OBJETIVOS

- **Criptografia de chaves públicas ou assimétricas (algoritmo RSA).**

CRIPTOGRAFIA DE CHAVE SIMÉTRICA X ASSIMÉTRICA

Algoritmo de Chave Simétrica	Algoritmo de Chave Assimétrica
Compartilha da mesma chave (criptografia e decriptografia)	Usa um par de chaves (criptografia e decriptografia)
Tamanho usual da chave (de 56 a 256 bits)	Tamanho usual da chave (de 512 a 4096 bits)
Emissor e receptor compartilham o algoritmo e a chave	Emissor e receptor precisam ter, cada um, uma chave do par (não a mesma)
Os algoritmos são geralmente mais rápidos, pois são baseados em operações matemáticas simples	Os algoritmos são relativamente mais lentos, pois são baseados em algoritmos computacionais mais difíceis
Exemplos: DES, 3DES, AES, IDEA, RC4/5/6 e Blowfish	Exemplos: RSA, Elgamal, curva elíptica, Diffie-Helmann.

CRIPTOGRAFIA DE CHAVE ASSIMÉTRICA

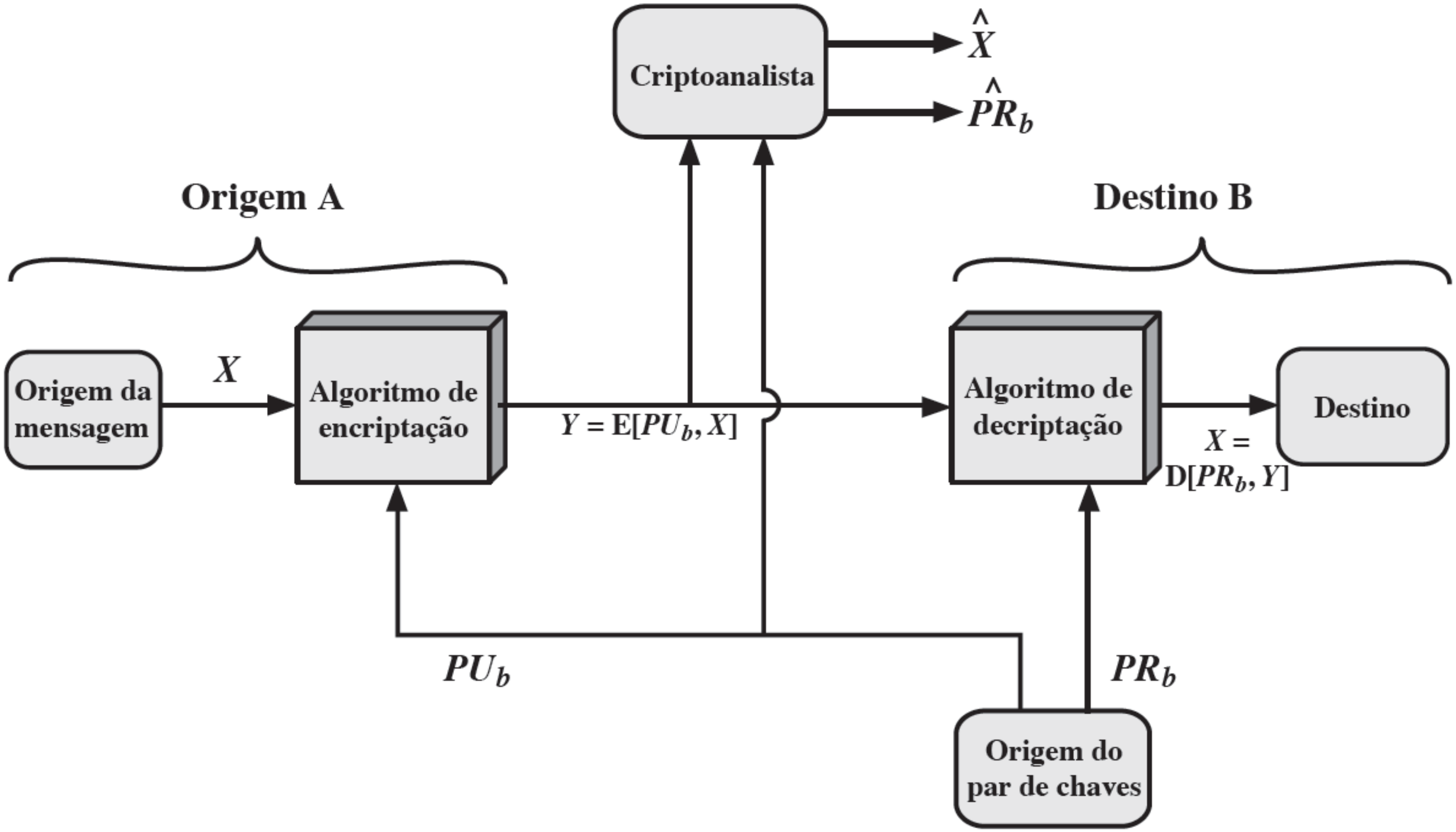
- Usa uma chave para criptografar diferente da usada para decriptografar. Alguns exemplos de algoritmos assimétricos:
  - **RSA (Rivest_Shamir-Adleman)** - uso o produto de 2 números primos grandes com um tamanho entre 100 e 200 digitos. Os browsers usam o RSA para estabelecimento de conexão segura.
  - **Diffie-Hellman** - prove de um método de troca de chave secreta compartilhada. Protocolos seguros, tais como SSL, TLS, SSH, Ipsec usam o Diffie-Hellman.
  - **ElGamal** - padrão do governo dos USA para assinaturas digitais.
  - **Elliptic Curve Cryptography (ECC)** - usa curva elíptica como parte do algoritmo. Nos USA, a NSA (National Security Agency) usa o ECC para troca de chaves e assinaturas digitais.

CRIPTOGRAFIA DE CHAVE ASSIMÉTRICA OU PÚBLICA

ALGORITMO	ENCRIPTAÇÃO/ DECRIPTAÇÃO	ASSINATURA DIGITAL	TROCA DE CHAVE
RSA	Sim	Sim	Sim
Curva elíptica	Sim	Sim	Sim
Diffie-Hellman	Não	Não	Sim
DSS	Não	Sim	Não

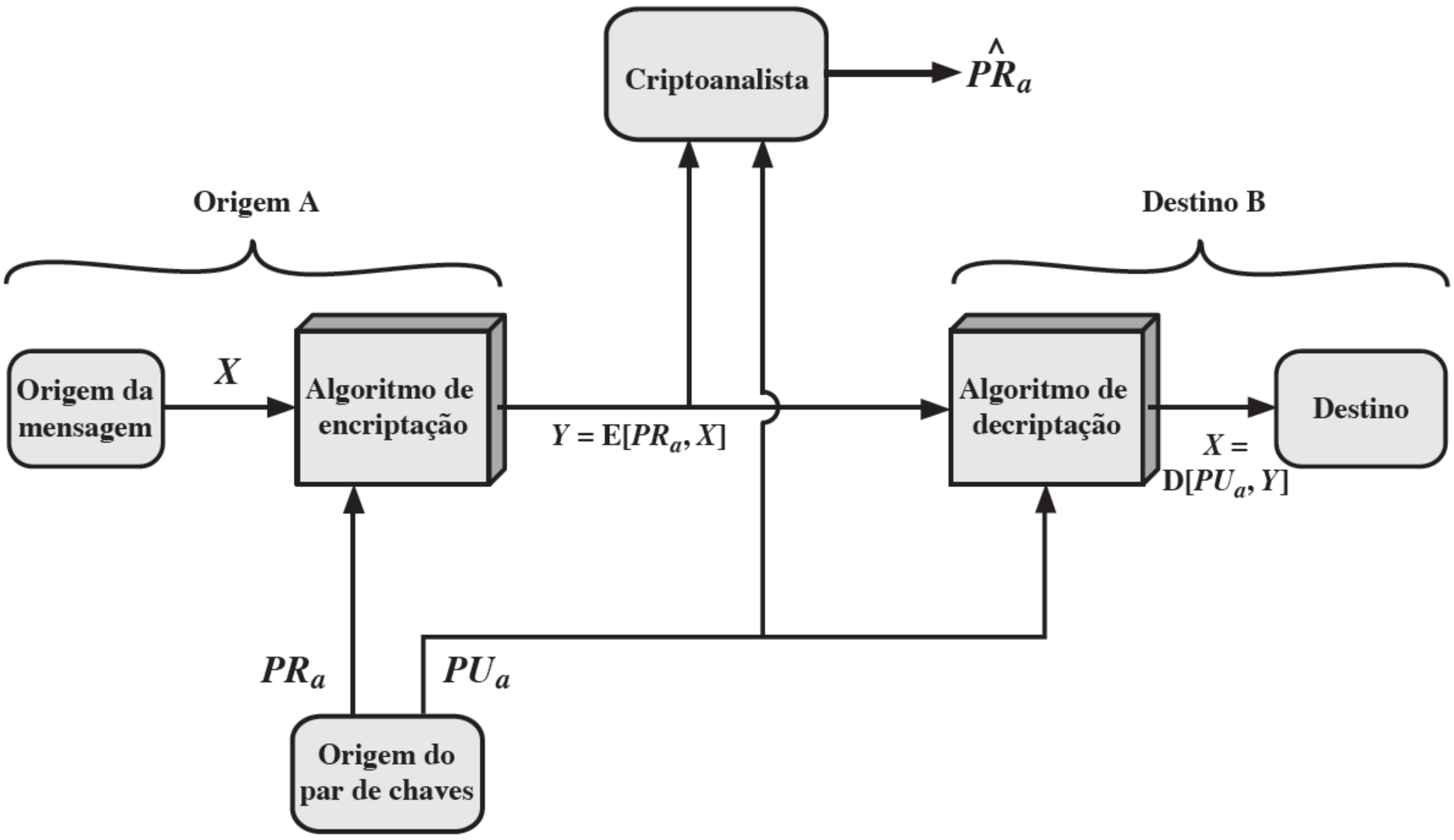
CRIPTOGRAFIA DE CHAVE ASSIMÉTRICA OU PÚBLICA

- Criptossistema de chave pública – sigilo:



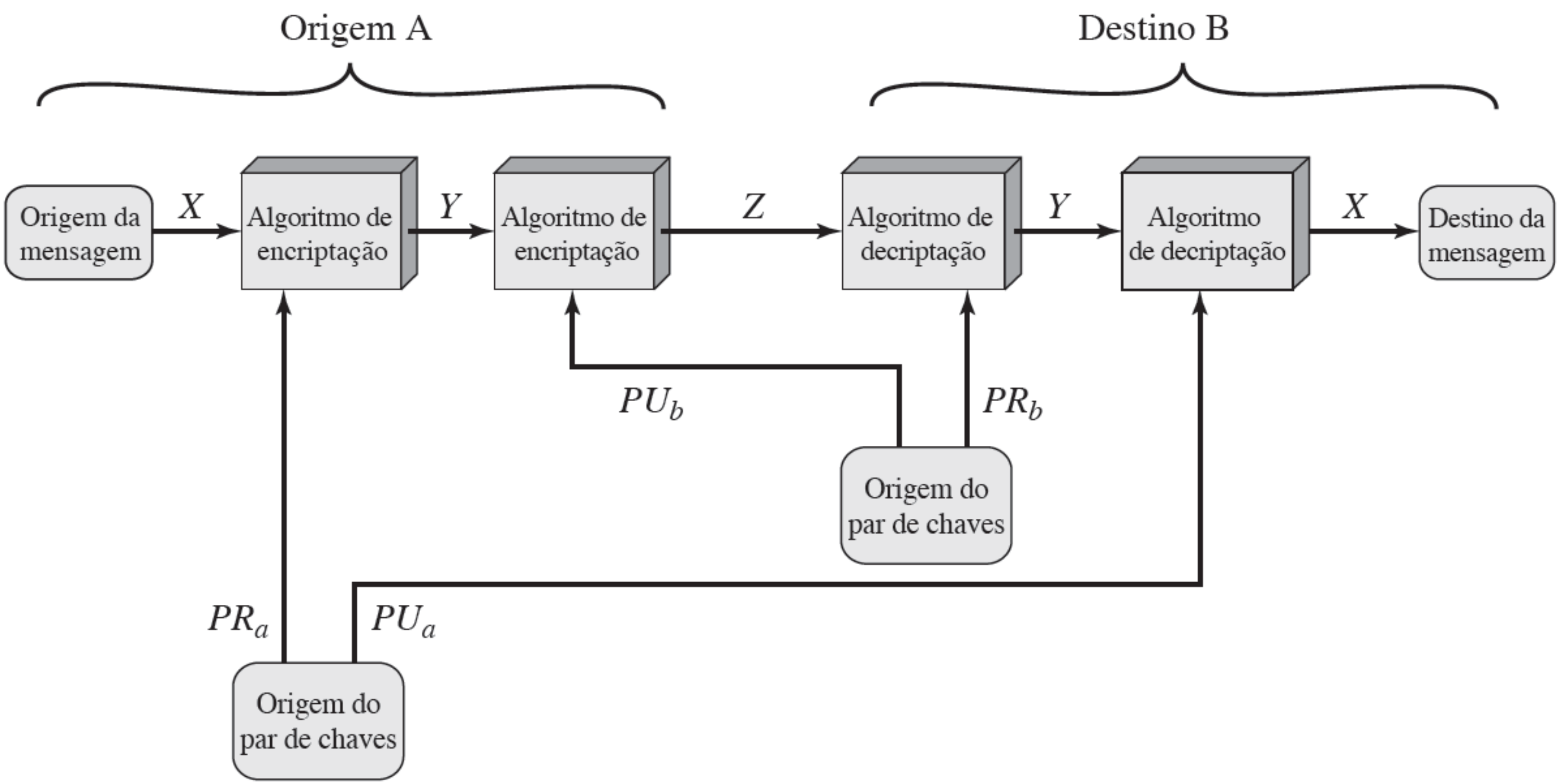
CRIPTOGRAFIA DE CHAVE ASSIMÉTRICA OU PÚBLICA

- Criptossistema de chave pública – autenticação:



CRIPTOGRAFIA DE CHAVE ASSIMÉTRICA OU PÚBLICA

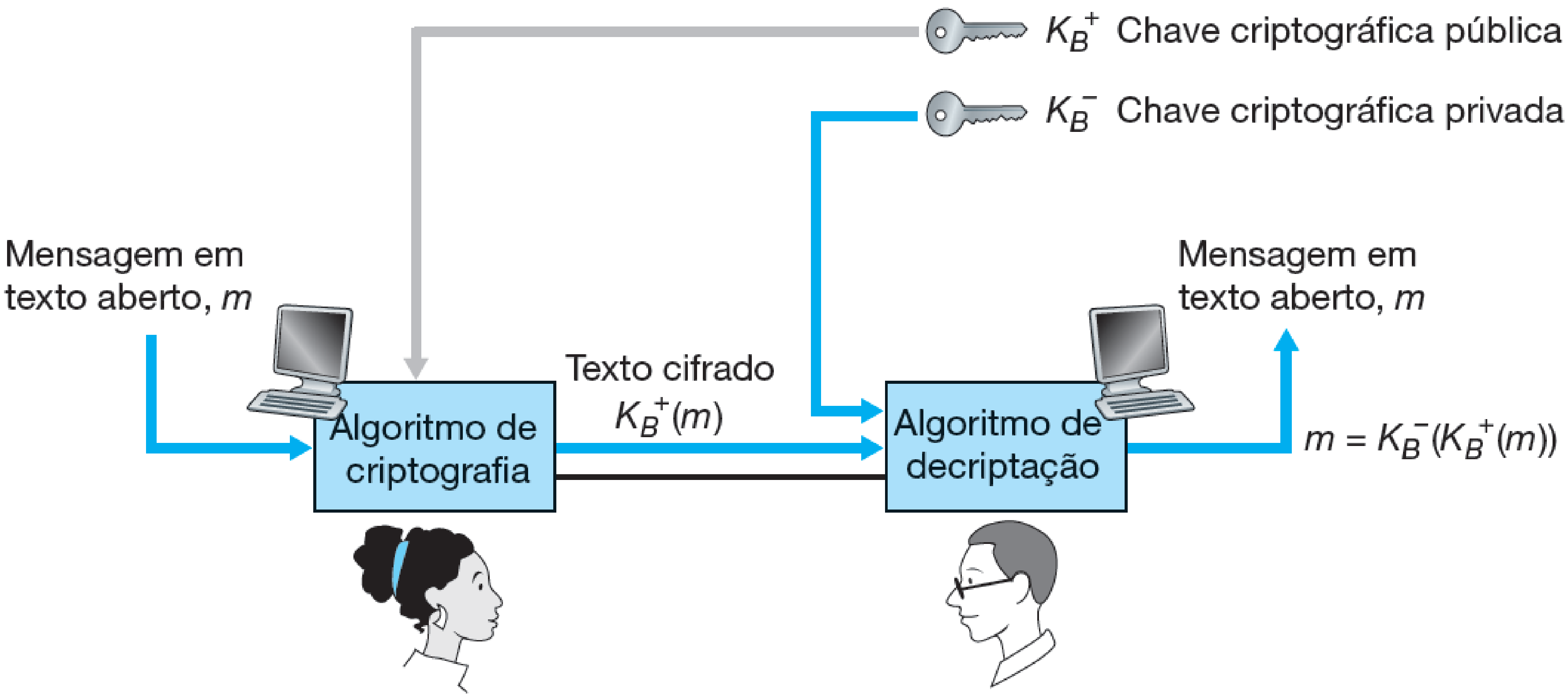
- Criptossistema de chave pública – autenticação e sigilo:



CRIPTOGRAFIA ASSIMÉTRICA

- Características:
  - Possuem duas chaves;
  - Uma chave criptografa (chamada ***chave pública***) enquanto que a outra decriptografa (chamada ***chave privada***),ou vice-versa;
  - Funções cuja inversa são de complexidade computacional elevada;
  - Chaves assimétricas são muito mais longas (512 à 1024 bits);
  - Possui um método de obtenção própria para as chaves (uma delas é aleatória enquanto a outra é calculada em função da primeira);
  - A obtenção da chave privada é secreta e ninguém além do dono precisa conhece-la;
  - A chave pública é obtida a partir da chave privada;
  - A chave pública pode ser distribuída sem a necessidade de sigilo.

CRIPTOGRAFIA ASSIMÉTRICA



ALGORITMO RSA

- O Algoritmo RSA ("Rivest, Shamir, Adleman") foi descoberto em 1978 no MIT
  - O método se baseia em alguns princípios da teoria dos números (Fatoração e Logaritmo Discreto)
  - A segurança do método se baseia na dificuldade de fatorar números extensos. Um número de 200 dígitos requer 4 bilhões de anos, supondo que o tempo de processamento de cada instrução é de 1 microsegundo

ALGORITMO RSA

- Problema da fatoração
  - Dados dois números primos p e q:
    - Obter n = p.q é bastante simples.
  - Dados um número n, produto de dois números primos p e q:
    - Obter p e q é bastante complexo.
  - Produto fácil e fatoração difícil.
  - Exemplo 1
    - p = 120.899 q = 136.739
    - n = p*q = ?
    - n = 16.531.608.361

ALGORITMO RSA

- Problema da fatoração
  - Exemplo 2
    - n = 11.916.442.787
    - p = ? q = ?
    - p = 117.319 q = 101.573

ALGORITMO RSA

- O esquema RSA é uma cifra de bloco em que o texto claro e o cifrado são inteiros entre 0 e $n - 1$, para algum n.
- Um tamanho típico para n é 1024 bits, ou 309 dígitos decimais.
- Ou seja, n é menor que 2^{1024}.
- RSA utiliza uma expressão com exponenciais.
- O texto claro é encriptado em blocos, com cada um tendo um valor binário menor que algum número n.

ALGORITMO RSA

- **Geração de chaves:**

1. **Encontre dois números primos grandes p, q.**
   (ex., 1024 bits cada um)

2. **Calcule $n = pq$, $z = (p-1)(q-1)$**

3. **Escolha e *(com $e<n$)* que não tem fatores primos em comum com z. (e, z são "primos entre si").**

4. **Escolha d tal que $ed-1$ é exatamente divisível por z.**
   (em outras palavras: $ed \bmod z = 1$).

5. **Chave *Pública* é (n,e). Chave *Privada* é (n,d).**

RSA: CRIPTOGRAFIA E DECRIPTOGRAFIA

- **Criptografia por Bob usando com a chave pública de Alice:**

0. Dado (n,e) e (n,d) como calculado anteriormente

1. Para criptografar o padrão de bits, m, calcule

$$c = m^e \bmod n$$ **(i.e., resto quando m^e é dividido por n)**

- **Decriptografia por Alice usando com a chave privada de Alice:**

2. Para decriptografar o padrão de bits recebidos, c, calcule

$$m = c^d \bmod n$$ **(i.e., resto quando c^d é dividido n)**

RSA: CRIPTOGRAFIA E DECRIPTOGRAFIA

- Tanto encriptação quanto decriptação no RSA envolvem elevar um inteiro a uma potência inteira, mod *n*.
- Se a exponenciação fosse feita sobre os inteiros e depois reduzida módulo *n*, os valores intermediários seriam gigantescos.
- Felizmente, podemos utilizar uma propriedade da aritmética modular:

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$$

RSA: CRIPTOGRAFIA E DECRIPTOGRAFIA

- Para agilizar a operação do algoritmo RSA usando a chave pública, normalmente é feita uma escolha específica de *e*.
- A mais comum é 65537 ($2^{16} + 1$); duas outras escolhas populares são 3 e 17.
- Cada uma delas tem apenas dois bits 1, e, por isso, o número de multiplicações exigidas para realizar a exponenciação é minimizado.
- Porém, com uma chave pública muito pequena, como $e = 3$, o RSA torna-se vulnerável a um ataque simples.

RSA: EXEMPLO

- **Bob escolhe** ***p=5, q=7.***
  - **n=pxq=5x7=35**
  - **z=(p-1)x(q-1)=4x6=24**
    - **24 | 2**
    - **12 | 2**
    - **6 | 2**
    - **3 | 3**
    - **1**
    - **e = 5 (*e, z* são primos entre si).**

- ***e=5*** **(assim *e, z* são primos entre si).**
- **Chave pública = (n,e) = (35,5)**

RSA: EXEMPLO

- **Escolha de d (método de tentativa)**
  - ***ed*****mod*****z*****=1**
  - **(5x1)mod24=5**
  - **(5x2)mod24=10**
  - **(5x3)mod24=15**
  - **(5x4)mod24=20**
  - **(5x5)mod24=1**
  - **(5x29)mod24=1**
- **Escolha: *d=5* ou *d=29* (assim *ed-1* é exatamente divisível por z).**
- **Chave privada: (n,d) = (35,5) ou (35,29)**

RSA: EXEMPLO

	letra	m	m^e	$c = m^e \bmod n$
criptografia:	l	12	248832	17

	c	c^d	$m = c^d \bmod n$	letra
decriptografia:	17	481968572106750915091411825223072000	12	l

ALGORITMO DE EUCLIDES ESTENDIDO

- O **Euclides Estendido** é uma das formas de se encontrar o (MDC) de dois números inteiros. Nele, ao invés de retornar um valor único, fornece a combinação linear, muito útil quando os inteiros são primos entre si.
- **MDC(120,23)** = 1
- **Onde: 120** e **23** são inteiros primos entre si porque não existe um divisor maior do que 1 que divida ambos. O algoritmo de Euclides estendido retorna:
- **ax + by = MDC(a, b)** , ou seja:
- **MDC(120,23)** = 120 * (− 9) + 23 * 47

ALGORITMO DE EUCLIDES ESTENDIDO

- Entendendo o algoritmo de **Euclides Estendido:**
- Para encontrar o MDC(120,23), coloca-se da seguinte forma:

```
(1)     120÷23 = 5 resta 5
(2)     23÷5 = 4 resta 3
(3)     5÷3 = 1 resta 2
(4)     3÷2 = 1 resta 1
(5)     2÷1 = 2 resta 0
```

- **MDC(120,23)** = 1
- Levando-se em conta apenas os restos encontrados, pode-se dizer que:

```
(1)     5 = 1*120 - 5*23
(2)     3 = 1*23 - 4*5    Substituindo o 5 temos
        3 = 1*23 - 4*(1*120 - 5*23)
        3 = -4*120 + 21*23
(3)     2 = 1*5 - 1*3     Substituindo o valor de 5 e 3 temos
        2 = 1(1*120 - 5*23) - 1(-4*120 + 21*23)
        2 = 5*120 - 26*23
(4)     1 = 1*3 - 1*2    Novamente substituindo 3 e 2
        1 = 1(-4*120 + 21*23) - 1(5*120 - 26*23)
        1 = -9*120 + 47*23
```

ALGORITMO DE EUCLIDES ESTENDIDO

- Portanto, **x = -9** e **y = 47** e temos:
- **MDC(120,23)** $= 120 * (-9) + 47 * 23$

RSA: EXEMPLO

- Seja dois números primos p=19 e q=23
- Cálculo da chave pública

```
N = p x q
N = 19 x 23
N = 437

(p-1) x (q-1) = 18 x 22
(p-1) x (q-1) = 396


396 | 2
198 | 2
 99 | 3
 33 | 3
 11 | 11
  1 |

ou seja, 396 = 2 x 2 x 3 x 3 x 11
```

- N = 437 e e = 13 são a chave pública.

RSA: EXEMPLO

- Cálculo da chave privada

```
(1)    13 ÷ 396 =  0 com resto 13   ...   divide-se o valor pelo módulo
(2)   396 ÷  13 = 30 com resto  6   ...   divide-se o divisor anterior pelo resto
(3)    13 ÷   6 =  2 com resto  1   ...   divide-se o divisor anterior pelo resto
        6 ÷   1 =  6 com resto  0   ...   divide-se o divisor anterior pelo resto
```

- Euclides estendido

```
(1)   13 = (1 x 13) - (0 x 396)
      13 = (1 x 13)

(2)    6 = (1 x 396) - (30 x 13)         ...   e como (1) nos diz que 13 = (1 x 13)
       6 = (1 x 396) - (30 x (1 x 13))
       6 = (1 x 396) - (30 x 13)

(3)    1 = (1 x 13) - (2 x 6)            ...   e como (2) nos diz que 6 = (1 x 396) - (30 x 13)
       1 = (1 x 13) - 2 x ((1 x 396) - (30 x 13))
       1 = (1 x 13) - (2 x 396) + (60 x 13)
       1 = (61 x 13) - (2 x 396)
```

SEGURANÇA DO RSA

- Técnicas possíveis para atacar o algoritmo RSA são as seguintes:
  - **Força bruta**: isso envolve tentar todas as chaves privadas possíveis.
  - **Ataques matemáticos**: existem várias técnicas, todas equivalentes em esforço a fatorar o produto de dois primos.
  - **Ataques de temporização**: estes dependem do tempo de execução do algoritmo de decriptação.
  - **Ataques baseados em falha de hardware**: estes envolvem a indução de falhas de hardware no processador que está gerando as assinaturas digitais.
  - **Ataques de texto cifrado escolhido**: esse tipo de ataque explora as propriedades do algoritmo RSA.

SEGURANÇA DO RSA

- A defesa contra a técnica de força bruta é a mesma para o RSA e para outros criptossistemas, ou seja, usar um espaço de chave grande.
- Assim, quanto maior o número de bits em d, melhor.

BIBLIOGRAFIA

- **Bibliografia:**

  - STALLINGS, W. Criptografia e Segurança de Redes - Princípios e Práticas - 6ed., Pearson, 2015.
  - KUROSE, James F; ROSS, Keith W. Redes de computadores e a internet: uma abordagem Top-Down. 6. ed. São Paulo: Pearson, c2014.
  - Popyack, J.L. RSA Calculator. Disponível em: <https://www.cs.drexel.edu/~jpopyack/IntroCS/HW/RSAWorksheet.html>. Acesso em: 16.08.2024.
  - Notas de aula.