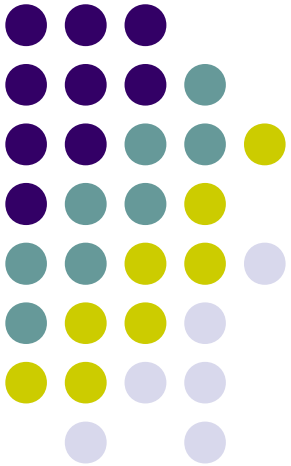
Universidade Federal de São Paulo – UNIFESP

Segurança Computacional

Aula 04: Gerenciamento de chaves e Criptografia de chave pública

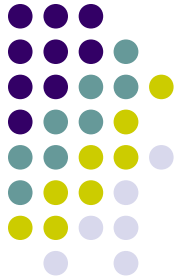


Prof.
Valério Rosset

2015-1

Criptografia simétrica

Gerenciamento de chaves



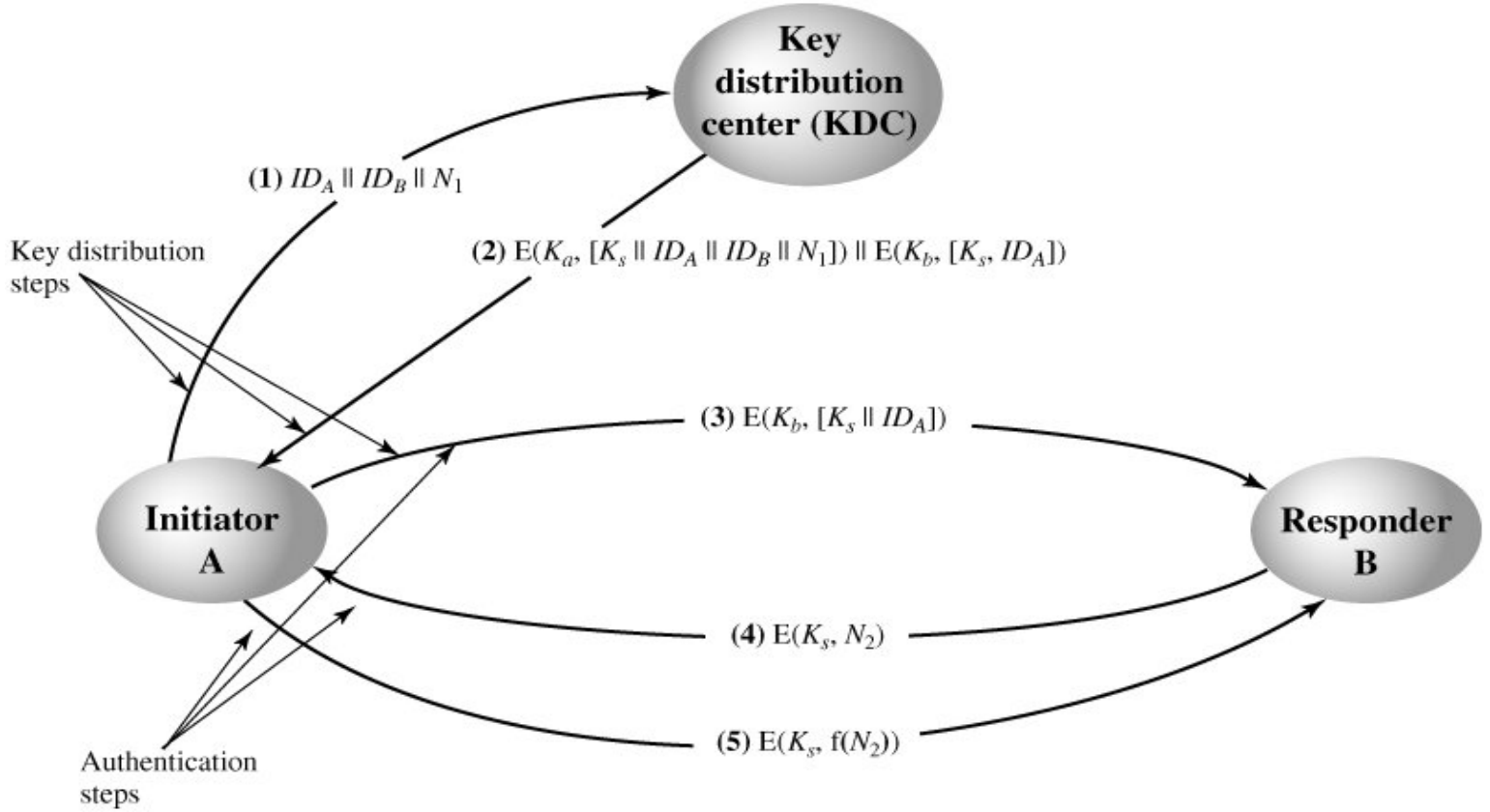
- Utilização puramente de criptografia simétrica em ambientes distribuídos é pouco aplicável em virtude do problema de distribuição chaves.

- Esse problema ocorre porque nesse modelo de criptografia se torna necessário o conhecimento prévio de uma chave pelas partes envolvidas.

Criptografia simétrica

Gerenciamento de chaves

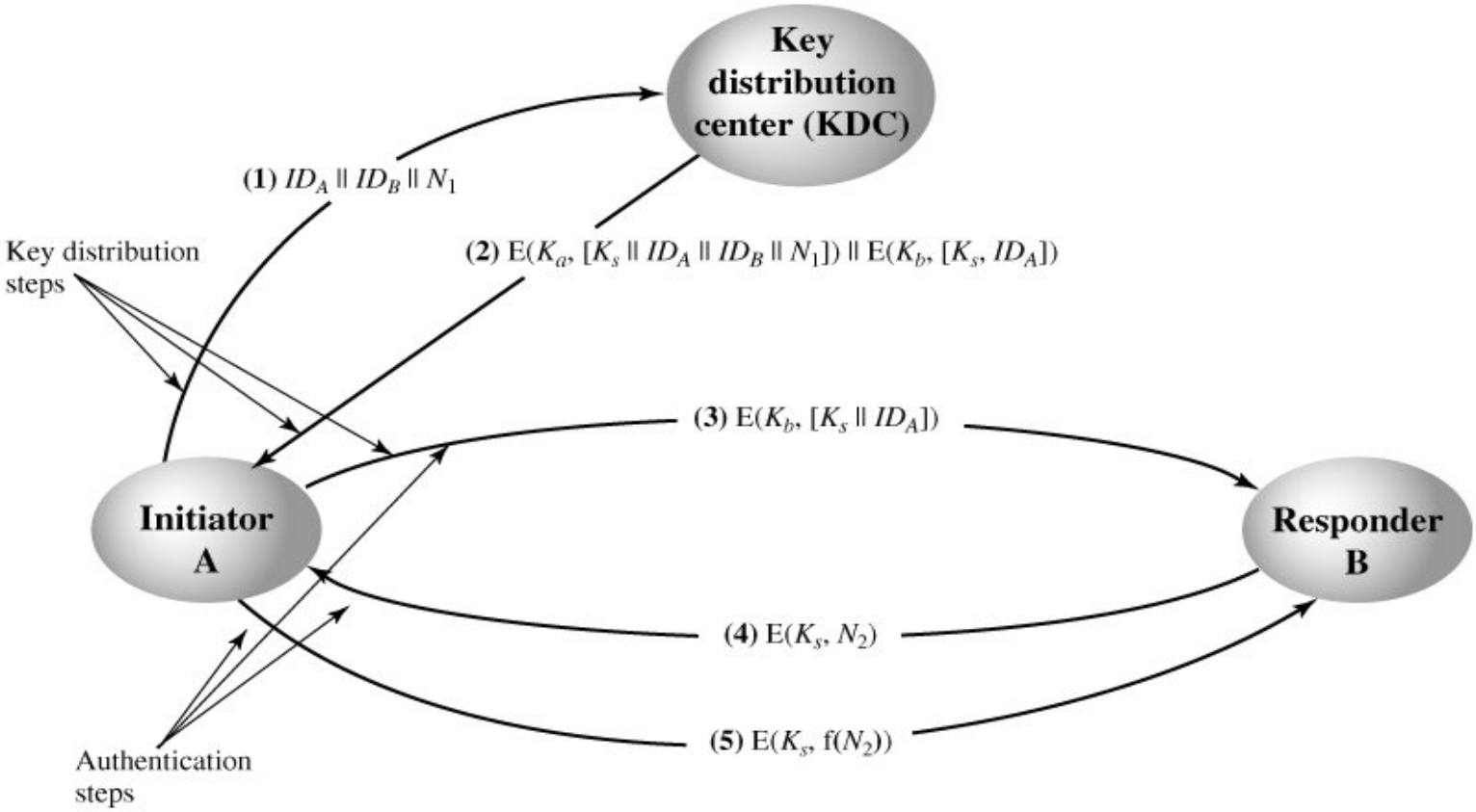
- **Podemos imaginar um sistema onde um KDC distribui chaves de sessão para uma comunicação entre 2 participantes.**



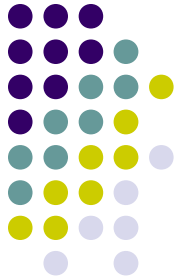
Criptografia simétrica

Gerenciamento de chaves

- Para que isso ocorra é necessário que A e o KDC conhecam Ka e B e KDC conheçam Kb (essas chaves são chamadas ***chaves mestras***).



Criptografia Assimétrica ou de chave Pública

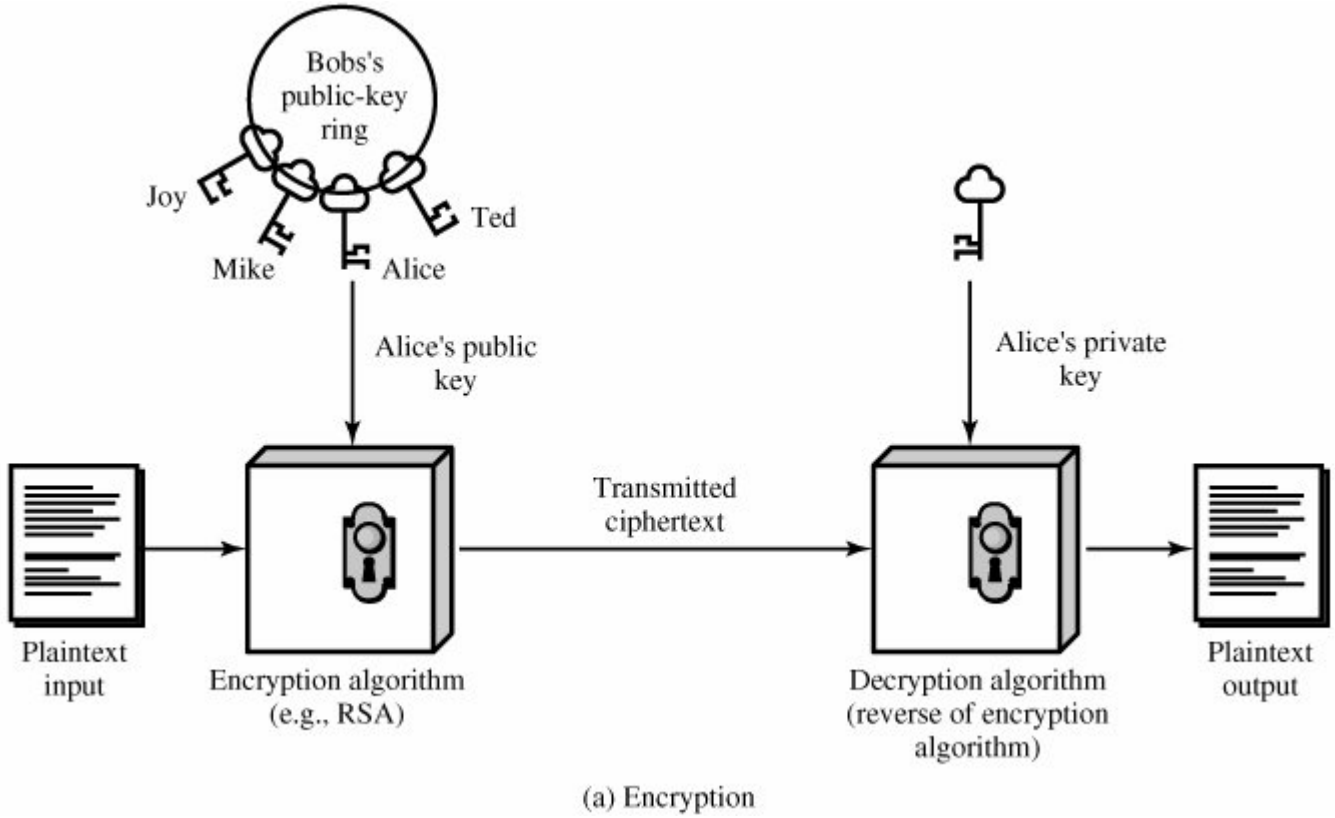


- Princípios:
  - Existem duas chaves distintas que podem ser usadas para decifrar ou cifrar informações.
  - Cada uma dessas chaves decifra o resultado da outra
  - Uma dessas chaves é um segredo e é chamada ***chave privada***
  - Outra é distribuída livremente para qualquer indivíduo (***chave pública***)

Criptografia de chave Pública

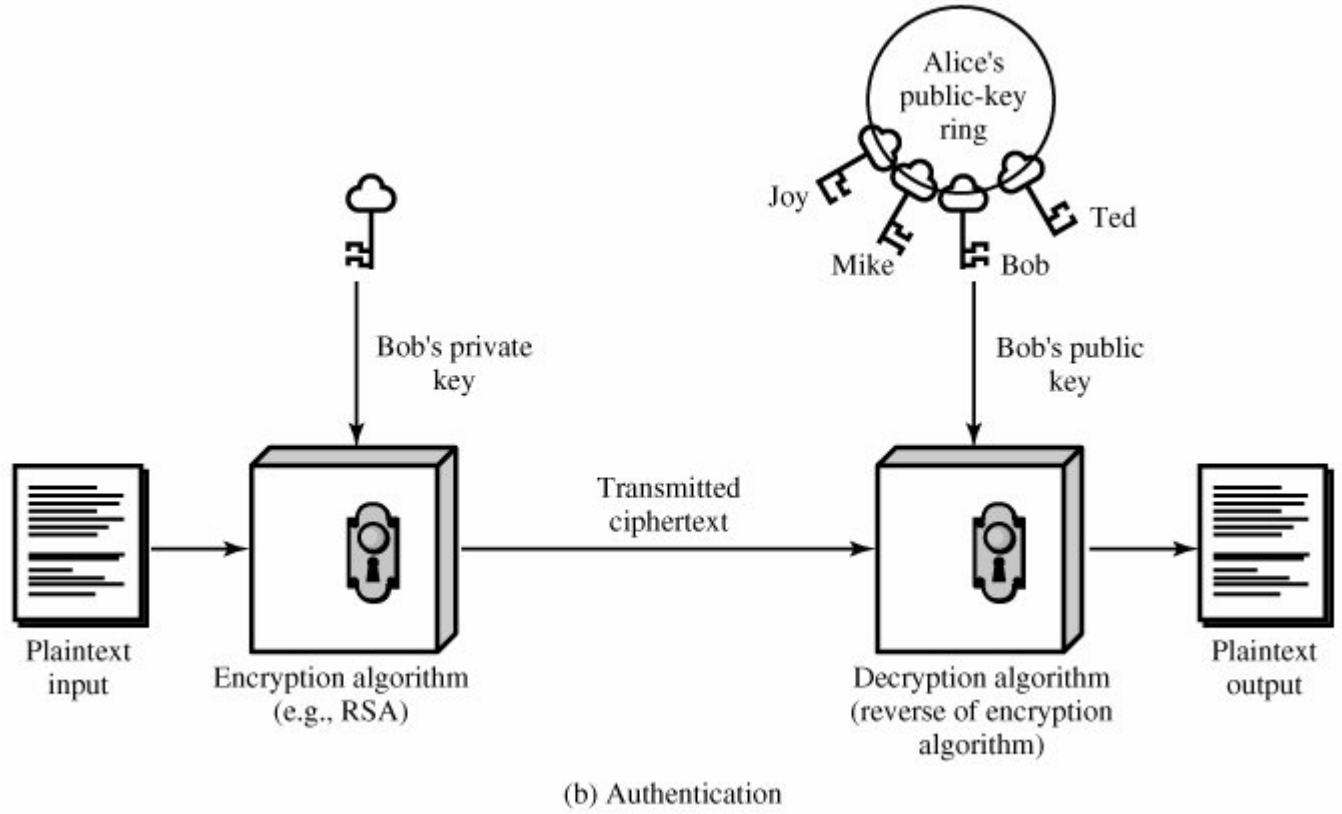
Propriedades

- ***Confidencialidade***



Criptografia de chave Pública
Propriedades

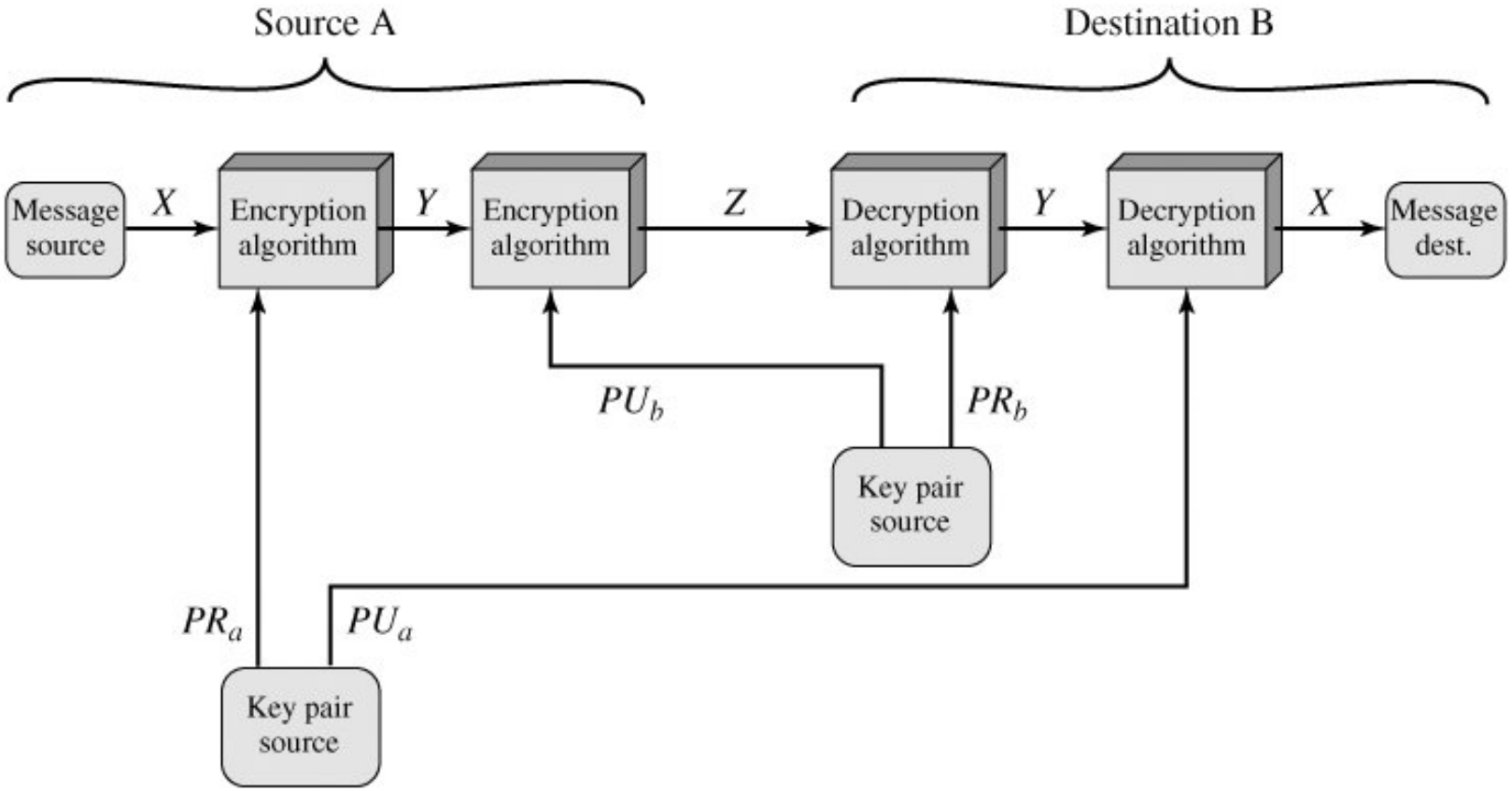
- **Autenticidade**



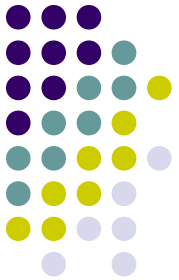
(b) Authentication

Criptografia de chave Pública
Propriedades

- **Confidencialidade + Autenticidade**



Criptografia de chave Pública
RSA

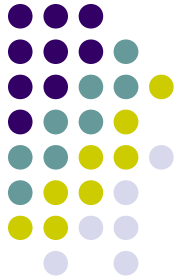


- **Formato**
  - **$C = M^e \bmod n$**
  - **$M = C^d \bmod n$**
- **Requisitos:**
  - **p,q : dois primos (*priv. escolhidos*)**
  - **n = p.q; (*público, calculado*)**
  - **e, com mdc (Φ(n), e) = 1; (*pub, escolhido*)**
  - **$d \equiv e^{-1} \pmod{\Phi(n)}$ (*priv. calculado*)**

Dois # são relativamente primos se o único divisor comum for 1. Equivalente à mdc(a,b)=1.

Onde: Φ(n) é a função Totiente de Euler, # de inteiros positivos relativamente primos e menores a n, Φ(1)=1, Φ(6)=2, Φ(11)=10 …

RSA – *Geração de chaves*

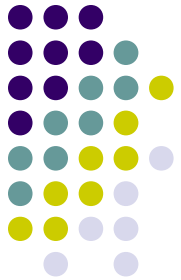


Algorithm: RSA Key Generation

Output: public key: $k_{pub} = (n, e)$ and private key $k_{pr} = d$

1. Choose two large primes p, q
2. Compute $n = p * q$
3. Compute $\Phi(n) = (p-1) * (q-1)$
4. Select the public exponent $e \varepsilon \{1, 2, \ldots, \Phi(n)-1\}$ such that $\gcd(e, \Phi(n)) = 1$
5. Compute the private key d such that $d * e \equiv 1 \bmod \Phi(n)$
6. **RETURN** $k_{pub} = (n, e)$, $k_{pr} = d$

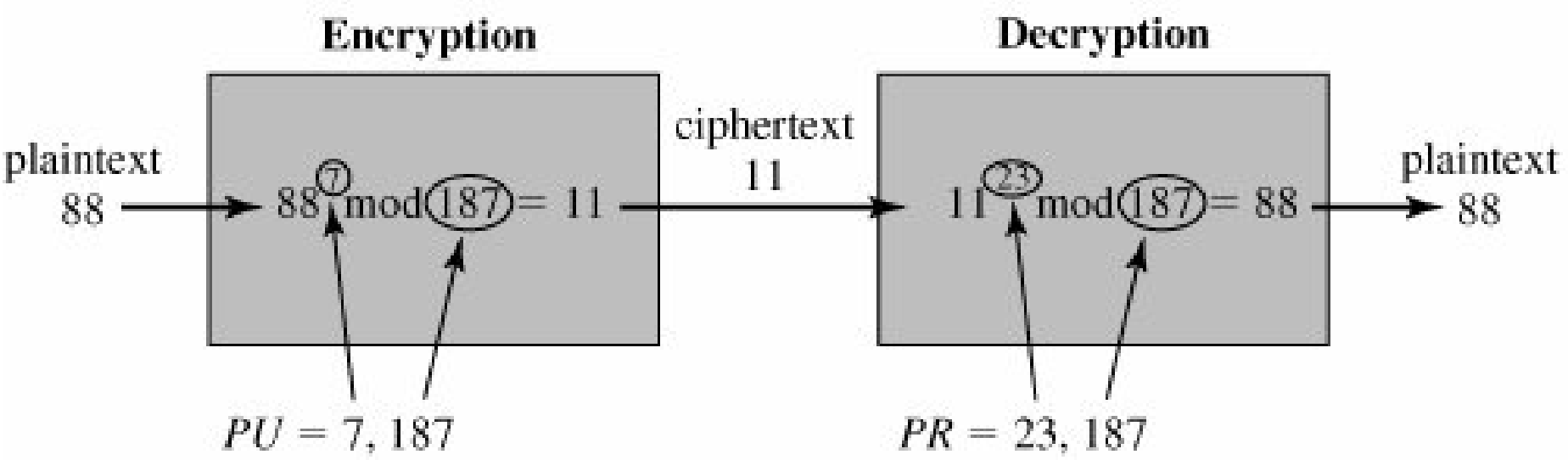
RSA – *Geração de chaves*
Exemplo



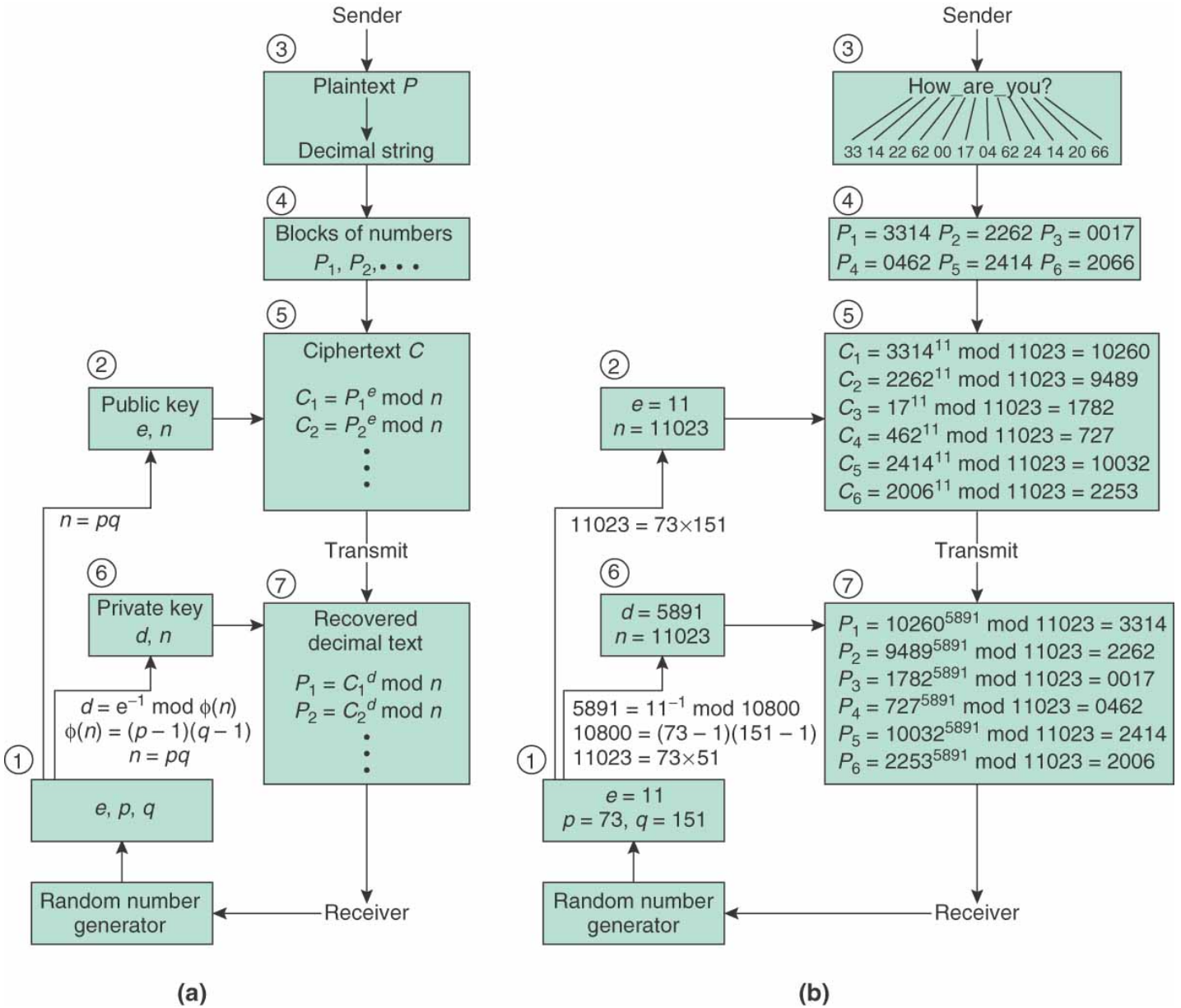
1. Selecione 2 primos, p = 17 and q = 11.
2. Calcule n = pq = 17 x 11 = 187.
3. Calcule Φ(n) = (p-1)(q-1) = 16 x 10 = 160.
4. selecione *e* sendo relativamente primo a Φ(n) = 160 e menor que Φ(n): *e* = 7.
5. Determine *d* assumindo que: *de* ≡ 1 (mod 160), *d* < 160. o valor será *d* = 23, porque 23 x 7 = 161 = 10 x 160 + 1; *d* poderá ser calculado usando o Alg. Ext. de Euclides.

RSA – Aplicação

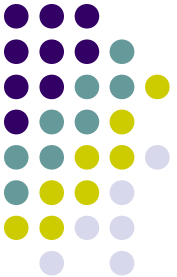
Exemplo



RSA - Exemplo


Sender
3
Plaintext P
Decimal string
4
Blocks of numbers
P1, P2, . . .
5
Ciphertext C
C1 = P1^e mod n
C2 = P2^e mod n
2
Public key
e, n
n = pq
Transmit
6
Private key
d, n
7
Recovered decimal text
P1 = C1^d mod n
P2 = C2^d mod n
d = e^–1 mod φ(n)
φ(n) = (p – 1)(q – 1)
n = pq
1
e, p, q
Random number generator
Receiver
(a)
Sender
3
How_are_you?
33 14 22 62 00 17 04 62 24 14 20 66
4
P1 = 3314 P2 = 2262 P3 = 0017
P4 = 0462 P5 = 2414 P6 = 2066
5
C1 = 3314^11 mod 11023 = 10260
C2 = 2262^11 mod 11023 = 9489
C3 = 17^11 mod 11023 = 1782
C4 = 462^11 mod 11023 = 727
C5 = 2414^11 mod 11023 = 10032
C6 = 2006^11 mod 11023 = 2253
2
e = 11
n = 11023
11023 = 73×151
Transmit
6
d = 5891
n = 11023
7
P1 = 10260^5891 mod 11023 = 3314
P2 = 9489^5891 mod 11023 = 2262
P3 = 1782^5891 mod 11023 = 0017
P4 = 727^5891 mod 11023 = 0462
P5 = 10032^5891 mod 11023 = 2414
P6 = 2253^5891 mod 11023 = 2006
5891 = 11^–1 mod 10800
10800 = (73 – 1)(151 – 1)
11023 = 73×51
1
e = 11
p = 73, q = 151
Random number generator
Receiver
(b)

Criptografia de chave pública
Gerenciamento de chaves



- Modelos:
  - Anúncio público
  - Diretório
  - Autoridade de Chave pública
  - Certificados

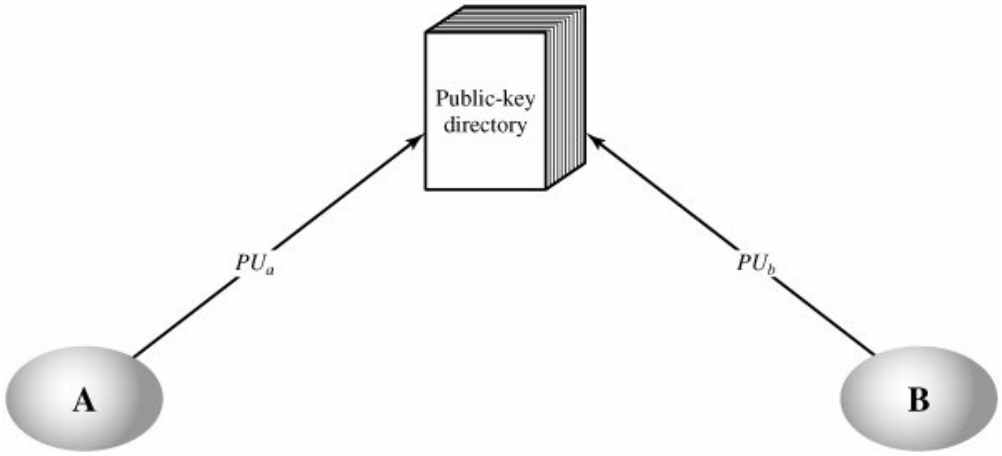
Criptografia de chave pública
Gerenciamento de chaves

- Anúncio público
  - Enviam a todos a sua chave pública

- Problema:
  - **Falsificação** de anúncio …

Criptografia de chave pública
Gerenciamento de chaves

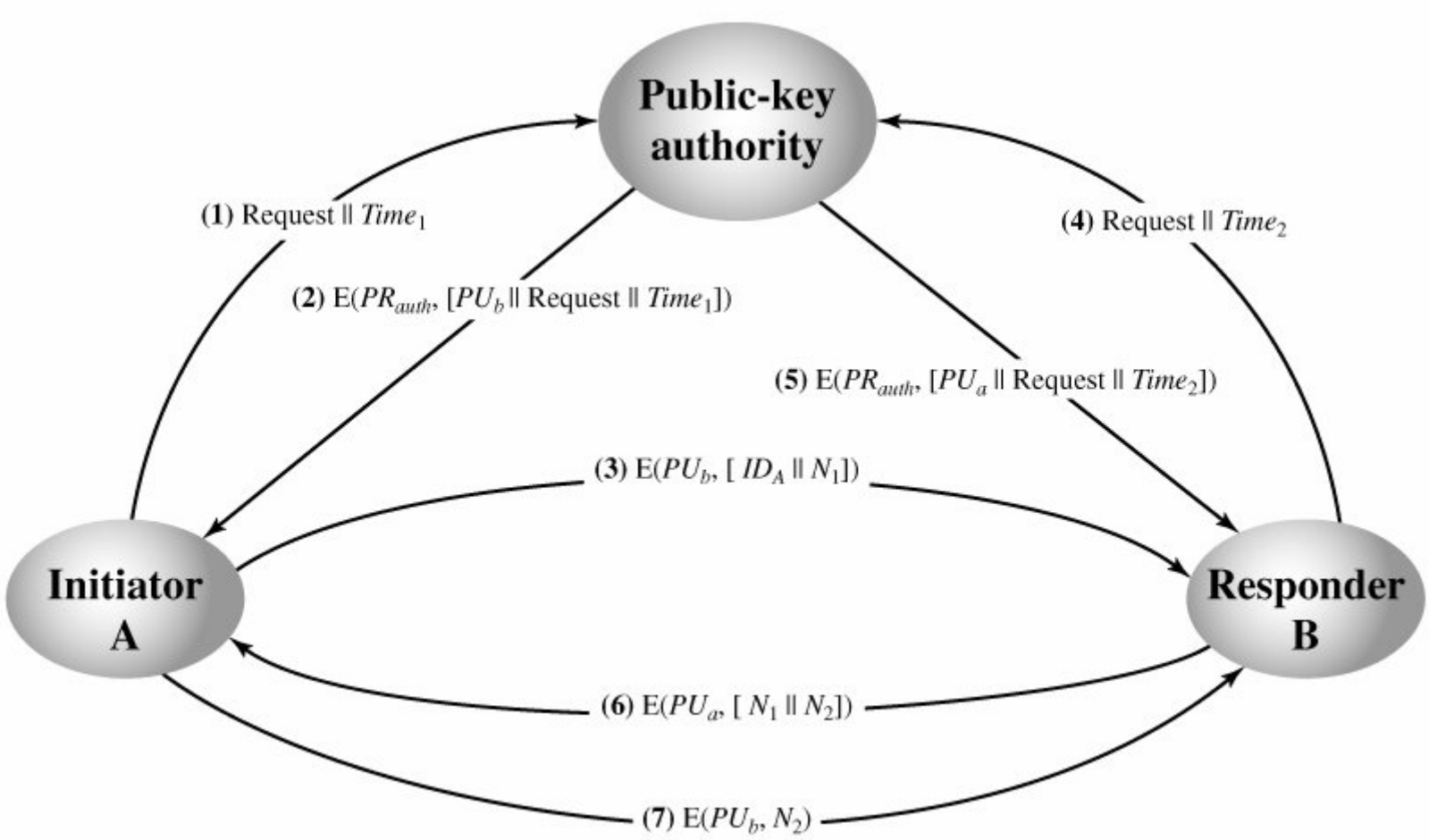
- **Diretório**
  - Usuários publicam suas chaves em um ***diretório***.
  - Inclusão dessas chaves necessita de acordo prévio.



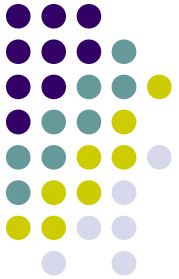
  - Necessita de uma autoridade de controle, porém pode sofrer com ataques sobre as chaves dessa autoridade.

Criptografia de chave pública
Gerenciamento de chaves

- ***Autoridade de Chave Pública***

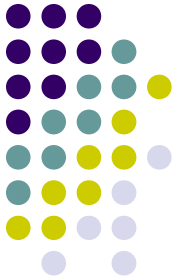


Criptografia de chave pública
Gerenciamento de chaves



- ***Autoridade de Chave Pública***
  - Desvantagens:
    - Gargalo para essa entidade
    - Diretório sujeito a violação

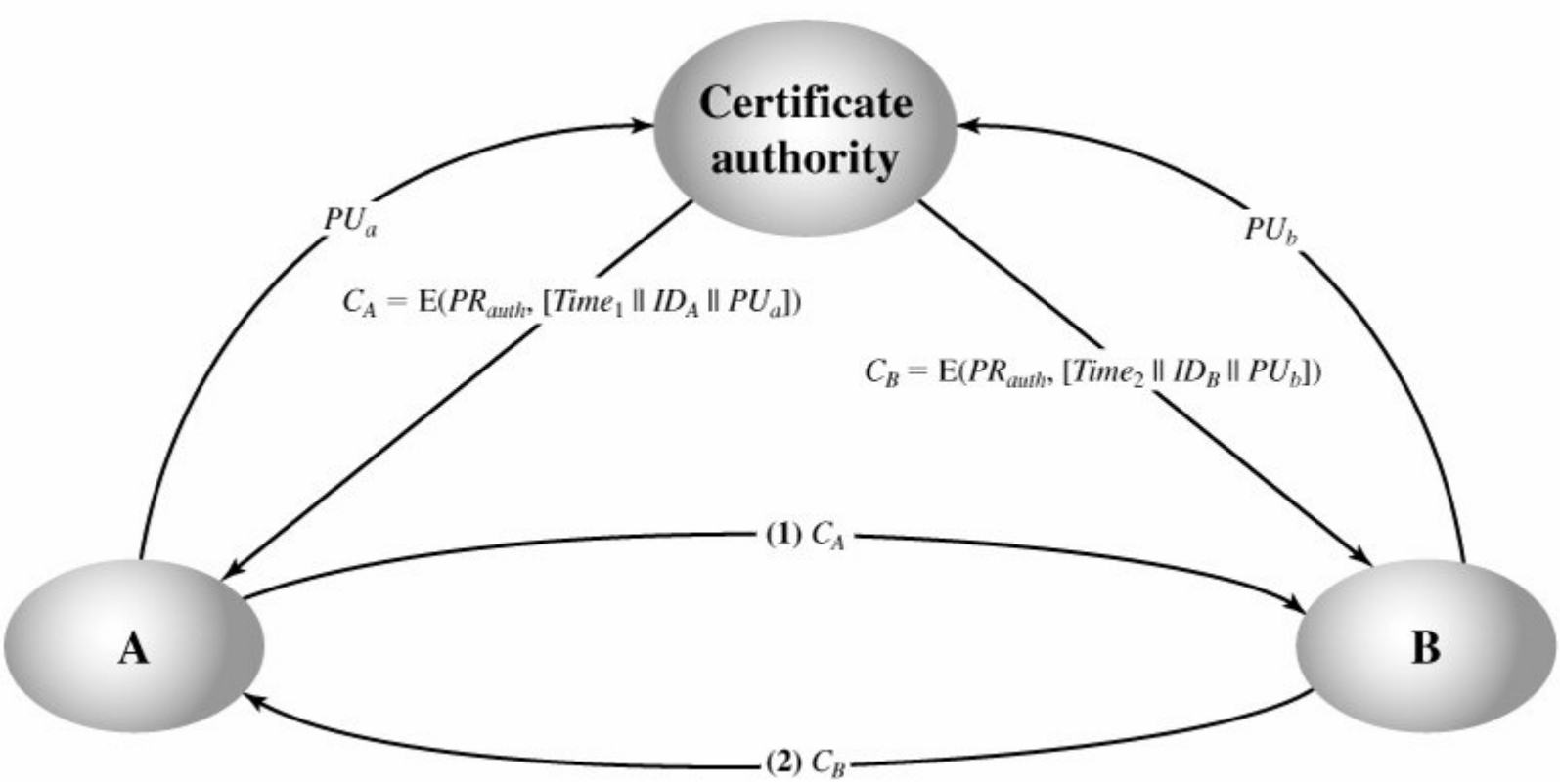
Criptografia de chave pública
Gerenciamento de chaves



- ***Certificados de Chave Pública***
  - Não necessita contactar a autoridade certificadora
  - Ideia básica é disponibilizar um conjunto de chave publica de usuários criado com a chave privada da certificadora.
  - Qualquer usuário pode usar esse certificado para confirmar a chave publica de outro.

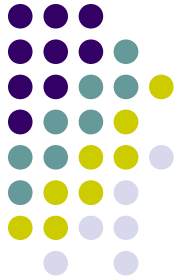
Criptografia de chave pública
Gerenciamento de chaves

- ***Certificados de Chave Pública***



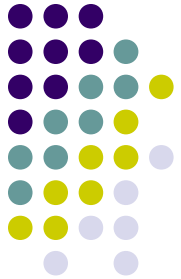
Criptografia de chave pública

Criptoanálise



- Vúlnerável a força bruta
  - Contramedida natural é o aumento do tamanho da chave
  - Entretanto quanto maior a chave menor o desenpenho do processo de cifragem e decifragem.
- Encontrar a chave privada a partir da chave pública
- Ataque de mensagem conhecida.
  - Reduçao do problema ao número de mensagens possíveis

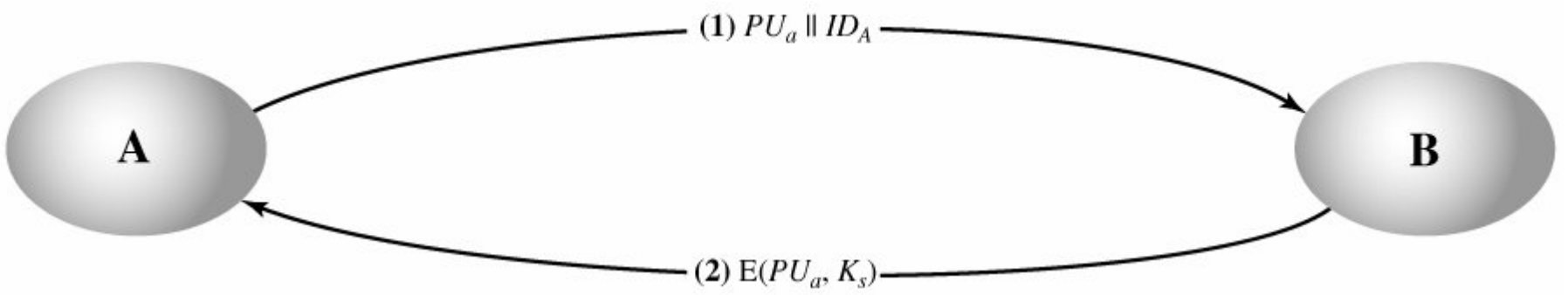
Criptografia de chave pública
Distribuição de chaves secretas (sessão)



- A distribuição de chaves públicas impede a interceptação e violação de informações.
- Desempenho dos protocolos de chave pública quando aplicados a troca de mensagens em grande volume não é satisfatório.
- Utilizamos o esquema de chave pública para transferir de modo seguro uma chave de sessão comum.

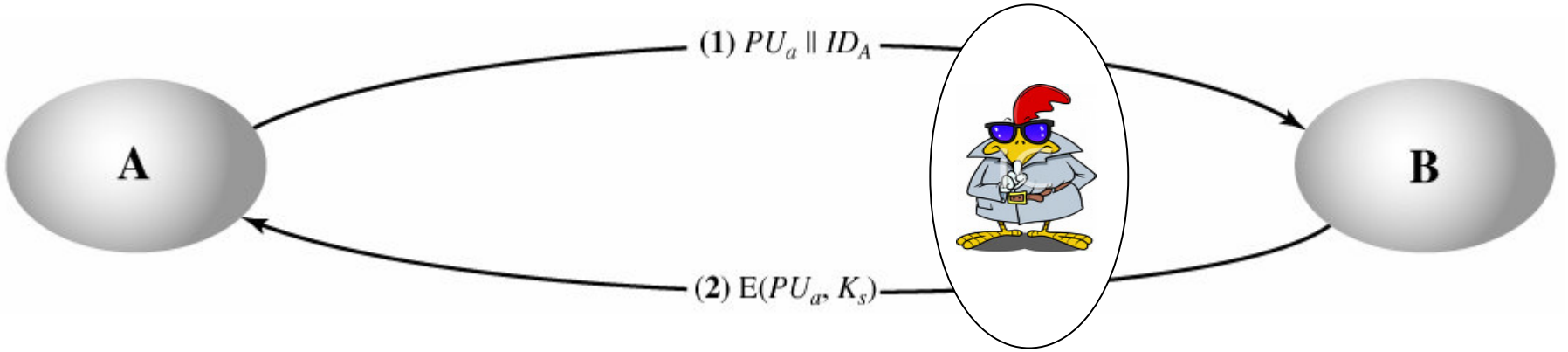
Criptografia de chave pública
Distribuição de chaves secretas (sessão)

- Distribuição simples de chave (Merkle 79).
  - 1) A calcula PUa e envia a B (PUa || IDa)e
  - 2) B gera Ks e envia para A.
  - 3) A calcula D(PRa, E(Pua,Ks))
  - 4) A e B trocam mensagens usando Ks



Criptografia de chave pública
Distribuição de chaves secretas (sessão)

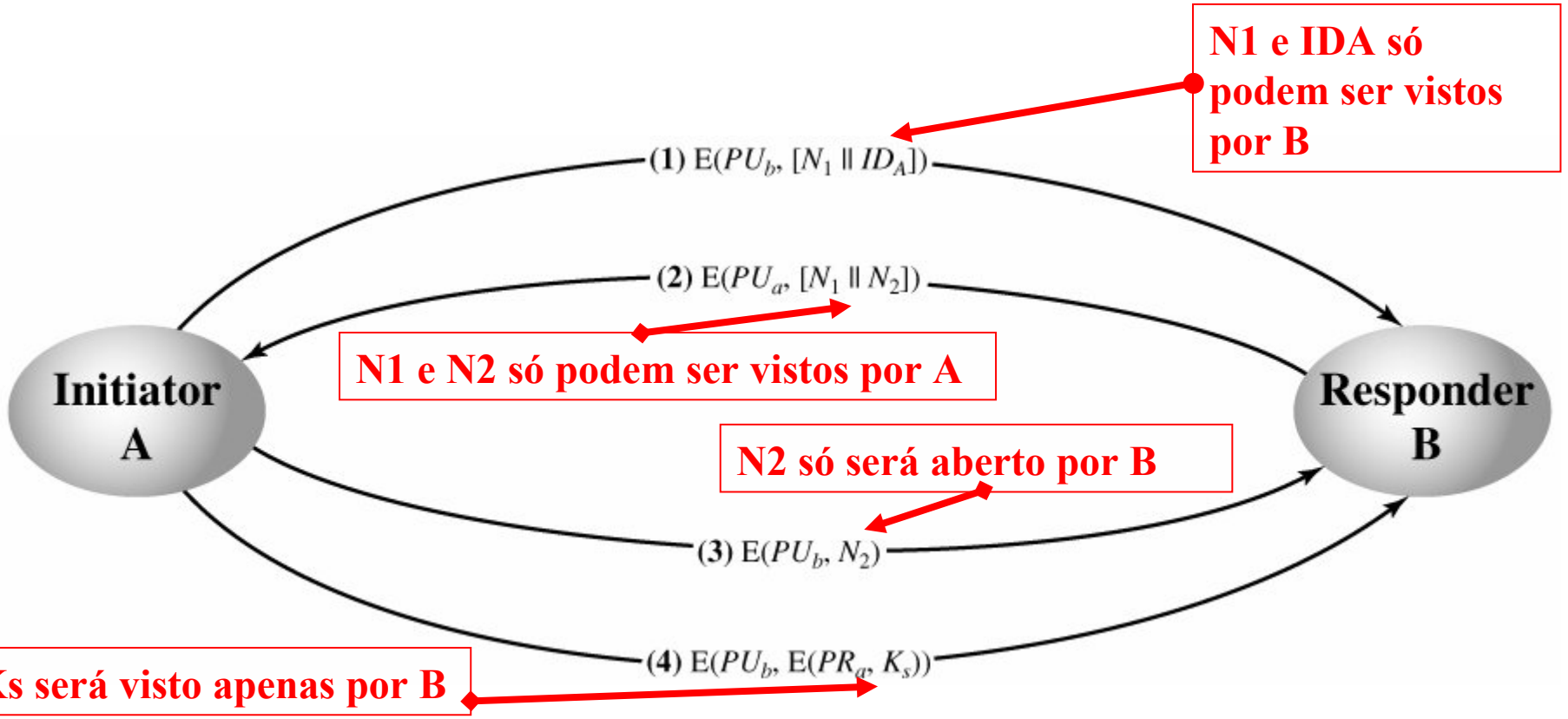
- **PROBLEMA**: O esquema proposto por Merkle é vulnerável ao ataque *man-in-the-middle*:
  - 1) Bisbilhoteiro intercepta a mensagem inicial de A e envia a B : (PUbisbilhteiro, IDA).
  - 2) B gera Ks e envia E(PUbisbilhoteiro,Ks)
  - 3) Bibilhoteiro agora tem Ks e o transmite para A: E(Pua,Ks)



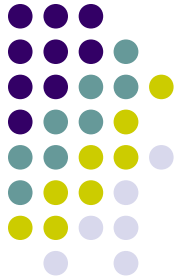
Criptografia de chave pública
Distribuição de chaves secretas (sessão)

- **Solução 1 : (NEED 78)**

  1) Assumindo que ambos já conheçam suas chaves públicas.



Criptografia de chave pública
Distribuição de chaves secretas (sessão)



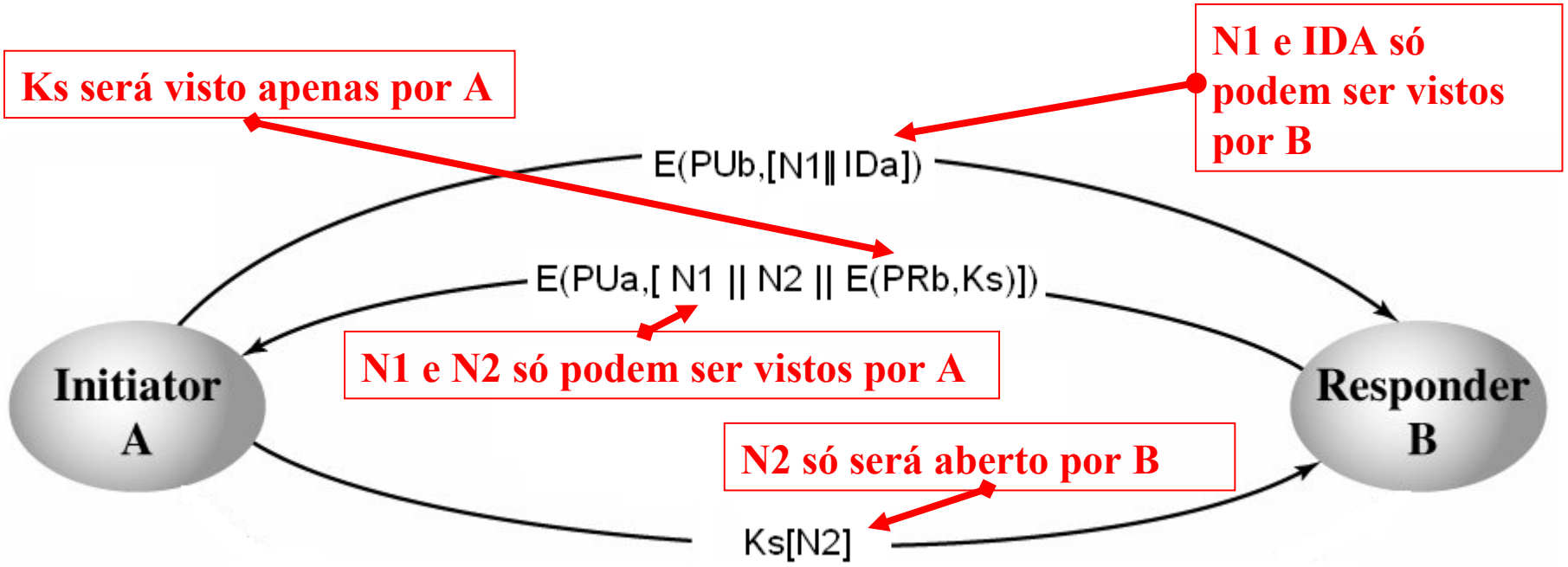
- Exercício :

1) Formule um processo de troca de chaves otimizado para a solução 1 proposta.

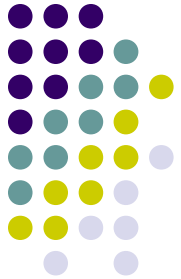
Criptografia de chave pública
Distribuição de chaves secretas (sessão)

- **Solução 2:**

  2) Um esquema mais enxuto. A é cliente e B é um servidor.



Criptografia de chave pública
Distribuição de chaves secretas (sessão)

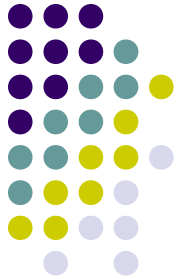


- Os esquemas apresentados precisam necessariamente de um cálculo de um par de chaves e também sua publicação em algum local.

- Um ***esquema alternativo*** para a troca de chaves de sessão pode ser alcançado através do mesmo princípio da criptografia de chave pública, porém sem a necessidade do cálculo e distribuição do par de chaves dos pares envolvidos na comunicação.

Criptografia de chave pública

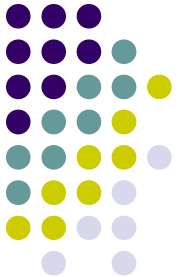
Acordo de chaves Diffie-Hellman



- Esse esquema deu origem a criptografia de chave pública [1976].
- Algoritmo permite acordar sobre um valor de chave comum entre 2 participantes.
- Funcionalidade restrita apenas para troca de chaves.

Criptografia de chave pública

Acordo de chaves Diffie-Hellman



- Fundamentação:
  - Consiste na dificuldade em computar logaritmos discretos.
  - Sendo ***a*** *uma raiz primitiva de um primo* **p** *e* ***b*** *um inteiro qualquer, existe um expoente* ***i*** *único que satifaz a seguinte condição:*

$$b \equiv a^i \pmod{p} \text{ em que } 0 \leq i \leq (p-1)$$

A raiz primitiva de um primo p é aquele cuja as potências módulo p geram todos os inteiros entre 1 até p-1 (em Z_p^*). Onde os números: a mod p, a^2 mod p, …, a^{p-1} mod p; são distintos.

Criptografia de chave pública

Acordo de chaves Diffie-Hellman

- Regra:

  $K = (Yb)^{Xa} \equiv (Ya)^{Xb} \bmod q$

- Existem dois números públicos:
  - Um primo q
  - Uma raiz primitiva de q : α
- ***A*** seleciona Xa < q (priv.) e calcula
  - $Ya = \alpha^{Xa} \bmod q$ (pública)
- ***B*** faz o mesmo Xb < q :
  - $Yb = \alpha^{Xb} \bmod q$ (pública)

Global Public Elements

q	prime number
α	$\alpha < q$ *and* α a primitive root of q

User A Key Generation

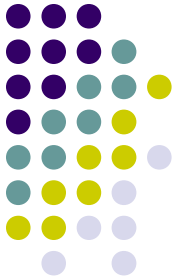
Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{X_A} \bmod q$

User B Key Generation

Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{X_B} \bmod q$

Criptografia de chave pública

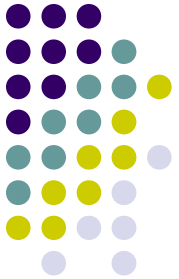
Acordo de chaves Diffie-Hellman



- Para finalizar o processo, A e B trocam suas chaves públicas e calculam uma chave de sessão K:
- A calcula:
  - $K = (Yb)^{Xa} \mod q$
- B calcula:
  - $K = (Ya)^{Xb} \mod q$

Criptografia de chave pública

Acordo de chaves Diffie-Hellman

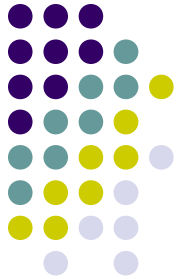


Exemplo:

- Sejam q = 353 e α= 3:
  - A seleciona Xa = 97 < 353
  - B seleciona Xb = 233 < 353
- **A** e **B** Calculam:
  - Ya = $3^{97} \bmod 353 =$ **40**
  - Yb = $3^{233} \bmod 353 =$ **248**
- Depois
  - A : K = $248^{97} \bmod 353 =$ **160**
  - B : K = $40^{233} \bmod 353 =$ **160**

Criptografia de chave pública

Acordo de chaves Diffie-Hellman

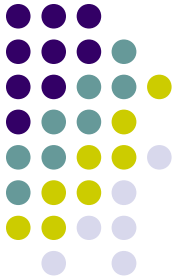


Exercício:

1) Calcule a Chave K, assumindo que q = 353, α= 3, Xa= 98 e Xb = 234.

Criptografia de chave pública

Acordo de chaves Diffie-Hellman

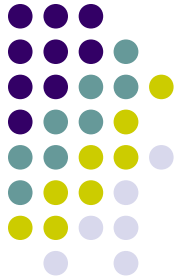


Exercício 1:

- Sejam q = 353 e α= 3:
  - A seleciona Xa = 98 < 353
  - B seleciona Xb = 234 < 353
- **A** e **B** Calculam:
  - Ya = $3^{98} \bmod 353 = 120$
  - Yb = $3^{234} \bmod 353 = 38$
- Depois
  - A : K = $(38)^{98} \bmod 353 =$ **336**
  - B : K = $(120)^{234} \bmod 353 =$ **336**

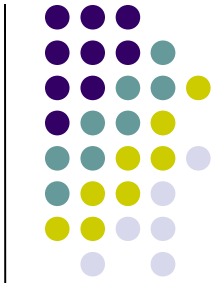
Criptografia de chave pública

Acordo de chaves Diffie-Hellman



- ***Quebra da Chave :***
- *Um atacante pode calcular Xa e Xb visto que :*
  - *q =353, α = 3, Ya = 40 e Yb = 248 : são conhecidos*
- *Por :*
  - $3^{Xa} \bmod 353 = 40$ ou $3^{Xb} \bmod 353 = 248$
  - *Nesse caso é possível estimar os valores através de força bruta*
- Porém a complexidade em se determinar Xa e Xb com números primos grandes torna esse processo impraticável.

Atividade



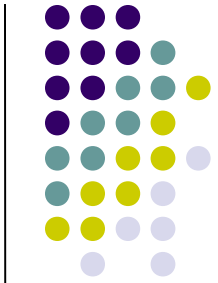
- Desenvolva um método/classe que implemente a de troca de chaves de sessão baseado no modelo de Diffie e Hellman para a aplicação de chat seguro.

Prática 3

- Implemente um algoritmo de força bruta para encontrar a chave K utilizada para a criptografia de mensagens utilizadas no Chat_Seguro com RC4.

- Busca por palavras ou mensagens conhecidas:

- Por convenção assuma que as mensagens trocadas no Chat será utilizado o seguinte conjunto de strings conhecidas:



Ola; Como vai?; Bom dia!; Boa Tarde.; Tudo bem?; Sim.; Nao.; Adeus; Ate logo; lol; :); :(; ok; td bem; sem problemas; tchau; reuniao; me liga; obrigado!; de nada; muito obrigado; :0;.

- Simule a captura de algumas conversas entre dois grupos e aplique o algoritmo de força bruta para encontrar o valor da chave do RC4.