CYB 241 Digital Cryptography Techniques

Public-Key Cryptography and RSA

Birth of Public-Key Cryptosystems

- Beginning to 1960’s: permutations and substitutions (Caesar, rotor machines, DES, . . .)
- 1960’s: NSA secretly discovered public-key cryptography
- 1970: first known (secret) report on public-key cryptography by CESG, UK
- 1976: Diffie and Hellman public introduction to public-key cryptography
  - Avoid reliance on third-parties for key distribution
  - Allow digital signatures

Principles of Public-Key Cryptosystems

- Symmetric algorithms used same secret key for encryption and decryption
- Asymmetric algorithms in public-key cryptography use one key for encryption and different but related key for decryption

Misconceptions Concerning Public-Key Encryption

- Is public-key encryption more secure than symmetric encryption?
  - The security of any encryption scheme depends on the length of the key
- Does public-key encryption make symmetric encryption obsolete?
  - Symmetric encryption takes less time than public key encryption.
- Is the key distribution in public-key trivial compared to symmetric encryption?
  - Public-key encryption needs some form of key distribution protocol involving a central agent and are not simpler nor any more efficient than symmetric encryption

Terminology Related to Asymmetric Encryption

- **Asymmetric Keys**
  - Two related keys, a public key and a private key, that are used to perform encryption and digital signature.
- **Public Key Certificate**
  - A document issued and signed by the private key of a Certification Authority that binds the name of a subscriber to a public key.
- **Public Key (Asymmetric) Cryptographic Algorithm**
  - A cryptographic algorithm that uses two related keys, a public key and a private key.
- **Public Key Infrastructure (PKI)**
  - Used for the purpose of administering certificates and public-private key pairs

Symmetric Encryption Problems

- The concept of public-key cryptography evolved from an attempt to attack two of the most difficult problems associated with symmetric encryption:

Key distribution

- **How to have secure communications without trusting a key distribution center (KDC) with your key**

Digital signatures

- **How to verify that a message comes intact from the claimed sender**

Public-Key Encryption Characteristics

- Two keys: private, public where one key for encryption, other for decryption
- Public Key (PU)
  - For secrecy: used in encryption
  - For authentication: used in decryption
  - Available to anyone
- Private Key (PR)
  - For secrecy: used in decryption
  - For authentication: used in encryption
  - Secret, known only by owner
- Computationally infeasible to determine private key using ciphertext and public key
- Some algorithms (RSA): Either of the two related keys can be used for encryption, with the other used for decryption.

Public-Key Cryptosystems

- A public-key encryption scheme has six ingredients:

Plaintext

The readable message or data that is fed into the algorithm as input

Encryption algorithm

Performs various transform-ations on the plaintext

Public key

Used for encryption or decryption

Private key

Used for encryption or decryption

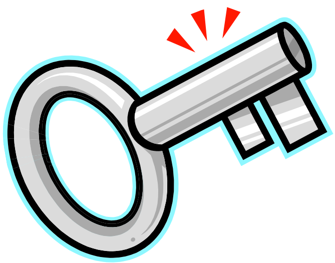
Ciphertext

The scrambled message produced as output

Decryption algorithm

Accepts the ciphertext and the matching key and produces the original plaintext

Public-Key Cryptography



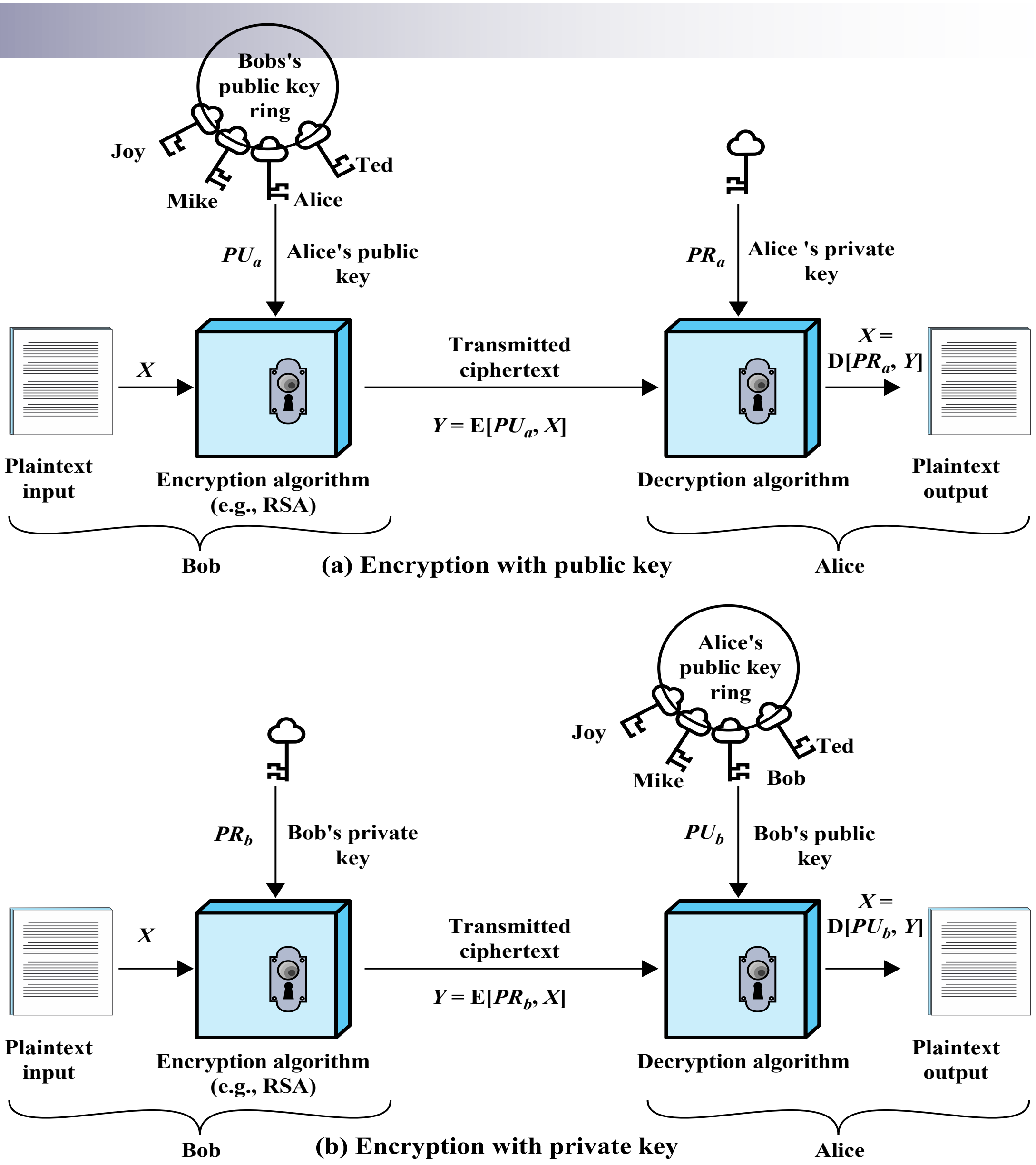


Figure 9.1 Public-Key Cryptography

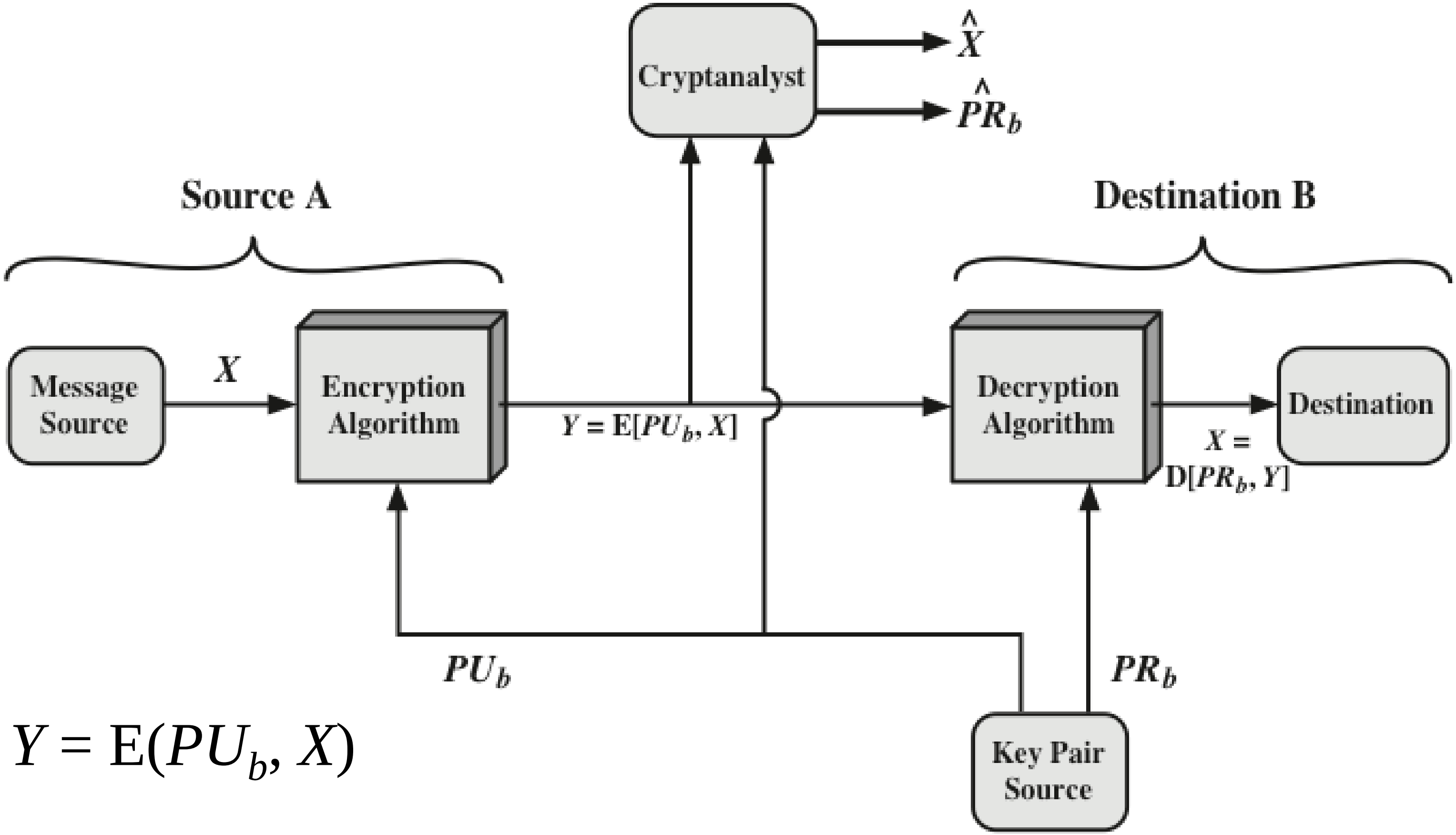
Operation

- Each user generates pair of keys
- Place one of keys in public register or accessible file (public key)
- Keep other companion key (private key)
- If Bob wants to send confidential message to Alice: encrypt with Alice’s public key
- Only Alice can decrypt message with her private key

Advantages

- Private keys generated locally
- Private keys need not to be distributed
- Keys can be changed at any time

Applications: Confidentiality

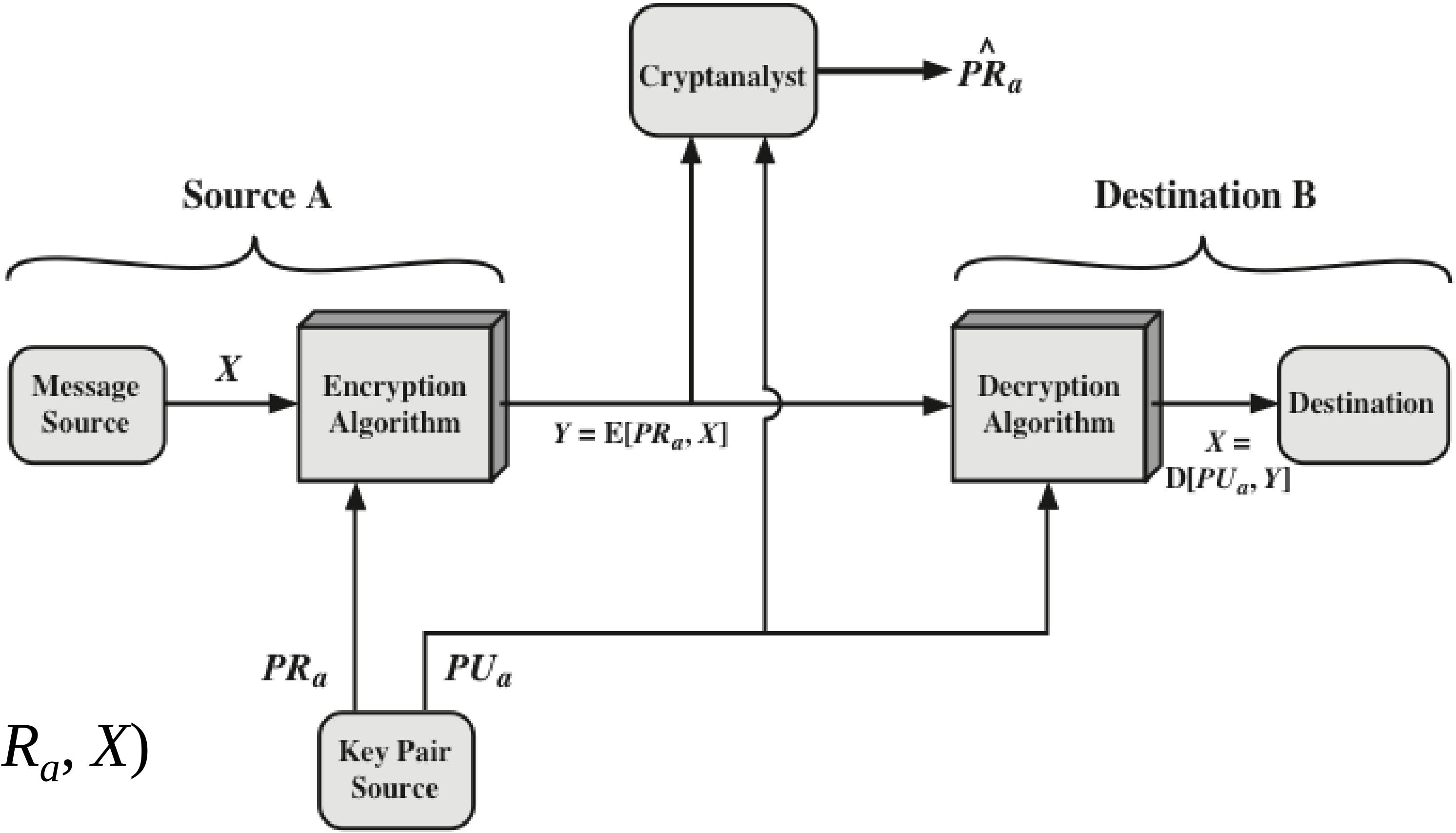


$$Y = E(PU_b, X)$$

$$X = D(PR_b, Y)$$

Figure 9.2 Public-Key Cryptosystem: Secrecy

Applications: Authentication

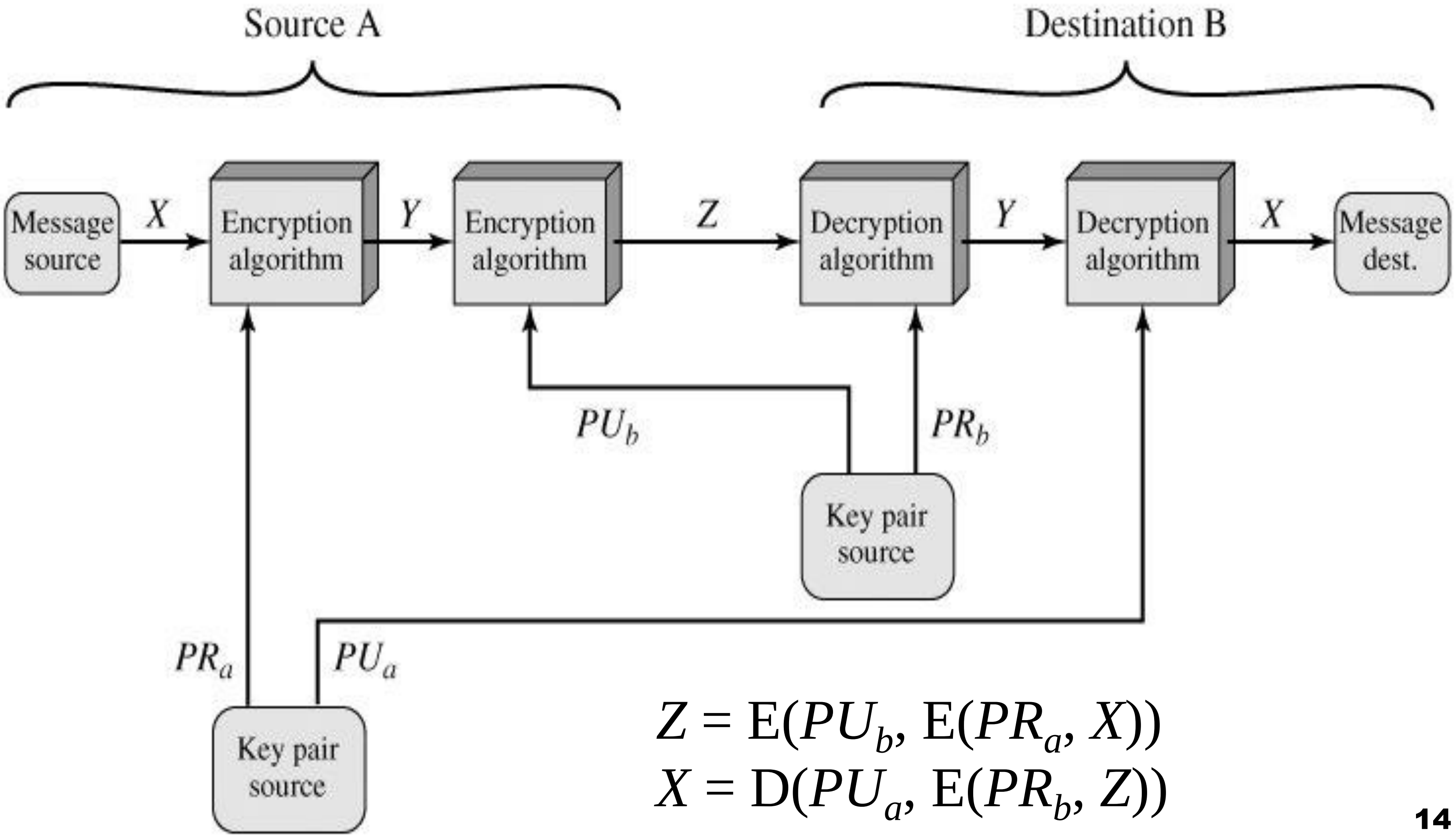


$$Y = \mathrm{E}(PR_a, X)$$

$$X = \mathrm{D}(PU_a, Y)$$

Figure 9.3 Public-Key Cryptosystem: Authentication

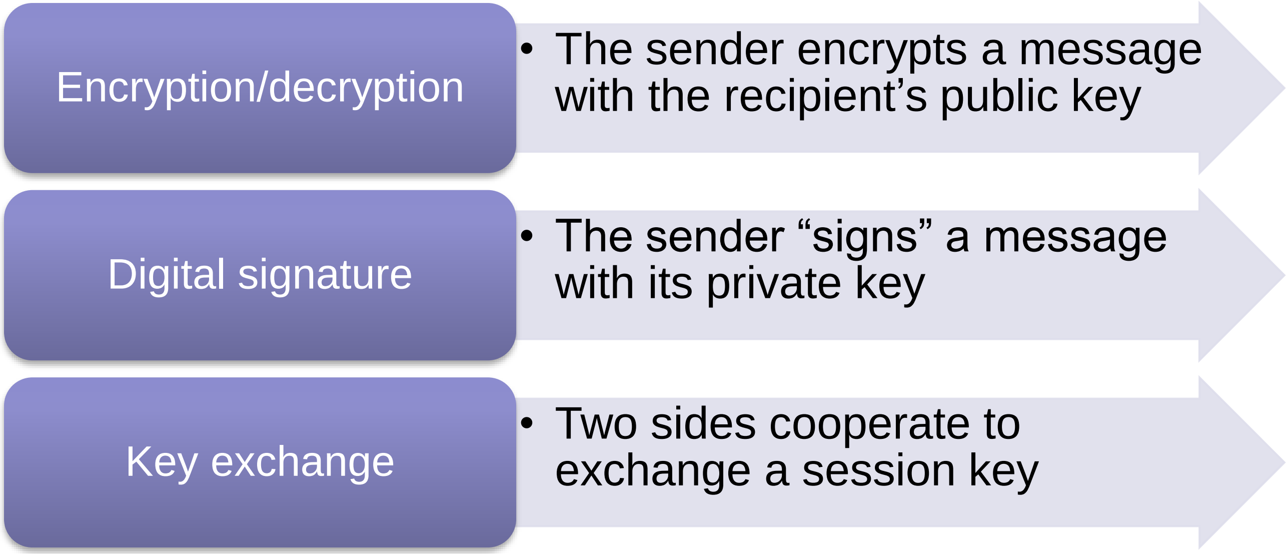
Applications: Confidentiality + Authentication



$$Z = \mathrm{E}(PU_b, \mathrm{E}(PR_a, X))$$
$$X = \mathrm{D}(PU_a, \mathrm{E}(PR_b, Z))$$

Applications for Public-Key Cryptosystems

- Public-key cryptosystems can be classified into three categories:



Encryption/decryption

- The sender encrypts a message with the recipient’s public key

Digital signature

- The sender “signs” a message with its private key

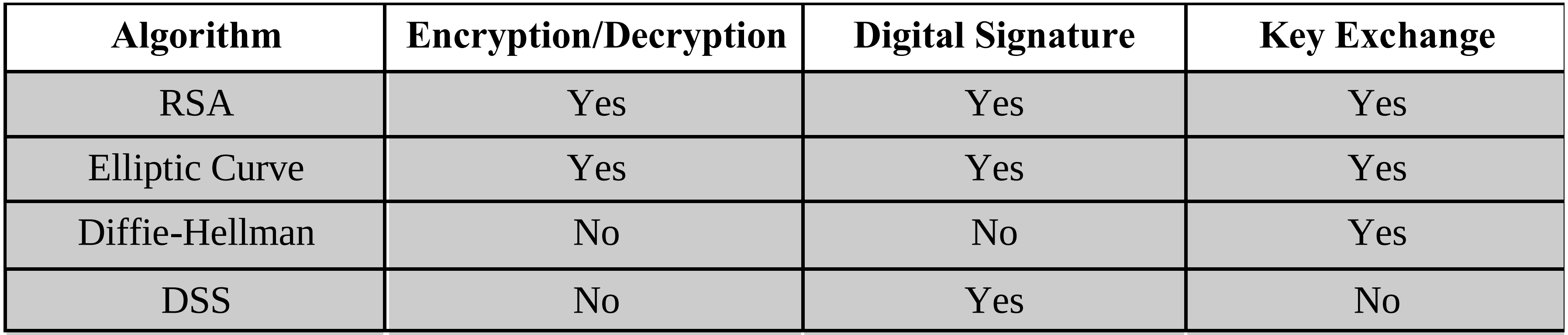
Key exchange

- Two sides cooperate to exchange a session key

- Some algorithms are suitable for all three applications, whereas others can be used only for one or two

Table 9.3

Applications for Public-Key Cryptosystems



Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

Table 9.3 Applications for Public-Key Cryptosystems

Rivest-Shamir-Adleman (RSA) Scheme

- Developed in 1977 at MIT by Ron Rivest, Adi Shamir & Len Adleman
- Most widely used general-purpose approach to public-key encryption
- Is a cipher in which the plaintext and ciphertext are integers between 0 and $n - 1$ for some n
  - □ A typical size for n is 1024 bits, or 309 decimal digits

RSA Algorithm

- Plaintext M is encrypted in blocks
- M < n
- Encryption (plaintext block M, ciphertext C)
  - $C = M^e \bmod n$
- Decryption
  - $M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$

Assumptions

- Sender and receiver know n
- Sender knows e
- Only the receiver knows d
- Public key PU = $\{e, n\}$
- Private key PR = $\{d, n\}$

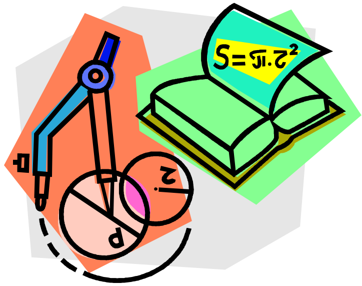
Algorithm Requirements

- For this algorithm to be satisfactory for public-key encryption, the following requirements must be met:

1. It is possible to find values of *e, d, n* such that M^{ed} mod n = M for all $M < n$

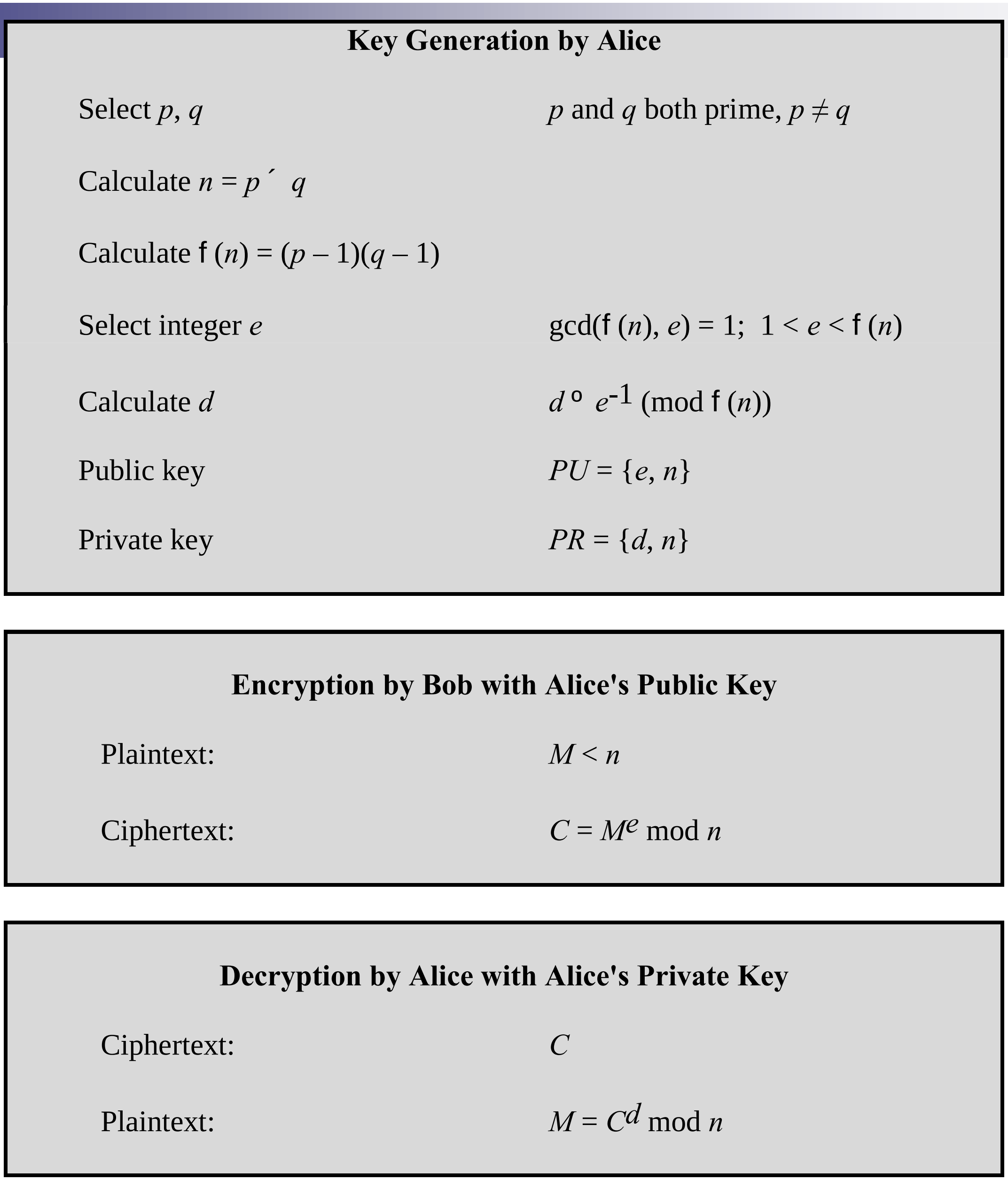
2. It is relatively easy to calculate (M^e mod *n)* and (C^d mod *n)* for all values of $M < n$

3. It is infeasible to determine *d* given *e* and *n*



Ingredients

p, q, two large prime numbers	private	chosen
$n = pq$	public	calculated
e, with $\gcd(\phi(n), e) = 1$; $1 < e < \phi(n)$; $\phi(n)=(p-1)(q-1)$	public	chosen
$d \equiv e^{-1} \pmod{\phi(n)}$ Or $(d \times e) \bmod \phi(n)=1$	private	calculated



Key Generation by Alice

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p$ ´ q	
Calculate f (n) = (p – 1)(q – 1)	
Select integer e	gcd(f (n), e) = 1; 1 < e < f (n)
Calculate d	d º e^{-1} (mod f (n))
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption by Bob with Alice's Public Key

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

Decryption by Alice with Alice's Private Key

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

Figure 9.5 The RSA Algorithm

Example 1

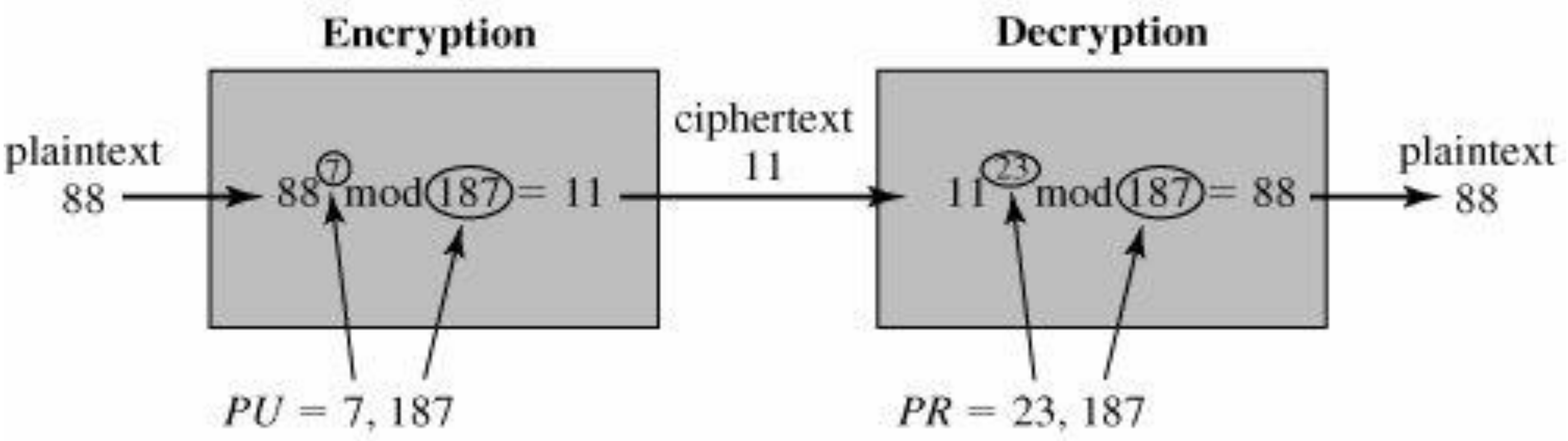
- Choose p=3 and q=11
- Compute n=p*q= 3*11=33
- Compute ϕ(*n*)=(p-1)*(q-1)= 2*10=20
- Choose e such that 1<e<ϕ(*n*) and relatively prime to ϕ(*n*). Let e= 7
- Compute d such that (d*e)mod ϕ(*n*) =1. One solution is d=3 since [3*7 mod 20] =1
- PU={e,n} = {7,33}
- PR={d,n} = {3,33}
- The encryption of m=2 is c=2^7 mod 33= 29
- The decryption of c=29 is m=29^3 mod 33 = 2

Example 2

- Select $p = 17$, $q = 11$
- Calculate $n = pq = 17 \times 11 = 187$
- Calculate $\phi(n) = (p - 1)(q - 1) = 160$
- Select $e < 160$, relatively prime to 160: $e = 7$
- Calculate d < 160, d*e mod 160 = 1
  - □ repeat $d = (k * 160 + 1) / e$
  - □ increment k until you get an integer value
  - □ 1 × 160 + 1 = 161; d = 161/7 = 23

Example 2

- PU = {7, 187}, PR = {23, 187}
- Let M = 88



Exponentiation in Modular Arithmetic

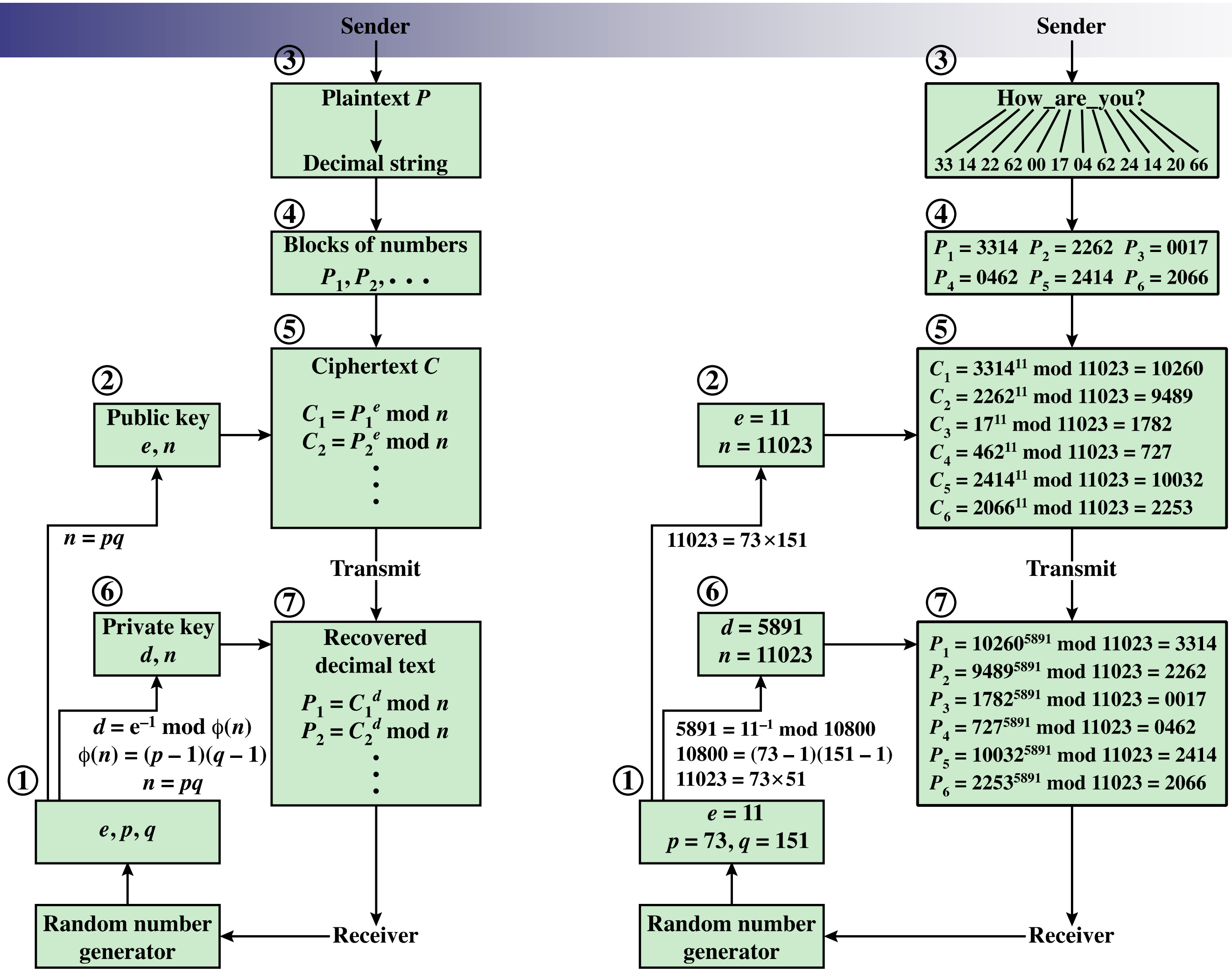
- If exponentiation is done first, intermediate results will be huge
- we can use property of modular arithmetic
  $[(a \text{ mod } n) \times (b \text{ mod } n)] \text{ mod } n = (a \times b) \text{ mod } n$
- To calculate x^{11} mod n
  - □ $x^{11} = x^{1+2+8} = (x)(x^2)(x^8)$
  - □ compute x mod n, x^2 mod n, x^4 mod n, x^8 mod n
  - □ calculate $[(x \text{ mod } n) \times (x^2 \text{ mod } n) \times (x^8 \text{ mod } n)] \text{ mod } n$

Example 2

- $88^7 \bmod 187 = [(88^4 \bmod 187) \times (88^2 \bmod 187) \times (88^1 \bmod 187)] \bmod 187$
- $88^1 \bmod 187 = 88$
- $88^2 \bmod 187 = 7744 \bmod 187 = 77$
- $88^4 \bmod 187 = (88^2)^2 \bmod 187 = 77^2 \bmod 187$
  $= 132$
- $88^7 \bmod 187 = (88 \times 77 \times 132) \bmod 187 = \mathbf{11}$

Example 2

- 11^{23} mod 187 = [(11^{1} mod 187) × (11^{2} mod 187) × (11^{4} mod 187) × (11^{8} mod 187) × (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 = 3025 mod 187 = 33
- 11^{16} mod 187 = 1089 mod 187 = 154
- 11^{23} mod 187 = (11 × 121 × 55 × 154) mod 187 = **88**



Sender

③ Plaintext P → Decimal string

④ Blocks of numbers $P_1, P_2, \ldots$

⑤ Ciphertext C

$C_1 = P_1^e \bmod n$

$C_2 = P_2^e \bmod n$

⋮

② Public key e, n

$n = pq$

Transmit

⑥ Private key d, n

⑦ Recovered decimal text

$P_1 = C_1^d \bmod n$

$P_2 = C_2^d \bmod n$

⋮

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

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

$n = pq$

① e, p, q

Random number generator

Receiver

(a) General approach

Sender

③ How_are_you?

33 14 22 62 00 17 04 62 24 14 20 66

④ $P_1 = 3314$ $P_2 = 2262$ $P_3 = 0017$

$P_4 = 0462$ $P_5 = 2414$ $P_6 = 2066$

⑤ $C_1 = 3314^{11} \bmod 11023 = 10260$

$C_2 = 2262^{11} \bmod 11023 = 9489$

$C_3 = 17^{11} \bmod 11023 = 1782$

$C_4 = 462^{11} \bmod 11023 = 727$

$C_5 = 2414^{11} \bmod 11023 = 10032$

$C_6 = 2066^{11} \bmod 11023 = 2253$

② $e = 11$

$n = 11023$

11023 = 73×151

Transmit

⑥ $d = 5891$

$n = 11023$

⑦ $P_1 = 10260^{5891} \bmod 11023 = 3314$

$P_2 = 9489^{5891} \bmod 11023 = 2262$

$P_3 = 1782^{5891} \bmod 11023 = 0017$

$P_4 = 727^{5891} \bmod 11023 = 0462$

$P_5 = 10032^{5891} \bmod 11023 = 2414$

$P_6 = 2253^{5891} \bmod 11023 = 2066$

$5891 = 11^{-1} \bmod 10800$

$10800 = (73 - 1)(151 - 1)$

11023 = 73×51

① $e = 11$

$p = 73, q = 151$

Random number generator

Receiver

(b) Example

Figure 9.7 RSA Processing of Multiple Blocks

CYB 241 Digital Cryptography Techniques

Other Public-Key Cryptosystems

Diffie-Hellman Key Exchange

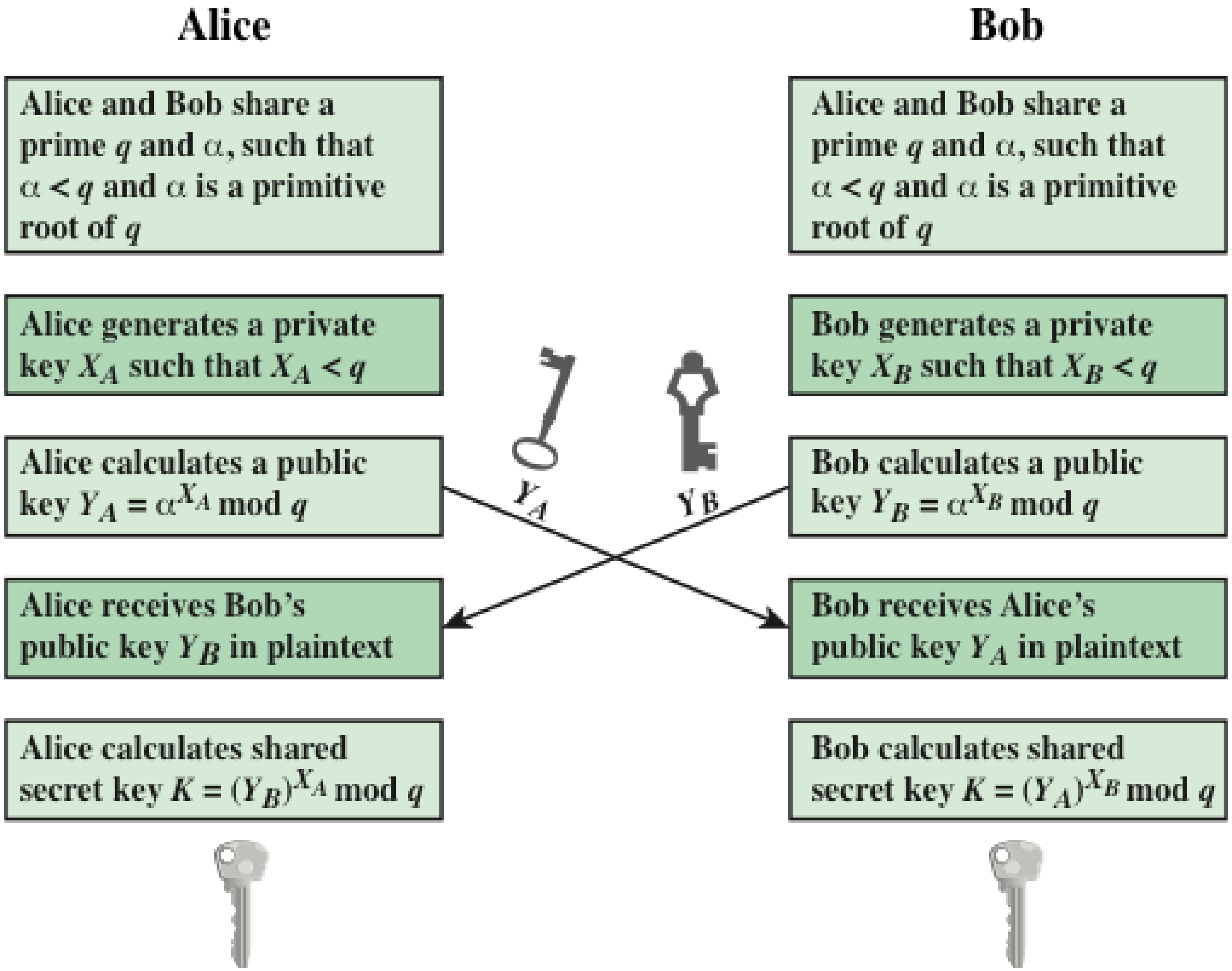
- First published public-key algorithm
- A number of commercial products employ this key exchange technique
- Purpose is limited to enable two users to securely exchange a key that can then be used for subsequent symmetric encryption of messages
- Two parties A and B using this algorithm for creating a shared secret key first agree on a large prime number *p* and primitive root of *p*
- Its effectiveness depends on the difficulty of computing discrete logarithms

Primitive root

- A primitive root of a prime number *p* is one whose powers modulo *p* generate all the integers from 1 to *p* - 1.
- If *a* is a primitive root of the prime number *p*, then
  - the numbers a mod p, a^2 mod p, ..., a^{p-1} mod p are distinct and consist of the integers from 1 through *p* - 1 in some permutation.
- For any integer *b* and a primitive root *a* of prime number *p*, we can find a unique exponent *i* such that
  - $b = a^i \pmod{p}$ where $1 \le i \le (p - 1)$
  - The exponent *i* is referred to as the **discrete logarithm** of *b* for the base *a*, mod *p*.

Example

- If we choose *p*= 7 then the primitive root *a = 3*
  - 3^1 mod 7 = 3
  - 3^2 mod 7 = 2
  - 3^3 mod 7 = 6
  - 3^4 mod 7 = 4
  - 3^5 mod 7 = 5
  - 3^6 mod 7 = 1



Alice

Alice and Bob share a prime q and α, such that $\alpha < q$ and α is a primitive root of q

Alice generates a private key X_A such that $X_A < q$

Alice calculates a public key $Y_A = \alpha^{X_A} \bmod q$

Alice receives Bob's public key Y_B in plaintext

Alice calculates shared secret key $K = (Y_B)^{X_A} \bmod q$

Bob

Alice and Bob share a prime q and α, such that $\alpha < q$ and α is a primitive root of q

Bob generates a private key X_B such that $X_B < q$

Bob calculates a public key $Y_B = \alpha^{X_B} \bmod q$

Bob receives Alice's public key Y_A in plaintext

Bob calculates shared secret key $K = (Y_A)^{X_B} \bmod q$

Figure 10.1 Diffie-Hellman Key Exchange

Example

- Let q = 353 and a primitive root of 353, in this case α = 3.
- A and B select private keys X_A = 97 and X_B = 233
- A computes $Y_A = 3^{97}$ mod 353 = 40, B computes Y_B = 3^{233} mod 353 = 248.
- A computes $K = (Y_B)^{XA}$ mod 353 = 24897 mod 353 = 160.
- B computes $K = (Y_A)^{XB}$ mod 353 = 40233 mod 353 = 160.

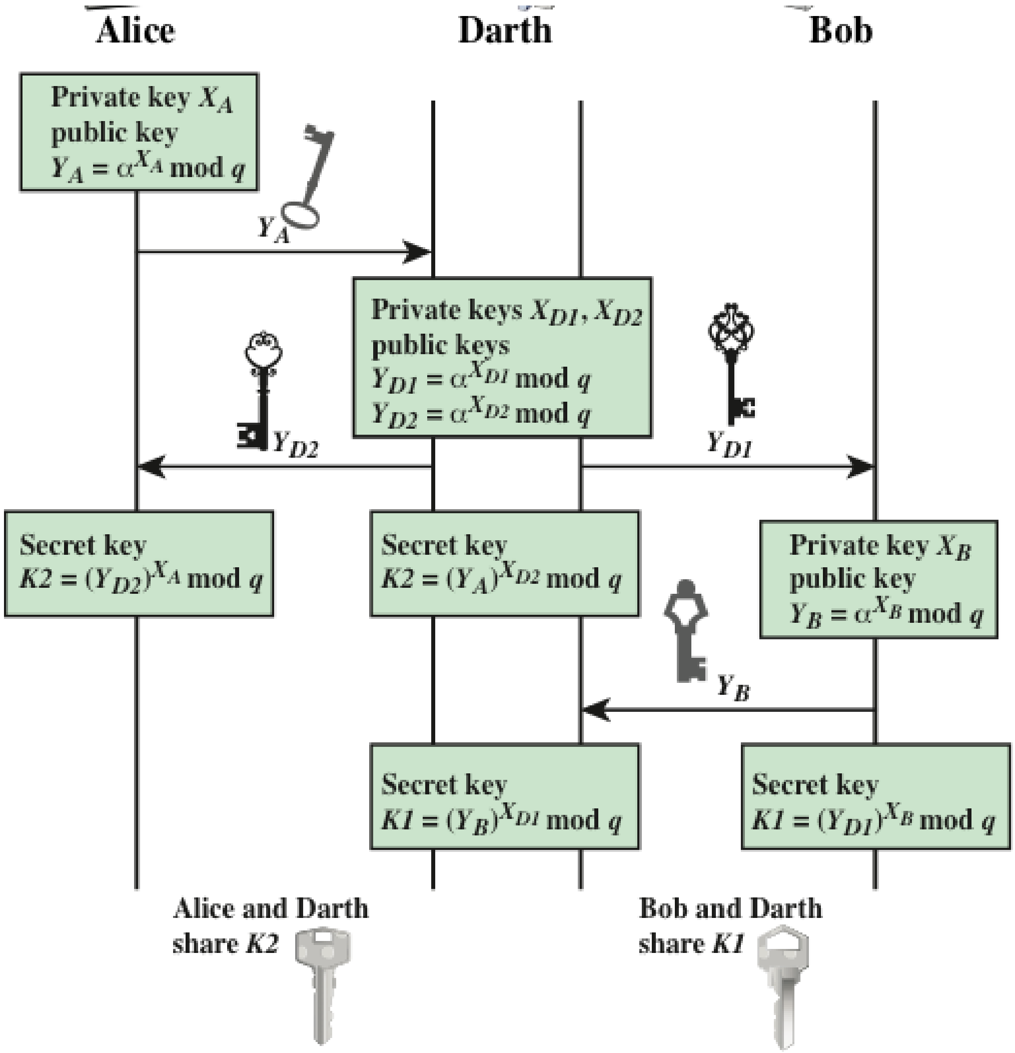

Alice
Darth
Bob
Private key X_A
public key
$Y_A = \alpha^{X_A} \bmod q$
Y_A
Private keys X_{D1}, X_{D2}
public keys
$Y_{D1} = \alpha^{X_{D1}} \bmod q$
$Y_{D2} = \alpha^{X_{D2}} \bmod q$
Y_{D2}
Y_{D1}
Secret key
$K2 = (Y_{D2})^{X_A} \bmod q$
Secret key
$K2 = (Y_A)^{X_{D2}} \bmod q$
Private key X_B
public key
$Y_B = \alpha^{X_B} \bmod q$
Y_B
Secret key
$K1 = (Y_B)^{X_{D1}} \bmod q$
Secret key
$K1 = (Y_{D1})^{X_B} \bmod q$
Alice and Darth
share $K2$
Bob and Darth
share $K1$

Figure 10.2 Man-in-the-Middle Attack

Other Public-Key Cryptosystems

- ElGamal Cryptosystem
  - Similar concepts to Diffie-Hellman
  - Used in Digital Signature Standard and secure email
- Elliptic Curve Cryptography
  - Uses elliptic curve arithmetic (instead of modular arithmetic in RSA)
  - Equivalent security to RSA with smaller keys (better performance)
  - Used for key exchange and digital signatures