Trường Đại Học Công Nghệ Thông Tin
Khoa Mạng Máy Tính và Truyền Thông

AN TOÀN MẠNG MÁY TÍNH

ThS. Tô Nguyễn Nhật Quang

NỘI DUNG MÔN HỌC

1. Tổng quan về an ninh mạng
2. Các phần mềm gây hại
3. Các giải thuật mã hoá dữ liệu
4. Mã hoá khoá công khai và quản lý khoá
5. Chứng thực dữ liệu
6. Một số giao thức bảo mật mạng
7. Bảo mật mạng không dây
8. Bảo mật mạng vành đai
9. Tìm kiếm phát hiện xâm nhập

BÀI 4

MÃ HOÁ
KHOÁ CÔNG KHAI
& QUẢN LÝ KHOÁ

Mã hoá khoá công khai và quản lý khoá

1. Số nguyên tố
2. Hệ mã hoá khoá công khai
3. Giao thức trao đổi khoá Diffie-Hellman
4. Hệ RSA
5. Quản lý khoá
6. Bài tập

1. Số nguyên tố

- **Giới thiệu**
  - Bất kỳ số nguyên a > 1 đều có thể viết dưới dạng:

    $a = p_1^{a1} p_2^{a2} p_3^{a3} \ldots p_t^{at}$

    trong đó $p_1 < p_2 < \ldots < p_t$ là các số nguyên tố.

    Ví dụ:

    $85 = 5 \times 17$

    $91 = 7 \times 13$

    $1200 = 2^4 \times 3 \times 5^2$

    $11011 = 7 \times 11^2 \times 13$

1. Số nguyên tố

- **Giới thiệu**
  - Một số nguyên p> 1 là số nguyên tố nếu và chỉ nếu ước duy nhất của nó là ± 1 và ± p.
  - Số nguyên tố đóng vai trò quan trọng trong lý thuyết số và trong các kỹ thuật mã hoá khoá công khai thảo luận trong chương này.
  - Bảng dưới đây trình bày các số nguyên tố nhỏ hơn 2000.

1. Số nguyên tố

2	101	211	307	401	503	601	701	809	0	1009	1103	1201	1301	1409	1511	1601	1709	1801	1901
3	103	223	311	409	509	607	709	811	911	1013	1109	1213	1303	1423	1523	1607	1721	1811	1907
5	107	227	313	419	521	613	719	821	919	1019	1117	1217	1307	1427	1531	1609	1723	1823	1913
7	109	229	317	421	523	617	727	823	929	1021	1123	1223	1319	1429	1543	1613	1733	1831	1931
11	113	233	331	431	541	619	733	827	937	1031	1129	1229	1321	1433	1549	1619	1741	1847	1933
13	127	239	337	433	547	631	739	829	941	1033	1151	1231	1327	1439	1553	1621	1747	1861	1949
17	131	241	347	439	557	641	743	839	947	1039	1153	1237	1361	1447	1559	1627	1753	1867	1951
19	137	251	349	443	563	643	751	853	953	1049	1163	1249	1367	1451	1567	1637	1759	1871	1973
23	139	257	353	449	569	647	757	857	967	1051	1171	1259	1373	1453	1571	1657	1777	1873	1979
29	149	263	359	457	571	653	761	859	971	1061	1181	1277	1381	1459	1579	1663	1783	1877	1987
31	151	269	367	461	577	659	769	863	977	1063	1187	1279	1399	1471	1583	1667	1787	1879	1999
37	157	271	373	463	587	661	773	877	983	1069	1193	1283		1481	1597	1669	1789	1889	1997
41	163	277	379	467	593	673	787	881	991	1087		1289		1483		1693			1999
43	167	281	383	479	599	677	797	883	997	1091		1291		1487		1697			
47	173	283	389	487		683		887		1093		1297		1489		1699			
53	179	293	397	491		691				1097				1493					
59	181			499										1499					
61	191																		
67	193																		
71	197																		
73	199																		
79																			
83																			
89																			
97																			

1. Số nguyên tố

- **Thuật toán tìm dãy số nguyên tố nhỏ hơn n -** dùng thuật toán của nhà toán học Hy lạp Eratosthenes.
  - Liệt kê tất cả các số nguyên từ 2 đến n.
  - Số đầu tiên (2) là số nguyên tố.
  - Loại tất cả các bội của 2 ra khỏi bảng.
  - Số nguyên ngay sau số 2 sau khi loại (sàng) là số nguyên tố (số 3).
  - Loại bỏ tất cả các bội của 3.
  - ...
  - Khi tìm được một số nguyên tố lớn hơn căn bậc 2 của n, tất cả các số còn lại không bị loại ra đều là số nguyên tố.

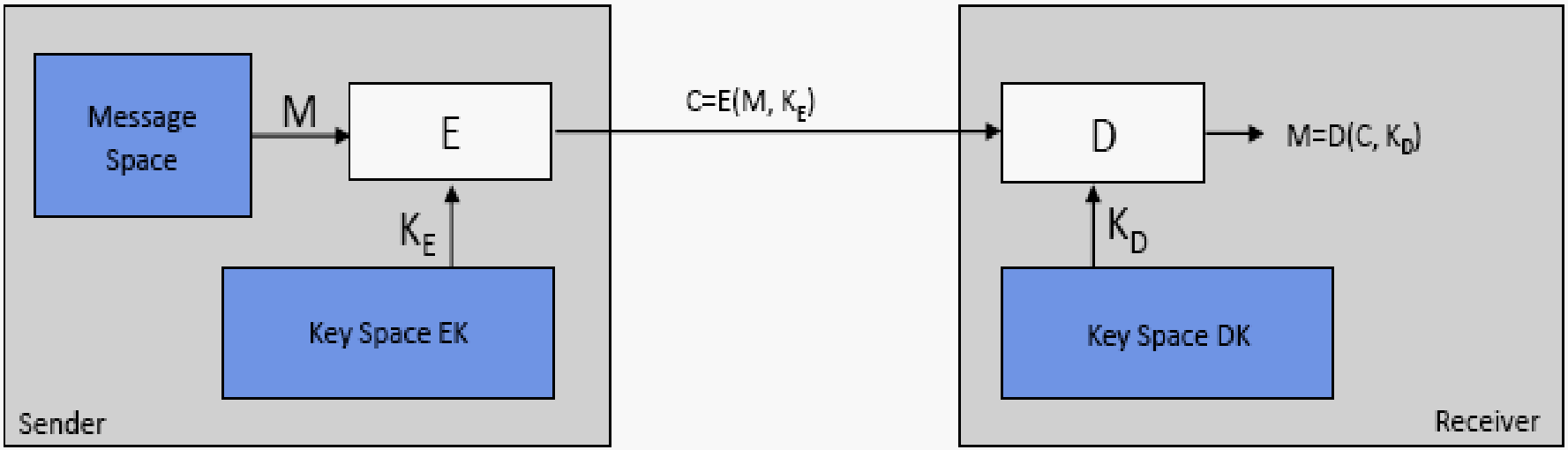
1. Số nguyên tố

- **Thuật toán tìm dãy số nguyên tố nhỏ hơn n:**

```
L = {2, 3, ..., n};
i = 1;
While (L[i]^2 <= n) Do {
        If (L[i] <> 0)
                k = i^2 + 2i;
                While (k <= n) Do {
                        L[k] = 0;
                        k = k + i;
                }
        i++;
}
```

2. Hệ mã hoá khoá công khai

- Được xây dựng trên ý tưởng hàm một chiều.



a) Symmetric Encryption: $K_E = K_D$ (e.g. AES)

b) Asymmetric Encryption: $K_E \neq K_D$ (e.g. RSA)

secret

public

private/secret

2. Hệ mã hoá khoá công khai

Các bước chủ yếu khi thực hiện mã hoá khoá công khai:

1. Mỗi user tạo ra một cặp khoá được sử dụng cho việc mã hoá và giải mã thông điệp.
2. Mỗi user đặt một trong hai khoá trong một đăng ký công cộng. Đây là khoá công khai. Khoá còn lại được giữ kín.
3. Nếu Bob muốn gửi một tin nhắn bí mật cho Alice, Bob mã hoá tin nhắn này bằng cách sử dụng khoá công khai của Alice.
4. Khi Alice nhận được tin nhắn, cô giải mã nó bằng cách sử dụng khoá riêng của mình. Không có ai khác có thể giải mã thông điệp bởi vì chỉ có Alice biết khoá riêng của Alice.

2. Hệ mã hoá khoá công khai

- **Lịch sử hình thành**:
  - Năm 1976, Whitfield Diffie và Martin Hellman công bố một hệ thống mật mã hoá khoá bất đối xứng trong đó nêu ra phương pháp trao đổi khóa công khai.
  - Trao đổi khoá Diffie-Hellman là phương pháp có thể áp dụng trên thực tế đầu tiên để phân phối khoá bí mật thông qua một kênh thông tin không an toàn.

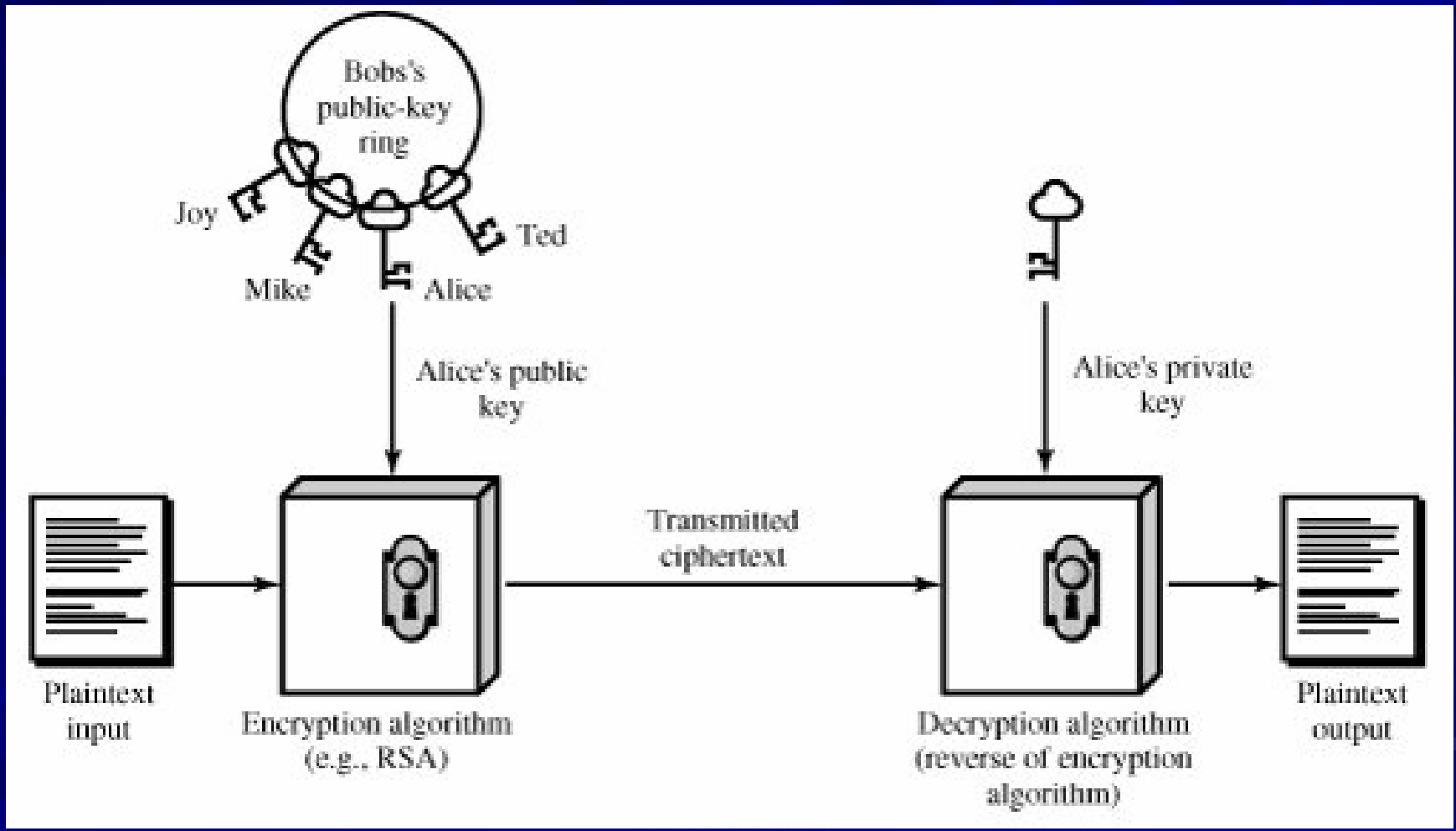
2. Hệ mã hoá khoá công khai

- **Lịch sử hình thành**:
  - Thuật toán đầu tiên được Rivest, Shamir và Adleman tìm ra vào năm 1977 tại MIT. Công trình này được công bố vào năm 1978 và thuật toán được đặt tên là RSA.
  - RSA sử dụng phép toán tính hàm mũ môđun (môđun được tính bằng tích số của 2 số nguyên tố lớn) để mã hóa và giải mã cũng như tạo chữ ký số. An toàn của thuật toán được đảm bảo với điều kiện là không tồn tại kỹ thuật hiệu quả để phân tích một số rất lớn thành thừa số nguyên tố.

2. Hệ mã hoá khoá công khai

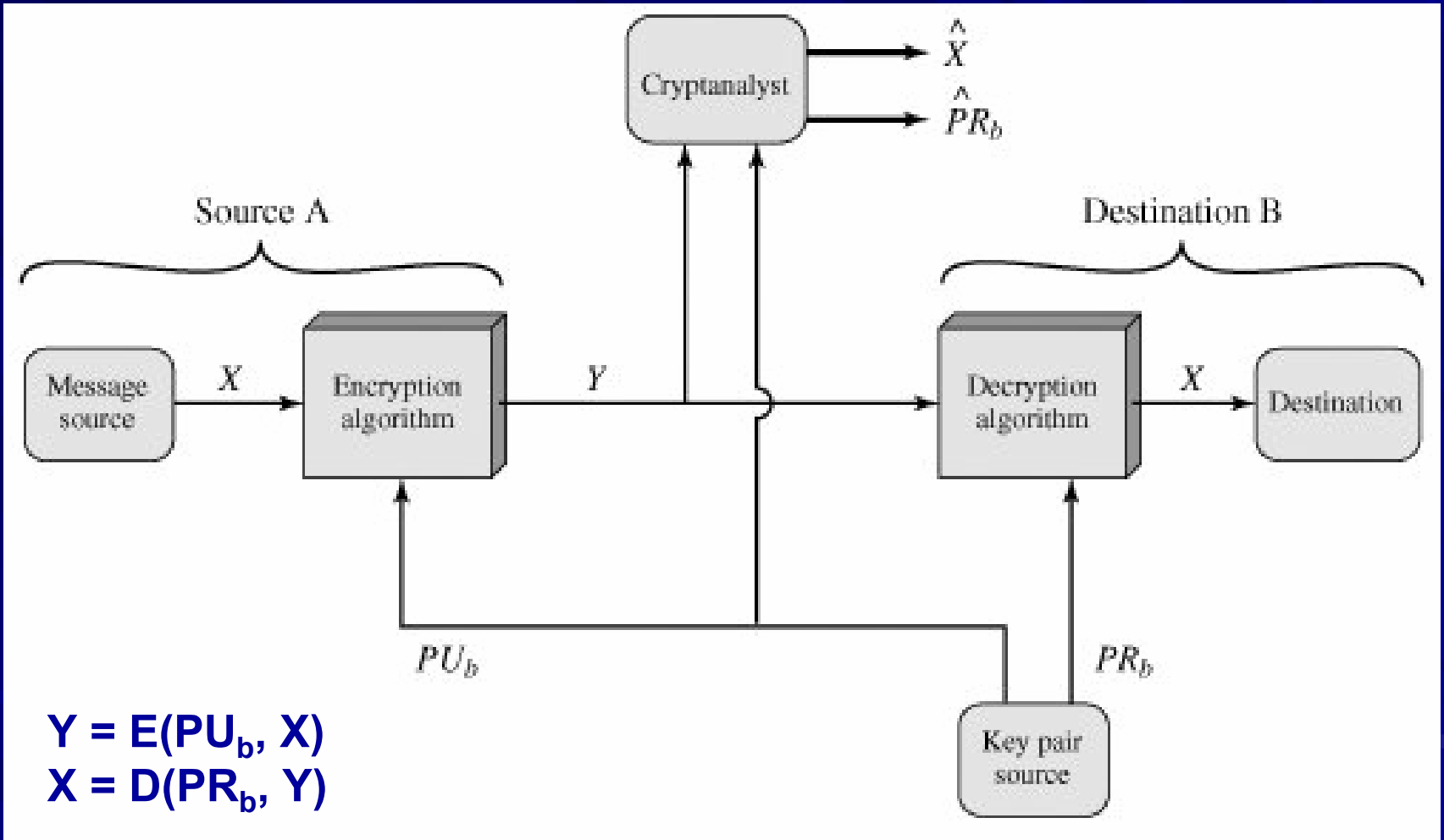
- **Ứng dụng**:
  - Ứng dụng thông dụng nhất của mật mã hoá khoá công khai là bảo mật (mã hoá/giải mã): một văn bản được mã hoá bằng **khoá công khai** của một người sử dụng thì chỉ có thể giải mã với **khoá bí mật** của người đó.

2. Hệ mã hoá khoá công khai



Encryption

2. Hệ mã hoá khoá công khai



$Y = E(PU_b, X)$

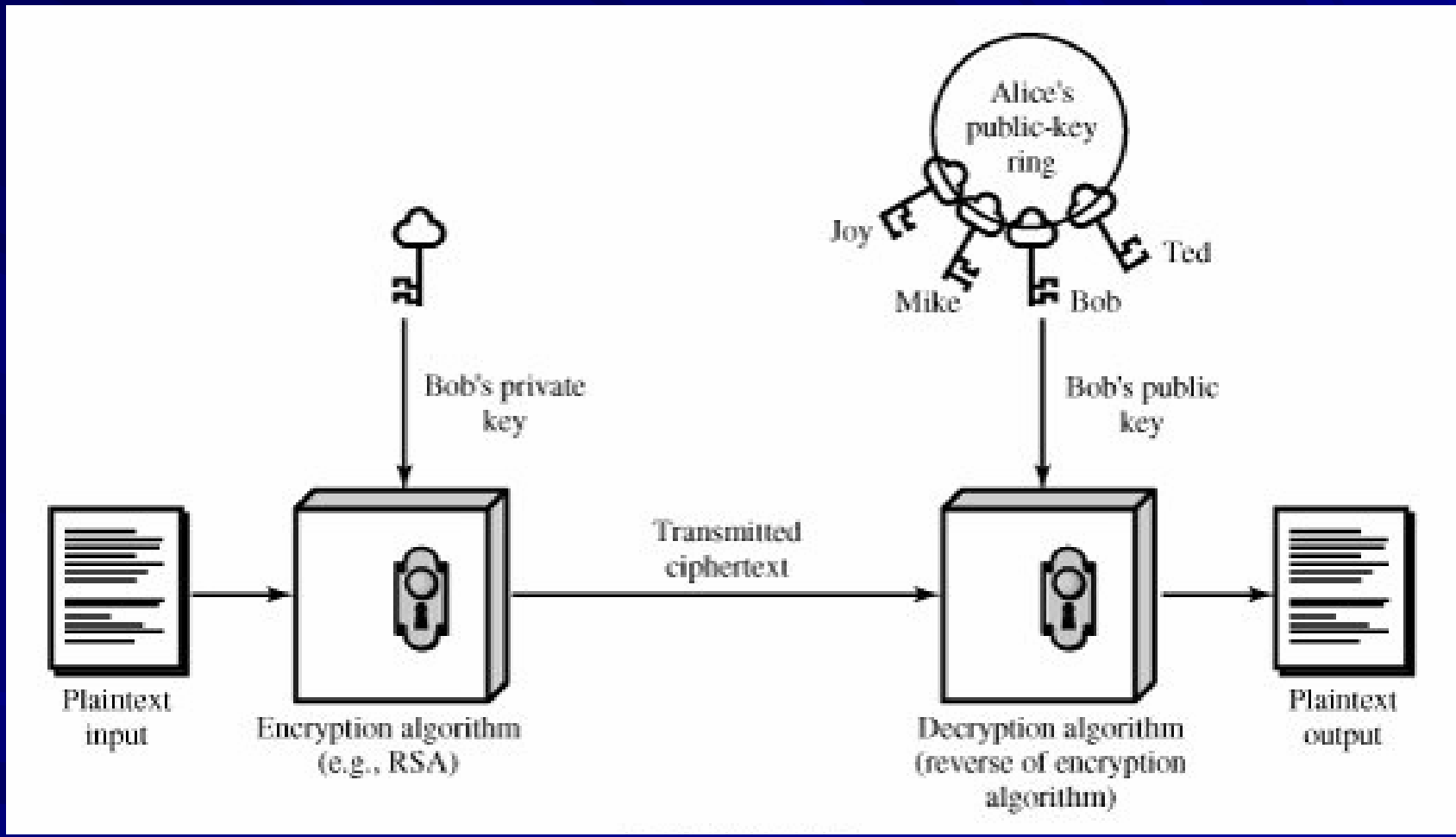
$X = D(PR_b, Y)$

Secrecy

2. Hệ mã hoá khoá công khai

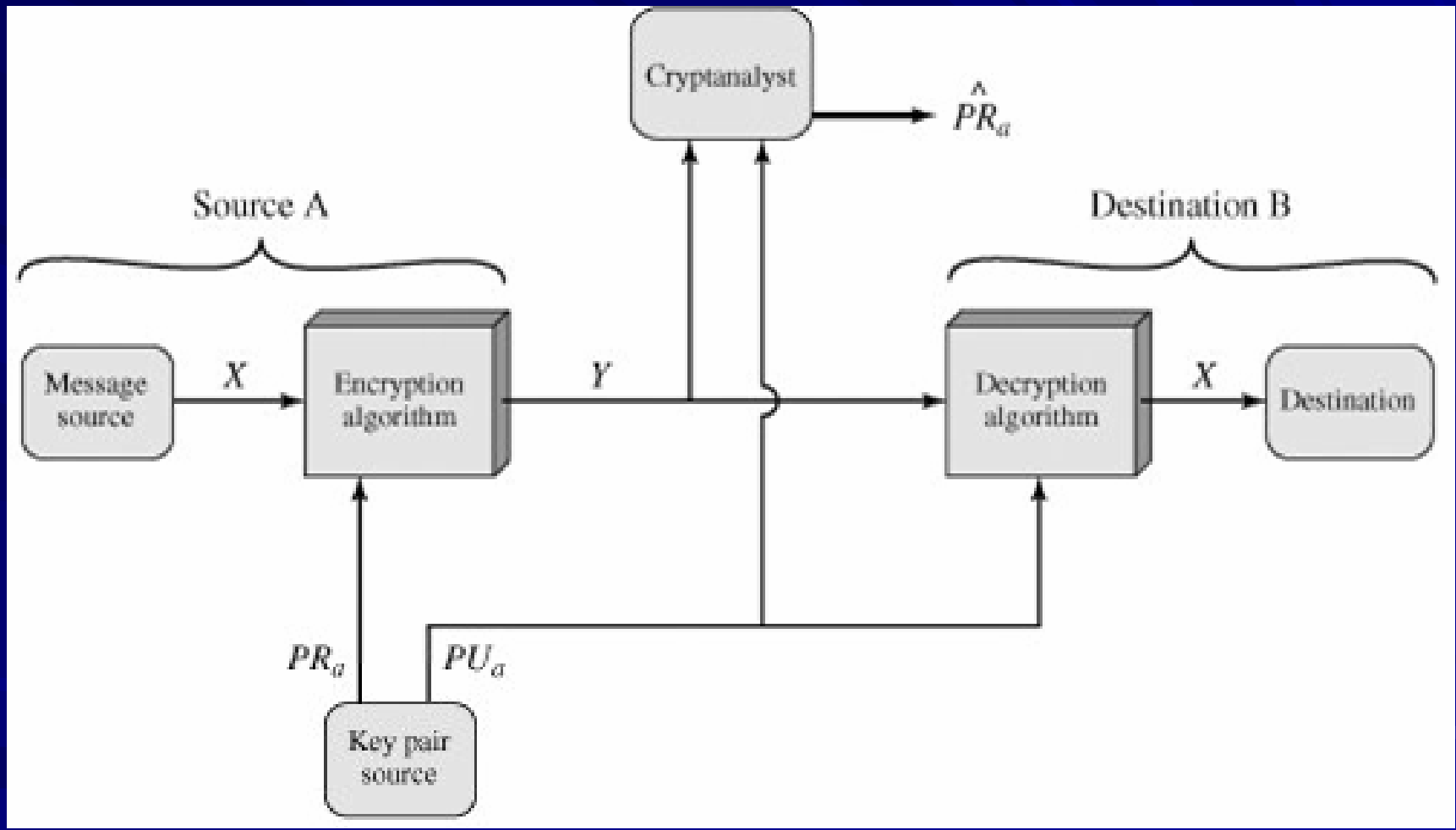
- **Ứng dụng**:
  - Các thuật toán tạo chữ ký số khoá công khai có thể dùng để chứng thực: Một người sử dụng có thể mã hoá văn bản với khoá bí mật của mình. Nếu một người khác có thể giải mã với **khoá công khai** của người gửi thì có thể tin rằng văn bản thực sự xuất phát từ người gắn với khoá công khai đó.

2. Hệ mã hoá khoá công khai



Authentication

2. Hệ mã hoá khoá công khai

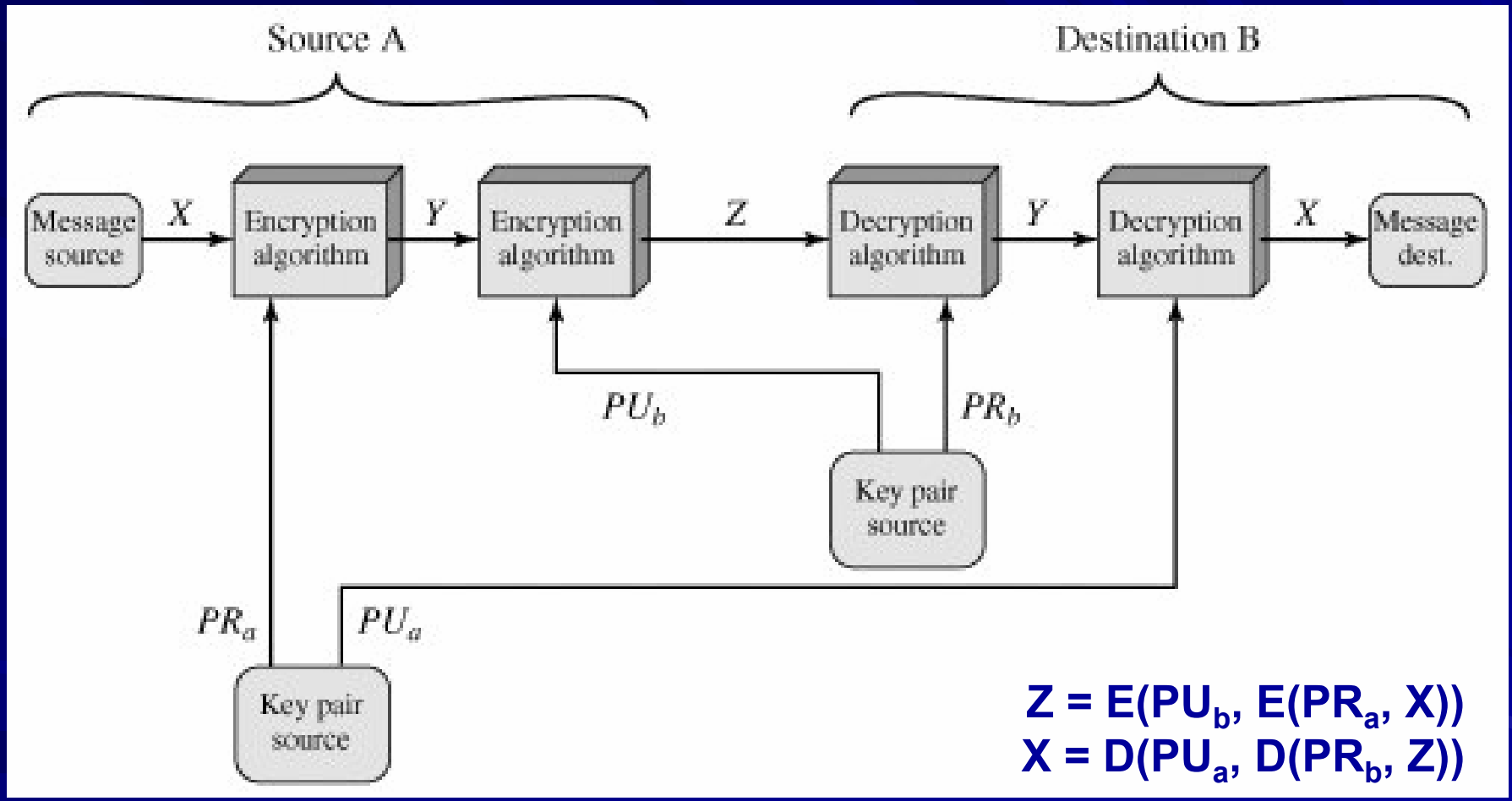


Authentication

2. Hệ mã hoá khoá công khai

- **Ứng dụng**:
  - Trao đổi khoá: Hai bên hợp tác để trao đổi session key. Có một số phương pháp tiếp cận khác nhau liên quan đến các khóa bí mật của một hoặc cả hai bên.
    - Trước tiên, mã hoá thông điệp X sử dụng khoá secret của người gởi (cung cấp chữ ký số) để được Y.
    - Kế đó, mã hoá tiếp Y với khoá public của người nhận.
    - Chỉ có người nhận đã xác định trước mới có khoá secret của người nhận và khoá public của người gởi để giải mã hai lần để được X.

2. Hệ mã hoá khoá công khai



Authentication và Secrecy

2. Hệ mã hoá khoá công khai

- Một số giải thuật hệ mã hoá khoá công khai

Algorithm	Encryption/ Decryption	Digital Signature	Key Exchange
RSA	x	x	x
Elliptic Curve	x	x	x
Diffie-Hellman			x
DSS		x	

2. Hệ mã hoá khoá công khai

- **Định nghĩa**:

  Cho các tập hữu hạn S và T.

  Hàm một chiều f: S $\rightarrow$ T là hàm khả nghịch thoả:

  - f dễ thực hiện; cho x $\in$ S, dễ dàng tính được y = f(x).
  - f^{-1} là hàm ngược của f, khó thực hiện; cho y $\in$ T, rất khó tính được x = $f^{-1}(y)$.
  - f^{-1} chỉ có thể tính được khi biết thêm một số thông tin cần thiết.

2. Hệ mã hoá khoá công khai

- **Ví dụ:**

  f: pq $\rightarrow$ n là hàm một chiều với p và q là các số nguyên tố lớn.

  - Có thể dễ dàng thực hiện phép nhân pq (độ phức tạp đa thức).
  - Tính f^{-1} (phân tích ra thừa số nguyên tố - độ phức tạp mũ) là bài toán cực kỳ khó.

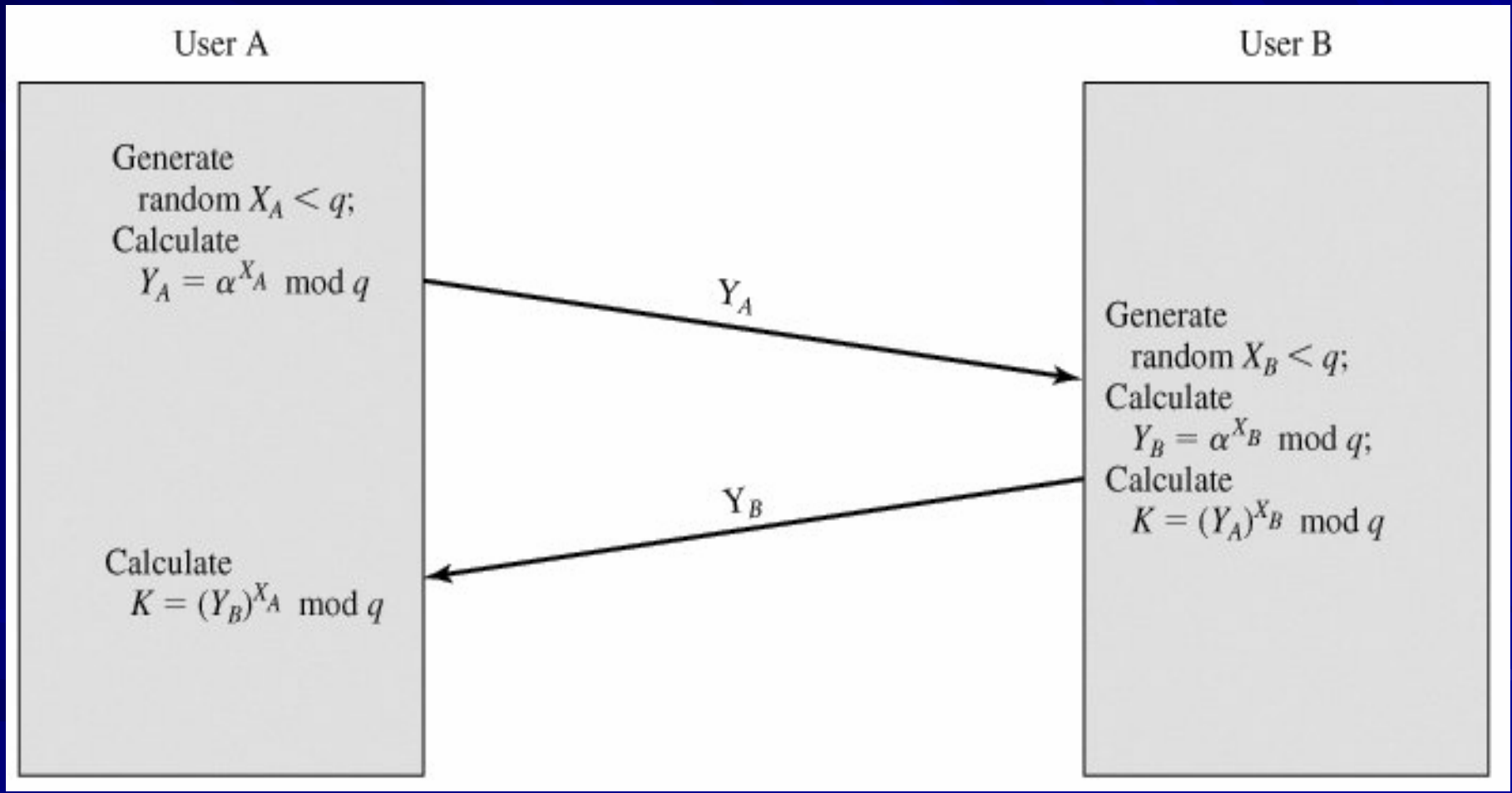
3. Giao thức trao đổi khoá Diffie-Hellman

- Mục đích của thuật toán là cho phép hai người dùng trao đổi khóa bí mật dùng chung trên mạng công cộng, sau đó có thể sử dụng để mã hóa các thông điệp.
- Thuật toán tập trung vào giới hạn việc trao đổi các giá trị bí mật, xây dựng dựa trên bài toán khó logarit rời rạc.

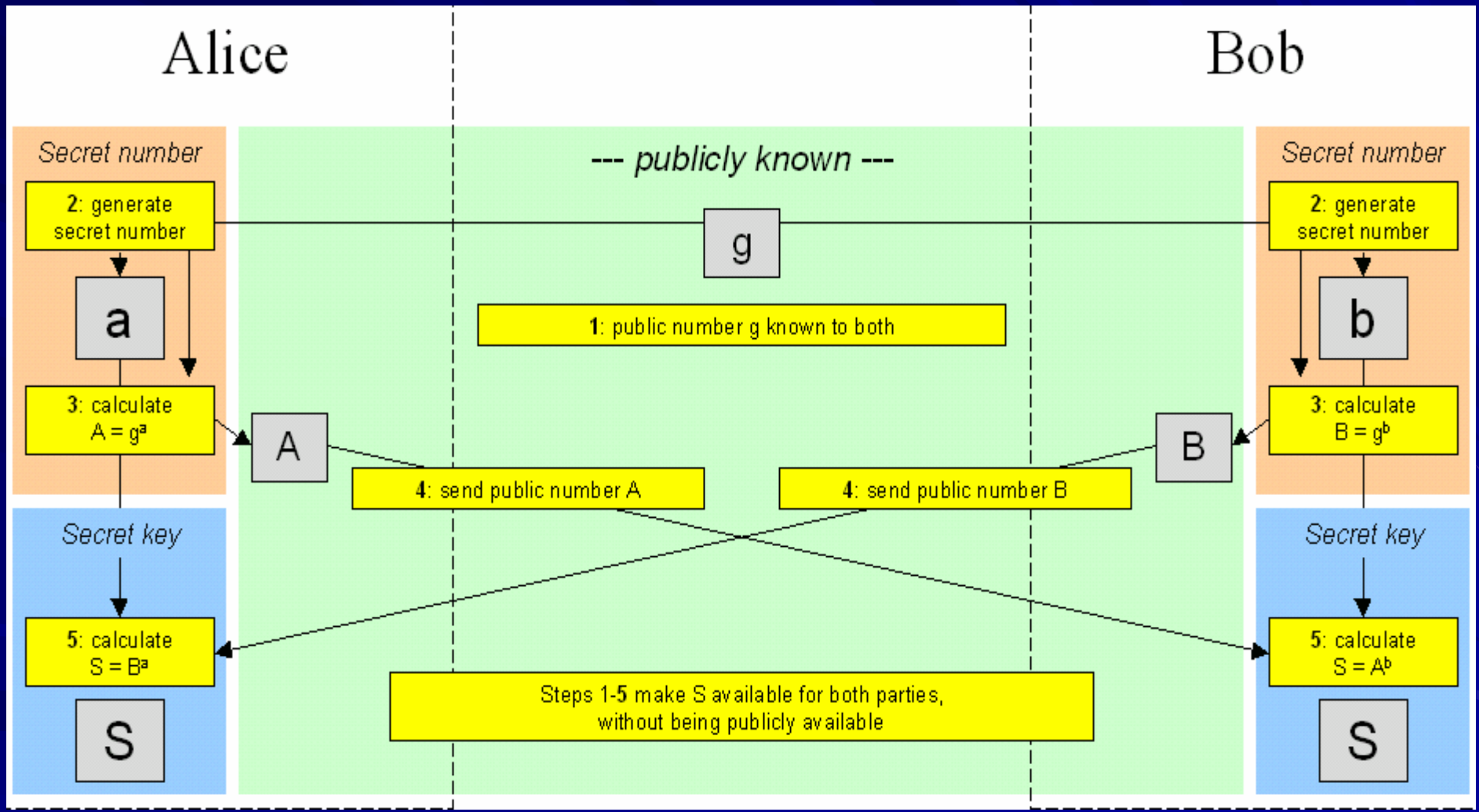
3. Giao thức trao đổi khoá Diffie-Hellman

- Giao thức trao đổi khoá giữa A và B:
  - A và B thống nhất chọn chung một số nguyên tố q và một phần tử sinh α.
  - A chọn ngẫu nhiên một số $X_A \in \{1, 2, ..., q-1\}$ rồi gởi cho B kết quả $Y_A = \alpha^{XA} \bmod q$.
  - B chọn ngẫu nhiên một số $X_B \in \{1, 2, ..., q-1\}$ rồi gởi cho A kết quả $Y_B = \alpha^{XB} \bmod q$.
  - A tính khoá bí mật: $K=(\alpha^{XB})^{XA} \bmod q = \alpha^{XAXB} \bmod q$
  - B tính khoá bí mật: $K=(\alpha^{XA})^{XB} \bmod q = \alpha^{XAXB} \bmod q$

3. Giao thức trao đổi khoá Diffie-Hellman



3. Giao thức trao đổi khoá Diffie-Hellman



3. Giao thức trao đổi khoá Diffie-Hellman

- Ví dụ:
  - A và B chọn số nguyên tố chung là 353 và phần tử sinh g là 3.
  - A chọn X_A=97 rồi gởi cho B giá trị kết quả của 3^{97} mod 353 = 40.
  - B chọn X_B=233 rồi gởi cho A giá trị kết quả của 3^{233} mod 353 = 248.
  - Cả A và B đều tính được K = 248^{97} mod 353 = 160 = 40^{233} mod 353.

4. Hệ RSA

- Giải thuật được phát triển bởi Rivest, Shamir và Adleman này sử dụng một biểu thức với hàm mũ.
- Văn bản rõ được mã hóa ở dạng khối, kích cỡ của khối phải nhỏ hơn hoặc bằng $\log_2(n)$.
- Trong thực tế, kích thước khối là i bit, với $2^i < n <= 2^{i+1}$.
- Mã hóa và giải mã được thực hiện với một số khối rõ M (plaintext) và khối mã C (cyphertext):

$C = M^e \bmod n$

$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$

4. Hệ RSA

- Giải thuật:
  - Mã hoá:
    - Từ khoá công khai (n, e) và thông điệp là plaintext dưới dạng số nguyên $M \in [0, n)$.
    - Tính cyphertext $C = M^e \bmod n$
  - Giải mã:
    - $M = C^d \bmod n$, với d là khoá bí mật.

4. Hệ RSA

- Cả người gửi và người nhận phải biết giá trị của n. Người gửi biết giá trị của e, và chỉ người nhận mới biết giá trị của d.
- Như vậy, đây là một thuật toán mã hoá khoá công khai với một khóa công khai PU={n, e} và một khoá riêng PU={d, n}.
- Các yêu cầu sau đây phải được đáp ứng:
  - Phải có khả năng tìm được giá trị của e, d, n sao cho M^{ed} mod n = M, với M < n.
  - Phải dễ dàng tính toán được mod M^e mod n và C^d cho tất cả các giá trị của M < n.
  - Nó là không khả thi để xác định d khi cho e và n.
  - Để an toàn, RSA đòi hỏi p và q phải là các số nguyên tố rất lớn để không thể phân tích được n=pq.

4. Hệ RSA

Key Generation

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p - 1)(q - 1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

4. Hệ RSA

Encryption

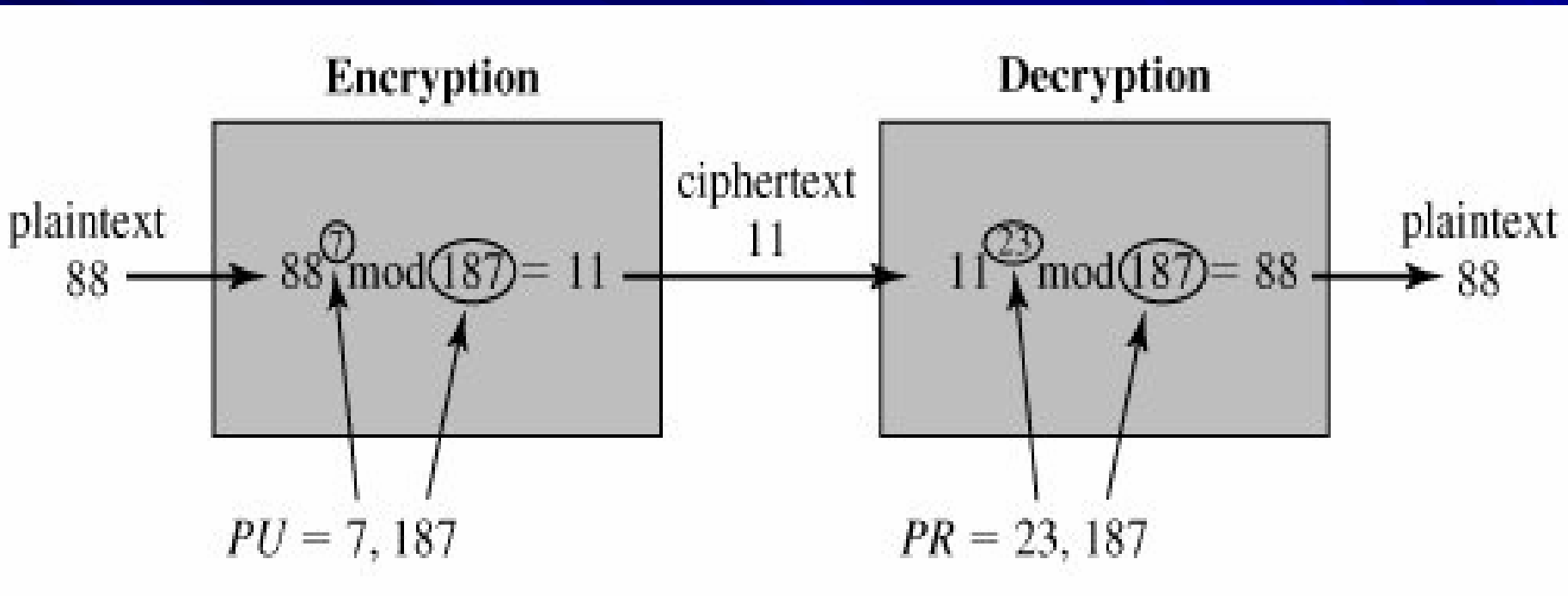
Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption

Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

4. Hệ RSA

Ví dụ:


Encryption
Decryption
plaintext
88
$88^{7} \bmod 187 = 11$
ciphertext
11
$11^{23} \bmod 187 = 88$
plaintext
88
$PU = 7, 187$
$PR = 23, 187$

4. Hệ RSA

- **Tính 88^7 mod 187**
  - 88^7 mod 187 = [(88^4 mod 187) x (88^2 mod 187) x (88^1 mod 187)] mod 187
  - 88^1 mod 187 = 88
  - 88^2 mod 187 = 7744 mod 187 = 77
  - 88^4 mod 187 = 59,969,536 mod 187 = 132
  - 88^7 mod 187 = (88 x 77 x 132) mod 187 = 894,432 mod 187 = **11**

4. Hệ RSA

- **Tính 11^{23} mod 187**
  - 11^{23} mod 187 = [(11^1 mod 187) x (11^2 mod 187) x (11^4 mod 187) x (11^8 mod 187) x (11^8 mod 187)] mod 187
  - 11^1 mod 187 = 11
  - 11^2 mod 187 = 121
  - 11^4 mod 187 = 14,641 mod 187 = 55
  - 11^8 mod 187 = 214,358,881 mod 187 = 33
  - 11^{23} mod 187 = (11 x 121 x 55 x 33 x 33) mod 187 = 79,720,245 mod 187 = 88

4. Hệ RSA

- **Ví dụ**:

  Cho các số nguyên tố p=2357 và q=2551.

  Tính được:

  n = pq = 6012707

  $\phi(n) = (p-1)(q-1) = 6007800$

  Chọn số nguyên e $\in (1, \phi(n))$ là 3674911

  $d \equiv e^{-1} \bmod \phi(n) = 422191$

  Khoá công khai: (n, e) = (6012707, 3674911)

  Khoá bí mật: d = 422191

4. Hệ RSA

- **Ví dụ**:

Để mã hoá bản rõ

$M = 5234673 \in [0, 6012707)$

tính $C = M^e \bmod n = 3650502$

Để giải mã

tính $C^d \bmod n = 5234673$

5. Quản lý khoá

1. *Thẩm quyền thu hồi khoá*

- Thu hồi khoá khi khoá bị sai sót hoặc có tính phá hoại.
- Thường được tham gia bởi từ hai thực thể trở lên. Ví dụ: cả Alice và Bob cùng thoả thuận thu hồi khoá.
- Cần đảm bảo:
  - Càng nhiều bên tham gia càng tốt (chống phá hoại).
  - Càng ít bên tham gia càng tốt (thu hồi nhanh).

5. Quản lý khoá

2. *Phân phối khoá mới*

- Phải phân phối khoá mới sau khi khoá cũ bị thu hồi nhằm đảm bảo hệ thống tiếp tục hoạt động một cách an toàn.
- Cần giảm thời gian giữa thời điểm thu hồi khoá và thời điểm phân phối khoá mới tới mức tối thiểu.
- Phải đảm bảo yêu cầu về an ninh và yêu cầu về tính sẵn sàng của hệ thống.

5. Quản lý khoá

3. Thông báo thông tin về thu hồi khoá

- Thông báo về một khóa nào đó bị thu hồi cần đến được tất cả những người đang sử dụng nó trong thời gian ngắn nhất có thể.
- Hai cách:
  - Thông tin được chuyển từ trung tâm tới người dùng.
  - Người dùng lấy thông tin từ trung tâm.
- Cung cấp các chứng thực có thời hạn.

5. Quản lý khoá

4. *Các biện pháp thực hiện khi lộ khoá*

- Hầu hết các trường hợp thu hồi khoá xảy ra khi khoá bí mật đã bị lộ. Hai khả năng xảy ra:
  - Các văn bản mã hóa với khóa công khai sau thời điểm T không còn được xem là bí mật.
  - các chữ ký số thực hiện với khóa bí mật sau thời điểm T không còn được xem là thật.
- Cần xác định người có quyền thu hồi khóa, cách thức truyền thông tin tới người dùng, cách thức xử lý các văn bản mã hóa với khóa bị lộ.

6. Bài tập

1. Viết chương trình nhập vào một số nguyên dương n, xuất ra:
   - n có phải là số nguyên tố hay không?
   - Dãy số nguyên tố nhỏ hơn hoặc bằng n.
   - n số nguyên tố đầu tiên.
2. Cho *p* là một số nguyên tố và *n* < *p* là một số nguyên dương. Chứng minh rằng a^2 mod *p* = 1 nếu và chỉ nếu a mod *p* = 1 hoặc a mod *p* = -1.

6. Bài tập

3. Hacker có thể lợi dụng điểm yếu trong giao thức trao đổi khoá Diffie-Hellman để thực hiện một cuộc tấn công Man-in-the-Middle.

- Mô tả cuộc tấn công này.
- Vẽ hình minh hoạ.

6. Bài tập

4. Nếu cho số nguyên tố p = 353 thì a = 3 là một primitive root modulo p. Sử dụng hai số này để xây dựng một hệ thống trao đổi khoá Diffiel-Hellman.
   a. Nếu Alice chọn một private key X_A = 97, giá trị public key Y_A của Alice là?
   b. Nếu Bob chọn một private key X_B = 233, giá trị public key Y_B của Bob là?
   c. Giá trị của khoá bí mật thống nhất giữa cả Alice và Bob là bao nhiêu?

6. Bài tập

5. Cho p = 13.
   a. Chứng minh rằng a = 2 là một primitive root modulo p. Sử dụng hai tham số này để xây dựng một hệ thống trao đổei khoá Diffie-Hellman.
   b. Nếu public key của Alice là Y_A = 7, giá trị private key X_A của cô ấy là bao nhiêu?
   c. Nếu public key của Bob là Y_B = 11, giá trị private key X_B của anh ấy?

6. Bài tập

6. Cho $n = 187 = 11 \times 17$.
   a. Cho $e = 7$, $M = 89$. Tính giá trị RSA ciphertext C.
   b. Từ C tính được ở (a), tính toán plaintext *M.*
   c. Cho $e = 7$, $M = 88$. Tính toán giá trị RSA ciphertext C. C có thể sử dụng $n = 187$? Giải thích.

6. Bài tập

7. Alice sử dụng phương pháp dưới đây để mã hoá văn bản rõ (plaintext messages) tiếng Anh với toàn các ký tự viết hoa:

- Ánh xạ mỗi ký tự viết hoa đến các số từ 100 đến 125; cụ thể là, ánh xạ A thành 100, B thành 101, ..., và Z thành 125.
- Sau đó cô ấy mã hoá các số nguyên này sử dụng các giá trị lớn của n và e.
- Phương pháp này có an toàn? Giải thích.

6. Bài tập

8. Giả sử rằng Alice mã hoá một thông điệp M sử dụng RSA với public key n = 437, e = 3, ciphertext C = 75.

- Nếu ai đó nói với Malice rằng $M \in \{8,9\}$, thì Malice có thể xác định giá trị đúng của M mà không cần n.
- Malice thực hiện điều này như thế nào?

6. Bài tập

9. Viết một ứng dụng client-server sử dụng socket API để thực hiện giao thức trao đổi khoá Diffie-Hellman.
10. Viết một ứng dụng client-server sử dụng để thực hiện mã hoá và giải mã RSA, với các tham số của RSA được cho trước.


THANK YOU!