - b$

2.2) $Q \leftarrow Q + 1$

3) Retorne o valor em Q com o quociente e o valor em R como o resto.

Vamos mostrar que esse algoritmo sempre termina:

- Vamos listar os valores armazenados em R ao longo da execução:

$$\underbrace{a > a-b > a-2b > a-3b > \dots \geq b}$$

TODOS OS NÚMEROS NESSA SEQUÊNCIA SÃO INTEIROS

- Como entre dois números inteiros sempre existe uma quantidade finita de inteiros, não há como o algoritmo executar infinitamente o laço.

Vamos mostrar que esse algoritmo produz o resultado correto.

REPETIÇÕES	RESTO	QUOCIENTE
0	a	0
1	a-b	1
2	a-2b	2
⋮	⋮	⋮
K	a-Kb	K
⋮	⋮	⋮
q-1	a-(q-1)b	q-1
q	a-qb	q
FIM DO PROGRAMA		

Os valores retornados são: QUOCIENTE: q

RESTO: q-b -> r

Estes valores satisfazem a equação $a = qb + r$

Como o algoritmo só termina quando o valor da variável R é menor do que b, então o valor retornado r é menor do que b ($r < b$)

Por outro lado, na repetição anterior, como o algoritmo não terminou, o valor em R naquele momento era maior ou igual a b.

$a-(q-1)b \geq b$

$a-(qb-b)$

$a-qb+b \geq b$

$\underbrace{a-qb}_{r} \geq 0$

$r \geq 0$