RSA
(Stallings, Capítulo 8 e 9)

Cifras Simétricas vs Cifras Assimétricas

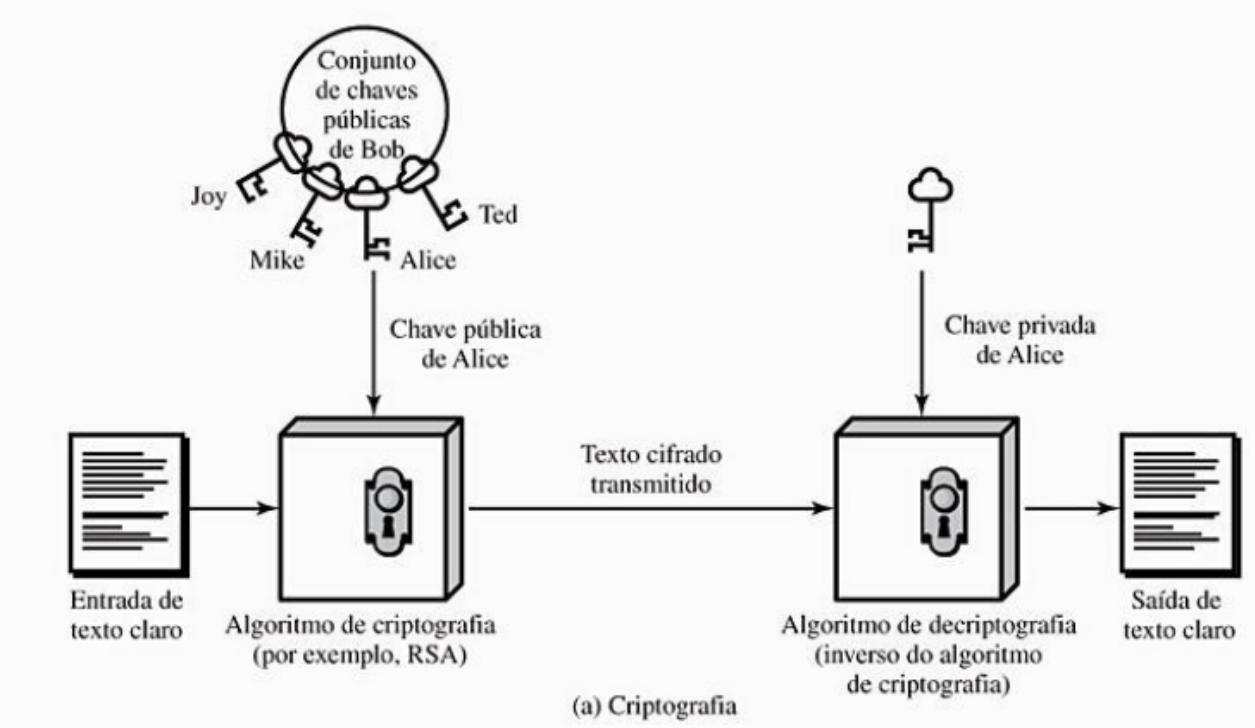
Cifras Simétricas: única chave utilizada na criptografia e decriptografia

$$C = E(M, K) \text{ e } M = D(C, K)$$

Cifras Assimétricas: chaves diferentes utilizadas na criptografia e decriptografia

$$C = E(M, K_E) \text{ e } M = D(C, K_D)$$

Cifras de Chave Pública



Cifras de Chave Pública

Etapas:

1. cada usuário gera par de chaves para criptografia e decriptografia das mensagens
2. cada usuário disponibiliza publicamente a chave pública K_{PU} (registro público, início da troca de mensagens, arquivo em comum, etc.)
3. cada usuário mantem a chave privada K_{PR} apenas de seu conhecimento
4. se Bob deseja enviar uma mensagem para Alice, Bob criptografa a mensagem com a chave pública de Alice
5. Alice decriptografa a mensagem usando sua chave privada
6. outros usuários não podem decriptografar a mensagem, apenas Alice

Aplicações de Chave Pública

Criptografia/decriptografia: emissor criptografa uma mensagem com a chave pública do destinatário

Assinatura Digital: o emissor assina uma mensagem com sua chave privada ao criptografar um texto conhecido pelo receptor

Troca de chave: os dois lados cooperam para trocar uma chave simétrica de sessão

Requisitos de Chave Pública

Definidos por Diffie e Hellman

1. ser computacionalmente fácil para uma parte B gerar um par (chave pública K_{PU_b}, chave privada K_{PR_b}).

2. ser computacionalmente fácil para um emissor A encriptar uma mensagem, isto é, computar

$$C = E(M, K_{PU_b})$$

3. ser computacionalmente fácil para o receptor B decriptografar um texto cifrado, isto é, computar

$$M = D(C, K_{PR_b}) = D(E(M, K_{PU_b}), K_{PR_b})$$

4. ser computacionalmente inviável para um adversário determinar a chave privada K_{PR_b} apenas conhecendo a chave pública K_{PU_b} e as funções $D(\cdot)$ e $E(\cdot)$

5. ser computacionalmente inviável para um adversário determinar o texto cifrado M apenas conhecendo a chave pública K_{PU_b} e as funções $D(\cdot)$ e $E(\cdot)$

6. as duas chaves podem ser aplicadas em qualquer ordem

$$M = D(E(M, K_{PU_b}), K_{PR_b}) = D(E(M, K_{PR_b}), K_{PU_b})$$

RSA - Rivest-Shamir-Adleman

Geração de Chaves

- Selecione p e q primos e $p \neq q$
- Calcule $n = p \times q$
- $\phi(n) = (p-1) \times (q-1)$, $\phi(n)$ é o de tociente de n
- Selecione o inteiro e, tal que $1 < e < \phi(n)$ e $MDC(\phi(n), e) = 1$, isto é, tociente de n e e são relativamente primos
- Calcule $d = e^{-1} \mod \phi(n)$
- Chave pública: $K_{PU} = \{e, n\}$
- Chave privada: $K_{PR} = \{d, n\}$

Propriedade

- se $M < n$, então $M^{ed} \mod n = M$

Criptografia

- Texto claro é um número $M < n$
- Texto cifrado é calculado por $C = M^e \mod n$

Decriptografia

- Texto cifrado é um número $C < n$
- Texto claro é obtido por $M = C^d \mod n$

Exercício

Escolha dois números primos pequenos p e q

Calcule $n = p \times q$

Calcule o tociente $\phi(n) = (p - 1) \times (q - 1)$

Escolha e tal que $1 < e < \phi(n)$ e $MDC(\phi(n), e) = 1$

Calcule $d = e^{-1} \mod \phi(n)$

Escolha um número (mensagem) aberto $M < n$

Calcule o número (mensagem) cifrado $C = M^e \mod n$

Teste se $M = C^d \mod n$

Prova - $M^{ed} \mod n = M$

Definição: a função tociente de Euler $\phi(n)$ é definida como o número de inteiros positivos menores que n e relativamente primos de n.

Propriedade 1: se p é primo, então $\phi(p) = p - 1$

Propriedade 2: se p e q são primos e $p \neq q$, então $\phi(pq) = (p-1)(q-1)$

Prova - $M^{ed} \mod n = M$

Teorema de Euler: seja a e n relativamente primos, então

$$a^{\phi(n)} \mod n = 1$$

Prova: considere o conjunto ordenado R dos inteiros positivos menores que n e relativamente primos de n e rotule-os da seguinte forma:

$$R = \{x_1, x_2, \ldots, x_{\phi(n)}\}.$$

Construa também o conjunto ordenado S da seguinte forma:

$$S = \{(ax_1 \mod n), (ax_2 \mod n), \ldots, (ax_{\phi(n)} \mod n)\}$$

Temos que S é uma permutação de R, pois:

1. $ax_i \mod n$ é relativamente primo de n e todos elementos em S são menor que n e relativamente primo de n.

2. Não existem duplicatas em S, pois se $ax_i \mod n = ax_j \mod n$, então $x_i = x_j$.

Então temos:

$$\begin{aligned}
\prod_{i=1}^{\phi(n)}(ax_i \mod n) &= \prod_{i=1}^{\phi(n)} x_i \\
(\prod_{i=1}^{\phi(n)} ax_i \mod n) \mod n &= \prod_{i=1}^{\phi(n)} x_i \mod n \\
a^{\phi(n)} \prod_{i=1}^{\phi(n)} x_i \mod n &= \prod_{i=1}^{\phi(n)} x_i \mod n \\
a^{\phi(n)} \mod n &= 1 \mod n
\end{aligned}$$

Corolário: seja a e n relativamente primos, então

$$a^x \mod n = a^{(x \mod \phi(n))} \mod n.$$

Prova - $M^{ed} \mod n = M$

Caso 1: M e n são relativamente primos. Temos:

$$\begin{aligned} M^{ed} \mod n &= M^{(ed \mod \phi(n))} \mod n \\ &= M^1 \mod n = M \end{aligned}$$

Caso 2: M e n não são relativamente primos, então $M \in \{0, 1p, 2p, \ldots, (q-1)p, 1q, 2q, \ldots, (p-1)q\}$. O caso $M = 0$ é trivial, considere o caso $M = xp$. Temos:

$$(M^{ed} - M) = (xp)^{ed} - (xp)$$
$$((xp)^{ed} - (xp)) \mod q =$$
$$((xp)^{(ed) \mod \phi(q)} \mod q - xp \mod q) \mod q =$$
$$((xp)^1 \mod q - xp \mod q) \mod q = 0$$
$$((xp)^{ed} - (xp)) \mod p = 0$$
$$((xp)^{ed} - (xp)) = kpq \Leftrightarrow (M^{ed} - M) \mod n = 0$$

Implementação

Requisitos:

- Ser possível encontrar valores de e, d, n tais que $M^{ed} \mod n = M$ para todo $M < n$
- Ser relativamente fácil calcular $M^e \mod n$ e $C^d \mod n$ para todo $M, C < n$
- Ser inviável determinar d dados e e n

Chaves:

- p, q são dois números primos (privados e escolhidos)
- $n = pq$ (público e calculado)
- e com $MDC(\phi(n), e) = 1$ e $1 < e < \phi(n)$ (público e escolhido)
- $d = e^{-1} \mod \phi(n)$ (privado e calculado)

Exercício

1. Considere que se deseje criptografar textos com 8 bits. Determine um par de chaves pública e privada para criptografar tais textos.

2. Considere que você intercepte a seguinte chave pública de um servidor $e = 77, n = 1829$ e uma mensagem encriptada $C = 56$ utilizando tal chave.

   (a) Determine a chave privada do servidor.

   (b) Determine o texto claro M.

Implementação - Exponenciação

Como calcular $a^b \mod n$?

Técnica Ingênua: multiplique a b vezes e depois tire o módulo $a \times a \times a \times \ldots \times a \times a \mod n$.
Problemas: valor pode ficar muito grande e é muito lento.

Técnica Ingênua com módulo: multiplique a b vezes, mas tire o módulo após cada multiplicação $(a \times a \mod n \times a \mod n \times \ldots \times a \mod n \times a \mod n)$.
Problema: é muito lento.

Exponenciação com Representação Binária: considere o exemplo $x^{11} \mod n$

1. pode-se representar $x^{11} \mod n = x^{8+2+1} \mod n = ((x^8 \mod n) \times (x^2 \mod n) \times (x^1 \mod n)) \mod n$

2. pode-se obter x^{2^n} realizando n multiplicações: $x^{2^0} = x^1 = x$ e $x^{2^n} = x^{2^{n-1}} \times x^{2^{n-1}} \mod n$

3. se expressarmos b como um número binário $b_k b_{k-1} \ldots b_0$, temos:

$$b = \sum_{i=1}^{k} b_i 2^i \qquad a^b = a^{\sum_{i=1}^{k} b_i 2^i} = \prod_{i=1}^{k} a^{b_i 2^i}$$

$$a^b \mod n = \left[\prod_{i=1}^{k} a^{b_i 2^i}\right] \mod n = \left(\prod_{i=1}^{k} \left[a^{b_i 2^i} \mod n\right]\right) \mod n$$

Implementação - Exponenciação

```
EXPONENCIACAO(a,b,n)
```

1. $c \leftarrow 0$
2. $f \leftarrow 1$
3. $k \leftarrow \lfloor \log(b) \rfloor$
4. for $i=k:-1:0$
5. $\quad c \leftarrow 2 \times c$
6. $\quad f \leftarrow f \times f \bmod n$
7. $\quad$ if $b_i = 1$
8. $\quad\quad c \leftarrow c + 1$
9. $\quad\quad f \leftarrow f \times a \bmod n$

Implementação - Números Primos

Para escolher as chaves, precisamos escolher números primos p e q

Solução Ingênua: dado um candidato ímpar a número primo n, testa para todo $1 < a < \sqrt{n}$ se $n \mod a = 0$

Solução Miller-Rabin: considera propriedade de números primos para realizar o teste de primalidade e executa o teste várias vezes.
Resultado 1: a probabilidade de que um número não primo passe no teste é **menor que** $\frac{1}{4}$
Resultado 2: em média $\frac{\ln N}{2}$ testes são necessários para encontrar um primo próximo a N

Implementação - Números Primos

Propriedades (Miller-Rabin):

1. Qualquer inteiro positivo ímpar $n \geqslant 3$ pode ser expresso da seguinte forma:

$$n - 1 = 2^k q, \text{ com } k > 0, \ q \text{ impar}$$

2. Se n é primo e $0 < a < n$, então $a^2 \mod n = 1$ se e somente se:

$$a \mod n = 1 \quad \text{ou} \quad a \mod n = n - 1$$

3. Se n é primo e $1 < a < n - 1$, então uma das duas condições é verdadeira:

(a) $a^q \mod n = 1$

(b) um dos números $a^q, a^{2q}, a^{4q}, \ldots, a^{2^{k-1}q}$ tem módulo em n igual a $n - 1$

Implementação - Números Primos

`TEST-PRIMO(n,k,q)`

1. selecione um inteiro aleatório `a`, tal que `1<a<n-1`
2. if a^q mod `n=1` então return `inconclusivo`
3. for `j=0` até `k-1` faça
4. if $a^{2^j q}$ mod `n=n-1` então return `inconclusivo`
5. return `composto`

Implementação - Relativamente Primos

Solução: escolha um número aleatório e utilize o algoritmo de Euclides para testar se é relativamente primo.

Resultado: a probabilidade de que dois números aleatórios sejam relativamente primos é 0,6.

Experiência 5

Implemente a criptografia e decriptografia RSA.

1. considere que o número $n = p \times q$ pode ter 32 bits (use representação unsigned int com 64 bits, no octave basta usar uint64(4) para criar o valor 4 do tipo inteiro sem sinal e 64 bits)
2. use a técnica de exponenciação apresentada em aula
3. implemente o algoritmo de Miller-Rabbin e avalie empiricamente quantos testes em média são necessários para encontrar um número primo (avalie com relação ao tamanho do número primo)
4. utilizando o algoritmo de Euclides estendido, avalie empiricamente a probabilidade de que dois números aleatórios

sejam relativamente primos (novamente, considere o tamanho dos números)

Os resultados deverão ser apresentados em conjunto com os resultados das Experiências 6

Data: 22/07/2014 (impresso e TIDIA)