Introdução à Criptografia

Criptografia de Chave Pública RSA

Prof. Rodrigo Minetto
rminetto@dainf.ct.utfpr.edu.br
Universidade Tecnológica Federal do Paraná

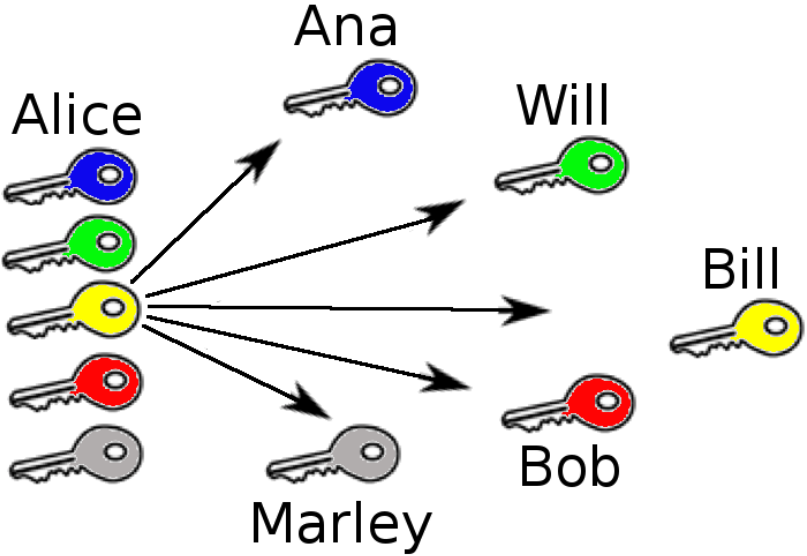
Baseado em: Understanding Cryptography by Paar e Pelzl

Sumário

Introdução

A troca de chaves imaginada por Diffie-Hellman-Merkle proporcionou uma evolução imensa para a criptografia moderna, mas o sistema não era perfeito. Imagine um cenário onde Alice deseja enviar uma mensagem para Bob, mas ambos não estão conectados ao mesmo tempo. Como proceder? Alice e Bob podem utilizar o e-mail para essa tarefa, mas qual o delay associado ao procedimento?

Número grande de chaves no DHKE:



Introdução

O número de chaves em sistema simétrico de criptografia é alto. Se cada par de usuários precisar combinar uma chave para a criptografia então em uma rede com n usuários existem

$$\frac{n \times (n-1)}{2}$$

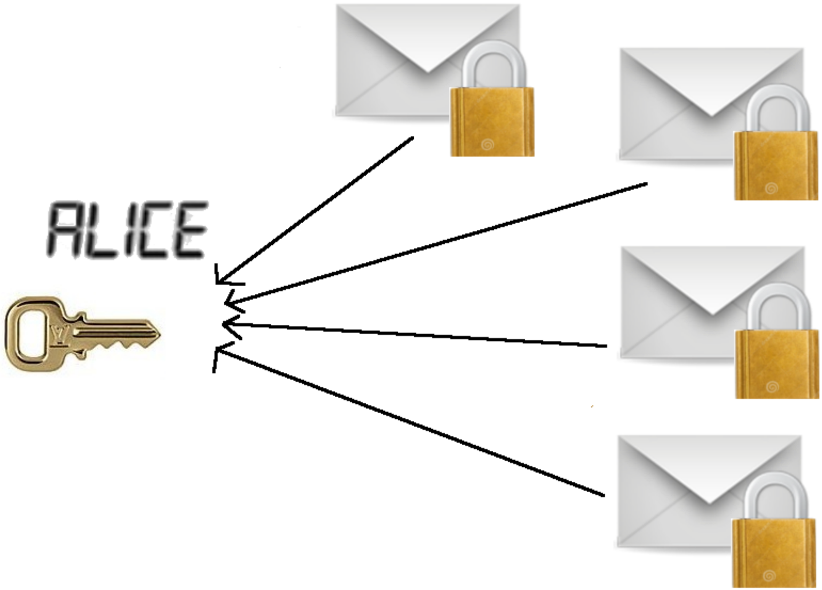
chaves diferentes e cada usuário deve guardar em segredo $n-1$ chaves.

Introdução

Em 1975, Diffie desenvolveu uma abordagem completamente diferente para resolver o problema da distribuição de chaves. A abordagem é conhecida como criptografia por **chave assimétrica**, sistema onde a chave de cifragem e decifragem não são idênticas. Em uma cifra assimétrica, se Alice sabe a chave de cifragem, ela pode cifrar mas não decifrar a mensagem. Esta distinção entre cifragem e decifragem é o que torna a cifra simétrica muito especial.

RSA

RSA



Introdução

É necessário enfatizar que embora Diffie tivesse concebido a ideia de uma cifra assimétrica, ele não tinha nenhuma que servisse de exemplo. Ele publicou um resumo de sua ideia em 1975 na esperança que alguém pudesse ajudar a encontrar uma função que permitisse cifrar uma mensagem usando uma chave pública, mas que fosse inviável decifrar a mensagem sem a chave privada (**trapdoor**).

Introdução

Em 1975 e 1976 vários cientistas buscaram a função mágica mas sem resultados e começaram a duvidar que ela existisse. Rivest, Shamir e Adleman (RSA), pesquisadores do MIT, anunciaram a descoberta da função em 1977 em um artigo intitulado "Um novo tipo de cifra que levará milhões de anos para ser decifrado" da Scientific American.

Introdução

No artigo, os leitores forem desafiados a fatorar N em dois primos p e q e usá-los para decifrar a mensagem adicionada ao artigo

$$N = 114{,}381{,}625{,}757{,}888{,}867{,}669{,}235{,}779{,}976{,}146{,}612{,}010{,}218{,}296{,}721{,}242{,}362{,}562{,}561{,}842{,}935{,}706{,}935{,}245{,}733{,}897{,}830{,}597{,}123{,}563{,}958{,}705{,}058{,}989{,}075{,}147{,}599{,}290{,}026{,}879{,}543{,}541$$

Após 17 anos a mensagem foi decifrada com voluntários de países como Austrália, USA, Inglaterra, etc.

Sumário

Seja a chave pública $k_{\mathrm{pub}} = (e, \mathbf{n})$ e o texto em claro $\mathbf{x}$, a função de ciframento é definida por:

Ciframento

$$\mathbf{y} = \mathbf{x}^{e} \bmod \mathbf{n}$$

onde $x, y \in \mathrm{Z}_n$.

RSA

Seja a chave privada $k_{\text{priv}} = (\mathbf{d}, \mathbf{n})$ e o texto cifrado $\mathbf{y}$, a função de deciframento é definida por:

Deciframento

$$\mathbf{x} = \mathbf{y}^{\mathbf{d}} \bmod \mathbf{n}$$

onde $x, y \in \mathbb{Z}_n$.

Exemplo de cifragem e decifragem:

Exemplo de cifragem e decifragem:

Alice		Bob
	$\leftarrow$	$k_{\mathrm{pub}} = (\mathbf{3}, \mathbf{33})$

RSA

Exemplo de cifragem e decifragem:

Alice

Bob

$\leftarrow \quad k_{\mathrm{pub}} = (\mathbf{3}, \mathbf{33})$

$\mathbf{x} = \mathbf{4}$

RSA

Exemplo de cifragem e decifragem:

Alice

Bob

$\leftarrow \quad k_{\text{pub}} = (\mathbf{3}, \mathbf{33})$

$\mathbf{x} = \mathbf{4}$

$\mathbf{y} = \mathbf{x}^{e} \bmod \mathbf{n}$

RSA

Exemplo de cifragem e decifragem:

Alice		Bob
	$\leftarrow$	$k_{\text{pub}} = (\mathbf{3}, \mathbf{33})$

$\mathbf{x} = \mathbf{4}$
$\mathbf{y} = \mathbf{x}^{\boldsymbol{e}} \bmod \mathbf{n}$
$\mathbf{y} = \mathbf{4}^{\mathbf{3}} \bmod \mathbf{33}$

Exemplo de cifragem e decifragem:

Alice		Bob
	$\leftarrow$	$k_{\mathrm{pub}} = (3, 33)$
$x = 4$		
$y = x^{e} \bmod n$		
$y = 4^{3} \bmod 33$		
$y = 31$	$\rightarrow$	

Exemplo de cifragem e decifragem:

Alice		Bob
	$\leftarrow$	$k_{\text{pub}} = (\mathbf{3}, \mathbf{33})$
$\mathbf{x} = \mathbf{4}$		
$\mathbf{y} = \mathbf{x}^{e} \bmod \mathbf{n}$		
$\mathbf{y} = \mathbf{4}^{3} \bmod \mathbf{33}$		
$\mathbf{y} = \mathbf{31}$	$\rightarrow$	
		$k_{\text{priv}} = (\mathbf{7}, \mathbf{33})$

Exemplo de cifragem e decifragem:

Alice		Bob
	$\leftarrow$	$k_{\text{pub}} = (3, 33)$
$x = 4$		
$y = x^{e} \bmod n$		
$y = 4^{3} \bmod 33$		
$y = 31$	$\rightarrow$	
		$k_{\text{priv}} = (7, 33)$
		$x = y^{d} \bmod n$

RSA

Exemplo de cifragem e decifragem:

Alice		Bob
	$\leftarrow$	$k_{\text{pub}} = (3, 33)$
$x = 4$		
$y = x^e \bmod n$		
$y = 4^3 \bmod 33$		
$y = 31$	$\rightarrow$	
		$k_{\text{priv}} = (7, 33)$
		$x = y^d \bmod n$
		$x = 31^7 \bmod 33$

Exemplo de cifragem e decifragem:

Alice		Bob
	$\leftarrow$	$k_{\mathrm{pub}} = (\mathbf{3}, \mathbf{33})$
$\mathbf{x} = \mathbf{4}$		
$\mathbf{y} = \mathbf{x}^{e} \bmod \mathbf{n}$		
$\mathbf{y} = \mathbf{4}^{3} \bmod \mathbf{33}$		
$\mathbf{y} = \mathbf{31}$	$\rightarrow$	
		$k_{\mathrm{priv}} = (\mathbf{7}, \mathbf{33})$
		$\mathbf{x} = \mathbf{y}^{\mathbf{d}} \bmod \mathbf{n}$
		$\mathbf{x} = \mathbf{31}^{\mathbf{7}} \bmod \mathbf{33}$
		$\mathbf{x} = \mathbf{4}$

Sumário

Geração de chaves

1. Selecione dois primos grandes p e q;
2. Calcule $\mathbf{n} = p \times q$;
3. Calcule $\phi(\mathbf{n}) = (p - 1) \times (q - 1)$;
4. Escolha $0 < e < \phi(\mathbf{n})$ tal que
   $\textbf{MDC-}\textsc{Euclides-Estendido}(\phi, e, a, \mathbf{b}) = 1$
   Enquanto $(\mathbf{b} < 0)$ $\{$ $\mathbf{b} = \mathbf{b} + \phi(\mathbf{n})$; $\}$
5. Calcule $\mathbf{d} = \mathbf{b} \bmod \phi(\mathbf{n})$; $(\mathbf{b} = e^{-1})$
6. Publique a chave $k_{\text{pub}} = (e, \mathbf{n})$;
7. Proteja a chave $k_{\text{priv}} = (\mathbf{d}, \mathbf{n})$;

RSA

Geração de chaves

1. Números primos $p = 3$ e $q = 11$;
2. Calcule $\mathbf{n} = p \times q$;
3. Calcule $\phi(\mathbf{n}) = (p - 1) \times (q - 1)$;
4. Escolha $0 < e < \phi(\mathbf{n})$ tal que
   $\textbf{MDC-}\textsc{Euclides-Estendido}(\phi, e, a, \mathbf{b}) = 1$
   Enquanto $(\mathbf{b} < 0)$ $\{\ \mathbf{b} = \mathbf{b} + \phi(\mathbf{n});\ \}$
5. Calcule $\mathbf{d} = \mathbf{b} \bmod \phi(\mathbf{n})$; $(\mathbf{b} = e^{-1})$
6. Publique a chave $k_{\mathrm{pub}} = (e, \mathbf{n})$;
7. Proteja a chave $k_{\mathrm{priv}} = (\mathbf{d}, \mathbf{n})$;

Geração de chaves

1. Números primos $p = 3$ e $q = 11$;
2. Calcule $\mathbf{n} = 3 \times 11$;
3. Calcule $\phi(\mathbf{n}) = (p - 1) \times (q - 1)$;
4. Escolha $0 < e < \phi(\mathbf{n})$ tal que
   $\textbf{MDC}\text{-}\textsc{Euclides-Estendido}(\phi, e, a, \mathbf{b}) = 1$
   Enquanto $(\mathbf{b} < 0)$ $\{ \mathbf{b} = \mathbf{b} + \phi(\mathbf{n}); \}$
5. Calcule $\mathbf{d} = \mathbf{b} \bmod \phi(\mathbf{n})$; $(\mathbf{b} = e^{-1})$
6. Publique a chave $k_{\mathrm{pub}} = (e, \mathbf{n})$;
7. Proteja a chave $k_{\mathrm{priv}} = (\mathbf{d}, \mathbf{n})$;

Geração de chaves

1. Números primos $p = 3$ e $q = 11$;
2. Calcule $\mathbf{n} = 33$;
3. Calcule $\phi(\mathbf{n}) = (p - 1) \times (q - 1)$;
4. Escolha $0 < e < \phi(\mathbf{n})$ tal que
   MDC-$\textsc{Euclides-Estendido}(\phi, e, a, \mathbf{b}) = 1$
   Enquanto $(\mathbf{b} < 0)$ $\{$ $\mathbf{b} = \mathbf{b} + \phi(\mathbf{n})$; $\}$
5. Calcule $\mathbf{d} = \mathbf{b} \bmod \phi(\mathbf{n})$; $(\mathbf{b} = e^{-1})$
6. Publique a chave $k_{\mathrm{pub}} = (e, \mathbf{n})$;
7. Proteja a chave $k_{\mathrm{priv}} = (\mathbf{d}, \mathbf{n})$;

RSA

Geração de chaves

1. Números primos $p = 3$ e $q = 11$;
2. Calcule $\mathbf{n} = 33$;
3. Calcule $\phi(\mathbf{n}) = (3 - 1) \times (11 - 1)$;
4. Escolha $0 < e < \phi(\mathbf{n})$ tal que
   $\textbf{MDC}\text{-}\textsc{Euclides-Estendido}(\phi, e, a, \mathbf{b}) = 1$
   Enquanto $(\mathbf{b} < 0)$ $\{$ $\mathbf{b} = \mathbf{b} + \phi(\mathbf{n})$; $\}$
5. Calcule $\mathbf{d} = \mathbf{b} \bmod \phi(\mathbf{n})$; $(\mathbf{b} = e^{-1})$
6. Publique a chave $k_{\mathrm{pub}} = (e, \mathbf{n})$;
7. Proteja a chave $k_{\mathrm{priv}} = (\mathbf{d}, \mathbf{n})$;

RSA

Geração de chaves

1. Números primos $p = 3$ e $q = 11$;
2. Calcule $\mathbf{n} = 33$;
3. Calcule $\phi(\mathbf{n}) = 2 \times 10$;
4. Escolha $0 < e < \phi(\mathbf{n})$ tal que
   MDC-$\textsc{Euclides-Estendido}(\phi, e, a, \mathbf{b}) = 1$
   Enquanto $(\mathbf{b} < 0)$ $\{$ $\mathbf{b} = \mathbf{b} + \phi(\mathbf{n})$; $\}$
5. Calcule $\mathbf{d} = \mathbf{b} \bmod \phi(\mathbf{n})$; $(\mathbf{b} = e^{-1})$
6. Publique a chave $k_{\mathrm{pub}} = (e, \mathbf{n})$;
7. Proteja a chave $k_{\mathrm{priv}} = (\mathbf{d}, \mathbf{n})$;

RSA

Geração de chaves

1. Números primos $p = 3$ e $q = 11$;
2. Calcule $\mathbf{n} = 33$;
3. Calcule $\phi(\mathbf{n}) = \mathbf{20}$;
4. Escolha $0 < e < \phi(\mathbf{n})$ tal que
   $\textbf{MDC}\text{-}\textsc{Euclides-Estendido}(\phi, e, a, \mathbf{b}) = 1$
   Enquanto $(\mathbf{b} < 0)$ $\{\ \mathbf{b} = \mathbf{b} + \phi(\mathbf{n});\ \}$
5. Calcule $\mathbf{d} = \mathbf{b} \bmod \phi(\mathbf{n})$; $(\mathbf{b} = e^{-1})$
6. Publique a chave $k_{\text{pub}} = (e, \mathbf{n})$;
7. Proteja a chave $k_{\text{priv}} = (\mathbf{d}, \mathbf{n})$;

RSA

Geração de chaves

1. Números primos $p = 3$ e $q = 11$;
2. Calcule $\mathbf{n} = 33$;
3. Calcule $\phi(\mathbf{n}) = \mathbf{20}$;
4. Escolha $0 < e = \mathbf{3} < \mathbf{20}$ tal que
   $\textbf{MDC}\text{-}\textsc{Euclides-Estendido}(\phi, e, a, \mathbf{b}) = 1$
   Enquanto $(\mathbf{b} < 0)$ $\{$ $\mathbf{b} = \mathbf{b} + \phi(\mathbf{n})$; $\}$
5. Calcule $\mathbf{d} = \mathbf{b} \bmod \phi(\mathbf{n})$; $(\mathbf{b} = e^{-1})$
6. Publique a chave $k_{\mathrm{pub}} = (e, \mathbf{n})$;
7. Proteja a chave $k_{\mathrm{priv}} = (\mathbf{d}, \mathbf{n})$;

RSA

Geração de chaves

1. Números primos $p = 3$ e $q = 11$;
2. Calcule $\mathbf{n} = 33$;
3. Calcule $\phi(\mathbf{n}) = \mathbf{20}$;
4. Escolha $0 < e = \mathbf{3} < \mathbf{20}$ tal que
   $\textbf{MDC}\text{-}\textsc{Euclides-Estendido}(\mathbf{20}, \mathbf{3}, a, \mathbf{7}) = 1$
   Enquanto ($\mathbf{b} < 0$) { $\mathbf{b} = \mathbf{b} + \phi(\mathbf{n})$; }
5. Calcule $\mathbf{d} = \mathbf{b} \bmod \phi(\mathbf{n})$; $(\mathbf{b} = e^{-1})$
6. Publique a chave $k_{\text{pub}} = (e, \mathbf{n})$;
7. Proteja a chave $k_{\text{priv}} = (\mathbf{d}, \mathbf{n})$;

RSA

Geração de chaves

1. Números primos $p = 3$ e $q = 11$;
2. Calcule $\mathbf{n} = 33$;
3. Calcule $\phi(\mathbf{n}) = \mathbf{20}$;
4. Escolha $0 < e = \mathbf{3} < \mathbf{20}$ tal que

   $\textbf{MDC}\text{-}\textsc{Euclides-Estendido}(\mathbf{20}, \mathbf{3}, a, \mathbf{7}) = 1$

   Enquanto $(\mathbf{7} < 0)$ $\{$ $\mathbf{b} = \mathbf{b} + \phi(\mathbf{n})$; $\}$
5. Calcule $\mathbf{d} = \mathbf{b} \bmod \phi(\mathbf{n})$; $(\mathbf{b} = e^{-1})$
6. Publique a chave $k_{\text{pub}} = (e, \mathbf{n})$;
7. Proteja a chave $k_{\text{priv}} = (\mathbf{d}, \mathbf{n})$;

RSA

Geração de chaves

1. Números primos $p = 3$ e $q = 11$;
2. Calcule $\mathbf{n} = 33$;
3. Calcule $\phi(\mathbf{n}) = \mathbf{20}$;
4. Escolha $0 < e = \mathbf{3} < \mathbf{20}$ tal que

   $\textbf{MDC}\text{-}\textsc{Euclides-Estendido}(\mathbf{20}, \mathbf{3}, a, \mathbf{7}) = 1$

   Enquanto $(\mathbf{7} < 0)$ $\{$ $\mathbf{b} = \mathbf{b} + \phi(\mathbf{n})$; $\}$
5. Calcule $\mathbf{d} = \mathbf{7} \bmod \mathbf{20}$; $(\mathbf{b} = \mathbf{7})$
6. Publique a chave $k_{\text{pub}} = (e, \mathbf{n})$;
7. Proteja a chave $k_{\text{priv}} = (\mathbf{d}, \mathbf{n})$;

RSA

Geração de chaves

1. Números primos $p = 3$ e $q = 11$;
2. Calcule $\mathbf{n} = 33$;
3. Calcule $\phi(\mathbf{n}) = \mathbf{20}$;
4. Escolha $0 < e = \mathbf{3} < \mathbf{20}$ tal que
   $\textbf{MDC-}\textsc{Euclides-Estendido}(\mathbf{20}, \mathbf{3}, a, \mathbf{7}) = 1$
   Enquanto $(\mathbf{7} < 0)$ $\{$ $\mathbf{b} = \mathbf{b} + \phi(\mathbf{n})$; $\}$
5. Calcule $\mathbf{d} = \mathbf{7}$;
6. Publique a chave $k_{\mathrm{pub}} = (e, \mathbf{n})$;
7. Proteja a chave $k_{\mathrm{priv}} = (\mathbf{d}, \mathbf{n})$;

RSA

Geração de chaves

1. Números primos $p = 3$ e $q = 11$;
2. Calcule $\mathbf{n} = 33$;
3. Calcule $\phi(\mathbf{n}) = \mathbf{20}$;
4. Escolha $0 < e = \mathbf{3} < \mathbf{20}$ tal que
   $\textbf{MDC}\text{-}\textsc{Euclides-Estendido}(\mathbf{20}, \mathbf{3}, a, \mathbf{7}) = 1$
   Enquanto $(\mathbf{7} < 0)$ $\{\ \mathbf{b} = \mathbf{b} + \phi(\mathbf{n});\ \}$
5. Calcule $\mathbf{d} = \mathbf{7}$;
6. Publique a chave $k_{\text{pub}} = (\mathbf{3}, \mathbf{33})$;
7. Proteja a chave $k_{\text{priv}} = (\mathbf{d}, \mathbf{n})$;

RSA

Geração de chaves

1. Números primos $p = 3$ e $q = 11$;
2. Calcule $\mathbf{n} = 33$;
3. Calcule $\phi(\mathbf{n}) = \mathbf{20}$;
4. Escolha $0 < e = \mathbf{3} < \mathbf{20}$ tal que
   $\textbf{MDC}\text{-}\textsc{Euclides-Estendido}(\mathbf{20}, \mathbf{3}, a, \mathbf{7}) = 1$
   Enquanto $(\mathbf{7} < 0)$ $\{$ $\mathbf{b} = \mathbf{b} + \phi(\mathbf{n})$; $\}$
5. Calcule $\mathbf{d} = \mathbf{7}$;
6. Publique a chave $k_{\text{pub}} = (\mathbf{3}, \mathbf{33})$;
7. Proteja a chave $k_{\text{priv}} = (\mathbf{7}, \mathbf{33})$;

Sumário

Máximo divisor comum - **mdc** (greatest common divisor - gdc): é o maior número inteiro positivo que divide dois ou mais números inteiros. Seja o mdc de dois números inteiros positivos **a** e **b** representado por:

$$\mathrm{mdc}\,(a, b)$$

Algoritmo de Euclides

O mdc pode ser determinado pela decomposição dos números em fatores primos. Algoritmo:

- Decompor os números dados em fatores primos.
- Separar os fatores primos comuns.
- Fazer o produtos desses fatores.

$$\mathrm{mdc}\,(a, b) = \mathrm{mdc}\,(84, 60)$$

$$\begin{aligned} a &= 84 = 2 \times 2 \times 3 \times 7 \\ b &= 60 = 2 \times 2 \times 3 \times 5 \end{aligned}$$

$$2 \times 2 \times 3 = 12 = \mathrm{mdc}\,(84, 30)$$

Algoritmo de Euclides

Um algoritmo famoso para calcular mdc é conhecido como **Algoritmo de Euclides** e se baseia na simples observação

$$\mathrm{mdc}\,(a, b) = \mathrm{mdc}\,(a - b, b)$$

assumindo que $a > b$ e que ambos são positivos inteiros.

Algoritmo de Euclides

Por exemplo $\mathrm{mdc}\,(a, b) = \mathrm{mdc}\,(84, 30)$

$\mathrm{mdc}\,(84, 30) = \mathrm{mdc}\,(84-30, 30) = \mathrm{mdc}\,(54, 30)$

pois a fatoração

$$\begin{aligned} a &= 54 = 2 \times 3 \times 3 \times 3 \\ b &= 30 = 2 \times 3 \times 5 \end{aligned}$$

O mdc é o produto dos divisores comuns

$$2 \times 3 = 6 = \mathrm{mdc}\,(84, 30)$$

Algoritmo de Euclides

A observação $\mathrm{mdc}\ (a, b) = \mathrm{mdc}\ (a - b, b)$ permite ainda concluir que

$$\mathrm{mdc}\,(a - 2b, b) = \cdots = \mathrm{mdc}\,(a - mb, b)$$

até que $(a - mb) > 0$, ou seja

$$\mathrm{mdc}\,(a, b) = \mathrm{mdc}\,(a \bmod b, b)$$

como $a \bmod b < b$, então o processo continua com a troca de lado dos valores

$$\mathrm{mdc}\,(a, b) = \mathrm{mdc}\,(b, a \bmod b)$$

Algoritmo de Euclides

Exemplo, calcule o $mdc\,(84, 30)$

$$\begin{array}{r|l} (\textit{dividendo})\ 84 & 30 \quad (\text{divisor}) \\ \hline (\textit{resto})\ 24 & 2\ (\textit{quociente}) \end{array}$$

o processo continua trocando dividendo por divisor e divisor por resto

$$\begin{array}{r|l} 30 & 24 \\ \hline \mathbf{6} & 1 \end{array} \qquad\qquad \begin{array}{r|l} 24 & 6 \\ \hline 0 & 4 \end{array}$$

o **mdc** é o último resto não nulo do processo das divisões sucessivas, no caso o valor **6**.

Exemplo, calcule o *mdc* $(973, 301)$

$$\begin{array}{r|l} 973 & 301 \\ \hline 70 & 3 \end{array} \qquad \begin{array}{r|l} 301 & 70 \\ \hline 21 & 4 \end{array} \qquad \begin{array}{r|l} 70 & 21 \\ \hline \mathbf{7} & 3 \end{array} \qquad \begin{array}{r|l} 21 & 7 \\ \hline 0 & 3 \end{array}$$

o **mdc** é o último resto não nulo do processo das divisões sucessivas, no caso o valor **7**.

Entrada: inteiros positivos a, b com $a > b$.
Saída: mdc (a, b).

Algoritmo de Euclides

Euclides (a, b)
  Se $(b = 0)$
    Retorne a;
  Retorne Euclides $(b, a \bmod b)$;
Fim

Algoritmo Estendido de Euclides

O algoritmo de Euclides pode ser facilmente modificado para nos fornecer mais que o $\mathrm{mdc}\,(a, b)$. Podemos estender o algoritmo para que ele compute números inteiros x e y tais que (identidade de Bézout)

$$xa + yb = \mathrm{mdc}\,(a, b)$$

Note que os números x e y são uma prova ou certificado da correção da resposta.

Algoritmo Estendido de Euclides

$$mdc\,(973, 301) = 7 = 973\times ? + 301\times ?$$

$$\begin{array}{r|l} 973 & 301 \\ \cline{2-2} 70 & \;\;3 \end{array} \qquad \begin{array}{r|l} 301 & 70 \\ \cline{2-2} 21 & \;4 \end{array} \qquad \begin{array}{r|l} 70 & 21 \\ \cline{2-2} \mathbf{\color{blue}7} & \;3 \end{array} \qquad \begin{array}{r|l} 21 & 7 \\ \cline{2-2} 0 & 3 \end{array}$$

$$\begin{aligned} 70 &= 973 + 301 \times (-3) \\ 21 &= 301 + 70 \times (-4) \\ 7 &= 70 + 21 \times (-3) \end{aligned}$$

Algoritmo Estendido de Euclides

$$mdc\ (973, 301) = 7 = 973 \times ? + 301 \times ?$$

$$\begin{aligned} 70 &= 973 + 301 \times (-3) \\ 21 &= 301 + 70 \times (-4) \\ 7 &= 70 + 21 \times (-3) \end{aligned}$$

Algoritmo Estendido de Euclides

$$mdc\,(973, 301) = 7 = 973\times ? + 301\times ?$$

$$\begin{aligned} 70 &= 973 + 301 \times (-3) \\ 21 &= 301 + 70 \times (-4) \\ 7 &= 70 + 21 \times (-3) \end{aligned}$$

Algoritmo Estendido de Euclides

$$mdc\,(973, 301) = 7 = 973{\times}? + 301{\times}?$$

$$\begin{array}{rcl}
70 & = & 973 + 301 \times (-3) \\
21 & = & 301 + 70 \times (-4) \\
7 & = & 70 + 21 \times (-3) \\
\hline
7 & = & 70 + (301 + 70 \times (-4)) \times (-3)
\end{array}$$

Algoritmo Estendido de Euclides

$$mdc\ (973, 301) = 7 = 973 \times ? + 301 \times ?$$

$$\begin{array}{rcl}
70 & = & 973 + 301 \times (-3) \\
21 & = & 301 + 70 \times (-4) \\
7 & = & 70 + 21 \times (-3) \\
\hline
7 & = & 70 + (301 + 70 \times (-4)) \times (-3) \\
7 & = & 301 \times (-3) + 70 \times (13)
\end{array}$$

Algoritmo Estendido de Euclides

$$mdc\,(973, 301) = 7 = 973 \times ? + 301 \times ?$$

$$\begin{aligned}
70 &= 973 + 301 \times (-3) \\
21 &= 301 + 70 \times (-4) \\
7 &= 70 + 21 \times (-3) \\
\hline
7 &= 70 + (301 + 70 \times (-4)) \times (-3) \\
7 &= 301 \times (-3) + 70 \times (13)
\end{aligned}$$

Algoritmo Estendido de Euclides

$$mdc\,(973, 301) = 7 = 973\times? + 301\times?$$

$$\begin{aligned}
{\color{blue}70} &= 973 + 301 \times (-3) \\
21 &= 301 + 70 \times (-4) \\
7 &= 70 + 21 \times (-3) \\
\hline
7 &= 70 + (301 + 70 \times (-4)) \times (-3) \\
7 &= 301 \times (-3) + {\color{blue}70} \times (13)
\end{aligned}$$

Algoritmo Estendido de Euclides

$$mdc\,(973, 301) = 7 = 973\times? + 301\times?$$

$$
\begin{aligned}
{\color{blue}70} &= 973 + 301 \times (-3)\\
21 &= 301 + 70 \times (-4)\\
7 &= 70 + 21 \times (-3)\\
\hline
7 &= 70 + (301 + 70 \times (-4)) \times (-3)\\
7 &= 301 \times (-3) + ({\color{blue}973 + 301 \times (-3)}) \times (13)
\end{aligned}
$$

Algoritmo Estendido de Euclides

$$mdc\,(973, 301) = 7 = 973\times ? + 301\times ?$$

$$\begin{aligned}
70 &= 973 + 301 \times (-3) \\
21 &= 301 + 70 \times (-4) \\
7 &= 70 + 21 \times (-3)
\end{aligned}$$

$$\begin{aligned}
7 &= 70 + (301 + 70 \times (-4)) \times (-3) \\
7 &= 301 \times (-3) + (973 + 301 \times (-3)) \times (13) \\
7 &= 973 \times (13) + 301 \times (-42)
\end{aligned}$$

Algoritmo Estendido de Euclides

$$mdc\ (973, 301) = 7 = 973\times ? + 301\times ?$$

$$\begin{array}{rcl}
70 & = & 973 + 301 \times (-3) \\
21 & = & 301 + 70 \times (-4) \\
7 & = & 70 + 21 \times (-3) \\
\hline
7 & = & 70 + (301 + 70 \times (-4)) \times (-3) \\
7 & = & 301 \times (-3) + (973 + 301 \times (-3)) \times (13) \\
7 & = & 973 \times \color{red}{(13)} + 301 \times \color{red}{(-42)}
\end{array}$$

Algoritmo Estendido de Euclides

Algoritmo Estendido de Euclides

Euclides-estendido $(a, b, *x, *y)$

Se $(b = 0)$

$*x \leftarrow 1; \quad *y \leftarrow 0;$

Retorne a;

Senão

d $=$ Euclides-estendido $(b,\ a \bmod b,\ x,\ y)$;

$\hat{x} \leftarrow *x; \quad \hat{y} \leftarrow *y;$

$*x \leftarrow \hat{y};$

$*y \leftarrow \hat{x} - \hat{y} \times (a/b);$

Retorne d;

Algoritmo Estendido de Euclides

O algoritmo estendido de Euclides permite ainda calcular inversos modulares

$$d = e^{-1} \bmod n$$

Note nesse caso que $ex + ny = 1$ e que o inverso d é o valor de x.

Algoritmo Estendido de Euclides

Calcular $d = 3^{-1} \bmod 20$ $\quad (e^{-1} \bmod n)$

$ex + ny = 1 \qquad 3x + 20y = 1$

Passos:

1) Euclides-estendido $(n, e, \&y, \&x)$

2) Resultado: $y = -1$ e $x = 7$.

3) Resultado: $1 = -1 \times 20 + 7 \times 3$.

4) Inverso $= 7$.

Se o inverso for negativo some n ao resultado.

Algoritmo Estendido de Euclides

Calcular $d = 3^{-1} \bmod 20$ $\qquad (e^{-1} \bmod n)$

$ex + ny = 1 \qquad 3x + 20y = 1$

Passos:

1) Euclides-estendido $(20, 3, \&y, \&x)$

2) Resultado: $y = -1$ e $x = 7$.

3) Resultado: $1 = -1 \times 20 + 7 \times 3$.

4) Inverso $= 7$.

Se o inverso for negativo some n ao resultado.

Sumário

Seja o processo de cifragem e decifragem

$$\mathbf{y} = \mathbf{x}^{e} \bmod \mathbf{n} \qquad \mathbf{x} = \mathbf{y}^{\mathbf{d}} \bmod \mathbf{n}$$

A espiã **Eva** conhece quais informações?

Seja o processo de cifragem e decifragem

$$\mathbf{y} = \mathbf{x}^{e} \bmod \mathbf{n} \qquad \mathbf{x} = \mathbf{y}^{\mathbf{d}} \bmod \mathbf{n}$$

A espiã **Eva** conhece as seguintes informações:

- **y** (mensagem cifrada)
- **e** (expoente público)
- **n** (produto primos $p \times q$ desconhecidos).

Questão: como recuperar **d**?

Ataque:

- Fatore $\mathbf{n} = p \times q$
- Calcule $\phi(\mathbf{n}) = (p - 1) \times (q - 1)$
- Calcule $\mathbf{d} = e^{-1} \bmod \phi(\mathbf{n})$

Sabemos pelo teorema fundamental da aritmética que só existe uma combinação de p e q que retorna em $\mathbf{n}$. O cálculo $\mathbf{d} = e^{-1} \bmod \phi(\mathbf{n})$ é trivial para o algoritmo de Euclides estendido (módulo inverso visto em aula).

Fatoração

Observe que se um número n for composto, então ele pode ser escrito por $n = a \times b$. Note também que $a \leq b$ ou $b \leq a$, suponha que $a \leq b$ então

$$a \leq b \Rightarrow a^2 \leq a \times b \Rightarrow a^2 \leq n \Rightarrow a \leq \sqrt{n}$$

Em resumo, o algoritmo deve buscar um número que divida n, começando de 2 e avançando até $\sqrt{n}$.

Segurança

Um algoritmo simples, conhecido como **trial divison**, consiste em tentar dividir $\mathbf{n}$ por cada inteiro no intervalo $2, 3, \ldots, \sqrt{\mathbf{n}}$. A complexidade do algoritmo é $\mathcal{O}(\sqrt{\mathbf{n}})$. Note no entanto que um inteiro n é representado por $\beta = \lceil \log_2(n+1) \rceil$ bits e

$$2^{\beta} \leq \mathbf{n} < \; 2^{\beta+1}$$

e desta forma $\sqrt{2^{\beta}} = 2^{\beta/2} \leq \sqrt{\mathbf{n}}$ (algoritmo exponencial no número de bits).

Sumário

Performance

Um aspecto interessante do RSA é que a escolha de um expoente público **e** pequeno, por exemplo $\mathbf{e = 3}$, não afeta a segurança da cifragem. Na prática três valores são de particular importância (o expoente privado de forma geral é grande):

Public key e	e as binary string	#MUL + #SQ
3	11_2	3
17	10001_2	5
$2^{16}+1$	$1\,0000\,0000\,0000\,0001_2$	17

Performance

Nível de segurança de n bits $= 2^n$ tentativas para o melhor ataque conhecido.

Algorithm Family	Cryptosystems	Security Level (bit)			
		80	128	192	256
Integer factorization	RSA	1024 bit	3072 bit	7680 bit	15360 bit
Discrete logarithm	DH, Elgamal	1024 bit	3072 bit	7680 bit	15360 bit
Elliptic curves	ECDH, ECDSA	160 bit	256 bit	384 bit	512 bit
Symmetric-key	AES, 3DES	80 bit	128 bit	192 bit	256 bit

O RSA é 2 ou 3 ordens de magnitude mais lento que o 3DES/AES. Um aumento no tamanho da chave de 1024 para 3076 bits deixa o processo de cifragem $27\times$ mais lento.

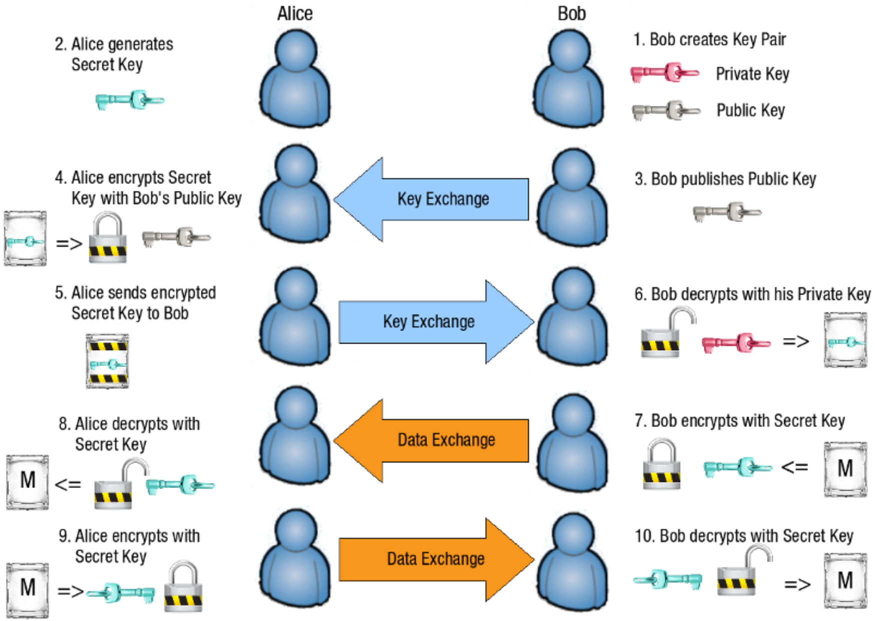
Considerações finais

Here is the complexity for the GNFS (pulled from the linked Wikipedia article):

$$\exp\left(\left(\sqrt[3]{\frac{64}{9}}+o(1)\right)(\ln n)^{\frac{1}{3}}(\ln\ln n)^{\frac{2}{3}}\right)$$

where n is a number to factor. Evaluating the above expression at 2^b is a rough approximation of the the time needed to factor a b-bit integer. Here's a table showing the bit-length of the evaluation at $2^{1024}, 2^{2048}, \ldots$:

```
Strength  RSA modulus size    Complexity bit-length
  80        1024                86.76611925028119
 112        2048                116.8838132958159
 128        3072                138.7362808527251
 192        7680                203.01873594417665
 256       15360                269.38477262128384
```



Alice
Bob
1. Bob creates Key Pair
Private Key
Public Key
2. Alice generates Secret Key
3. Bob publishes Public Key
4. Alice encrypts Secret Key with Bob's Public Key
=>
Key Exchange
5. Alice sends encrypted Secret Key to Bob
Key Exchange
6. Bob decrypts with his Private Key
=>
7. Bob encrypts with Secret Key
<=
M
Data Exchange
8. Alice decrypts with Secret Key
M
<=
9. Alice encrypts with Secret Key
M
=>
Data Exchange
10. Bob decrypts with Secret Key
=>
M

Sumário

Prova de corretude

A função $\phi(m)$ de Euler representa o número de inteiros em $\mathbb{Z}_m = \{0, 1, 2, \ldots, m-1\}$ que são coprimos a m (**mdc** $= 1$). Exemplo: $\phi(6) = 2$

$$\begin{aligned}
\text{mdc }(0{,}6) &= 6 \\
\text{mdc }(1{,}6) &= 1 \star \\
\text{mdc }(2{,}6) &= 2 \\
\text{mdc }(3{,}6) &= 3 \\
\text{mdc }(4{,}6) &= 2 \\
\text{mdc }(5{,}6) &= 1 \star
\end{aligned}$$

Prova de corretude

Note que $\phi(p) = p - 1$ para qualquer número primo p. Exemplos:

mdc $(0,5) = 5$	mdc $(0,7) = 7$
mdc $(1,5) = 1 \star$	mdc $(1,7) = 1 \star$
mdc $(2,5) = 1 \star$	mdc $(2,7) = 1 \star$
mdc $(3,5) = 1 \star$	mdc $(3,7) = 1 \star$
mdc $(4,5) = 1 \star$	mdc $(4,7) = 1 \star$
	mdc $(5,7) = 1 \star$
	mdc $(6,7) = 1 \star$

Teorema e propriedades da função de Euler:

Teorema: Se m é um inteiro positivo e a é um inteiro positivo co-primo de m então:

$$a^{\phi(m)} \equiv 1 \bmod m$$

Generalização do teorema de Fermat :

$$a^{p-1} \equiv 1 \bmod p$$

Propriedades:

1. $\phi(1) = 1$ pois $\mathrm{mdc}(0{,}1) = 1$.
2. $\phi(xy) = \phi(x) \times \phi(y)$

Passo 3 do RSA temos que $\phi(n) = \phi(p \times q)$

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

Prova de corretude

No passo 4 do RSA (geração de chaves) temos:

MDC-EUCLIDES-ESTENDIDO($\phi(n)$, e, a, $\mathbf{b}$)=1

Lembre-se da aula de MDC (a, $\mathbf{b}$ certificados):

$$\begin{aligned} \phi(n) \times a \ + \ e \times \mathbf{b} = 1 \\ e \times \mathbf{b} \ = \ 1 - \phi(n) \times a \end{aligned}$$

Aplicando-se mod $\phi(\mathbf{n})$ na equação

$$\begin{aligned} e \times \mathbf{b} \bmod \phi(n) &= 1 \\ e \times \mathbf{b} &\equiv 1 \bmod \phi(n) \\ \mathbf{b} &\equiv e^{-1} \bmod \phi(n) \end{aligned}$$

Como $\mathbf{d} = \mathbf{b}$, então $e \times d \equiv 1 \bmod \phi(n)$

Prova de corretude

Seja o processo de cifragem e decifragem

$$\mathbf{y} = \mathbf{x}^{e} \bmod \mathbf{n} \qquad\qquad \mathbf{x} = \mathbf{y}^{\mathbf{d}} \bmod \mathbf{n}$$

Mostre que $\mathbf{x}^{e \times \mathbf{d}} \text{mod } \mathbf{n} = \mathbf{x}$

Prova de corretude

Seja o processo de cifragem e decifragem

$$\mathbf{y} = \mathbf{x}^{e} \bmod \mathbf{n} \qquad \mathbf{x} = \mathbf{y}^{\mathbf{d}} \bmod \mathbf{n}$$

Mostre que $\mathbf{x}^{e \times \mathbf{d}} \bmod \mathbf{n} = \mathbf{x}$

$$\mathbf{x}^{e \times \mathbf{d}} \bmod \mathbf{n}$$

Prova de corretude

Seja o processo de cifragem e decifragem

$$\mathbf{y} = \mathbf{x}^{e} \bmod \mathbf{n} \qquad \mathbf{x} = \mathbf{y}^{\mathbf{d}} \bmod \mathbf{n}$$

Mostre que $\mathbf{x}^{e \times \mathbf{d}} \text{mod } \mathbf{n} = \mathbf{x}$

$$\mathbf{x}^{e \times \mathbf{d}} \bmod \mathbf{n}$$

seja $e \times \mathbf{d} = 1 + k\phi(n)$

Prova de corretude

Seja o processo de cifragem e decifragem

$$\mathbf{y} = \mathbf{x}^{e} \bmod \mathbf{n} \qquad \mathbf{x} = \mathbf{y}^{\mathbf{d}} \bmod \mathbf{n}$$

Mostre que $\mathbf{x}^{e \times \mathbf{d}} \mathrm{mod}\ \mathbf{n} = \mathbf{x}$

$$\mathbf{x}^{e \times \mathbf{d}} \bmod \mathbf{n}$$

$$\mathbf{x}^{1+k\phi(n)} \bmod \mathbf{n}$$

Prova de corretude

Seja o processo de cifragem e decifragem

$$\mathbf{y} = \mathbf{x}^{e} \bmod \mathbf{n} \qquad \mathbf{x} = \mathbf{y}^{\mathbf{d}} \bmod \mathbf{n}$$

Mostre que $\mathbf{x}^{e \times \mathbf{d}} \bmod \mathbf{n} = \mathbf{x}$

$$\mathbf{x}^{e \times \mathbf{d}} \bmod \mathbf{n}$$

$$\mathbf{x}^{1 + k\phi(n)} \bmod \mathbf{n}$$

$$\mathbf{x}^{1} (\mathbf{x}^{\phi(n)})^{k} \bmod n$$

Prova de corretude

Seja o processo de cifragem e decifragem

$$\mathbf{y} = \mathbf{x}^{e} \bmod \mathbf{n} \qquad \mathbf{x} = \mathbf{y}^{\mathbf{d}} \bmod \mathbf{n}$$

Mostre que $\mathbf{x}^{e \times \mathbf{d}} \mathrm{mod}\ \mathbf{n} = \mathbf{x}$

$$\mathbf{x}^{e \times \mathbf{d}} \bmod \mathbf{n}$$

$$\mathbf{x}^{1+k\phi(n)} \bmod \mathbf{n}$$

$$\mathbf{x}^{1}(\mathbf{x}^{\phi(n)})^{k} \bmod \mathbf{n}$$

seja $a^{\phi(n)} \equiv 1 \text{ mod } n$ (Teorema de Euler).

Prova de corretude

Seja o processo de cifragem e decifragem

$$\mathbf{y} = \mathbf{x}^{e} \mod \mathbf{n} \qquad \mathbf{x} = \mathbf{y}^{\mathbf{d}} \mod \mathbf{n}$$

Mostre que $\mathbf{x}^{e\times\mathbf{d}} \bmod \mathbf{n} = \mathbf{x}$

$$\mathbf{x}^{e\times\mathbf{d}} \mod \mathbf{n}$$
$$\mathbf{x}^{1+k\phi(n)} \mod \mathbf{n}$$
$$\mathbf{x}^{1}(\mathbf{x}^{\phi(n)})^{k} \mod \mathbf{n}$$
$$\mathbf{x}^{1}(1)^{k} \mod \mathbf{n}$$

Prova de corretude

Seja o processo de cifragem e decifragem

$$\mathbf{y} = \mathbf{x}^{e} \bmod \mathbf{n} \qquad \mathbf{x} = \mathbf{y}^{\mathbf{d}} \bmod \mathbf{n}$$

Mostre que $\mathbf{x}^{e\times \mathbf{d}} \text{mod } \mathbf{n} = \mathbf{x}$

$$\mathbf{x}^{e\times \mathbf{d}} \bmod \mathbf{n}$$
$$\mathbf{x}^{1+k\phi(n)} \bmod \mathbf{n}$$
$$\mathbf{x}^{1}(\mathbf{x}^{\phi(n)})^{k} \bmod \mathbf{n}$$
$$\mathbf{x}^{1}(1)^{k} \bmod n$$
$$\mathbf{x} \bmod \mathbf{n} = \mathbf{x}$$