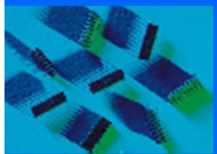


Lecture 10
Introduction to Network Security

Textbook: Ch. 31

Main Topics

A. **Security Goal (31.1)**

B. **Cryptography (31.2)**

- **Symmetric-Key Cryptography (31.2.1)**
  - **Monoalphabetic Substitution**
  - **Polyalphabetic Substitution**
  - **Transpositional Encryption**
- **Asymmetric-key cryptography (31.2.2)**
  - **Requirements for Public Key**
  - **RSA**

C. **Security Aspects (31.3)**

- **Message Integrity (31.3.1)**
- **Message Authentication (31.3.2)**
- **Digital Signature (31.3.3)**

A. Security Goals

- Information needs to be secured from attacks.
- To be secured, information needs to be
  - hidden from unauthorized access (**confidentiality**),
  - protected from unauthorized change (**integrity**),
  - available to an authorized entity when it is needed (**availability**).

Attacks

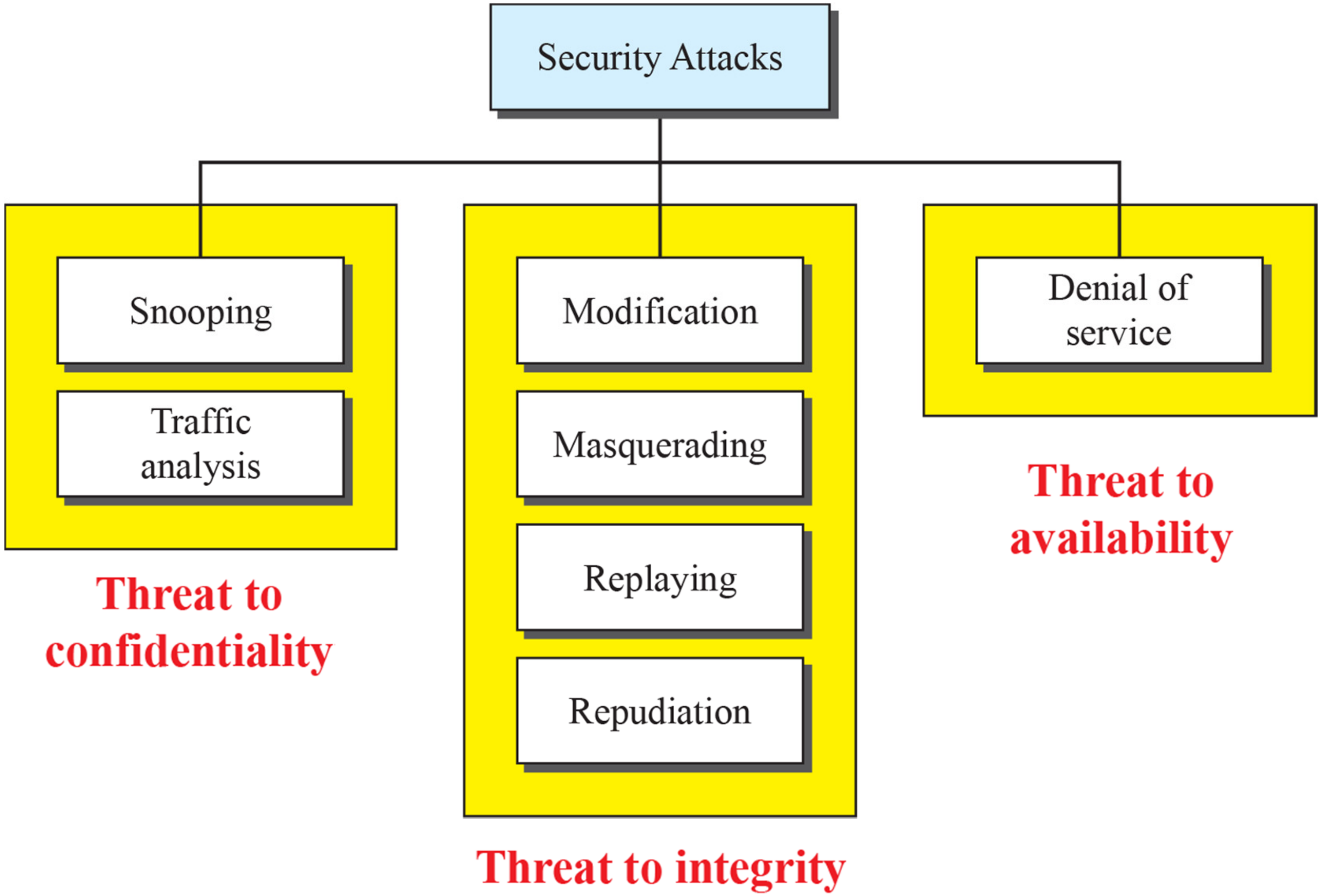
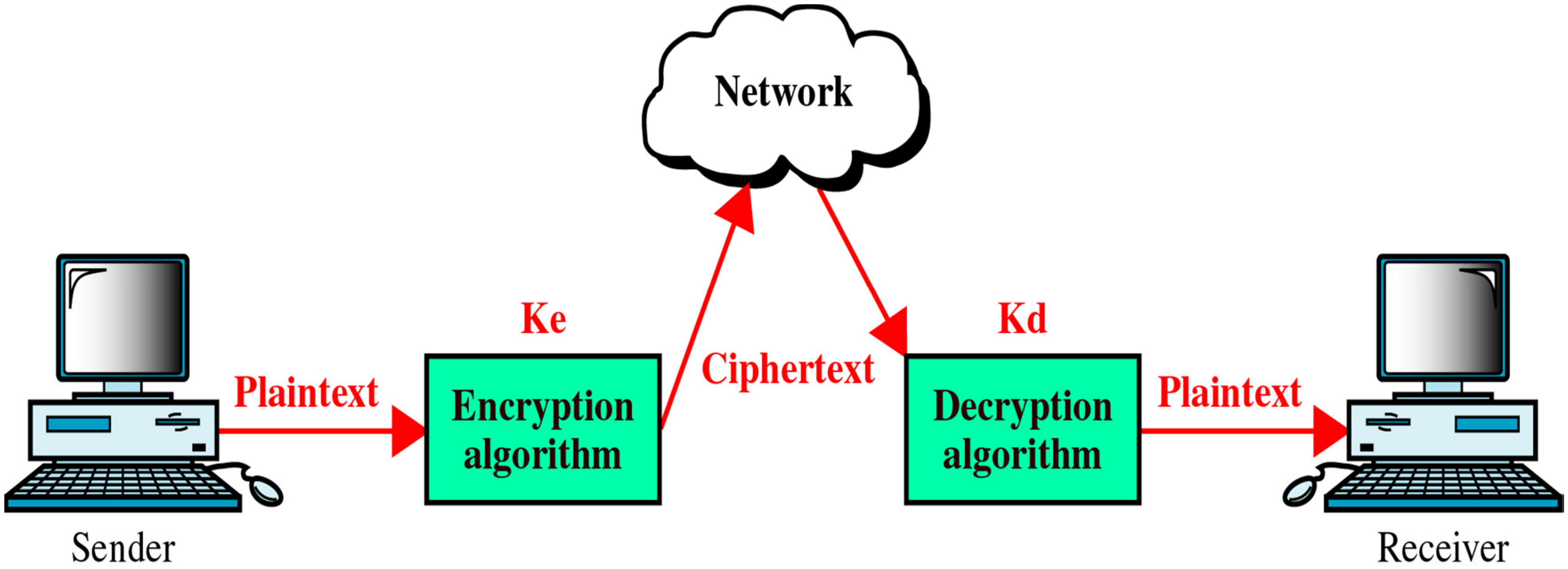


Figure 31.1: Taxonomy of attacks with relation to security goals

B. Cryptography

- Network security is mostly achieved through the use of cryptography.
  - Cryptography is the science of transforming messages to make them secure and immune to attack.
- Aim
  - Confidentiality
  - Integrity
  - Authentication

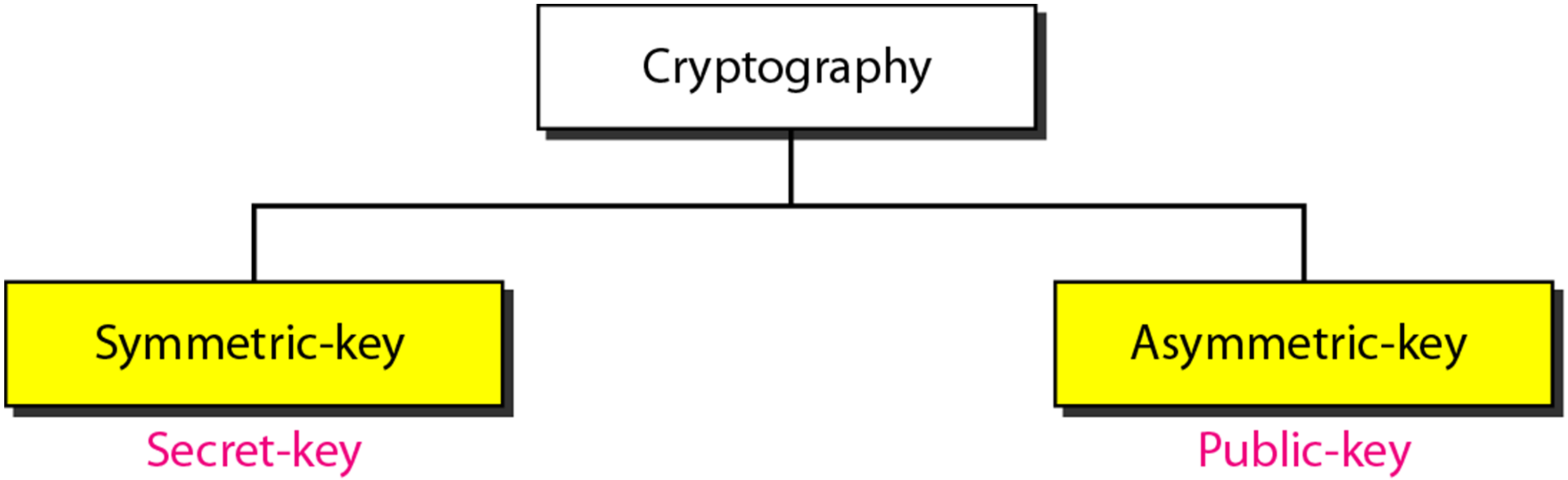
Concept of Encryption and Decryption



Ke is the encryption key

Kd is the decryption key

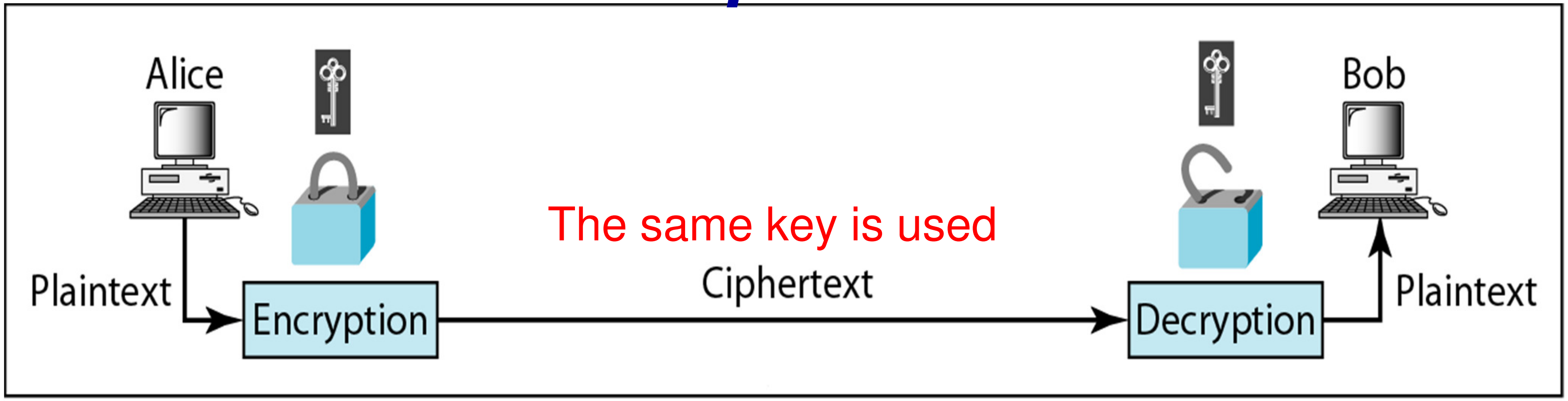
Encryption/Decryption Methods



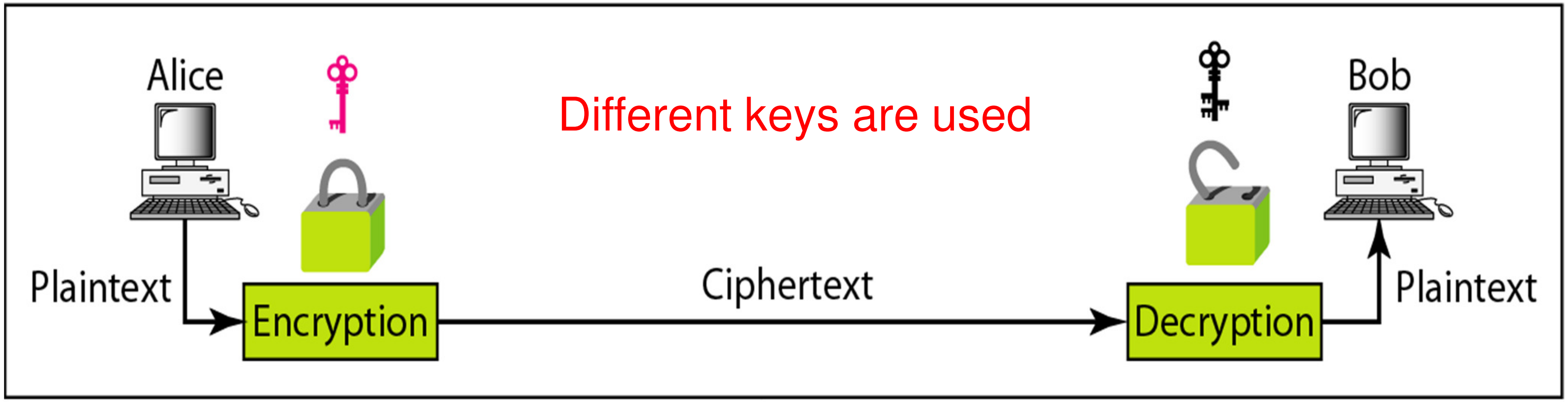
- In **traditional encryption (symmetric)**, the encrypting algorithm is known to everyone but the key is secret except to the sender and receiver

- In **public key encryption (asymmetric)**, both the encrypting algorithm and the encryption key are known to everyone but the decryption key is known only to the receiver

Comparison

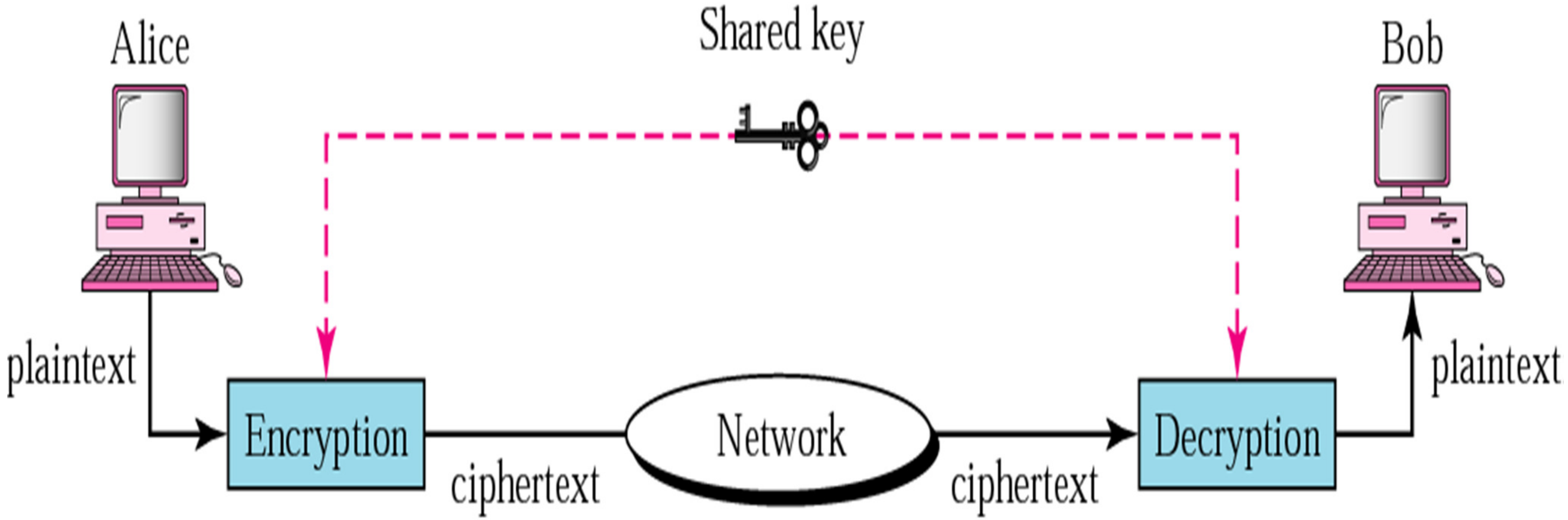


a. Symmetric-key cryptography



b. Asymmetric-key cryptography

I. Symmetric-Key Cryptography

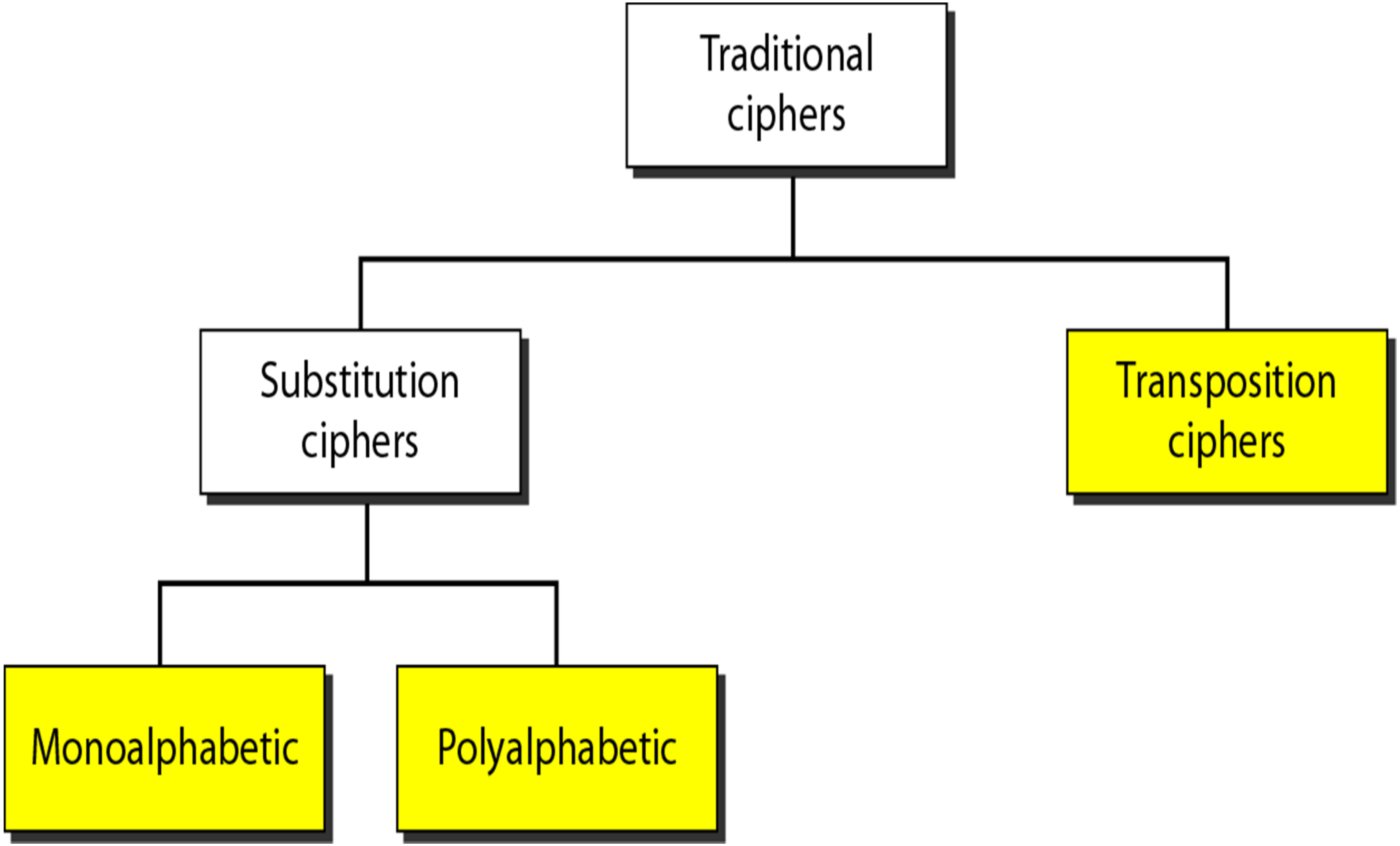


-- The same key (called shared key) is used by the sender (for encryption) and the receiver (for decryption)

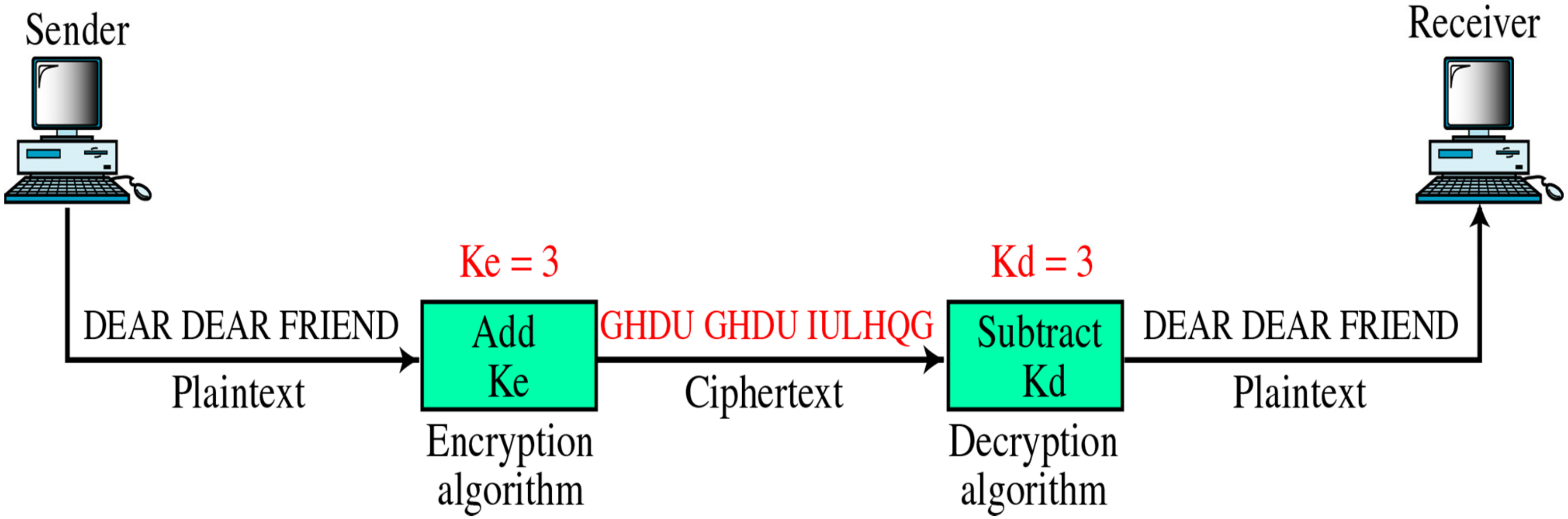
-- e.g. the methods in the following slides

-- Each pair of users must have a unique symmetric key

Traditional Ciphers (Symmetric-Key)

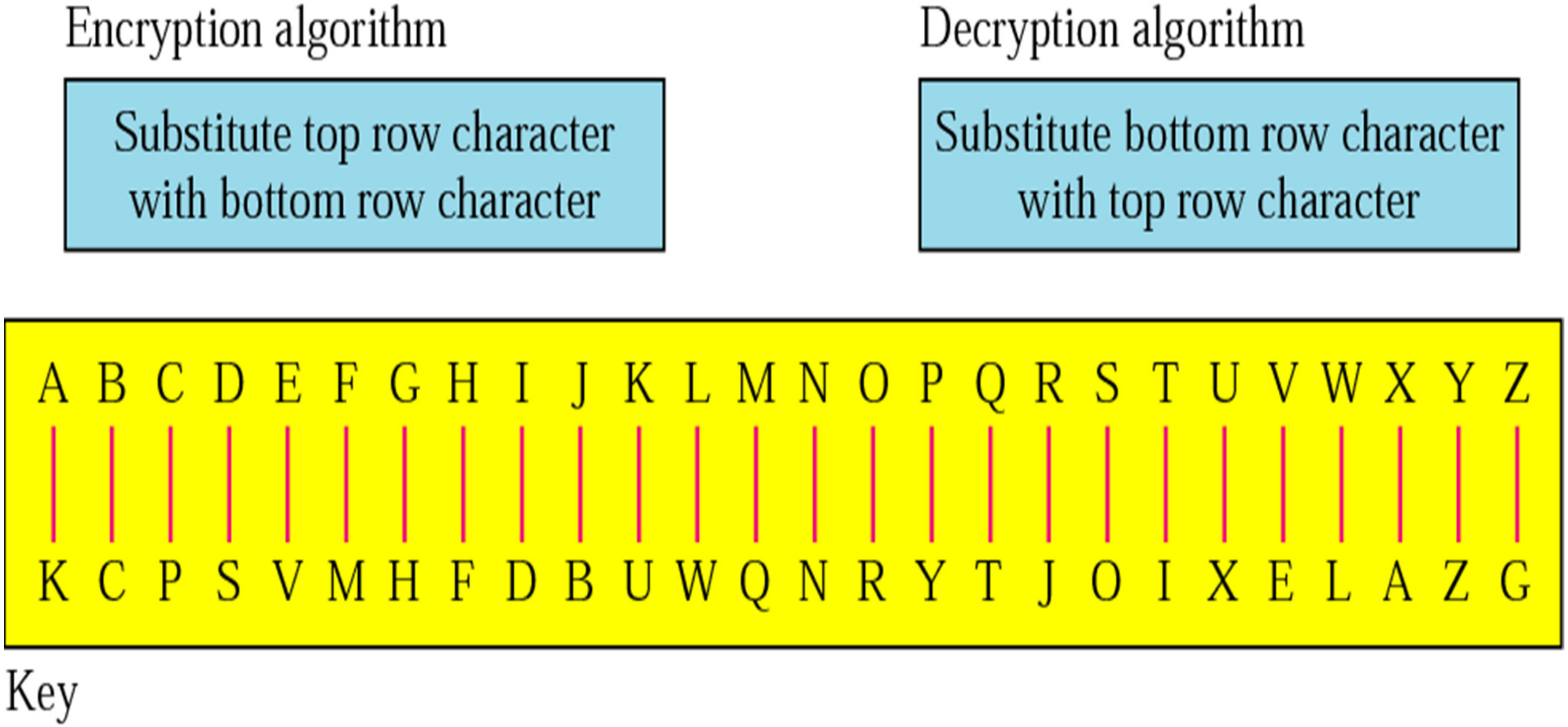


1. Monoalphabetic Substitution



- *Map* every alphabet to another (unique) alphabet. **OR**
- *Shift* the plaintext alphabet by n places (n is the key)
- In monoalphabetic substitution, the relationship between a character in the plaintext to the character in the ciphertext is always one-to-one.

Example of monoalphabetic substitution

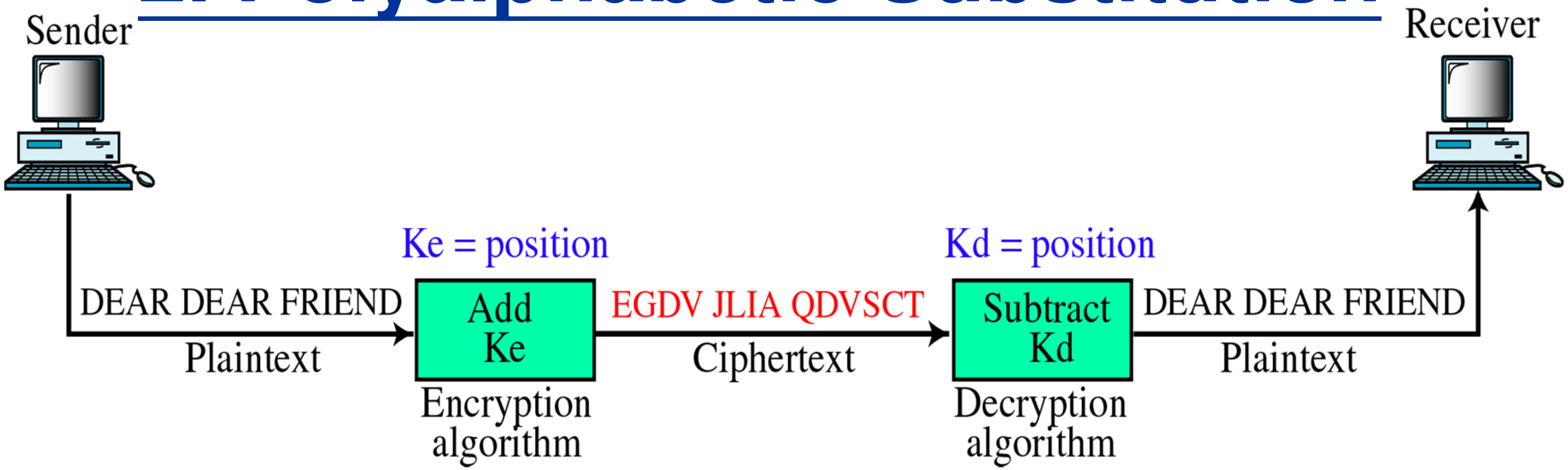


Problem?

-- can be attacked easily

-- cannot hide natural frequencies of characters

2. Polyalphabetic Substitution



- Use different monoalphabetic substitutions as one proceeds through the plaintext message.
- e.g. use the position of the character in the text as the key (of substitution).
- e.g. define a table which maps every plaintext alphabet to a ciphertext alphabet.

Example

Character in plaintext

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	W	R	K	D	O	V	C	A	S	B	Y	Q	M	L	H	I	T	U	F	E	Z	N	G	J	P	X
1	H	Q	B	G	W	E	R	K	F	C	O	A	Z	J	M	S	L	V	N	I	P	U	D	T	X	Y
2	P	I	D	Z	X	V	S	T	O	C	M	J	N	L	B	Q	R	U	W	K	H	G	E	F	A	Y
⋮													⋮													
25	M	C	I	D	A	X	V	S	T	O	N	L	K	U	R	E	W	Z	H	F	P	G	Y	J	B	Q

Character in Ciphertext

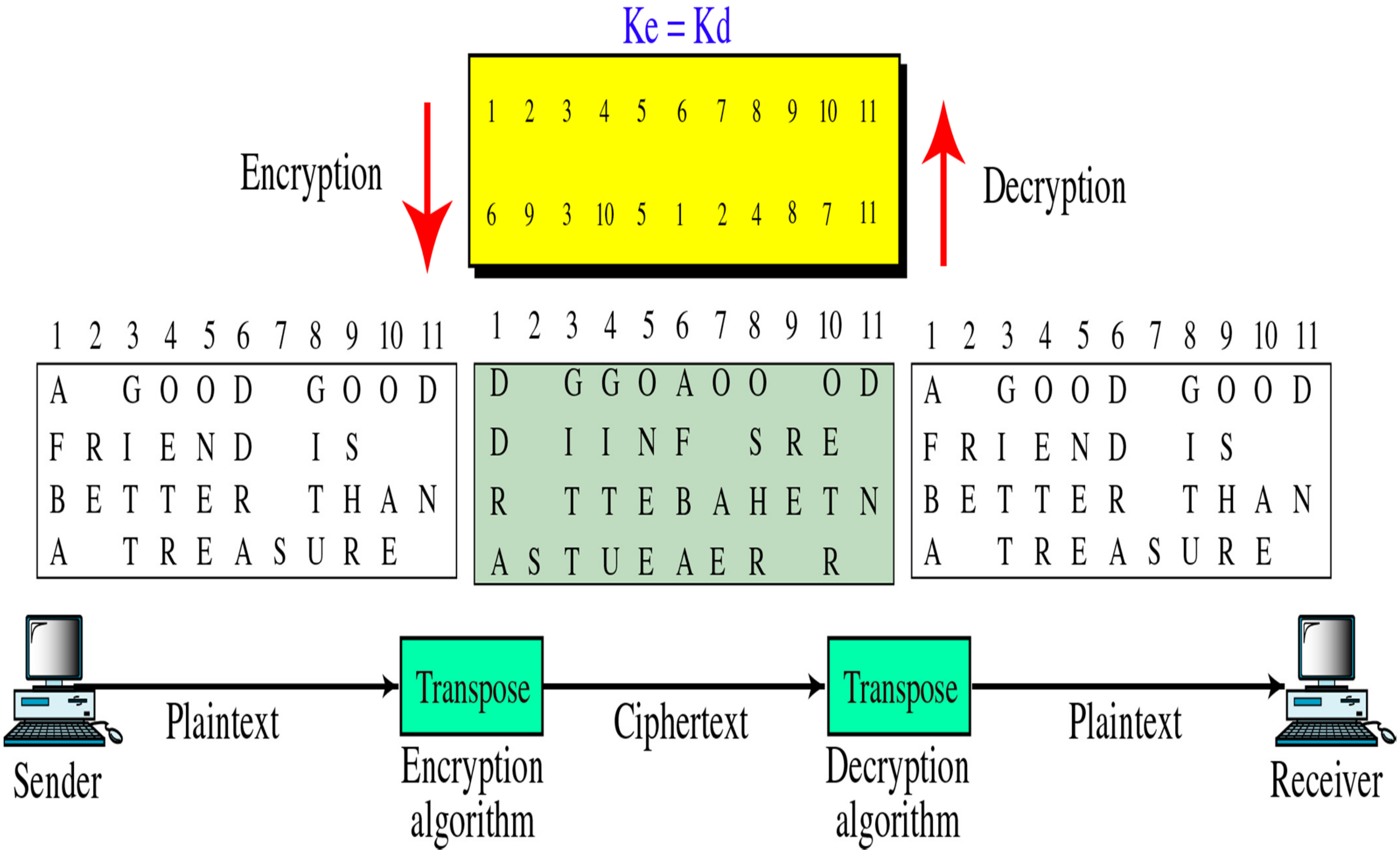
Key = (Position of character in the text) mod 26

- According to this table, A is encrypted as W if it is in position 0 and as M if it is in position 25.

3. Transpositional Encryption

- Re-order the positions of the characters in the plaintext
- e.g. Organize the plaintext into a table of n columns (n is the key length)
  - The columns are interchanged according to the key, which is a series of numbers
  - After exchanging the columns, the “encrypted” data is outputted “row by row”
- e.g. The key in the following slide is
  - 6, 9, 3, 10, 5, 1, 2, 4, 8, 7, 11 (and the key length is 11)
- Means column 1 becomes column 6,
- column 2 becomes column 9 and so on

Transpositional Encryption



II. Asymmetric-key cryptography

- It is also called **Public Key Cryptography**
- Encryption uses the key E called ***public key***, while decryption uses another key D called ***private key***
- i.e. encryption and decryption use different keys (this is an asymmetric method)
- (Here E(P) represents the ciphertext formed by encrypting the plaintext P using the key E)

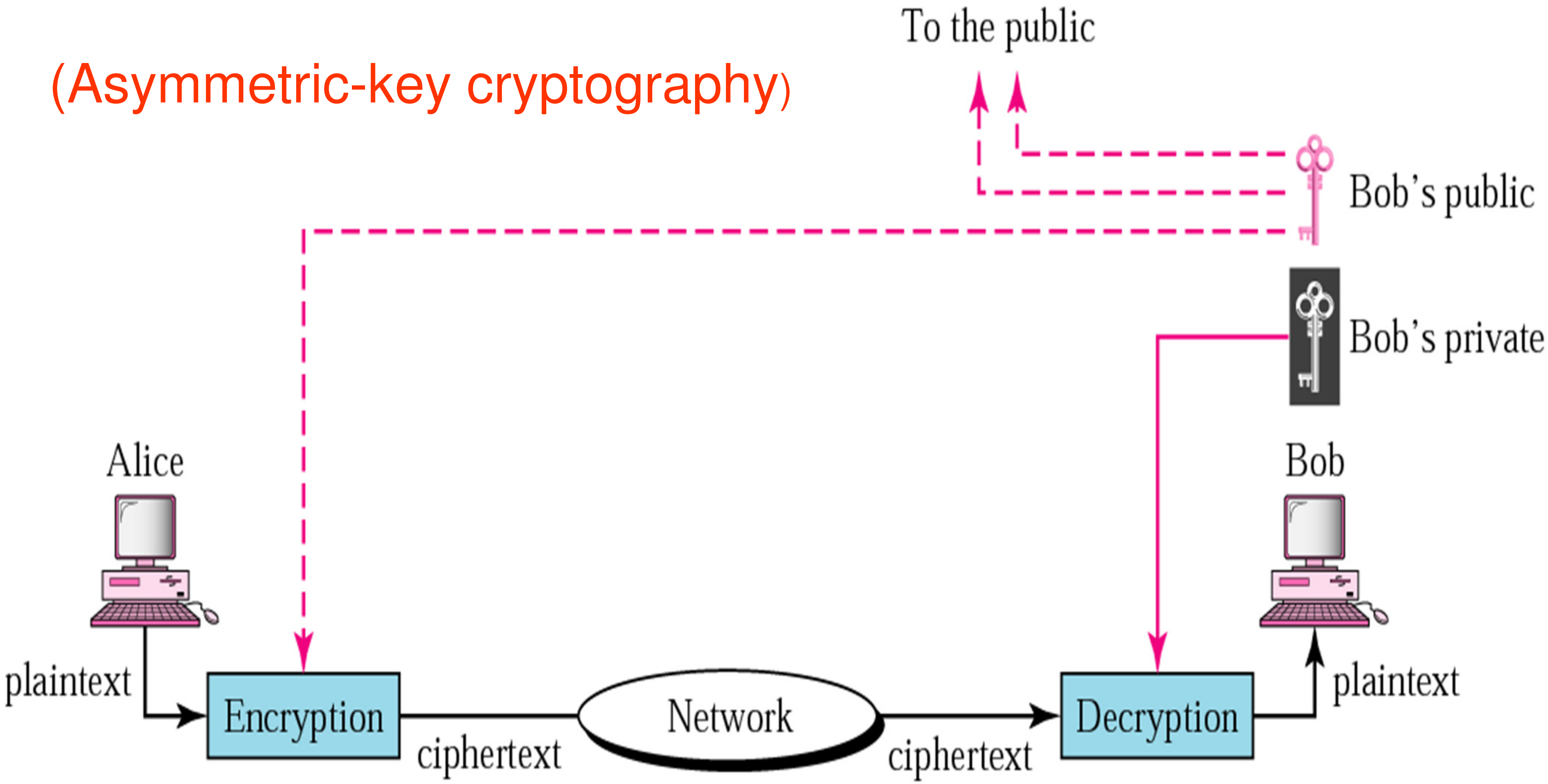
1. Requirements for Public Key

- 1) The encryption key (called public key) is made public, while the decryption key (called private key) is kept by the user securely
- 2) D(E(P)) = P ,i.e. using D to decrypt a ciphertext message which is encrypted by E can get back the original message P
- 3) It is very, very difficult to deduce D from E

- e.g. The RSA method
- Each user creates a pair of keys (E & D), which can be used to communicate with any other users

Public-key cryptography

(Asymmetric-key cryptography)

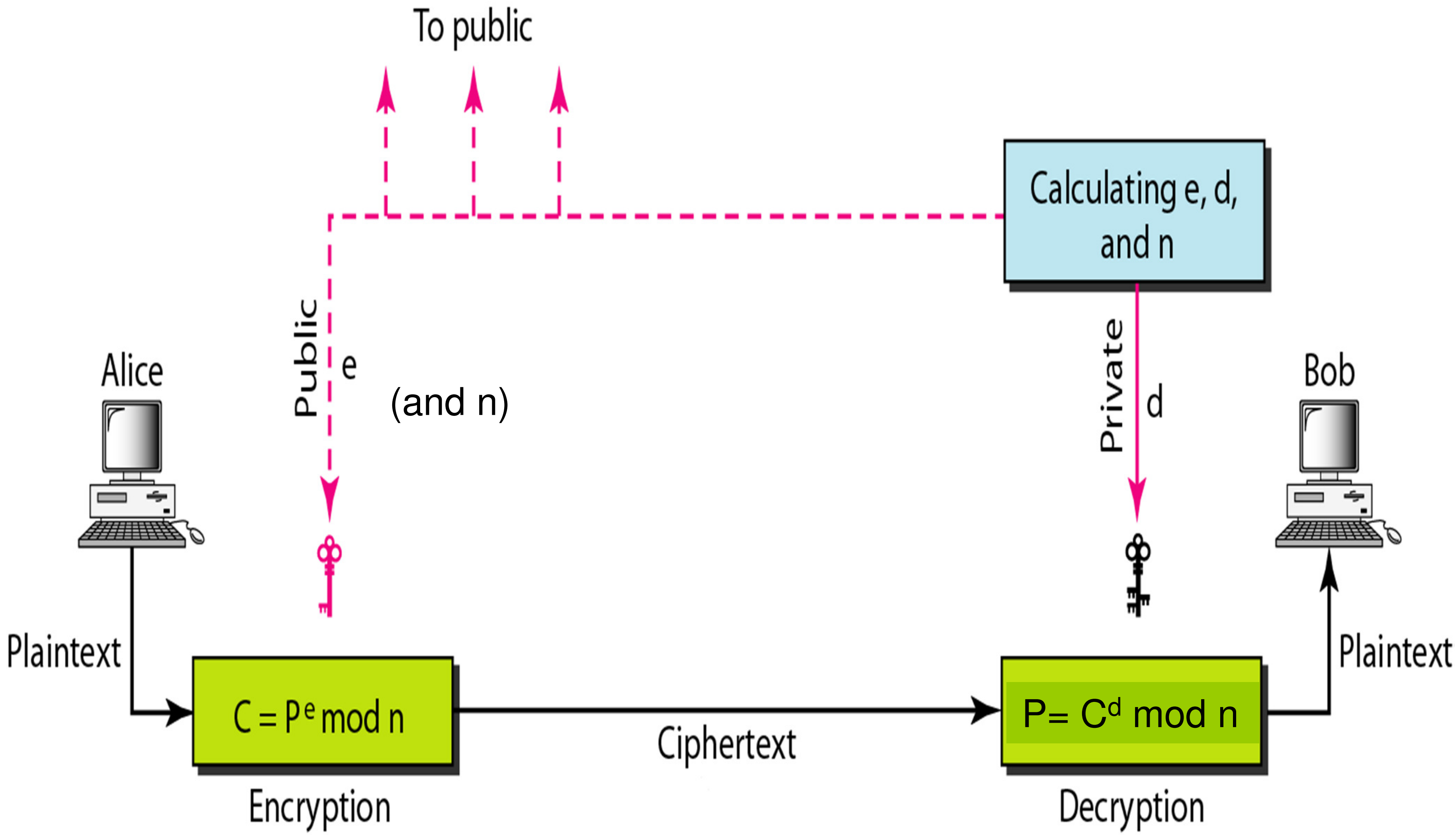


Sender uses the receiver's *public key* to encrypt the message

Receiver uses its own *private key* to decrypt the ciphertext

2. RSA Cryptosystem

RSA is named for its inventors Rivest, Shamir, and Adleman.



Selecting Key for RSA

- Bob uses the following steps to select the private and public keys:

1. Chooses two very large prime numbers p and q.
2. Get n and Φ by $n = p \times q$ and $\Phi = (p\text{-}1) \times (q\text{-}1)$
3. Choose a random integer e and calculate d so that $d \times e \bmod \Phi = 1$.
4. ***e*** **and** ***n*** **are announced to the public;** ***d*** **and Φ are kept secret.**

In RSA, *e* and *n* are announced to the public; *d* and Φ are kept secret.

Encryption

$$C = P^{e} \pmod{n}$$

- ***Example 31.7***

Bob chooses 7 and 11 as p and q
and calculates $= 7 \cdot 11 = 77 = n$.
The value of $\Phi = (7-1)(11-1)$ or 60.
Now he chooses two keys, e and d. If he chooses e to be 13, then d is 37.
Now imagine Alice sends the plaintext 5 to Bob.
She uses the public key 13 to encrypt 5.

37 x 13 mod 60 = 1

Plaintext: 5
$C = 5^{13} \bmod 77 = 26$
Ciphertext: 26

Decryption

$$P = C^{d} \pmod{n}$$

- ***Example 31.7 (continued)***

Bob receives the ciphertext 26 and uses the private key 37 to decipher the ciphertext:

Ciphertext: 26
$P = 26^{37} \bmod 77 = 5$
Plaintext: 5

The plaintext 5 sent by Alice is received as plaintext 5 by Bob.

How many keys are needed?

- N users in a network
  - a) Total number of keys?
  - b) Each user needs to know/store how many keys?
- ***Symmetric-key System***
  - a) N(N-1)/2 b) N-1 Why?
- ***Asymmetric-key System***
  - a) 2N b) N+1 Why?

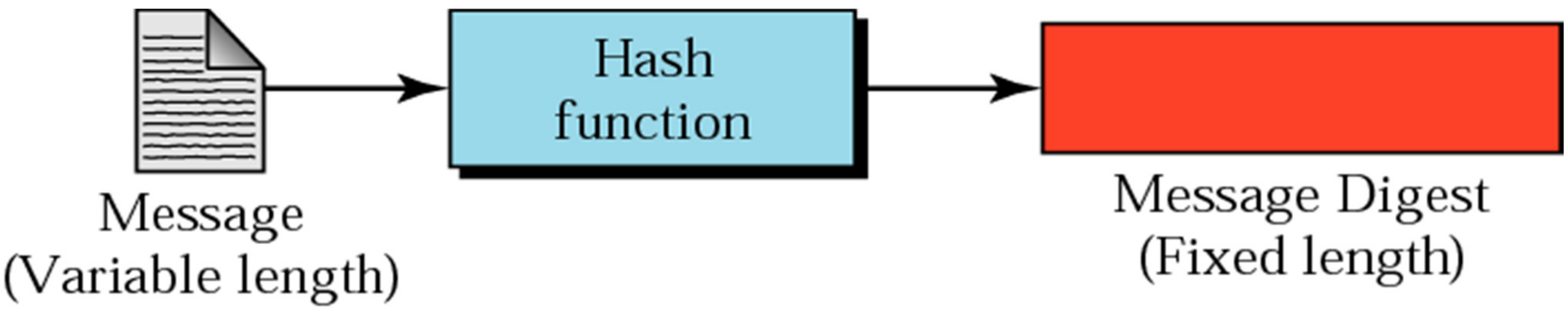
C. Security Aspects

1. Message Integrity

- There are occasions where we may not even need secrecy but instead must have integrity: the message should **remain unchanged**.
- For example, Alice may write a will to distribute her estate upon her death. The will does not need to be encrypted. After her death, anyone can examine the will.
- The integrity of the will, however, needs to be preserved. Alice does not want the contents of the will to be changed.

Message Digest

- A *miniature version (**digest**)* of the message (like a *fingerprint*)
- Created by a one-way hash function: the digest can only be created from the message, not vice versa
- Common hash functions: MD5 and SHA-1



long

short

Message and Digest for checking the Integrity

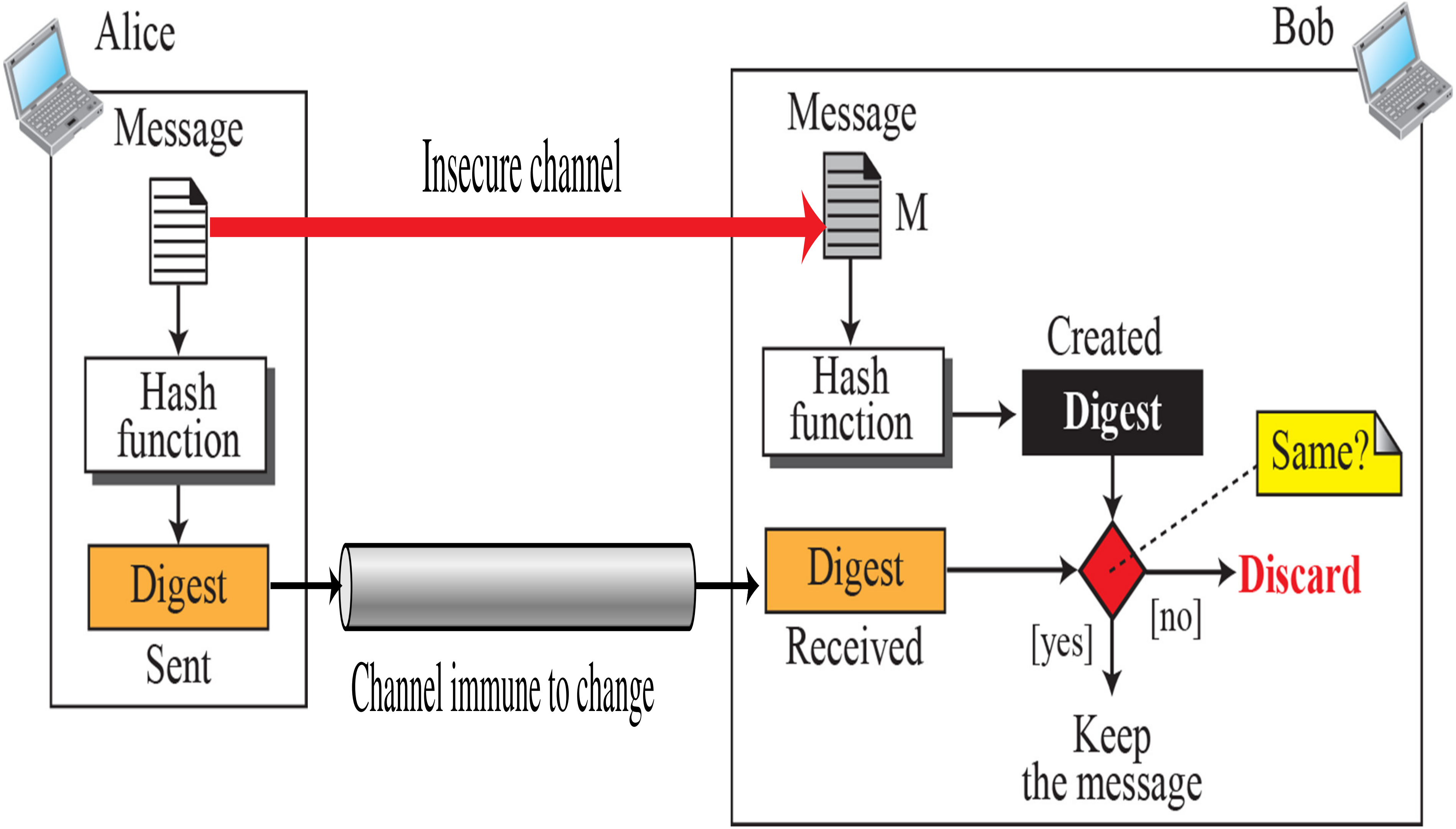
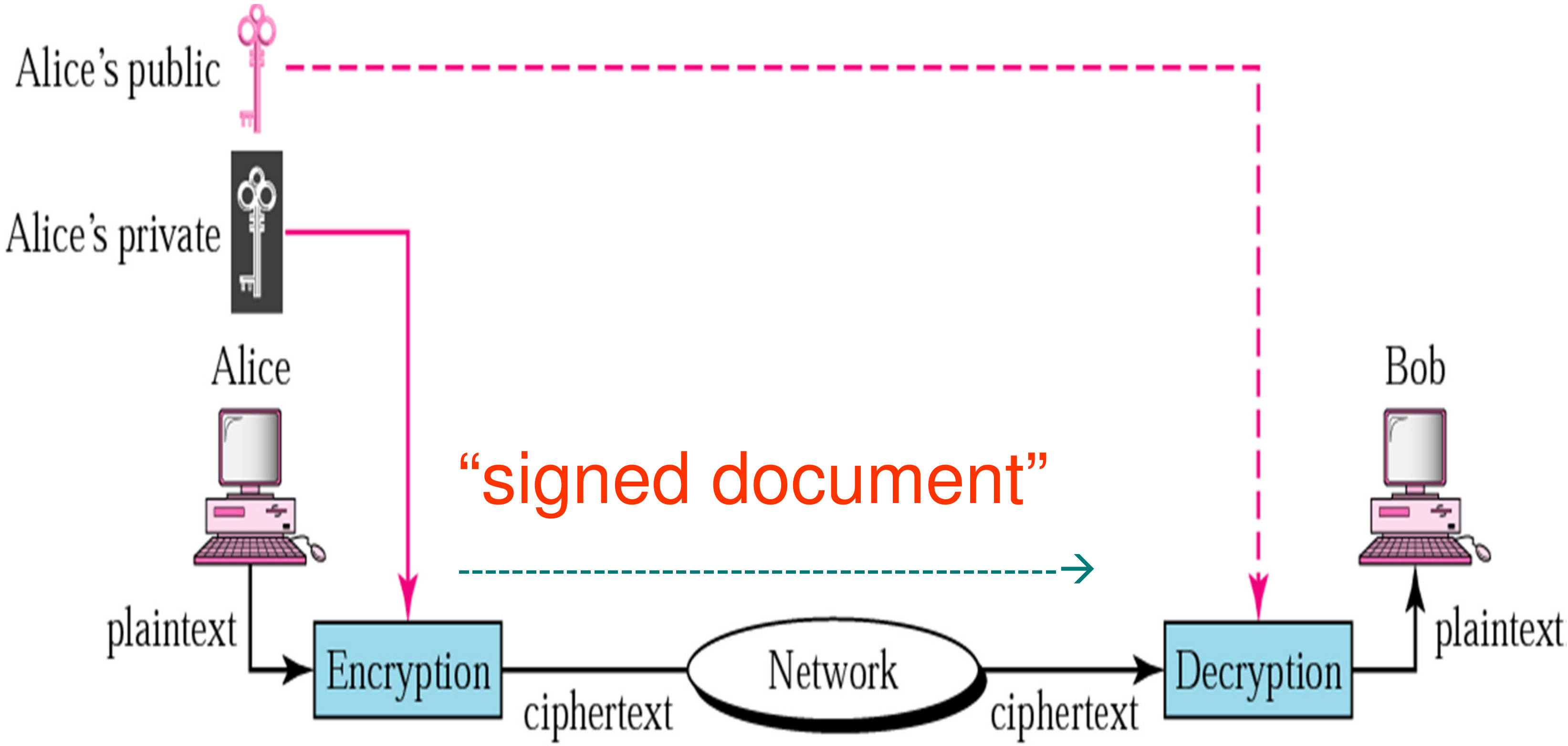


Figure 31.16

2. Message Authentication

- Means verifying the identity of a sender
- One method called ***digital signature*** is based on public key cryptography
- To ***prevent*** a user from ***repudiating*** the message that he has sent
- Additional Requirement: E(D(P)) = P
- (Both encryption and decryption are just transformation algorithms)

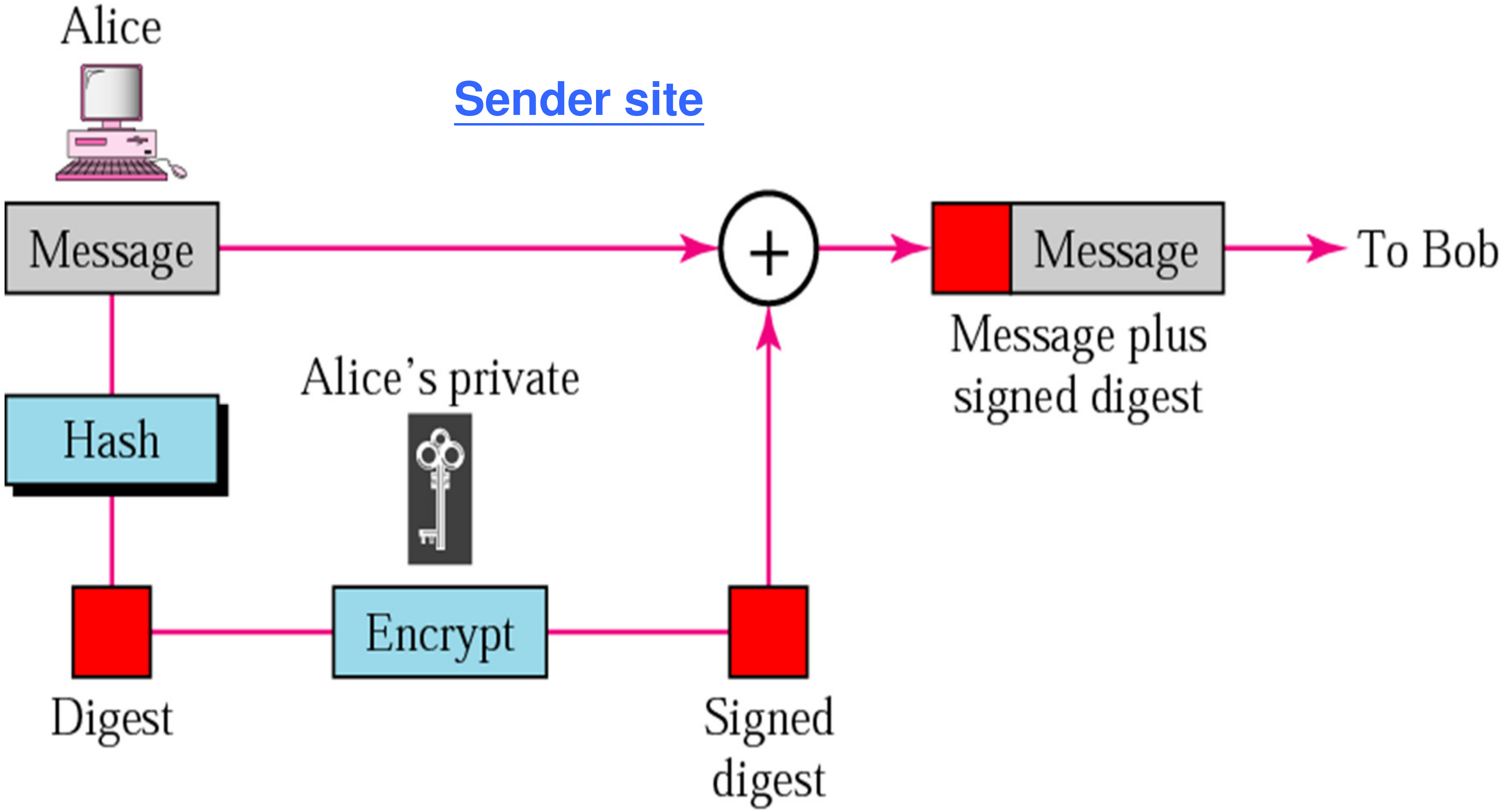
Signing the whole document



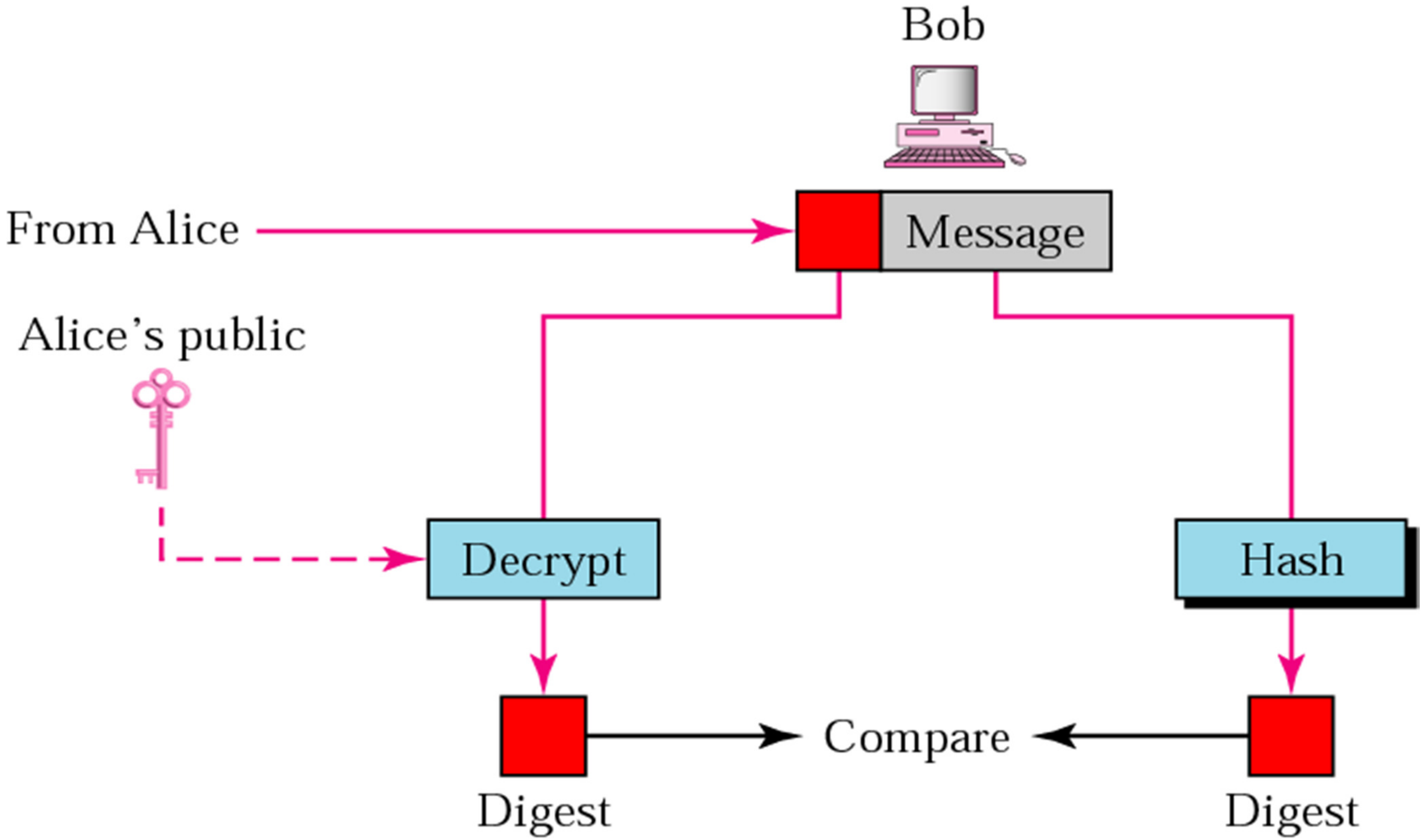
- Sender uses its own *private key* to sign (/encrypt)
- Receiver uses the sender's *public key* to verify (/decrypt)
- Digital signature does not provide privacy (i.e. secret of the message)

Signing the Digest

- **Digital Signature - Signing the Digest Only**



Receiver site (verify)



3. Digital Signature together with Encryption

- For user A, denote
  - E_A = public key
  - D_A = private key
  - E_A (P) = encrypt message P using the key E_A
  - D_A (P) = decrypt message P using the key D_A

- The encryption and decryption algorithms should have the property that
  - D(E(P)) = P
  - E(D(P)) = P

Digital Signature together with Encryption

- User A sends a message P to user B by transmitting $E_B(D_A(P))$
- B decrypts the ciphertext using its own private key:
  - $D_B(E_B(D_A(P))) = D_A(P)$
- User B **stores $D_A(P)$** in a safe place and then decrypts it (check A's signature) using the public key E_A of user A to get the original message P
- **Message Nonrepudiation**
- When A denies having sent the message P to B
  - User B can show both P and $D_A(P)$ as evidence
  - (since $D_A(P)$ can only be produced by user A)

Summary

- Cryptography
  - Symmetric-Key Cryptography
  - Asymmetric-key cryptography

- Security Aspects
  - Message Integrity
  - Message Authentication
  - Digital Signature

References

- Video on Distributed Denial of Service (DDOS) Attacks
  - http://www.youtube.com/watch?v=NogCN78XN2w
  - http://www.youtube.com/watch?v=SCcpauJp63c

- Revision Quiz
  - http://highered.mheducation.com/sites/0073376221/student_view0/chapter31/quizzes.html