Segurança Computacional

Criptografia assimétrica

Prof. Carlos Maziero

DInf UFPR, Curitiba PR

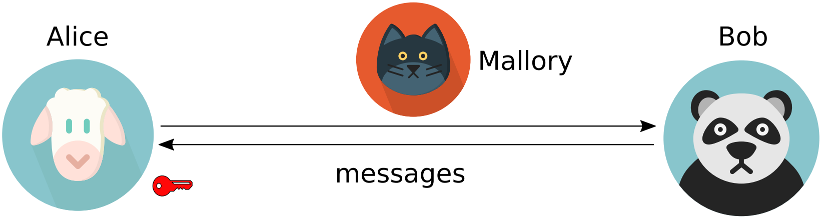
Julho de 2019

Conteúdo

Acordo de chaves

O problema do acordo de chaves

- Alice e Bob querem trocar mensagens via rede
- As mensagens serão cifradas com um algoritmo simétrico
- Eles precisam definir uma **chave comum**
- Mallory pode capturar as mensagens da rede
- Como definir a chave comum **através da rede**?



Acordo de chave de Diffie-Hellman-Merkle

- Permite estabelecer uma chave secreta comum
- Primeiro algoritmo de acordo de chaves (1976)
- Pode ser usado com canais inseguros (*sniffing*)
- Baseado em aritmética inteira modular
- Define os conceitos de chaves **pública** e **privada**

Troca de chaves de Diffie-Hellman-Merkle

Sejam p um número primo e g uma raiz primitiva módulo p

Raiz primitiva módulo p

Inteiro positivo $g < p$ tal que todo número n coprimo de p é congruente a uma potência de g módulo p.

Números coprimos

Dois números p e q são coprimos (ou primos entre si) sse o único inteiro que divide ambos é 1. Exemplos: 14 e 15, ou 6 e 35.

Troca de chaves de Diffie-Hellman-Merkle

passo	Alice	rede (Mallory vê)	Bob
1	escolhe p e g	$\xrightarrow{(p,g)}$	recebe p e g
2	escolhe a		escolhe b
3	$A = g^a mod\ p$		$B = g^b mod\ p$
4	envia A	$\xrightarrow{A}$	recebe A
5	recebe B	$\xleftarrow{B}$	envia B
6	$k = B^a mod\ p$		$k = A^b mod\ p$
	$k = g^{ba} mod\ p$		$k = g^{ab} mod\ p$
7	$m' = \{m\}_k$	$\xrightarrow{m'}$	$m = \{m'\}_k^{-1}$

Troca de chaves de Diffie-Hellman-Merkle

Que informação Mallory possui?

- O número primo p
- O número gerador g
- $A = g^a \mod p$ (*chave pública* de Alice)
- $B = g^b \mod p$ (*chave pública* de Bob)

Para obter k ela precisa calcular a ou b

- $A = g^a \mod p$, ou seja, $a = \log_g A \mod p$
- $B = g^b \mod p$, ou seja, $b = \log_g B \mod p$

Problema do logaritmo discreto

Calcular $A = g^a mod\ p$ é fácil e rápido!

mas...

Calcular $a = \log_g A \mod p$ é **muito difícil!**

Problema do logaritmo discreto:

- complexidade exponencial (sem solução eficiente)
- impraticável se o primo p for muito grande
- Em sistemas reais usa-se p com 1.024 bits ou mais

Algoritmos assimétricos

Algoritmos assimétricos

Usam um par de chaves complementares:

- Uma *chave pública kp*: conhecida por todos os usuários
- Uma *chave privada kv*: conhecida só pelo proprietário
- O que *kp* cifrar, *kv* decifra, e vice-versa (nem sempre!)

Estas chaves estão fortemente relacionadas:

- para cada *kp* há uma única *kv* correspondente
- para cada *kv* há uma única *kp* correspondente
- Não é possível calcular uma chave a partir da outra

Exemplos de chaves

Chave pública (SSHv2, algoritmo RSA):


```
ssh-rsa AAAAB3GzaC1yz2EAAAABIwCAAQEAq9iq5glqnwm4kQGUJ0EE7VoNBlNL
t7BJyUVJSC0j7E+JJYDDmkdwgTkgEF0CWkeBeQ3M1abgZthog1AIeDf9hfL2WPy6
XAfPZ2FtDze53qr/5akVfzLYvKj4c+ewVGYL+2Cjw9fqCpuVDzmG+0dRqfxk2rY2
jLTi79x8GWMpOWQMlrOiEpElopTB9VLxPCrh4SePnGWI+0/YS8T3m7K702dHXell
FQSFasNga7n9RtVIEHjjgSPV8iv3rVomxuOemKMXpUbsW56UrQAsMQ0G3KF4/Rfl
ACoHzM+Ib2JaN5sTBO0g4ImPjU5xjKl1lFkvAuM76U
vBImNjnClvNJa5BQ== maziero@localhost
```


Chave privada (SSHv2, algoritmo RSA):


```
-----BEGIN RSA PRIVATE KEY-----
MIIEoQIBAAKCHGFAq9iq5glqnwm5kQGUJ0EE7werBlWLt7BJyUVJSC0j7E+JJYGD
mkdwgTkgEF0CKkeBeQ3M1xbgZthog1AIeDf9mkjuWPy6XAfPZ2FtDze53qr/5aiV
... (22 linhas omitidas)
D4Bxn3bX9CUASkJicmD6S91sj+10HHZN+2bLGhbcYaAMPljGbA==
-----END RSA PRIVATE KEY-----
```

Algoritmos assimétricos

Para um usuário u com chave pública kp e privada kv:

$$\{\, \{\, x \,\}_{kp} \,\}_k^{-1} = x \iff k = kv$$

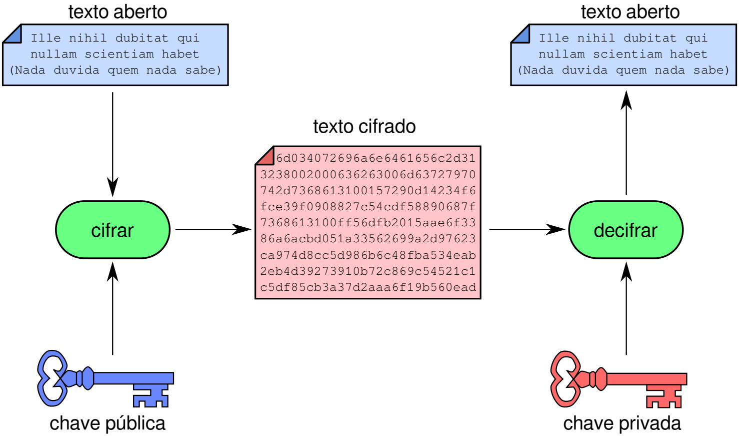
$$\{\, \{\, x \,\}_{kv} \,\}_k^{-1} = x \iff k = kp$$

ou

$$x \xrightarrow{kp} x' \xrightarrow{kv} x$$

$$x \xrightarrow{kv} x' \xrightarrow{kp} x$$

Algoritmos assimétricos



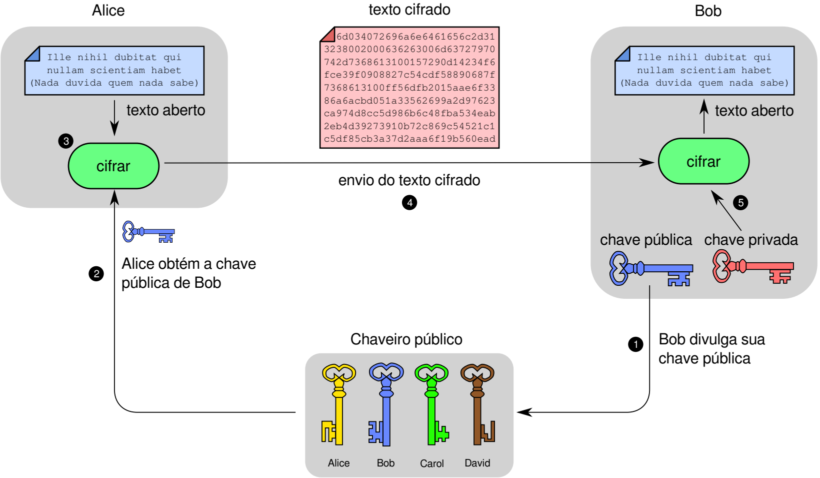
Exemplo de uso: comunicação confidencial

Alice quer enviar um documento secreto a Bob

1. Bob deixa sua chave pública em um local público
2. Alice busca a chave pública de Bob no repositório
3. Alice usa a chave pública de Bob para cifrar o documento
4. Alice envia o documento cifrado a Bob
5. Bob usa sua chave privada para decifrar o documento

Outros usuários lêem o texto cifrado mas não podem decifrá-lo

Algoritmos assimétricos



Algoritmos assimétricos

Podem ser usados para verificar a **autoria** de um documento:

- As chaves públicas estão publicamente acessíveis
- se Alice cifrar um documento com sua chave privada, qualquer um poderá decifrá-lo
- Se o documento puder ser decifrado com a chave pública de Alice, então ela é a autora do mesmo
- Este mecanismo é usado para criar **assinaturas digitais**

Quadro comparativo

Cifrador	Simétrico	Assimétrico
Chaves	Uma única chave para cifrar e decifrar	Chaves complementares para cifrar e decifrar
Tamanho das chaves	Pequena (AES: 64 a 256 bits)	Grande (RSA: 2.048 a 15.360 bits)
Tamanho dos dados	Qualquer (blocos ou fluxo)	Blocos com o tamanho da chave
Velocidade	Alta (centenas de MB/s)	Baixa (centenas de KB/s)
Uso	Grandes volumes de dados (tráfego de rede, arquivos, áudio, etc)	Pequenos volumes de dados (troca de chaves, assinaturas digitais)
Exemplos	RC4, A/51, DES, 3DES, AES	Diffie-Hellman, RSA, ElGamal, ECC

Criptografia híbrida

Vantagens da criptografia simétrica:

- é rápida e flexível
- pode cifrar grandes volumes de dados

Vantagens da criptografia assimétrica:

- permite realizar o acordo de chaves
- permite verificar autenticidade

Uso conjunto de ambas:

- Usar algoritmo assimétrico para definir chave de sessão
- Usado no SSL e TLS (SSH, HTTPS, etc)

Criptografia híbrida

Passo	Alice	rede insegura	Bob
1	$k = random()$		
2	$k' = \{k\}_{kp(Bob)}$		
3	$k'' = \{k'\}_{kv(Alice)}$		
4	envia k''	k''	recebe k''
5			$k' = \{k''\}^{-1}_{kp(Alice)}$
6			$k = \{k'\}^{-1}_{kv(Bob)}$
7	$m' = \{m\}_k$		
8	envia m'	m'	recebe m'
9			$m = \{m'\}^{-1}_k$

O algoritmo RSA

RSA

RSA - Rivest, Shamir & Adleman (MIT), 1977

É o cifrador assimétrico mais utilizado hoje

Problema: fatoração do produto de números primos:

- Calcular $p \times q$ é fácil:

$$1.300.511 \times 67.883.743 = 88.283.554.492.673$$

- Obter os fatores primos de um produto é **muito difícil**:

$$104.741.680.862.209.687 = p \times q$$

RSA - Conceitos básicos

Conjunto de inteiros módulo p:

$$\mathbb{Z}_p = \{0, 1, 2, \ldots, p-1\}$$

Aritmética modular:

$$15 \bmod 4 = 3$$

$$x \bmod 4 = 3 \Rightarrow x \equiv 3, 7, 11, 15, \ldots$$

RSA - Geração das chaves (1)

- Sortear dois números primos p e q
  - Números de magnitude similar
  - Técnica: sortear aleatórios e testar **primalidade**
  - Teste rápido: Miller-Rabin (probabilístico)
- Calcular o **módulo**: $n = p \times q$
- Calcular **lambda**: $\lambda(n) = mmc\,(p-1, q-1)$
  - λ: totiente de Carmichael
  - $mmc\,(a, b)$: mínimo múltiplo comum entre a e b

RSA - Geração das chaves (2)

- Calcular expoente e da chave pública
  - $1 < e < \lambda(n)$ e $mdc\ (e, \lambda(n)) = 1$
  - $mdc\ (a, b)$: máximo divisor comum
  - e e $\lambda(n)$ são **coprimos**

- Calcular expoente d da chave privada
  - $d \equiv e^{-1} \bmod \lambda(n)$
  - d é o **inverso multiplicativo** de e
    (ou seja, $d \times e = 1 \bmod \lambda(n)$)
  - calculado usando o Algoritmo de Euclides estendido

- Chave pública: $\{e, n\}$ e chave privada: $\{d, n\}$

RSA

Operação de cifragem ($m \longrightarrow c$) usa $\{e, n\}$:

$$c \equiv m^e \bmod n$$

Operação de decifragem ($c \longrightarrow m$) usa $\{d, n\}$:

$$m \equiv c^d \equiv (m^e)^d \bmod n$$

No mundo real:

- p e q têm centenas de dígitos
- operações com inteiros de precisão arbitrária
- exponenciações são **muito** pesadas

RSA - Exemplo com números pequenos

1. Escolher dois primos: $p = 61$ e $q = 53$
2. Calcular o módulo: $n = p \times q = 61 \times 53 = 3233$
3. Calcular o totiente: $\lambda(3233) = mmc(60, 52) = 780$
4. Escolher expoente $1 < e < 780$, e coprimo de 780
   - $e = 17$
   - 17 é primo e não é divisor de 780
5. Calcular expoente d tal que $d \times e = 1 \bmod \lambda(n)$
   - $d = 413$
   - $413 \times 17 \bmod \lambda(n) = 1$

6. **Chave pública**: $\{n = 3233, e = 17\}$
7. **Chave privada**: $\{n = 3233, d = 413\}$

RSA - Exemplo

Mensagem aberta:	$m = 65$ (letra "A" em ASCII)
Cifrando ($m \longrightarrow c$):	$c \equiv m^e \bmod n = 65^{17} \bmod 3233 = 2790$
Mensagem cifrada:	$c = 2790$
Decifrando ($c \longrightarrow m$):	$m \equiv c^d \bmod n = 2790^{413} \bmod 3233 = 65$
Mensagem decifrada:	$m = 65$