

# Cloud Security

M.Sc. Sem III - 2022

Dr Mukti Padhya

Assistant Professor

NFSU, Gandhinagar

# Encryption

- Why Encryption
  - To achieve Confidentiality
  - Protection against unauthorized disclosure of information.
- What is Encryption?
- Types

# Encryption

- can characterize Encryption system by:
  - ❖ type of encryption operations used
    - Substitution ( each element of PT is mapped into another element)
    - Transposition ( elements in PT are rearranged)
    - Product ( combination of both)
  - ❖ number of keys used
    - single-key or private or symmetric
    - two-key or public or asymmetric
  - ❖ way in which plaintext is processed
    - block
    - stream

# Some Basic Terminology

❑ **plaintext** - original message

❑ **ciphertext** - coded message

❑ **cipher** - algorithm for transforming plaintext to ciphertext

❑ **key** - info used in cipher known only to sender/receiver

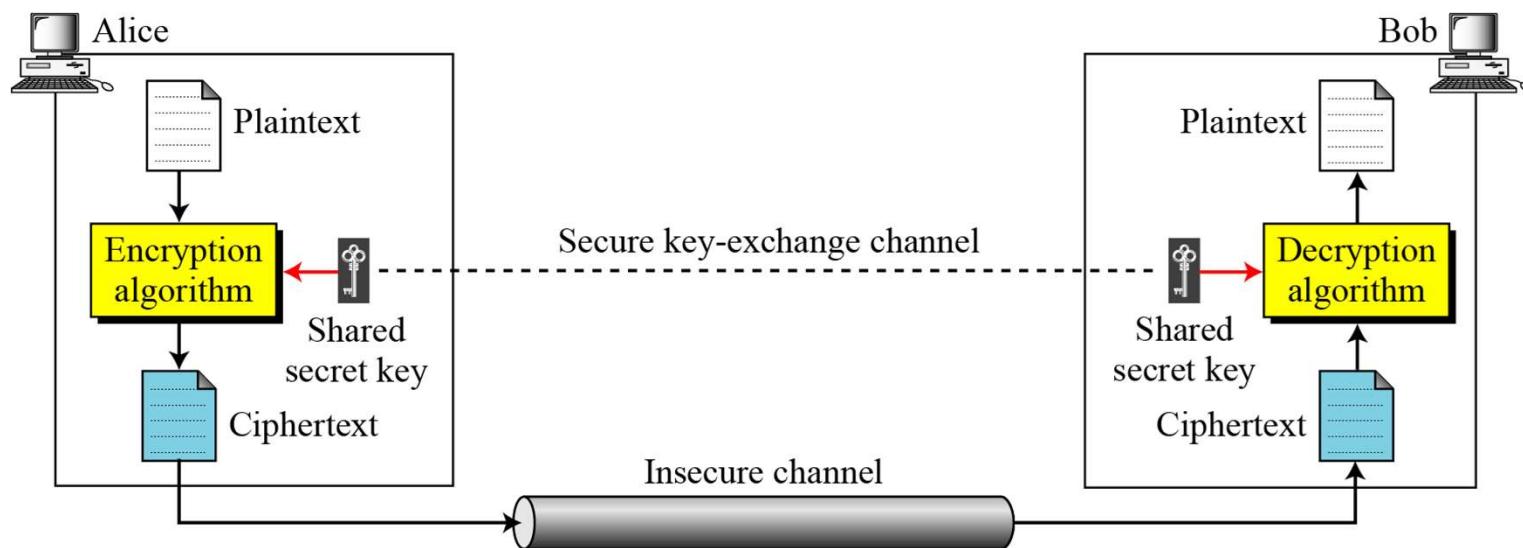
❑ **encipher (encrypt)** - converting plaintext to ciphertext

❑ **decipher (decrypt)** - recovering ciphertext from plaintext

❑ **cryptography** - study of encryption principles/methods

❑ **cryptanalysis (codebreaking)** - study of principles/ methods of deciphering ciphertext *without* knowing key

# symmetric-key cipher



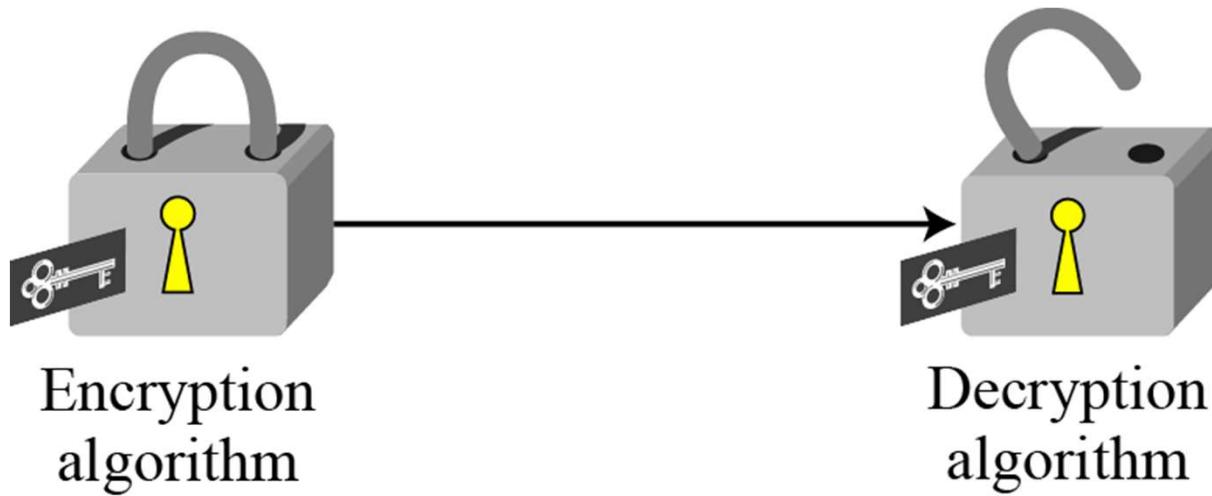
If P is the plaintext, C is the ciphertext, and K is the key,

Encryption:  $C = E_k(P)$

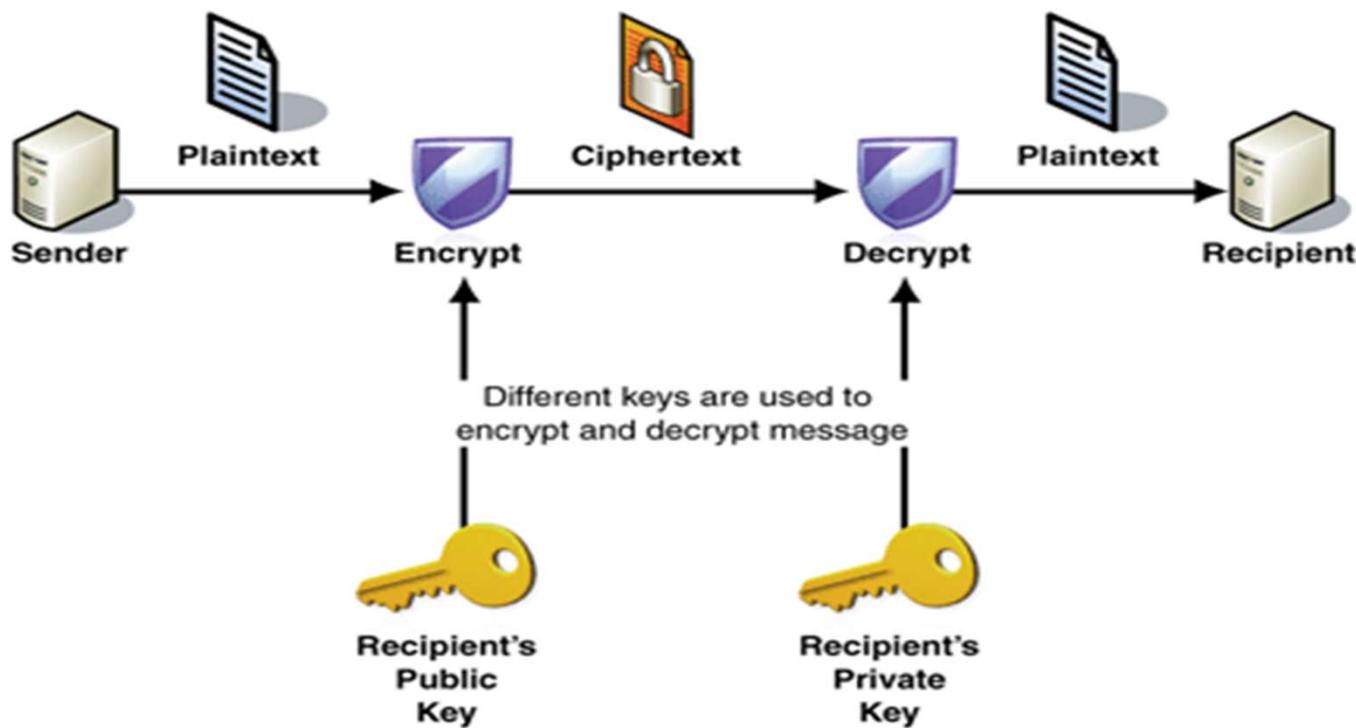
Decryption:  $P = D_k(C)$

In which,  $D_k(E_k(x)) = E_k(D_k(x)) = x$

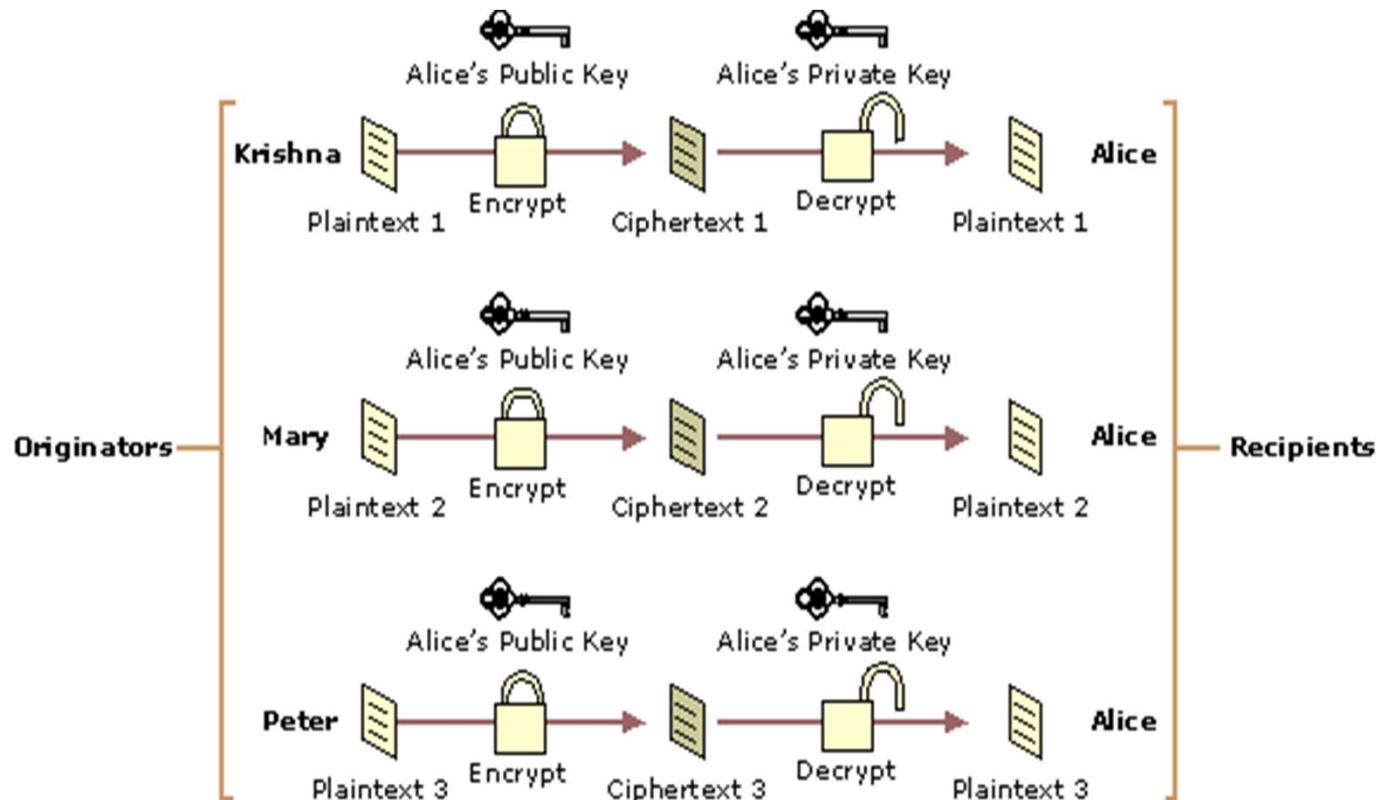
Locking and unlocking  
with the same key



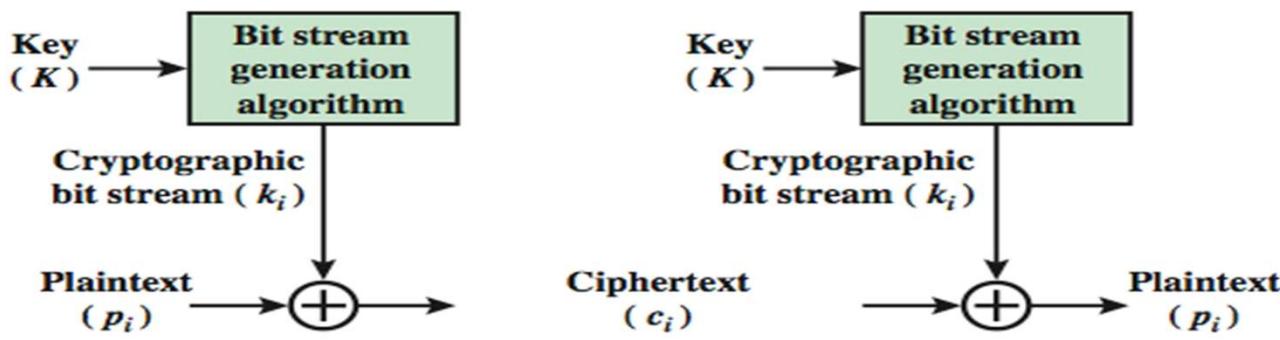
# Asymmetric-key Cipher



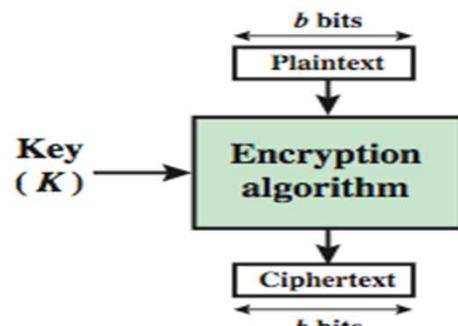
# Asymmetric-key Cipher



# Block Vs Stream Ciphers



(a) Stream Cipher Using Algorithmic Bit Stream Generator



Dr Mukti Padhye : Cloud Sec./@ MSc. CS 2022  
(b) Block Cipher

# Block vs Stream Ciphers

- block ciphers process messages in blocks, each of which is then en/decrypted
  - like a substitution on very big characters
  - 64-bits or more
- stream ciphers process messages a bit or byte at a time when en/decrypting
- many current ciphers are block ciphers
  - broader range of applications

# Stream Ciphers

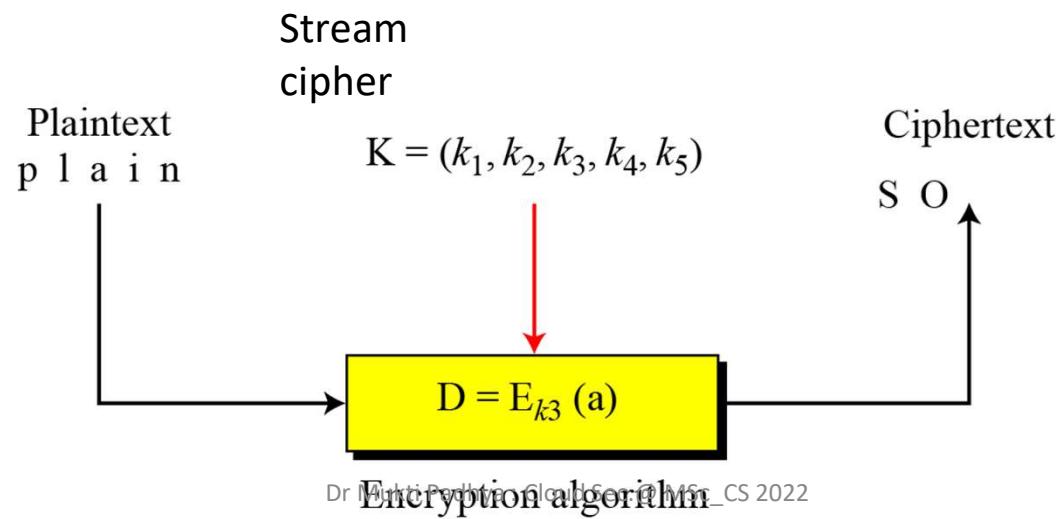
Call the plaintext stream P, the ciphertext stream C, and the key stream K.

$$P = P_1 P_2 P_3, \dots$$

$$C = C_1 C_2 C_3, \dots$$

$$\mathbf{K} = (k_1, k_2, k_3, \dots)$$

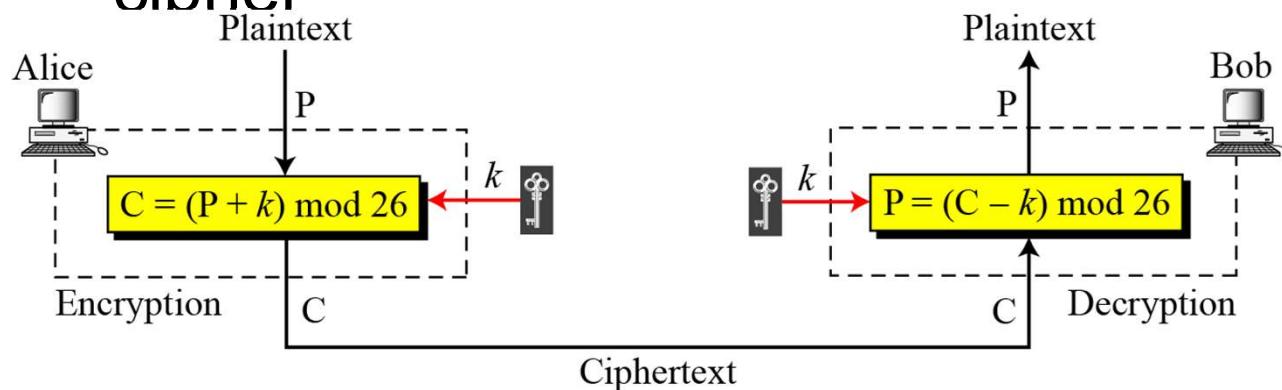
$$C_1 = E_{k1}(P_1) \quad C_2 = E_{k2}(P_2) \quad C_3 = E_{k3}(P_3) \dots$$



- 
- Additive ciphers can be categorized as stream ciphers in which the key stream is the repeated value of the key. In other words, the key stream is considered as a predetermined stream of keys or  $K = (k, k, \dots, k)$ .
  - The monoalphabetic substitution ciphers are also stream ciphers.
  - However, each value of the key stream in this case is the mapping of the current plaintext character to the corresponding ciphertext character in the mapping table.

# Continued

## Additive cipher



The simplest monoalphabetic cipher is the additive cipher. This cipher is sometimes called a **shift cipher** and sometimes a **Caesar cipher**, but the term additive cipher better reveals its mathematical nature.

When the cipher is additive, the plaintext, ciphertext, and key are integers in  $\mathbb{Z}_{26}$ .

## Continued

### Example

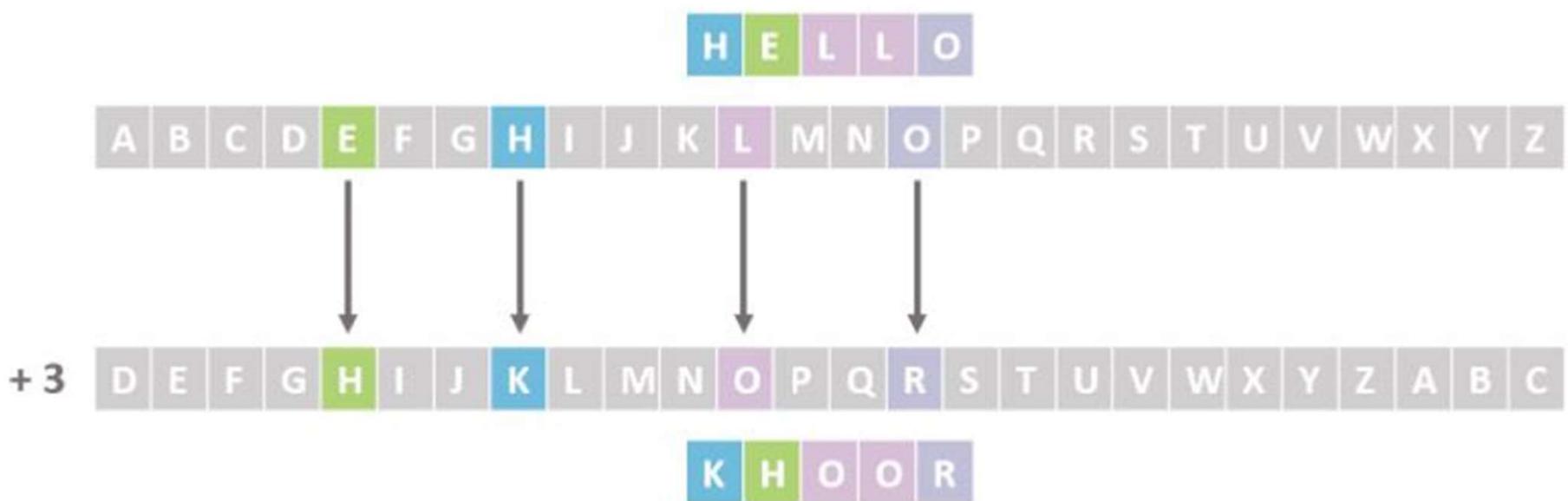
Use the additive cipher with key = 15 to encrypt the message “hello”.

a-0, b-1,c-2.....,z-25

### Solution

We apply the encryption algorithm to the plaintext, character by character:

Plaintext: h → 07	Encryption: $(07 + 15) \text{ mod } 26$	Ciphertext: $22 \rightarrow W$
Plaintext: e → 04	Encryption: $(04 + 15) \text{ mod } 26$	Ciphertext: $19 \rightarrow T$
Plaintext: l → 11	Encryption: $(11 + 15) \text{ mod } 26$	Ciphertext: $00 \rightarrow A$
Plaintext: l → 11	Encryption: $(11 + 15) \text{ mod } 26$	Ciphertext: $00 \rightarrow A$
Plaintext: o → 14	Encryption: $(14 + 15) \text{ mod } 26$	Ciphertext: $03 \rightarrow D$



- mathematically give each letter a number

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	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

- then have Caesar cipher as:

$$c = E(p) = (p + k) \bmod (26)$$

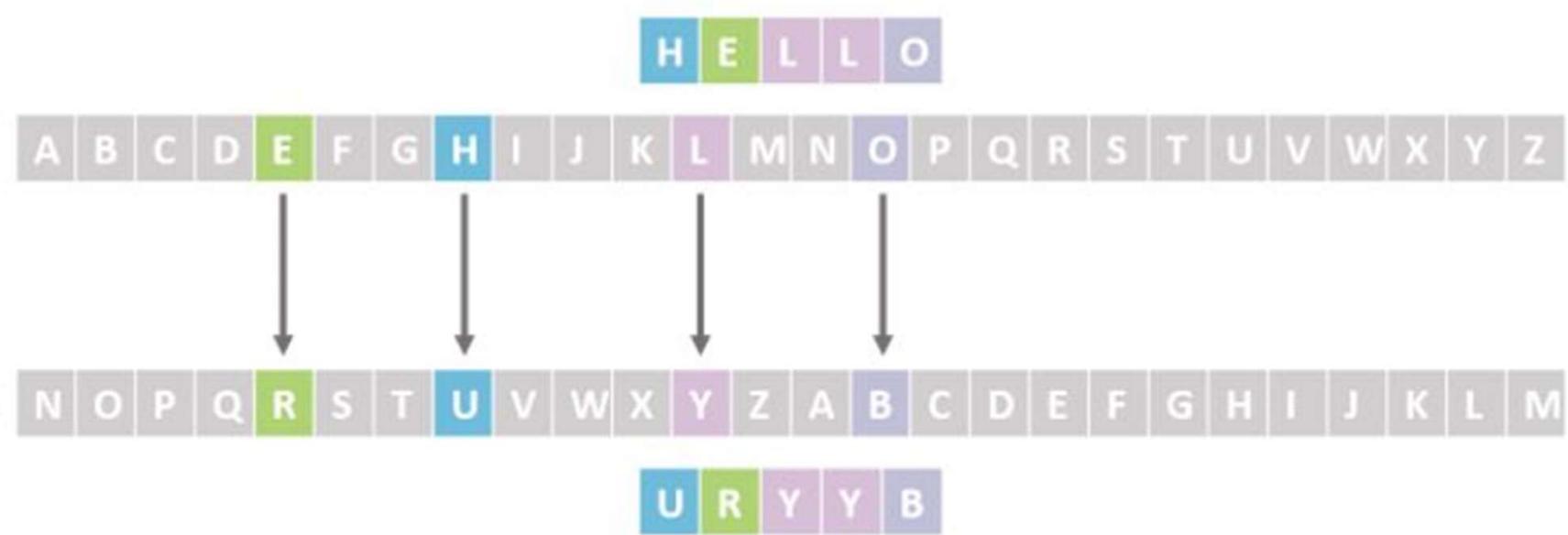
$$p = D(c) = (c - k) \bmod (26)$$

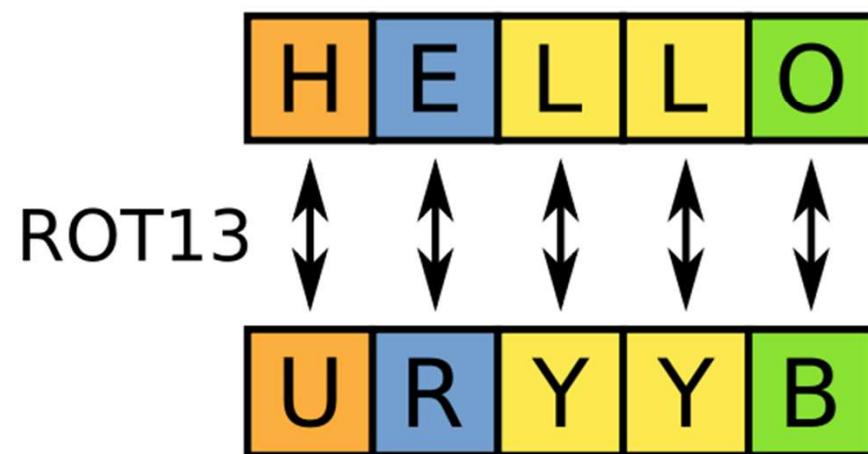
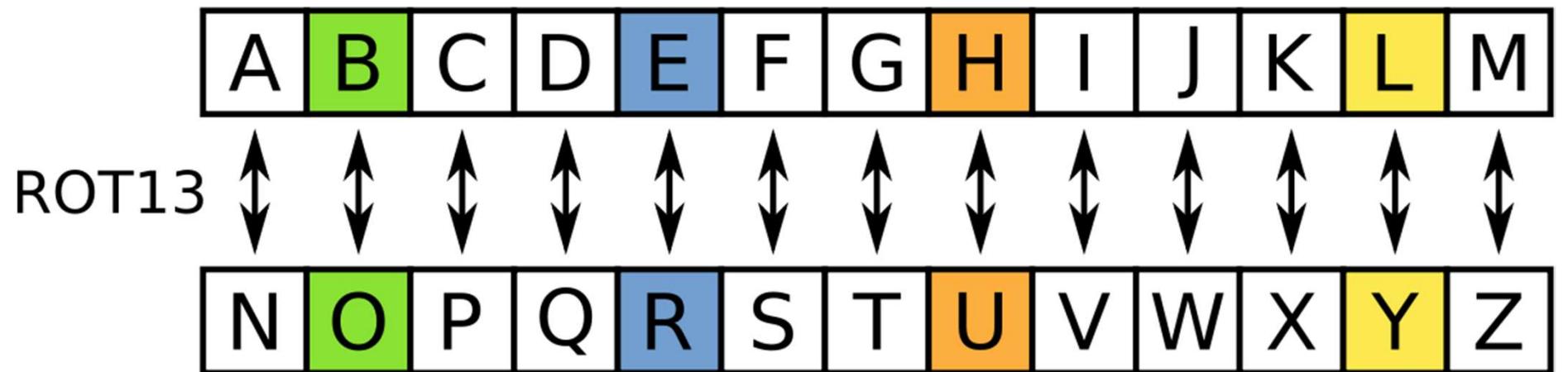
Plaintext : HELLO Key =13 Ciphertext : ?

Note: mod operation  $\equiv$   $x \bmod n$  then if  $x > n$  and  $n$  can divide  $x$  so output of mod is ZERO

If  $x < n$  then output will be  $x$  itself

If  $x$  is negative then output will be  $x+n$





## Continued

### Monoalphabetic Substitution Cipher

- Because additive, multiplicative, and affine ciphers have small key domains, they are very vulnerable to brute-force attack.
- A better solution is to create a mapping between each plaintext character and the corresponding ciphertext character. Alice and Bob can agree on a table showing the mapping for each character.

An example key for monoalphabetic substitution cipher

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
Ciphertext →	N	O	A	T	R	B	E	C	F	U	X	D	Q	G	Y	L	K	H	V	I	J	M	P	Z	S	W

## Continued

Example  
encrypt the message

this message is easy to encrypt but hard to find the key

The ciphertext is

ICFVQRVVNEFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

Vigenere ciphers are also stream ciphers according to the definition.

In this case, the key stream is a repetition of m values, where m is the size of the keyword.

In other words,

$$K = (k_1, k_2, \dots, k_m, k_1, k_2, \dots, k_m, \dots)$$

# Polyalphabetic Ciphers

In polyalphabetic substitution, each occurrence of a character may have a different substitute. The relationship between a character in the plaintext to a character in the ciphertext is one-to-many.

Autokey

Cipher

$$P = P_1 P_2 P_3 \dots$$

$$C = C_1 C_2 C_3 \dots$$

$$k = (k_1, P_1, P_2, \dots)$$

$$\text{Encryption: } C_i = (P_i + k_i) \bmod 26$$

$$\text{Decryption: } P_i = (C_i - k_i) \bmod 26$$

## Continued

### Vigenere Cipher

$$P = P_1 P_2 P_3 \dots$$

$$C = C_1 C_2 C_3 \dots$$

$$K = [(k_1, k_2, \dots, k_m), (k_1, k_2, \dots, k_m), \dots]$$

$$\text{Encryption: } C_i = P_i + k_i$$

$$\text{Decryption: } P_i = C_i - k_i$$

Example  
We can encrypt the message “She is listening” using the 6-character keyword “PASCAL”.

# Continued

## Example

Let us see how we can encrypt the message “She is listening” using the 6-character keyword “PASCAL”. The initial key stream is (15, 0, 18, 2, 0, 11). The key stream is the repetition of this initial key stream (as many times as needed).

<b>Plaintext:</b>	s      h      e      i      s      l	i      s      t      e      n      i	n      g
<b>P's values:</b>	18    07    04    08    18    11	08    18    19    04    13    08	13    06
<b>Key stream:</b>	<b>15    00    18    02    00    11</b>	<b>15    00    18    02    00    11</b>	<b>15    00</b>
<b>C's values:</b>	<b>07    07    22    10    18    22</b>	<b>23    18    11    6    13    19</b>	<b>02    06</b>
<b>Ciphertext:</b>	<b>H    H    W    K    S    W</b>	<b>X    S    L    G    N    T</b>	<b>C    G</b>

Ciphertext : DATG

Key : 19

### DECRYPTION

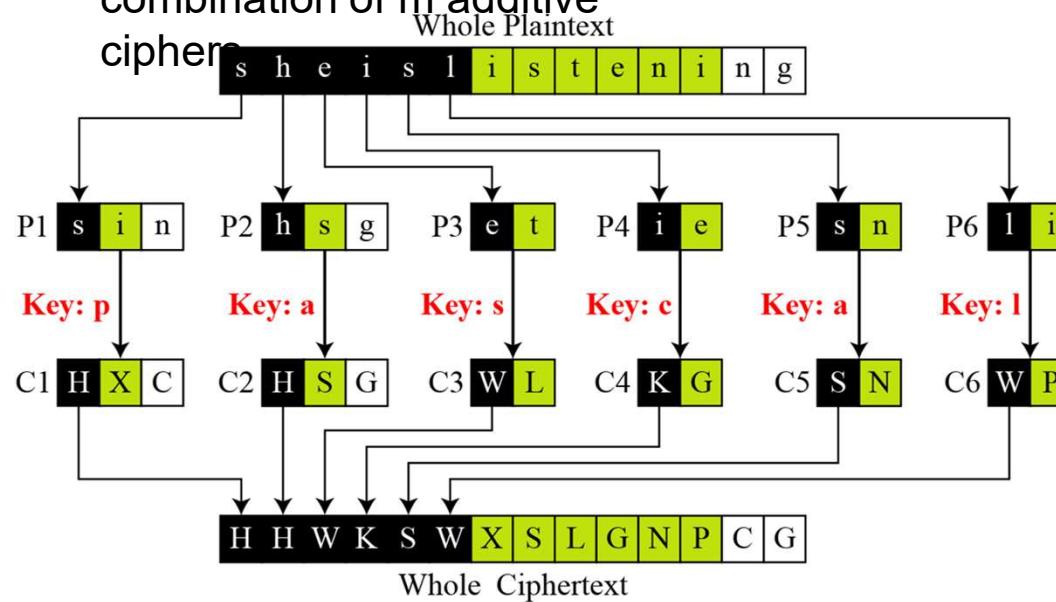
$$\begin{array}{r} \text{D} \quad \text{A} \quad \text{T} \quad \text{G} \\ 3 \quad 0 \quad 19 \quad 6 \\ - \quad 19 \quad 19 \quad 19 \quad 19 \\ \hline ( \quad -16 \quad -19 \quad 0 \quad -13 \quad ) \text{ mod } 26 \\ 10 \quad 7 \quad 0 \quad 13 \\ \hline \text{K} \quad \text{H} \quad \text{A} \quad \text{N} \end{array}$$

# Continued

## Example

Vigenere cipher can be seen as combinations of m additive ciphers.

A Vigenere cipher as a combination of m additive ciphers



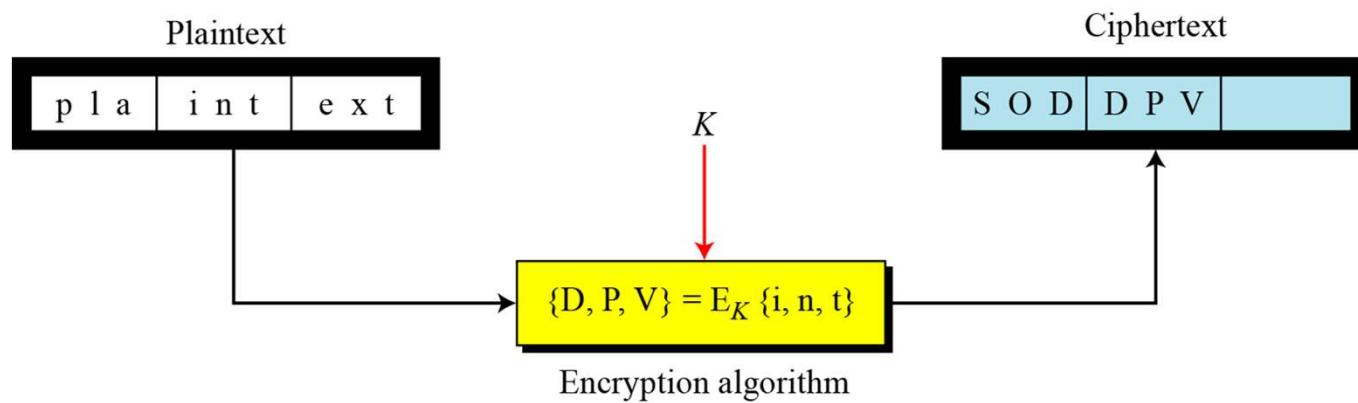
# Continued

## A Vigenere Table

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	v	v	w	x	y	z
A	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
B	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	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	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

# Block Ciphers

- In a block cipher, a group of plaintext symbols of size  $m$  ( $m > 1$ ) are encrypted together creating a group of ciphertext of the same size.
- A single key is used to encrypt the whole block even if the key is made of multiple values.





- Playfair ciphers are block ciphers. The size of the block is  $m = 2$ . Two characters are encrypted together.
- Hill ciphers are block ciphers. A block of plaintext, of size 2 or more is encrypted together using a single key (a matrix).
- Although the key is made of  $m \times m$  values, it is considered as a single key.

# Playfair Cipher

- not even the large number of keys in a monoalphabetic cipher provides security
- one approach to improving security was to encrypt multiple letters
- the **Playfair Cipher** is an example
- invented by Charles Wheatstone in 1854, but named after his friend Baron Playfair

# Playfair Key Matrix

- a 5X5 matrix of letters based on a keyword
- fill in letters of keyword (without duplicates)
- fill rest of matrix with other letters
- eg. using the keyword MONARCHY

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

# Encrypting and Decrypting

plaintext is encrypted two letters at a time

- if a pair is a repeated letter, insert filler like 'X'
- ABC AB CX

*	*	*	*	*	*
*	O	Y	R	Z	
*	*	*	*	*	
*	*	*	*	*	
*	*	*	*	*	

*	*	O	*	*
*	*	B	*	*
*	*	*	*	*
*	*	R	*	*
*	*	Y	*	*

Hence, OR -> BY

Hence, OR -> YZ

Z	*	*	O	*
*	*	*	*	*
*	*	*	*	*
R	*	*	X	*
*	*	*	*	*

Hence, OR ->  
ZX

# Playfair Cipher

Examp

le An example of a secret key in the Playfair cipher

Secret Key =

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

encrypt the plaintext “hello”

he → EC

lx → QZ

lo → BX

Plaintext: hello

Ciphertext: ECQZBX

# Example

❑ Key: playfair example

❑ Message "Hide the gold in the tree stump"

❑ (note the null "X" used to separate the repeated "E"s)

# Example

-Key: playfair example 007

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

-Message "Hide the gold in the tree stump"

(note the null "X" used to separate the repeated "E"s)

HI DE TH EG OL DI NT HE TR EX ES TU MP



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

HI

Shape: Rectangle  
Rule: Pick Same Rows,  
Opposite Corners

BM

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

DE

Shape: Column  
Rule: Pick Items Below Each  
Letter, Wrap to Top if Needed

OD

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

TH

Shape: Rectangle  
Rule: Pick Same Rows,  
Opposite Corners

ZB

P	L	A	Y	F
I	R	E-X		M
B	C	D-G		H
K	N	O	Q	S
T	U	V	W	Z

EG

Shape: Rectangle  
Rule: Pick Same Rows,  
Opposite Corners

XD

- ⑥. The pair DI forms a rectangle, replace it with BE
- ⑦. The pair NT forms a rectangle, replace it with KU
- ⑧. The pair HE forms a rectangle, replace it with DM
- ⑨. The pair TR forms a rectangle, replace it with UI

P	L-A	Y	F
I	R E	X M	
B	C D	G H	
K	N-O	Q S	
T	U V	W Z	

OL

Shape: Rectangle  
Rule: Pick Same Rows,  
Opposite Corners

NA

- ¶11. The pair ES forms a rectangle, replace it with MO
  - ¶12. The pair TU is in a row, replace it with UV
  - ¶13. The pair MP forms a rectangle, replace it with IF
- ¶Ciphertext: BM OD ZB XD NA BE KU DM UI XM MO UV IF

# Hill Cipher

Key in the Hill cipher

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1m} \\ k_{21} & k_{22} & \dots & k_{2m} \\ \vdots & \vdots & & \vdots \\ k_{m1} & k_{m2} & \dots & k_{mm} \end{bmatrix}$$

$$\begin{aligned} C_1 &= P_1 k_{11} + P_2 k_{21} + \dots + P_m k_{m1} \\ C_2 &= P_1 k_{12} + P_2 k_{22} + \dots + P_m k_{m2} \\ &\dots \\ C_m &= P_1 k_{1m} + P_2 k_{2m} + \dots + P_m k_{mm} \end{aligned}$$

The key matrix in the Hill cipher needs to have a multiplicative inverse.

# Continu

## Examp le

For example, the plaintext “code is ready” can make a  $3 \times 4$  matrix when adding extra bogus character “z” to the last block and removing the spaces. The ciphertext is “OHKNIHGKLSS”.

$$\begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}^C = \begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}^P \begin{bmatrix} 09 & 07 & 11 & 13 \\ 04 & 07 & 05 & 06 \\ 02 & 21 & 14 & 09 \\ 03 & 23 & 21 & 08 \end{bmatrix}^K$$

a. Encryption

$$\begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}^P = \begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}^C \begin{bmatrix} 02 & 15 & 22 & 03 \\ 15 & 00 & 19 & 03 \\ 09 & 09 & 03 & 11 \\ 17 & 00 & 04 & 07 \end{bmatrix}^{K^{-1}}$$

b. Decryption

# Example:

Message: “ pay more money”

PAY 15 0 24

Key : 17 17 5  
21 18 21  
2 2 19

15  
0  
24

$$P.T = \begin{bmatrix} p & m & e & n \\ a & o & m & e \\ y & r & o & y \end{bmatrix} = \begin{bmatrix} 15 & 12 & 4 & 13 \\ 0 & 14 & 12 & 4 \\ 24 & 17 & 14 & 24 \end{bmatrix}$$

$$P.T_1 = \begin{bmatrix} p \\ a \\ y \end{bmatrix} = \begin{bmatrix} 15 \\ 0 \\ 24 \end{bmatrix}$$

$$C.T_1 = Key \times P.T_1 \bmod 26 = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix} \begin{bmatrix} 15 \\ 0 \\ 24 \end{bmatrix} \bmod 26 = \begin{bmatrix} 11 \\ 13 \\ 18 \end{bmatrix} = \begin{bmatrix} L \\ N \\ S \end{bmatrix}$$

$$C.T_2 = Key \times P.T_2 \bmod 26 = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix} \begin{bmatrix} 12 \\ 14 \\ 17 \end{bmatrix} \bmod 26 = \begin{bmatrix} 7 \\ 3 \\ 11 \end{bmatrix} = \begin{bmatrix} H \\ D \\ L \end{bmatrix}$$

► And so on... then the C.T = **LNS HDL.....**

# Example

② Ciphertext : WIIT

② Key: 3 7  
      1 5

Message: ?

HELP

## Inverse : 2 x 2 Matrix

$$\begin{bmatrix} 4 & 7 \\ 2 & 6 \end{bmatrix}^{-1} = \frac{1}{4 \times 6 - 7 \times 2} \begin{bmatrix} 6 & -7 \\ -2 & 4 \end{bmatrix}$$
$$= \frac{1}{10} \begin{bmatrix} 6 & -7 \\ -2 & 4 \end{bmatrix}$$
$$= \begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.4 \end{bmatrix}$$

## Continu

### Examp le

Assume that Eve knows that  $m = 3$ . She has intercepted three plaintext/ciphertext pair blocks (not necessarily from the same message)

$$\begin{bmatrix} 05 & 07 & 10 \end{bmatrix} \longleftrightarrow \begin{bmatrix} 03 & 06 & 00 \end{bmatrix}$$

$$\begin{bmatrix} 13 & 17 & 07 \end{bmatrix} \longleftrightarrow \begin{bmatrix} 14 & 16 & 09 \end{bmatrix}$$

$$\begin{bmatrix} 00 & 05 & 04 \end{bmatrix} \longleftrightarrow \begin{bmatrix} 03 & 17 & 11 \end{bmatrix}$$

P

C

## Example (Continued)

She makes matrices P and C from these pairs. Because P is invertible, she inverts the P matrix and multiplies it by C to get the K matrix

$$\begin{bmatrix} 02 & 03 & 07 \\ 05 & 07 & 09 \\ 01 & 02 & 11 \end{bmatrix} = \begin{bmatrix} 21 & 14 & 01 \\ 00 & 08 & 25 \\ 13 & 03 & 08 \end{bmatrix}^{-1} \begin{bmatrix} 03 & 06 & 00 \\ 14 & 16 & 09 \\ 03 & 17 & 11 \end{bmatrix}$$

Now she has the key and can break any ciphertext encrypted with that key.

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \\ 5 & 6 & 0 \end{pmatrix} \quad M = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \\ 5 & 6 & 0 \end{pmatrix} \quad M = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \\ 5 & 6 & 0 \end{pmatrix}$$

$\hookrightarrow$  ex.  $|((1 \times 0) - (6 \times 4))|$

$$\det(M) = |(0 - 24) - 2(0 - 20) + 3(0 - 5)| \\ = 1$$

wikiHow

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \\ 5 & 6 & 0 \end{pmatrix}$$

$$M = \begin{pmatrix} 1 & 0 & 5 \\ 2 & 1 & 6 \\ 3 & 4 & 0 \end{pmatrix}$$

wikiHow

$$M = \begin{pmatrix} 1 & 0 & 5 \\ 2 & 1 & 6 \\ 3 & 4 & 0 \end{pmatrix}$$

$$M = \begin{pmatrix} 1 & 0 & 5 \\ 2 & 1 & 6 \\ 3 & 4 & 0 \end{pmatrix}$$

$$M_{21} = \begin{vmatrix} 0 & 5 \\ 4 & 0 \end{vmatrix} = -20$$

$$M_{11} = \begin{vmatrix} 1 & 6 \\ 4 & 0 \end{vmatrix} = -24$$

$$M_{12} = \begin{vmatrix} 2 & 6 \\ 3 & 0 \end{vmatrix} = -18$$

$$M_{13} = \begin{vmatrix} 2 & 1 \\ 3 & 4 \end{vmatrix} = 5$$

$$M_{21} = \begin{vmatrix} 0 & 5 \\ 4 & 0 \end{vmatrix} = -20$$

$$M_{22} = \begin{vmatrix} 1 & 5 \\ 3 & 0 \end{vmatrix} = -15$$

$$M_{23} = \begin{vmatrix} 1 & 0 \\ 3 & 4 \end{vmatrix} = 4$$

$$M_{31} = \begin{vmatrix} 0 & 5 \\ 1 & 6 \end{vmatrix} = -5$$

$$M_{32} = \begin{vmatrix} 1 & 5 \\ 2 & 6 \end{vmatrix} = -4$$

$$M_{33} = \begin{vmatrix} 1 & 0 \\ 2 & 1 \end{vmatrix} = 1$$

wikiHow

$$M^T = \begin{pmatrix} 1 & 0 & 5 \\ 2 & 1 & 6 \\ 3 & 4 & 0 \end{pmatrix}$$

↳  $M = \begin{pmatrix} -24 & -18 & 5 \\ -20 & -15 & 4 \\ -5 & -4 & 1 \end{pmatrix} \times \begin{pmatrix} + & - & + \\ - & + & - \\ + & - & + \end{pmatrix}$

↳  $M = \begin{pmatrix} -24 & 18 & 5 \\ 20 & -15 & -4 \\ -5 & 4 & 1 \end{pmatrix}$

wikiHow

$$M^{-1} = \frac{1}{\det(M)} (\text{adj}(M)) \quad \det(M) = 1$$

$$\text{adj}(M) = \begin{pmatrix} -24 & 18 & 5 \\ 20 & -15 & -4 \\ -5 & 4 & 1 \end{pmatrix}$$

$M^{-1} = \frac{1}{1} \begin{pmatrix} -24 & 18 & 5 \\ 20 & -15 & -4 \\ -5 & 4 & 1 \end{pmatrix}$

$M^{-1} = \frac{1}{1} \begin{pmatrix} -24 & 18 & 5 \\ 20 & -15 & -4 \\ -5 & 4 & 1 \end{pmatrix}$

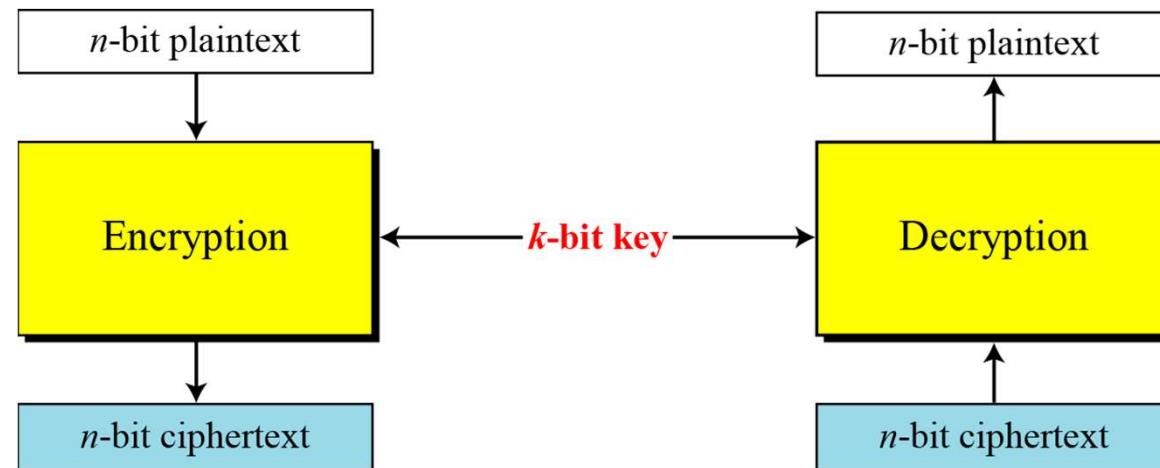
$$M^{-1} = \begin{pmatrix} -24 & 18 & 5 \\ 20 & -15 & -4 \\ -5 & 4 & 1 \end{pmatrix}$$

## MODERN BLOCK CIPHERS

A symmetric-key modern block cipher encrypts an n-bit block of plaintext or decrypts an n-bit block of ciphertext.

The encryption or decryption algorithm uses a k-bit key.

## *A modern block cipher*



## **Substitution or Transposition**

A modern block cipher can be designed to act as a substitution cipher or a transposition cipher.

Fiestel Structure 16 rounds

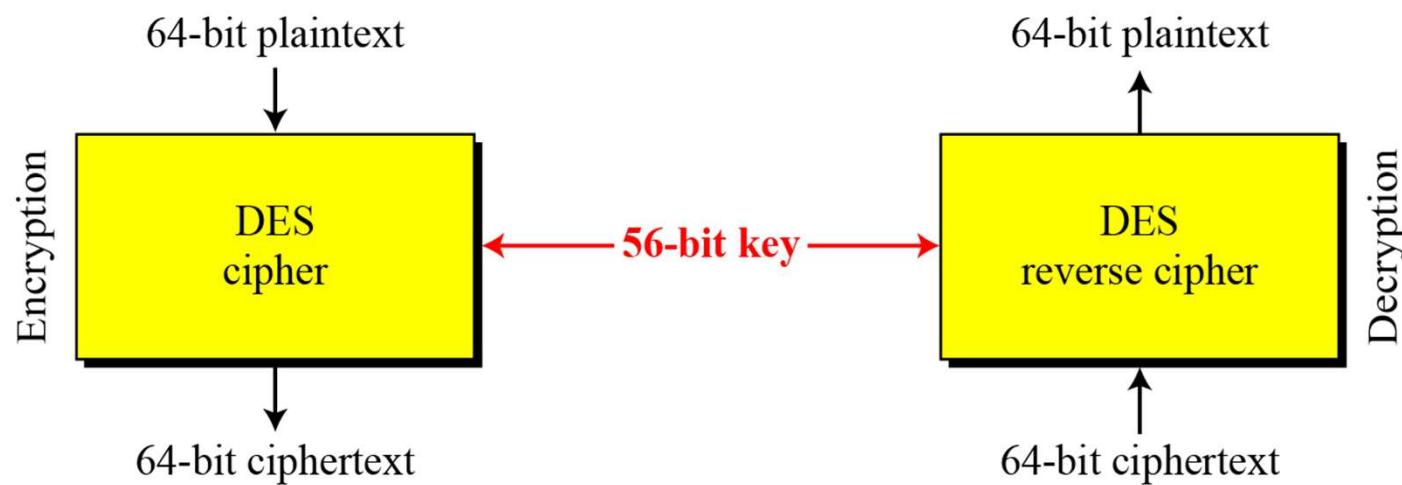
Example DES AES

To be resistant to exhaustive-search attack,  
a modern block cipher needs to be  
designed as a substitution cipher.

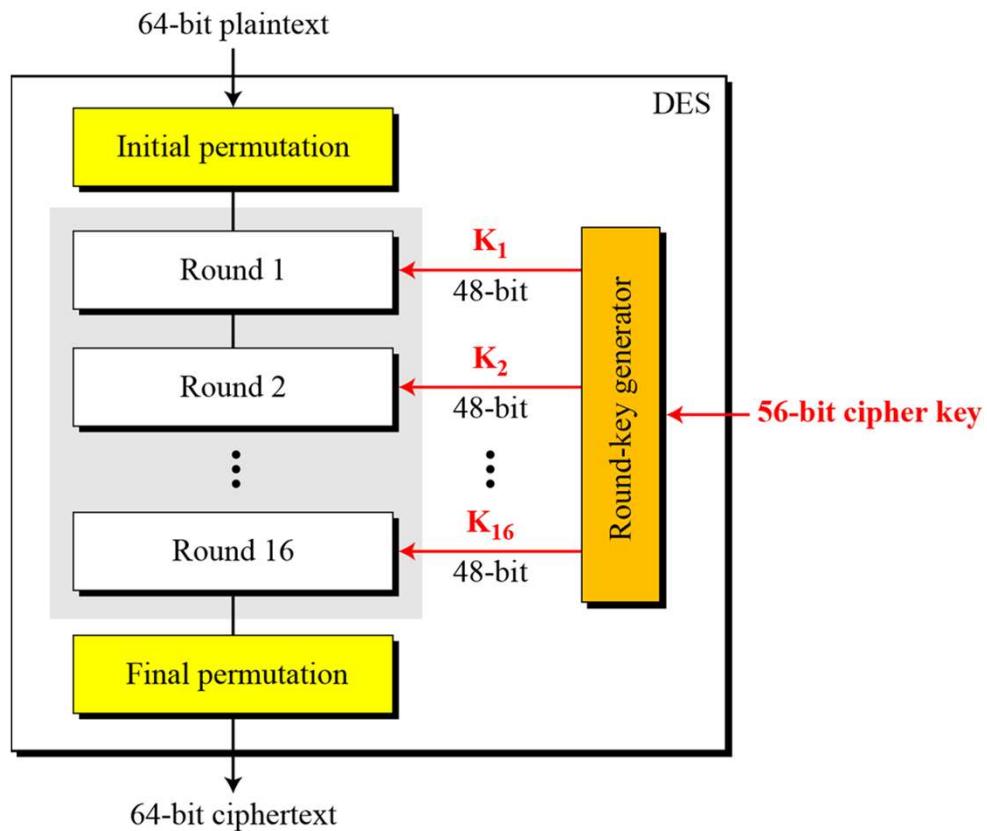
# Encryption

- can characterize Encryption system by:
  - ❖ type of encryption operations used
    - Substitution ( each element of PT is mapped into another element)
      - HELLO → AXMRW
      - Transposition ( elements in PT are rearranged)
        - HELLO → LHLOE
      - Product ( combination of both)
    - ❖ number of keys used
      - single-key or private or symmetric
      - two-key or public or asymmetric
    - ❖ way in which plaintext is processed
      - block
      - stream

## *Encryption and decryption with DES*



# DES STRUCTURE



Continued

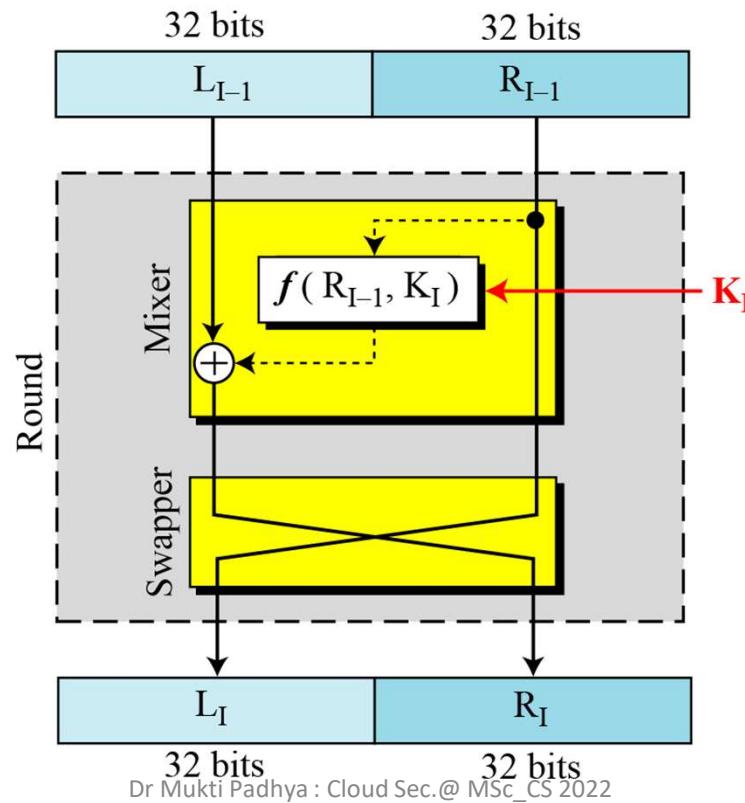
The initial and final permutations are straight P-boxes that  
are inverses of each other.  
They have no cryptography significance in DES.

Round

s

*DES uses 16 rounds. Each round of DES is a Feistel cipher.*

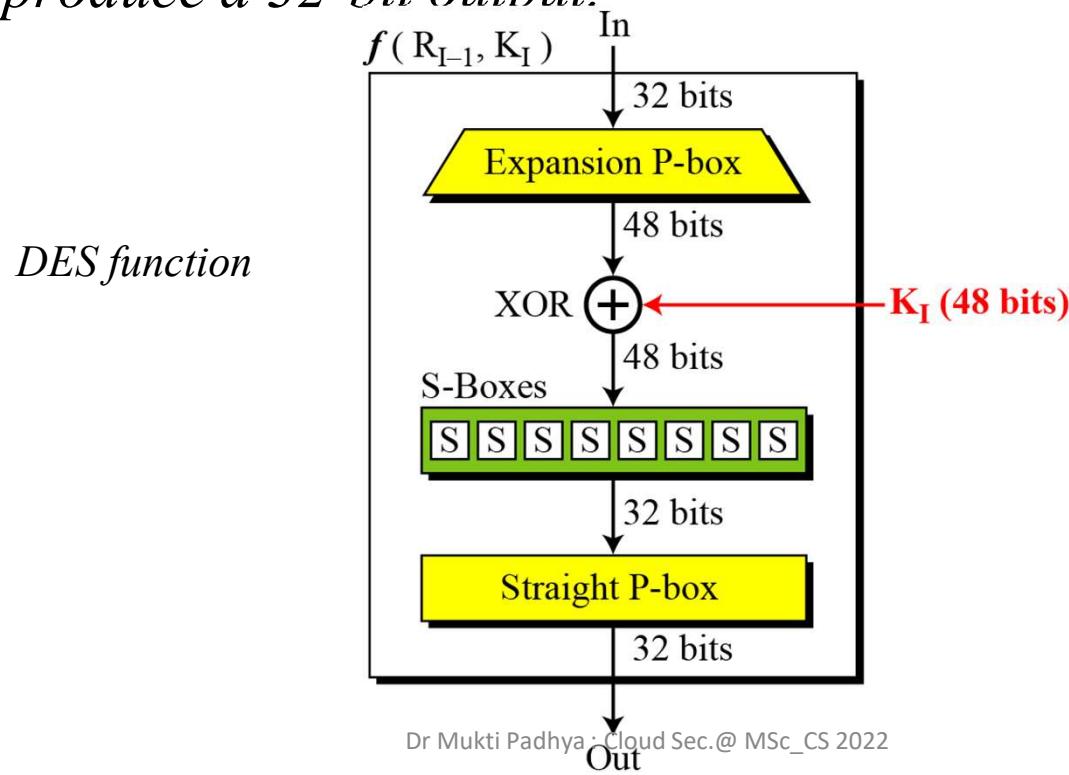
*A round in DES  
(encryption site)*



*Continued*

## DES Function

*The heart of DES is the DES function. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.*



# Asymmetric-Key Cryptography

## Note

Symmetric-key cryptography is based on sharing secrecy;  
asymmetric-key cryptography is based on personal secrecy.

# Symmetric-Key Cryptography

- traditional **private/secret/single key** cryptography uses **one key**
- shared by both sender and receiver
- if this key is disclosed communications are compromised
- also is **symmetric**, parties are equal
- hence does not protect sender from receiver forging a message & claiming is sent by sender

# Asymmetric-Key Cryptography

- probably most significant advance in the 3000 year history of cryptography
- uses **two** keys – a public & a private key
- **asymmetric** since parties are **not** equal
- uses clever application of number theoretic concepts to function
- complements **rather than** replaces private key crypto

# Why Public-Key Cryptography?

- developed to address two key issues:
  - **key distribution** – how to have secure communications in general without having to trust a KDC with your key
  - **digital signatures** – how to verify a message comes intact from the claimed sender
- public invention due to Whitfield Diffie & Martin Hellman at Stanford University in 1976
  - known earlier in classified community

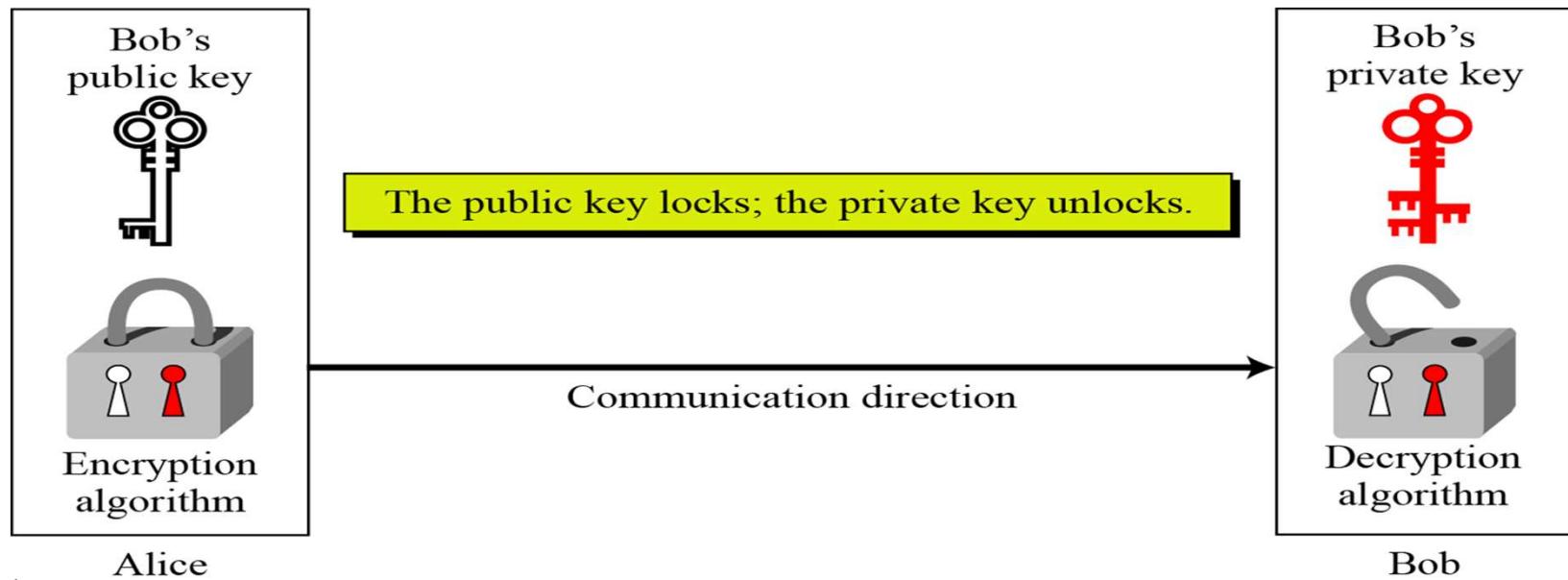
# Public-Key Cryptography

- **public-key/two-key/asymmetric** cryptography involves the use of **two keys**:
  - a **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
  - a related **private-key**, known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures**
- **infeasible to determine private key from public**
- **is asymmetric** because
  - those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures

## Keys

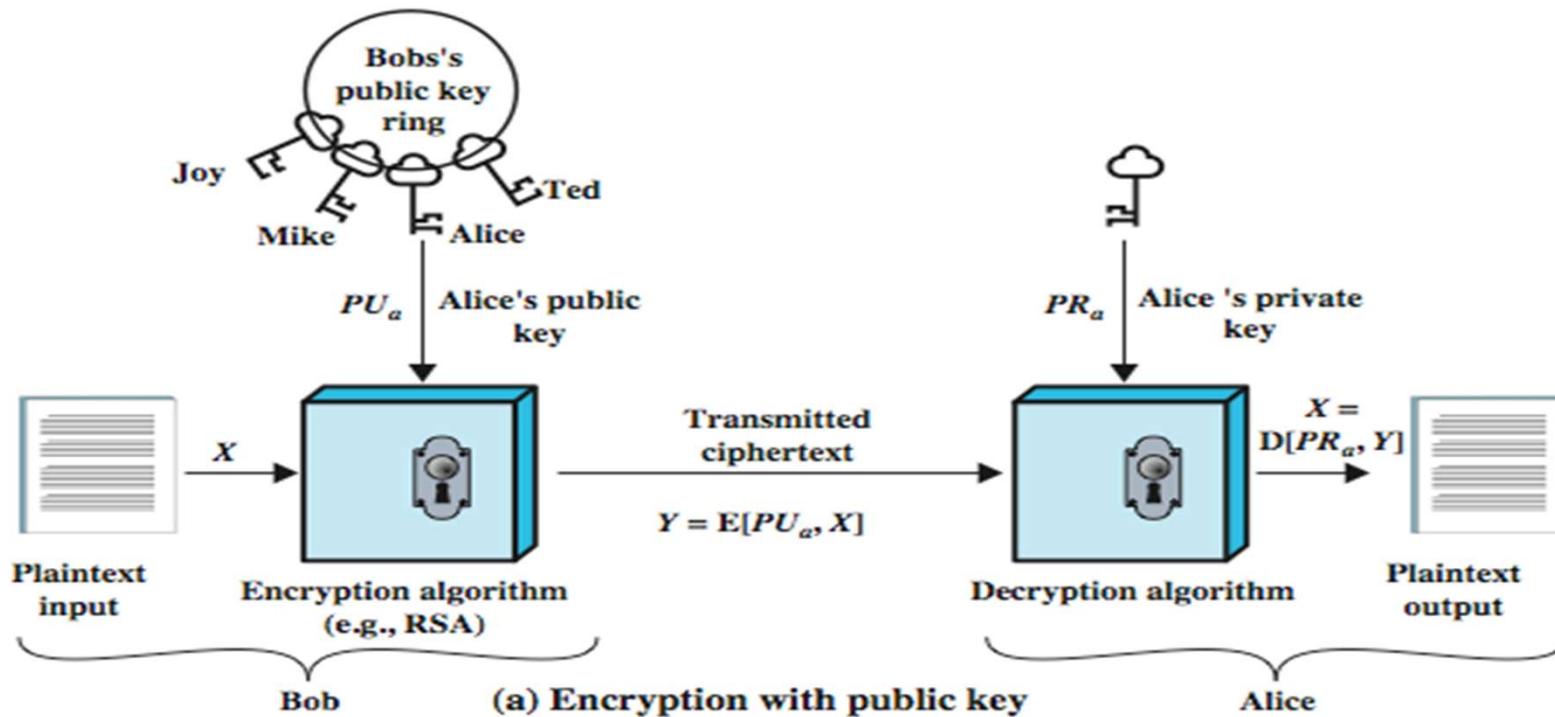
*Asymmetric key cryptography uses two separate keys: one private and one public.*

*Locking and unlocking in asymmetric-key cryptosystem*



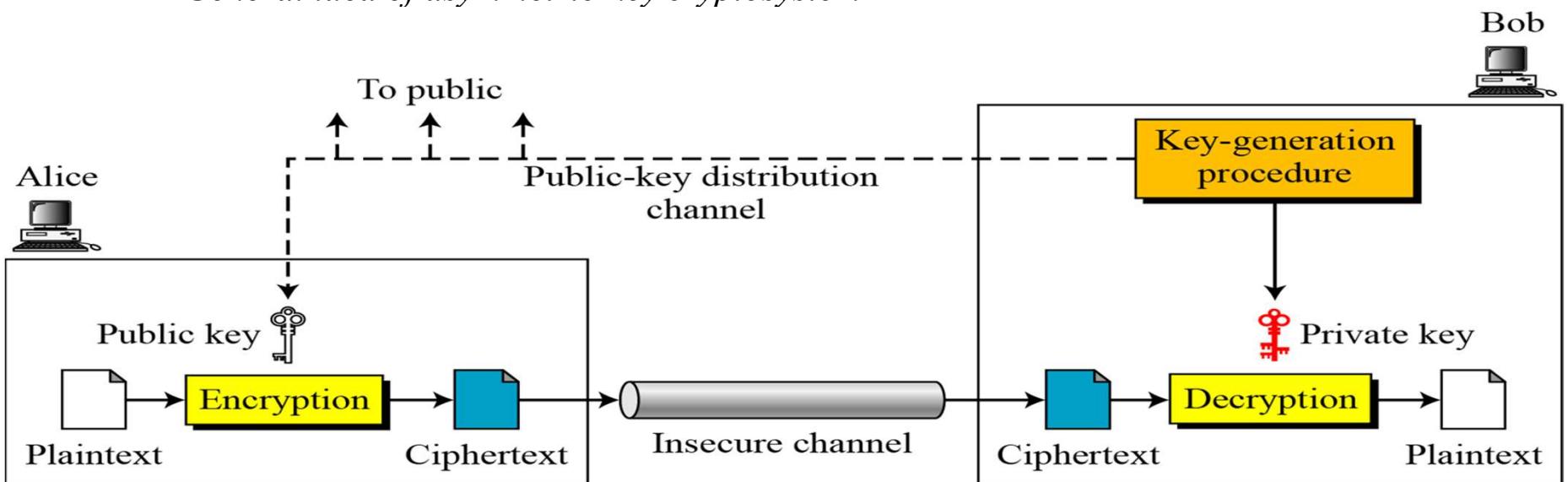
# Asymmetric/Public-Key Cryptography

## Many to one communication



## *General Idea*

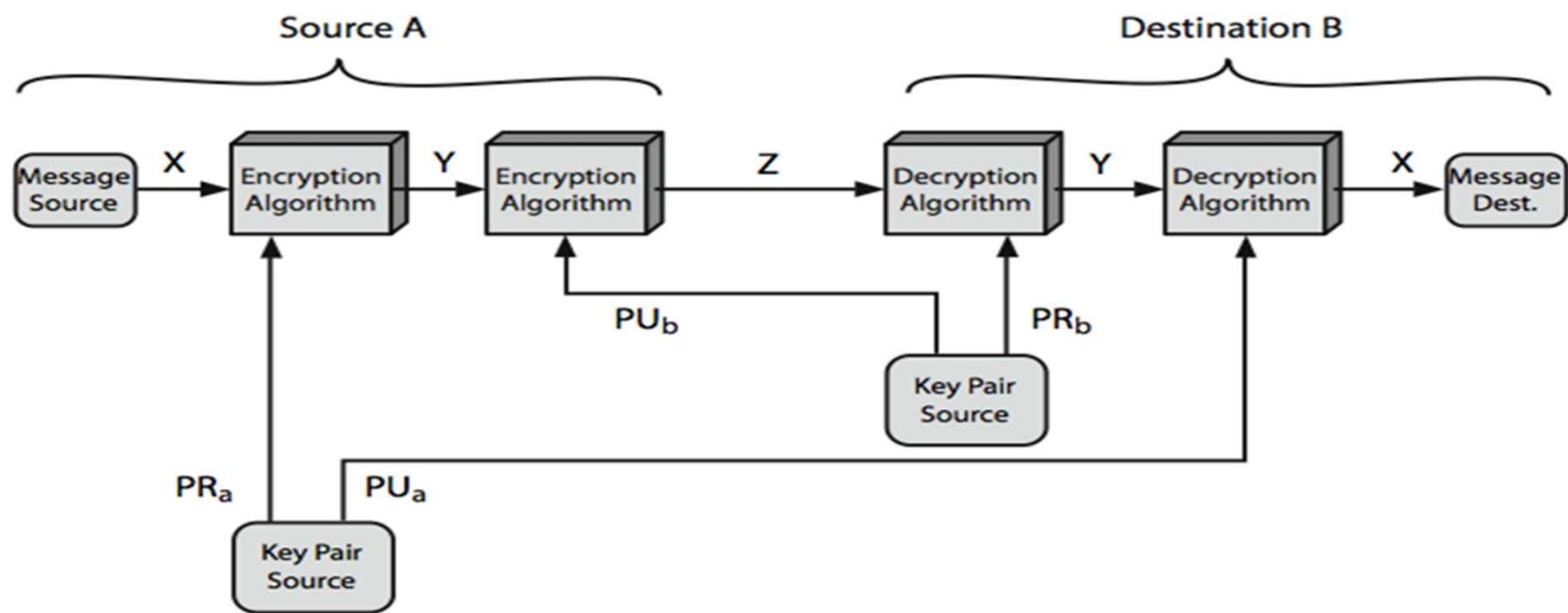
*General idea of asymmetric-key cryptosystem*



# Symmetric vs Public-Key

Conventional Encryption	Public-Key Encryption
<p><i>Needed to Work:</i></p> <ol style="list-style-type: none"><li>1. The same algorithm with the same key is used for encryption and decryption.</li><li>2. The sender and receiver must share the algorithm and the key.</li></ol> <p><i>Needed for Security:</i></p> <ol style="list-style-type: none"><li>1. The key must be kept secret.</li><li>2. It must be impossible or at least impractical to decipher a message if no other information is available.</li><li>3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.</li></ol>	<p><i>Needed to Work:</i></p> <ol style="list-style-type: none"><li>1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption.</li><li>2. The sender and receiver must each have one of the matched pair of keys (not the same one).</li></ol> <p><i>Needed for Security:</i></p> <ol style="list-style-type: none"><li>1. One of the two keys must be kept secret.</li><li>2. It must be impossible or at least impractical to decipher a message if no other information is available.</li><li>3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.</li></ol>

# Public-Key Cryptosystems



# Public-Key Applications

- can classify uses into 3 categories:
  - **encryption/decryption** (provide secrecy)
  - **digital signatures** (provide authentication)
  - **key exchange** (of session keys)
- some algorithms are suitable for all uses, others are specific to one

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

# Public-Key Requirements

- Public-Key algorithms rely on two keys where:
  - it is computationally infeasible to find decryption key knowing only algorithm & encryption key
  - it is computationally easy to en/decrypt messages when the relevant (en/decrypt) key is known
  - either of the two related keys can be used for encryption, with the other used for decryption (for some algorithms)
- these are formidable requirements which only a few algorithms have satisfied

# Security of Public Key Schemes

- like private key schemes brute force **exhaustive search** attack is always theoretically possible
- but keys used are too large (>512bits)
- security relies on a **large enough** difference in difficulty between **easy** (en/decrypt) and **hard** (cryptanalyse) problems
- more generally the **hard** problem is known, but is made hard enough to be impractical to break
- requires the use of **very large numbers**
- hence is **slow** compared to private key schemes

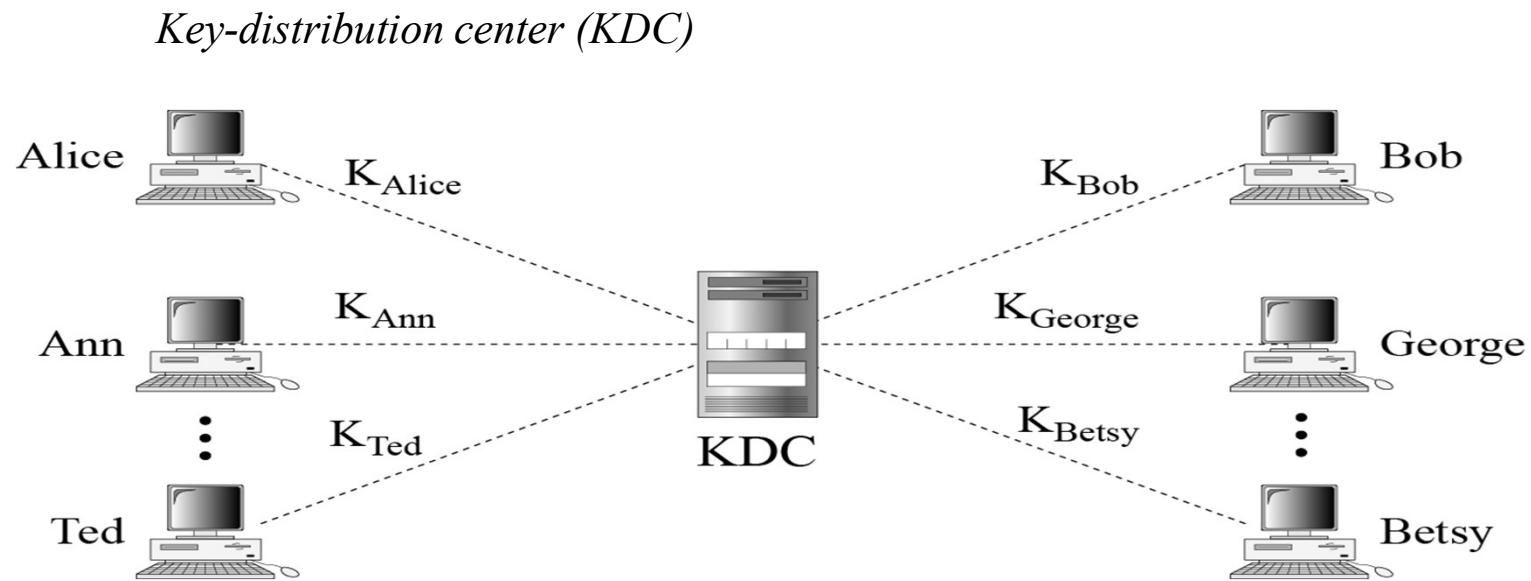
# Symmetric-key Distribution

Symmetric-key cryptography is more efficient than asymmetric-key cryptography for enciphering large messages. Symmetric-key cryptography, however, needs a shared secret key between two parties. The distribution of keys is another problem.

## Two Possible Ways:

- 1      Key-Distribution Center: KDC
- 2      Session Keys

# Key-Distribution Center: KDC



# Symmetric-key Agreement

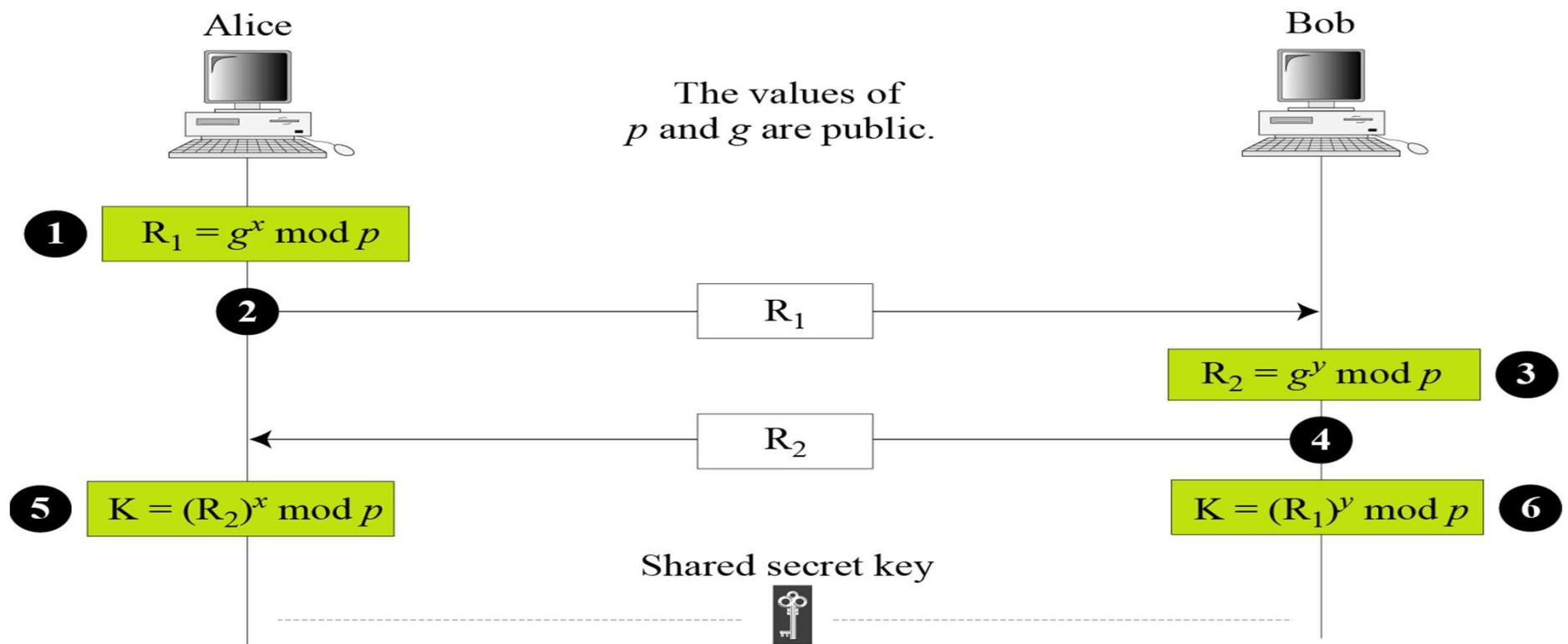
*Alice and Bob can create a session key between themselves without using a KDC. This method of session-key creation is referred to as the symmetric-key agreement.*

## Techniques for Key

- 1      Diffie-Hellman ~~Agreement~~ Key Agreement
- 2      Station-to-Station Key Agreement

# Diffie-Hellman Key Agreement

*Diffie-Hellman method*



*Note*

The symmetric (shared) key in the Diffie-Hellman method is  
 $K = g^{xy} \text{ mod } p.$

## ***Example***

Let us give a trivial example to make the procedure clear. Our example uses small numbers, but note that in a real situation, the numbers are very large. Assume that  $g = 7$  and  $p = 23$ . The steps are as follows:

1. Alice chooses  $x = 3$  and calculates  $R_1 = 7^3 \bmod 23 = 21$ .
2. Bob chooses  $y = 6$  and calculates  $R_2 = 7^6 \bmod 23 = 4$ .
3. Alice sends the number 21 to Bob.
4. Bob sends the number 4 to Alice.
5. Alice calculates the symmetric key  $K = 4^3 \bmod 23 = 18$ .
6. Bob calculates the symmetric key  $K = 21^6 \bmod 23 = 18$ .
7. The value of  $K$  is the same for both Alice and Bob;  
 $g^{xy} \bmod p = 7^{18} \bmod 35 = 18$ .

## ***Example***

Let us give a more realistic example. We used a program to create a random integer of 512 bits (the ideal is 1024 bits). The integer  $p$  is a 159-digit number. We also choose  $g$ ,  $x$ , and  $y$  as shown below:

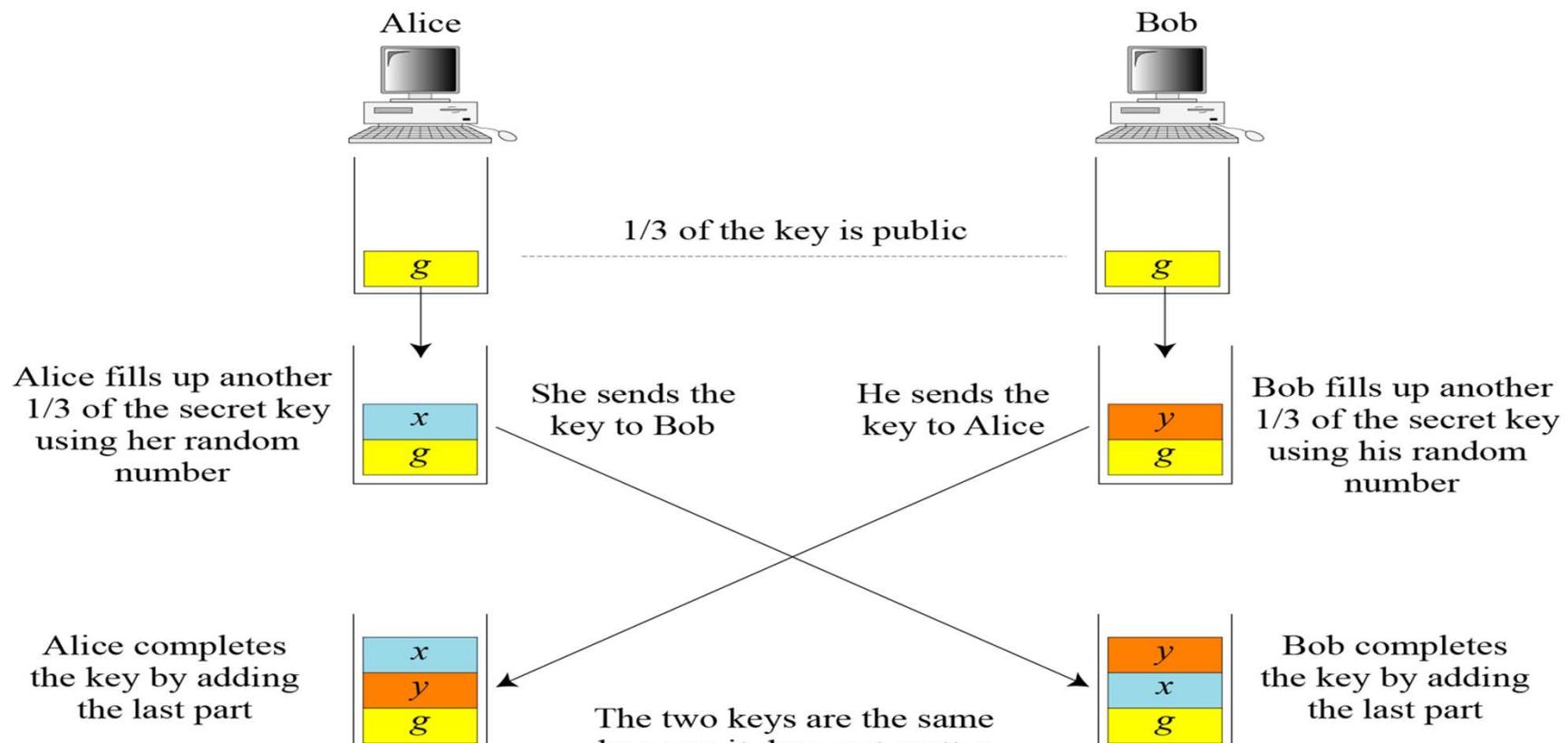
<b><i>p</i></b>	764624298563493572182493765955030507476338096726949748923573772860925 235666660755423637423309661180033338106194730130950414738700999178043 6548785807987581
<b><i>g</i></b>	2
<b><i>x</i></b>	557
<b><i>y</i></b>	273

*Continued*

*The following shows the values of  $R_1$ ,  $R_2$ , and  $K$ .*

<b>R<sub>1</sub></b>	844920284205665505216172947491035094143433698520012660862863631067673 619959280828586700802131859290945140217500319973312945836083821943065 966020157955354
<b>R<sub>2</sub></b>	435262838709200379470747114895581627636389116262115557975123379218566 310011435718208390040181876486841753831165342691630263421106721508589 6255201288594143
<b>K</b>	155638000664522290596225827523270765273218046944423678520320400146406 500887936651204257426776608327911017153038674561252213151610976584200 1204086433617740

## Diffie-Hellman idea



**Example:**

Step 1: Alice and Bob get public numbers  $P = 23$ ,  $G = 9$

Step 2: Alice selected a private key  $x = 4$  and Bob selected a private key  $y = 3$

Step 3: Alice and Bob compute public values

Alice:  $R_1 = ?$

Bob:  $R_2 = ?$

Step 4: Alice and Bob exchange public numbers

Step 5: Alice receives public key  $R_2$  and Bob receives public key  $R_1$

Step 6: Alice and Bob compute symmetric keys

Alice:  $k_a = ?$

Bob:  $k_b = ?$

Step 7: What is the value of shared secret key  $k??.$

**Example:**

Step 1: Alice and Bob get public numbers  $P = 23$ ,  $G = 9$

Step 2: Alice selected a private key  $x = 4$  and Bob selected a private key  $y = 3$

Step 3: Alice and Bob compute public values

$$\text{Alice: } R_1 = (9^4 \bmod 23) = (6561 \bmod 23) = 6$$

$$\text{Bob: } R_2 = (9^3 \bmod 23) = (729 \bmod 23) = 16$$

Step 4: Alice and Bob exchange public numbers

Step 5: Alice receives public key  $R_2 = 16$  and Bob receives public key  $R_1 = 6$

Step 6: Alice and Bob compute symmetric keys

$$\text{Alice: } k_a = R_2^a \bmod p = 65536 \bmod 23 = 9$$

$$\text{Bob: } k_b = R_1^b \bmod p = 216 \bmod 23 = 9$$

Step 7:  $K=9$  is the shared secret.

## RSA Cryptosystem

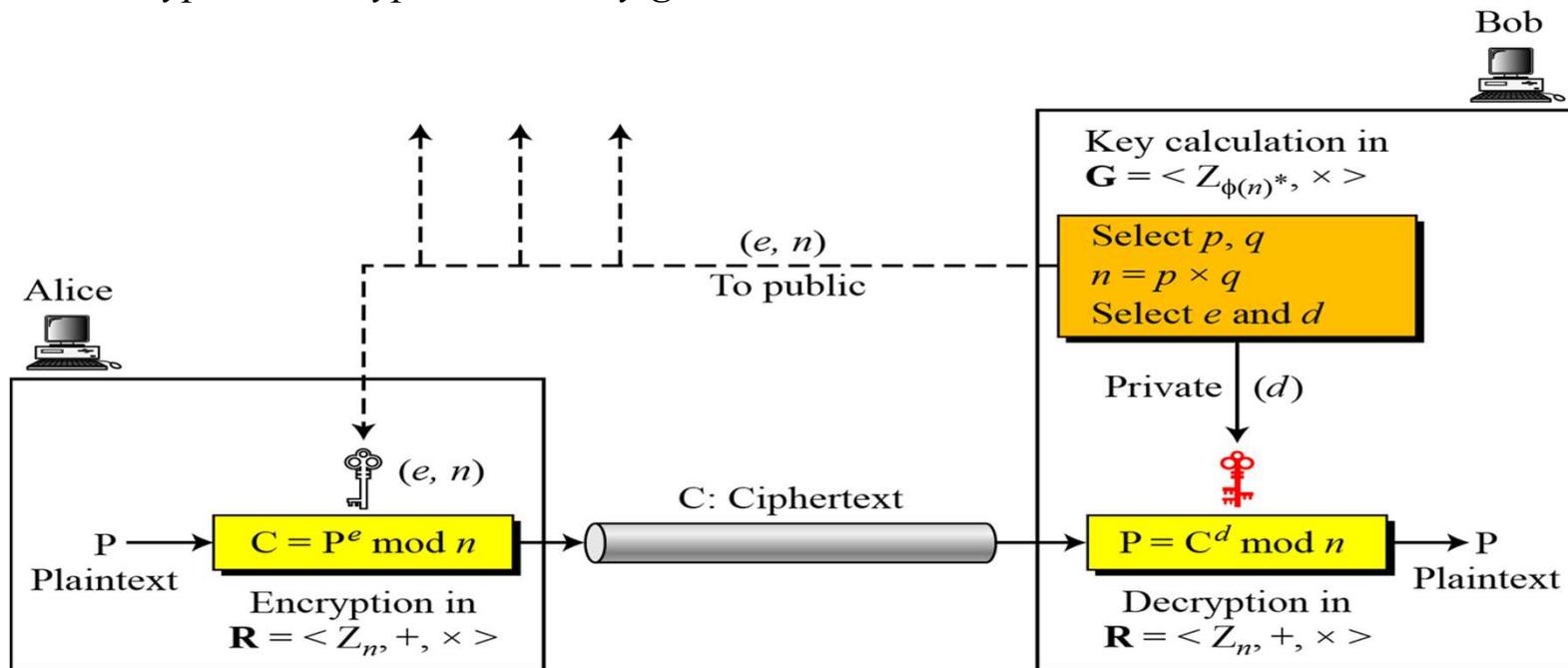
The most common public-key algorithm is the RSA cryptosystem, named for its inventors (Rivest, Shamir, and Adleman).

# RSA

- by Rivest, Shamir & Adleman of MIT in 1977
- best known & widely used public-key scheme
- based on exponentiation in a finite (Galois) field over integers modulo a prime
  - nb. exponentiation takes  $O((\log n)^3)$  operations (easy)
- uses large integers (eg. 1024 bits)
- security due to cost of factoring large numbers
  - nb. factorization takes  $O(e^{\log n \log \log n})$  operations (hard)

# Procedure

*Encryption, decryption, and key generation in RSA*



# RSA Key Generation

- users of RSA must:
  - determine two primes at random -  $p, q$
  - select either  $e$  or  $d$  and compute the other
- primes  $p, q$  must not be easily derived from modulus  $n=p \cdot q$ 
  - means must be sufficiently large
  - typically guess and use probabilistic test
- exponents  $e, d$  are inverses, so use Inverse algorithm to compute the other

# RSA Key Setup

- each user generates a public/private key pair by:
- selecting two large primes at random:  $p, q$
- computing their system modulus  $n=p \cdot q$ 
  - note  $\phi(n) = (p-1)(q-1)$
  - $\text{Gcd}(3,7) = ?$
- selecting at random the encryption key  $e$ 
  - where  $1 < e < \phi(n)$ ,  $\text{gcd}(e, \phi(n)) = 1$
- solve following equation to find decryption key  $d$ 
  - $e \cdot d \equiv 1 \pmod{\phi(n)}$  and  $0 \leq d \leq n$
- publish their public encryption key:  $PU=\{e,n\}$
- keep secret private decryption key:  $PR=\{d,n\}$

---

## **Algorithm 10.2 RSA Key Generation**

### **RSA\_Key\_Generation**

{

Select two large primes  $p$  and  $q$  such that  $p \neq q$ .

$n \leftarrow p \times q$

$\phi(n) \leftarrow (p - 1) \times (q - 1)$

Select  $e$  such that  $1 < e < \phi(n)$  and  $e$  is coprime to  $\phi(n)$

$d \leftarrow e^{-1} \bmod \phi(n)$  //  $d$  is inverse of  $e$  modulo  $\phi(n)$

Public\_key  $\leftarrow (e, n)$  // To be announced publicly

Private\_key  $\leftarrow d$  // To be kept secret

return Public\_key and Private\_key

}

# RSA En/decryption

- to encrypt a message  $M$  the sender:
  - obtains **public key** of recipient  $PU = \{e, n\}$
  - computes:  $C = M^e \text{ mod } n$ , where  $0 \leq M < n$
- to decrypt the ciphertext  $C$  the owner:
  - uses their private key  $PR = \{d, n\}$
  - computes:  $M = C^d \text{ mod } n$
- note that the message  $M$  must be smaller than the modulus  $n$  (block if needed)

## *Encryption*

### **Algorithm 10.3 RSA encryption**

```
RSA_Encryption (P, e, n)          // P is the plaintext in  $Z_n$  and  $P < n$ 
{
    C  $\leftarrow$  Fast_Exponentiation (P, e, n)      // Calculation of  $(P^e \bmod n)$ 
    return C
}
```

**In RSA,  $p$  and  $q$  must be at least 512 bits;  $n$  must be at least 1024 bits.**



## *Decryption*

### **Algorithm 10.4 RSA decryption**

```
RSA_Decryption (C, d, n)          //C is the ciphertext in  $Z_n$ 
{
    P  $\leftarrow$  Fast_Exponentiation (C, d, n)      // Calculation of  $(C^d \bmod n)$ 
    return P
}
```

# Why RSA Works

➤ because of Euler's Theorem:

- $a^{\phi(n)} \mod n = 1$  where  $\gcd(a, n) = 1$

➤ in RSA have:

- $n = p \cdot q$
- $\phi(n) = (p-1)(q-1)$
- carefully chose  $e$  &  $d$  to be inverses mod  $\phi(n)$
- hence  $e \cdot d = 1 + k \cdot \phi(n)$  for some  $k$

➤ hence :

$$\begin{aligned} C^d &= M^e \cdot d = M^{1+k \cdot \phi(n)} = M^1 \cdot (M^{\phi(n)})^k \\ &= M^1 \cdot (1)^k = M^1 = M \mod n \end{aligned}$$

# RSA Example - Key Setup

1. Select primes:  $p=17$  &  $q=11$
2. Calculate  $n = pq = 17 \times 11 = 187$
3. Calculate  $\phi(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. Select  $e$ :  $\gcd(e, 160) = 1$ ; choose  $e=7$
5. Determine  $d$ :  $de \equiv 1 \pmod{160}$  and  $d < 160$  Value is  $d=23$  since  $23 \times 7 = 161 = 10 \times 160 + 1$
6. Publish public key PU={7, 187}
7. Keep secret private key PR={23, 187}

# RSA Example - En/Decryption

- sample RSA encryption/decryption is:
- given message  $M = 88$  (nb.  $88 < 187$ )
- encryption:

$$C = 88^7 \bmod 187 = 11$$

- decryption:

$$M = 11^{23} \bmod 187 = 88$$



## Example

---

Bob chooses 7 and 11 as  $p$  and  $q$  and calculates  $n = 77$ .

The value of  $\phi(n) = (7 - 1)(11 - 1)$  or 60. Now he chooses two exponents,  $e$  and  $d$ , from  $Z_{60}^*$ . If he chooses  $e$  to be 13, then  $d$  is 37. Note that  $e \times d \bmod 60 = 1$  (they are inverses of each other). Now imagine that Alice wants to send the plaintext 5 to Bob. She uses the public exponent 13 to encrypt 5.

$$C = M^e \bmod n$$

## Example

Bob chooses 7 and 11 as  $p$  and  $q$  and calculates  $n = 77$ . The value of  $\phi(n) = (7 - 1)(11 - 1)$  or 60. Now he chooses two exponents,  $e$  and  $d$ , from  $Z_{60}^*$ . If he chooses  $e$  to be 13, then  $d$  is 37. Note that  $e \times d \bmod 60 = 1$  (they are inverses of each other). Now imagine that Alice wants to send the plaintext 5 to Bob. She uses the public exponent 13 to encrypt 5.

Plaintext: 5

$$C = 5^{13} \bmod 77$$

Ciphertext: 26

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

Ciphertext: 26

$$P = 26^{37} \bmod 77$$

Plaintext: 5

---

## Example

Now assume that another person, John, wants to send a message to Bob. John can use the same public key announced by Bob (probably on his website), 13; John's plaintext is 63. John calculates the following:

Plaintext: 63

$$C = 63^{13} = 28 \text{ mod } 77$$

Ciphertext: 28

Bob receives the ciphertext 28 and uses his private key 37 to decipher the ciphertext:

Ciphertext: 28

$$P = 28^{37} = 63 \text{ mod } 77$$

Plaintext: 63

---

## Example

Jennifer creates a pair of keys for herself.

She chooses  $p = 397$  and  $q = 401$ .

She calculates  $n = ?$ .

She then calculates  $\phi(n) = ?$ .

She then chooses  $e = 343$  and  $d = 12007$ .

Show how Ted can send a message to Jennifer if he knows  $e$  and  $n$ .

Suppose Ted wants to send the message “NO” to Jennifer.

The plaintext is ?.

---

## Example

Jennifer creates a pair of keys for herself.

She chooses  $p = 397$  and  $q = 401$ .

She calculates  $n = ?$ .

She then calculates  $\phi(n) = ?$ .

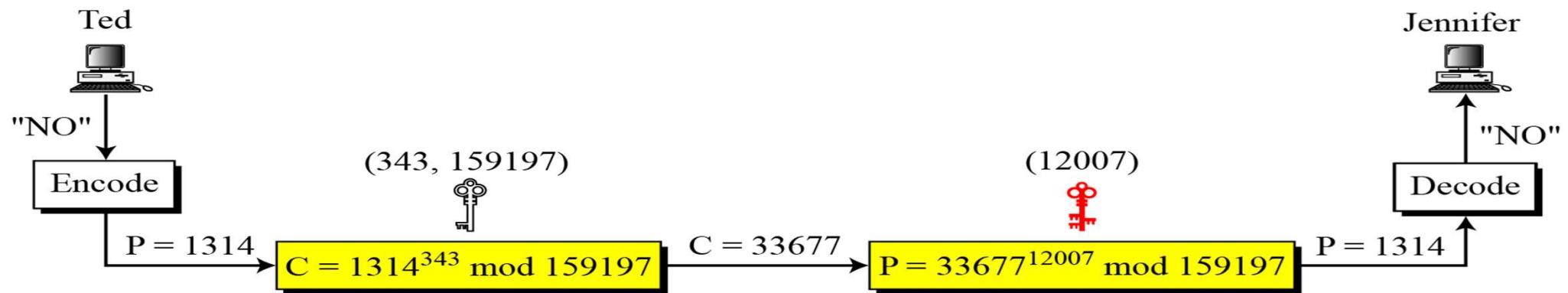
She then chooses  $e = 343$  and  $d = 12007$ .

Show how Ted can send a message to Jennifer if he knows  $e$  and  $n$ .

Suppose Ted wants to send the message “NO” to Jennifer.

He changes each character to a number (from 00 to 25), with each character coded as two digits. He then concatenates the two coded characters and gets a four-digit number. The plaintext is ?.

*Encryption and decryption Example*



# Exponentiation

- can use the Square and Multiply Algorithm
- a fast, efficient algorithm for exponentiation
- concept is based on repeatedly squaring base
- and multiplying in the ones that are needed to compute the result
- look at binary representation of exponent
- only takes  $O(\log_2 n)$  multiples for number n
  - eg.  $7^5 = 7^4 \cdot 7^1 = 3 \cdot 7 = 10 \bmod 11$
  - eg.  $3^{129} = 3^{128} \cdot 3^1 = 5 \cdot 3 = 4 \bmod 11$

# RSA Security

- possible approaches to attacking RSA are:
  - brute force key search - infeasible given size of numbers
  - mathematical attacks - based on difficulty of computing  $\phi(n)$ , by factoring modulus n
  - timing attacks - on running of decryption
  - chosen ciphertext attacks - given properties of RSA

# Factoring Problem

➤ mathematical approach takes 3 forms:

- factor  $n=p \cdot q$ , hence compute  $\phi(n)$  and then  $d$
- determine  $\phi(n)$  directly and compute  $d$
- find  $d$  directly

➤ currently believe all equivalent to factoring

- have seen slow improvements over the years
  - as of May-05 best is 200 decimal digits (663) bit with LS
- biggest improvement comes from improved algorithm
  - cf QS to GHFS to LS
- currently assume 1024-2048 bit RSA is secure
  - ensure  $p, q$  of similar size and matching other constraints