

Criptografia de chave assimétrica

Segurança da Informação
Curso Técnico em Redes de Computadores
ETER – FAETEC - Rio de Janeiro - RJ

O problema da criptografia simétrica.

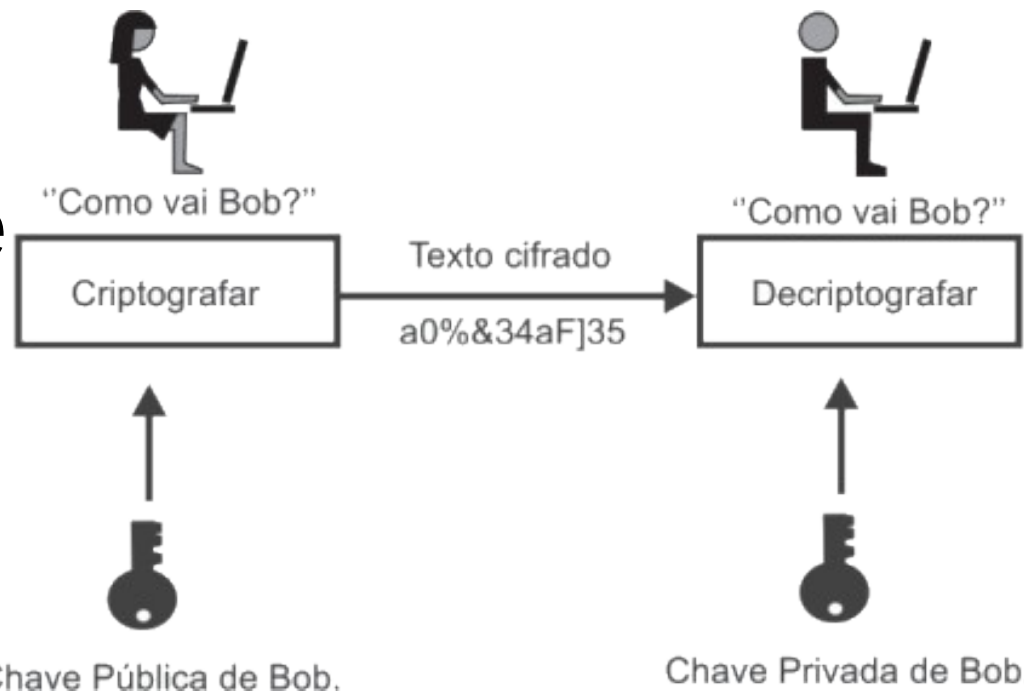
- A mesma chave é usada para os processos de cifragem e decifragem.
 - Como compartilhar a chave, de forma segura entre as partes, *a priori*?
 - A segurança é baseada no segredo da chave e na complexidade do algoritmo.
- Na prática, a criptografia simétrica foi pouco usada sozinha, e os princípios da criptografia assimétrica já eram conhecidos desde os anos 1970.

Chaves para cifragem.

- Também são medidas em bits, sendo que é comum o uso de chaves de até 2048 bits.
- Cada usuário possui um par de chaves:
 - Uma é a chave pública, compartilhada livremente e usada para cifragem.
 - Outra é a chave privada, mantida em segredo e usada para decifragem.

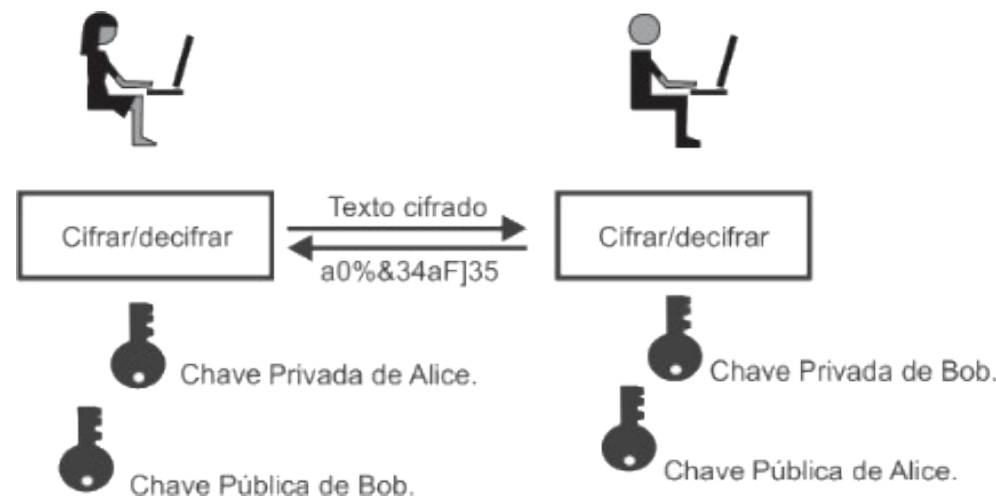
Como funciona?

- Quando o usuário A quer falar com o usuário B, é preciso que ele tenha a chave pública de A, para cifrar a mensagem antes de fazer o envio.
- Logo, há duas chaves envolvidas.



Como funciona?

- Se B quiser falar com A, ele precisa ter a chave pública de B para cifrar a mensagem de retorno. Logo, teremos quatro chaves envolvidas.



Protocolos de criptografia assimétrica.

- Os primeiros conceitos de criptografia assimétrica surgem nos trabalhos de Whitfield Diffie e Martin Hellman na Universidade de Stanford, em 1976, trazendo uma revolução no campo da criptografia.
- Temos então que: M é a mensagem, e M' é a mensagem cifrada. C é a função de cifragem, e D é a função de decifragem.
- Usar a chave pública do destinatário para cifrar e a chave privada do remetente para decifrar é uma operação **típica**.
- Usar a chave privada do remetente para decifrar e a chave pública do destinatário para cifrar é uma operação **atípica**.

Características da criptografia assimétrica.

- Todos os participantes tem acesso às chaves públicas.
- As chaves privadas são geradas localmente e não precisam ser distribuídas (aliás, é melhor que não sejam!).
- Enquanto o usuário proteger a chave privada, a comunicação que chega é segura.
- Um usuário pode mudar as chaves a qualquer tempo e publicar novamente sua chave pública.

Princípios da criptografia assimétrica.

- 1) Ao decifrar a mensagem cifrada, obtemos a mesma mensagem novamente. Ou seja, $D(C(M)) = D(M') = M$.
- 2) As funções C e D são facilmente computáveis. Logo, cifragem e decifragem podem ser feitas sem grande esforço computacional.
- 3) Conhecer C não facilita determinar D . Logo, não basta conhecer a função de cifragem para decifrar a mensagem.
- 4) A cifragem da forma decifrada da mensagem obterá a mesma mensagem novamente: $C(D(M)) = C(M') = M$.

Sobre os princípios.

- O 1o princípio define as funções típicas das chaves.
- O 4o princípio define as funções atípicas das chaves.
- Tanto o 1o quanto o 4o princípios definem que cada chave no par tem uma função primária e uma função secundária.
- O 2o e 3o princípios estão ligados às propriedades matemáticas dos protocolos assimétricos mais conhecidos.
- Estes princípios influenciam toda a criptografia assimétrica. Por exemplo, protocolos como o Oakley (parte do protocolo IKE, usado em VPNs) usa o algoritmo Diffie-Hellman para troca de chaves a serem usadas na criptografia assimétrica.

Um pouco de matemática envolvida.

- O algoritmo Diffie-Hellman usa logaritmos discretos. Logo, para dois inteiros, **a** e **b** e um número primo **p**, é possível encontrar um expoente **i**, tal que:

$$b = a^i \bmod p, \text{ onde } 0 \leq i \leq (p-1)$$

- A função mod é o o resto da divisão inteira.
- O expoente **i** é o logaritmo discreto de **b** na base **a**. Logo, **a** é a raiz primitiva de **p**.
 - Se tivermos **a**, **i** e **p**, é fácil calcular **b**.
 - Se tivermos **a**, **b** e **p**, é difícil calcular **i**.

Como se dá esse diálogo?

- Dados **a** e **p**, o usuário A escolhe um número, **Xa**, menor do que **p**, e calcula **Ya** = $a^{Xa} \bmod p$. Aí ele envia para o usuário B os valores de a, p e Ya.
- O usuário B sorteia **Xb** (também menor do que p), e calcula **Yb** = $a^{Xb} \bmod p$. O usuário B envia para A o valor Yb, mas mantém segredo sobre Xb.
- Após este procedimento, ambos podem gerar uma chave, K, de forma independente. Se um terceiro quiser gerar a chave K, terá que resolver o problema do logaritmo, mas sem saber Xa ou Xb.

Para facilitar o processamento.

- Uma maneira de facilitar o processo de cálculo do resto da divisão é usando o método da congruência. Logo:
- Então:

$$X^{(a+b)} \bmod Y = [(X^a \bmod Y) * (X^b \bmod Y)] * \bmod Y$$

$$5^{36} \bmod 97 = 5^{(20+16)} \bmod 97 = ((5^{20} \bmod 97) * (5^{16} \bmod 97)) * \bmod 97$$

Exemplo.

- Suponhamos que o sistema determine $a = 5$ e $p = 97$.
- O usuário A sorteia um número, X_a , 36 (menor do que 97), e calcula Y_a . Ele obtém 50 ($Y_a = a^{X_a} \bmod p$ – nesse caso, $50 = 5^{36} \bmod 97$).
- O usuário A envia 50, 97 e 5 para B.
- O usuário B sorteia X_b , por exemplo 58, e calcula $Y_b = 44$ ($44 = 5^{58} \bmod 97$). Ele envia Y_b para A.
- As chaves são iguais?

Verificação das chaves.

- A chave K_a , gerada por A, é $Yb^{x_a} \bmod p$. Logo, $K_a = 44^{36} \bmod 97$, que pode ser calculada como $(44^6 \bmod 97)^6 \bmod 97$. Obteremos $K_a = 75$.
- A chave K_b , gerada por B, é $Ya^{x_b} \bmod p$. Logo, $K_b = 50^{58} \bmod 97$. que pode ser calculada como $((50^7 \bmod 97)^8 \times (50^2 \bmod 97)) \bmod 97$. O resultado é que $K_b = 75$!

RSA

- O protocolo RSA é uma criação de três pesquisadores do MIT: Ron Rivest, Adi Shamir e Leonard Adleman.
- O protocolo foi criado em 1978, e ao contrário do DES, é amplamente usado até hoje.
- Os três pesquisadores fundaram a RSA Security em 1982, e o protocolo foi **patenteado**, em 1983, e a patente durou até 2000. Hoje o uso do RSA é livre.
- Hoje em dia, a empresa RSA Security pertence à Dell, e o protocolo domina 90% das aplicações comerciais que usam criptografia assimétrica.

Como funciona o RSA?

- A complexidade do RSA reside em fazer a fatoração de números primos muito grandes.
- Note que multiplicar dois números primos muito grandes é rápido e fácil. Mas fatorar o produto dessa multiplicação, de forma a encontrar os dois números primos originais... Isto é bem difícil.
- Esta ideia está relacionada diretamente ao 2º princípio estabelecido por Diffie e Hellman.

Como funciona o RSA, passo a passo?

- 1) Escolha dois números primos muito grandes, **p** e **q**. Se os números forem pequenos, é fácil verificar que são primos. Por isso é que se usam números primos muito grandes, com muitos algarismos.
- 2) Calcula-se **n**, que é o produto de **p** e **q**. Este número fará parte tanto da chave pública quanto da chave privada:

$$n = p \times q$$

Como funciona o RSA, passo a passo?

- 3) Calcule **z**, que é um número intermediário.

$$z = (p-1) \times (q-1)$$

- 4) Escolha **d**, que é um número menor do que **n**. Os números **d** e **z** devem ser primos entre si, ou seja: *não há fatores em comum entre eles*. Por exemplo, 9 e 10 não são números primos, mas são números primos entre si, já que:

$$9 = 3 \times 3; 10 = 2 \times 5.$$

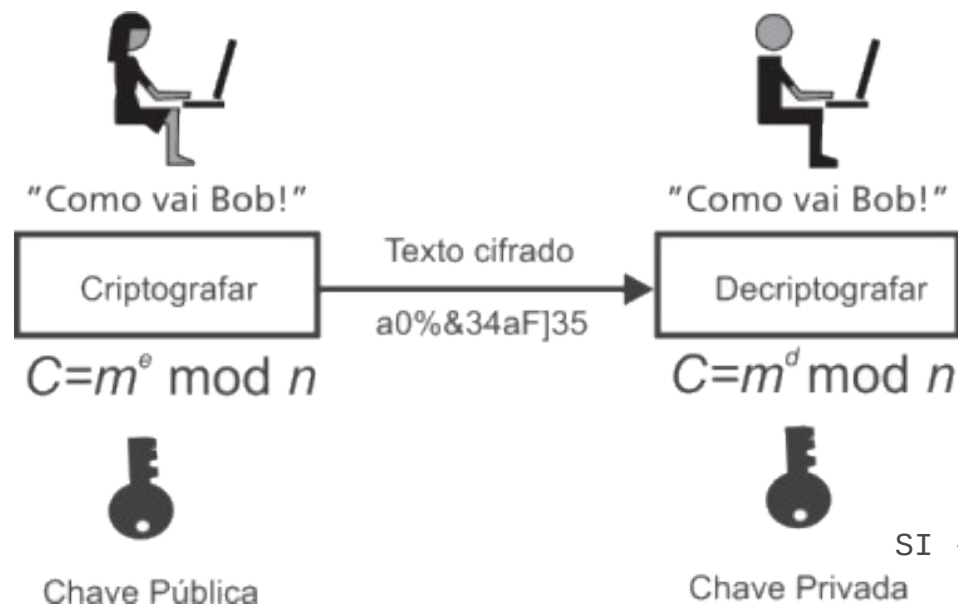
Como funciona o RSA, passo a passo?

5) Finalmente, calcule **e**, a partir de **z** e **d**. O número **e** é tal que ele, multiplicado por **d - 1** é divisível por **z**. Ou seja, o resto de **e** multiplicado por **d** e dividido por **z** é igual a 1.

$$(e \times d) \bmod z = 1$$

E as chaves?

- O RSA usa o par (n, d) como a chave privada, e o par (n, e) como a chave pública.



Vamos a um exemplo.

- 1) Vamos escolher **p** e **q**, que serão 5 e 7, respectivamente.
- 2) A partir da 1ª equação, **n**=35 (5 x 7).
- 3) A partir da 2ª equação, **z**=24 (4 x 6).
- 4) Agora, obteremos **d** < **n**, tal que **d** e **z** sejam primos entre si. Tomaremos **d**=29.
- 5) Escolheremos **e**, tal que a 3ª equação seja satisfeita. Tomaremos **e**=5, pois:

$$(29 \times 5) - 1 = 144.$$

$$144 \div 24 = 6 (\text{resto } 0).$$

Pondo em prática o exemplo.

- Temos que **p**=5; **q**=7; **n**=35; **z**=24; **d**=29 e **e**=5.
- Tomemos o texto em claro **12** (m=12). O processo de cifragem consiste em realizar a seguinte operação para gerar o texto cifrado **c**:
- Logo: $m^e \bmod n$
- E o texto cifrado é $12^5 \bmod 35 = 248.832 \bmod 35 = 17$.

Decifrando o exemplo.

- O processo de decifragem consiste em realizar a seguinte operação:

$$c^d \bmod n$$

- Como $c=17$, então a operação é:

$$17^{29} \bmod 35 = 481968572106750915091411825223071697 \bmod 35 = 12$$

- E o texto decifrado é 12.

Vamos a outro exemplo.

- 1) Vamos escolher **p** e **q**, que serão 11 e 3, respectivamente.
- 2) A partir da 1ª equação, **n**=**p** x **q** = 11 x 3 = 33.
- 3) A partir da 2ª equação, **z**=(**p**-1) x (**q**-1) = 10 x 2 = 20.
- 4) Agora, obteremos **d**, tal que **d** seja primo entre ele e **z** = 20. Tomaremos **d** = 3.
- 5) Procuraremos um valor para **e**, tal que **e** x **d** = 1 mod 20. Verificamos que **e** = 7 funciona.
- 6) Logo, a chave pública será **(n, e) = (33, 7)**.
- 7) A chave privada será **(n, d) = (33, 3)**.

Pondo em prática o exemplo.

- Tomemos o texto em claro **8** ($m=8$). O texto cifrado será então $m^e \bmod n$. Logo $8^7 = 2097152 = 2 \bmod 33$.
- O texto cifrado será 2. Para decifrar o texto, faremos $c^d \bmod n$.
- Como $c = 2$, teremos então que $2^3 = 8 \bmod 33 = 8$
- O texto decifrado é 8.

Vamos a mais um exemplo.

- 1) Vamos escolher **p** e **q**, que serão 17 e 11, respectivamente.
- 2) A partir da 1ª equação, **n**=**p** x **q** = 17 x 11 = 187.
- 3) A partir da 2ª equação, **z**=(**p**-1) x (**q**-1) = 16 x 10 = 160.
- 4) Agora, obteremos **e**, tal que **e** seja primo entre ele e **z**. Tomaremos **e** = 7.
- 5) Procuraremos um valor para **d**, tal que **e** x **d** = 1 mod 160. Verificamos que **d** = 23 funciona.
- 6) Logo, a chave pública será **(n, e) = (187, 7)**.
- 7) A chave privada será **(n, d) = (187, 23)**.

Pondo em prática o exemplo.

- Tomemos o texto em claro **88** ($m = 88$). O texto cifrado será então $c = m^e \bmod n$. Logo $88^7 \bmod 187 = 11$.
- O texto cifrado será 11. Para decifrar o texto, faremos $c^d \bmod n$.
- Como $c = 11$, teremos então que $11^{23} \bmod 187 = 88$.
- O texto decifrado é 88.

Alguns sites para experimentar o protocolo RSA.

1) https://bit.ly/RSA_1

2) <https://bit.ly/RSA-gerador>

3) <https://bit.ly/RSA-passo-a-passo>

- Nota: No exemplo anterior, usamos um texto cifrado muito pequeno (**12**). Se você desejar trabalhar com textos maiores (como por exemplo, **2191** ou **Turma de SI**), é necessário usar números primos maiores.

Exercício para fazer com o site.

- No site 1, tente com a mensagem **Turma de SI**, gere as chaves privada e pública e experimente. O resultado será um número em Base 64.
- No site 3, temos que $p=1039$, $q=1753$, $e=13$ e a mensagem é **Turma de SI**. O texto cifrado deverá ser: 283215, 301355, 1266313, 1107662, 1452891, 1047912.
- No site 3, temos que $p=3$, $q=11$, $e=7$ e a mensagem a ser cifrada é 6. O texto cifrado deverá ser 30.

Duas abordagem para tentar quebrar o RSA.

- 1) Força bruta: Tentar todas as possíveis combinações. Quanto maior o número de bits em e e d , mais seguro é o algoritmo, mas mais lento ele é para processar os dados.
- 2) Criptoanálise: Fatorando n em dois números primos, p e q . Em 1994, 1600 computadores foram usados para quebrar uma chave onde n continha 428 bits. E eles levaram 8 meses!

Segurança do RSA.

- O maior n quebrado até hoje é o número
123018668453011775513049495838496272077285356959533479219732
245215172640050726365751874520219978646938995647494277406384
592519255732630345373154826850791702612214291346167042921431
1602221240479274737794080665351419597459856902.
- Após dois anos de trabalho, foi descoberto que este n é o produto de
 $p=3347807169895689878604416984821269081770479498371376856891243$
1388982883793878002287614711652531743087737814467999489
 e
 $q=3674604366679959042824463379962795263227915816434308764267603$
2283815739666511279233373417143396810270092798736308917
- Esta é a chave **RSA-768**. Mas o algoritmo permite chaves de até 2048 bits, com variações que aceitam até 4096 bits.

Outros algoritmos – El Gammal.

- Criado pelo criptólogo egípcio Taher El Gammal em 1984. Atende aos princípios enunciados por Diffie-Hellman, e aplica a teoria do logaritmo discreto para criar um protocolo completo, que realiza a troca de chaves e cifra e decifra informações.
- Oferece chaves de cifragem e decifragem menores do que o protocolo RSA.

Outros algoritmos – curvas elípticas.

- As curvas elípticas não são um sistema de criptografia, mas uma maneira diferente de fazer os cálculos em um sistema de chave pública.
- Também conhecido como ECC, é uma variação do El Gammal e foi proposto em 1985 com base na teoria matemática das curvas elípticas envolvendo corpos finitos (álgebra pura...).
- É um desdobramento da teoria dos logaritmos discretos, e oferece chaves de cifragem e decifragem menores do que o protocolo RSA – ou seja, pode ser mais eficiente.

Outros algoritmos - Knapsack

- Método da “mochila”. É um algoritmo que é fácil e rápido para cifragem, mas que é inseguro (foi quebrado em 1983!)
- <https://bit.ly/KNAPSACK>

Conclusão e resumo.

- A criptografia assimétrica envolve um par de chaves, uma pública (que cifra) e uma privada (que decifra).
- W. Diffie e M. Hellman enunciaram os quatro princípios da criptografia assimétrica que são seguidos em todas as implementações.
- O algoritmo Diffie-Hellman é muito usado isoladamente para troca de chaves de sessão de forma segura em um canal teoricamente inseguro.
- O principal protocolo é o RSA, amplamente usado até hoje. Existem outros protocolos, como o El Gammal e o ECC, mas são pouco utilizados.