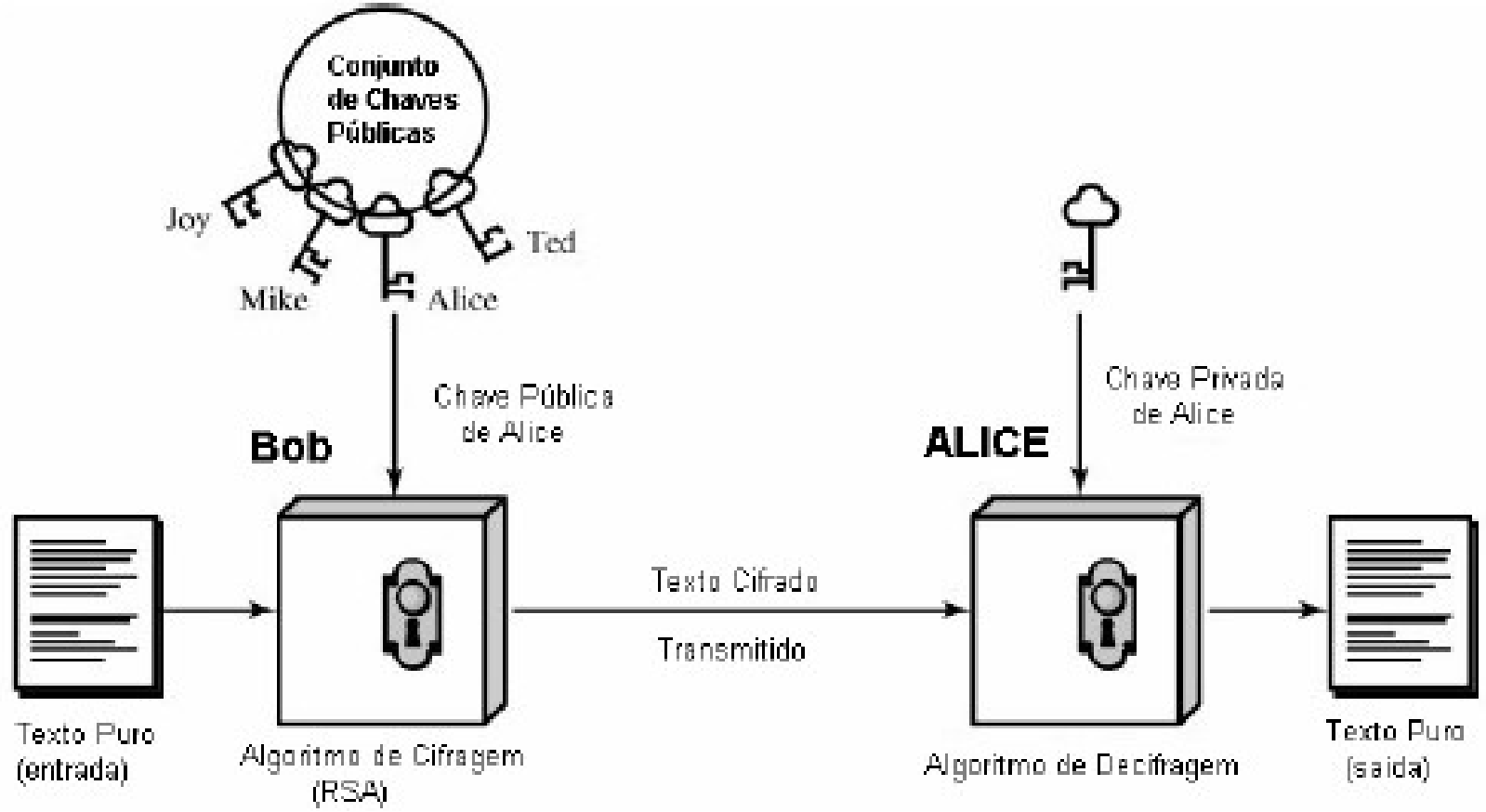
Criptografia Assimétrica

- Cada entidade que se comunica possui um par de chaves que são utilizadas no processo de cifragem e decifragem das mensagens
- Uma das chaves é pública e desta forma é conhecida por todas as entidades
- A outra é privada e somente é conhecida por uma das partes
- Quando a chave pública é utilizada para cifrar a mensagem e a chave privada pode ser utilizada para decifrá-la e vice-versa.
- A maneira como o algoritmo se comporta depende da chave utilizada na
- entrada: a pública ou a privada
- Cada entidade mantém uma coleção de chaves públicas, pertencentes a outras partes, para seu uso
- A criptografia assimétrica também é conhecida como criptografia por chave pública
- No sistema de chave pública além do processo da confidencialidade de uma mensagem é possível também implementar a autenticação de uma parte.

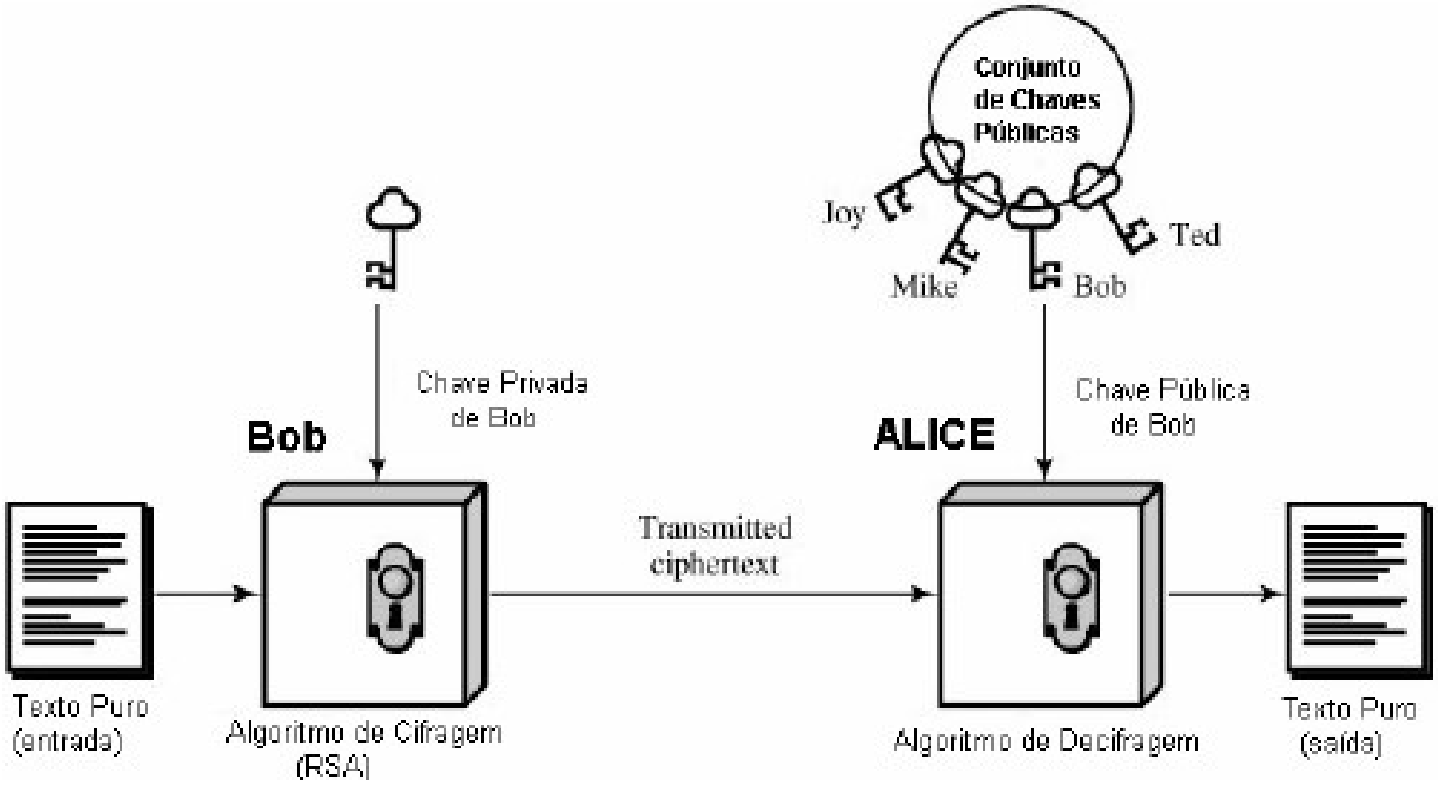
Criptografia Assimétrica Cifragem e Decifragem

- Para enviar uma mensagem para Alice, Bob cifra a mesma a chave pública de Alice
- Ao receber a mensagem Alice a decifra utilizando sua chave privada



Criptografia Assimétrica Autenticação

- Bob Cifra uma mensagem com sua chave privada
- Alice recebe a mensagem e a consegue decifrar utilizando a chave pública de Bob
- Neste caso Bob prova para Alice que é quem afirma ser

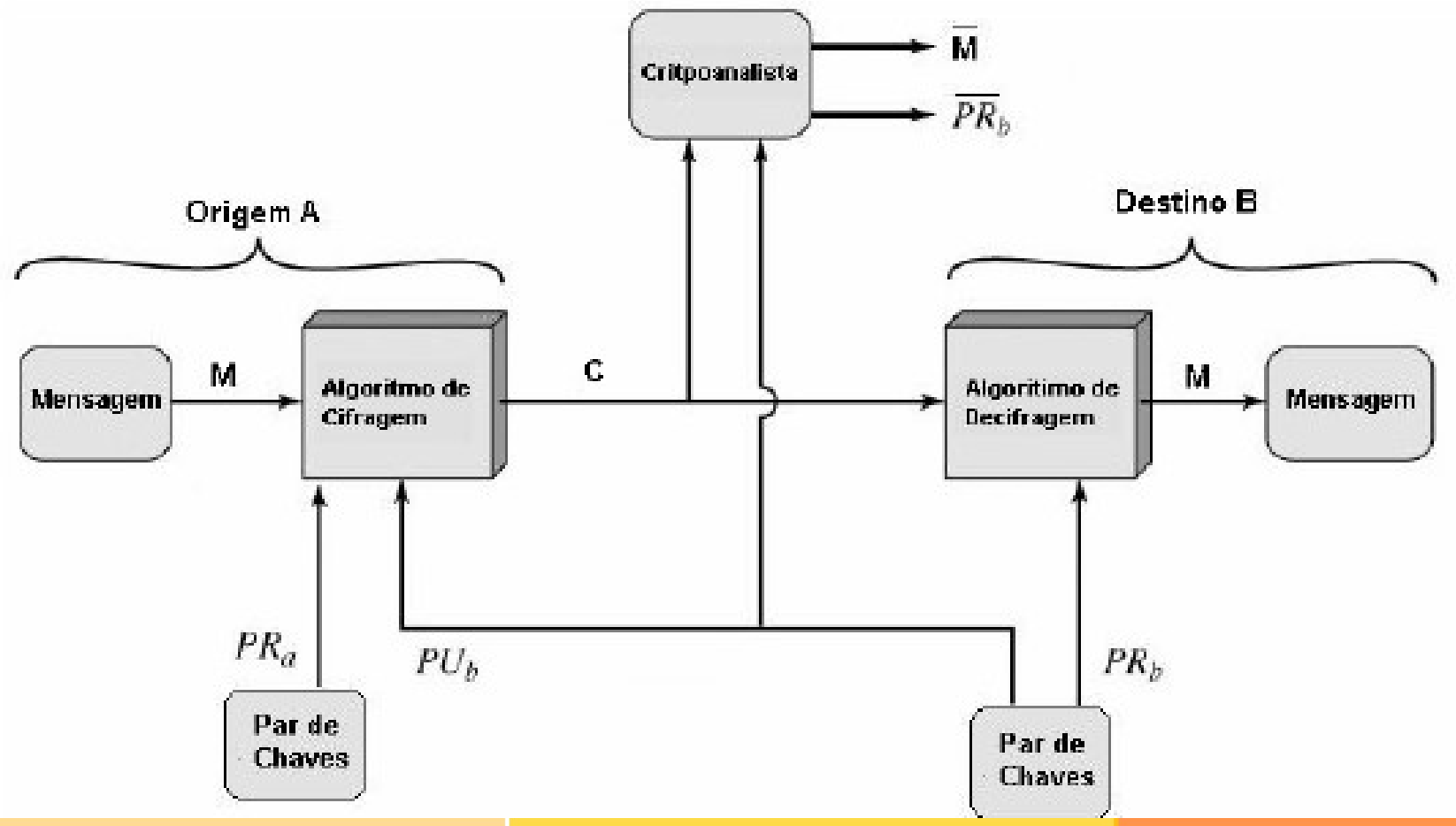


Algoritmos Assimétricos Sistema Criptográfico

- Seja M = [P_1, P_2,…P_m] uma mensagem, o texto puro, com m elementos.
- Os elementos da mensagem são símbolos definidos em um alfabeto finito
- A mensagem será enviada de A para B
- Seja PU_A e PU_B as chaves públicas de A e B
- Seja PR_A e PR_B as chaves privadas de A e B
- Seja C = [C_1, C_2,…C_m] a mensagem M cifrada por uma função de cifragem E, utilizando a chave PU_B
  - C = E(PU_B, M)
- No destino, de posse sua chave privada, é possível que B inverta a transformação, utilizado uma função de decifragem e sua chave privada
  - M = D(PR_B, C)
- Um adversário observando C e que possui acesso a PU_B mas não possui acesso a PR_B e também não conhece M, tentará calcular X e/ou PR_B.
- Caso o interesse seja apenas em M, então seu esforço será para estimar obter uma estimativa da mesma (M)
- Caso o interesse seja em futuras mensagens estão seu esforço será para estimar a chave - PR_B

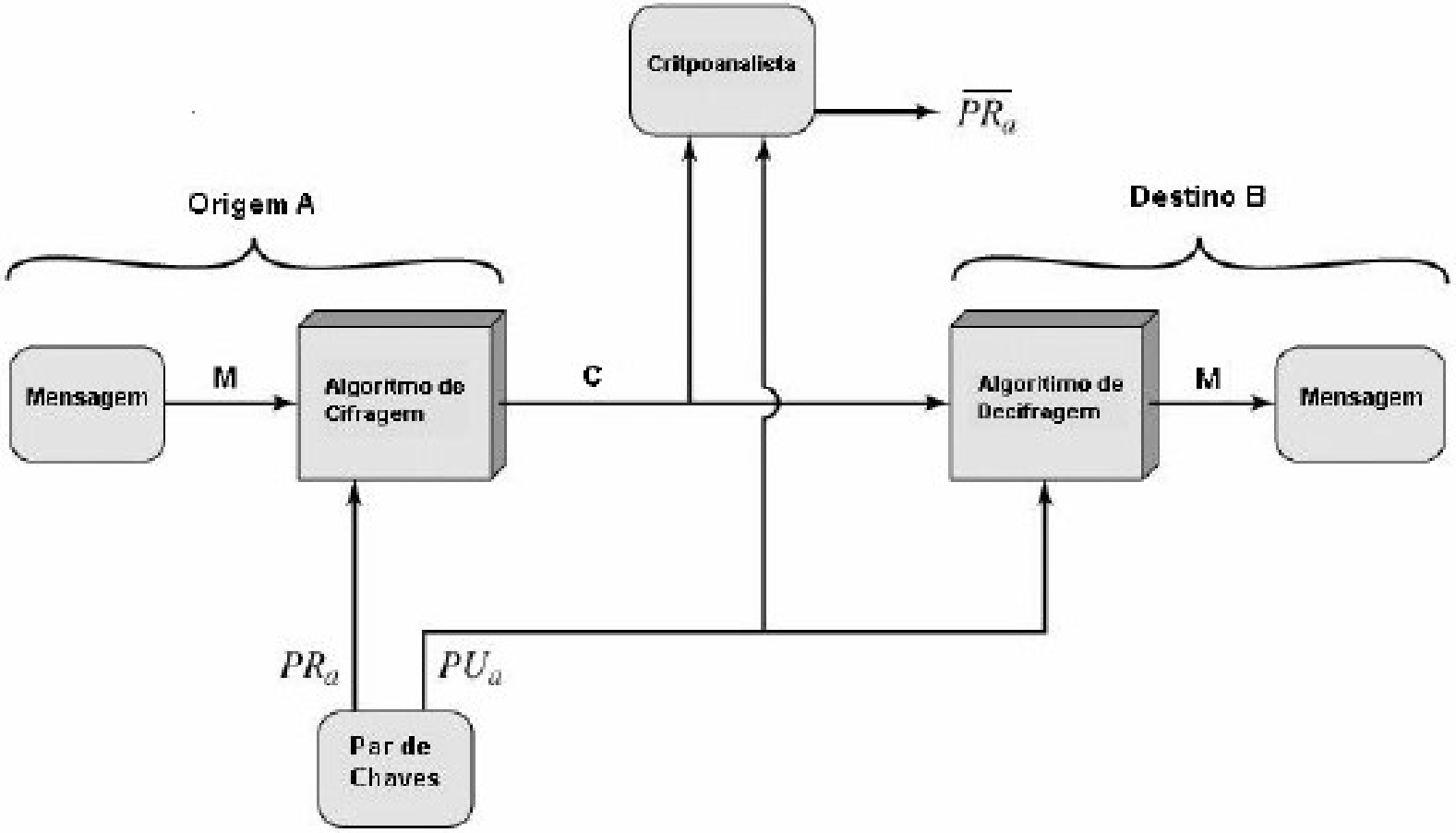
Algoritmos Assimétricos Sistema Criptográfico

- Sistema utilizado para cifragem
  - $C = E(PU_B, M)$
  - $M = D(PR_B, C)$



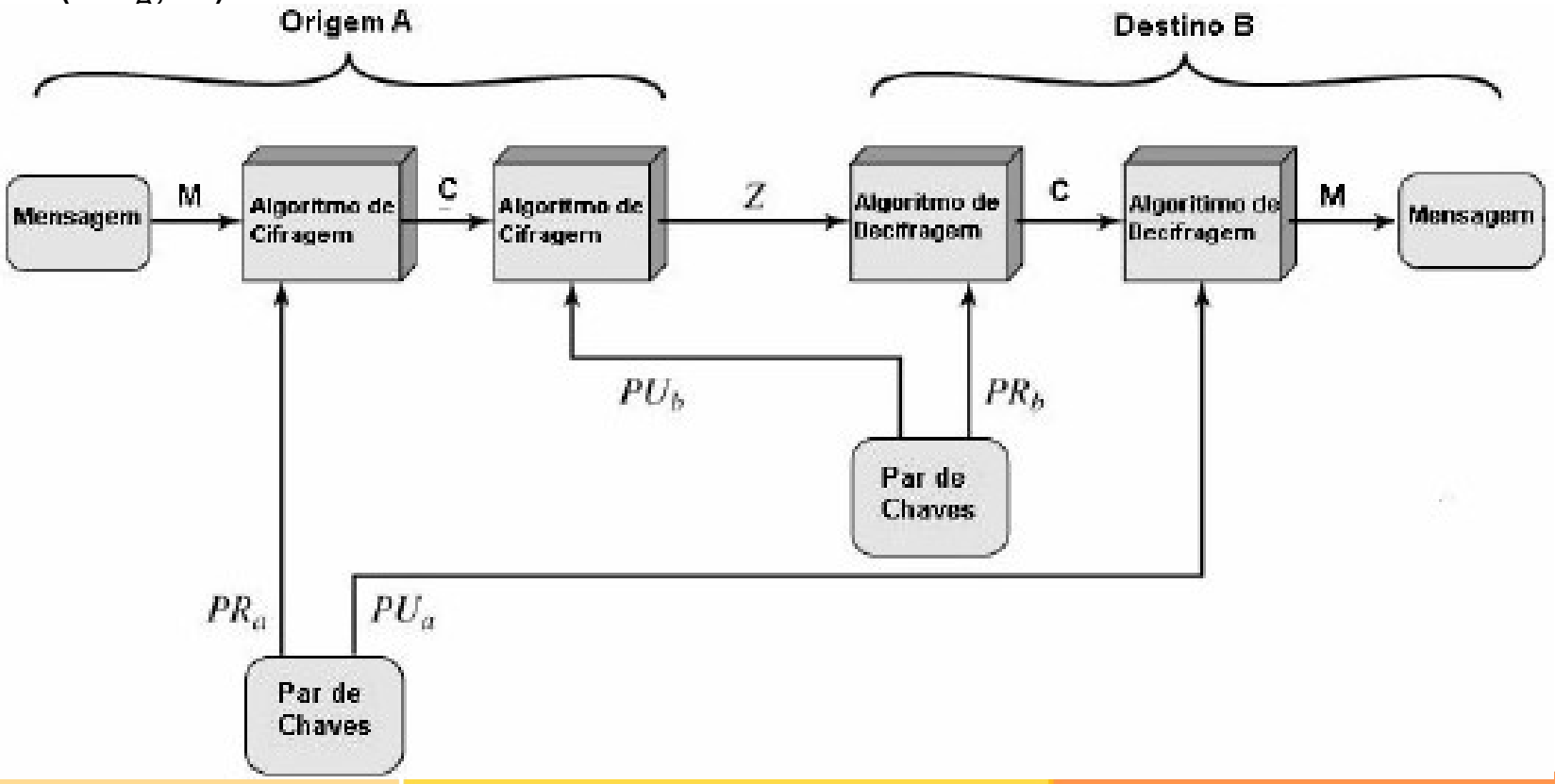
Algoritmos Assimétricos Sistema Criptográfico

- Sistema utilizado para autenticação (não fornece confidencialidade!)
  - $C = E(PR_A, M)$
  - $M = D(PU_A, C)$



Algoritmos Assimétricos Sistema Criptográfico

- Sistema utilizado para confidencialidade e autenticação
  - $C = E(PR_A, M)$
  - $Z = E(PU_B, C)$
  - $C = D(PR_B, E(PU_B, C))$
  - $M = D(PU_A, C)$



Algoritmos Assimétricos Requisitos

- Os requisitos de algoritmos assimétricos, publicados por Diffie e Hellman, são os seguintes:
  - A geração do par de chaves (PU_B e PR_B) é computacionalmente viável
  - É computacionalmente viável para o transmissor (A) cifrar o texto utilizando a chave pública de B (PU_B) → $C = E(PU_B, M)$
  - É computacionalmente viável para o receptor (B) decifrar o texto utilizando sua chave privada (PR_B) → $M = D(PR_B, C) = D[PR_B, E(PU_B, M)]$
  - Deve ser computacionalmente inviável, para um adversário, conhecendo a chave pública de B (PU_B) determinar a sua chave privada (PR_B)
  - Deve ser computacionalmente inviável, para um adversário, conhecendo a chave pública de B (PU_B), decifrar e obter a mensagem M
- O atendimento de tais requisitos não e tarefa fácil
- Poucos algoritmos possuem credibilidade, no sentido de atender tais requisitos

Algoritmos Assimétricos CriptoAnálise

- Assim como os algoritmos simétricos os algoritmos de chave pública são vulneráveis a alguns tipos de ataque:
  - Força Bruta
    - A alternativa neste caso é aumentar o tamanho da chave.
    - Como sistemas de chaves públicas envolvem o uso de funções inversas, o aumento linear da chave pode provocar um aumento não linear no cálculo destas funções
    - Isto pode tornar o processo de cifragem e decifragem lento, impedindo o seu uso nas aplicações comuns.
  - Cálculo da Chave Privada
    - Neste caso o objetivo é encontrar uma maneira de calcular a chave privada a partir da chave pública.
    - Ainda não foi matematicamente provado que esta forma de ataque é inviável para um determinado algoritmo
  - Mensagens Prováveis
    - Neste ataque mensagens corretas prováveis são criadas
    - Por exemplo, suponha que a mensagem a ser transmitida é uma senha com 4 dígitos decimais
    - Neste caso existem apenas 10000 possíveis mensagens.
    - Um adversário poderia produzir estas mensagens cifradas com a chave pública e enviar até obter sucesso.

Algoritmos Assimétricos Aplicações

- A partir dos algoritmos assimétricos, ou de chave pública, podem ser utilizados a fim prover algumas funcionalidades relativas à segurança, dependendo do algoritmo utilizado e da maneira como o mesmo é utilizado
- As seguintes funcionalidades são suportadas:
  - Cifragem e Decifragem
    - Neste caso o transmissor (A) cifra a mensagem utilizando a chave pública do receptor (PU_B)
  - Assinatura Digital (Digital Signatures)
    - Neste caso o transmissor (A) cifra (assina) a mensagem utilizando a sua chave privada (PR_A)
  - Troca de Chaves (Key Exchange)
    - Os dois lados trocam mensagens a fim de compartilhar uma chave de sessão.
    - Este processo pode envolver as chaves privadas de um ou de ambos, bem como suas chaves públicas
- Nem todos os algoritmos de chave pública oferecem todas estas aplicações
- Na prática o tamanho das chaves tornaram o processo de cifragem lento e desta forma algoritmos basicamente para assinaturas digitais e troca de chaves

Algoritmos Assimétricos

- A tabela abaixo mostra alguns algoritmos de chave públicas aceitos e normalmente utilizados

Algoritmo	Cifragem e Decifragem	Assinatura Digital	Troca de Chaves	Problema Matemático
RSA	Sim	Sim	Sim	IFS
Elliptic Curve Crytography (ECC)	Sim	Sim	Sim	ECDLP
Diffie-Hellman	Não	Não	Sim	DLS
Digital Signature Standard (DSS)	Não	Sim	Não	DLS

- Todos baseiam-se nos mesmos princípios, porém diferem no problema matemático em que se baseiam:
  - Fatoração de Inteiros (Integer Factorization Problem – IFP)

http://www.rsasecurity.com/rsalabs/node.asp?id=2093

  - Logaritmo Discreto (Discrete Logarithm Problem – DLP)
  - Curvas Elípticas (Elliptic Curve Discrete Logarithm Problem – ECDLP)

Algoritmo RSA
Visão Geral

- O algoritmo pode ser divido em duas partes
  - Cálculo da Chave Pública
    - Escolha dois grandes números primos **p e q, tais que p ≠ q**
    - Calcule o valor n, tal que **n = p · q**
    - Calcule o função Totiente **[φ(n)]**
    - Escolha um valor **e, tal que 1 < e < φ(n) e seja co-primo de φ(n)**
    - Calcule o valor de d, tal que o produto **e · d seja congruente com 1 em** módulo n ou seja: **e · d ≡ 1 mod φ(n)**
  - Cifragem ou Decifragem
    - Cifragem
      - $C = M^e \bmod n$, sendo $M < n$; $PR = \{e,n\}$
    - Decifragem
      - $M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$; $PU = \{d,n)$

Conceitos Matemáticos
Divisor e Máximo Divisor Comum

- Divisor
  - a é divisível por b se o resto da divisão de a por b for zero, ou seja, na expressão abaixo tem-se r = 0:
    - a = b · q + r, logo
    - a = b · q
    - Neste caso o número b é divisor de a
- Máximo Divisor Comum (mdc ou gcd)
  - Sejam dados dois números inteiros a e b.
  - O máximo divisor comum entre a e b (mdc ou gcd) é o maior número que divide a e divide b.
  - mdc (a, b) = n

ou

  - gdc (a, b) = n

a	b	Divisores Comuns	mdc(a,b)
36	80	1,2, 4	4
45	18	1,3, 5, 9	9

Conceitos Matemáticos
Número Primo e Congruência

- Número Primo
  - É um número que possui apenas dois divisores: o próprio número e a unidade.
  - Dois números a e b são primos entre si, primos relativos ou co-primos, se e somente se:
    - mdc (a, b) = 1
- Congruência
  - É a relação entre dois números inteiros que, divididos por um terceiro chamado módulo de congruência, deixam o mesmo resto.
  - Por exemplo, 20 é côngruo ou congruente de 14 com relação a 6 (20/6=3 restando 2 e 14/6=2 restando 2).
  - Suponha que a, b e n sejam números inteiros diferentes de zero.
  - Diz-se que a é congruente de b módulo n se n dividir a-b, ou seja:
    - a ≡ b mod n; do exemplo anterior temos 20 ≡ 14 mod 6

Conceitos Matemáticos
Função Totiente de Euler - ϕ(n)

- A função φ(n) indica a quantidade de números inteiros menores que n e que são primos entre si.

	a	1	2	3	4	5	6	7	8	9	10	11	12	13	14
n = 15	mdc(n,a)	1	1	3	1	5	3	1	1	3	5	1	3	1	1

  - φ(1) = 1; φ(2) = 2; φ(3) = 2; φ(15) = 8
- Se n é primo então
  - φ(n) = n-1
  - Pois
    - n só é divisível por 1 e por n;
    - todos os números inferiores a n não podem ser divisíveis por n.
    - Logo o único divisor comum é a unidade.
- A função φ(n) é multiplicativa, sendo n uma multiplicação de co-primos, logo:
  - n = p · q, sendo p e q co-primos logo:
  - φ(n) = φ(p · q) = φ(p) φ(q)
  - Portanto
  - φ(n) = φ(p · q) = φ(p) φ(q) = (p-1) (q-1)

Algoritmo RSA
Cálculo da Chave Pública

- Escolha dois grandes números primos **p e q, tais que p ≠ q**
  - Estes números devem ser secretos e normalmente possuem muitas casas decimais, por exemplo p e q, cada um com pelo menos 768 bits
  - Neste caso **p e q são co-primos**
- Calcule o valor n, tal que **n = p · q**
- Calcule o valor da função Totiente [**φ(n)**]
  - Como p e q são co-primos, temos que **φ(n) = (p-1)(q-1)**
- Escolha um valor **e, tal que 1 < e < φ(n) e seja co-primo de φ(n).**
  - Neste caso mdc [e, **φ(n)] = 1**
- Calcule o valor de d, tal que o produto **e · d seja congruente** com 1 em módulo φ(n) ou seja:
  - **e · d ≡ 1 mod φ(n)**
  - O valor **e o valor d devem ser co-primos de φ(n)**
  - Para obter **d** pode ser utilizado o algoritmo euclidiano estendido

Conceitos Matemáticos
Algoritmo Euclidiano

- **Cálculo do máximo divisor comum**
- O algoritmo Euclidiano permite calcular o máximo divisor comum entres dois números - mdc(a,b)
  - Sendo a maior que b, a será o primeiro dividendo e b o primeiro divisor
  - Nas próximas linhas
    - o dividendo é o divisor da linha anterior
    - o divisor é o resto da linha anterior
- O algoritmo pára quando o resto é zero. O resto na linha imediatamente anterior é o máximo divisor comum
- Quando o mdc(a,b) = 1 então
- Exemplo:
  - mdc(120,23) = 1

Dividendo	Divisor	Quociente	Resto
120	23	5	5
23	5	4	3
5	3	1	2
3	2	1	1
2	1	2	0

Conceitos Matemáticos
Algoritmo Euclidiano Estendido

- **Cálculo do máximo divisor comum**
- Se mdc(a,b) = r então existe x e y tais que
  - $a \cdot x + b \cdot y = mdc(a,b) = r$
- O algoritmo Euclidiano é o inverso do Algoritmo Euclidiano
  - Da linha onde foi encontrado o mdc(a,b) temos:
    - 1 = 3 - 1(2)
  - Substituindo o termo (2) pela linha anterior temos:
    - 1 = 3 – 1[**5 - 1(3)] = 3 - 1(5) +1(3) = 2(3) - 1(5)**
  - Substituindo o termo (3) temos:
    - 1 = 2[23-4(5)] - 1(5) = 2(23) - 8(5) - 1(5) = 2(23) - 9(5)
  - Finalmente substituindo o termo (5) tempos
    - 1 = 2(23) – 9[120-5(23)] = 2(23) - 9(120) + 45(23) = 47(23) - 9(120)
  - Neste caso como 120 e 23 são co-primos pode-se dizer:
    - $-9 \times 120 \equiv 1 \pmod{23}$
    - $47 \times 23 \equiv 1 \pmod{120}$

Dividendo	Divisor	Quociente	Resto
120	23	5	5
23	5	4	3
5	3	1	2
3	2	1	1
2	1	2	0

Algoritmo RSA
Exemplo - Cálculo da Chave Pública

- Inicialmente escolhe-se os número p e q, primos e diferentes entre si:
  - p = 11 e q = 13
- Calcule o valor n, tal que **n = p · q**
  - **n = p · q = 11 x 13 = 143**
- Calcule o valor da função Totiente [φ**(n)], tal que φ(n) = (p-1)(q-1)**
  - **φ(n) = (p-1)(q-1) = (11-1)(13-1) = 120**
- Escolha um valor **e, tal que 1 < e < φ(n) e seja co-primo de φ(n).**
  - Vamos escolher e = 23, pois é menor 120
  - Neste caso mdc [e, **φ(n)] = 1 ou seja mdc(23,120) = 1**
- Calcule o valor de d, tal que o produto **e · d seja congruente com 1 em** módulo φ(n) ou seja:
  - **e · d ≡ 1 mod φ(n)**
  - O valor **e o valor d deve ser co-primos de φ(n)**
  - Para obter d pode ser utilizado o algoritmo euclidiano estendido
  - Neste caso o valor de d = 47
- Finalmente temos PR = {23,143} e PU = {47,143}

Algoritmo RSA
Exemplo - Cifragem

- A partir das chaves obtidas anteriormente temos:
  - PR = {23,143} e PU = {47,143}
- Na cifragem será utilizada a chave pública
  - $C = M^e \bmod n$, sendo $M < n$; PU = {d,n) e PR = {e,n}
  - Logo teremos:
  - $C = M^{23} \bmod 143$
  - Supondo a mensagem igual o número 2
    - $C = 2^{23} \bmod 143 = ?$
    - $C = [(2^{16} \bmod 143) x (2^4 \bmod 143) x (2^2 \bmod 143) x (2^1 \bmod 143)] \bmod 143$
    - C = [42 x 16 x 4x 2] mod 143
    - C = 85
    - A mensagem será enviada como 85

Algoritmo RSA
Exemplo - Decifragem

- A partir das chaves obtidas anteriormente temos:
  - PR = {23,143} e PU = {47,143}
  - Na decifragem será utilizada a chave privada
  - $M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$
  - Logo teremos:
    - Como C = 85
    - $M = 85^{47} \bmod 143 = ?$
    - $[(85^{32} \bmod 143)x(85^8 \bmod 143)x(85^2 \bmod 143)x(85^1 \bmod 143)] \bmod 143$
    - Utilizando as propriedades da aritmética modular teremos que:
    - M = 2

Algoritmo RSA
Exemplo – Decifragem (Operações)

- $M = 85^{47} \bmod 143 = ?$
- Como
  - $85^{47} = 85^{8}_{1} \times 85^{8}_{2} \times 85^{8}_{3} \times 85^{8}_{4} \times 85^{8}_{5} \times 85^{4} \times 85^{2} \times 85^{1}$
- Sendo assim teremos:
  - $M = (85^{8}_{1} \times 85^{8}_{2} \times 85^{8}_{3} \times 85^{8}_{4} \times 85^{8}_{5} \times 85^{4} \times 85^{2} \times 85^{1}) \bmod 143$
- Logo teremos:
  - $85^{8} \bmod 143 = 272490525039062 5 \bmod 143 = 16$
  - $85^{4} \bmod 143 = 52206625 \bmod 143 = 48$
  - $85^{2} \bmod 143 = 7225 \bmod 143 = 75$
  - $85^{1} \bmod 143 = 85 \bmod 143 = 85$
- Pelas regras da aritmética modular podemos escrever:
  - $M=(16 \times 16 \times 16 \times 16 \times 16 \times 48 \times 75 \times 85) \bmod 143$
  - $M = 320864256000 \bmod 143$
- Finalmente teremos
  - $M = 2$

Algoritmo RSA
Outro Exemplo – Cálculo chaves

- Seja p = 11 e q = 17, calcular a chave pública e privada
  - n = p x q = 11 x 1 7 = 187
  - φ(n) = (p-1)(q-1) = (11-1)(17-1) = 160
  - Escolha um valor **e, tal que 1 < e < φ(n) e** mdc [e, φ**(n)].**
    - **e = ?**
    - **Candidatos: 3; 27; 7; 159 → Será escolhido o número 7**
  - Cálculo de **d**
    - **Algoritmo Euclidiano Estendido**

- Sendo M = 88, utilizar o RSA

Algoritmo RSA
Outro Exemplo

- Sendo as seguintes chaves:
  - PR = {7,187} e PU = {23,187}.
- Cifragem, sendo M = 88
  - $C = M^e \bmod n$, sendo $M < n$; PU = {d,n) e PR = {e,n}
- Logo teremos:
  - $C = 88^7 \bmod 187 = ?$
  - $C = [(88^4 \bmod 187) \times (88^2 \bmod 187) \times (88^1 \bmod 187)] \bmod 187$
  - $C = (88 \times 77 \times 132) \bmod 187 = 11$
- Decifragem será utilizada a chave privada
  - $M = C^d \bmod n$
  - $M = 11^{23} \bmod 187$
  - $M = [(11^1 \bmod 187) \times (11^2 \bmod 187) \times (11^4 \bmod 187) \times (11^8 \bmod 187) \times (11^8 \bmod 187)] \bmod 187$
  - Como: $11^1 \bmod 187 = 11$; $11^2 \bmod 187 = 121$; $114 \bmod 187 = 55$ e $11^8 \bmod 187 = 33$ temos:
  - **$M = (11 \times 121 \times 55 \times 33 \times 33) \bmod 187 = 79.720.245 \bmod 187 = 88$**