

**M.S. Ramaiah Institute of Technology  
(Autonomous Institute, Affiliated to VTU)**

**Department of Computer Science and Engineering**

**Course Name: Cryptography and Network Security**

**Course Code - CSE643**

**Credits - 3:0:0**

**UNIT -2**

**Term: March 2022 – July 2022**

---

**Prepared by: Dr. Sangeetha. V  
Assistant Professor**

# Unit II (Text1)

---

## **Traditional Symmetric-Key Ciphers: (Chapter 3)**

- Introduction
- Substitution Ciphers
- Trans positional Ciphers
- Stream and Block Ciphers

## **Data Encryption Standard (DES): (Chapter 6)**

- Introduction
- DES Structure
- DES Analysis
- Security of DES

## **Advanced Encryption Standard(AES): (Chapter 7)**

- Introduction
- Transformations
- Key Expansion
- The AES Ciphers
- Examples
- Analysis of AES.

# Introduction -Symmetric Key ciphers

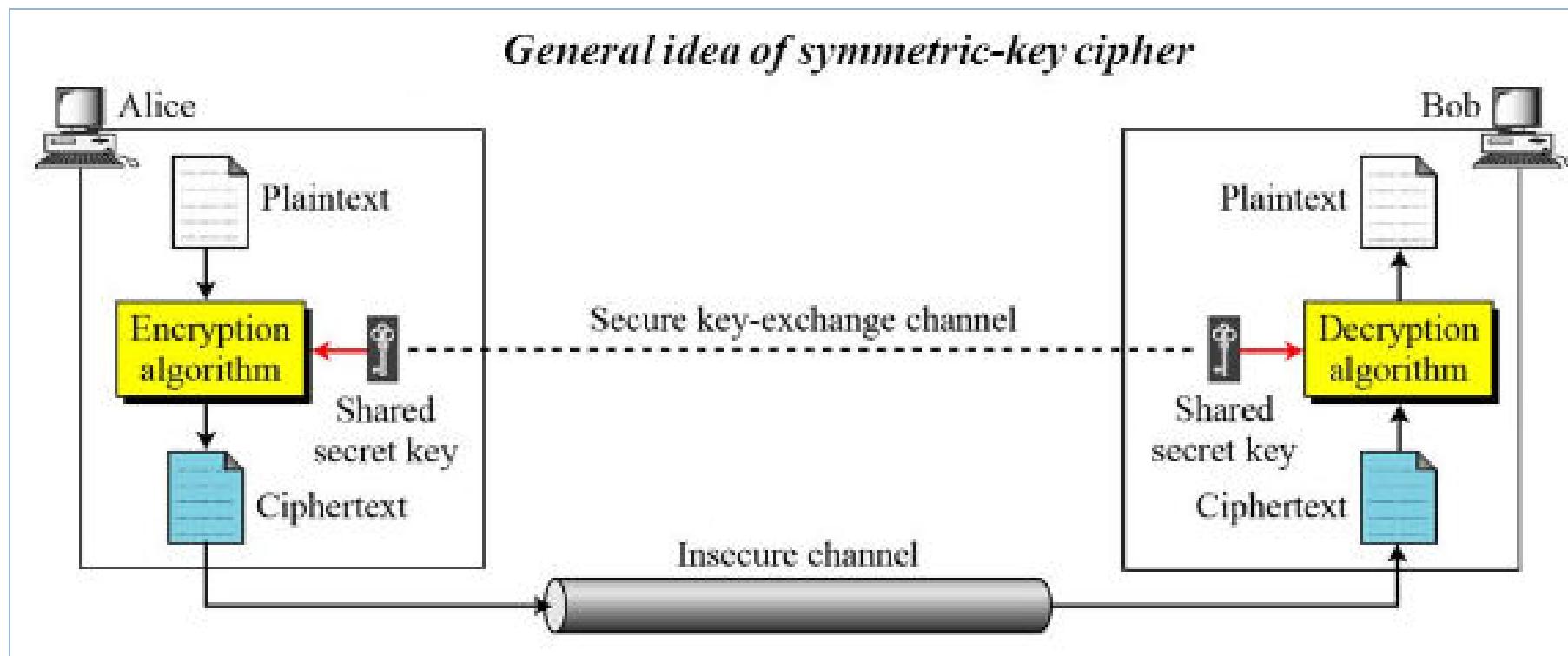
---

Traditional Symmetric-key ciphers are not used today, but we study for several reason:

1. They are simpler than modern ciphers and easier to understand
2. They show basic foundation of cryptography and encipherment.
3. They provide rationale for using modern ciphers, traditional ciphers can be easily attacked using computer

# Introduction -Symmetric Key ciphers

Figure shows the general idea behind a symmetric key cipher



# Introduction -Symmetric Key ciphers

---

The original message from Alice to Bob is called **plaintext**;

The message that is sent through the channel is called the **ciphertext**.

To create the ciphertext from the plaintext, Alice uses an **encryption algorithm** and a **shared secret key**.

To create the plaintext from ciphertext, Bob uses a **decryption algorithm** and the same secret key.

# Introduction -Symmetric Key ciphers

---

We refer to encryption and decryption algorithms as  
**Ciphers**

A **Key** is a set of values(Numbers) that the cipher operates on

# Introduction -Symmetric Key ciphers

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

Encryption:  $C = E_k(P)$

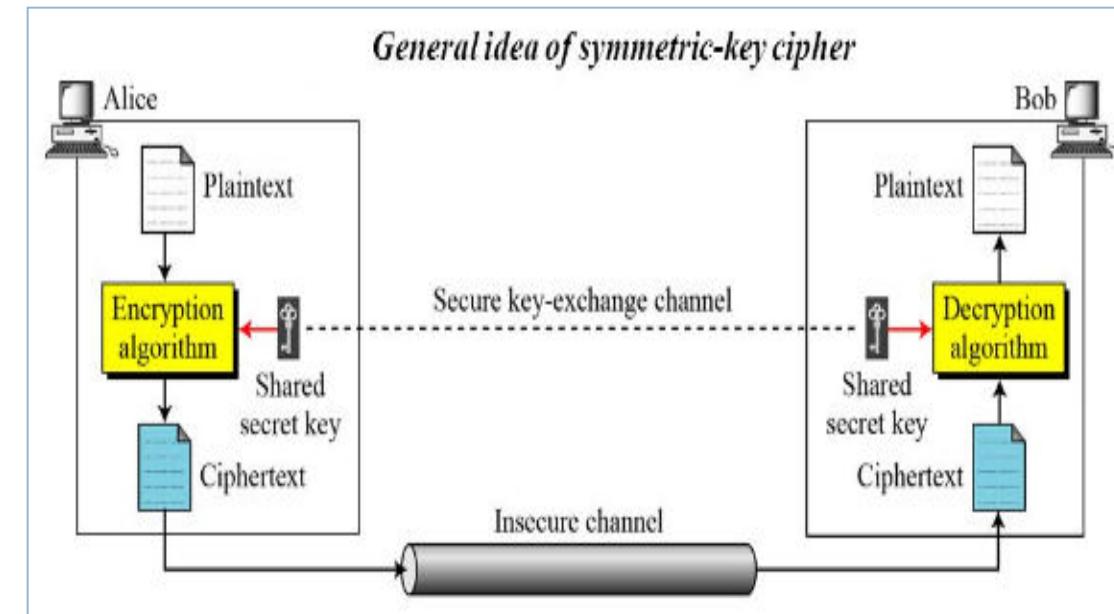
Decryption:  $P = D_k(C)$

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

**We assume that Bob creates  $P_1$ ; we prove that  $P_1 = P$ :**

**Alice:**  $C = E_k(P)$

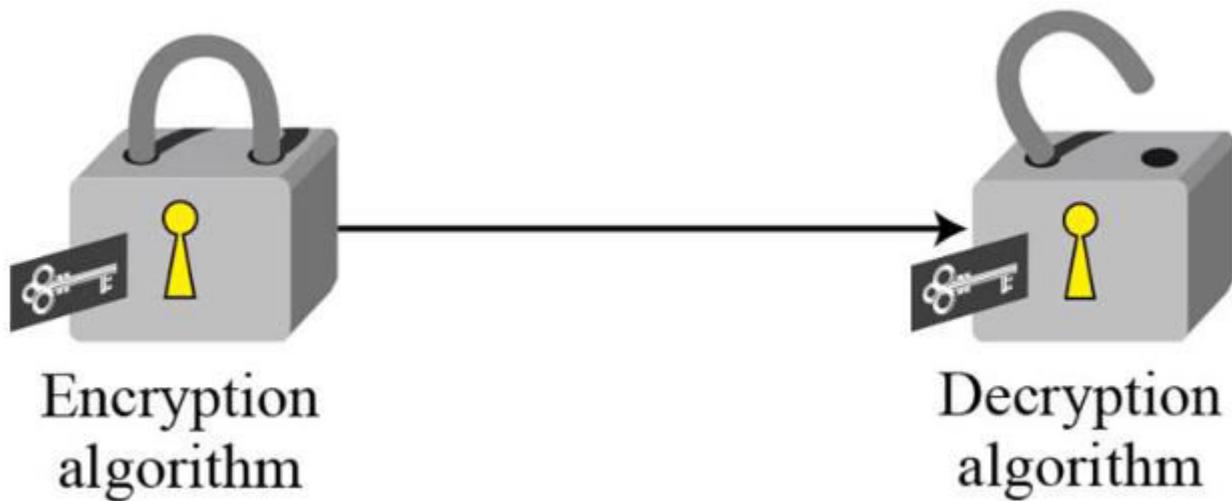
**Bob:**  $P_1 = D_k(C) = D_k(E_k(P)) = P$



# Introduction -Symmetric Key ciphers

Figure shows Locking and unlocking with the same key.

Alice and Bob meet once and exchange the key personally.



# Introduction -Symmetric Key ciphers

---

## Kerckhoff's Principle

- According to Kerckhoff's principle, it is better to make encryption and decryption public, but keep the shared key secret.
- Based on Kerckhoff's principle, one should always assume that the **adversary, Eve**, knows the encryption/decryption algorithm.
- The resistance of the cipher to attack must be based only on the secrecy of the key.

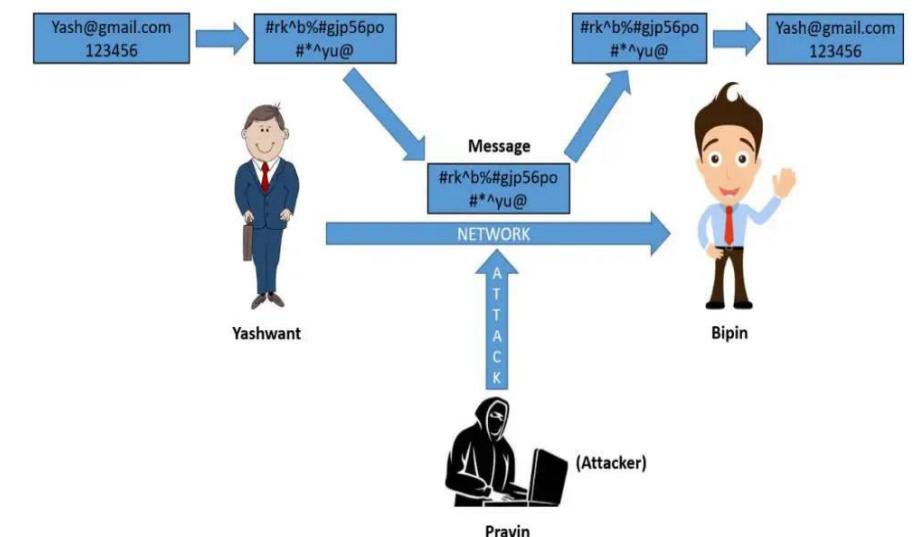
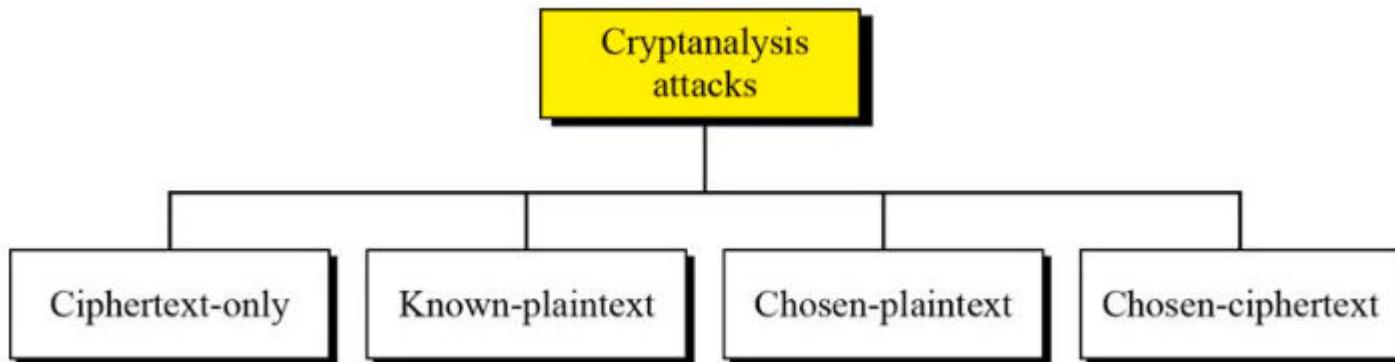
# Introduction -Symmetric Key ciphers

## Cryptanalysis

Cryptography is the science and art of creating secret codes.

Cryptanalysis is the science and art of breaking those codes.

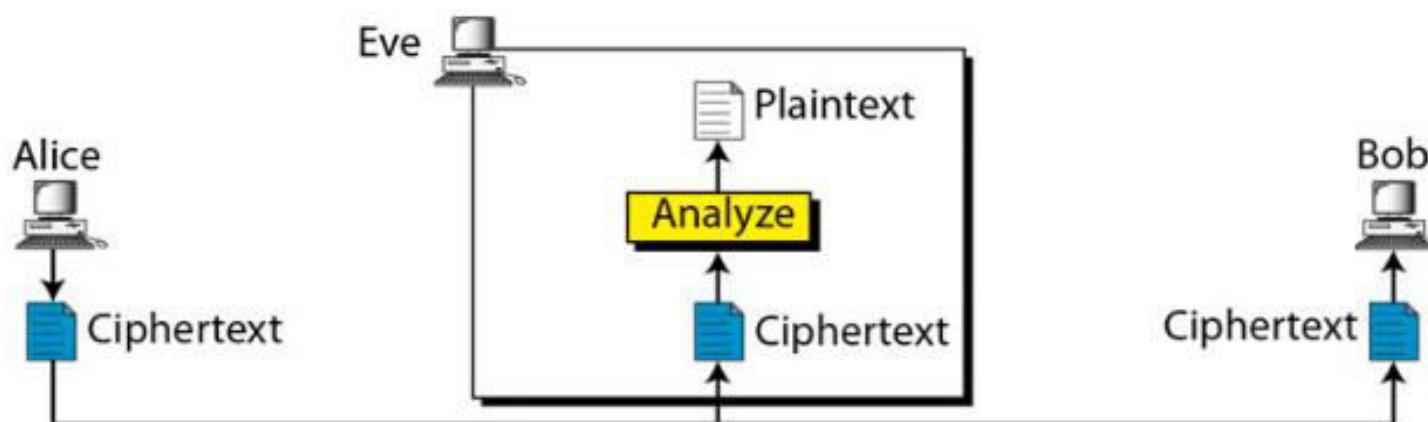
Figure shows Cryptanalysis attack froms



# Introduction -Symmetric Key ciphers

## Cryptanalysis - Ciphertext-only attack

- Eve has access to only some ciphertext, then finds the key and plaintext.
- Assume eve knows the encryption algorithm



# Introduction -Symmetric Key ciphers

---

## Cryptanalysis - Ciphertext-only attack

Various methods can be used in Ciphertext-only attack

1. Brute-Force attack: exhaustive key search attack
2. Statistical attack: benefit from inherent characteristics of the plaintext language. E.g. E is the most frequently used letter.
3. Pattern attack: discover pattern in ciphertext.

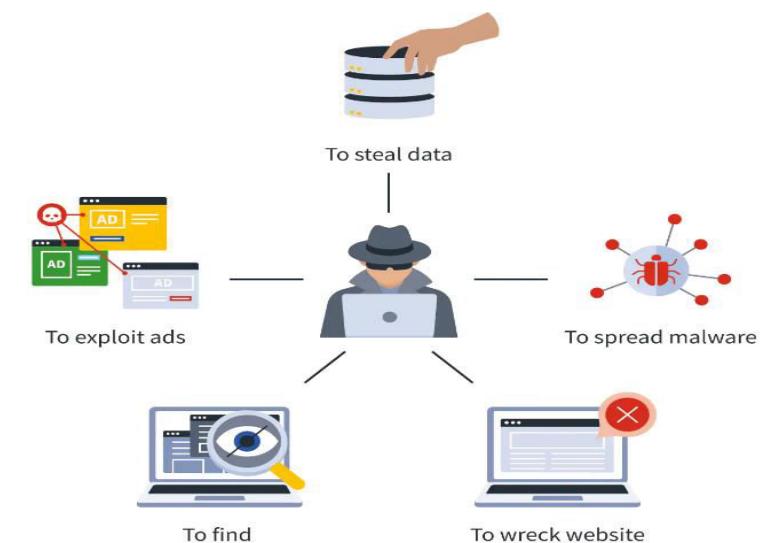
# Introduction -Symmetric Key ciphers

## 1. Brute-Force attack:

If the key is 8 bits long, then the number of possible keys is  $2^8 = 256$  keys

Often target popular platforms where many users store data.

Email domains, online tax services, or food apps could be likely targets.



# Introduction -Symmetric Key ciphers

---

## 2. Statistical attack:

It is the study of the frequency of letters or groups of letters in a ciphertext.

Benefit from inherent characteristics of the plaintext language. E.g. E is the most frequently used letter.

Statistical attacks target vulnerabilities in the **operating systems** or **hardware hosting the functional cryptography tool**.

# Introduction -Symmetric Key ciphers

---

## 2. Statistical attack:

For example, Database containing employee details may be used by others to calculate the average salary of employees based on particular criteria.

If a user discovered a criterion which only holds for one employee, and uses this information to find the average salary of all employees, then the employee's salary could be easily discovered.

# Introduction -Symmetric Key ciphers

---

## 3. Pattern attack:

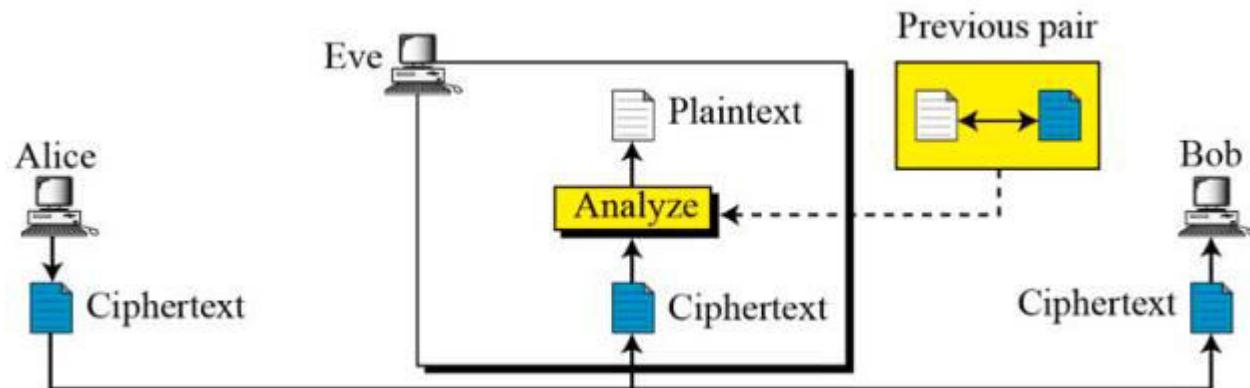
Some ciphers may hide the characteristics of the language, but create some patterns in the ciphertext

Cryptanalyst will use pattern to break the cipher

# Introduction -Symmetric Key ciphers

## Cryptanalysis -Known-Plaintext Attack

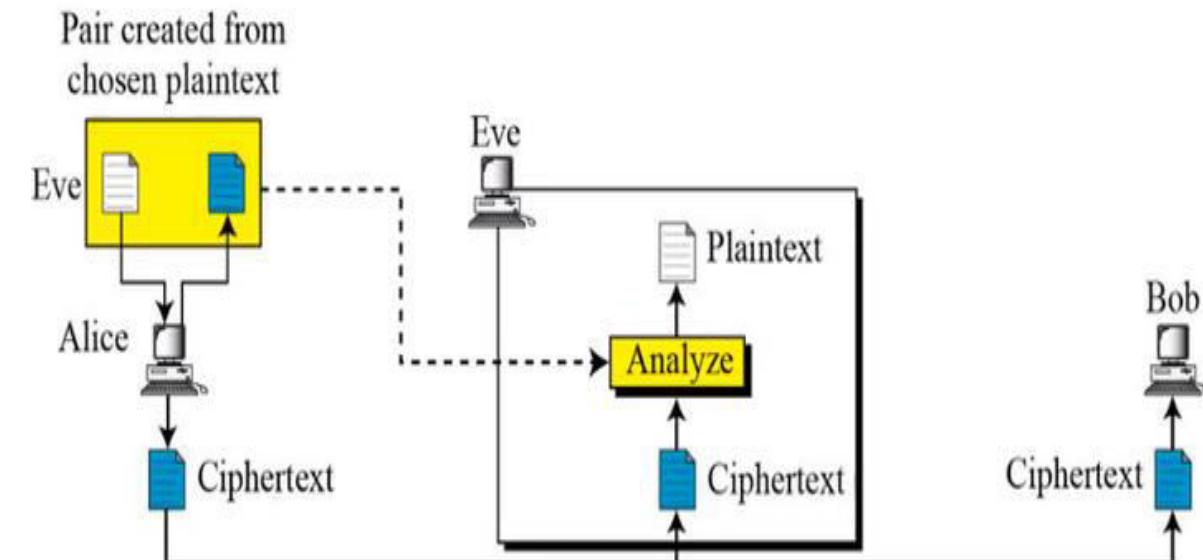
- Eve has access to some plaintext/ciphertext pairs in addition to the intercepted ciphertext that he/she wants to break.
- Plaintext/Ciphertext pairs have been collected earlier.



# Introduction -Symmetric Key ciphers

## Cryptanalysis -Chosen-Plaintext Attack

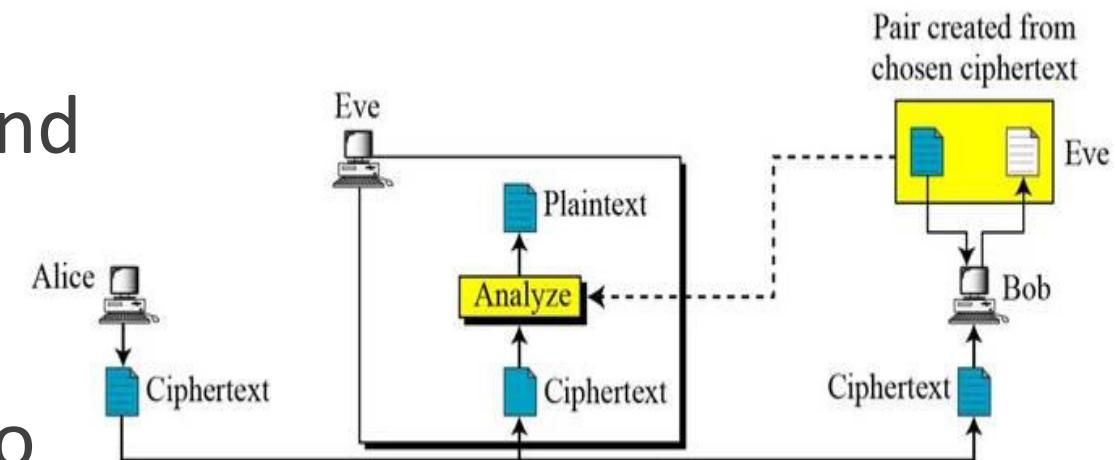
- Eve has access to Alice computer
- Cryptanalyst chose some plaintext and intercept the created plaintext.
- Observing the Plaintext/ciphertext gives Eve a strong foothold into the inner workings of the algorithm and secret key.



# Introduction -Symmetric Key ciphers

## Cryptanalysis -Chosen-Ciphertext Attack

- Eve has access to Bob computer
- Cryptanalyst chose some ciphertext and decrypts to form the pair plaintext/ciphertext.
- Cryptanalyst is not necessarily trying to find the plaintext, but rather they are trying to **decipher the algorithm and secret key** used to encrypt the plaintext. .



# Introduction -Symmetric Key ciphers

Type of Attack	Known to cryptanalyst
Ciphertext Only	<ul style="list-style-type: none"> <li>★ Encryption Algorithm</li> <li>★ Ciphertext</li> </ul>
Known Plaintext	<ul style="list-style-type: none"> <li>★ Encryption Algorithm</li> <li>★ Ciphertext</li> <li>★ One or more PT-CT pairs formed with secret key</li> </ul>
Chosen Plaintext	<ul style="list-style-type: none"> <li>★ Encryption Algorithm</li> <li>★ Ciphertext</li> <li>★ PT message chosen by cryptanalyst, together with its CT generated with the secret key</li> </ul>
Chosen Ciphertext	<ul style="list-style-type: none"> <li>★ Encryption Algorithm</li> <li>★ Ciphertext</li> <li>★ CT chosen by cryptanalyst, together with its corresponding decrypted PT generated with the secret key</li> </ul>

# Substitution Ciphers

---

A substitution cipher replaces one symbol with another.

If the symbols in the plain text are alphabetic characters,  
replaces one character with another.

Substitution ciphers can be categorized as either

- a. Monoalphabetic ciphers
- b. Polyalphabetic ciphers

# Substitution Ciphers

---

## **Monoalphabetic ciphers**

- a. Additive cipher(Shift cipher/Ceasar cipher)
- b. Multiplicative ciphers
- c. Affine cipher

# Substitution Ciphers

---

## Monoalphabetic ciphers

A Character in the plaint text is changed to the same character in the ciphertext regardless of its position in the plaintext

The relationship between a symbol in the plaintext to a symbol in the ciphertext is always one-to-one.

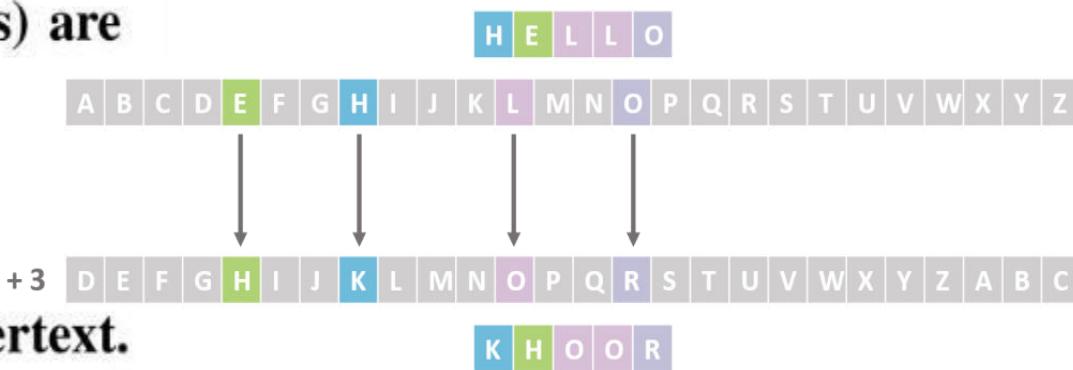
# Substitution Ciphers

## Monoalphabetic ciphers

The following shows a plaintext and its corresponding ciphertext. The cipher is probably monoalphabetic because both *l*'s (els) are encrypted as *O*'s.

**Plaintext:** hello

**Ciphertext:** KHOOR



The following shows a plaintext and its corresponding ciphertext. The cipher is not monoalphabetic because each *l* (el) is encrypted by a different character.

**Plaintext:** hello

**Ciphertext:** JKQNZ

# Substitution Ciphers

## Monoalphabetic ciphers – Additive cipher

The simplest monoalphabetic cipher is the additive cipher.

This cipher is sometimes called a shift cipher/Caesar cipher, but the term additive cipher better reveals its mathematical nature.

To apply mathematical operations on PT/CT, assign numerical values to each letter

Each character is assigned an integer in  $Z_{26}$

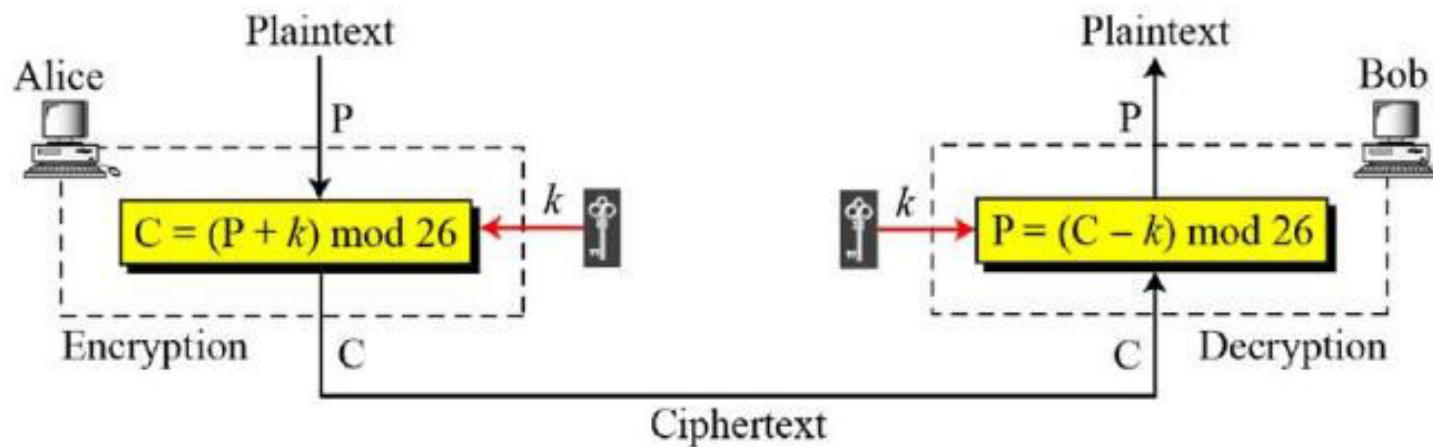
*Plaintext and ciphertext in  $Z_{26}$*

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

# Substitution Ciphers

## Monoalphabetic ciphers – Additive 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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25



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

# Substitution Ciphers

## Monoalphabetic ciphers – Additive 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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

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

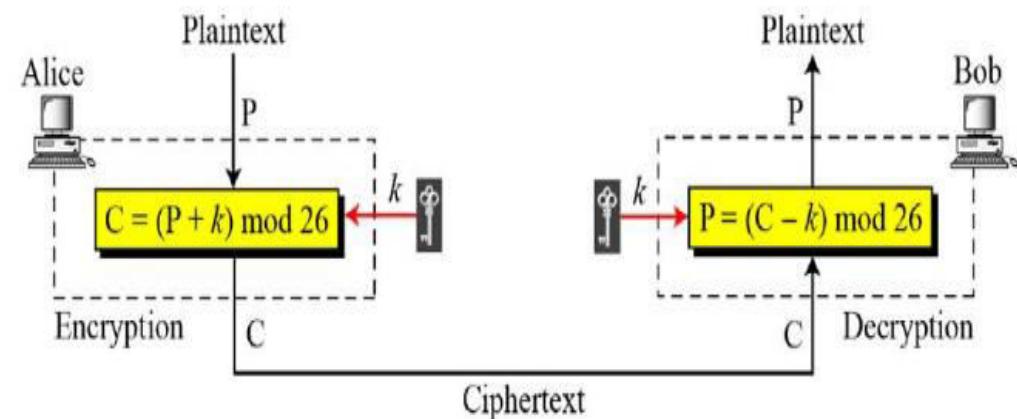
### Solution

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

Plaintext: h → 07  
 Plaintext: e → 04  
 Plaintext: l → 11  
 Plaintext: l → 11  
 Plaintext: o → 14

Encryption:  $(07 + 15) \bmod 26$   
 Encryption:  $(04 + 15) \bmod 26$   
 Encryption:  $(11 + 15) \bmod 26$   
 Encryption:  $(11 + 15) \bmod 26$   
 Encryption:  $(14 + 15) \bmod 26$

Ciphertext: 22 → W  
 Ciphertext: 19 → T  
 Ciphertext: 00 → A  
 Ciphertext: 00 → A  
 Ciphertext: 03 → D



# Substitution Ciphers

## Monoalphabetic ciphers – Additive 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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Use the additive cipher with key = 15 to decrypt the message “WTAAD”.

### Solution

We apply the decryption algorithm to the plaintext character by character:

Ciphertext: W → 22

Decryption:  $(22 - 15) \bmod 26$

Plaintext: 07 → h

Ciphertext: T → 19

Decryption:  $(19 - 15) \bmod 26$

Plaintext: 04 → e

Ciphertext: A → 00

Decryption:  $(00 - 15) \bmod 26$

Plaintext: 11 → l

Ciphertext: A → 00

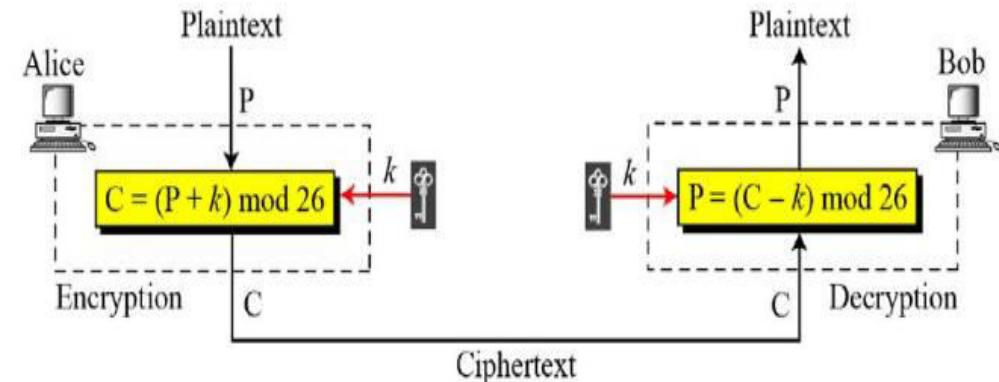
Decryption:  $(00 - 15) \bmod 26$

Plaintext: 11 → l

Ciphertext: D → 03

Decryption:  $(03 - 15) \bmod 26$

Plaintext: 14 → o

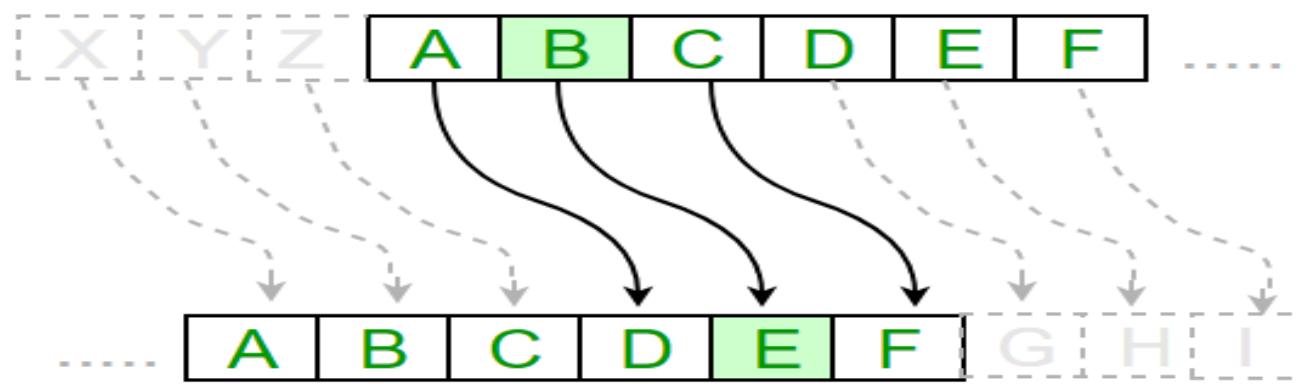


# Substitution Ciphers

## Monoalphabetic ciphers – Additive cipher

### Shift Cipher

A **shift cipher** involves replacing each letter in the message by a letter that is some fixed number of positions further along in the alphabet.



# Substitution Ciphers

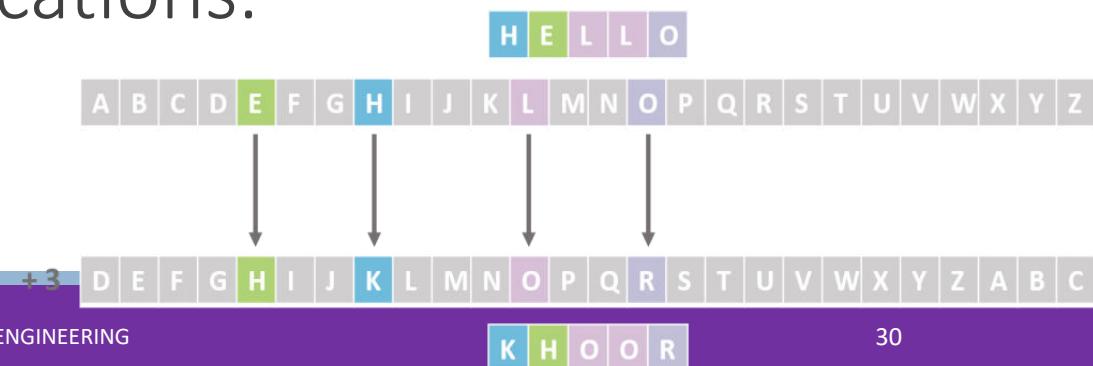
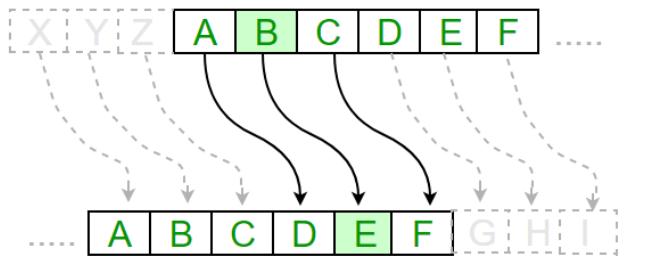
## Monoalphabetic ciphers – Additive cipher

### Caesar Cipher

Caesar used an additive cipher to communicate with his officers.

For this reason, additive ciphers are sometimes referred to as the Caesar cipher.

Caesar used a key of 3 for his communications.

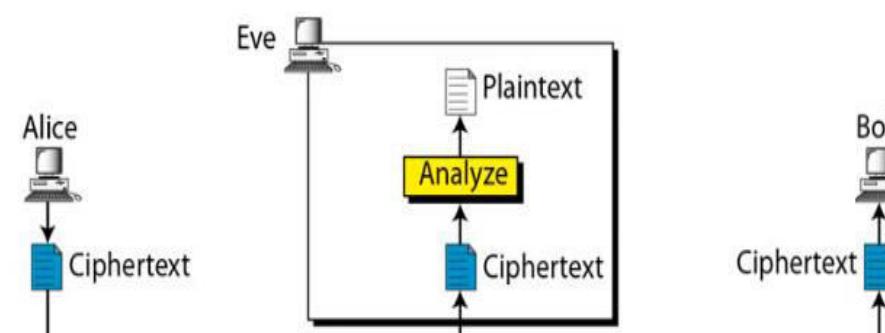
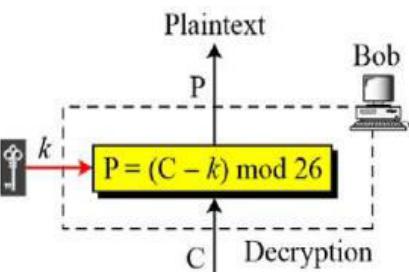
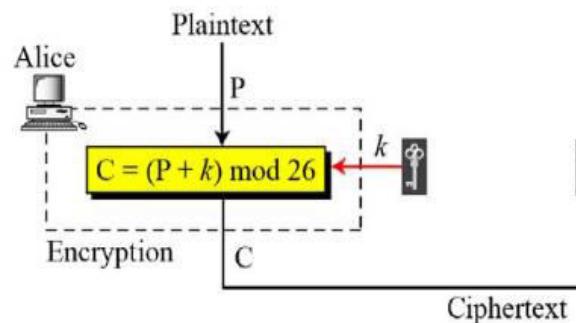


# Substitution Ciphers

## Monoalphabetic ciphers – Additive cipher

Cryptanalysis -Vulnerable to cipher-text only attack (Brute force attack)

**Eve has intercepted the ciphertext “UVACLYFZLJBYL”. Show how she can use a brute-force attack to break the 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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

**Eve has intercepted the ciphertext “UVACLYFZLJBYL”. Show how she can use a brute-force attack to break the cipher.**

### Solution

**Eve tries keys from 1 to 7. With a key of 7, the plaintext is “not very secure”, which makes sense.**

**Ciphertext: UVACLYFZLJBYL**

- K = 1** → **Plaintext:** tuzbkxeykiaxk
- K = 2** → **Plaintext:** styajwdxjhzwj
- K = 3** → **Plaintext:** rsxzivcwigyvi
- K = 4** → **Plaintext:** qrwyhubvhfxuh
- K = 5** → **Plaintext:** pqvxgtaugewtg
- K = 6** → **Plaintext:** opuwfsztfdvsf
- K = 7** → **Plaintext:** notverysecure

# Substitution Ciphers

## Monoalphabetic ciphers – Additive cipher

### Cryptanalysis

- Vulnerable to cipher-text only attack (Statistical attack)

*Frequency of characters in English*

Letter	Frequency	Letter	Frequency	Letter	Frequency	Letter	Frequency
E	12.7	H	6.1	W	2.3	K	0.08
T	9.1	R	6.0	F	2.2	J	0.02
A	8.2	D	4.3	G	2.0	Q	0.01
O	7.5	L	4.0	Y	2.0	X	0.01
I	7.0	C	2.8	P	1.9	Z	0.01
N	6.7	U	2.8	B	1.5		
S	6.3	M	2.4	V	1.0		

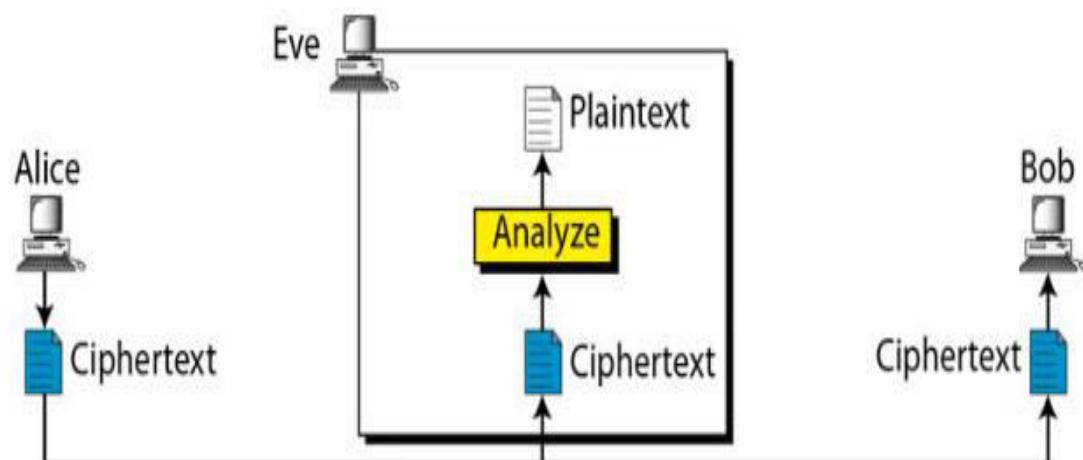
*Frequency of diagrams and trigrams*

Digram	TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI, OF
Trigram	THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR, DTH

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

**Eve has intercepted the following ciphertext. Using a statistical attack, find the plaintext.**

XLILSYWIMWRSAJSVWEPIJSVJSYVQMPPMSRHSPPEVWMXMWASVX-LQSVILY-  
VVCFIJSVIXLIWIPPIVVIGIMZIWQSVISJJIVW



**Eve tabulate the frequency of letter in this ciphertext**

I = 14 occurrences

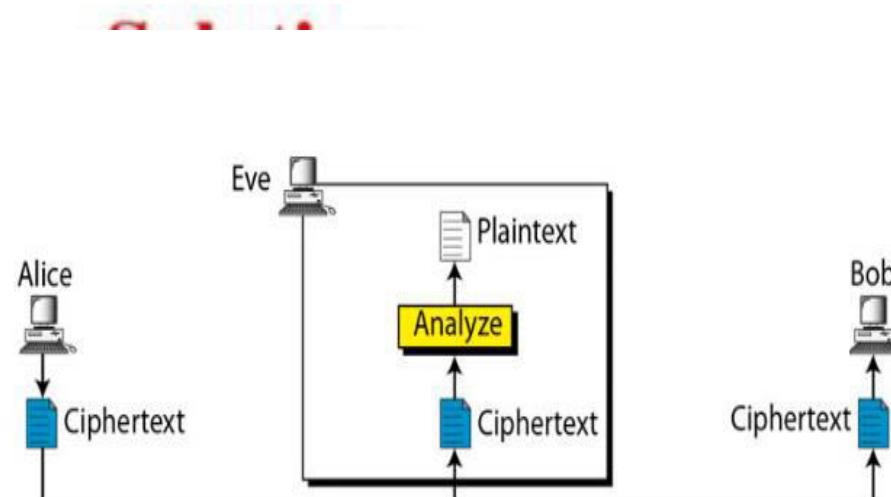
V= 13 occurrences

S =12 and so on

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

**Eve has intercepted the following ciphertext. Using a statistical attack, find the plaintext.**

XLILSYWIMWRSAJSVWEPIJSVJSYVQMPPMSRHSPPEVWMXMWASVX-LQSVILY-  
VVCFIJSVIXLIWIPPIVVIGIMZIWQSVISJJIVW



Letter	Frequency	Letter	Frequency	Letter	Frequency	Letter	Frequency
E	12.7	H	6.1	W	2.3	K	0.08
T	9.1	R	6.0	F	2.2	J	0.02
A	8.2	D	4.3	G	2.0	Q	0.01
O	7.5	L	4.0	Y	2.0	X	0.01
I	7.0	C	2.8	P	1.9	Z	0.01
N	6.7	U	2.8	B	1.5		
S	6.3	M	2.4	V	1.0		

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

XLILSYWIMWRSAJSVWEPIJSVJSYVQMPPMSRHSPEVWMXMWASVX-LQSVILY-  
 VVCFIJSVIXLIWIPPIVVIGIMZIWQSVISJJIVW

$$x \rightarrow 23 - 4 \bmod 26 \rightarrow 19 \bmod 26 = 19 \rightarrow T$$

$$L \rightarrow 11 - 4 \bmod 26 \rightarrow 7 \bmod 26 = 7 \rightarrow H$$

$$I \rightarrow 8 - 4 \bmod 26 \rightarrow 4 \bmod 26 = 4 \rightarrow E$$

# Substitution Ciphers

**Eve has intercepted the following ciphertext. Using a statistical attack, find the plaintext.**

XLILSYWIMWRSAJSVWEPIJSVJSYVQMPPMSRHSPEVWMXMWASVX-LQSVILY-  
VVCFIJSVIXLIWIPPIVIGIMZIWQSVISJJIVW

## Solution

**When Eve tabulates the frequency of letters in this ciphertext, she gets: I =14, V =13, S =12, and so on. The most common character is I with 14 occurrences. This means key = 4.**

the house is now for sale for four million dollars it is worth more hurry before the seller receives more offers

# Substitution Ciphers

---

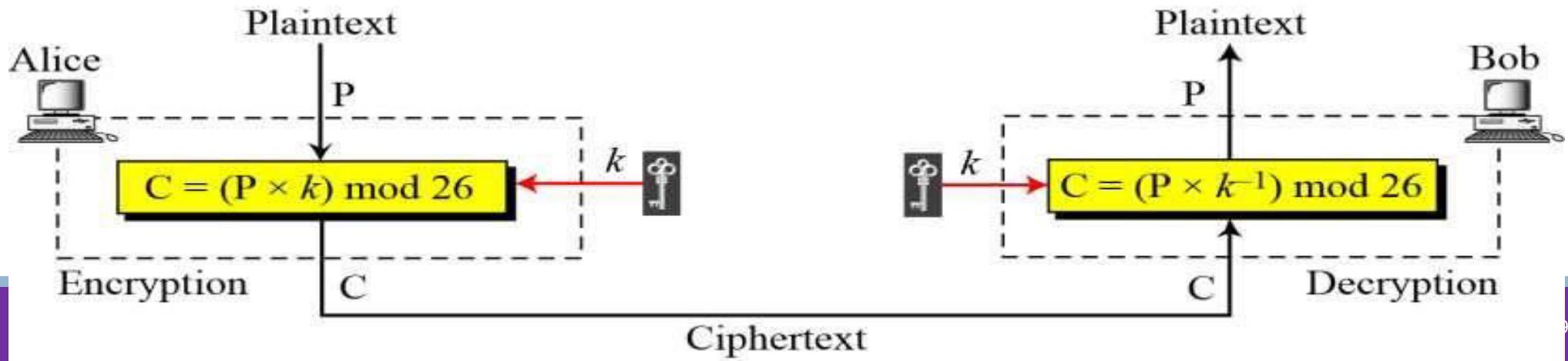
## **Monoalphabetic ciphers**

- a. Additive cipher(Shift cipher/Ceasar cipher)
- b. Multiplicative ciphers
- c. Affine cipher

# Substitution Ciphers

## Monoalphabetic ciphers - Multiplicative ciphers

- The encryption algorithm specifies **multiplication** of PT with key
- The decryption algorithm specifies division of CT by key
- Since operations are in  $Z_{26}$  Decryption here means multiplying PT with **multiplicative inverse of key**



# Substitution Ciphers

---

In Cryptography use

$Z_n$  when additive inverse are needed

$Z_n^*$  when multiplicative inverse are needed

# Substitution Ciphers

---

What is the key domain for any multiplicative inverse

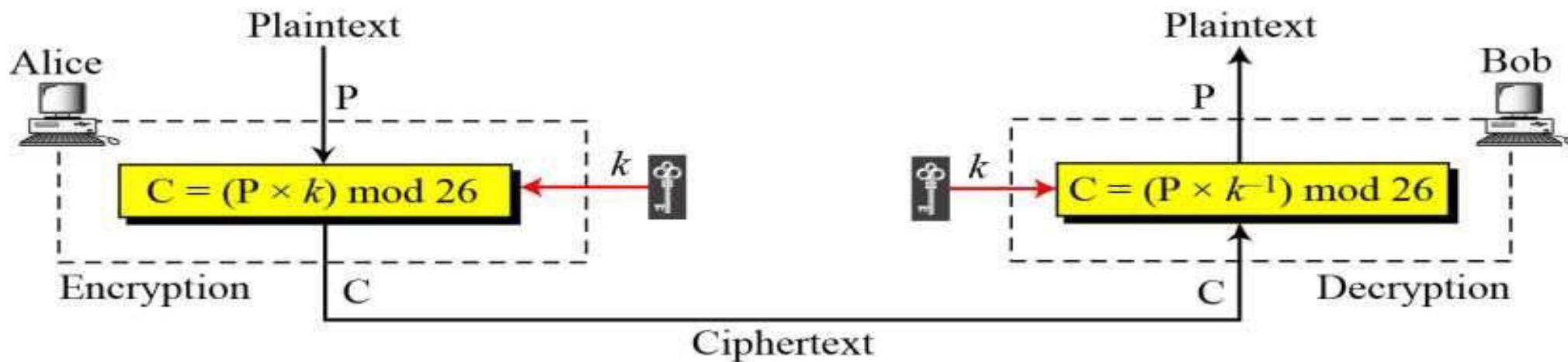
Key needs to be in  $Z_n^*$

$$Z_{26}^* = \{ 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25 \}$$

# Substitution Ciphers

## Monoalphabetic ciphers - Multiplicative ciphers

Use multiplicative cipher to encrypt the message “hello” with a key of 7.



# Substitution Ciphers

## Monoalphabetic ciphers - Multiplicative ciphers

Use multiplicative cipher to encrypt the message “hello” with a key of 7.

The ciphertext is “XCZZU”.

Plaintext: h → 07

Plaintext: e → 04

Plaintext: l → 11

Plaintext: l → 11

Plaintext: o → 14

Encryption:  $(07 \times 07) \bmod 26$

Encryption:  $(04 \times 07) \bmod 26$

Encryption:  $(11 \times 07) \bmod 26$

Encryption:  $(11 \times 07) \bmod 26$

Encryption:  $(14 \times 07) \bmod 26$

ciphertext: 23 → X

ciphertext: 02 → C

ciphertext: 25 → Z

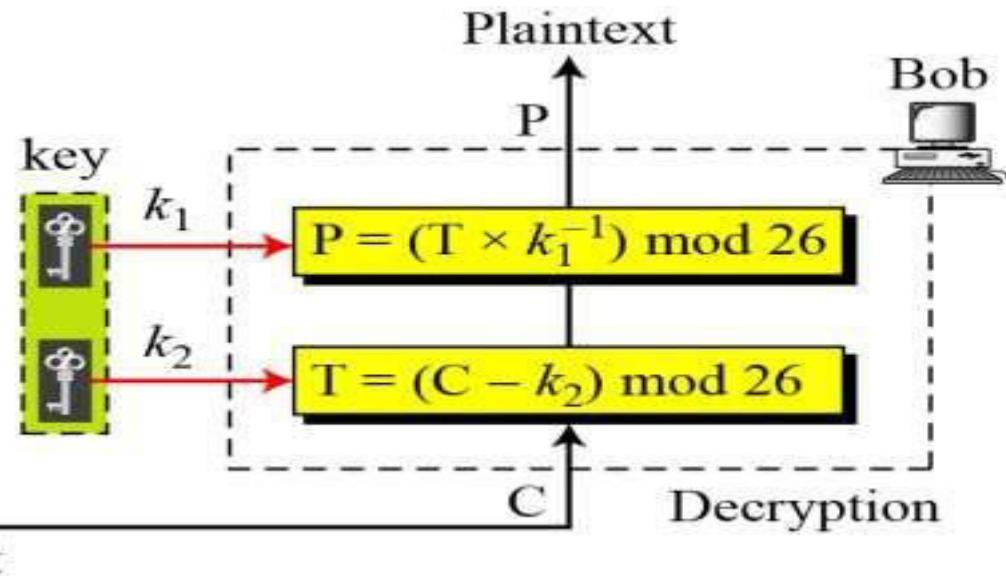
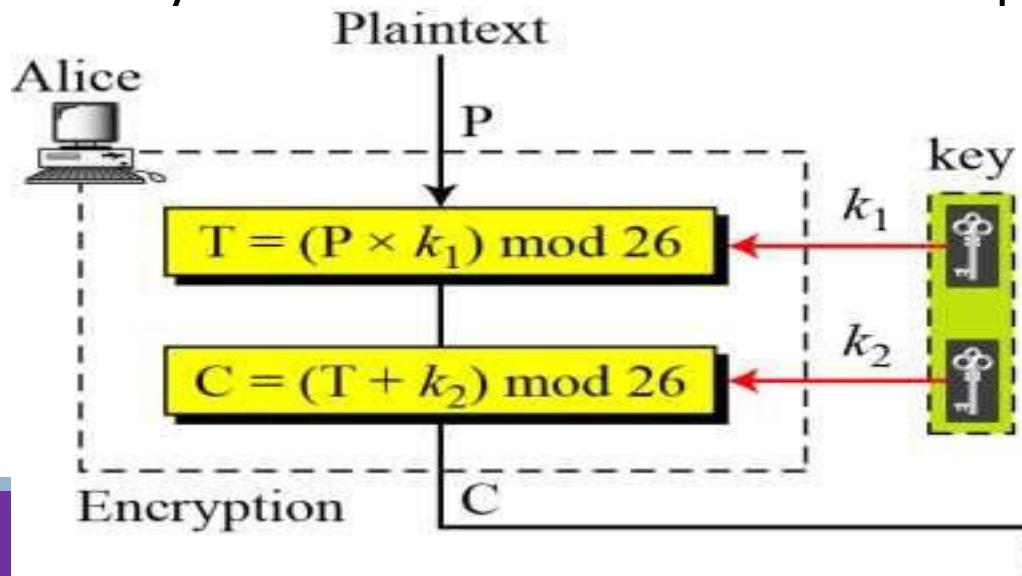
ciphertext: 25 → Z

ciphertext: 20 → U

# Substitution Ciphers

## **Monoalphabetic ciphers – Affine cipher**

- Combination of both cipher with a pair of keys
- First key is used with multiplicative cipher
- Second key is used with additive cipher



# Substitution Ciphers

---

What is the key domain for any affine inverse

First key is  $Z_n^*$  and second key needs to be in  $Z_{26}$

As we know that  $Z_{26}^* = \{ 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25 \}$

Size of key domain =  $12 * 26 \rightarrow 312$

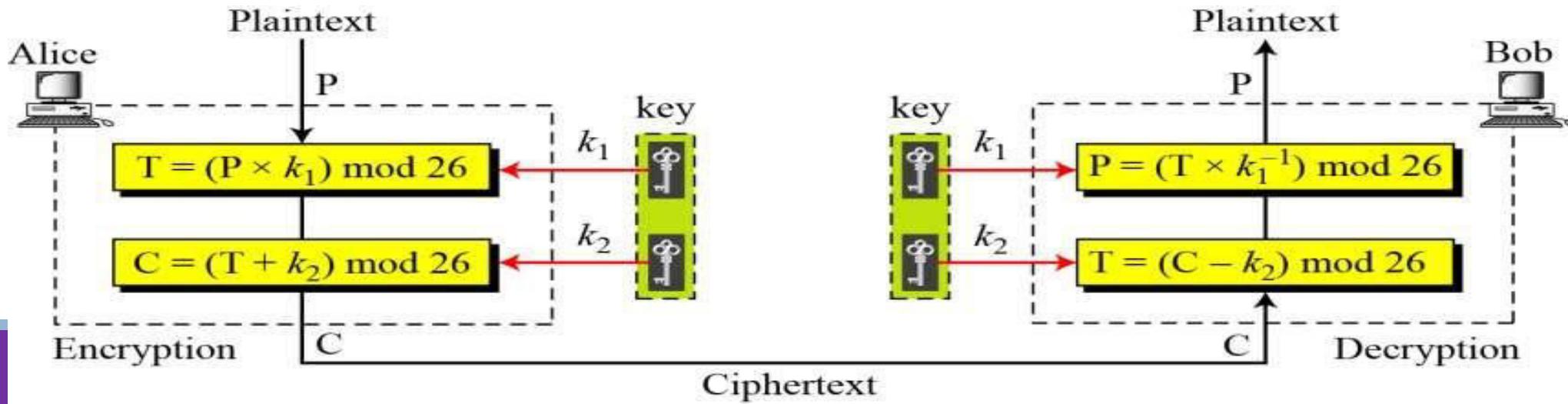
# Substitution Ciphers

Use an affine cipher to encrypt the message “hello” with the key pair  $(7, 2)$ .

$$P: h \rightarrow 07$$

$$\text{Encryption: } (07 \times 7 + 2) \bmod 26$$

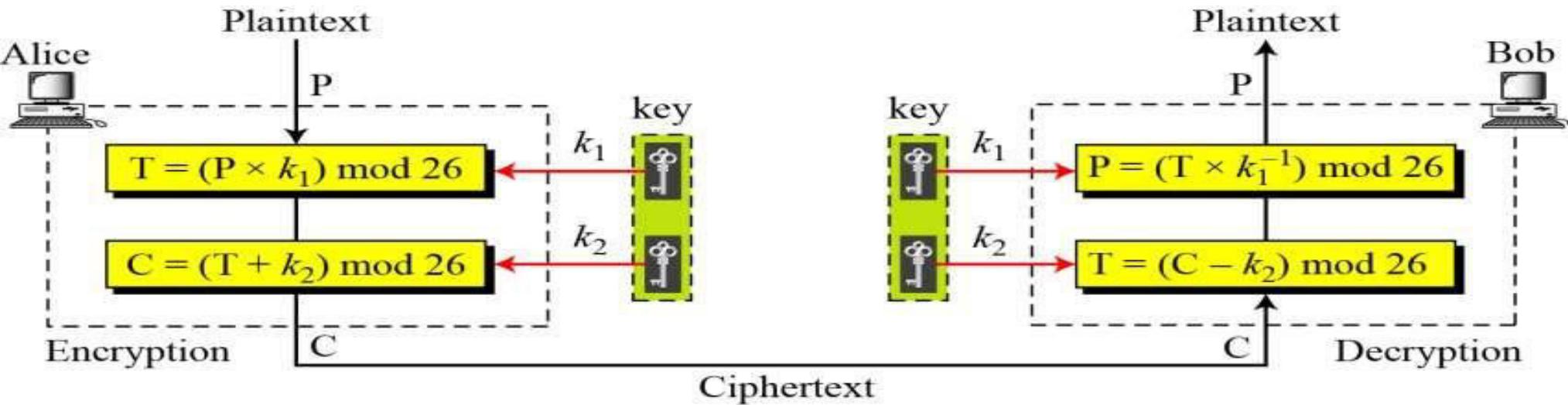
$$C: 25 \rightarrow Z$$



# Substitution Ciphers

Use an affine cipher to encrypt the message “hello” with the key pair  $(7, 2)$ .

P: h → 07	Encryption: $(07 \times 7 + 2) \text{ mod } 26$	C: 25 → Z
P: e → 04	Encryption: $(04 \times 7 + 2) \text{ mod } 26$	C: 04 → E
P: l → 11	Encryption: $(11 \times 7 + 2) \text{ mod } 26$	C: 01 → B
P: l → 11	Encryption: $(11 \times 7 + 2) \text{ mod } 26$	C: 01 → B
P: o → 14	Encryption: $(14 \times 7 + 2) \text{ mod } 26$	C: 22 → W



# Cipher = ZEBBW

$$C = Z \rightarrow 25$$

$$= ( (25 - 2) \times 7^{-1} ) \bmod 26$$

$$= (23 \times 7^{-1}) \bmod 26$$

$$= (23 \times 15) \bmod 26$$

$$= 345 \bmod 26$$

$$= 7$$

$$= H$$

$$C = E \rightarrow 04$$

$$= ( (4 - 2) \times 7^{-1} ) \bmod 26$$

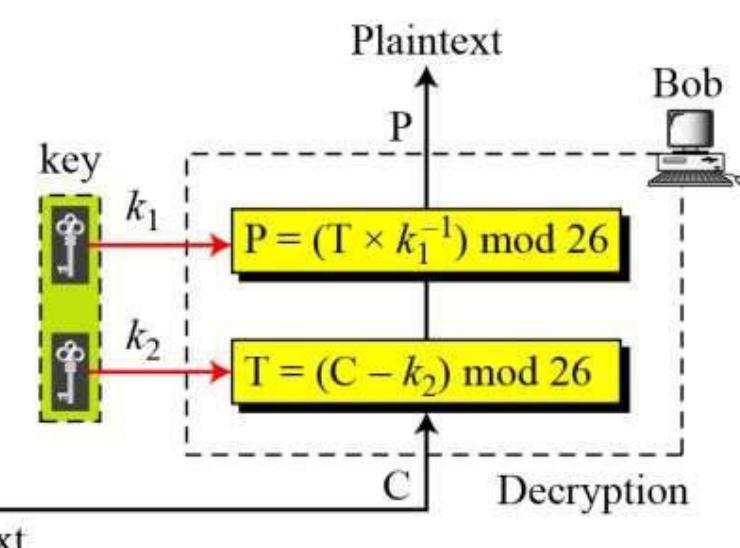
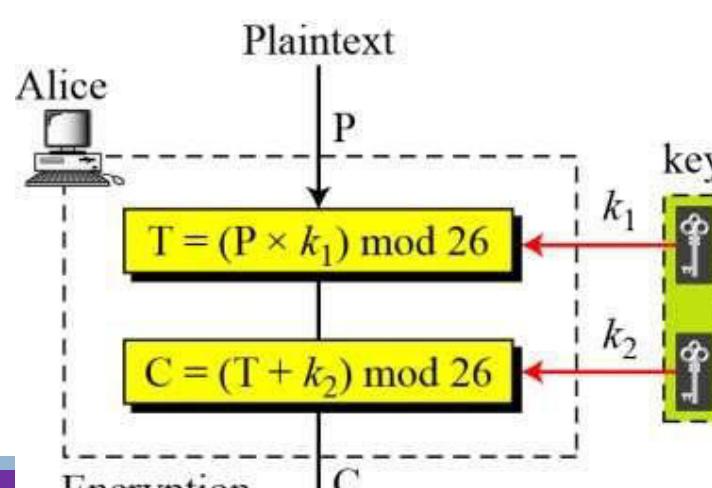
$$= (2 \times 7^{-1}) \bmod 26$$

$$= (2 \times 15) \bmod 26$$

$$= 30 \bmod 26$$

$$= 4$$

$$= 4$$



# Substitution Ciphers

Use the affine cipher to decrypt the message “ZEBBW” with the key pair  $(7, 2)$  in modulus 26.

$$C: Z \rightarrow 25$$

$$C: E \rightarrow 04$$

$$C: B \rightarrow 01$$

$$C: B \rightarrow 01$$

$$C: W \rightarrow 22$$

$$\text{Decryption: } ((25 - 2) \times 7^{-1}) \bmod 26$$

$$\text{Decryption: } ((04 - 2) \times 7^{-1}) \bmod 26$$

$$\text{Decryption: } ((01 - 2) \times 7^{-1}) \bmod 26$$

$$\text{Decryption: } ((01 - 2) \times 7^{-1}) \bmod 26$$

$$\text{Decryption: } ((22 - 2) \times 7^{-1}) \bmod 26$$

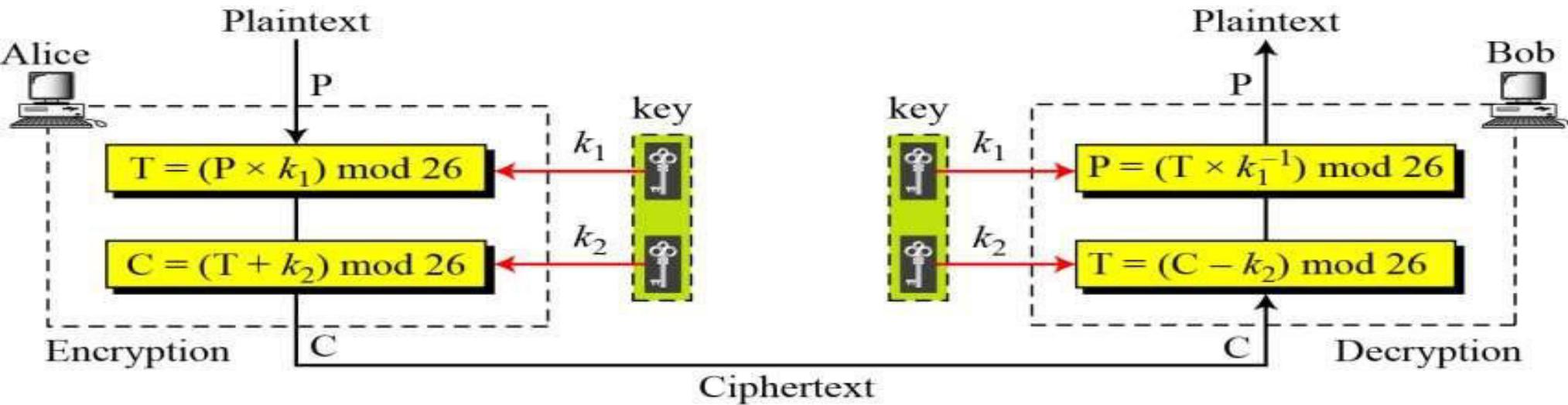
$$P: 07 \rightarrow h$$

$$P: 04 \rightarrow e$$

$$P: 11 \rightarrow l$$

$$P: 11 \rightarrow l$$

$$P: 14 \rightarrow o$$



# Substitution Ciphers

---

In general, In additive cipher

- First key  $k_1$  is used with multiplicative cipher
- Second key  $k_2$  is used with additive cipher

The additive cipher is a special case of an affine cipher in which  $k_1 = 1$ .

The multiplicative cipher is a special case of affine cipher in which  $k_2 = 0$ .

# Substitution Ciphers

---

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

# Substitution Ciphers

---

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

We can use the key to encrypt the message

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

The ciphertext is

# Substitution Ciphers

---

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

We can use the key to encrypt the message

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

The ciphertext is

ICFVQRVVNEFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

# 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.

PT = WELCOME

CT = XGPHUTR

		Plaintext																										
		Key																										
		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	
A	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	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		

# Substitution Ciphers

---

## **Polyalphabetic ciphers**

- a. Autokey cipher
- b. Playfair cipher
- c. Vigener cipher
- d. Hill cipher
- e. One time pad
- f. Rotor cipher

# Polyalphabetic Ciphers

---

## Autokey cipher

- In this cipher, key is a stream of subkeys, in which each subkey is used to encrypt the corresponding character in the plaintext.
- The first subkey is predetermined secret value agreed between Sender and receiver

$$P = P_1 P_2 P_3 \dots$$

$$C = C_1 C_2 C_3 \dots$$

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

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

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

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Assume that Alice and Bob agreed to use an autokey cipher with initial key value  $k_1 = 12$ .

Now Alice wants to send Bob the message “Attack is today”. Enciphering is done character by character.

Plaintext:	a	t	t	a	c	k	i	s	t	o	d	a	y
P's Values:	00	19	19	00	02	10	08	18	19	14	03	00	24
Key stream:	12	00	19										
C's Values:	12	19	12										
Ciphertext:	M	T	M										

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Assume that Alice and Bob agreed to use an autokey cipher with initial key value  $k_1 = 12$ .

Now Alice wants to send Bob the message “Attack is today”. Enciphering is done character by character.

Plaintext:	a	t	t	a	c	k	i	s	t	o	d	a	y
P's Values:	00	19	19	00	02	10	08	18	19	14	03	00	24
Key stream:	12	00	19	19	00	02	10	08	18	19	14	03	00
C's Values:	12	19	12	19	02	12	18	00	11	7	17	03	24
Ciphertext:	M	T	M	T	C	M	S	A	L	H	R	D	Y

# Polyalphabetic Ciphers

---

## Playfair cipher

- The best-known digraph substitution cipher, invented in 1854 by Charles Wheatstone but was named after Lord Playfair who promoted the use of the cipher.
- Used by British army during World war I.
- Secret key is made up of 25 characters arranged in 5\*5 matrix  
(I and J are considered same)

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

# Polyalphabetic Ciphers

## The Playfair Cipher Encryption Algorithm:

### 1. Generate the key Square(5×5):

The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext.

The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.

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

keyword = **ATHENS**

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

# Polyalphabetic Ciphers

---

The Playfair Cipher Encryption Algorithm:

**Before encryption:**

- Divide the plain text into digraphs
- If two letters in a pair are the same, a bogus letter is inserted to separate them
- After inserting bogus letter, if number of character is odd, one extra bogus character is added at the end to make number of characters even

# Polyalphabetic Ciphers

---

PT = attack

Digrams → at ta ck

PT = balloon

Digrams → ba ll oo n → ba lx lo on

PT = msit academy

Digrams → ms it ac ad em yx

# Polyalphabetic Ciphers

---

The Playfair Cipher Encryption Algorithm:

## 2. Encrypt the Plaintext → Three cipher rules

- a) If the two letters in a pair are located in the same row of the secret key, the corresponding encrypted character for each character is the next letter to the right in the same row(wrap to beginning of row)
- b) If the two letters in a pair are located in the same column of the secret key, the corresponding encrypted character for each character is the letter beneath in the same column(wrap to beginning of column)
- c) If the two letter in a pair are not in the same row or column of the secret, the corresponding encrypted character for each letter is a letter that is in its own row but in the same column as the other letter. (Form a rectangle)

# Polyalphabetic Ciphers

Encrypt the plaintext “hello” using the key shown

PT = hello

Digraphs = he ll o

he	Ix	lo
EC	QZ	BX

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

he → EC

Plaintext: hello

Ix → QZ

Ciphertext: ECQZBX

# Polyalphabetic Ciphers

**Example 1:** attack

**Digrams:** at ta ck

at	ta	ck
RS	SR	DE

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

1. Digrams.
2. Repeating Letters - Filler letter.
3. Same Column | ↓ | Wrap around.
4. Same row | → | Wrap around.
5. Rectangle | ⇄ | Swap

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

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

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

# Polyalphabetic Ciphers

Example 2: mosque

mo	sq	ue
ON	TS	ML

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

1. Digrams.
2. Repeating Letters - Filler letter.
3. Same Column | ↓ | Wrap around.
4. Same row | → | Wrap around.
5. Rectangle | ⇄ | Swap

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		

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

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

# Polyalphabetic Ciphers

Encrypt the plaintext “**PLEASE SAVE ME**” using the key = **CRYPTO**

Digraphs = PL EA SE SA VE ME

**CIPHER** = CQ OB ZK ME ZA SA

**CIPHER** = CQ OB ZK ME ZA SA

Digraphs = PL EA SE SA VE ME

c	r	y	p	t
o	a	b	d	e
f	g	h	i/j	k
l	m	n	o	s
u	v	w	x	z

1. Diagrams.
2. Repeating Letters - Filler letter.
3. Same Column | ↓ | Wrap around.
4. Same row | → | Wrap around.
5. Rectangle | ⇕ | Swap

# Polyalphabetic Ciphers

Encrypt the  
plaintext “**HIDE THE GOAL IN THE TREE STUMP**” using  
the key = **PLAYFAIR EXAMPLE**

p	i	a	y	f
i	r	e	x	m
b	c	d	g	h
k	n	o	q	s
t	u	v	w	z

# Vigenere Cipher

---

- Designed by Blaise de Vigenere, mathematician
- Vigenere cipher uses a different strategy to create the key stream .
- Keystream is a repetition of an initial secret key stream of length  $m$ , where  $1 \leq m \leq 26$ .
- Suppose Alice and Bob agree  $k = (k_1, k_2, k_3, \dots, k_m)$

$$P = P_1 P_2 P_3 \dots$$

$$C = C_1 C_2 C_3 \dots$$

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

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

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

---

Encrypt the message “She is listening” using the 6-character keyword “PASCAL”.

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

“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>	<i>15</i>	<i>00</i>	<i>18</i>	<i>02</i>	<i>00</i>	<i>11</i>								
<b>C's values:</b>	<b>07</b>	<b>07</b>	<b>22</b>	<b>10</b>	<b>18</b>	<b>22</b>								
<b>Ciphertext:</b>	<b>H</b>	<b>H</b>	<b>W</b>	<b>K</b>	<b>S</b>	<b>W</b>								

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

“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).

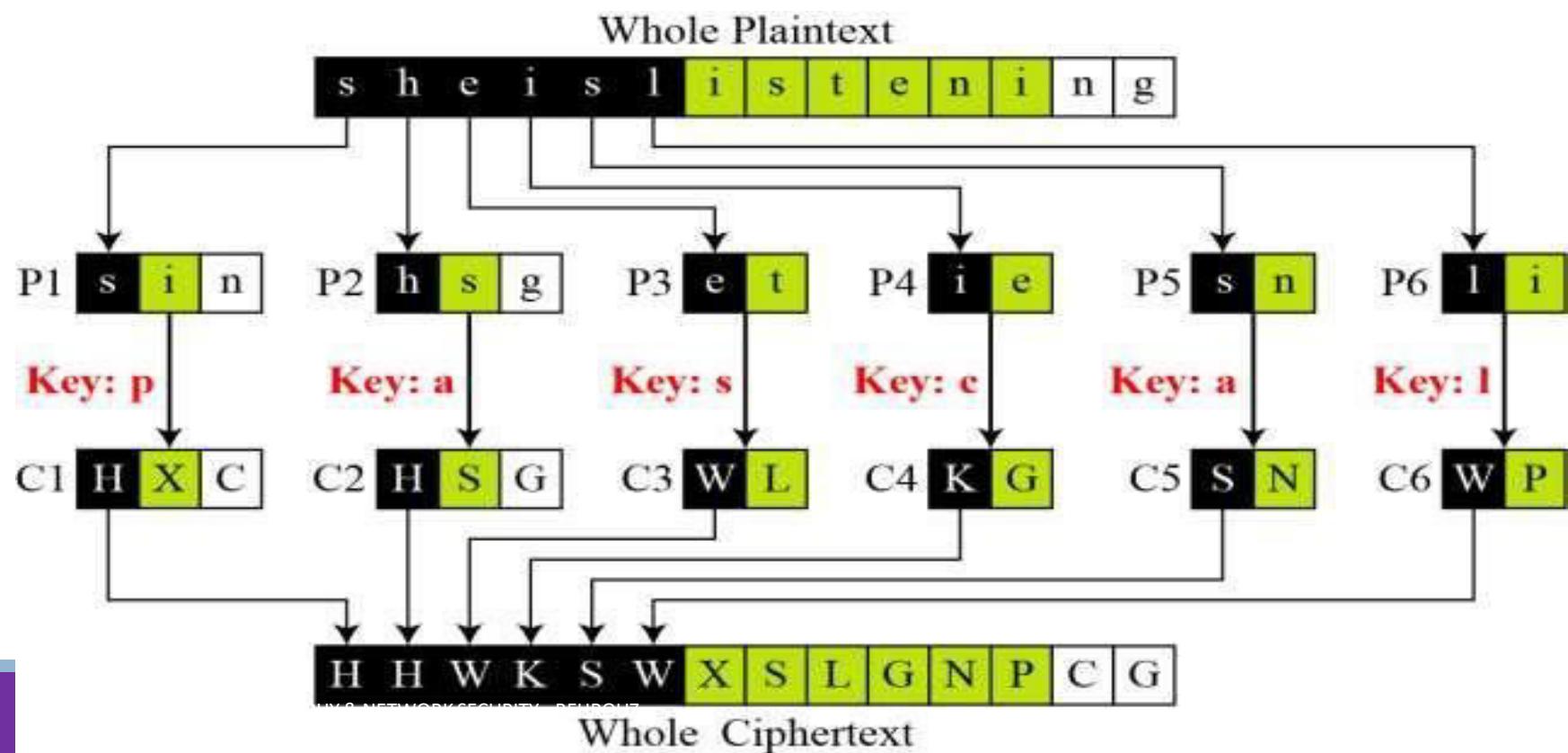
<b>Plaintext:</b>	s	h	e	i	s	l	i	s	t	e	n	i	n	g
<b>P's values:</b>	18	07	04	08	18	11	08	18	19	04	13	08	13	06
<b>Key stream:</b>	15	00	18	02	00	11	15	00	18	02	00	11	15	00
<b>C's values:</b>	07	07	22	10	18	22	23	18	11	6	13	19	02	06
<b>Ciphertext:</b>	H	H	W	K	S	W	X	S	L	G	N	T	C	G

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

PT "She is listening"

Keyword "PASCAL".

A Vigenere cipher as a combination of m additive ciphers





	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

We can say that the additive cipher is a special case of Vigenere cipher in which  $m = 1$ .

---

Plain text

**“She is listening”**

keyword

**“PASCAL”.**

s	h	e	i	s	l	i	s	t	e	n	i	n	g
18	07	04	08	18	11	08	18	19	04	13	08	13	06
<b>15</b>	<b>00</b>	<b>18</b>	<b>02</b>	<b>00</b>	<b>11</b>	<b>15</b>	<b>00</b>	<b>18</b>	<b>02</b>	<b>00</b>	<b>11</b>	<b>15</b>	<b>00</b>
07	07	22	10	18	22	23	18	11	6	13	19	02	06
H	H	W	K	S	W	X	S	L	G	N	T	C	G

## Cryptanalysis of Vigenere Ciphers

---

Eve can use technique to decipher the intercepted cipher text.

The cryptanalysis consist of :

- Finding the length of key
- Finding the key

## Cryptanalysis of Vigenere Ciphers

---

Several methods to find the length of key

- Kasiski test

---

## Vigenere Cipher (Crypanalysis)

The Kasiski test for repetition of three-character segments

Let us assume eve intercepted the following ciphertext:

LIOMWGFEGGDVWGHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTH-  
VTSGXQOVGCSVETQLTJSUMVVVEUVLXEWSLGFZMVVWLGYHCUSWXQH-  
KVGSHHEEVFLCFDGVSUMPHKIRZDMPHHBVWWJWIXGFWLTSHGJOUEEEHH-  
VUCFVGOWICQLTJSUXGLW

LIOMWGFEGGDVWGHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTH-  
VTSGXQOVGCSVETQLTJSUMVWVEUVLXEWSLGFZMVVWLGYHCUSWXQH-  
KVGSHHEEVFLCFDGVSUMPHKIRZDMPHHBVWWJWIXGFWLTSHGJOUEEEHH-  
VUCFVGOWICQLTJSUXGLW

## Vigenere Cipher (Crypanalysis)

The Kasiski test for repetition of three-character segments yields the results shown in Table

LIOMWGEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTH-  
 VTSGXQOVGCSVETQLTJSUMVWVEUVLXEWSLGFMVVWLGYHCUSWXQH-  
 KVGSHEEVFLCFDGSUMPHKIRZDMPHHBVWVWJWIXGFWLTSHGJOUEEEHH-  
 VUCFVGOWICQLTJSUXGLW

<i>String</i>	<i>First Index</i>	<i>Second Index</i>	<i>Difference</i>
JSU	68	168	100
SUM	69	117	48
VWV	72	132	60
MPH	119	127	8

# One Time pad

---

- One of the goals of cryptography is perfect secrecy.
- A study by Shannon has shown that perfect secrecy can be achieved if each plaintext symbol is encrypted with a key randomly chosen from a key domain.
- This idea is used in a cipher called one-time pad, invented by **Vernam**.

# One Time pad

---

- It is a method of encrypting alphabetic plain text.
- It is one of the Substitution techniques which converts plain text into ciphertext.
- In this mechanism, we assign a number to each character of the Plain-Text.

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

# One Time pad

---

- The key must be the same size as the message being sent.
- The key must be truly random.
- Keys must be securely shared between the sending and receiving parties.

Because of these strict conditions, the use of one-time pad over digital media is impracticable

# One Time pad

- The relation between the key and plain text: In this algorithm, the length of the key should be equal to that of plain text.
- Encrypt the Plaintext = HELLO and Key = MONEY
- HELLO → 7 4 11 11 14
- MONEY → 12 14 13 4 24

$$\begin{array}{ccccc} 19 & 18 & 24 & 15 & (38 - 26) \\ 19 & 18 & 24 & 15 & 12 \rightarrow \textcolor{red}{T S Y P M} \end{array}$$

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

# One Time pad

---

CT= TSYPM → 19 18 24 15 12

KEY=MONEY → 12 14 3 4 24

$$\begin{array}{r}
 7 \quad 4 \quad 11 \quad 11 \quad -12 + 26 \\
 7 \quad 4 \quad 11 \quad 11 \quad 14 \\
 \text{H} \quad \text{E} \quad \text{L} \quad \text{L} \quad \text{O}
 \end{array}$$

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

# One Time pad

Plaintext = TEST

Key = FVEB

PT = ENIGMA

Key = KEYWOR

Plaintext	T	19	+	F	5	=	24	
	E	4	+	V	21	=	25	Ciphertext YZWU
	S	18	+	E	4	=	22	
	T	19	+	B	1	=	20	

Keyword

# One Time pad

---

When a message is to be sent, the sender uses the secret key to encrypt each character one at a time.

If a computer is used, each bit in the character, which is usually eight bits in length is exclusively OR'ed with the corresponding bit in the secret key.

With a one-time pad, the encryption algorithm is simply the **XOR operation**.

# One Time pad

A	B	A <b>XOR</b> B
0	0	0
0	1	1
1	0	1
1	1	0

## ENCRYPT

$$\begin{array}{r} \oplus \\ \text{00110101 Plaintext} \\ \text{11100011 Secret Key} \\ \hline = \text{11010110 Ciphertext} \end{array}$$

## DECRYPT

$$\begin{array}{r} \oplus \\ \text{11010110 Ciphertext} \\ \text{11100011 Secret Key} \\ \hline = \text{00110101 Plaintext} \end{array}$$

# Hill Cipher

---

- The Hill Cipher was invented by Lester S. Hill in 1929
- It acts on groups of letters.
- It is a polygraphic substitution cipher, as it can work on digraphs, trigraphs (3 letter blocks) or theoretically any sized blocks.

# Hill Cipher

---

- Key is a square matrix of size  **$m \times m$  matrix** in which  $m$  is the size of the block(  $2 \times 2$  matrix for digraphs, a  $3 \times 3$  matrix for trigraphs).

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

# Hill Cipher

---

- **Encryption**

Turn the plaintext into digraphs (or trigraphs) and each of these into a column vector.

To encrypt a message, each block of n letters is multiplied by an  $m \times m$  matrix, against modulus 26.  $C = K \cdot P \text{ mod } 26$

- **Decryption**

To decrypt the message, each block is multiplied by the inverse of the matrix

# Hill Cipher

---

Encrypt the plaintext message "**short example**" using the keyword **hill** with a **2 x 2 matrix**.

# Hill Cipher

---

- The first step is to turn the keyword **hill** into a matrix.

- hill → 7 8 11 11

$$\begin{pmatrix} H & I \\ L & L \end{pmatrix} \rightarrow \begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix}$$

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

# Hill Cipher

---

Split the plaintext into digraphs, and write these as column vectors.

short example → sh      or      te      xa      mp      le  
                       18 7    14 17    19 4    23 0    12 15    11 4

$$\binom{s}{h} \binom{o}{r} \binom{t}{e} \binom{x}{a} \binom{m}{p} \binom{l}{e}$$

$$\binom{18}{7} \binom{14}{17} \binom{19}{4} \binom{23}{0} \binom{12}{15} \binom{11}{4}$$

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

# Hill 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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

**Encryption :** Perform matrix multiplication, multiply the key matrix by each column vector in turn.

$$\begin{pmatrix} 18 \\ 7 \end{pmatrix} \begin{pmatrix} 14 \\ 17 \end{pmatrix} \begin{pmatrix} 19 \\ 4 \end{pmatrix} \begin{pmatrix} 23 \\ 0 \end{pmatrix} \begin{pmatrix} 12 \\ 15 \end{pmatrix} \begin{pmatrix} 11 \\ 4 \end{pmatrix} \quad \text{Key} = \begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix}$$

$$C = K * P \bmod 26 \quad \begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{pmatrix} 18 \\ 7 \end{pmatrix} = \begin{pmatrix} 182 \\ 275 \end{pmatrix}$$

$$= \begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{pmatrix} 18 \\ 7 \end{pmatrix} \bmod 26 \quad = \begin{pmatrix} 182 \\ 275 \end{pmatrix} \bmod 26$$

$$= 7 \times 18 + 8 \times 7 = 182$$

$$11 \times 18 + 11 \times 7 = 275$$

$$= \begin{pmatrix} 0 \\ 15 \end{pmatrix} = \begin{pmatrix} A \\ P \end{pmatrix}$$

# Hill Cipher

$$C = K * P \bmod 26$$

$$\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{pmatrix} 14 \\ 17 \end{pmatrix} \bmod 26$$

$$7 \times 14 + 8 \times 17 = 234$$

$$11 \times 14 + 11 \times 17 = 341$$

$$= \begin{pmatrix} 234 \\ 341 \end{pmatrix} \bmod 26$$

$$= \begin{pmatrix} 0 \\ 3 \end{pmatrix} = \begin{pmatrix} A \\ D \end{pmatrix}$$

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

$$\begin{pmatrix} 18 \\ 7 \end{pmatrix} \begin{pmatrix} 14 \\ 17 \end{pmatrix} \begin{pmatrix} 19 \\ 4 \end{pmatrix} \begin{pmatrix} 23 \\ 0 \end{pmatrix} \begin{pmatrix} 12 \\ 15 \end{pmatrix} \begin{pmatrix} 11 \\ 4 \end{pmatrix}$$

# Hill Cipher

$$C = K * P \bmod 26$$

$$\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{pmatrix} 19 \\ 4 \end{pmatrix} \bmod 26$$

$$7 \times 19 + 8 \times 4 = 165$$

$$11 \times 19 + 11 \times 4 = 253$$

$$= \begin{pmatrix} 165 \\ 253 \end{pmatrix} \bmod 26$$

$$= \begin{pmatrix} 9 \\ 19 \end{pmatrix} = \begin{pmatrix} J \\ T \end{pmatrix}$$

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

$$\begin{pmatrix} 18 \\ 7 \end{pmatrix} \begin{pmatrix} 14 \\ 17 \end{pmatrix} \begin{pmatrix} 19 \\ 4 \end{pmatrix} \begin{pmatrix} 23 \\ 0 \end{pmatrix} \begin{pmatrix} 12 \\ 15 \end{pmatrix} \begin{pmatrix} 11 \\ 4 \end{pmatrix}$$

# Hill Cipher

$$C = K * P \bmod 26$$

$$\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{pmatrix} 23 \\ 0 \end{pmatrix} = \begin{pmatrix} 161 \\ 253 \end{pmatrix} \bmod 26$$

$$\begin{pmatrix} 5 \\ 19 \end{pmatrix} = \begin{pmatrix} F \\ T \end{pmatrix}$$

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

$$\begin{pmatrix} 18 \\ 7 \end{pmatrix} \begin{pmatrix} 14 \\ 17 \end{pmatrix} \begin{pmatrix} 19 \\ 4 \end{pmatrix} \begin{pmatrix} 23 \\ 0 \end{pmatrix} \begin{pmatrix} 12 \\ 15 \end{pmatrix} \begin{pmatrix} 11 \\ 4 \end{pmatrix}$$

# Hill Cipher

$$C = K \cdot P \bmod 26$$

$$\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{pmatrix} 12 \\ 15 \end{pmatrix} = \begin{pmatrix} 204 \\ 297 \end{pmatrix} \bmod 26$$

$$= \begin{pmatrix} 22 \\ 11 \end{pmatrix} = \begin{pmatrix} W \\ L \end{pmatrix}$$

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

$$\begin{pmatrix} 18 \\ 7 \end{pmatrix} \begin{pmatrix} 14 \\ 17 \end{pmatrix} \begin{pmatrix} 19 \\ 4 \end{pmatrix} \begin{pmatrix} 23 \\ 0 \end{pmatrix} \begin{pmatrix} 12 \\ 15 \end{pmatrix} \begin{pmatrix} 11 \\ 4 \end{pmatrix}$$

# Hill Cipher

$$C = K \cdot P \bmod 26$$

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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

$$\begin{pmatrix} 18 \\ 7 \end{pmatrix} \begin{pmatrix} 14 \\ 17 \end{pmatrix} \begin{pmatrix} 19 \\ 4 \end{pmatrix} \begin{pmatrix} 23 \\ 0 \end{pmatrix} \begin{pmatrix} 12 \\ 15 \end{pmatrix} \begin{pmatrix} 11 \\ 4 \end{pmatrix}$$

$$\begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{pmatrix} 11 \\ 4 \end{pmatrix} = \begin{pmatrix} 109 \\ 165 \end{pmatrix} \bmod 26 = \begin{pmatrix} 5 \\ 9 \end{pmatrix} = \begin{pmatrix} F \\ J \end{pmatrix}$$

Cipher text = APADJTFWLFJ

# Hill 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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

**Decryption :** Perform inverse matrix multiplication

Cipher text = APADJTFWTLFJ

$$\text{Key} = \begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$P = K^{-1} * C \bmod 26$$

Find out inverse matrix of given key matrix.

$$K^{-1} = \frac{1}{|K|} * K_{\text{adj}}$$

# Hill 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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Cipher text = APADJFTWLFJ

$$K = \begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Find out inverse matrix of given key matrix.

$$K^{-1} = \frac{1}{|K|} * K_{adj}$$

$$|K| = ad - bc = 7*11 - 8*11 = 77 - 88 \rightarrow -11 \bmod 26 = 15$$

$$K_{adj} = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \begin{bmatrix} 11 & -8 \\ -11 & 7 \end{bmatrix}$$

# Hill 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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Cipher text = APADJTFWLFJ

Find out inverse matrix of given key matrix.

$$K^{-1} = \frac{1}{|K|} * K_{adj}$$

$$= \frac{1}{15} * \begin{pmatrix} 11 & -8 \\ -11 & 7 \end{pmatrix}$$

$$= 15^{-1} * \begin{pmatrix} 11 & -8 \\ -11 & 7 \end{pmatrix} \text{ mod } 26$$

$$K = \begin{pmatrix} 7 & 8 \\ 11 & 11 \end{pmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$15^{-1} \text{ mod } 26 = ?$$

$$15(?) \text{ mod } 26 = 1$$

$$15(1) \text{ mod } 26 = 15 \text{ not equal to } 1$$

$$15(2) \text{ mod } 26 = 4 \text{ not equal to } 1$$

.

.

$$15(7) \text{ mod } 26 = 1$$

$$\text{So, } 15^{-1} \text{ mod } 26 = 7$$

Find the multiplicative inverse of 15 in  $\mathbb{Z}_{26}$ .

$$\begin{array}{ll} r_1 \leftarrow n; & r_2 \leftarrow b; \\ t_1 \leftarrow 0; & t_2 \leftarrow 1; \end{array}$$

while ( $r_2 > 0$ )

{  
 $q \leftarrow r_1 / r_2;$

$r \leftarrow r_1 - q \times r_2;$

$r_1 \leftarrow r_2;$        $r_2 \leftarrow r;$

$t \leftarrow t_1 - q \times t_2;$

$t_1 \leftarrow t_2;$        $t_2 \leftarrow t;$

}

if ( $r_1 = 1$ ) then  $b^{-1} \leftarrow t_1$

q	r1	r2	r	t1	t2	t
1	26	15	11	0	1	-1
1	15	11	4	1	-1	2
2	11	4	3	-1	2	-5
1	4	3	1	2	-5	7
3	3	1	0	-5	7	-26
	1	0		7	-26	

The gcd (26, 15 ) is 1; the inverse of 15 is 7

# Hill Cipher

---

Find out inverse matrix of given key matrix.

$$K^{-1} = \frac{1}{|K|} * K_{\text{adj}}$$

$$= 7 * \begin{pmatrix} 11 & -8 \\ -11 & 7 \end{pmatrix} \text{ mod } 26$$

$$= 7 * \begin{pmatrix} 11 & 18 \\ 15 & 7 \end{pmatrix} \text{ mod } 26$$

$$= \begin{pmatrix} 77 & 126 \\ 105 & 49 \end{pmatrix} \text{ mod } 26 \quad = \begin{pmatrix} 25 & 22 \\ 1 & 23 \end{pmatrix}$$

# Hill 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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Cipher text = APADJTFTWLFJ

$$P = K^{-1} * C \bmod 26 = \begin{pmatrix} 25 & 22 \\ 1 & 23 \end{pmatrix} \begin{pmatrix} 0 \\ 15 \end{pmatrix}$$

$$= 25*0 + 22*15 = 330 \bmod 26 = 18 = S$$

$$1*0 + 23 * 15 = 345 \bmod 26 = 7 = H$$

# Hill 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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Plain Text = SAF EME SSA GES

Cipher text = HDS IOE YQO CAA

$$P = K^{-1} * C \bmod 26$$

Find out inverse matrix of given key matrix.

$$K^{-1} = \frac{1}{|K|} * K_{adj}$$

# Hill 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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Cipher text = HDS IOE YQO CAA

Find out inverse matrix of given key matrix.

$$K^{-1} = \frac{1}{|K|} * K_{adj}$$

$$K = \begin{pmatrix} 2 & 8 & 15 \\ 7 & 4 & 17 \\ 8 & 13 & 6 \end{pmatrix}$$

$$\begin{aligned}
 |K| &= \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix} \\
 &= a(ei - fh) - b(di - fg) + c(dh - eg) \\
 &= aei - afh - bdi + bfg + cdh - ceg \\
 &= (aei + bfg + cdh) - (afh + bdi + ceg)
 \end{aligned}$$

# Hill 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 →	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
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

$$\begin{aligned}
 |\mathbf{K}| &= \begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix} \\
 &= a(ei - fh) - b(di - fg) + c(dh - eg) \\
 &= 2(4*6 - 17*13) - 8(7*6 - 17*8) + 15(7*13 - 4*8) \\
 &= 1243 \text{ mod } 26 \rightarrow 21
 \end{aligned}$$

(matA<sup>-1</sup>) \* (det(matA))  
 ==inverse  
 of matrix

$$= \begin{pmatrix} 2 & 8 & 15 \\ 7 & 4 & 17 \\ 8 & 13 & 6 \end{pmatrix}$$

$$\begin{aligned}
 21^{-1} \bmod 26 &=? \\
 21 ( ? ) \bmod 26 &= 1 \\
 21(1) \bmod 26 &= 21 \text{ not equal to } 1 \\
 21(2) \bmod 26 &= 42 \text{ not equal to } 1 \\
 \cdot & \\
 \cdot & \\
 21(5) \bmod 26 &= 1
 \end{aligned}$$

$$\text{So, } 21^{-1} \bmod 26$$

$$K^{-1} = \frac{1}{|D|} \text{adj}(K)$$

$$D^{-1} \text{adj}(K) = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}^{-1}$$

$$= \frac{1}{|D|} \begin{bmatrix} + (ei - fh) & - (di - fg) & + (dh - eg) \\ - (bi - ch) & + (ai - cg) & - (ah - bg) \\ + (bf - ce) & - (af - cd) & + (ae - bd) \end{bmatrix}^T$$

# Hill Cipher

$$\begin{aligned}
 K_{adj} &= \begin{pmatrix} 2 & 8 & 15 \\ 7 & 4 & 17 \\ 8 & 13 & 6 \end{pmatrix} \\
 &= \begin{pmatrix} 4*6 - 17*13 & 7*6 - 17*8 & 7 *13 - 4*8 \\ 8 *6 - 15*13 & 2*6 -15*8 & 2*13 - 8 *8 \\ 8*17 - 15*4 & 2*17 - 15*7 & 2*4 - 8*7 \end{pmatrix}
 \end{aligned}$$

$$\begin{pmatrix} -197 & -94 & 59 \\ 147 & -108 & 38 \\ 76 & 71 & -48 \end{pmatrix} \begin{pmatrix} + & - & + \\ - & + & - \\ + & - & + \end{pmatrix} \begin{pmatrix} -197 & 94 & 59 \\ 147 & -108 & 38 \\ 76 & 71 & -48 \end{pmatrix}$$

# Hill Cipher

Take  
transpose

$$\begin{pmatrix} -197 & 94 & 59 \\ 147 & -108 & 38 \\ 76 & 71 & -48 \end{pmatrix}$$

$$\begin{pmatrix} -197 & 147 & 76 \\ 94 & -108 & 71 \\ 59 & 38 & -48 \end{pmatrix}$$

$$\begin{pmatrix} 11 & 147 & 76 \\ 94 & 22 & 38 \\ 59 & 71 & 4 \end{pmatrix}$$

Repeatedly add +26 to  
make negative to  
positive

# Hill Cipher

---

$$= 5 * \begin{pmatrix} 11 & 147 & 76 \\ 94 & 22 & 38 \\ 59 & 71 & 4 \end{pmatrix}$$

$$\begin{pmatrix} 55 & 735 & 380 \\ 470 & 110 & 355 \\ 295 & 190 & 20 \end{pmatrix}$$

Mod 26 →

$$\begin{pmatrix} 3 & 7 & 16 \\ 2 & 6 & 17 \\ 9 & 8 & 20 \end{pmatrix}$$

$$P = K^{-1} C \bmod 26 = \begin{bmatrix} 3 & 7 & 16 \\ 2 & 6 & 17 \\ 9 & 8 & 20 \end{bmatrix} \begin{bmatrix} 7 \\ 3 \\ 18 \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 3(7) + 7(3) + 16 \\ 2(7) + 6(3) + 17 \\ 9(7) + 8(3) + 20 \end{bmatrix} = \begin{bmatrix} 330 \\ 338 \\ 447 \end{bmatrix} \bmod 26 = \begin{bmatrix} 18 \\ 0 \\ 5 \end{bmatrix} = \begin{bmatrix} s \\ a \\ f \end{bmatrix}.$$

$$= \begin{bmatrix} 330 \\ 338 \\ 447 \end{bmatrix} \bmod 26 = \begin{bmatrix} 18 \\ 0 \\ 5 \end{bmatrix} = \begin{bmatrix} s \\ a \\ f \end{bmatrix}.$$

$$P = K^{-1} C \bmod 26 = \begin{bmatrix} 3 & 7 & 16 \\ 2 & 6 & 17 \\ 9 & 8 & 20 \end{bmatrix} \begin{bmatrix} 8 \\ 14 \\ 4 \end{bmatrix} \bmod 26 = \begin{bmatrix} 3(8) + 7(14) + 16(4) \\ 2(8) + 6(14) + 17(4) \\ 9(8) + 8(14) + 20(4) \end{bmatrix} \bmod 26 = \begin{bmatrix} 186 \\ 168 \\ 264 \end{bmatrix} \bmod 26 = \begin{bmatrix} 4 \\ 12 \\ 4 \end{bmatrix} = \begin{bmatrix} e \\ m \\ e \end{bmatrix}.$$

$$P = K^{-1} C \bmod 26 = \begin{bmatrix} 3 & 7 & 16 \\ 2 & 6 & 17 \\ 9 & 8 & 20 \end{bmatrix} \begin{bmatrix} 24 \\ 16 \\ 14 \end{bmatrix} \bmod 26 = \begin{bmatrix} 3(24) + 7(16) + 16(14) \\ 2(24) + 6(16) + 17(14) \\ 9(24) + 8(16) + 20(14) \end{bmatrix} \bmod 26 = \begin{bmatrix} 18 \\ 18 \\ 0 \end{bmatrix} = \begin{bmatrix} s \\ s \\ a \end{bmatrix}.$$

$$P = K^{-1} C \bmod 26 = \begin{bmatrix} 3 & 7 & 16 \\ 2 & 6 & 17 \\ 9 & 8 & 20 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \bmod 26 = \begin{bmatrix} 3(2) + 7(0) + 16(0) \\ 2(2) + 6(0) + 17(0) \\ 9(2) + 8(0) + 20(0) \end{bmatrix} \bmod 26 = \begin{bmatrix} 6 \\ 4 \\ 18 \end{bmatrix} = \begin{bmatrix} g \\ e \\ s \end{bmatrix}.$$

---

Encrypt the plaintext “attack”, using Hill cipher for the given key =  $\begin{bmatrix} 2 & 3 \\ 3 & 6 \end{bmatrix}$ .

Ciphertext: “FKMFIO”.

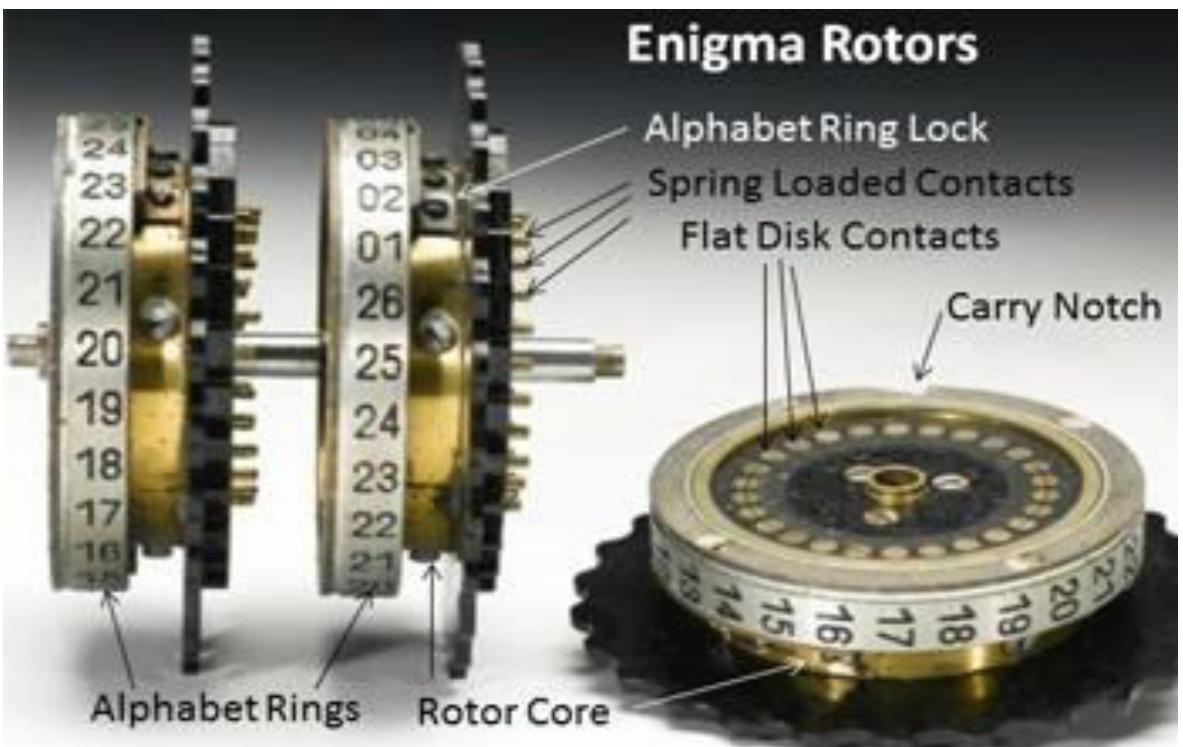
Encrypt the message “retreat now” using the keyphrase back up and a  $3 \times 3$  matrix.

$$\binom{r}{e} \binom{r}{e} \binom{t}{n} \binom{w}{x}$$

$$\begin{pmatrix} B & A & C \\ K & U & P \\ A & B & C \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 2 \\ 10 & 20 & 15 \\ 0 & 1 & 2 \end{pmatrix}$$

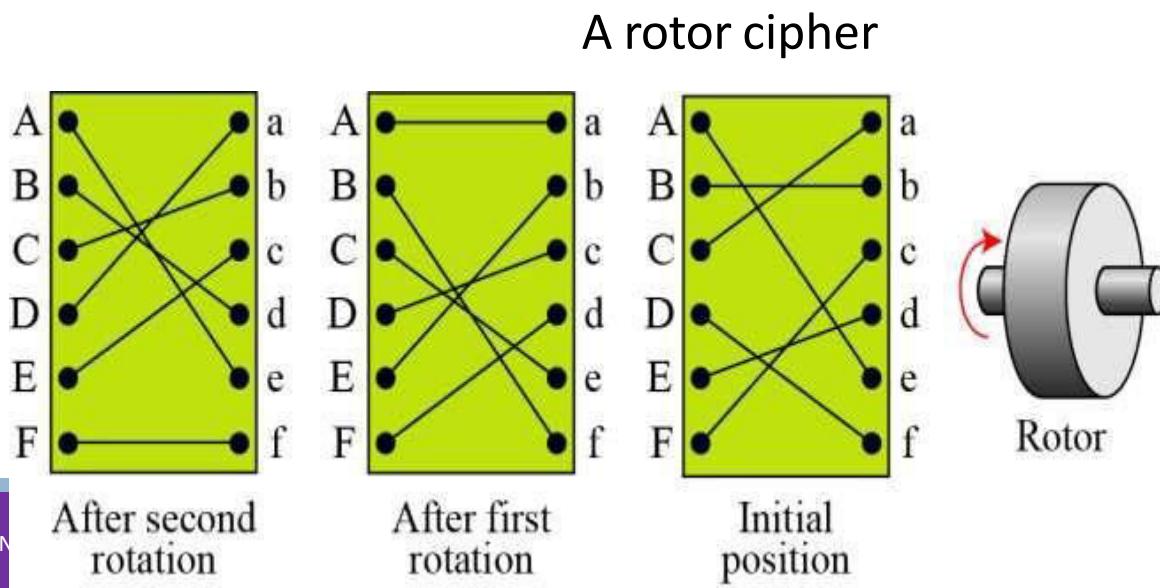
# ROTOR CIPHER

A rotor is a thick disk that has near its outer circumference on both sides as many electric contacts as letters in the alphabet (usually 26—early versions of the Enigma used 28 or 29 letters).



# ROTOR CIPHER

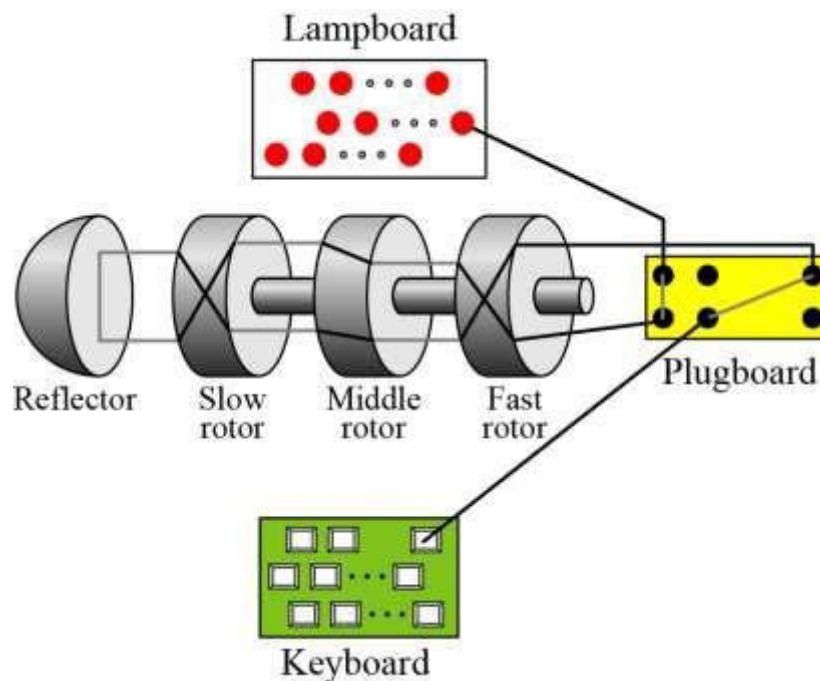
- Rotor uses only 6 letters
- Initial setting is key agreed between sender and receiver
- First character is encrypted using initial position
- Second character after first rotation
- Third character after second rotation
- Example : bee → **BCA**



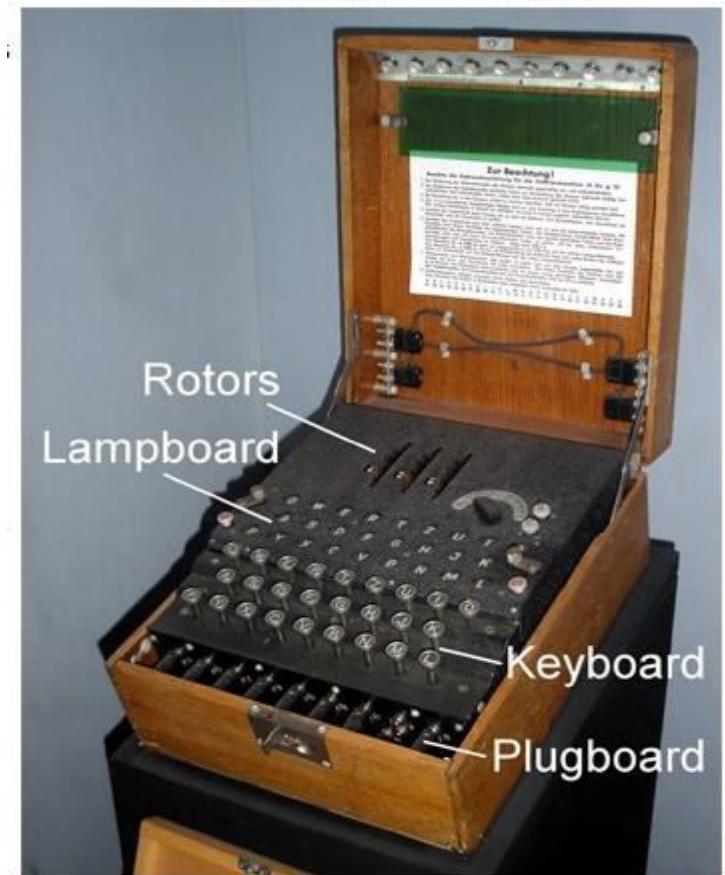
# ROTOR CIPHER

Main components of Enigma machine

1. Keyboard = 26keys
2. Lampboard =26 lamps
3. Plugboard = 26plugs
4. Three rotors
5. Reflector



**The Enigma Ciphertext Machine**

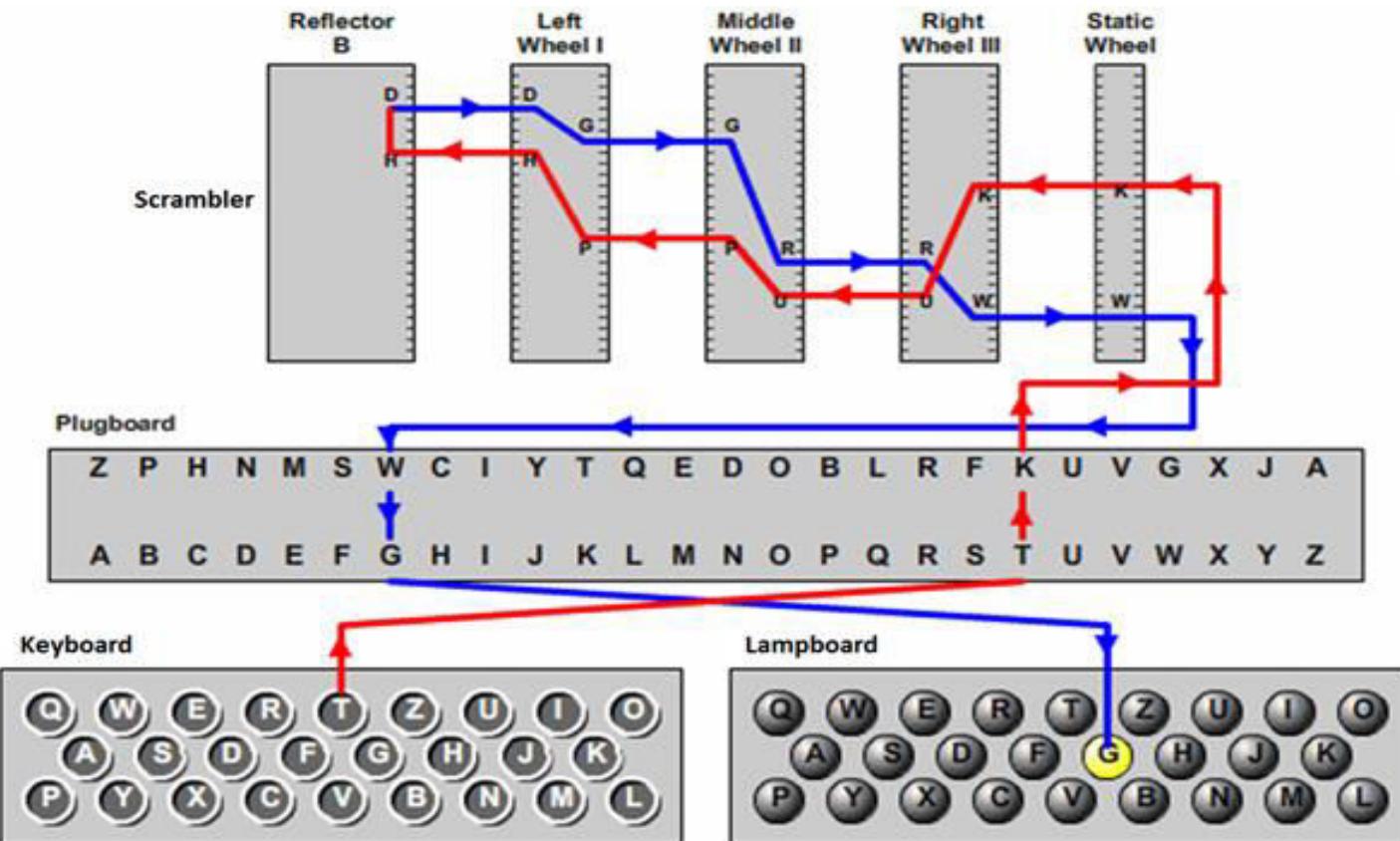


# ROTOR CIPHER

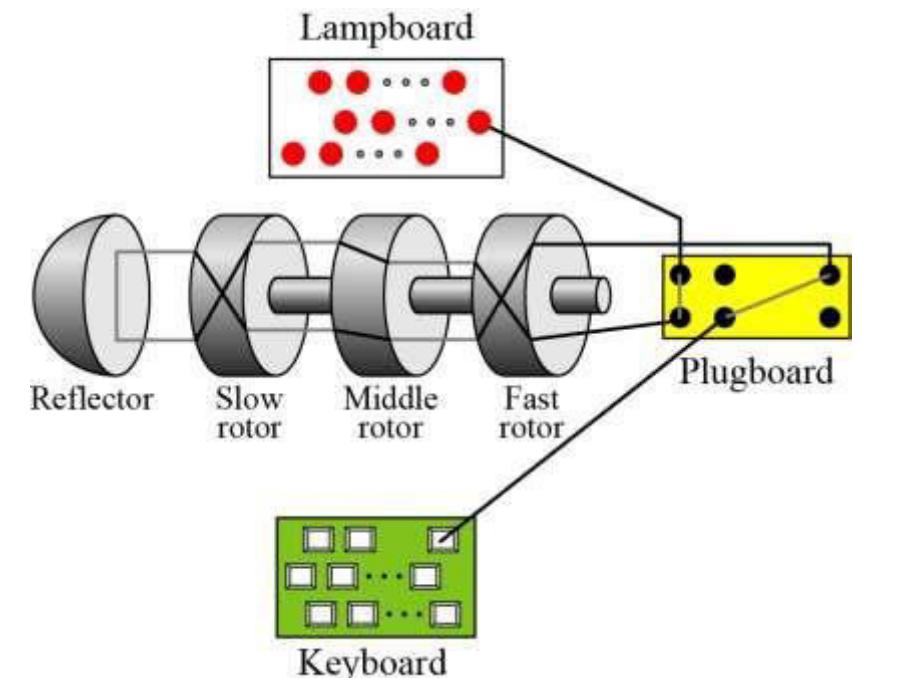
---

- **Keyboard (input)**
- **Lampboard (output)**
- **Rotors (polyalphabetic encryption)**
  - 3 - 4 rotors (depending on Enigma type)
  - 1 reflector
  - 26 contacts at both sides
  - Wired unregularly
  - Turns one step at each letter
  - Turnover is based on ring setting
  - Exchangeable
- **Plugboard (monoalphabetic encryption)**
  - Interchanges letters

# ROTOR CIPHER



A schematic of the Enigma machine



# ROTOR CIPHER

---

To use Engima machine , a code book is published that gives several settings for each day

- a. 3 rotor to be chosen, out of 5 available
- b. The order in which rotor to be installed
- c. Setting for plugboard
- d. A three letter code for the day

# ROTOR CIPHER

---

Procedure for Encrypting message

1. Set starting position of rotor to code of the day. For example code was “HUA”
2. Choose a random 3 letter code such as ACF

Encrypt           **ACFACF**(repeated code) using code from step1

Encrypted code is **OPNABT**

3. Set the starting position to OPN(half of encrypted code)
4. Append encrypted code to message →ACFOPNABT
5. Encrypt the message →ACFOPNABT. Send the encrypted message

# ROTOR CIPHER

---

Procedure for Decrypting message

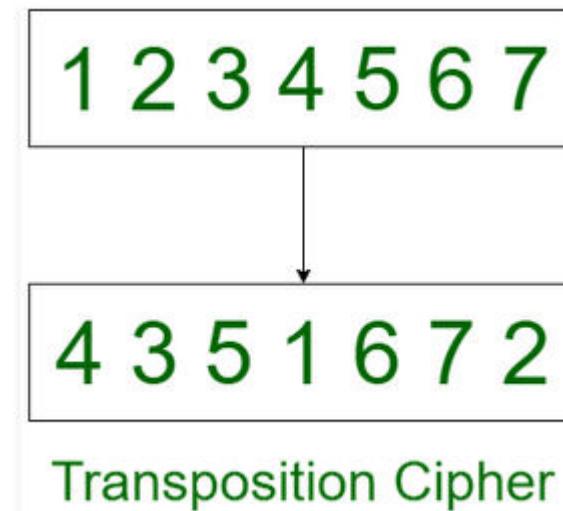
1. Receive the message and separate the first six letters
2. Set the starting position of the rotor to the code of the day
3. Decrypt the first six letter using initial setting in step2
4. Set the position of the rotor to the first half of the decrypted code
5. Decrypt the message (without the first six letter)

# TRANSPOSITION CIPHERS

---

Transposition Cipher rearranges the position of the characters of plain text.

It changes the position of the character but it does not change the identity of the character.



# TRANSPOSITION CIPHERS

---

A transposition cipher does not substitute one symbol for another, instead it changes the location of the symbols.

A transposition cipher reorders symbols.

---

1. Keyless Transposition Ciphers
2. Keyed Transposition Ciphers
3. Combining Two Approaches

# Keyless Transposition Ciphers

---

Simple transposition ciphers, which were used in the past, are keyless.

Two Methods are

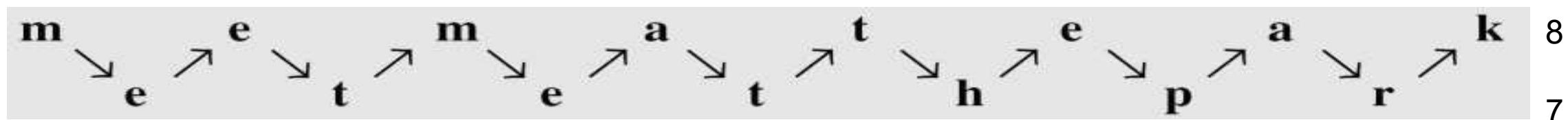
1. Text is written into table column by column and transmit row by row(Rail Fence cipher)
2. Text is written into table row by row and transmit column by column

# Keyless Transposition Ciphers

## Rail fence cipher

PT is arranged in two line as a zigzag pattern(Column by column)

For example: “Meet me at the park”



The ciphertext is created reading the pattern row by row.

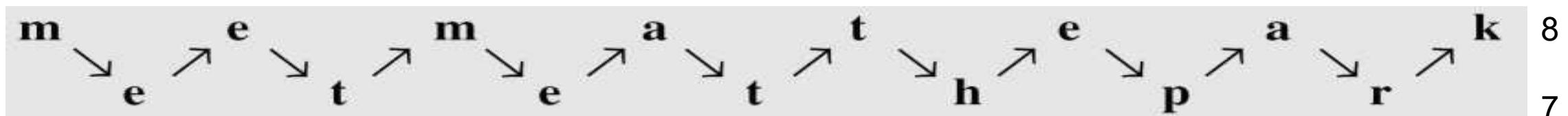
Ciphertext “**MEMATEAKETETHPR**”.

# Keyless Transposition Ciphers

## Rail fence cipher

Ciphertext “**MEMATEAKETETHPR**”

Receiver divides into half (first half form the first row and second half second row) and reads in zigzag



The ciphertext is created reading the pattern row by row.

# TRANSPOSITION CIPHERS

---

Alice and Bob can agree on the number of columns.

Alice writes the plaintext, row by row, in a table of four columns.

For example: “Meet me at the park”

<b>m</b>	<b>e</b>	<b>e</b>	<b>t</b>
<b>m</b>	<b>e</b>	<b>a</b>	<b>t</b>
<b>t</b>	<b>h</b>	<b>e</b>	<b>p</b>
<b>a</b>	<b>r</b>	<b>k</b>	

Ciphertext “**MMTAEEHREAEKTP**”.

Meet me at the park

MMTAEEHREAEKTP

---

The following shows the permutation of each character in the plaintext into the ciphertext based on the positions.

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
01	05	09	13	02	06	10	13	03	07	11	15	04	08	12

The second character in the plaintext has moved to the fifth position in the ciphertext; the third character has moved to the ninth position; and so on.

Although the characters are permuted, there is a pattern in the permutation: (01, 05, 09, 13), (02, 06, 10, 13), (03, 07, 11, 15), and (08, 12).

In each section, the difference between the two adjacent numbers is 4.

# Keyed Transposition Ciphers

---

- Divide the plaintext into groups of predetermined size, called blocks, and then use a key to permute the characters in each block separately.
- Alice needs to send the message “**Enemy attacks tonight**” to Bob.
- Alice and Bob agrees with block size =5



e n e m y   a t t a c k s   t o n i g h t z

- The key used for encryption and decryption is a permutation key, which shows how the character are permuted.

# Keyed Transposition Ciphers

---

- Divide the plaintext into groups of predetermined size, called blocks, and then use a key to permute the characters in each block separately.
- Alice needs to send the message “**Enemy attacks tonight**” to Bob.
- Alice and Bob agrees with block size =5



e n e m y      a t t a c k s      t o n i g h t z

# Keyed Transposition Ciphers

e n e m y      a t t a c      k s t o n      i g h t z

Encryption ↓

3	1	4	5	2
1	2	3	4	5

↑ Decryption

3rd bala 1st me jaye ga  
1st bala 2nd me jaye ga  
4th bala 3rd me jaye ga  
5th bala 4th me jaye ga  
2nd bala 5th me jaye ga

The permutation yields

E E M Y N      T A A C T      T K O N S      H I T Z G

1st me 3rd bala aaye ga  
2nd me 1st bala aaye ga  
3rd me 4th bala aaye ga  
4th me 5th bala aaye ga  
5th me 2nd bala aaye ga

# Keyed Transposition Ciphers

Encryption ↓

3	1	4	5	2
1	2	3	4	5

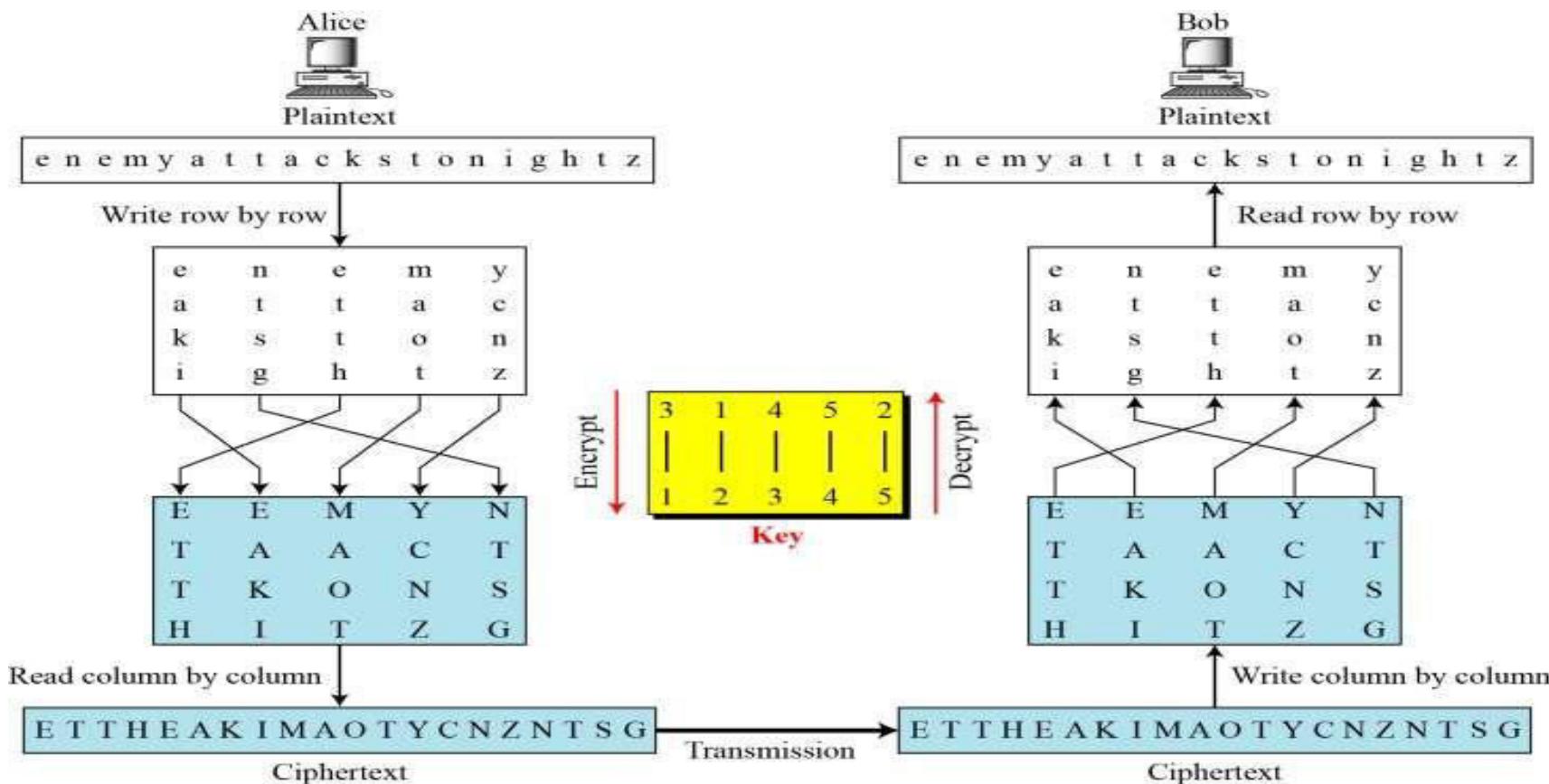
↑ Decryption

The permutation yields

E E M Y N      T A A C T      T K O N S      H I T Z G

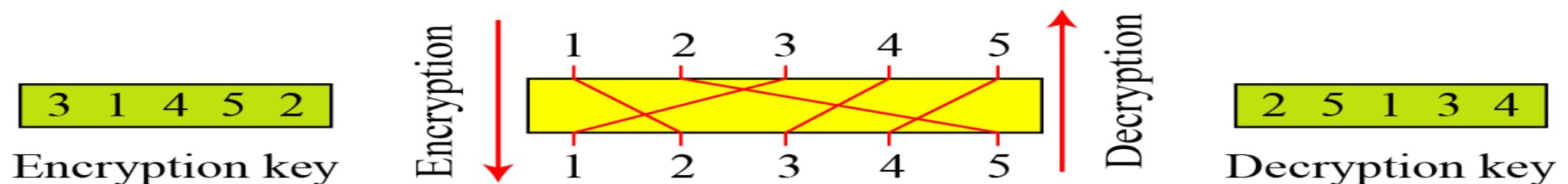
e n e m y      a t t a c      k s t o n      i g h t z

# Combining Two Approaches



# Keys

A single key was used in two directions for the column exchange: downward for encryption, upward for decryption. It is customary to create two keys.



Encryption/decryption keys in transpositional ciphers

Encryption ↓

3	1	4	5	2
1	2	3	4	5

↑ Decryption

# STREAM AND BLOCK CIPHERS

---

The literature divides the symmetric ciphers into two broad categories:

- Stream ciphers
- Block ciphers

# STREAM CIPHERS

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

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

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

$$K = (k_1, k_2, k_3, \dots)$$

$$C_1 = E_{k1}(P_1)$$

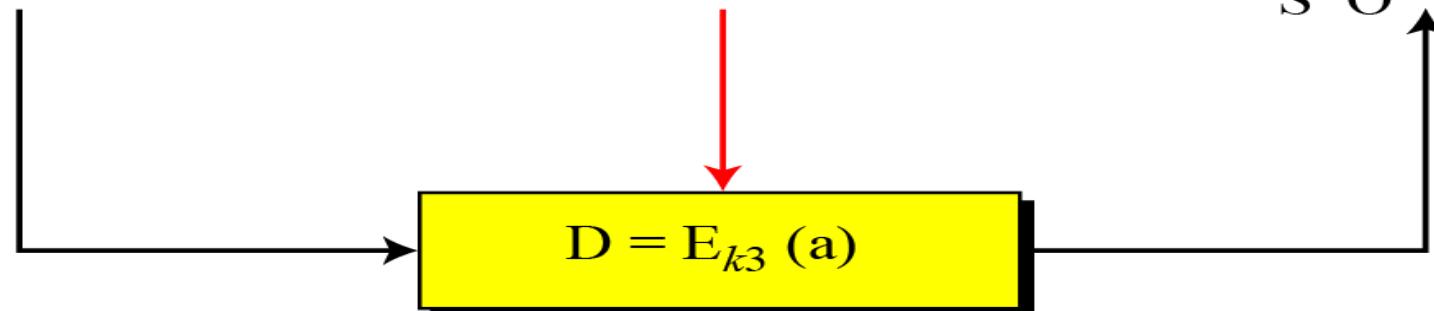
$$C_2 = E_{k2}(P_2)$$

$$C_3 = E_{k3}(P_3) \dots$$

Plaintext  
p l a i n

$$K = (k_1, k_2, k_3, k_4, k_5)$$

Ciphertext  
s o

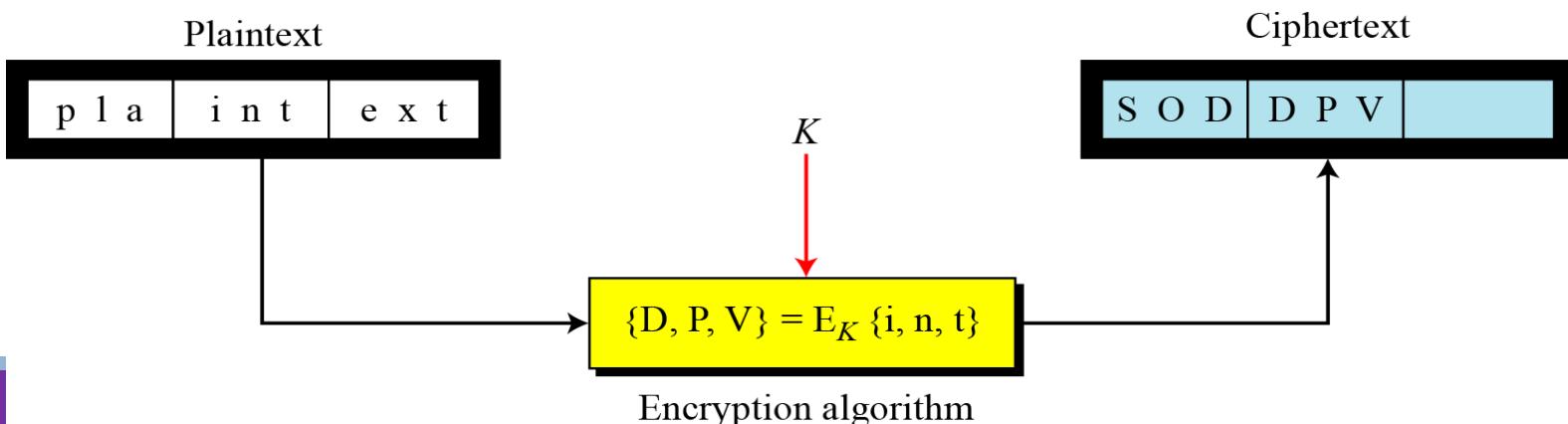


Encryption algorithm

# 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.



Block Cipher	Stream Cipher
Processing or encoding of the plain text is done as a fixed length block one by one. A block for example could be 64 or 128 bits in size.	Processing or encoding of plain text is done bit by bit. The block size here is simply one bit.
The same key is used to encrypt each of the blocks	A different key is used to encrypt each of the bits.
A Pad added to short length blocks	Bits are processed one by one in as in a chain
Uses Symmetric Encryption and is NOT used in asymmetric encryption	High speed and low hardware complexity
Confusion factor: The key to the cipher text relationship could be really very complicated.	Key is often combined with an initialization vector
Diffusion Factor: output depends on the input in a very complex method.	Long period with no repetition
Most block ciphers are based on Feistel cipher in structure	Statistically random
Looks more like an extremely large substitution and Using the idea of a product cipher	Depends on a large key and Large liner complexity
More secure in most cases	Equally secure if properly designed
Usually more complex and slower in operation	Usually very simple and much faster
<b>Examples of Block Cipher are:</b> Lucifer / DES, IDEA, RC5, Blowfish etc.	<b>Examples of Stream Cipher are:</b> FISH, RC4, ISAAC, SEAL, SNOW etc.

# UNIT 2

---

## Data Encryption Standard (DES): (Chapter 6)

- Introduction
- DES Structure
- DES Analysis
- Security of DES

# History

---

- In 1973, NIST published a request for proposals for a national symmetric-key cryptosystem.
- A proposal from IBM, a modification of a project called Lucifer, was accepted as DES.
- DES was published in the Federal Register in March 1975 as a draft of the Federal Information Processing Standard (FIPS).

# DES Overview

The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).

DES is a block cipher. The block size is 64-bit.

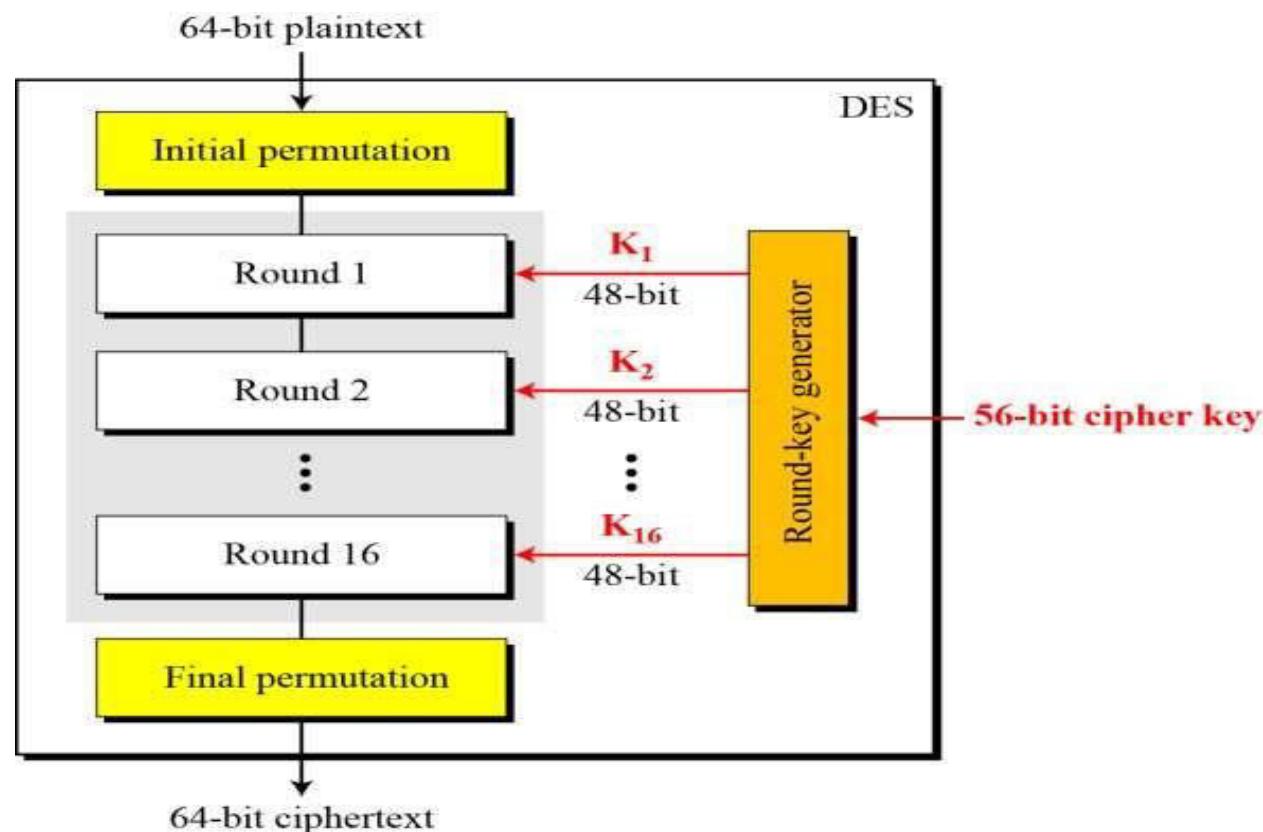


# DES Structure

DES is an implementation of a Feistel Cipher.

It uses **16 round Feistel structure**.

Key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm



# DES Structure

Two properties that a good cryptosystem should have to hinder statistical analysis:

1. **Diffusion** -If a single symbol in the plaintext is changed, several or all symbol in the ciphertext will also be changed
2. **Confusion** - If a single bit in the key is changed, most or all bits in the ciphertext will also be changed

**Confusion = Substitution**

a --> b

[Caesar Cipher](#)

**Diffusion = Transposition or Permutation**

abcd --> dacb

DES

# DES Structure

DES uses a **56-bit key**.

Actually, the initial key consists of **64 bits**.

However, before the DES process even starts, every 8th bit of the key is discarded to produce a **56-bit key**. That is bit positions 8, 16, 24, 32, 40, 48, 56, and 64 are discarded.

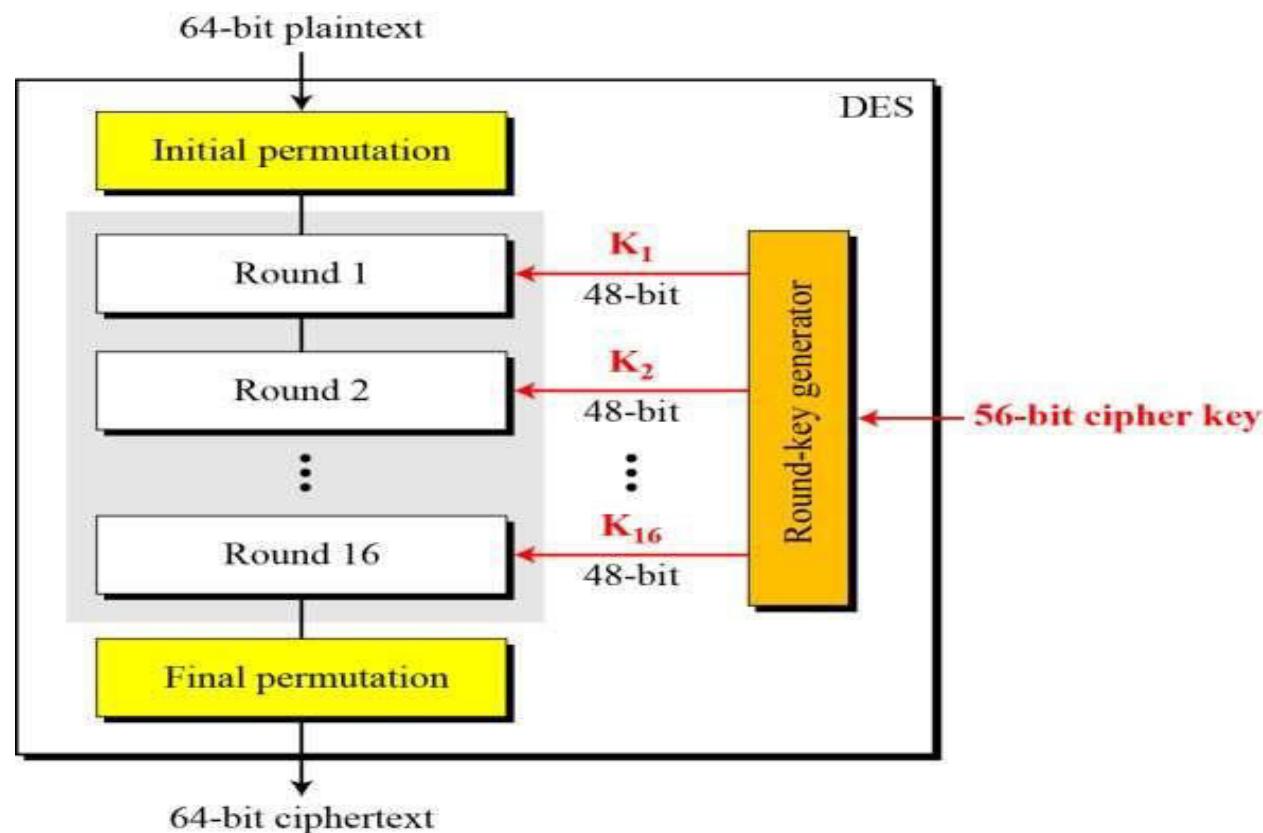
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64

1	2	3	4	5	6	7
9	10	11	12	13	14	15
17	18	19	20	21	22	23
25	26	27	28	29	30	31
33	34	35	36	37	38	39
41	42	43	44	45	46	47
49	50	51	52	53	54	55
57	58	59	60	61	62	63

# DES Structure

DES is based on the Feistel Cipher

- Initial and final permutation
- Round function
- Round Key Generator



# DES Structure

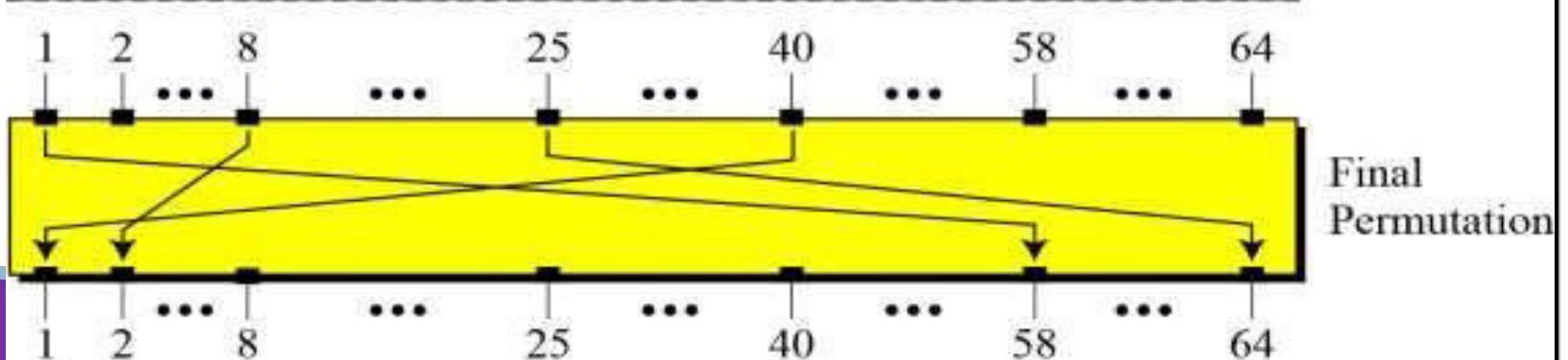
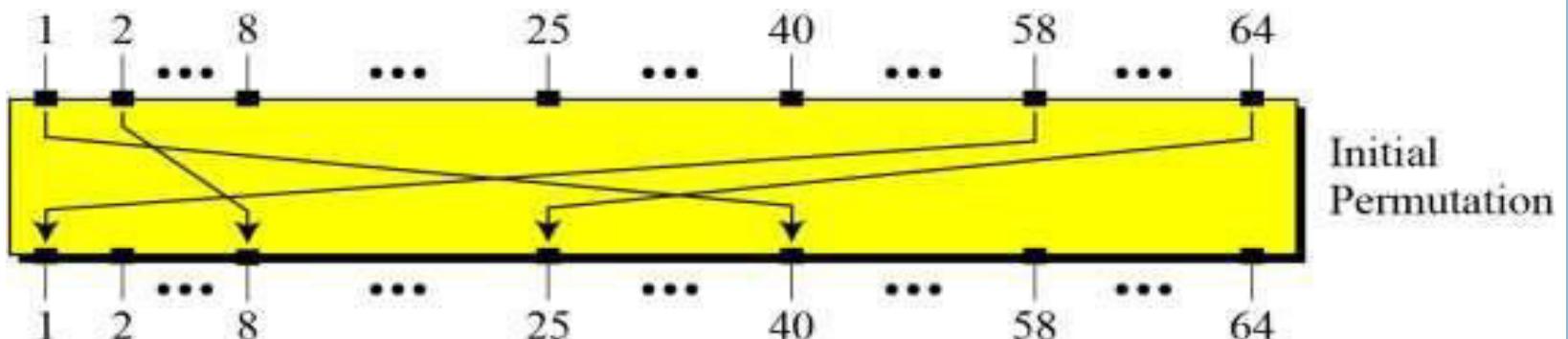
---

## Initial and final permutation

- The initial permutation (IP) happens only once and it happens before the first round.
- Transposition in IP is done, Both are keyless and predetermined.
- IP replaces the first bit of the original plain text block with the 58th bit of the original plain text, the second bit with the 50th bit of the original plain text block, and so on.

# DES Structure

Initial and final permutation step



Permutation Box

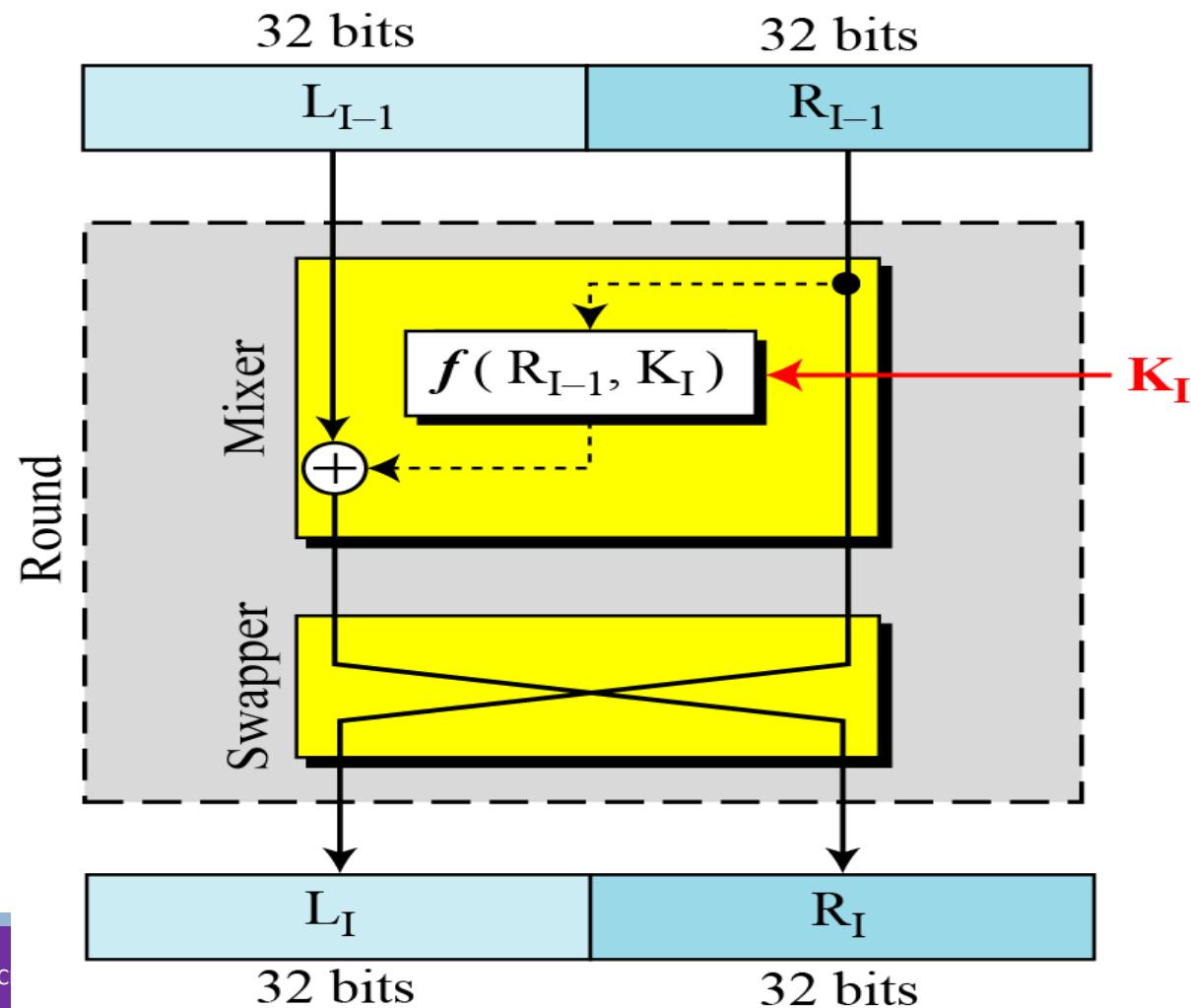
Initial Permutation									
58	50	42	34	26	18	10	02		
60	52	44	36	28	20	12	04		
62	54	46	38	30	22	14	06		
64	56	48	40	32	24	16	08		
57	49	41	33	25	17	09	01		
59	51	43	35	27	19	11	03		
61	53	45	37	29	21	13	05		
63	55	47	39	31	23	15	07		

Final Permutation									
40	08	48	16	56	24	64	32		
39	07	47	15	55	23	63	31		
38	06	46	14	54	22	62	30		
37	05	45	13	53	21	61	29		
36	04	44	12	52	20	60	28		
35	03	43	11	51	19	59	27		
34	02	42	10	50	18	58	26		
33	01	41	09	49	17	57	25		

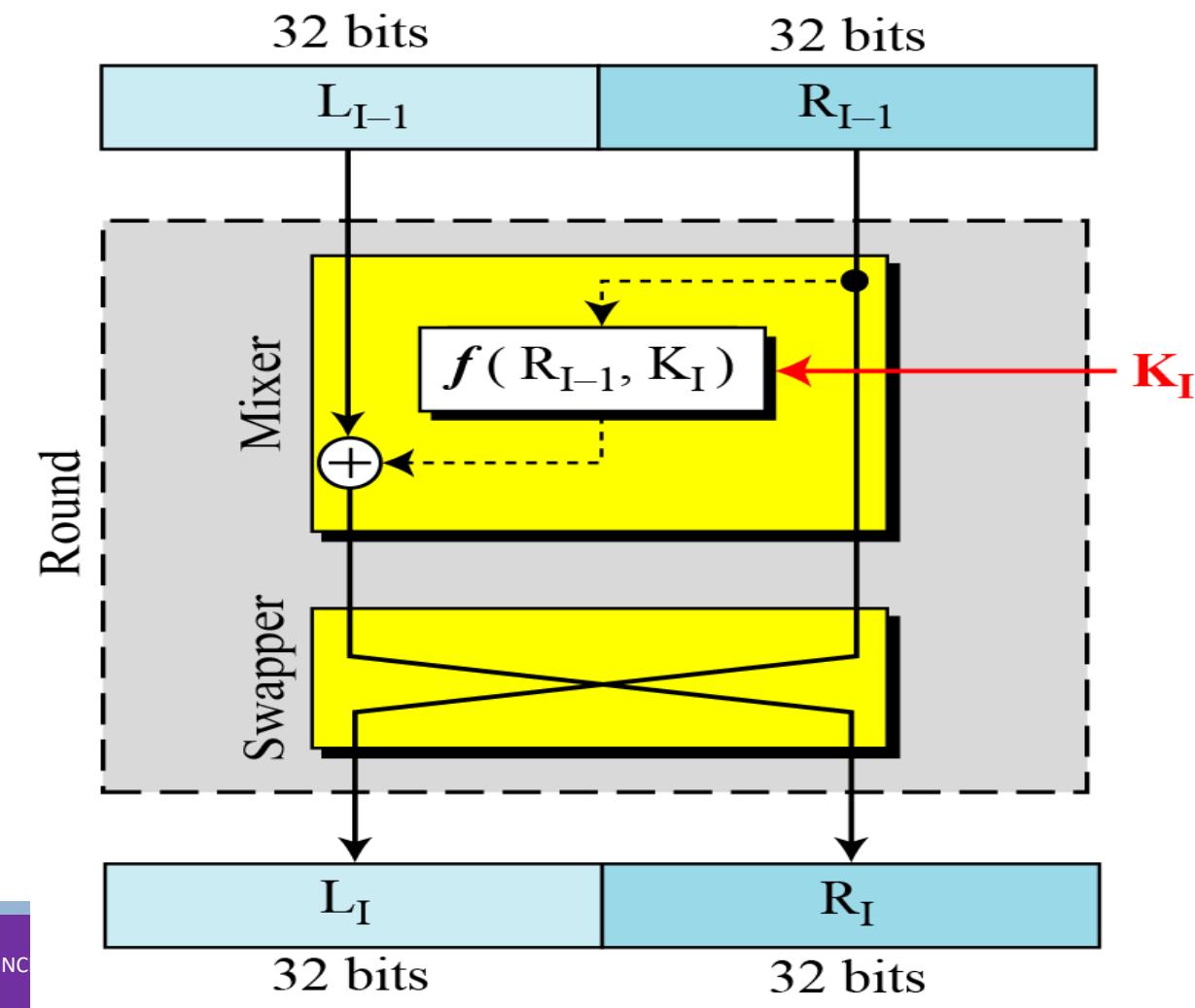
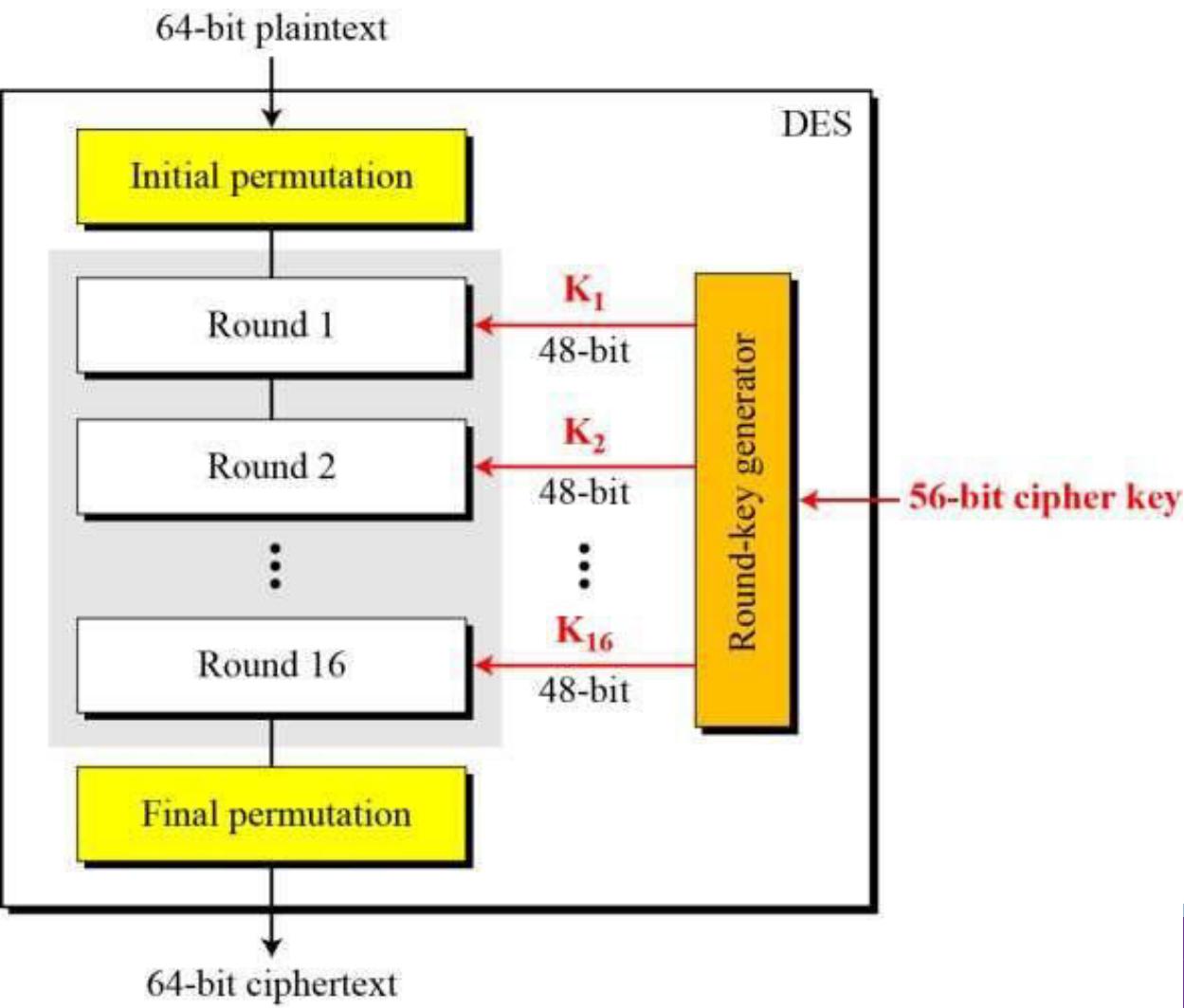
# DES Structure

## Round function

- DES uses 16 rounds
- Each round is a Feistel cipher is the DES **function  $f$ .**



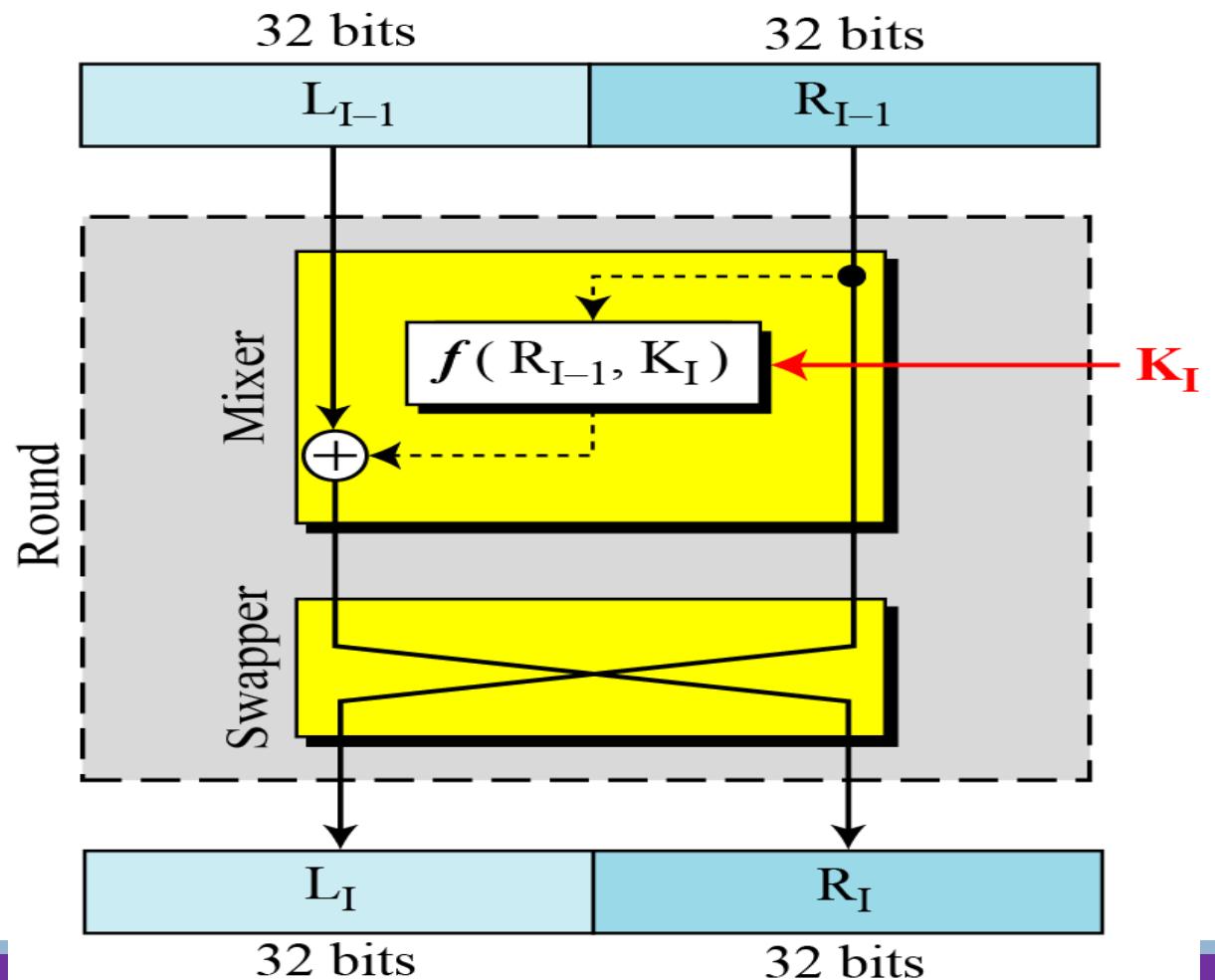
# DES Structure



# DES Structure

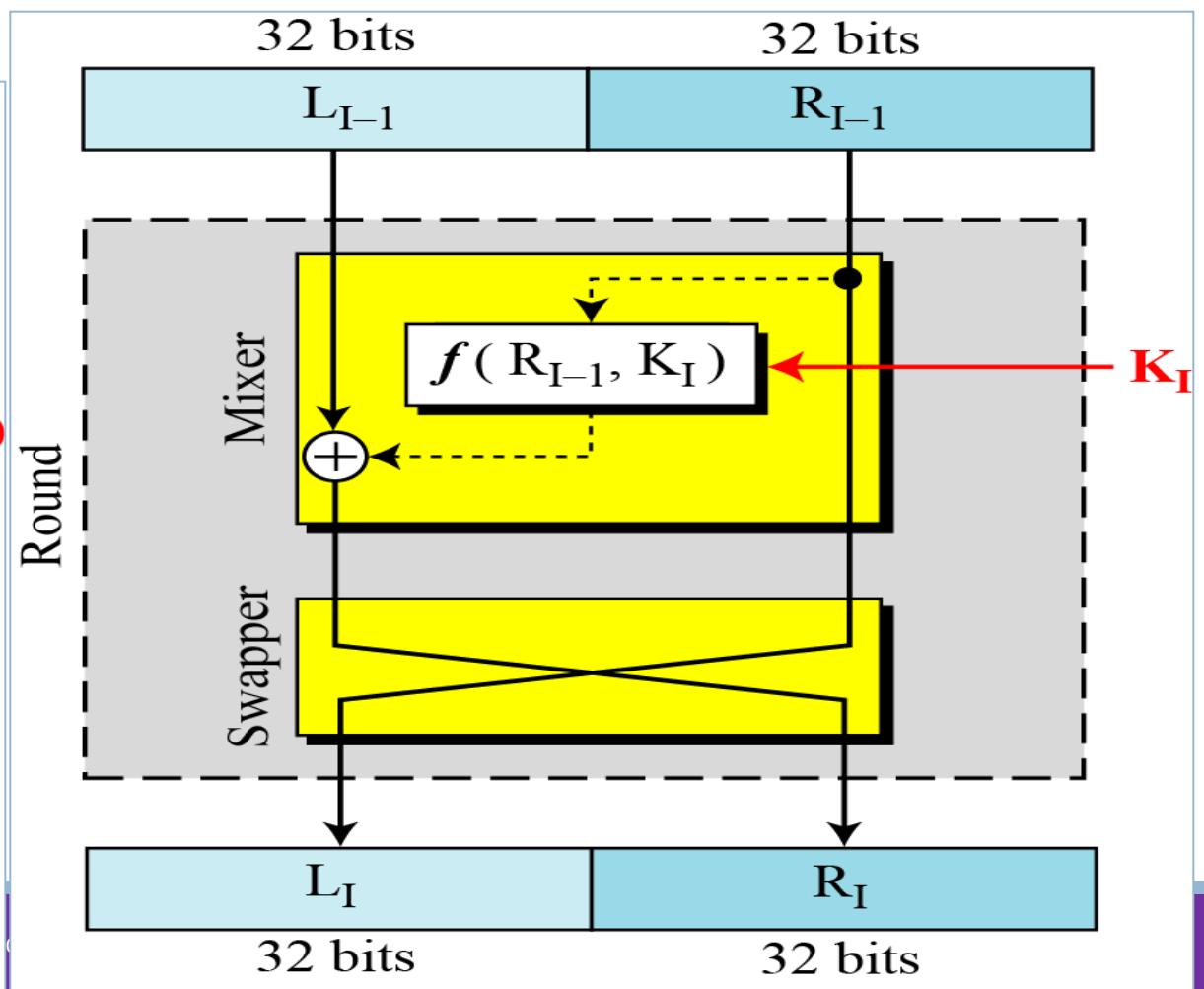
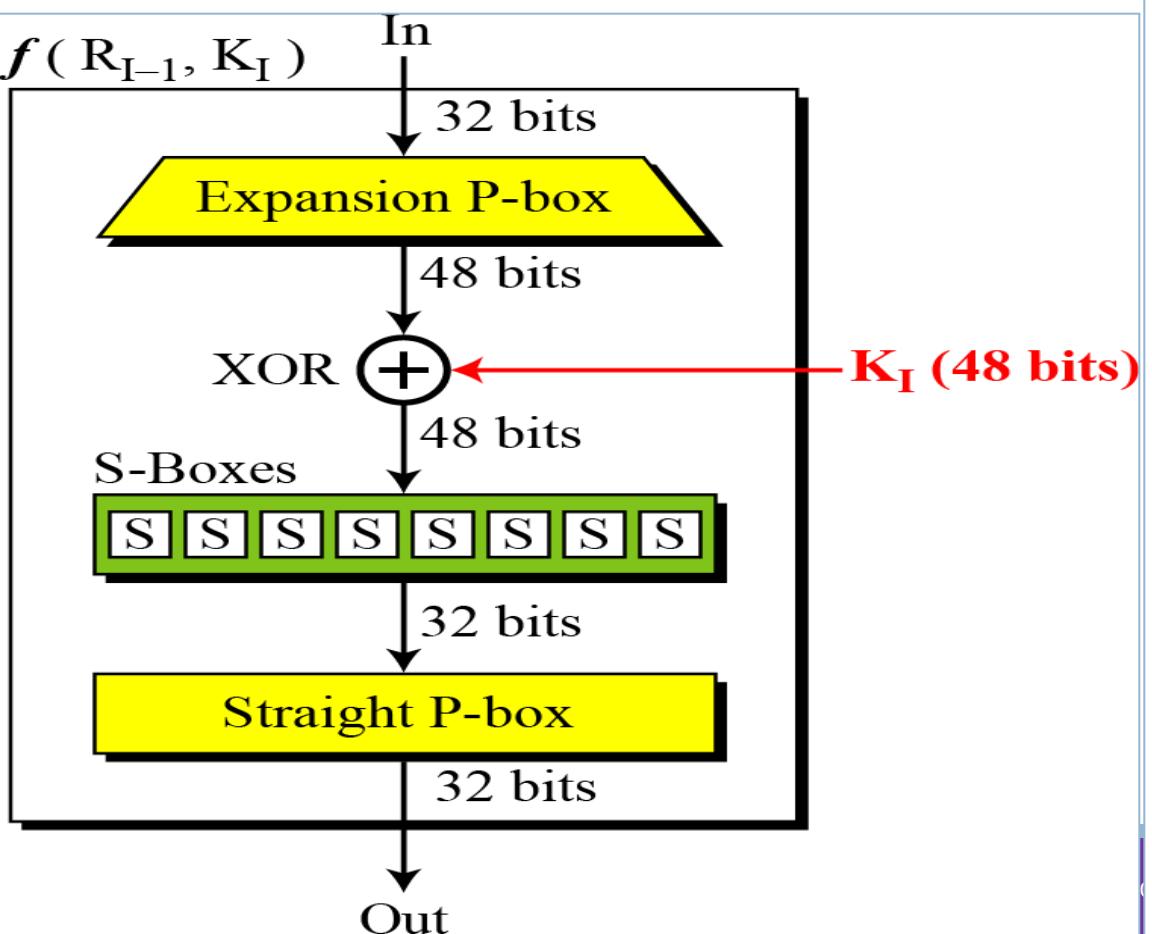
## Round function

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



# DES Structure

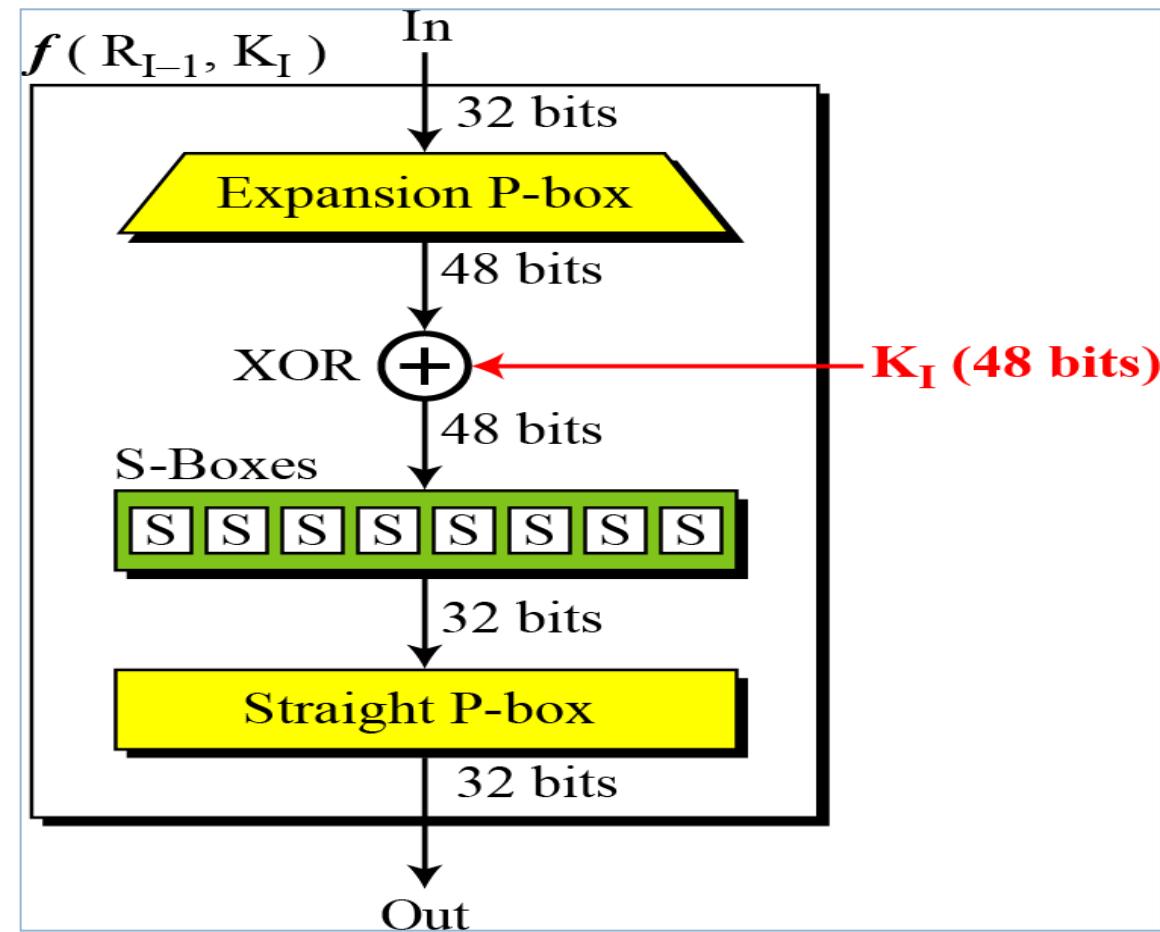
## DES function



# DES Structure

## DES function

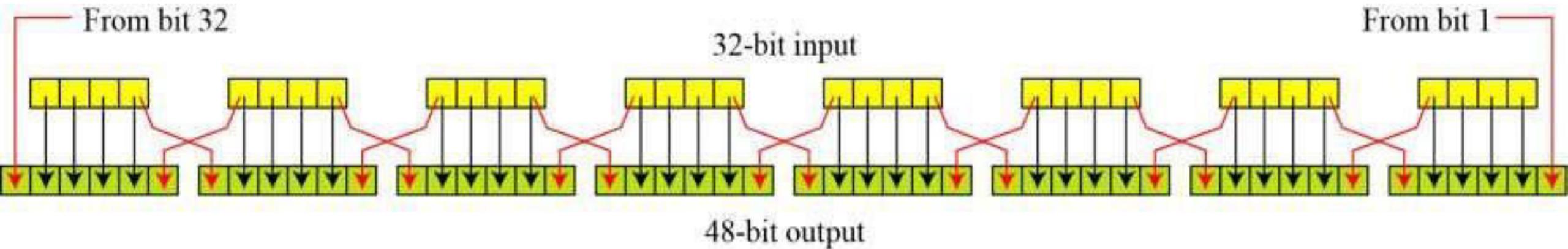
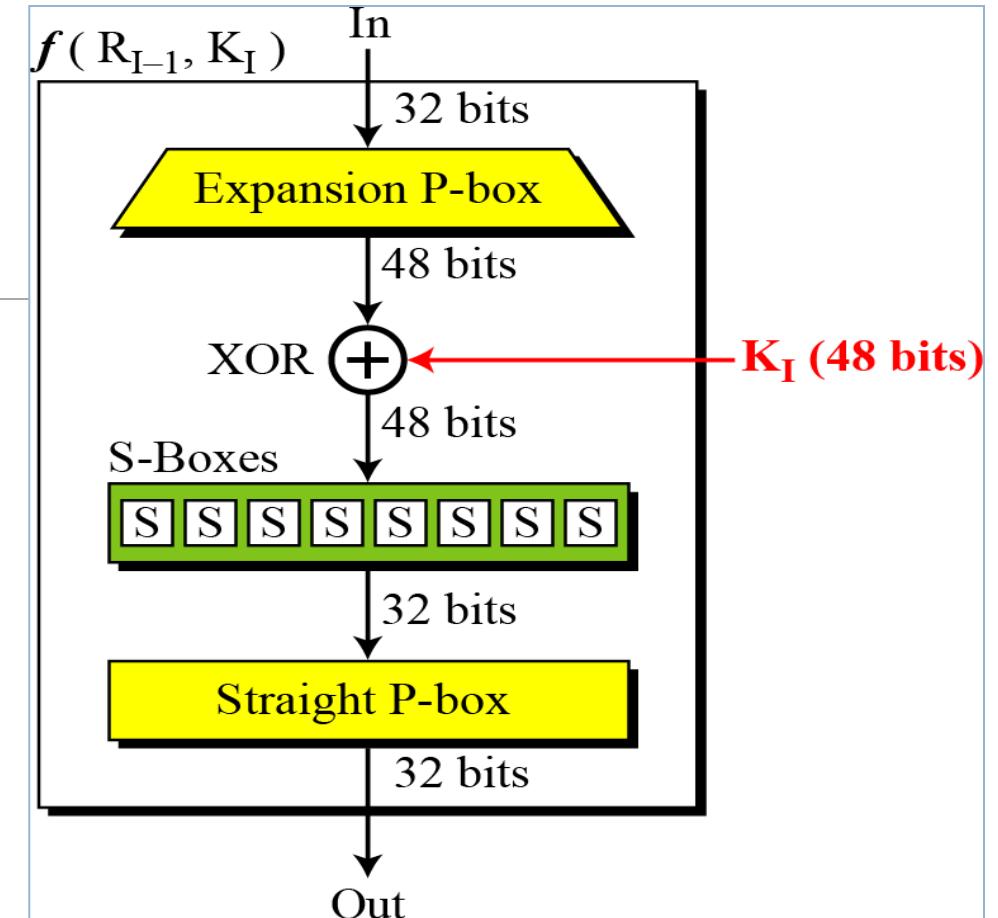
- Expansion P-Box
- Whitener (XOR)
- S-Boxes
- Straight P-Box



# DES Structure

## DES function - Expansion P-Box

- RPT is expanded from 32 bits to 48 bits.
- Bits are permuted as well hence called expansion permutation.
- This happens as the 32-bit RPT is divided into 8 blocks, with each block consisting of 4 bits.
- Then, each 4-bit block of the previous step is then expanded to a corresponding 6-bit block, i.e., per 4-bit block, 2 more bits are added.

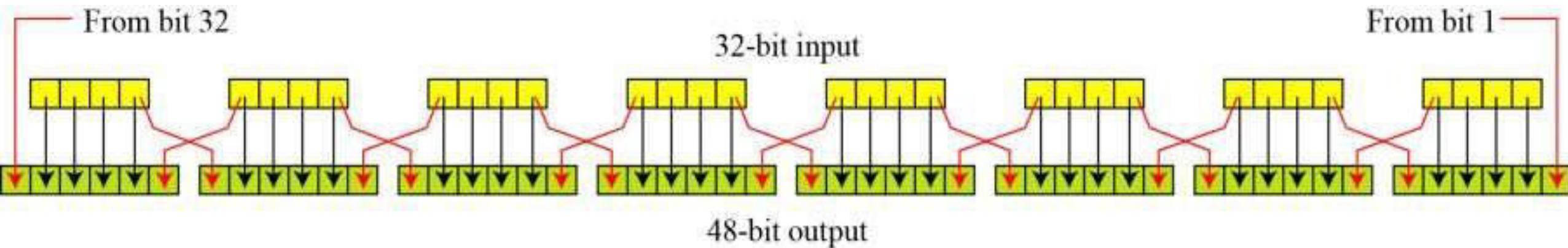


# DES Structure

## DES function - Expansion P-Box

- RPT is expanded from 32 bits to 48 bits.
- Bits are permuted as well hence called expansion permutation.
- This happens as the 32-bit RPT is divided into 8 blocks, with each block consisting of 4 bits.
- Then, each 4-bit block of the previous step is then expanded to a corresponding 6-bit block, i.e., per 4-bit block, 2 more bits are added.

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

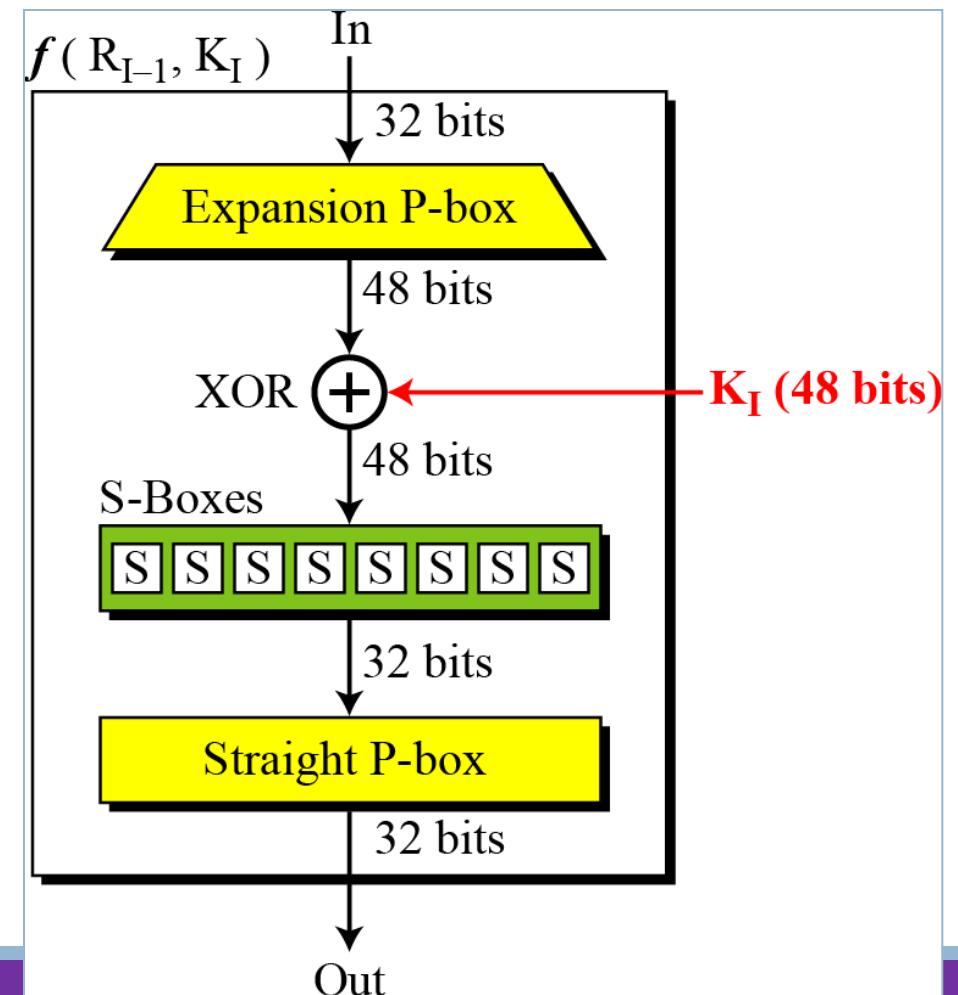


# DES Structure

## DES function - Whitener (XOR)

After the expansion permutation, DES does XOR operation on the expanded right section and the round key.

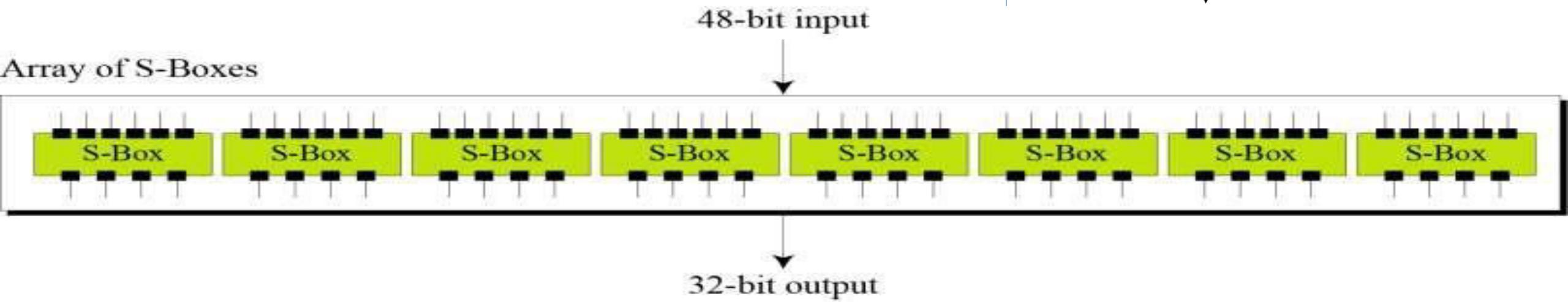
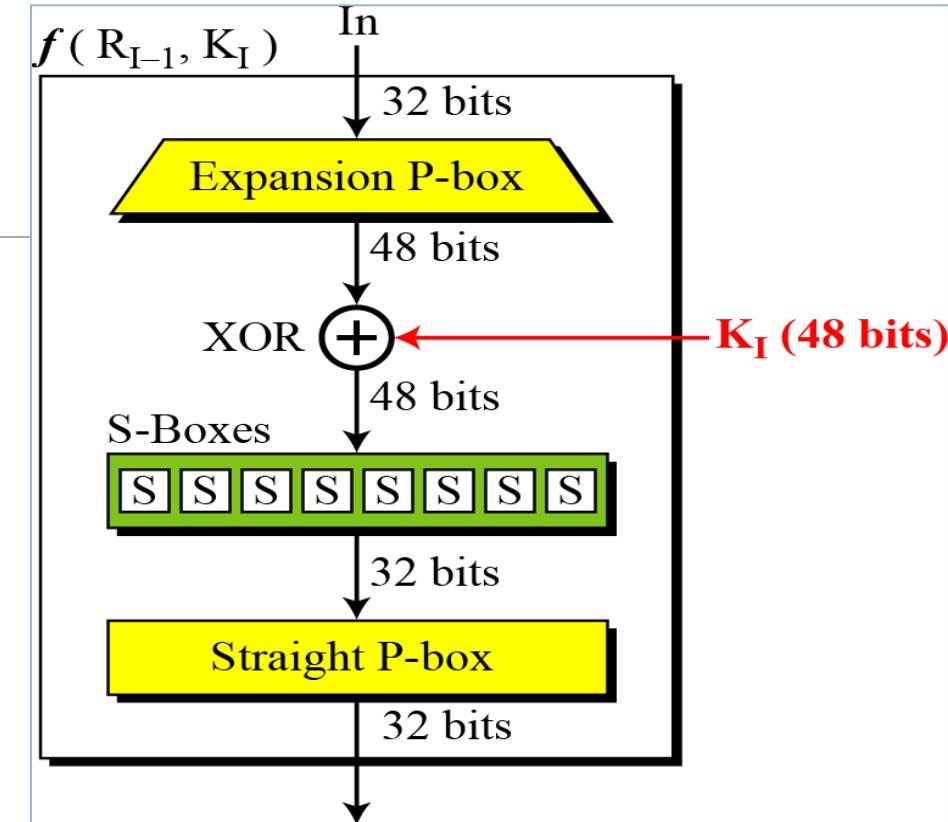
The round key is used only in this operation.



# DES Structure

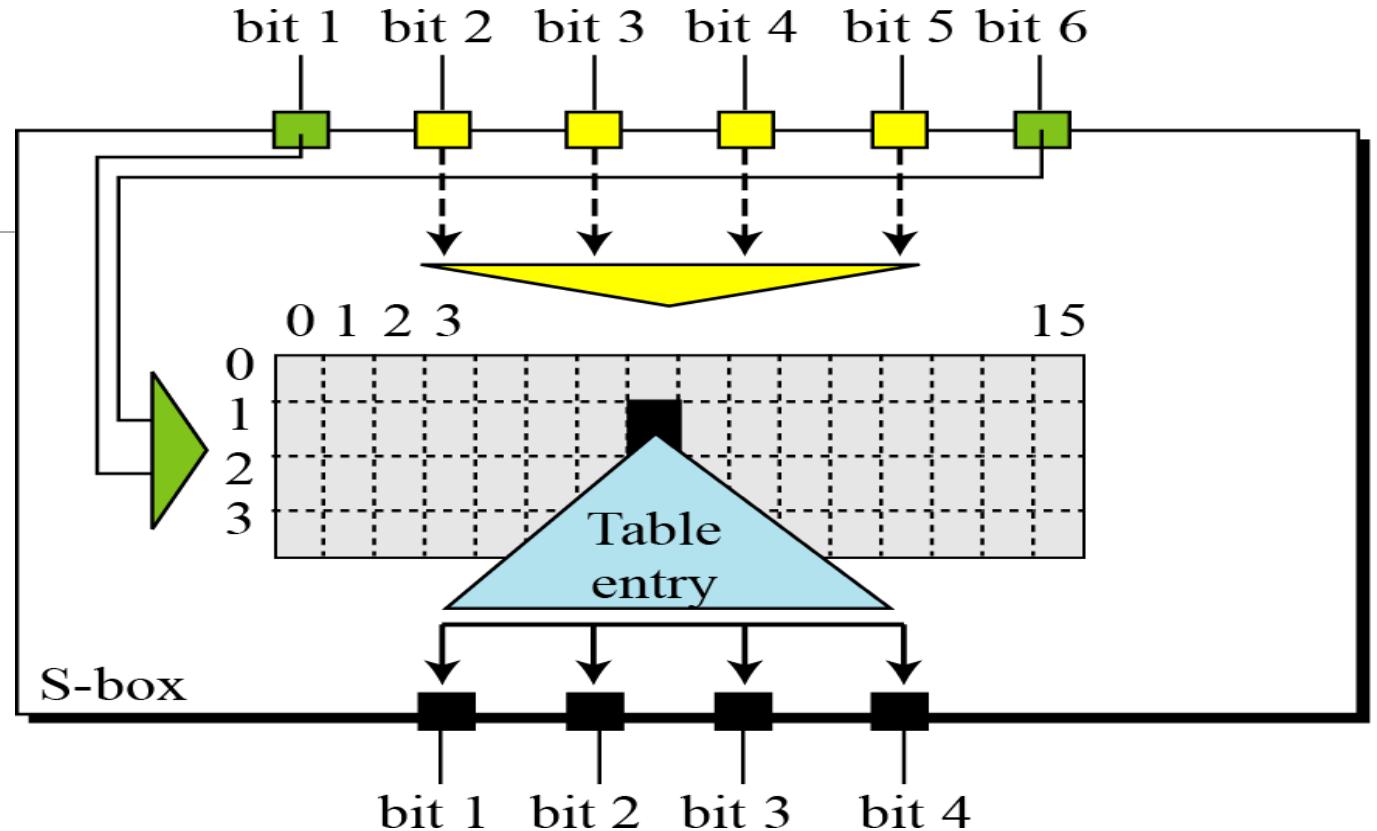
## DES function - S-Boxes

- The S-boxes carry out the real mixing (confusion).
- DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.

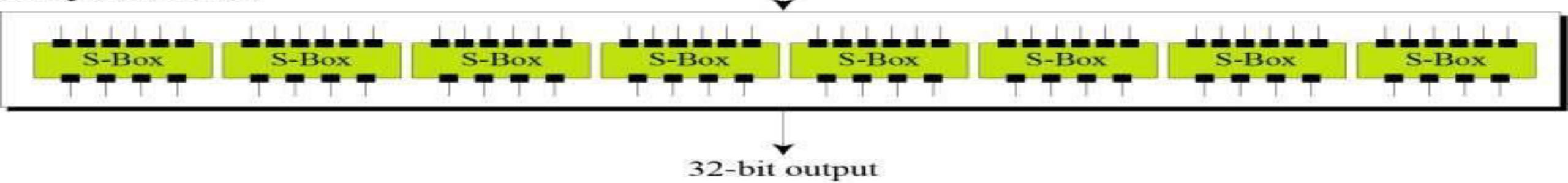


# DES Structure

## DES function - S-Boxes



Array of S-Boxes



# DES Structure

## DES function - S-Boxes

B = 011011 the first bit is "0" and the last bit "1" giving **01** as the row. This is row 1.

The middle four bits are "**1101**". This is the binary equivalent of decimal 13, so the column is column number 13.

In row 1, column 13 appears 5. This determines the output; 5 is binary 0101, so that the output is 0101.

Hence **S<sub>1</sub>**(011011) = 0101.

*S-box 1*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

# DES Structure

## DES function - S-Boxes

S-box 2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	01	08	14	06	11	03	04	09	07	02	13	12	00	05	10
1	03	13	04	07	15	02	08	14	12	00	01	10	06	09	11	05
2	00	14	07	11	10	04	13	01	05	08	12	06	09	03	02	15
3	13	08	10	01	03	15	04	02	11	06	07	12	00	05	14	09

S-box 3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	00	09	14	06	03	15	05	01	13	12	07	11	04	02	08
1	13	07	00	09	03	04	06	10	02	08	05	14	12	11	15	01
2	13	06	04	09	08	15	03	00	11	01	02	12	05	10	14	07
3	01	10	13	00	06	09	08	07	04	15	14	03	11	05	02	12

S-box 4

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	07	13	14	03	00	6	09	10	1	02	08	05	11	12	04	15
1	13	08	11	05	06	15	00	03	04	07	02	12	01	10	14	09
2	10	06	09	00	12	11	07	13	15	01	03	14	05	02	08	04
3	03	15	00	06	10	01	13	08	09	04	05	11	12	07	02	14

# DES Structure

## DES function - S-Boxes

S-box 6

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	01	10	15	09	02	06	08	00	13	03	04	14	07	05	11
1	10	15	04	02	07	12	09	05	06	01	13	14	00	11	03	08
2	09	14	15	05	02	08	12	03	07	00	04	10	01	13	11	06
3	04	03	02	12	09	05	15	10	11	14	01	07	10	00	08	13

S-box 7

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	00	08	13	03	12	09	07	05	10	06	01
1	13	00	11	07	04	09	01	10	14	03	05	12	02	15	08	06
2	01	04	11	13	12	03	07	14	10	15	06	08	00	05	09	02
3	06	11	13	08	01	04	10	07	09	05	00	15	14	02	03	12

S-box 8

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	02	08	04	06	15	11	01	10	09	03	14	05	00	12	07
1	01	15	13	08	10	03	07	04	12	05	06	11	10	14	09	02
2	07	11	04	01	09	12	14	02	00	06	10	10	15	03	05	08
3	02	01	14	07	04	10	8	13	15	12	09	09	03	05	06	11

# DES Structure

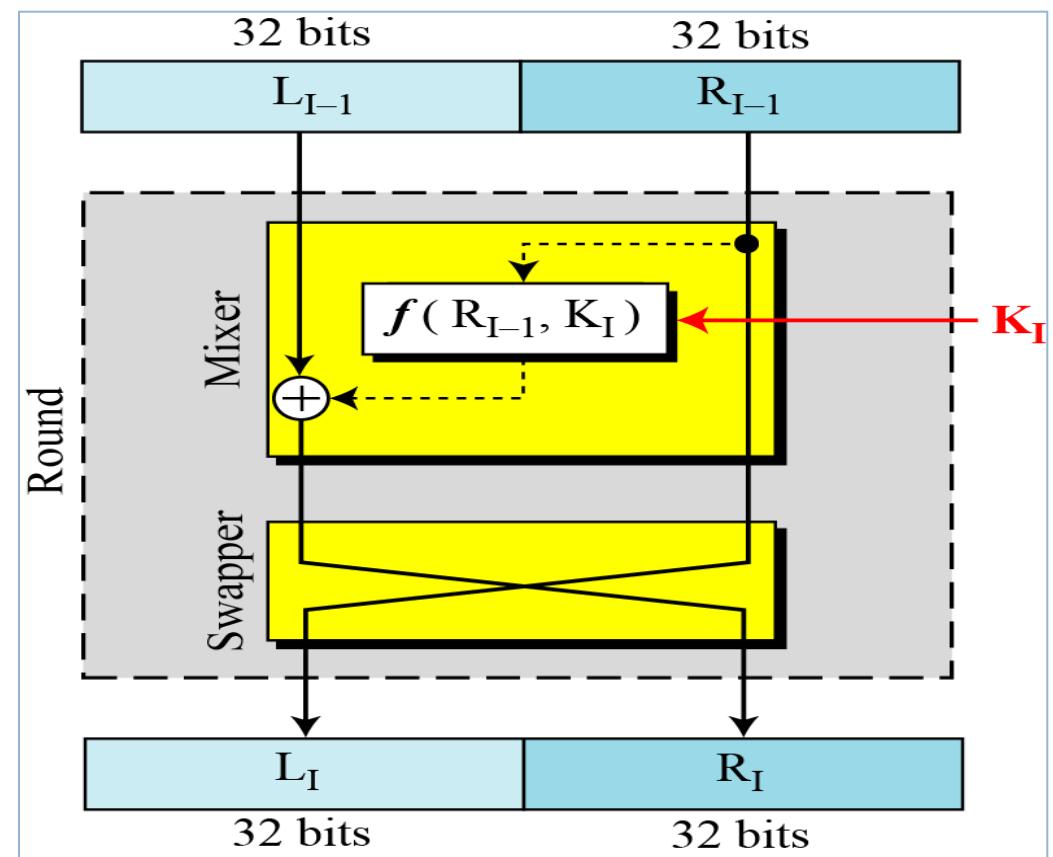
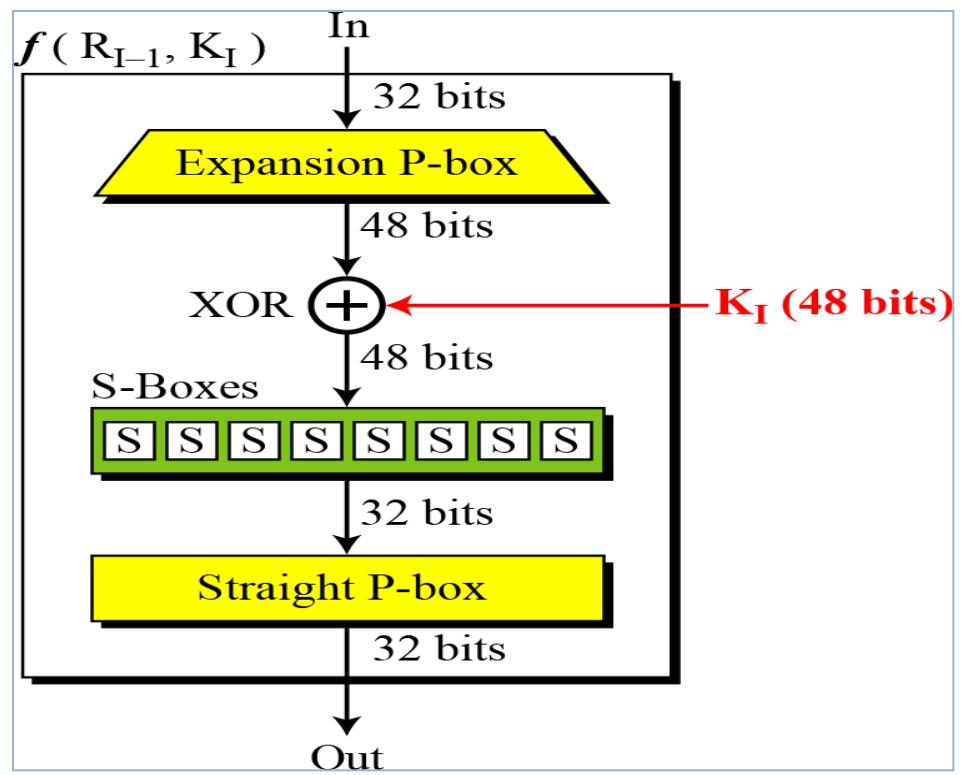
---

## **DES function - Straight P-Box**

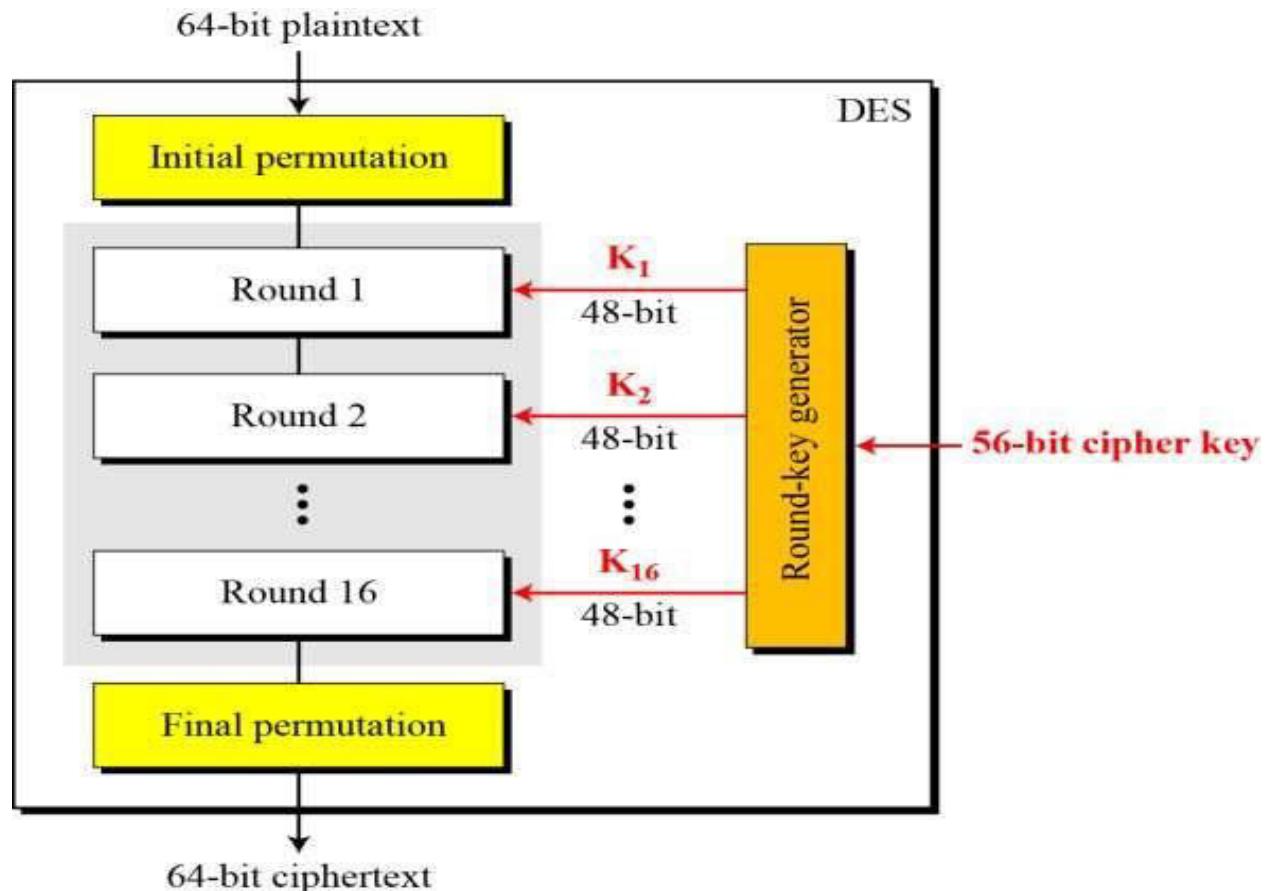
The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration:

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

# DES Structure

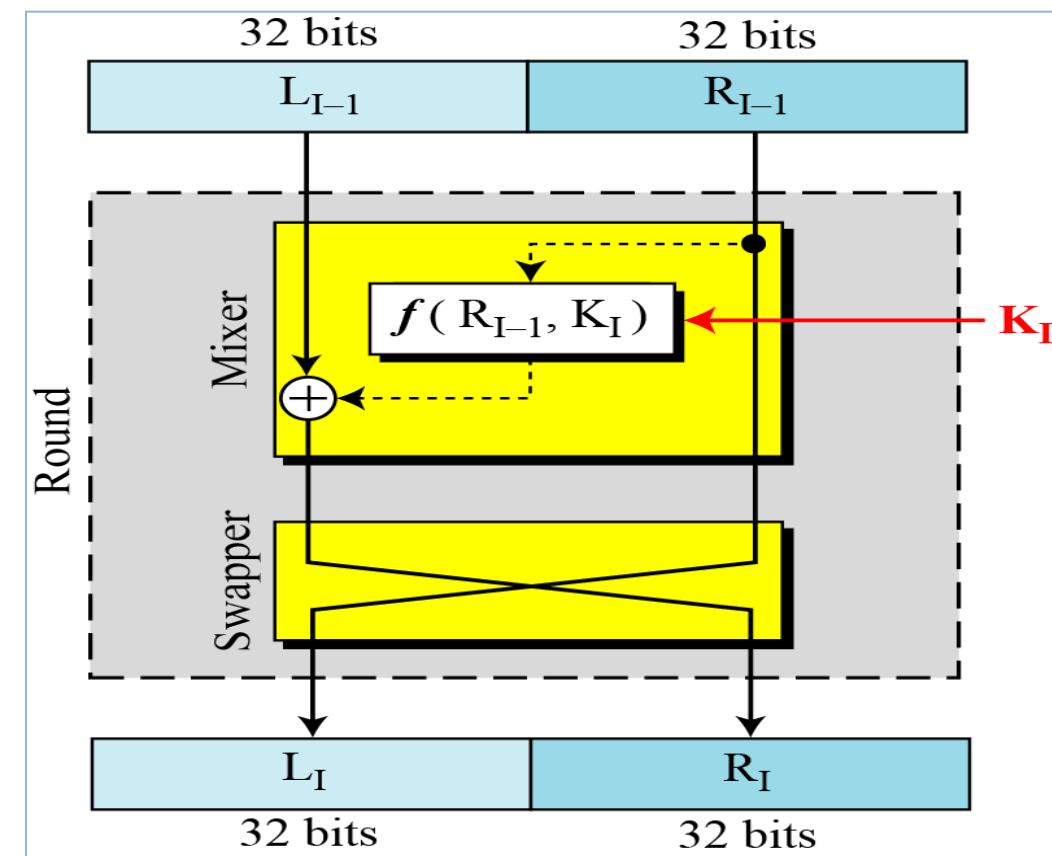


# DES Structure



# Cipher and Reverse Cipher

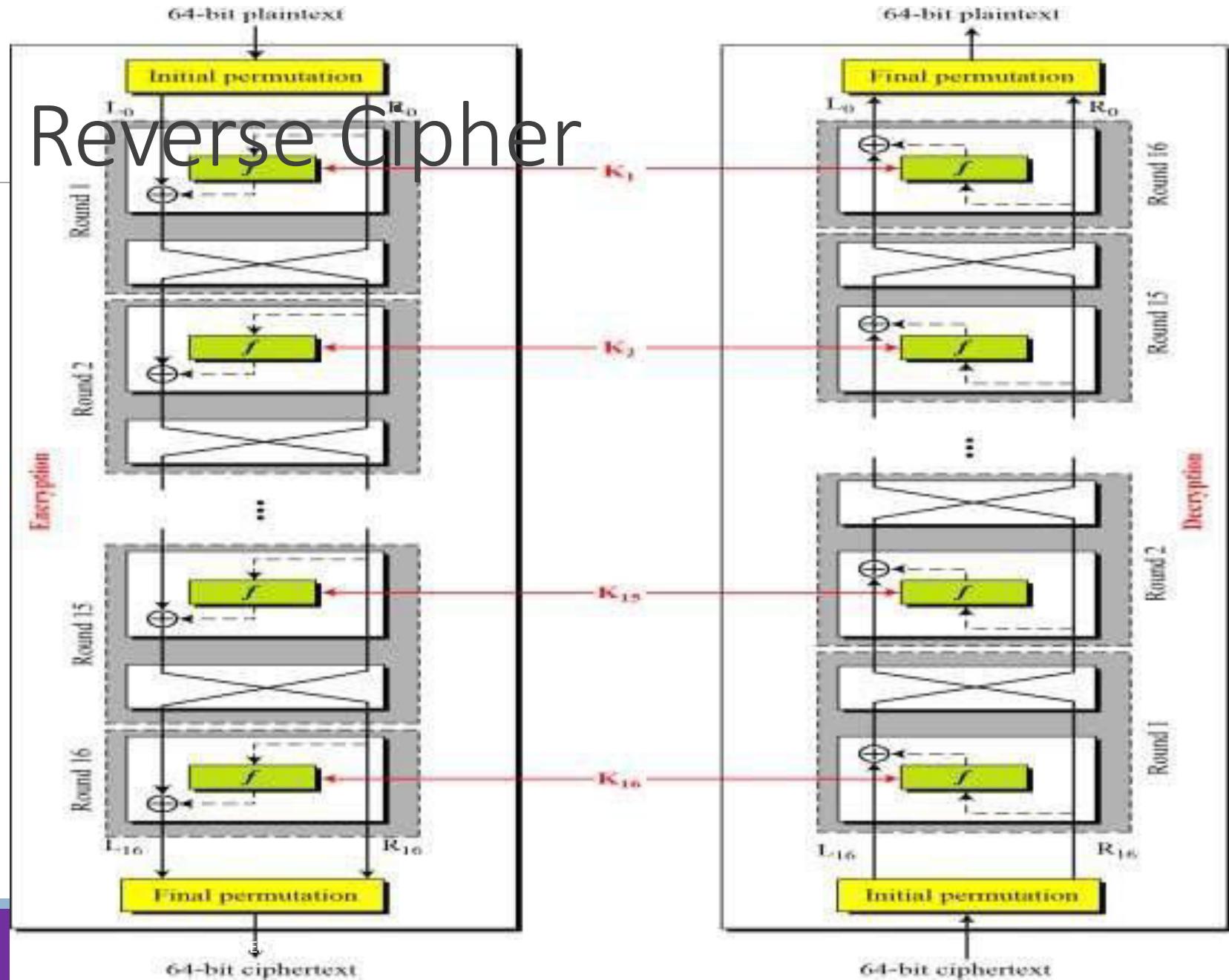
- Using Mixer and Swapper, we can create the cipher and reverse cipher, each having 16 rounds.
- Cipher is used at encryption site
- Reverse cipher is used at decryption site
- Whole idea is to make cipher and reverse cipher similar



# Cipher and Reverse Cipher

## First Approach

Make the last round (round 16) different from others, it has only mixer and no swapper



# Pseudocode for cipher – First approach

```
Cipher (plainBlock[64], RoundKeys[16, 48], cipherBlock[64])
{
    permute (64, 64, plainBlock, inBlock, InitialPermutationTable)
    split (64, 32, inBlock, leftBlock, rightBlock)
    for (round = 1 to 16)
    {
        mixer (leftBlock, rightBlock, RoundKeys[round])
        if (round!=16) swapper (leftBlock, rightBlock)
    }
    combine (32, 64, leftBlock, rightBlock, outBlock)
    permute (64, 64, outBlock, cipherBlock, FinalPermutationTable)
}
```

# Pseudocode for cipher – First approach

```
mixer (leftBlock[48], rightBlock[48], RoundKey[48])
{
    copy (32, rightBlock, T1)
    function (T1, RoundKey, T2)
    exclusiveOr (32, leftBlock, T2, T3)
    copy (32, T3, rightBlock)
}

swapper (leftBlock[32], rigthBlock[32])
{
    copy (32, leftBlock, T)
    copy (32, rightBlock, leftBlock)
    copy (32, T, rightBlock)
}
```

# Pseudocode for cipher – First approach

```
function (inBlock[32], RoundKey[48], outBlock[32])
{
    permute (32, 48, inBlock, T1, ExpansionPermutationTable)
    exclusiveOr (48, T1, RoundKey, T2)
    substitute (T2, T3, SubstituteTables)
    permute (32, 32, T3, outBlock, StraightPermutationTable)
}

substitute (inBlock[32], outBlock[48], SubstitutionTables[8, 4, 16])
{
    for (i = 1 to 8)
    {
        row ← 2 × inBlock[i × 6 + 1] + inBlock [i × 6 + 6]
        col ← 8 × inBlock[i × 6 + 2] + 4 × inBlock[i × 6 + 3] +
              2 × inBlock[i × 6 + 4] + inBlock[i × 6 + 5]

        value = SubstitutionTables [i][row][col]

        outBlock[[i × 4 + 1] ← value / 8;           value ← value mod 8
        outBlock[[i × 4 + 2] ← value / 4;           value ← value mod 4
        outBlock[[i × 4 + 3] ← value / 2;           value ← value mod 2
        outBlock[[i × 4 + 4] ← value
    }
}
```

# Cipher and Reverse Cipher

---

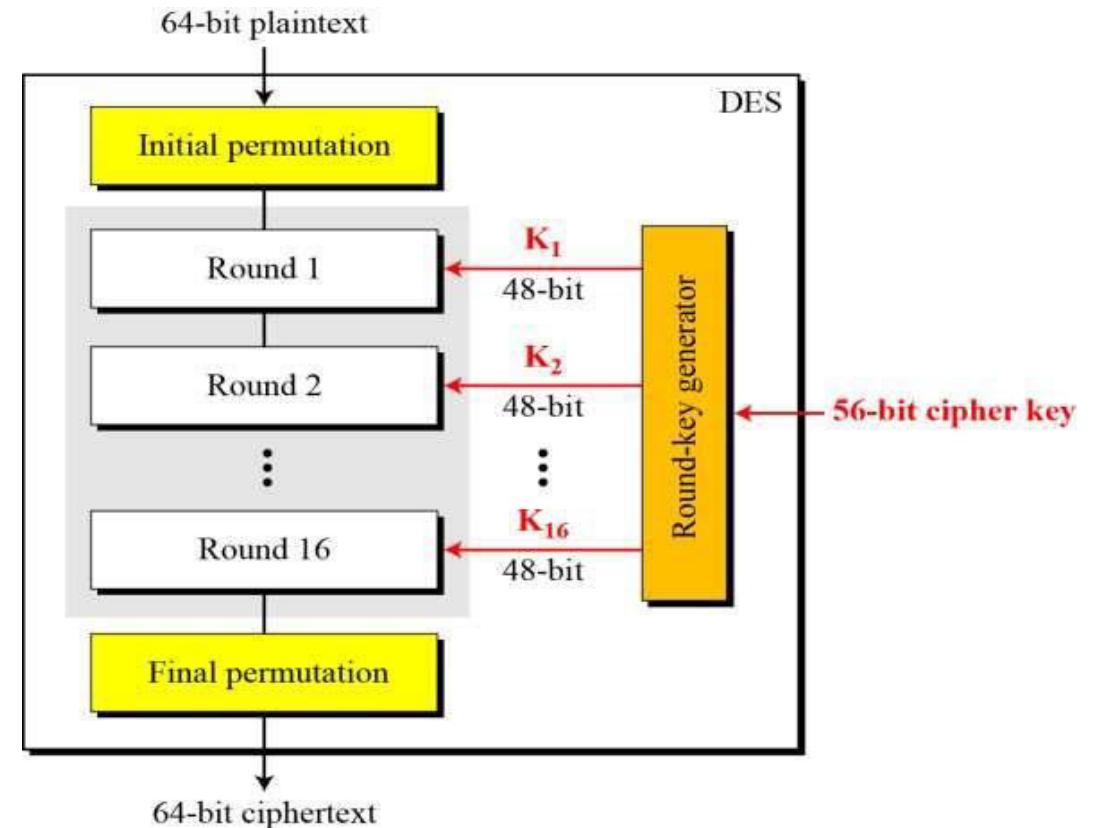
## Alternative Approach

Make the all 16 rounds the same by including one swapper to the 16<sup>th</sup> round and an extra swapper after that .

# DES Structure

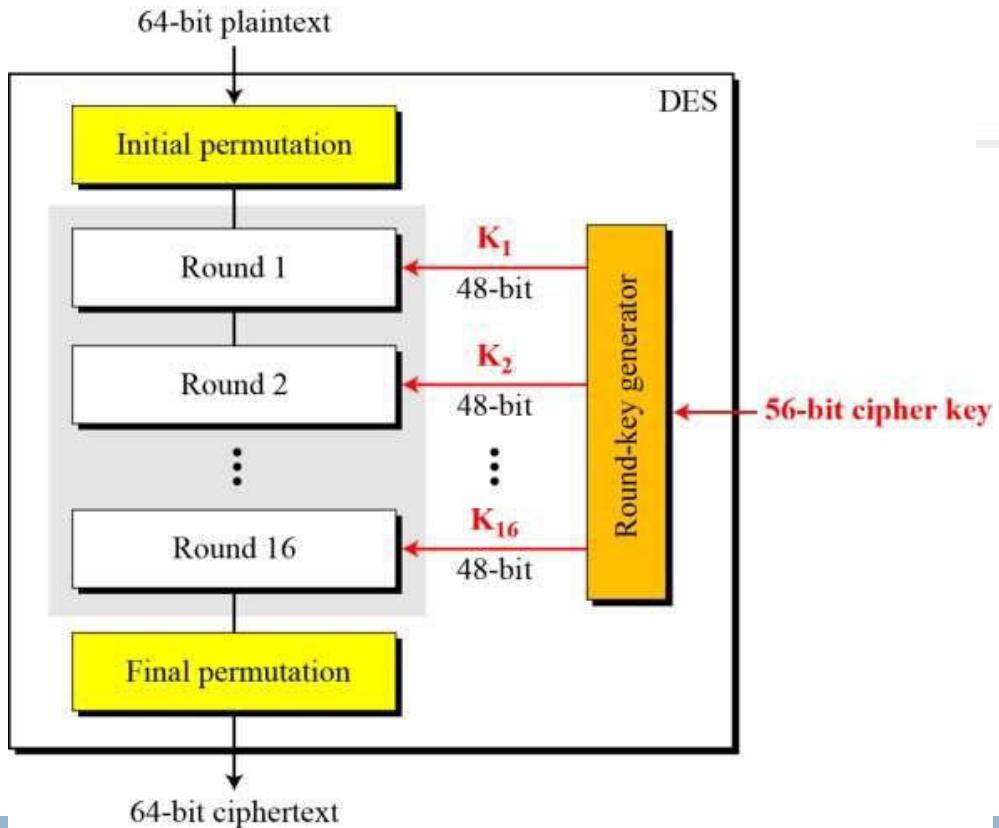
## Round Key Generator

- The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key.
- The process of key generation is depicted in the following illustration

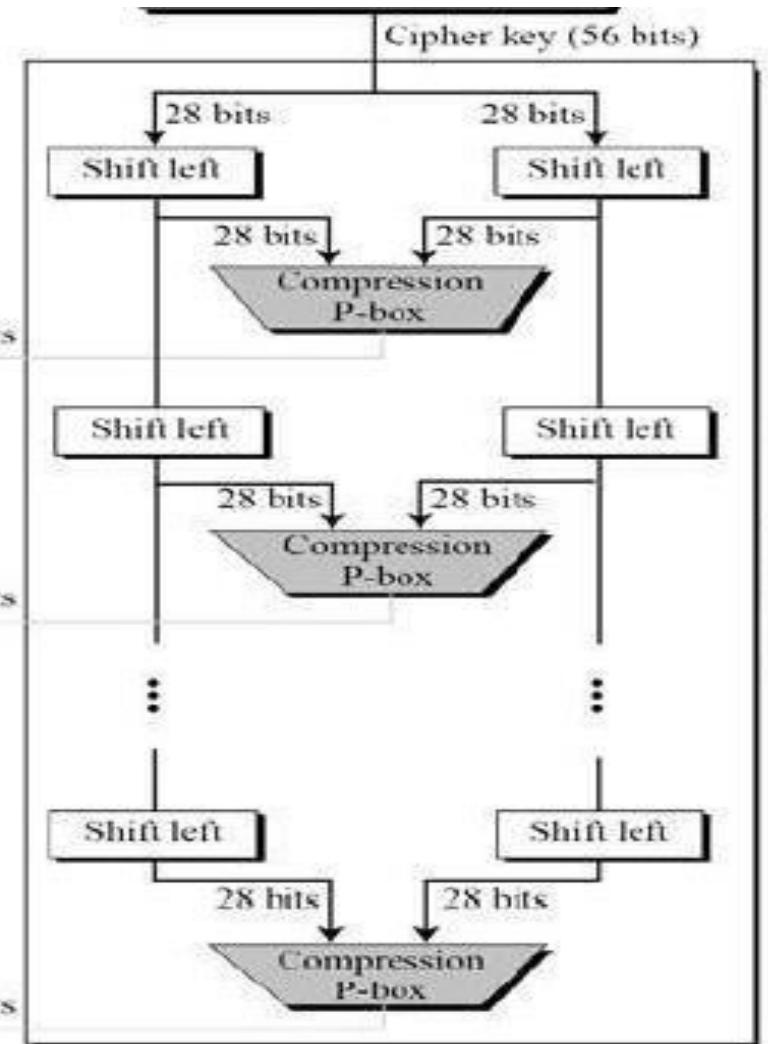


# DES Structure

## Round Key Generator



Shifting	
Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits



Round-Key Generator

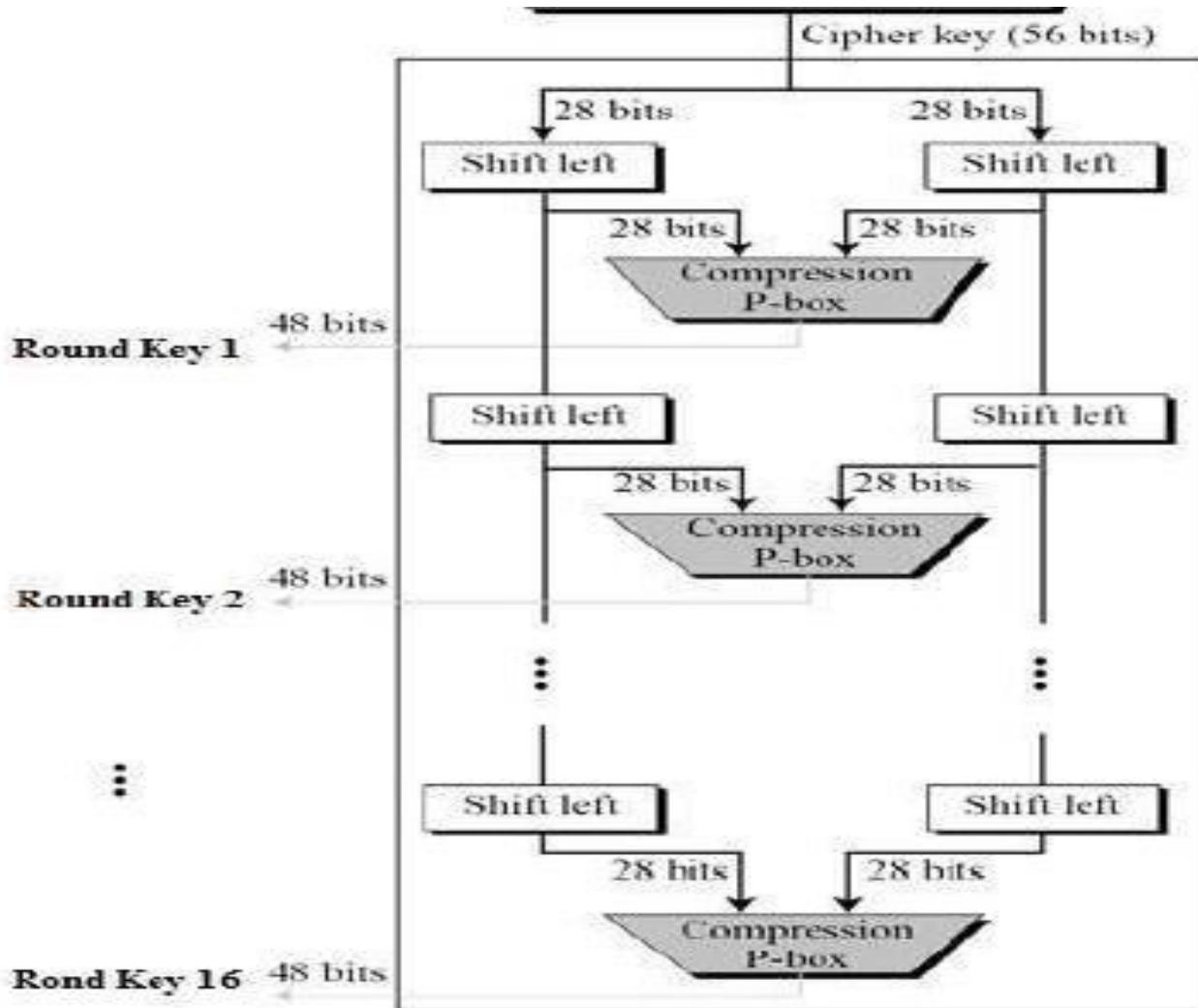
# DES Structure

## Round Key Generator

### Shifting

Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits

14	17	11	24	01	05	03	28
15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32



Round-Key Generator

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bit shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

# DES Structure

## Round Key Generator

```
Key_Generator (keyWithParities[64], RoundKeys[16, 48], ShiftTable[16])
{
    permute (64, 56, keyWithParities, cipherKey, ParityDropTable)
    split (56, 28, cipherKey, leftKey, rightKey)
    for (round = 1 to 16)
    {
        shiftLeft (leftKey, ShiftTable[round])
        shiftLeft (rightKey, ShiftTable[round])
        combine (28, 56, leftKey, rightKey, preRoundKey)
        permute (56, 48, preRoundKey, RoundKeys[round], KeyCompressionTable)
    }
}
```

# DES Analysis

---

## Properties of a Block Cipher    DES Weakness

1. Avalanche Effect
2. Completeness Effect

## Design Criteria

- S-Boxes
- P-Boxes
- Number of rounds

1. Weakness in cipher design
  - S-Boxes
  - P-Boxes
2. Weakness in cipher key
  - Key size
  - Weak keys
  - Semi-weak keys
  - Possible Weak keys

# DES Analysis

---

## Properties of a Block Cipher - Avalanche Effect

- Avalanche effect is considered as one of the desirable property of any encryption algorithm.
- A slight change in either the **key or the plain-text** should result in a significant change in the cipher-text. This property is termed as avalanche effect.

# DES Analysis

---

## Properties of a Block Cipher - Avalanche Effect

Plaintext: 0000000000000000

Key: 22234512987ABB23

Ciphertext: 4789FD476E82A5F1

Plaintext: 00000000000000001

Key: 22234512987ABB23

Ciphertext: 0A4ED5C15A63FEA3

# DES Analysis

---

**Properties of a Block Cipher - Completeness Effect**

Each bit of ciphertext depends on many bits of plaintext.

The diffusion and confusion produced by P-Box and S-Box in DES, shows strong completeness effect

# DES Analysis

---

## Design Criteria

- S-Boxes
- P-Boxes
- Number of rounds

# DES Analysis

## Design Criteria - S-Boxes

S-box 1																
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

- The entries of each row are permutations of values between 0 and 15.
- S-boxes are nonlinear.
- If we change a single bit in the input, two or more bits will be changed in the output.
- If two inputs to an S-box differ only in two middle bits (bits 3 and 4), the output must differ in at least two bits.
- If two inputs to an S-box differ in first two middle bits (bits 1 and 2) and the same in the last two bits (bits 5 and 6), the two outputs must different.

# DES Analysis

---

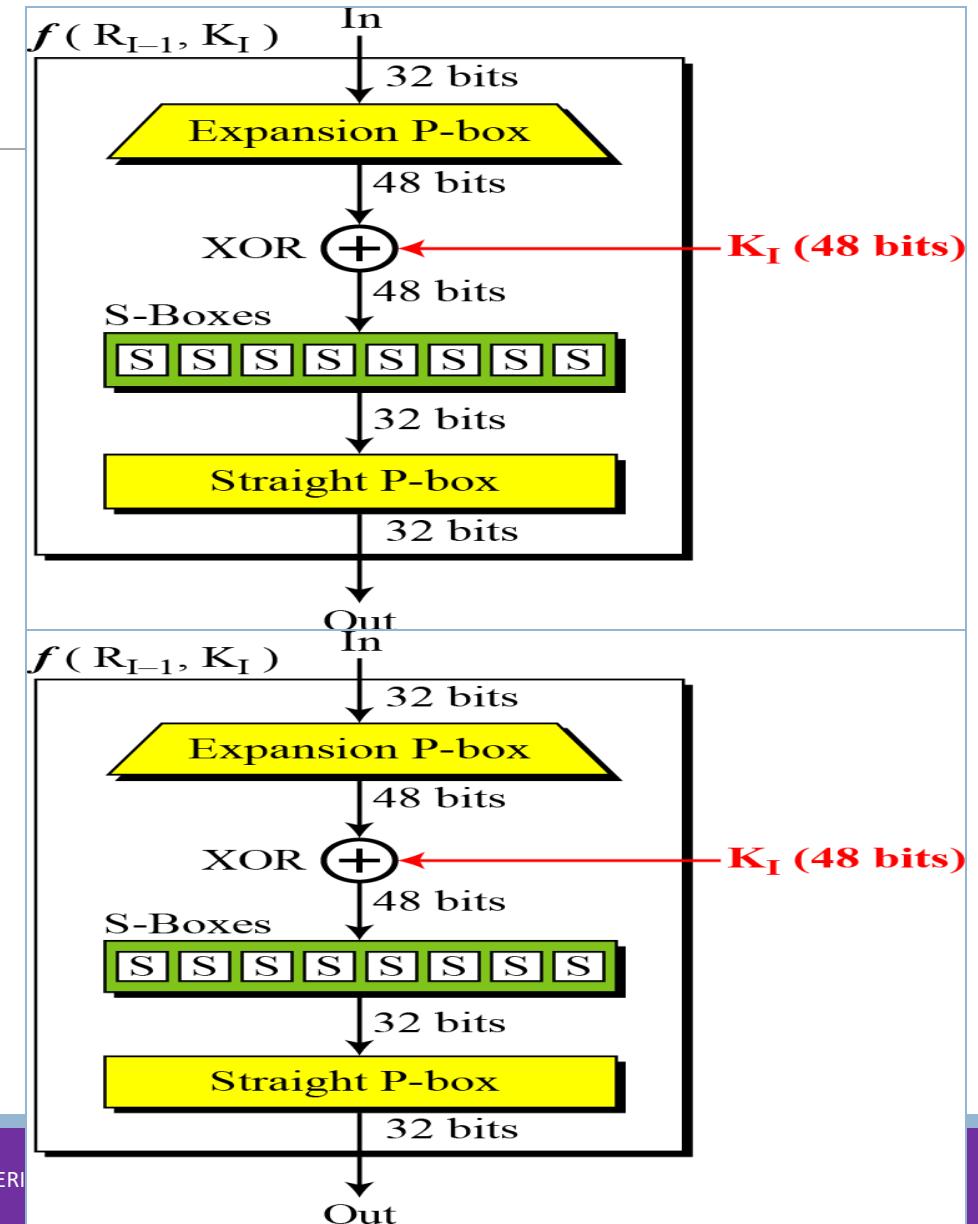
## Design Criteria - S-Boxes

6. There are only 32 6-bit input-word pairs ( $x_i$  and  $x_j$ ), in which  $x_i \oplus x_j \neq (000000)_2$ . These 32 input pairs create 32 4-bit output-word pairs. If we create the difference between the 32 output pairs,  $d = y_i \oplus y_j$ , no more than 8 of these  $d$ 's should be the same.
7. A criterion similar to #6 is applied to three S-boxes.
8. In any S-box, if a single input bit is held constant (0 or 1) and other bits are changed randomly, the differences between the numbers of 0s and 1s are minimized.

# DES Analysis

## Design Criteria - P-Boxes

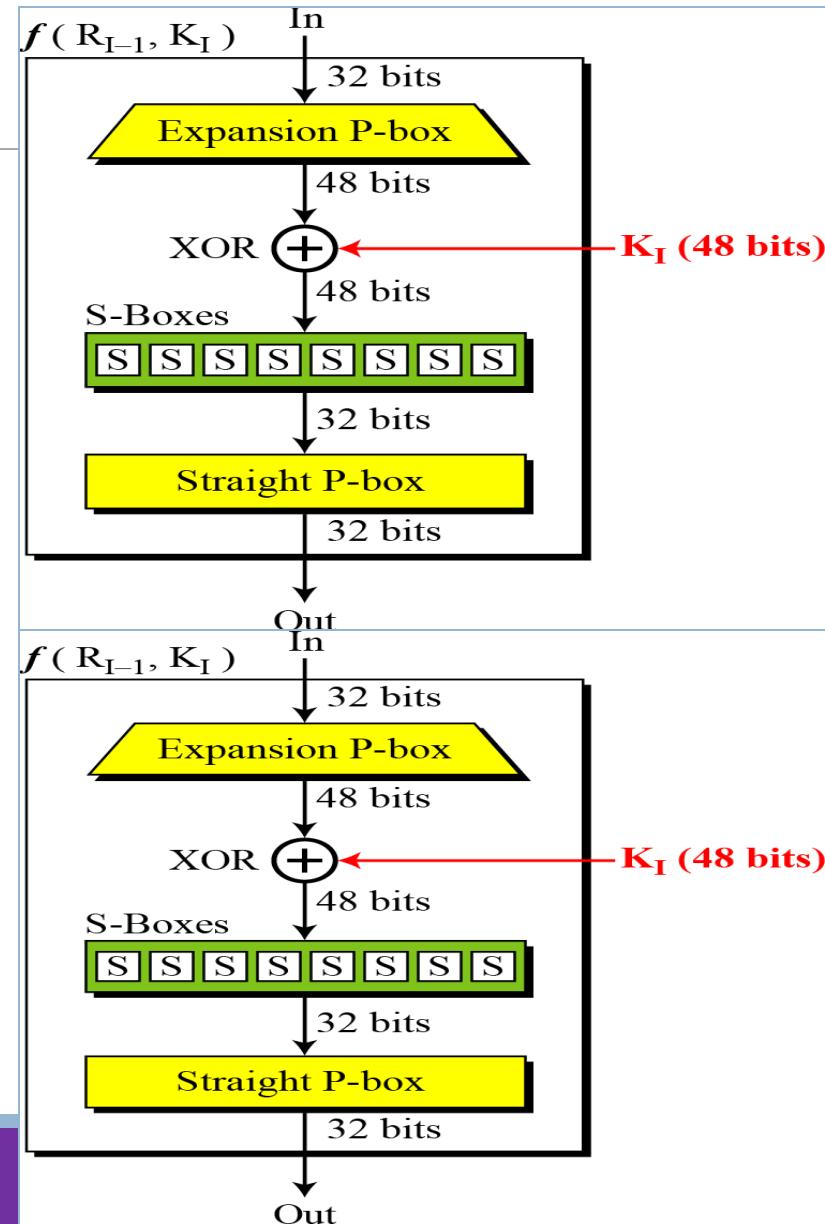
- Between two rows of S-boxes (in two subsequent rounds), there are one straight P-box and one expansion P-box.
- These two P-boxes provide diffusion of bits.



# DES Analysis

## Design Criteria - P-Boxes

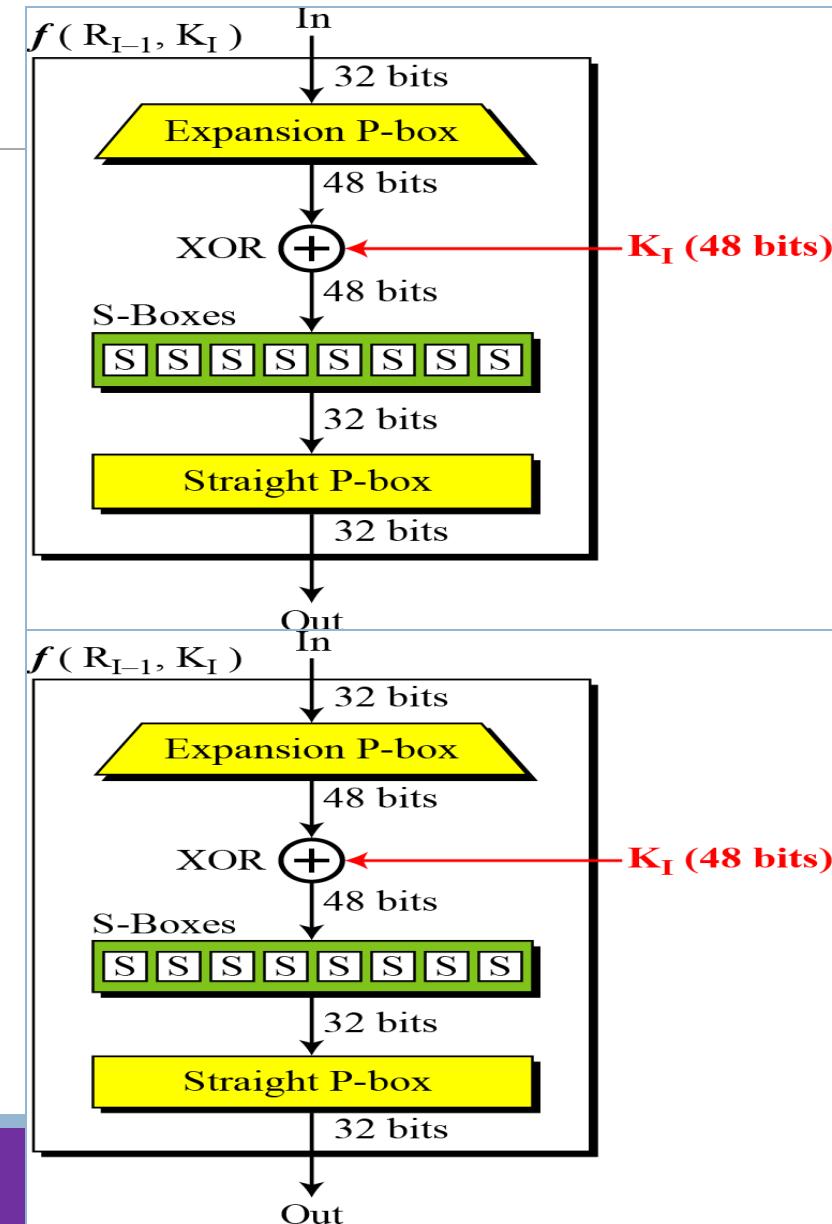
- Each S-box input comes from the output of a different S-box.
- No input to a given S-box comes from the output from the same box (in the previous round).
- The four outputs from each S-box go to six different S-boxes (in the next round).
- No two output bits from an S-box go to the same S-box (in the next round).



# DES Analysis

## Design Criteria - P-Boxes

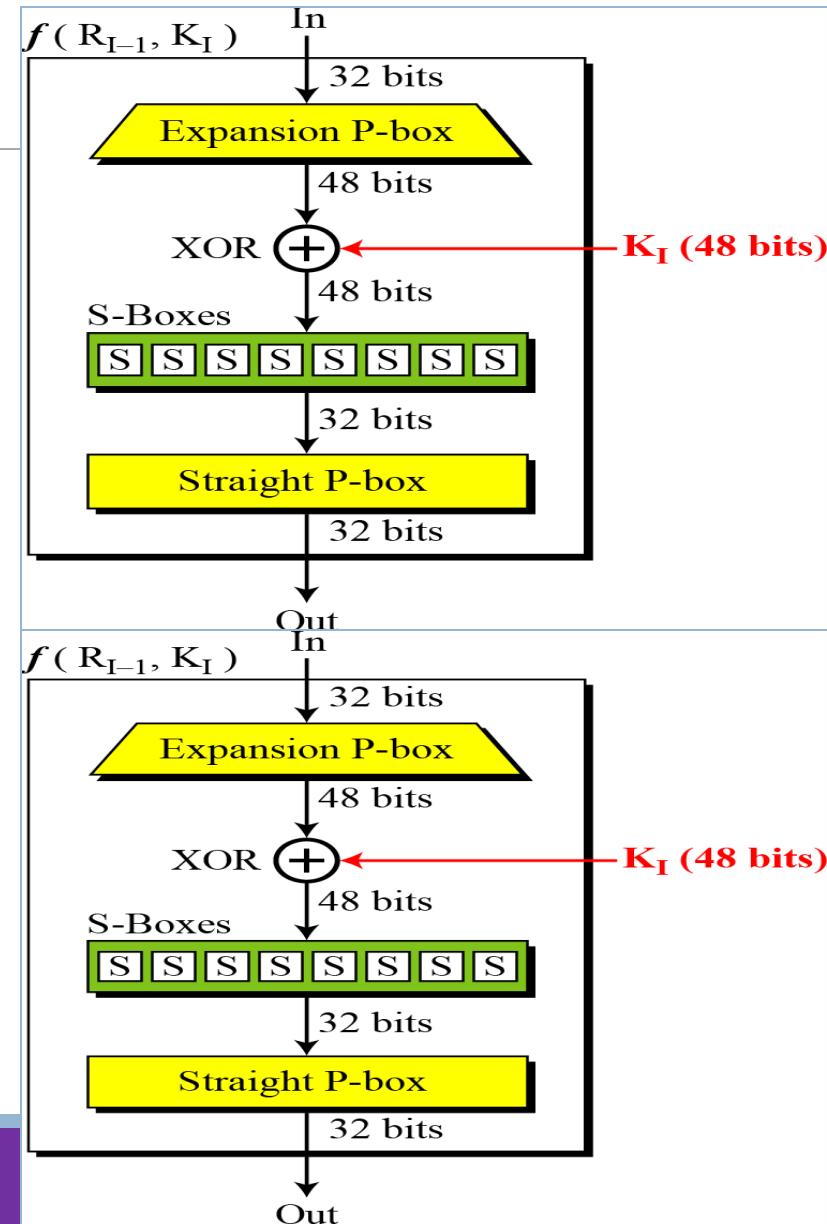
- If we number the eight S-boxes,  $S_1, S_2, \dots, S_8$ ,
  - An output of  $S_{j-2}$  goes to one of the first two bits of  $S_j$  (in the next round).
  - An output of  $S_{j-1}$  goes to one of the last two bits of  $S_j$  (in the next round).
  - An output of  $S_{j+1}$  goes to one of the two middle bits of  $S_j$  (in the next round).



# DES Analysis

## Design Criteria - P-Boxes

- For each S-box, the two output bits go to the first or last two bits of an S-box in the next round. The other two output bits go to the middle bits of an S-box in the next round.
- If an output bit from  $S_j$  goes to one of the middle bits in  $S_k$  (in the next round), then an output bit from  $S_k$  cannot go to the middle bit of  $S_j$ . If we let  $j=k$ , this implies that none of the middle bits of an S-box can go to one of the middle bits of the same S-box the next round.



# DES Analysis

---

## Design Criteria - Number of rounds

- DES uses sixteen rounds of Feistel ciphers.
- It has been proved that after eight rounds, each ciphertext is a function of every plaintext bit and every key bit; the ciphertext is thoroughly a random function of plaintext and ciphertext.
- Therefore, it looks like 8 rounds should be enough.
- However, experiments have found that DES versions with less than 16 rounds are even more vulnerable to known-plaintext than brute-force attack, which justify the use of 16 rounds by the designers of DES.

# DES Analysis

---

## DES Weakness

### 1. Weakness in cipher design

- S-Boxes
- P-Boxes

### 2. Weakness in cipher key

- Key size
- Weak keys
- Semi-weak keys
- Possible weak keys

# DES Analysis

## DES Weakness - Weakness in cipher design

### S-Boxes

- In S-box 4, the last three output bits can be derived in the same way as the first output bit by complementing some of the input bits.
- Two specifically chosen inputs to an S-box array can create the same output.
- It is possible to obtain the same output in a single round by changing bits in only three neighboring S-boxes.

S-box 4																
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	07	13	14	03	00	6	09	10	1	02	08	05	11	12	04	15
1	13	08	11	05	06	15	00	03	04	07	02	12	01	10	14	09
2	10	06	09	00	12	11	07	13	15	01	03	14	05	02	08	04
3	03	15	00	06	10	01	13	08	09	04	05	11	12	07	02	14

# DES Analysis

---

## DES Weakness - Weakness in cipher design

### P-Boxes

- It is not clear why designers of DES used the initial and final permutations; these have no security benefits.
- In the expansion permutation (inside the function), the first and fourth bits of every 4-bit series are repeated.

# DES Analysis

---

## DES Weakness - Weakness in cipher key

- Key size
- Weak keys
- Semi-weak keys
- Possible weak keys

# DES Analysis

---

## Weakness in cipher key - Key size(56 bits)

Brute force attack adversary will check with –  $2^{56}$  keys

- One computer with processor → more than thousand years
- Computer with parallel processing → 20 hours
- Computer network with parallel processing → 120 days (key challenged by RSA Laboratories)

Solution :

3DES with two keys(112bits)

3DES with three keys(168 bits)

# DES Analysis

---

## Weakness in cipher key-Weak keys

- Four out of  $2^{56}$  keys are called weak keys
- Round keys created from weak keys will have the same pattern as cipher key

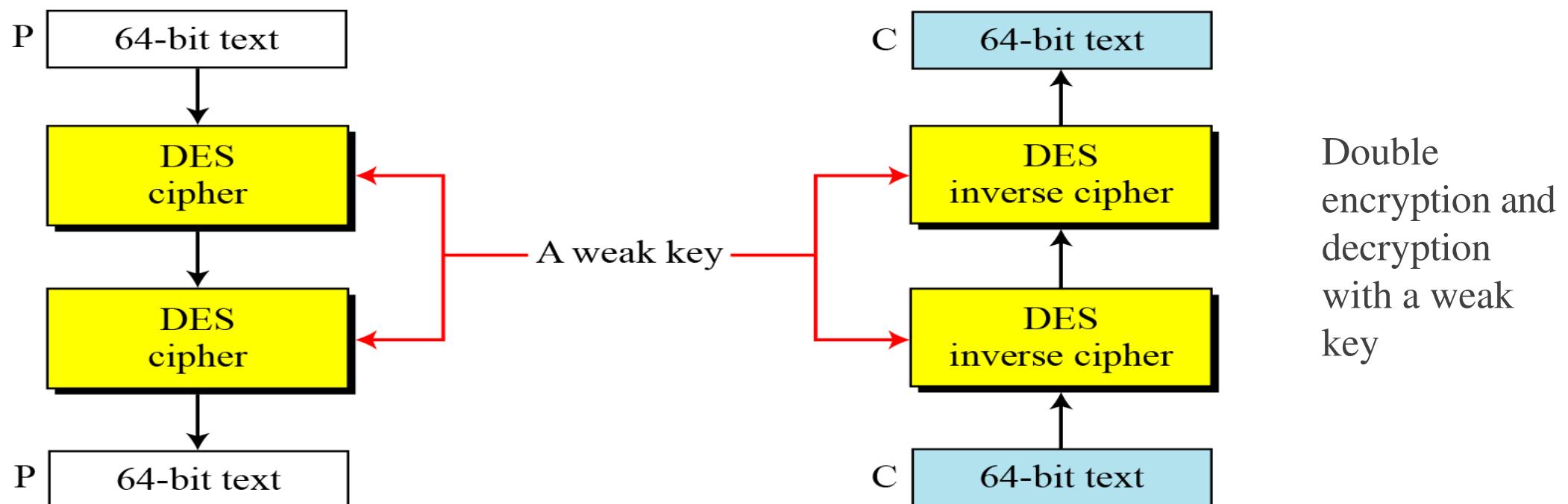
**Table 6.18** Weak keys

<i>Keys before parities drop (64 bits)</i>	<i>Actual key (56 bits)</i>
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

# DES Analysis

## Weakness in cipher key-Weak keys

- What is the advantage of using weak keys



# DES Analysis

## Weakness in cipher key-Weak keys

**Table 6.18** Weak keys

Keys before parities drop (64 bits)	Actual key (56 bits)
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFFF

Let us try the first weak key in Table 6.18 to encrypt a block two times.

After two encryptions with the same key the original plaintext block is created. Note that we have used the encryption algorithm two times, not one encryption followed by another decryption.

Key: 0x0101010101010101

Plaintext: 0x1234567887654321 → Ciphertext: 0x814FE938589154F7

Key: 0x0101010101010101

Plaintext: 0x814FE938589154F7 → Ciphertext: 0x1234567887654321

$$E_k(E_k(P)) = P$$

# DES Analysis

## Weakness in cipher key - Semi-weak keys

- There are six key pairs called semi weak keys
- A Semi weak keys creates two different round keys and each of them is repeated eight times
- Round key created from each pair are the same with different order

<i>First key in the pair</i>	<i>Second key in the pair</i>
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E
01E0 01E1 01F1 01F1	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1

# DES Analysis

## Weakness in cipher key - Semi-weak keys

<i>First key in the pair</i>	<i>Second key in the pair</i>
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01

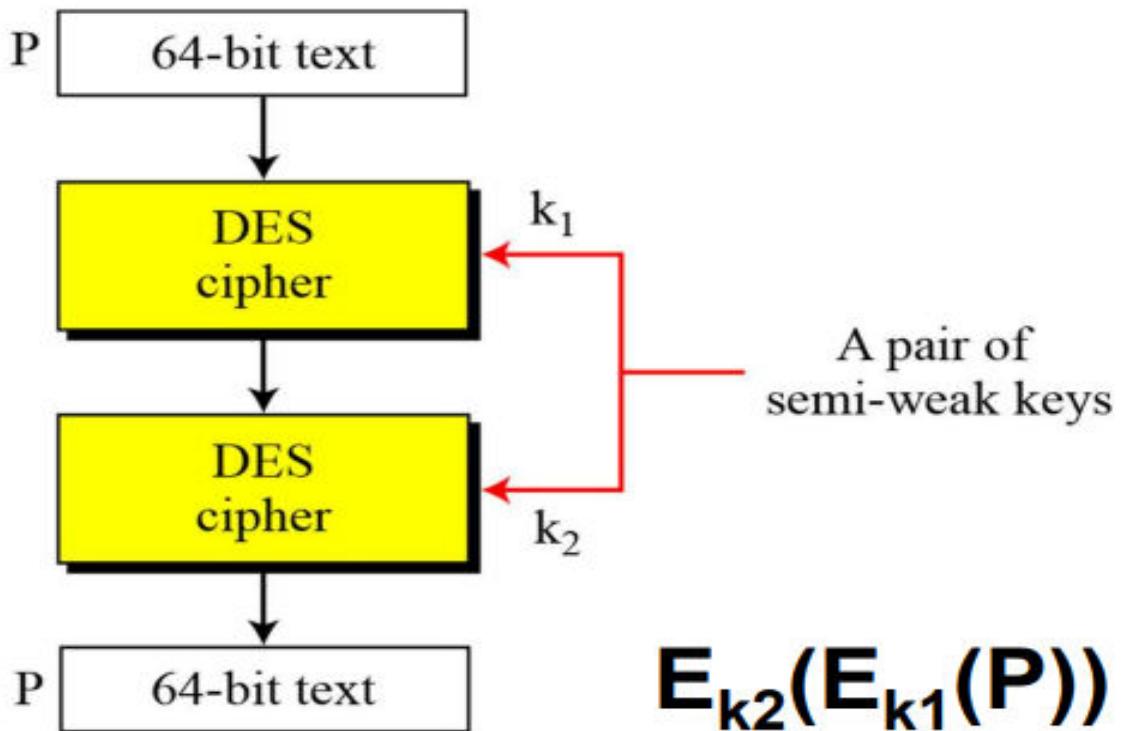
<i>Round key 1</i>	9153E54319BD	6EAC1ABCE642
<i>Round key 2</i>	6EAC1ABCE642	9153E54319BD
<i>Round key 3</i>	6EAC1ABCE642	9153E54319BD
<i>Round key 4</i>	6EAC1ABCE642	9153E54319BD
<i>Round key 5</i>	6EAC1ABCE642	9153E54319BD
<i>Round key 6</i>	6EAC1ABCE642	9153E54319BD
<i>Round key 7</i>	6EAC1ABCE642	9153E54319BD
<i>Round key 8</i>	6EAC1ABCE642	9153E54319BD
<i>Round key 9</i>	9153E54319BD	6EAC1ABCE642
<i>Round key 10</i>	9153E54319BD	6EAC1ABCE642
<i>Round key 11</i>	9153E54319BD	6EAC1ABCE642
<i>Round key 12</i>	9153E54319BD	6EAC1ABCE642
<i>Round key 13</i>	9153E54319BD	6EAC1ABCE642
<i>Round key 14</i>	9153E54319BD	6EAC1ABCE642
<i>Round key 15</i>	9153E54319BD	6EAC1ABCE642
<i>Round key 16</i>	6EAC1ABCE642	9153E54319BD

<i>First key in the pair</i>	<i>Second key in the pair</i>
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E
01E0 01E1 01F1 01F1	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1

# DES Analysis

## Weakness in cipher key - Semi-weak keys

A pair of semi-weak keys in encryption and decryption



$$E_{k2}(E_{k1}(P)) = P$$

# DES Analysis

---

## Weakness in cipher key – Possible weak keys

- 48 Keys are possible weak keys
- A possible weak key is a key that creates four distinct round keys
- 16round keys = 4 groups → each group 4 equal round key

# DES Analysis

---

## Weakness in cipher key – Key clustering

- 2 or more different keys can create same ciphertext from the same plaintext.

# Security of DES

---

DES, as the first important block cipher, has gone through much scrutiny.

Among the attempted attacks, three are of interest

- Brute Force attack
- Differential cryptanalysis
- Linear cryptanalysis

# Security of DES

---

## Brute Force attack

- DES can be broken using  $2^{56}$

---

**Cryptanalysis** is the process of transforming or decoding communications from non-readable to readable format without having access to the real key.

### **Different Forms of Cryptanalysis:**

1. **Differential Cryptanalysis:** The use of differential cryptanalysis is to get clues about some critical bits, reducing the need for an extensive search.
2. **Linear Cryptanalysis:** The use of linear cryptanalysis is to figure out what is the linear relationship present between some plaintext bits, ciphertext bits, and unknown key bits very easily.

# Security of DES

---

## Differential cryptanalysis

- It has been revealed that the designers of DES already knew about this type of attack and **designed S-boxes** and **chose 16 as the number of rounds** to make DES specifically resistant to this type of attack.

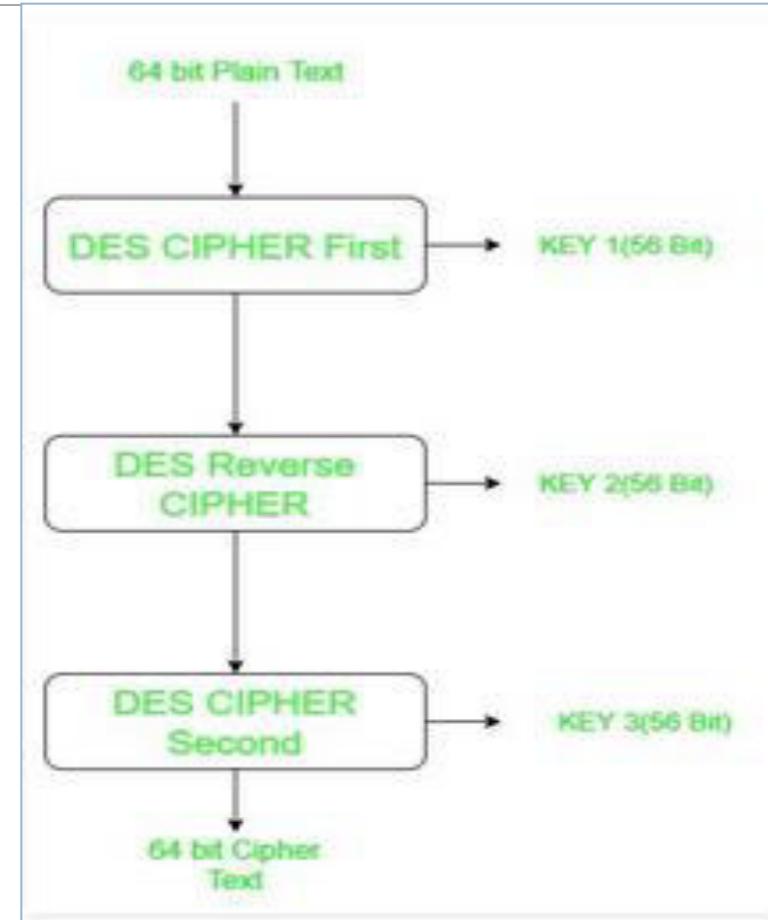
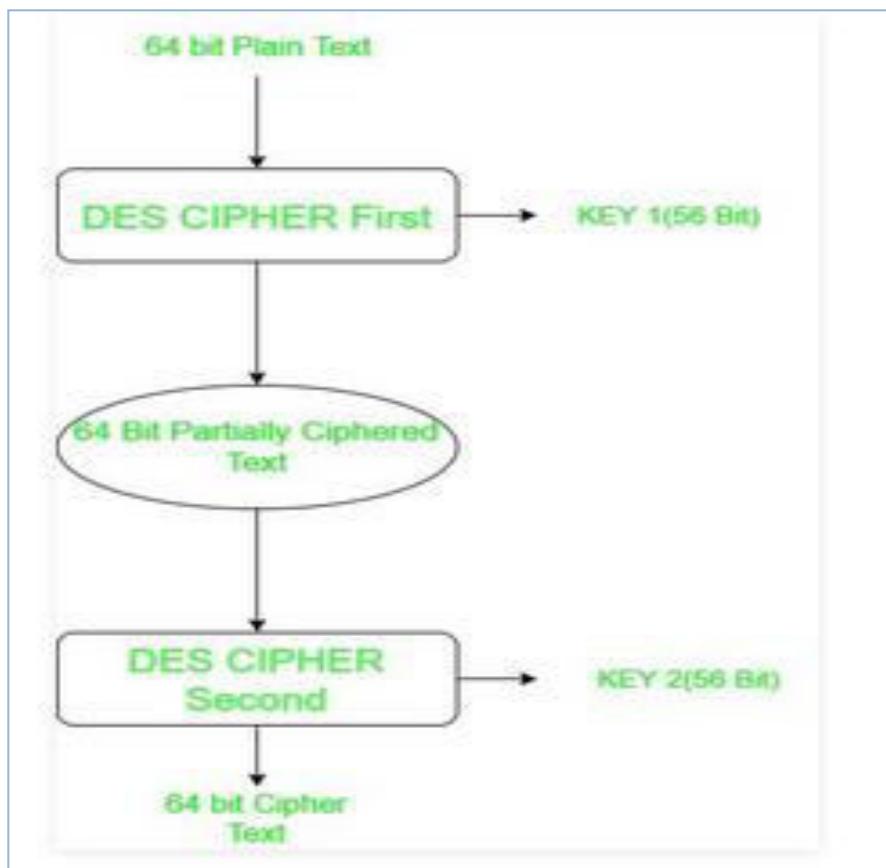
# Security of DES

---

## Linear cryptanalysis

- DES is more vulnerable to linear cryptanalysis than to differential cryptanalysis.
- S-boxes are not very resistant to linear cryptanalysis.
- DES can be broken using  $2^{43}$  pairs of known plaintexts. However, from the practical point of view, finding so many pairs is very unlikely.

# Double DES and Triple DES



# Why Was the AES Encryption Algorithm necessary?

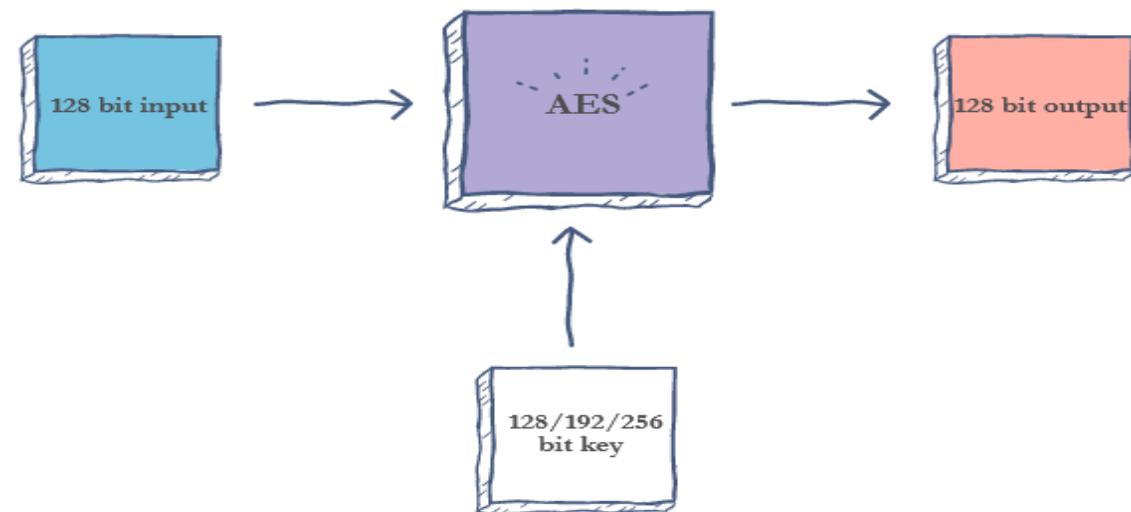
---

Chronology of DES Cracking	
Broken for the first time	1997
Broken in 56 hours	1998
Broken in 22 hours and 15 minutes	1999
Capable of broken in 5 minutes	2021

# Introduction

Advanced Encryption Standard (AES) features are as follows:

- Symmetric key symmetric block cipher
- 128-bit block data, 128/192/256-bit keys
- Stronger and faster than Triple-DES



	DES	AES
Developed	1977	2000
Cipher Type	Symmetric block cipher	Symmetric block cipher
Block size	64 bits	128 bits
Key length	56 bits	128/192/256 bits
Security	Rendered insecure	Considered secure

# Introduction

---

- AES is an iterative rather than Feistel cipher.
- It is based on ‘substitution–permutation network’.
- It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).
- AES performs all its computations on bytes rather than bits.

# Introduction

---

The Advanced Encryption Standard (AES) published by the National Institute of Standards and Technology (NIST) in December 2001.

- History
- Criteria
- Rounds
- Data Units
- Structure of Each Round

# Introduction

---

## History

- In February 2001, NIST announced that a draft of the Federal Information Processing Standard (FIPS) was available for public review and comment.
- Finally, AES was published as FIPS 197 in the Federal Register in December 2001.

# Introduction

---

## Criteria

The criteria defined by NIST for selecting AES fall into three areas:

1. Security -128bit key
2. Cost-Computational efficiency and storage requirements
3. Implementation -Flexibility(platform independent)

# Introduction

---

## Rounds

- AES is a non Feistel cipher that encrypts and decrypts a block of 128bit data.
- It uses 10,12 or 14 rounds → keysize-128/192/256bits

# Introduction

---

## Rounds

AES-128

AES-192

AES-256

But roundkey is

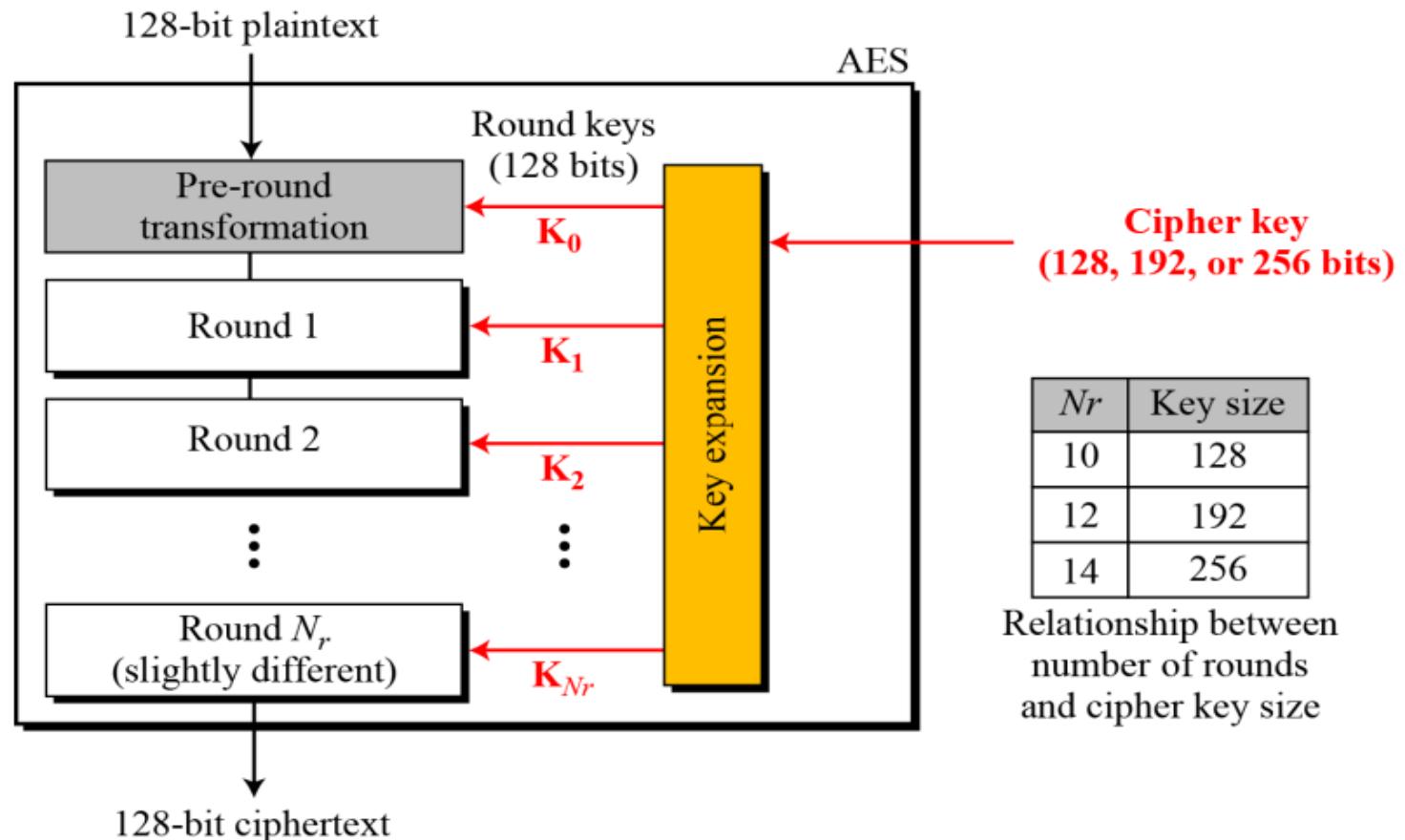
Always 128bits

# Introduction

## Rounds

The number of round key is one more than the number of rounds

Number of Round keys =  $Nr+1$



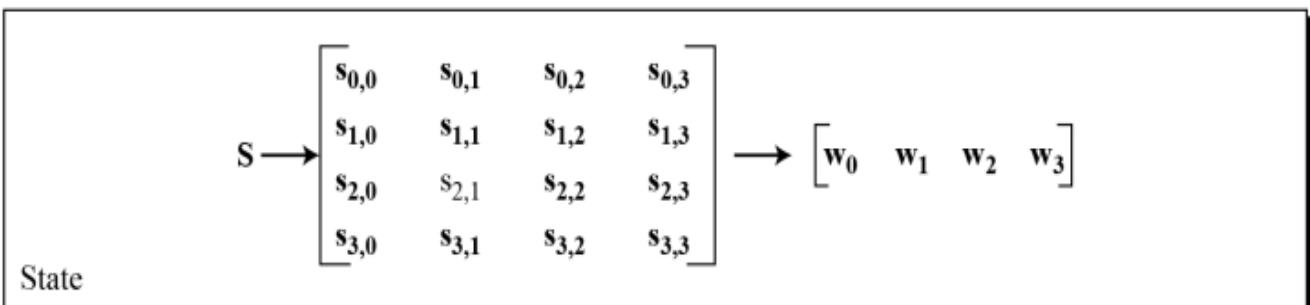
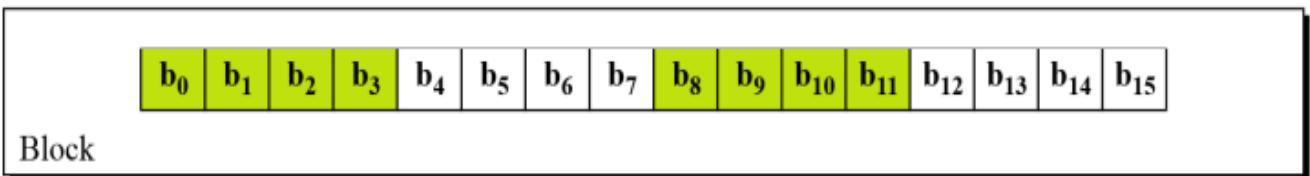
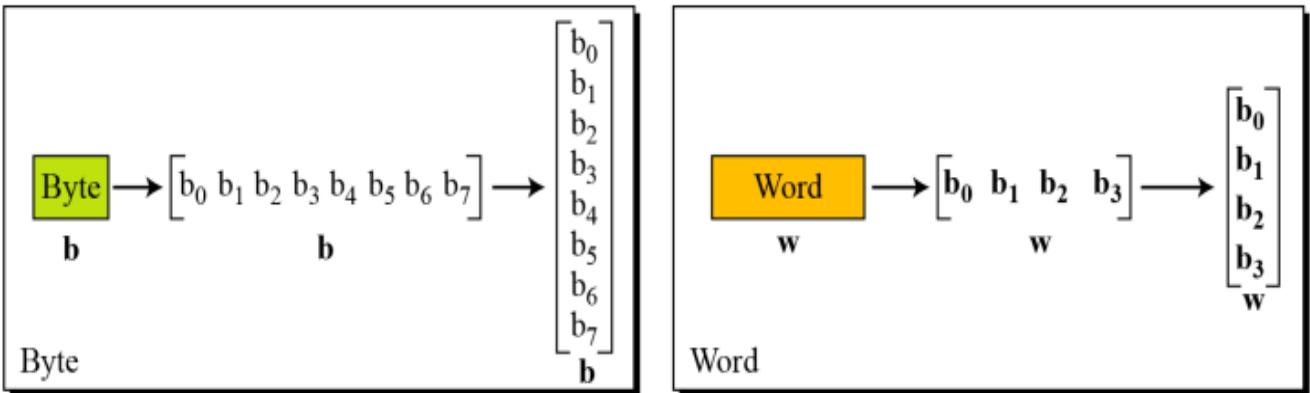
# Introduction

## Data Units

AES uses 5 units of measurement to refer to data

1. Bits
2. Bytes
3. Words
4. Blocks
5. State

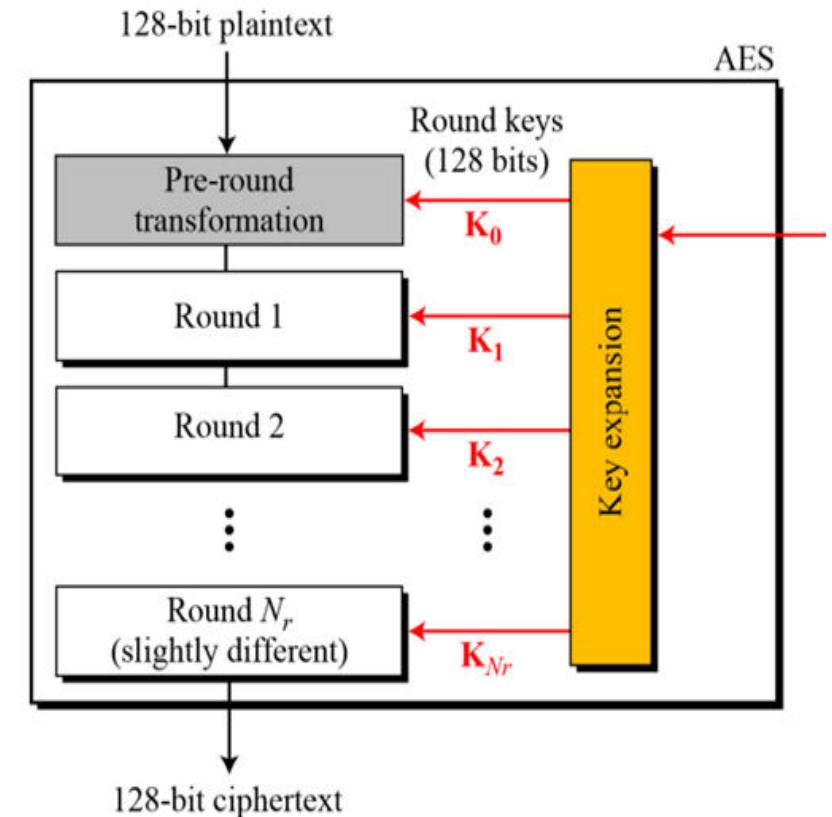
*Data units used in AES*



# Introduction

## Data Units – States

- AES uses several rounds
- Each round is made of several stages – Data blocks are transferred from one stage to other
- At the beginning and end of cipher – data blocks
- Before and after each stage – data block is referred as state

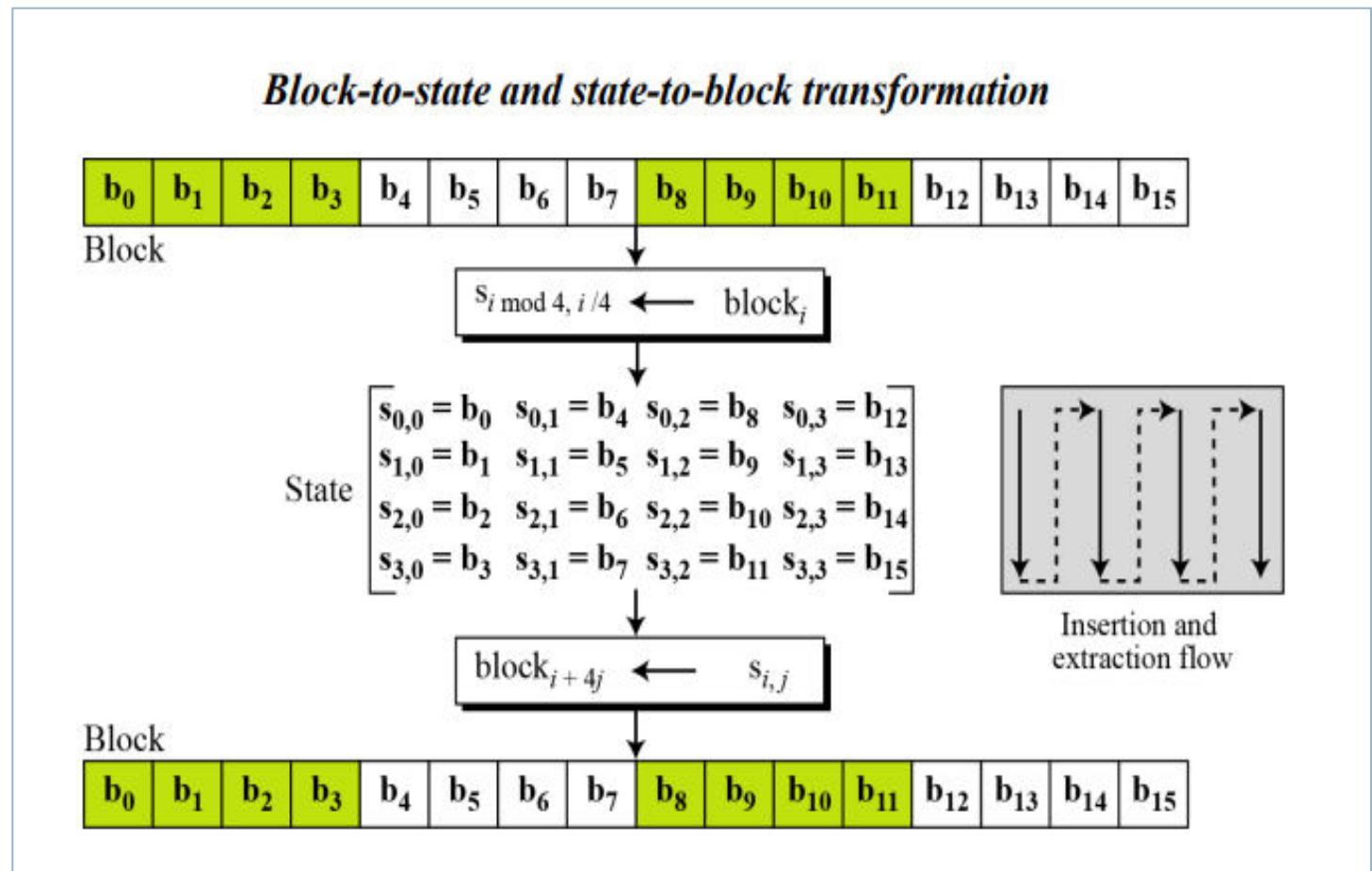


# Introduction

## Data Units – States

- S – State
- T- temporary state
- States are made up of 16bytes
- Matrix (4x4)

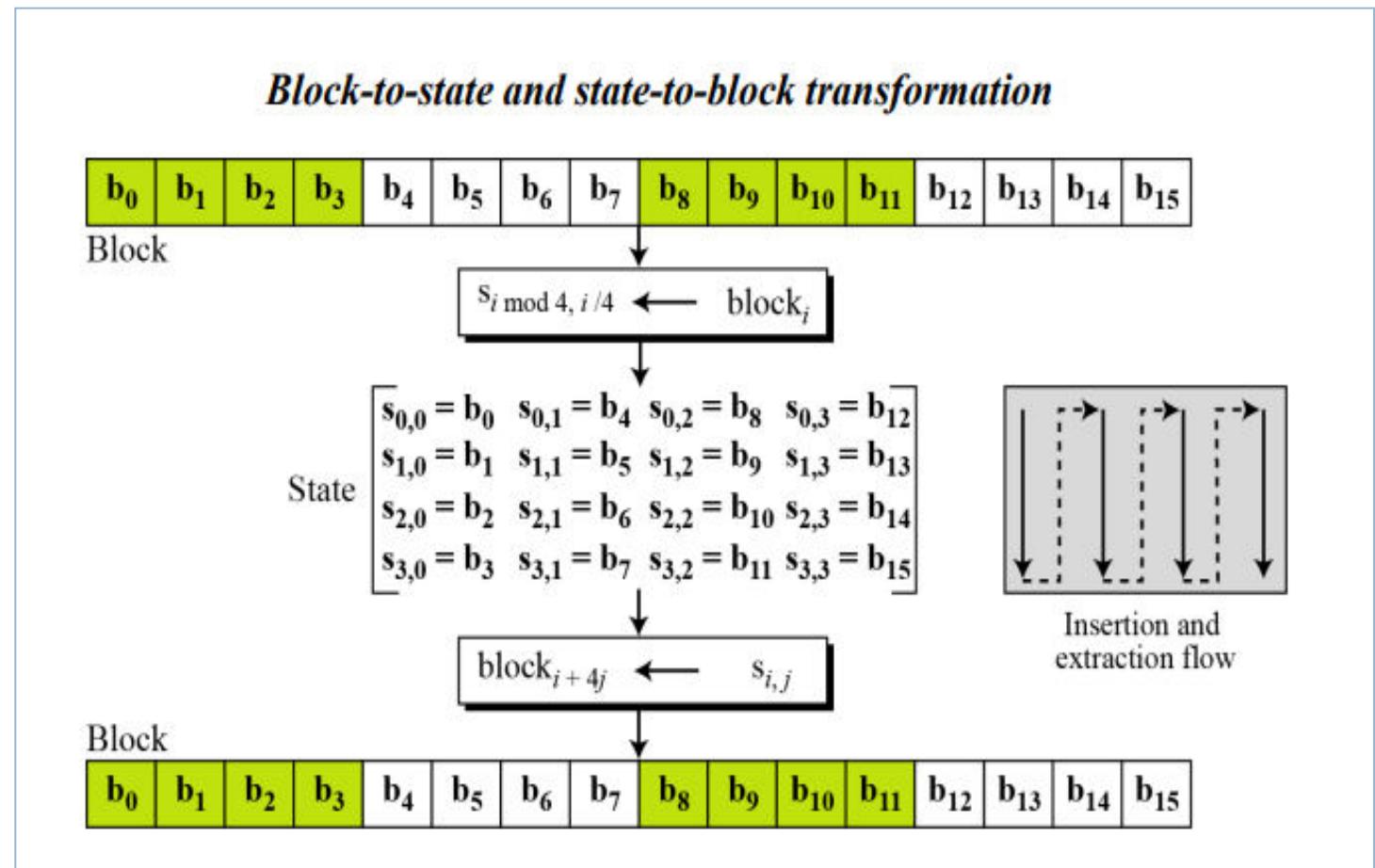
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15



# Introduction

## Data Units – States

- State is treated as row matrix( $1 \times 4$ ) of words



# Introduction

## Example

*Changing plaintext to state*

Text    A   E   S   U   S   E   S   A   M   A   T   R   I   X   Z   Z

Hexadecimal    00   04   12   14   12   04   12   00   0C   00   13   11   08   23   19   19

00	12	0C	08
04	04	00	23
12	12	13	19
14	00	11	19

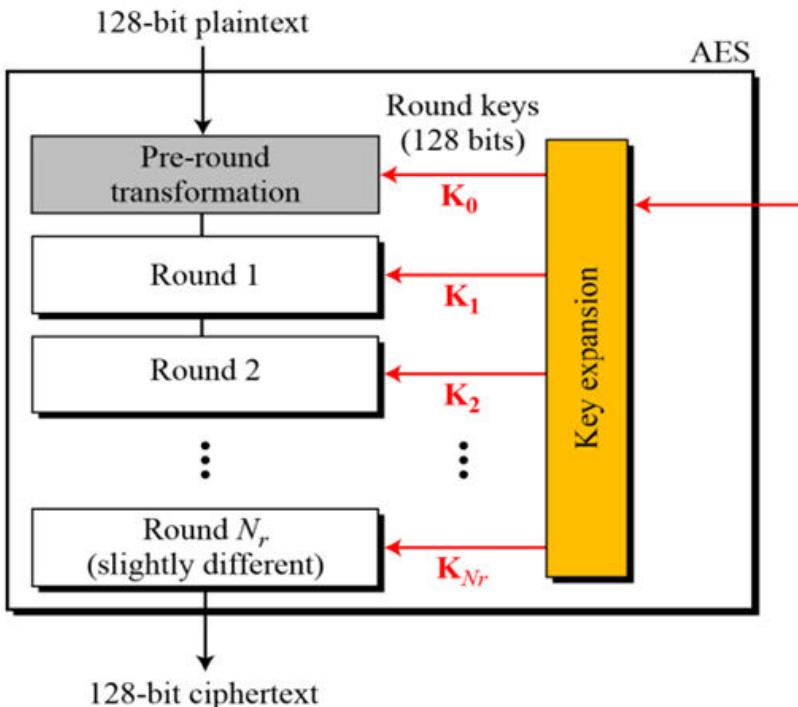
State

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 → 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  
 Value → 00   01   02   03   04   05   06   07   08   09   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   25

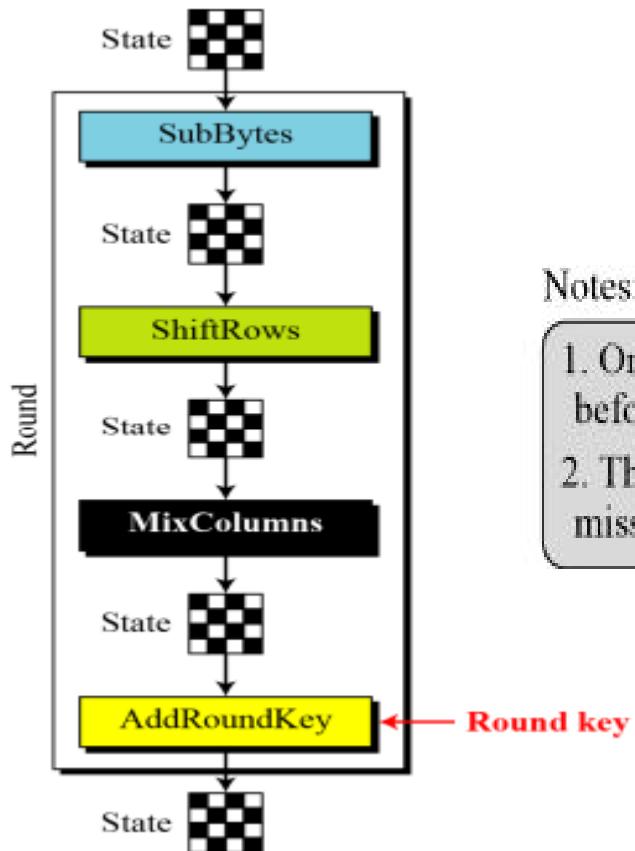
Decimal	Hex	Char
0	0	[NULL]
1	1	[START OF HEADING]
2	2	[START OF TEXT]
3	3	[END OF TEXT]
4	4	[END OF TRANSMISSION]
5	5	[ENQUIRY]
6	6	[ACKNOWLEDGE]
7	7	[BELL]
8	8	[BACKSPACE]
9	9	[HORIZONTAL TAB]
10	A	[LINE FEED]
11	B	[VERTICAL TAB]
12	C	[FORM FEED]
13	D	[CARRIAGE RETURN]
14	E	[SHIFT OUT]
15	F	[SHIFT IN]
16	10	[DATA LINK ESCAPE]
17	11	[DEVICE CONTROL 1]
18	12	[DEVICE CONTROL 2]
19	13	[DEVICE CONTROL 3]
20	14	[DEVICE CONTROL 4]
21	15	[NEGATIVE ACKNOWLEDGE]
22	16	[SYNCHRONOUS IDLE]
23	17	[END OF TRANS. BLOCK]
24	18	[CANCEL]
25	19	[END OF MEDIUM]
26	1A	[SUBSTITUTE]
27	1B	[ESCAPE]
28	1C	[FILE SEPARATOR]
29	1D	[GROUP SEPARATOR]
30	1E	[RECORD SEPARATOR]
31	1F	[UNIT SEPARATOR]

# Introduction

## Structure of Round



*Structure of each round at the encryption site*



Notes:

1. One AddRoundKey is applied before the first round.
2. The third transformation is missing in the last round.

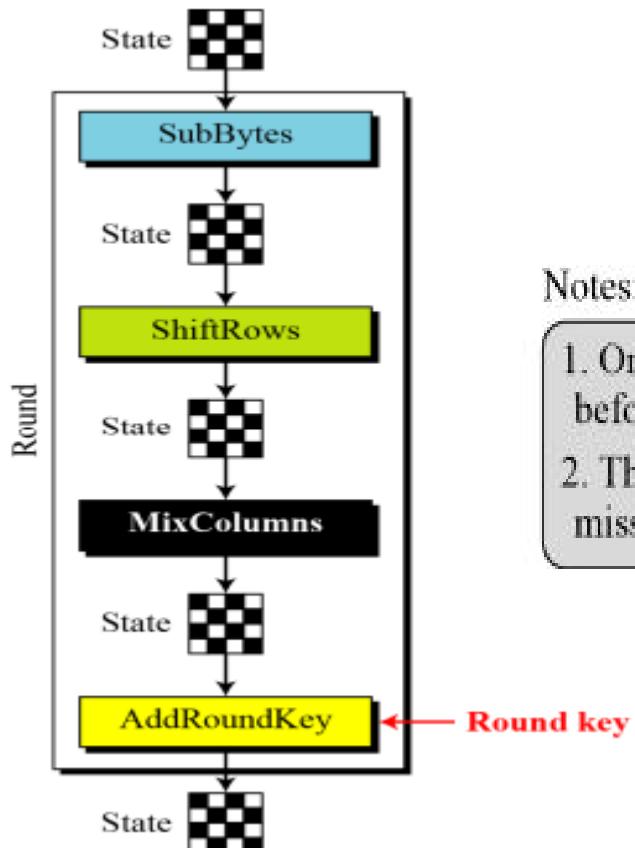
# Introduction

## Structure of Round

### 4 Transformations

1. SubBytes
2. ShiftRows
3. MixColumns
4. Add Round Keys

*Structure of each round at the encryption site*



Notes:

1. One AddRoundKey is applied before the first round.
2. The third transformation is missing in the last round.

# Transformation

---

1. SubBytes(Substitution)
2. ShiftRows (Permutation)
3. MixColumns (Mixing)
4. Add Round Keys (Adding)

# Transformation

---

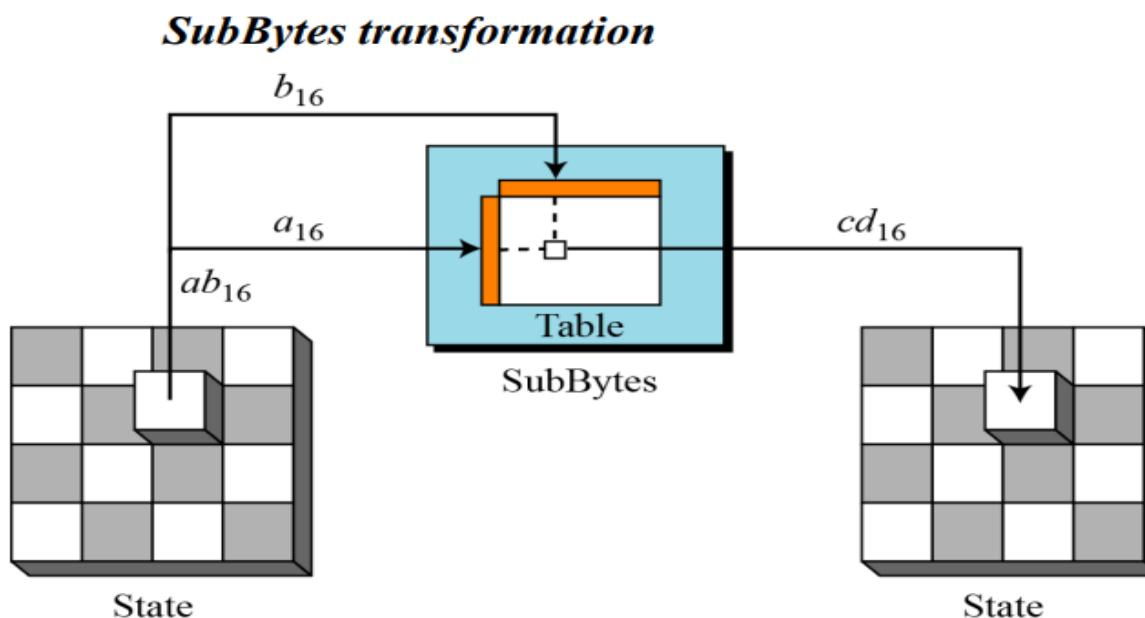
## SubBytes(Substitution)

- AES uses substitution
- Mechanism is different
  1. Substitution done for each byte
  2. Table is used for substitution for each byte
  3. Table Lookup process or mathematical calculation in  $GF(2^8)$  field

# Transformation

## SubBytes

- SubBytes is used at the encryption site.
- To substitute a byte, we interpret the byte as two hexadecimal digits.
- Left digit –row
- Right digit -column



# Transformation

## SubBytes- Transformation table

*SubBytes transformation table*

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	CB	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

INPUT

## InvSubBytes-Transformation table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5B	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3B	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

# Transformation

## Example 1

$$5A_{16} = BE_{16}$$

$$5B_{16} = 39_{16}$$

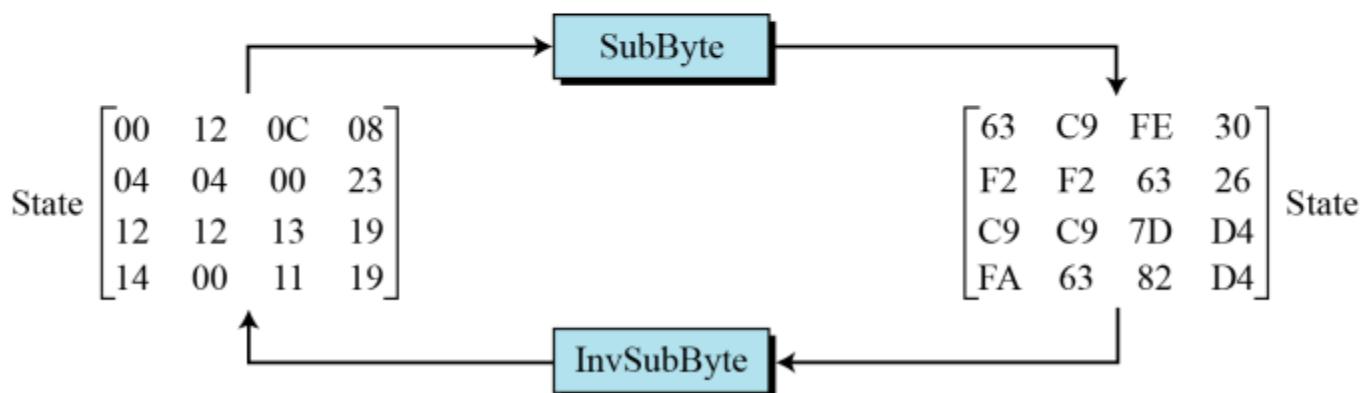
*SubBytes transformation table*

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	CB	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

# Transformation

## Example 2

shows how a state is transformed using the SubBytes transformation. The figure also shows that the InvSubBytes transformation creates the original one. Note that if the two bytes have the same values, their transformation is also the same.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

# Transformation

---

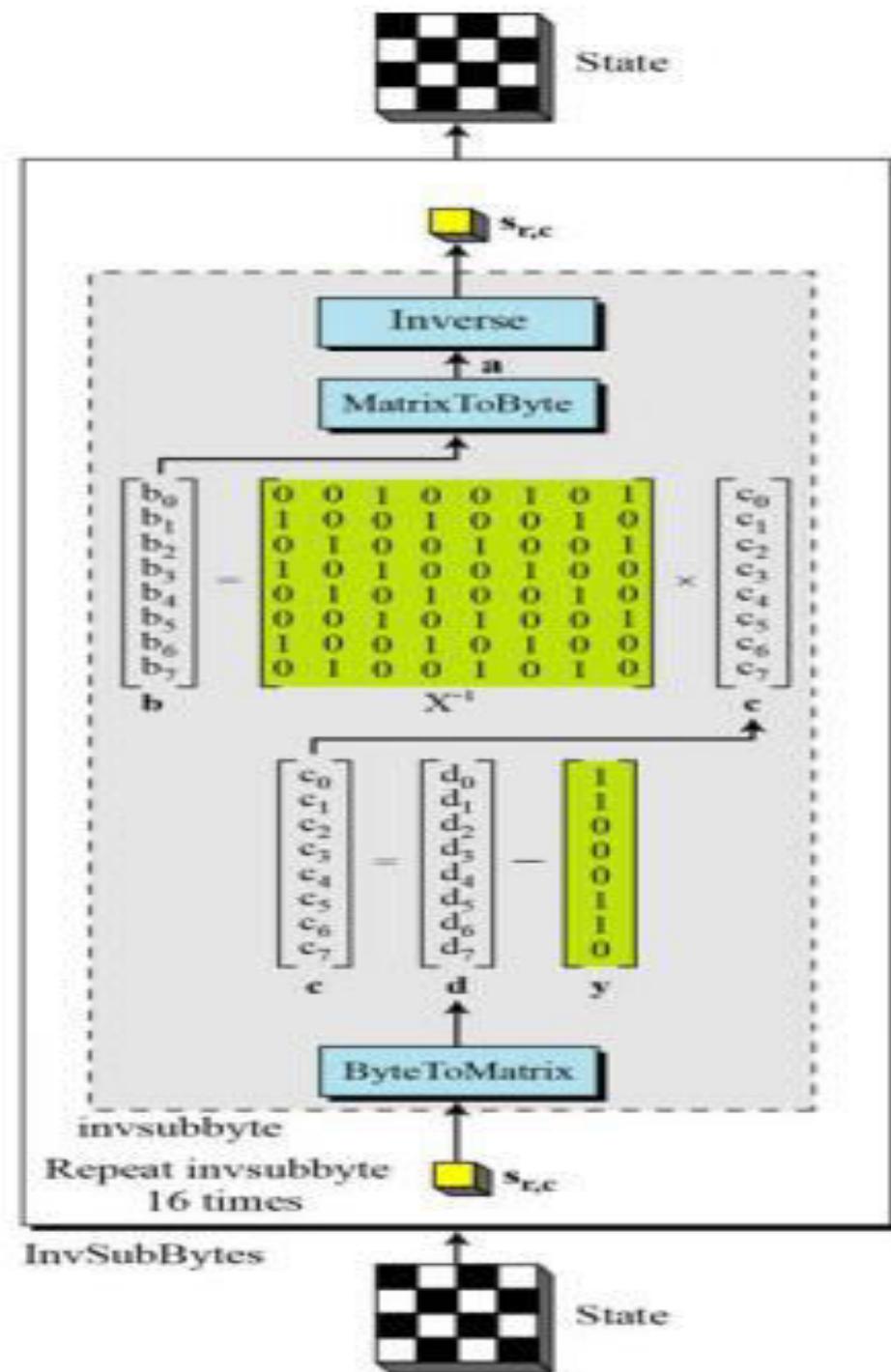
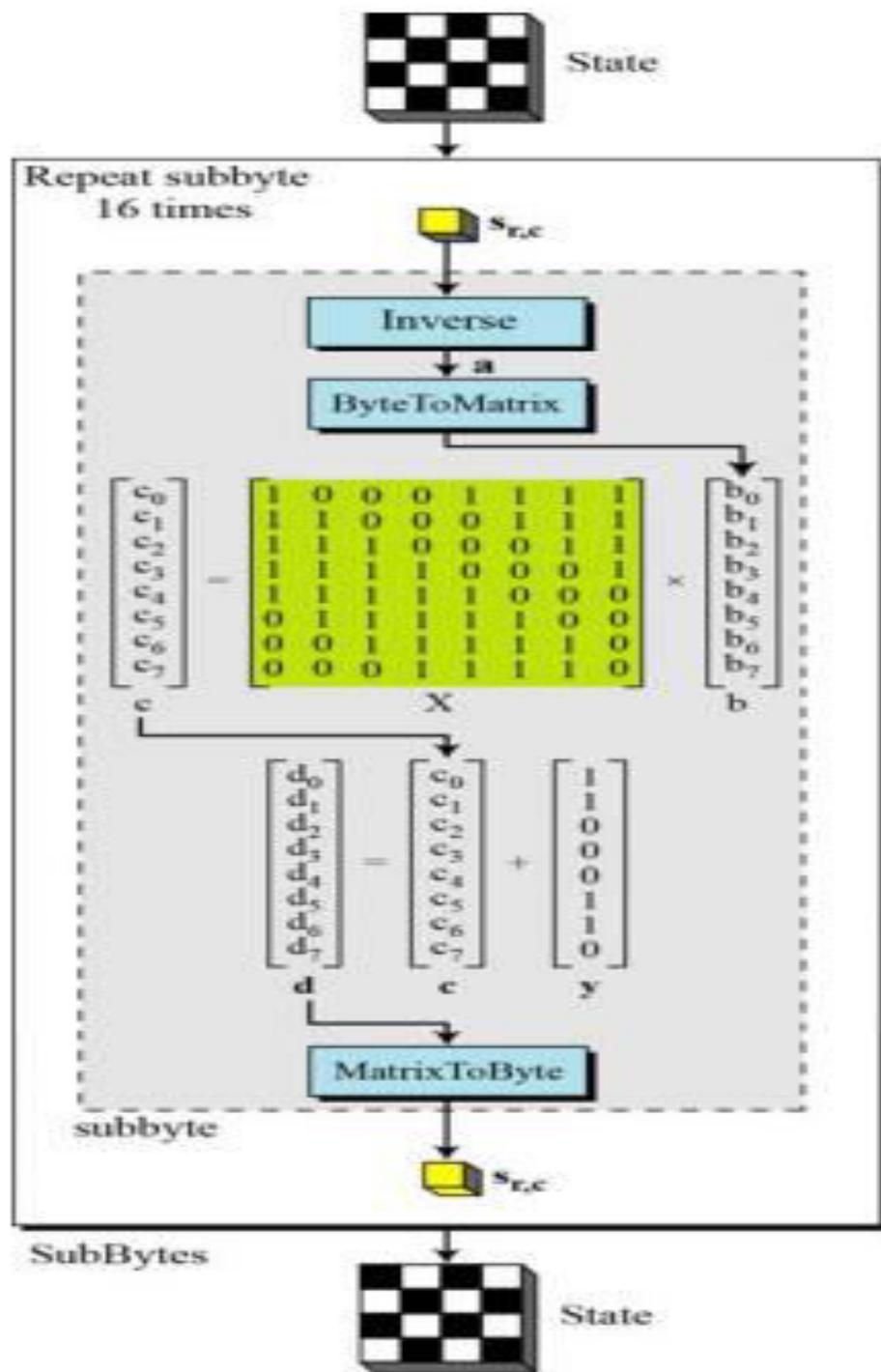
## SubBytes

### *Transformation Using the GF(2<sup>8</sup>) Field*

*AES also defines the transformation algebraically using the GF(28) field with the irreducible polynomials  $(x^8 + x^4 + x^3 + x + 1)$*

$$\text{subbyte: } \rightarrow \mathbf{d} = \mathbf{X} (s_{r,c})^{-1} \oplus \mathbf{y}$$

$$\text{invsubbyte: } \rightarrow [\mathbf{X}^{-1}(\mathbf{d} \oplus \mathbf{y})]^{-1} = [\mathbf{X}^{-1}(\mathbf{X} (s_{r,c})^{-1} \oplus \mathbf{y} \oplus \mathbf{y})]^{-1} = [(s_{r,c})^{-1}]^{-1} = s_{r,c}$$



# Transformation

*Pseudocode for SubBytes transformation*

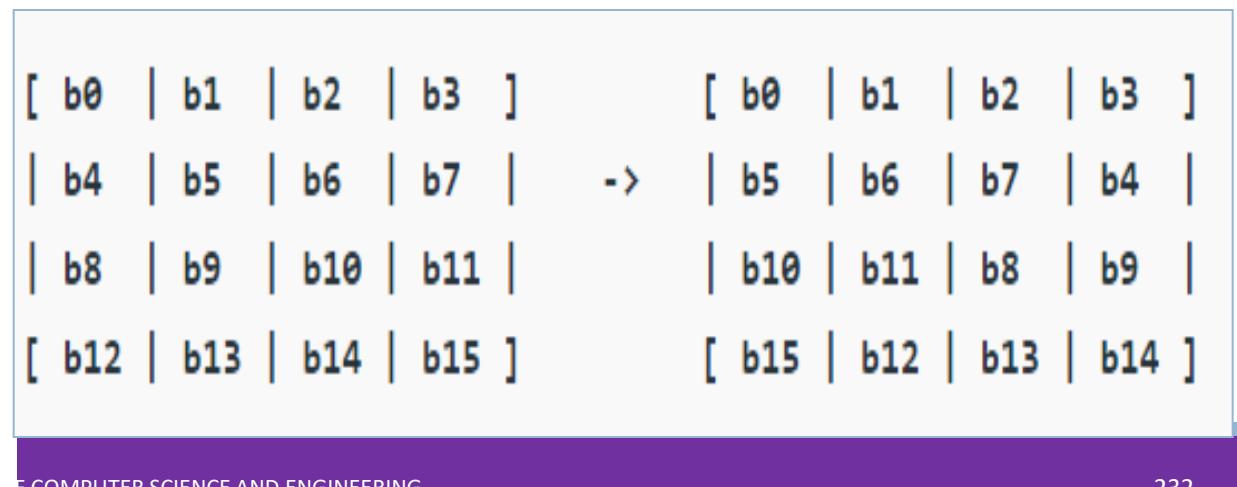
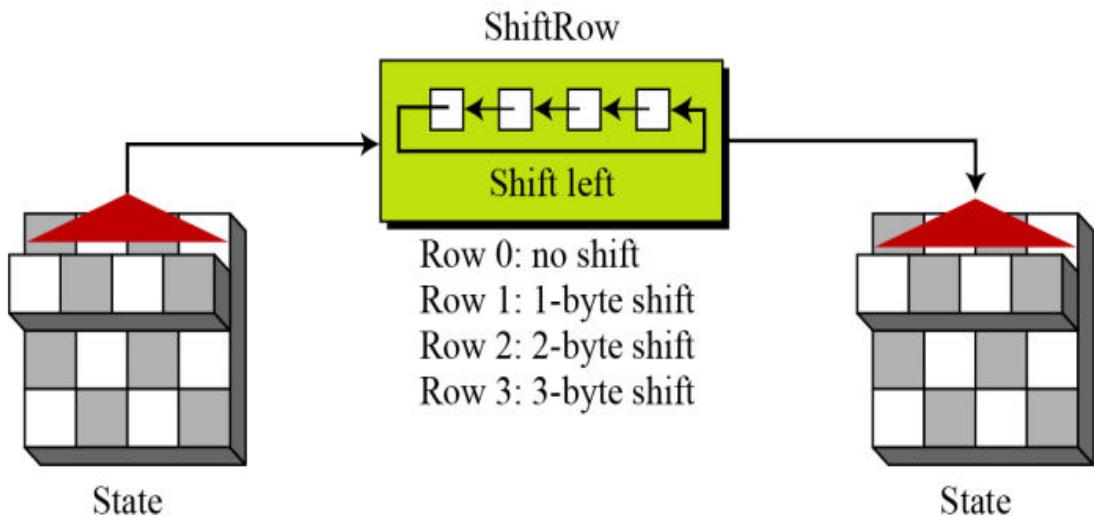
```
SubBytes (S)
{
    for (r = 0 to 3)
        for (c = 0 to 3)
            Sr,c = subbyte (Sr,c)
}

subbyte (byte)
{
    a ← byte-1          //Multiplicative inverse in GF(28) with inverse of 00 to be 00
    ByteToMatrix (a, b)
    for (i = 0 to 7)
    {
        ci ← bi ⊕ b(i+4)mod 8 ⊕ b(i+5)mod 8 ⊕ b(i+6)mod 8 ⊕ b(i+7)mod 8
        di ← ci ⊕ ByteToMatrix (0x63)
    }
    MatrixToByte (d, d)
    byte ← d
}
```

# Transformation

## ShiftRows (Permutation)

- Another transformation found in a round is shifting, which permutes the bytes.
- Each row is shifted a particular number of times.



# Transformation

## ShiftRows (Permutation)

- InvShiftRows In the decryption, the transformation is called InvShiftRows and the shifting is to the right.

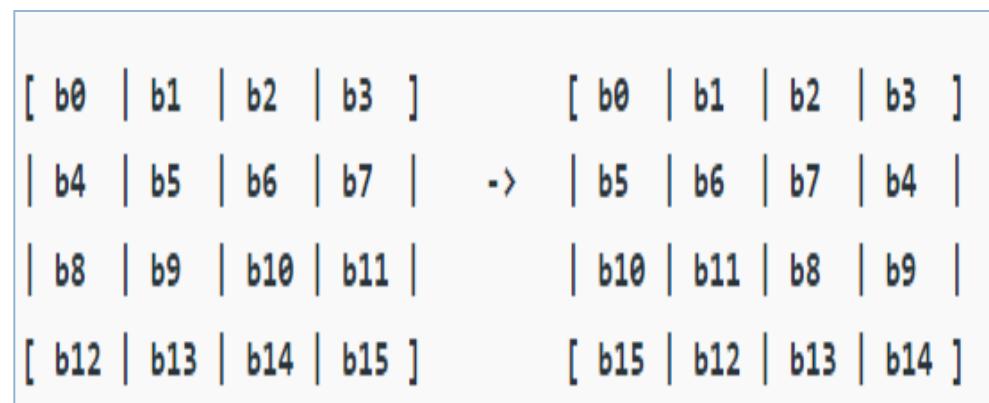
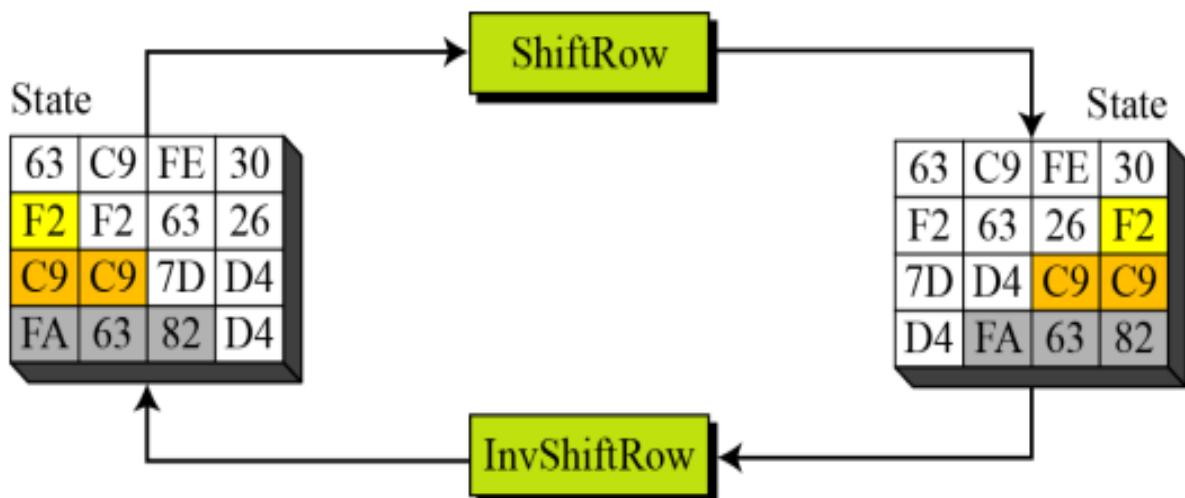
*Pseudocode for ShiftRows transformation*

```
ShiftRows (S)
{
    for ( $r = 1$  to  $3$ )
        shiftrow ( $s_r$ ,  $r$ )          //  $s_r$  is the  $r$ th row
}
shiftrow (row,  $n$ )           //  $n$  is the number of bytes to be shifted
{
    CopyRow (row, t)          // t is a temporary row
    for ( $c = 0$  to  $3$ )
        row( $c - n$ ) mod 4  $\leftarrow$  t $c$ 
}
```

# Transformation

## ShiftRows (Permutation)

**shows how a state is transformed using ShiftRows transformation. The figure also shows that InvShiftRows transformation creates the original state.**



# Transformation

## MixColumns (Mixing)

- This step is basically a matrix multiplication.
- Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.

*Mixing bytes using matrix multiplication*

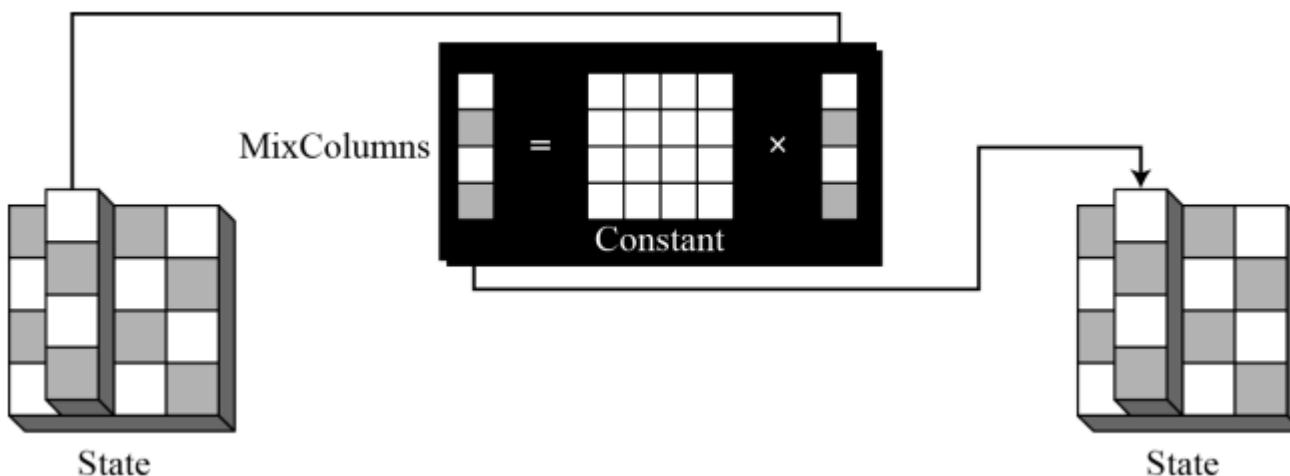
$$\begin{array}{l}
 ax + by + cz + dt \\
 ex + fy + gz + ht \\
 ix + jy + kz + lt \\
 mx + ny + oz + pt
 \end{array} \xrightarrow{\text{New matrix}} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix}$$

**Constant matrix**

# Transformation

## MixColumns (Mixing)

- The MixColumns transformation operates at the column level; it transforms each column of the state to a new column.



# Transformation

## MixColumns (Mixing)

*Pseudocode for MixColumns transformation*

```
MixColumns (S)
{
    for (c = 0 to 3)
        mixcolumn (sc)
}

mixcolumn (col)
{
    CopyColumn (col, t)          // t is a temporary column

    col0 ← (0x02) • t0 ⊕ (0x03 • t1) ⊕ t2 ⊕ t3

    col1 ← t0 ⊕ (0x02) • t1 ⊕ (0x03) • t2 ⊕ t3

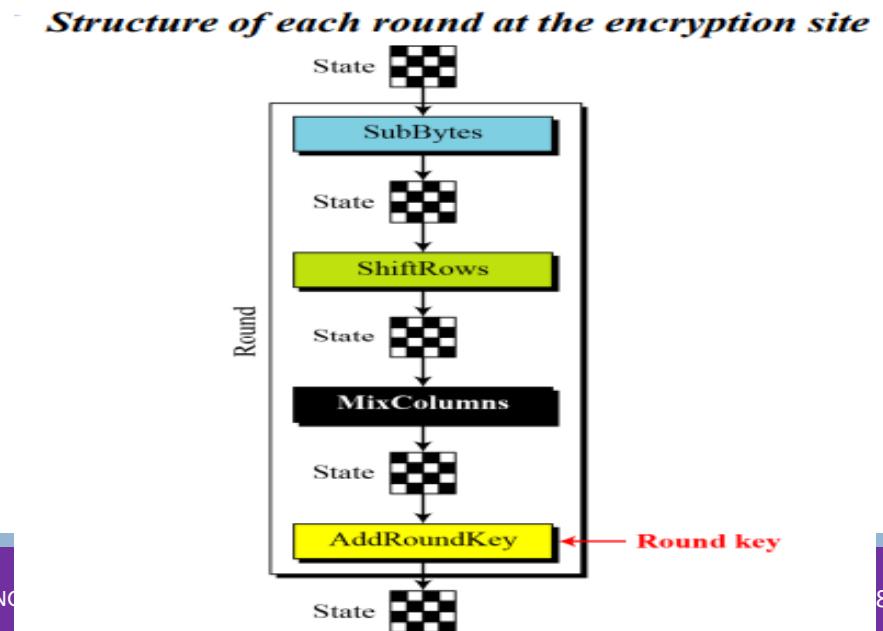
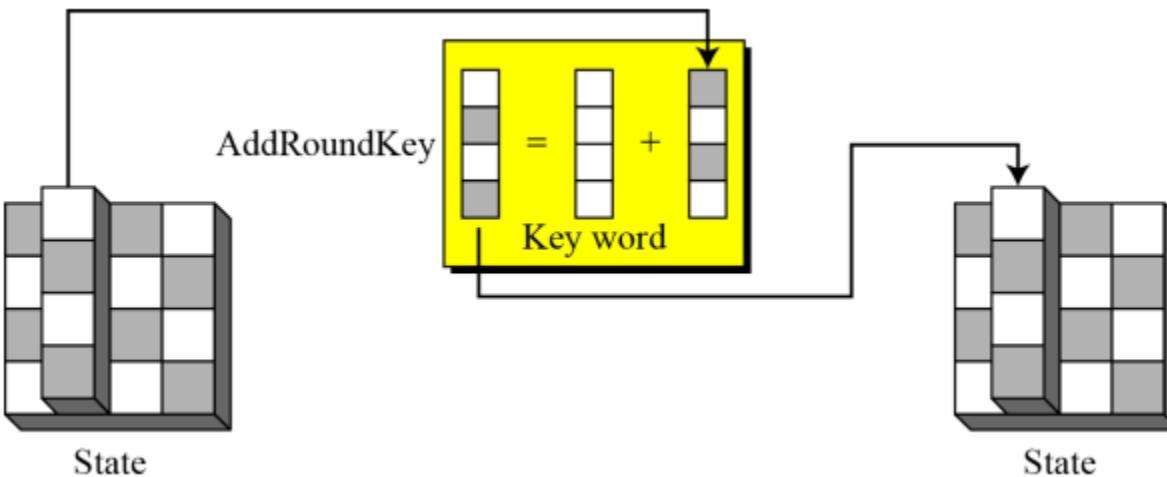
    col2 ← t0 ⊕ t1 ⊕ (0x02) • t2 ⊕ (0x03) • t3

    col3 ← (0x03 • t0) ⊕ t1 ⊕ t2 ⊕ (0x02) • t3
}
```

# Transformation

## Add Round Keys (Adding)

- AddRoundKey proceeds one column at a time.
- AddRoundKey adds a round key word with each state column matrix; the operation in AddRoundKey is matrix addition.



# Transformation

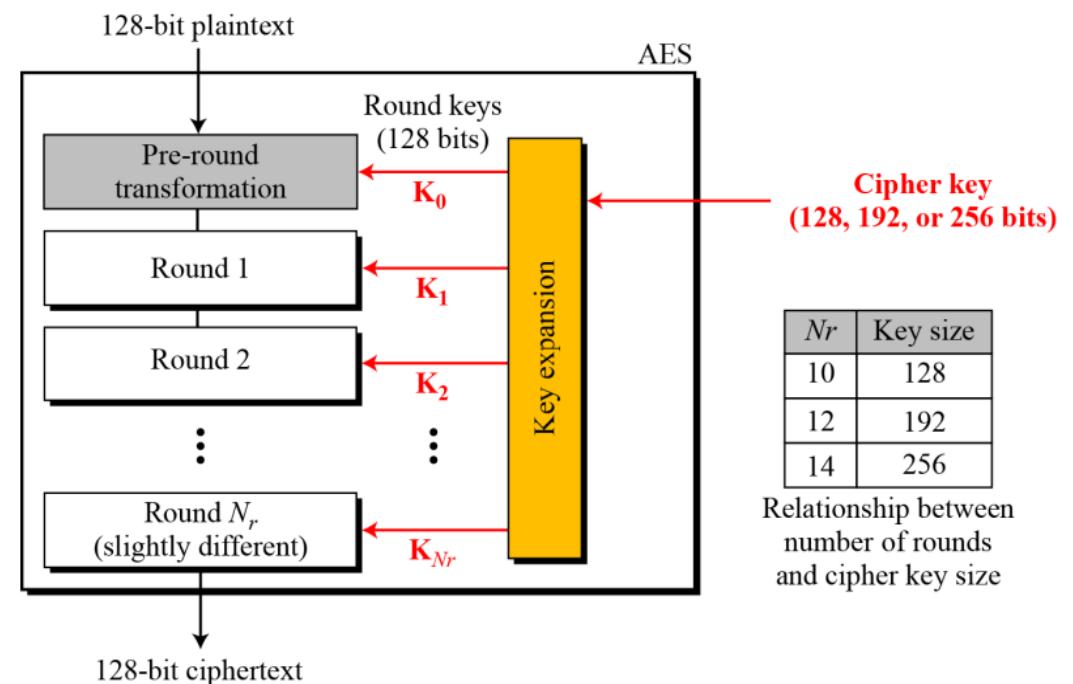
## Add Round Keys (Adding)

*Pseudocode for AddRoundKey transformation*

```
AddRoundKey (S)
{
    for (c = 0 to 3)
         $s_c \leftarrow s_c \oplus w_{\text{round} + 4c}$ 
}
```

# Key Expansion

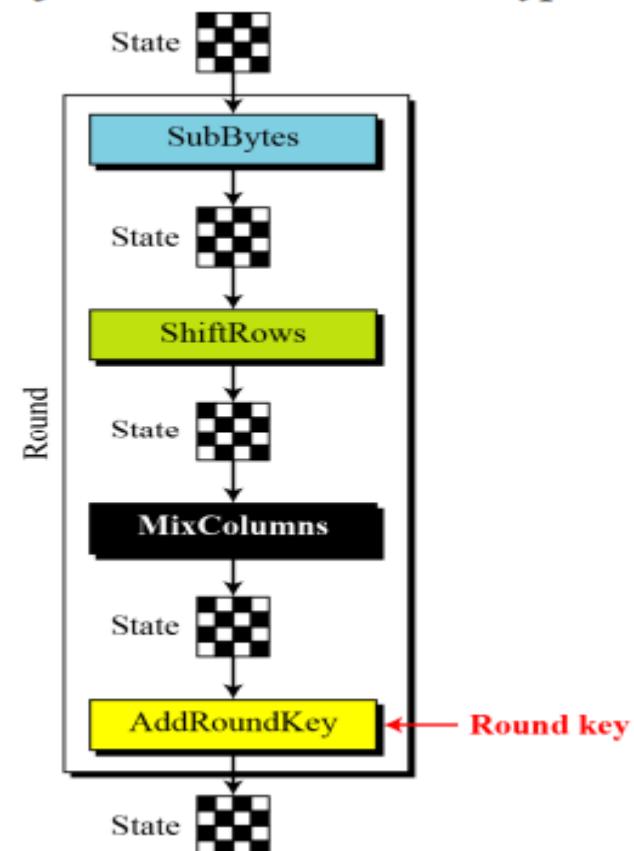
- To create round keys for each round, AES uses a key-expansion process.
- If the number of rounds is  $N_r$ , the key-expansion routine creates  $N_r + 1$  128-bit round keys from one single 128-bit cipher key



# Key Expansion

- First round key is used for pre-round transformation
- Remaining all for every round 4<sup>th</sup> transformation

*Structure of each round at the encryption site*



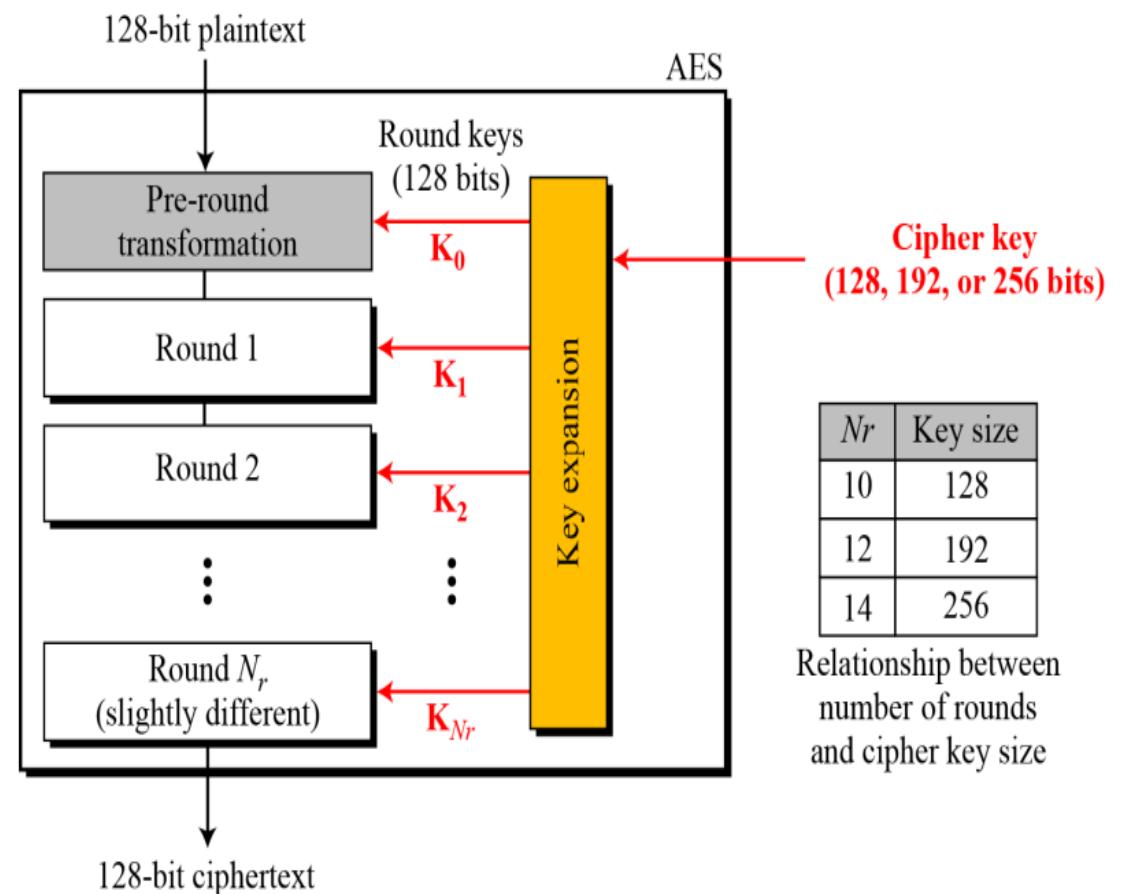
# Key Expansion

Key-expansion creates round key word by word, where a word is an array of 4 bytes.

- Key Expansion in AES-128
- Key Expansion in AES-192 and AES-256
- Key-Expansion Analysis

*Words for each round*

Round	Words			
Pre-round	$w_0$	$w_1$	$w_2$	$w_3$
1	$w_4$	$w_5$	$w_6$	$w_7$
2	$w_8$	$w_9$	$w_{10}$	$w_{11}$
...	...			
$N_r$	$w_{4N_r}$	$w_{4N_r+1}$	$w_{4N_r+2}$	$w_{4N_r+3}$



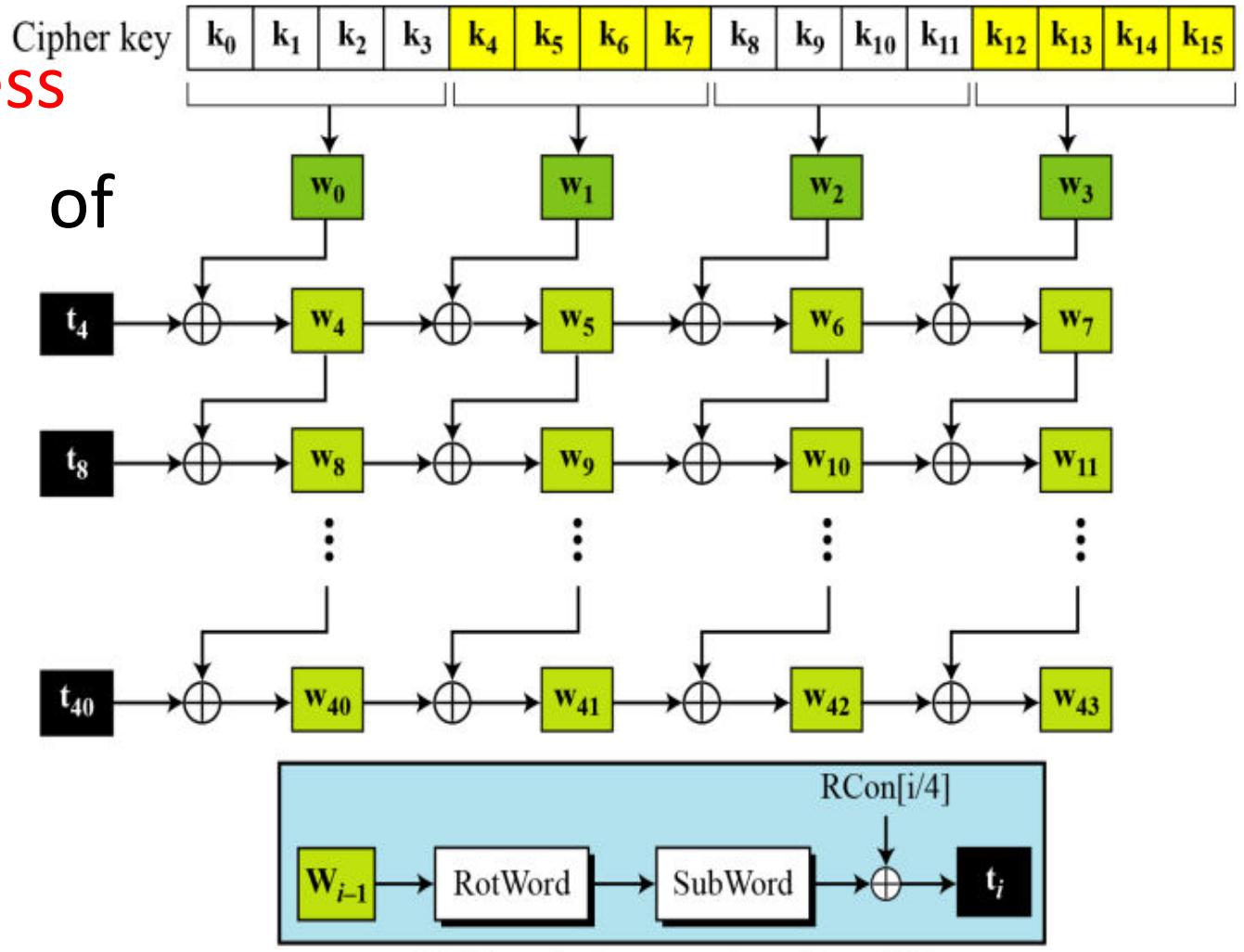
# Key Expansion

## Key Expansion in AES-128 process

1. Cipher key is an array of 16bytes(k0 to k15)

The first 4 words( $w_0, w_1, w_2, w_3$ ) are made from cipher key

- K0 to k3 ->  $w_0$
- k4 to k7 ->  $w_1$
- K8 to k11 -> $w_2$
- K12 to k15 -> $w_3$



# Key Expansion

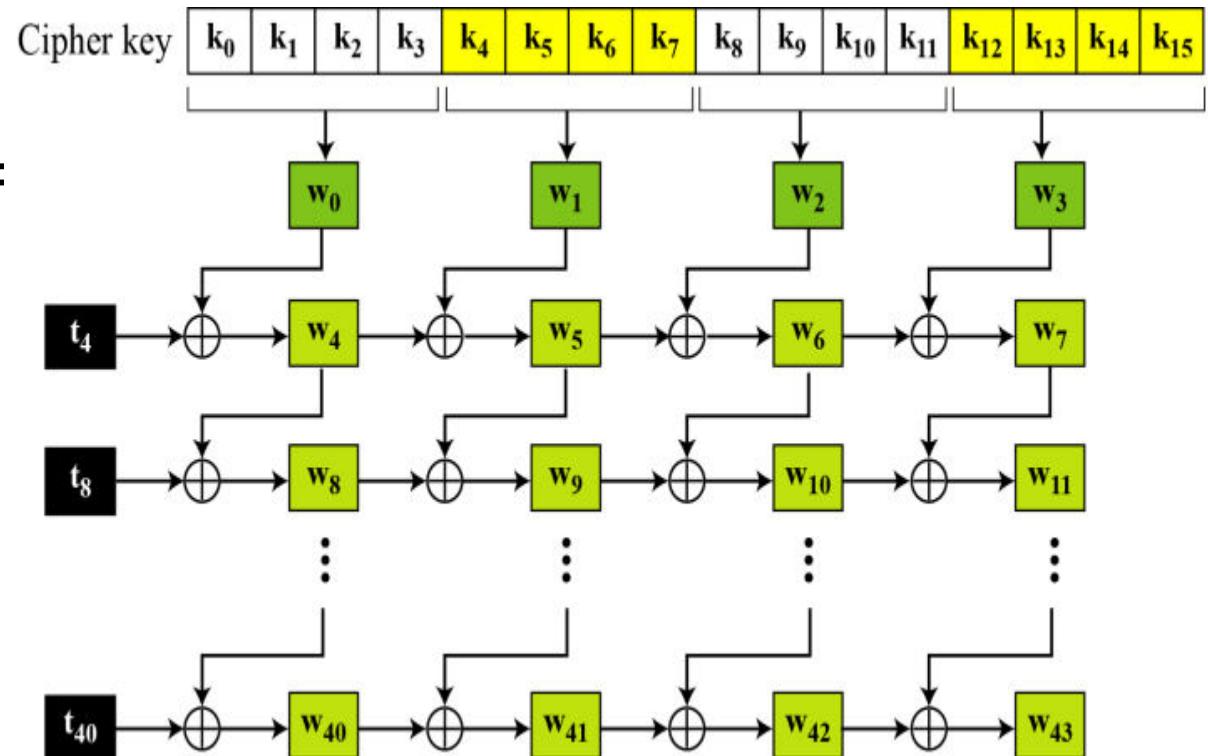
## Key Expansion in AES-128 process

2. The rest of the words ( $w_i$  for  $i > 3$ ) are made as follows

- i) if  $(i \bmod 4) \neq 0$ ,  $w_i = w_{i-1} \oplus w_{i-4}$
- ii) if  $(i \bmod 4) = 0$ ,  $w_i = t \oplus w_{i-4}$

Temporary word  $t$

$$t_i = \text{subword}(\text{Rotword}(w_{i-1})) \oplus Rcon_{i/4}$$



# Key Expansion

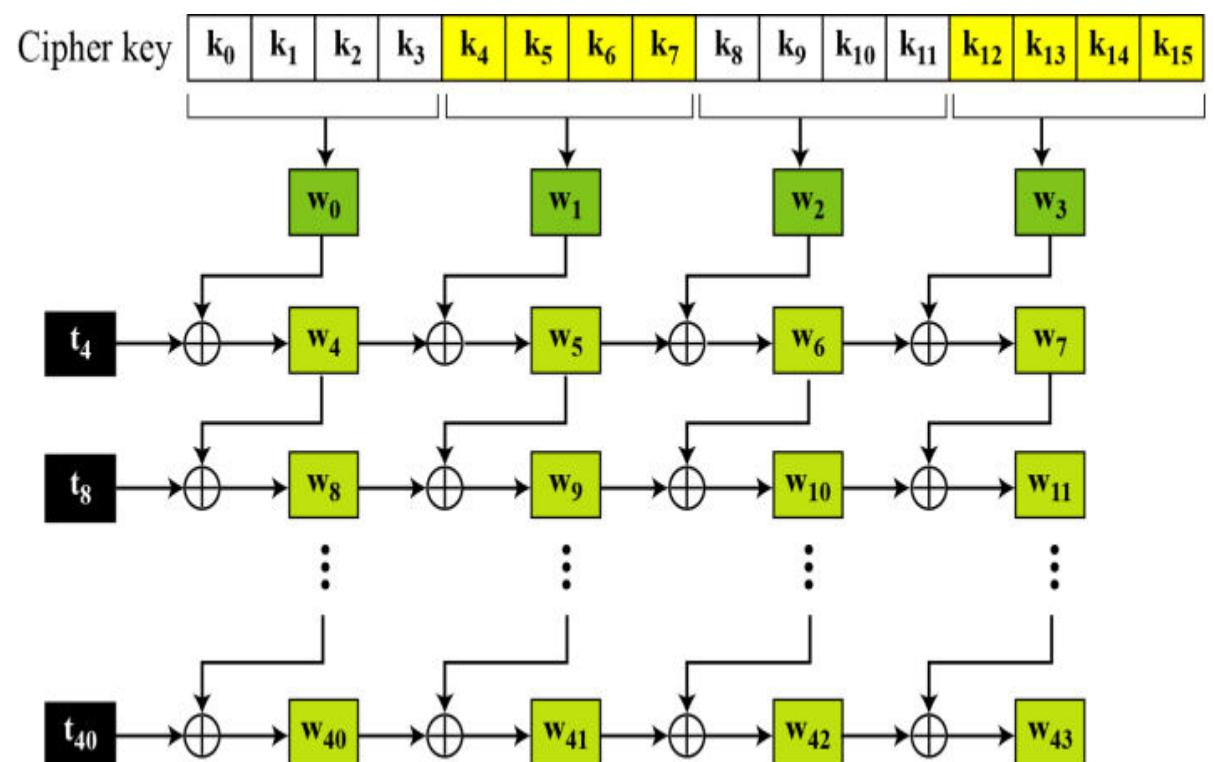
$$t_i = \text{subword}(\text{Rotword}(w_{i-1})) \oplus Rcon_{i/4}$$

**Rotword** : Applied to only one row

Rotate word routine takes a word as an array of 4bytes and shifts each byte to the left with wrapping.

**Subword** : Applied to 4 bytes.

Substitute word routine takes each byte in the word and substitute another byte for it.

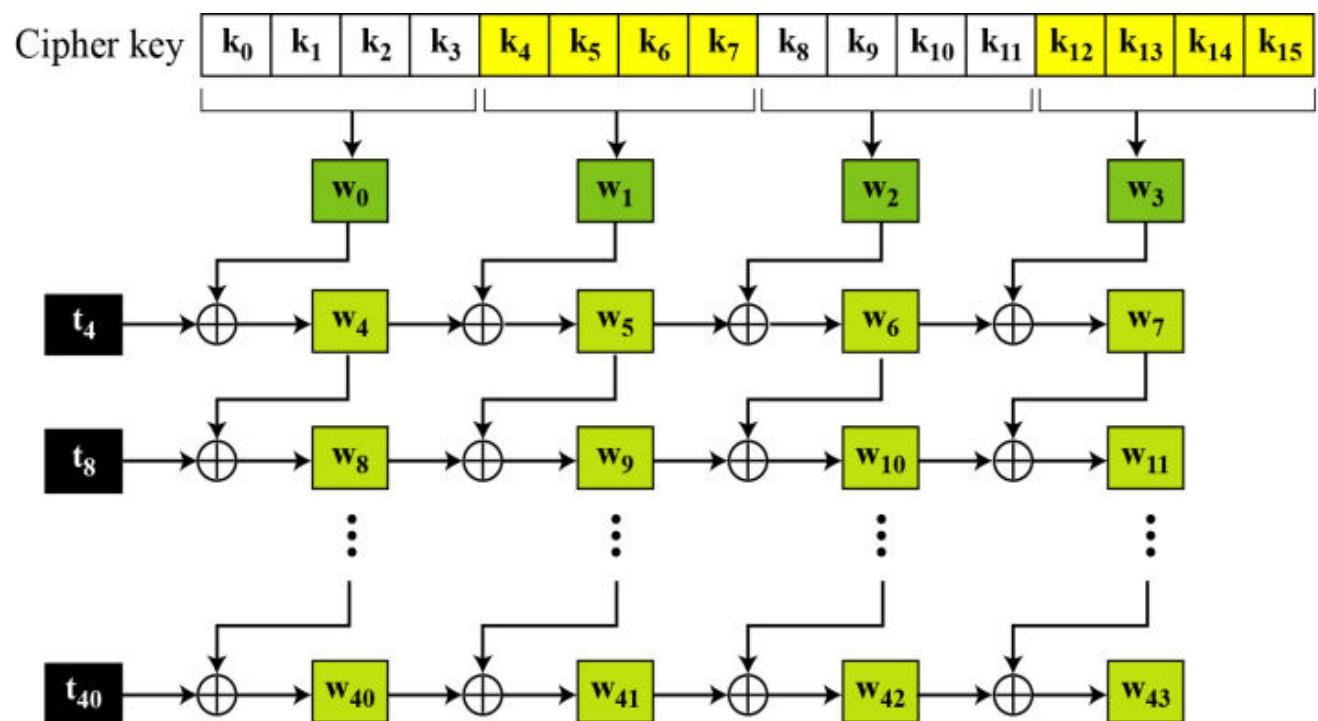


# Key Expansion

$$t_i = \text{subword}(\text{Rotword}(w_{i-1})) \oplus Rcon_{i/4}$$

Rcon : Round constant is a 4byte value in which the rightmost 3bytes are always zero

Round	Constant (RCon)	Round	Constant (RCon)
1	( <u>01</u> 00 00 00) <sub>16</sub>	6	( <u>20</u> 00 00 00) <sub>16</sub>
2	( <u>02</u> 00 00 00) <sub>16</sub>	7	( <u>40</u> 00 00 00) <sub>16</sub>
3	( <u>04</u> 00 00 00) <sub>16</sub>	8	( <u>80</u> 00 00 00) <sub>16</sub>
4	( <u>08</u> 00 00 00) <sub>16</sub>	9	( <u>1B</u> 00 00 00) <sub>16</sub>
5	( <u>10</u> 00 00 00) <sub>16</sub>	10	( <u>36</u> 00 00 00) <sub>16</sub>



# Key Expansion

The key-expansion routine can either use the table when calculating the words or use the GF(2<sup>8</sup>) field to calculate the leftmost byte dynamically, as shown below

RC <sub>1</sub>	$\rightarrow x^{1-1}$	$=x^0$	mod prime	$= 1$	$\rightarrow 00000001$	$\rightarrow 01_{16}$
RC <sub>2</sub>	$\rightarrow x^{2-1}$	$=x^1$	mod prime	$= x$	$\rightarrow 00000010$	$\rightarrow 02_{16}$
RC <sub>3</sub>	$\rightarrow x^{3-1}$	$=x^2$	mod prime	$= x^2$	$\rightarrow 00000100$	$\rightarrow 04_{16}$
RC <sub>4</sub>	$\rightarrow x^{4-1}$	$=x^3$	mod prime	$= x^3$	$\rightarrow 00001000$	$\rightarrow 08_{16}$
RC <sub>5</sub>	$\rightarrow x^{5-1}$	$=x^4$	mod prime	$= x^4$	$\rightarrow 00010000$	$\rightarrow 10_{16}$
RC <sub>6</sub>	$\rightarrow x^{6-1}$	$=x^5$	mod prime	$= x^5$	$\rightarrow 00100000$	$\rightarrow 20_{16}$
RC <sub>7</sub>	$\rightarrow x^{7-1}$	$=x^6$	mod prime	$= x^6$	$\rightarrow 01000000$	$\rightarrow 40_{16}$
RC <sub>8</sub>	$\rightarrow x^{8-1}$	$=x^7$	mod prime	$= x^7$	$\rightarrow 10000000$	$\rightarrow 80_{16}$
RC <sub>9</sub>	$\rightarrow x^{9-1}$	$=x^8$	mod prime	$= x^4 + x^3 + x + 1$	$\rightarrow 00011011$	$\rightarrow 1B_{16}$
RC <sub>10</sub>	$\rightarrow x^{10-1}$	$=x^9$	mod prime	$= x^5 + x^4 + x^2 + x$	$\rightarrow 00110110$	$\rightarrow 36_{16}$

# Key Expansion

## Key Expansion in AES-128 - Algorithm

```
KeyExpansion ([key0 to key15], [w0 to w43])
{
    for (i = 0 to 3)
        wi ← key4i + key4i+1 + key4i+2 + key4i+3

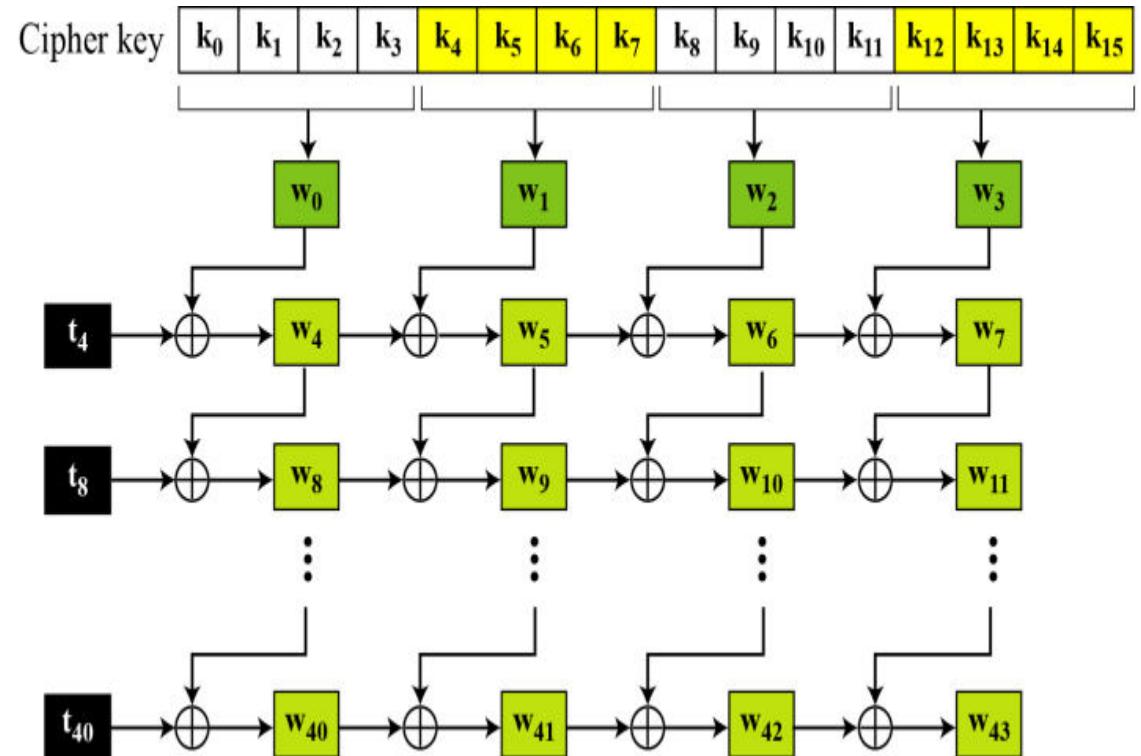
    for (i = 4 to 43)
    {
        if (i mod 4 ≠ 0)    wi ← wi-1 + wi-4
        else
        {
            t ← SubWord (RotWord (wi-1)) ⊕ RConi/4
            wi ← t + wi-4
        }
    }
}
```

# Key Expansion

Table 7.5 shows how the keys for each round are calculated assuming that the 128-bit cipher key agreed upon by Alice and Bob is  $(24\ 75\ A2\ B3\ 34\ 75\ 56\ 88\ 31\ E2\ 12\ 00\ 13\ AA\ 54\ 87)_{16}$ .

**Table 7.5** Key expansion example

Round	Values of t's	First word in the round	Second word in the round	Third word in the round	Fourth word in the round
—		$w_{00} = 2475A2B3$	$w_{01} = 34755688$	$w_{02} = 31E21200$	$w_{03} = 13AA5487$
1	AD20177D	$w_{04} = 8955B5CE$	$w_{05} = BD20E346$	$w_{06} = 8CC2F146$	$w_{07} = 9F68A5C1$
2	470678DB	$w_{08} = CE53CD15$	$w_{09} = 73732E53$	$w_{10} = FFB1DF15$	$w_{11} = 60D97AD4$
3	31DA48D0	$w_{12} = FF8985C5$	$w_{13} = 8CFAAB96$	$w_{14} = 734B7483$	$w_{15} = 2475A2B3$
4	47AB5B7D	$w_{16} = B822deb8$	$w_{17} = 34D8752E$	$w_{18} = 479301AD$	$w_{19} = 54010FFA$
5	6C762D20	$w_{20} = D454F398$	$w_{21} = E08C86B6$	$w_{22} = A71F871B$	$w_{23} = F31E88E1$
6	52C4F80D	$w_{24} = 86900B95$	$w_{25} = 661C8D23$	$w_{26} = C1030A38$	$w_{27} = 321D82D9$
7	E4133523	$w_{28} = 62833EB6$	$w_{29} = 049FB395$	$w_{30} = C59CB9AD$	$w_{31} = F7813B74$
8	8CE29268	$w_{32} = EE61ACDE$	$w_{33} = EAFe1F4B$	$w_{34} = 2F62A6E6$	$w_{35} = D8E39D92$
9	0A5E4F61	$w_{36} = E43FE3BF$	$w_{37} = 0EC1FCF4$	$w_{38} = 21A35A12$	$w_{39} = F940C780$
10	3FC6CD99	$w_{40} = DBF92E26$	$w_{41} = D538D2D2$	$w_{42} = F49B88C0$	$w_{43} = 0DDB4F40$



# Key Expansion

Table 7.5 shows how the keys for each round are calculated assuming that the 128-bit cipher key agreed upon by Alice and Bob is  $(24\ 75\ A2\ B3\ 34\ 75\ 56\ 88\ 31\ E2\ 12\ 00\ 13\ AA\ 54\ 87)_{16}$ .

Table 7.5 Key expansion example

Round	Values of t's	First word in the round	Second word in the round	Third word in the round	Fourth word in the round
—		$w_{00} = 2475A2B3$	$w_{01} = 34755688$	$w_{02} = 31E21200$	$w_{03} = 13AA5487$
1	AD20177D	$w_{04} = 8955B5CE$	$w_{05} = BD20E346$	$w_{06} = 8CC2F146$	$w_{07} = 9F68A5C1$
2	470678DB	$w_{08} = CE53CD15$	$w_{09} = 73732E53$	$w_{10} = FFB1DF15$	$w_{11} = 60D97AD4$
3	31DA48D0	$w_{12} = FF8985C5$	$w_{13} = 8CFAAB96$	$w_{14} = 734B7483$	$w_{15} = 2475A2B3$
4	47AB5B7D	$w_{16} = B822deb8$	$w_{17} = 34D8752E$	$w_{18} = 479301AD$	$w_{19} = 54010FFA$
5	6C762D20	$w_{20} = D454F398$	$w_{21} = E08C86B6$	$w_{22} = A71F871B$	$w_{23} = F31E88E1$
6	52C4F80D	$w_{24} = 86900B95$	$w_{25} = 661C8D23$	$w_{26} = C1030A38$	$w_{27} = 321D82D9$
7	E4133523	$w_{28} = 62833EB6$	$w_{29} = 049FB395$	$w_{30} = C59CB9AD$	$w_{31} = F7813B74$
8	8CE29268	$w_{32} = EE61ACDE$	$w_{33} = EAFe1F4B$	$w_{34} = 2F62A6E6$	$w_{35} = D8E39D92$
9	0A5E4F61	$w_{36} = E43FE3BF$	$w_{37} = 0EC1FCF4$	$w_{38} = 21A35A12$	$w_{39} = F940C780$
10	3FC6CD99	$w_{40} = DBF92E26$	$w_{41} = D538D2D2$	$w_{42} = F49B88C0$	$w_{43} = 0DDB4F40$

Round	Constant (RCon)	Round	Constant (RCon)
1	$(01\ 00\ 00\ 00)_{16}$	6	$(20\ 00\ 00\ 00)_{16}$
2	$(02\ 00\ 00\ 00)_{16}$	7	$(40\ 00\ 00\ 00)_{16}$
3	$(04\ 00\ 00\ 00)_{16}$	8	$(80\ 00\ 00\ 00)_{16}$
4	$(08\ 00\ 00\ 00)_{16}$	9	$(1B\ 00\ 00\ 00)_{16}$
5	$(10\ 00\ 00\ 00)_{16}$	10	$(36\ 00\ 00\ 00)_{16}$

$$t_i = \text{subword}(\text{Rotword}(w_{i-1})) \oplus Rcon_{i/4}$$

$$t_4 = \text{subword}(\text{Rotword}(w_{4-1})) \oplus Rcon_{4/4}$$

$$t_4 = \text{subword}(\text{Rotword}(w_3)) \oplus Rcon_1$$

$$\text{Rotword}(13AA5487) = AA548713$$

$$\text{Subword } (AA548713) = AC20177D$$

$$t_4 = AC20177D \oplus Rcon_1$$

$$= AC20177D \oplus 01\ 00\ 00\ 00 \rightarrow AD20177D$$

# Key Expansion –AES 192 and AES 256

---

Key-expansion algorithms in the AES-192 and AES-256 versions are very similar to the key expansion algorithm in AES-128, with the following differences:

1. In AES-192, the words are generated in groups of six instead of four.
  - a. The cipher key creates the first six words ( $w_0$  to  $w_5$ ).
  - b. If  $i \bmod 6 \neq 0$ ,  $w_i \leftarrow w_{i-1} + w_{i-6}$ ; otherwise,  $w_i \leftarrow t + w_{i-6}$ .
2. In AES-256, the words are generated in groups of eight instead of four.
  - a. The cipher key creates the first eight words ( $w_0$  to  $w_7$ ).
  - b. If  $i \bmod 8 \neq 0$ ,  $w_i \leftarrow w_{i-1} + w_{i-8}$ ; otherwise,  $w_i \leftarrow t + w_{i-8}$ .
  - c. If  $i \bmod 4 = 0$ , but  $i \bmod 8 \neq 0$ , then  $w_i = \text{SubWord}(w_{i-1}) + w_{i-8}$ .

# Key expansion analysis

---

The key-expansion mechanism in AES has been designed to provide several features that thwart the cryptanalyst.

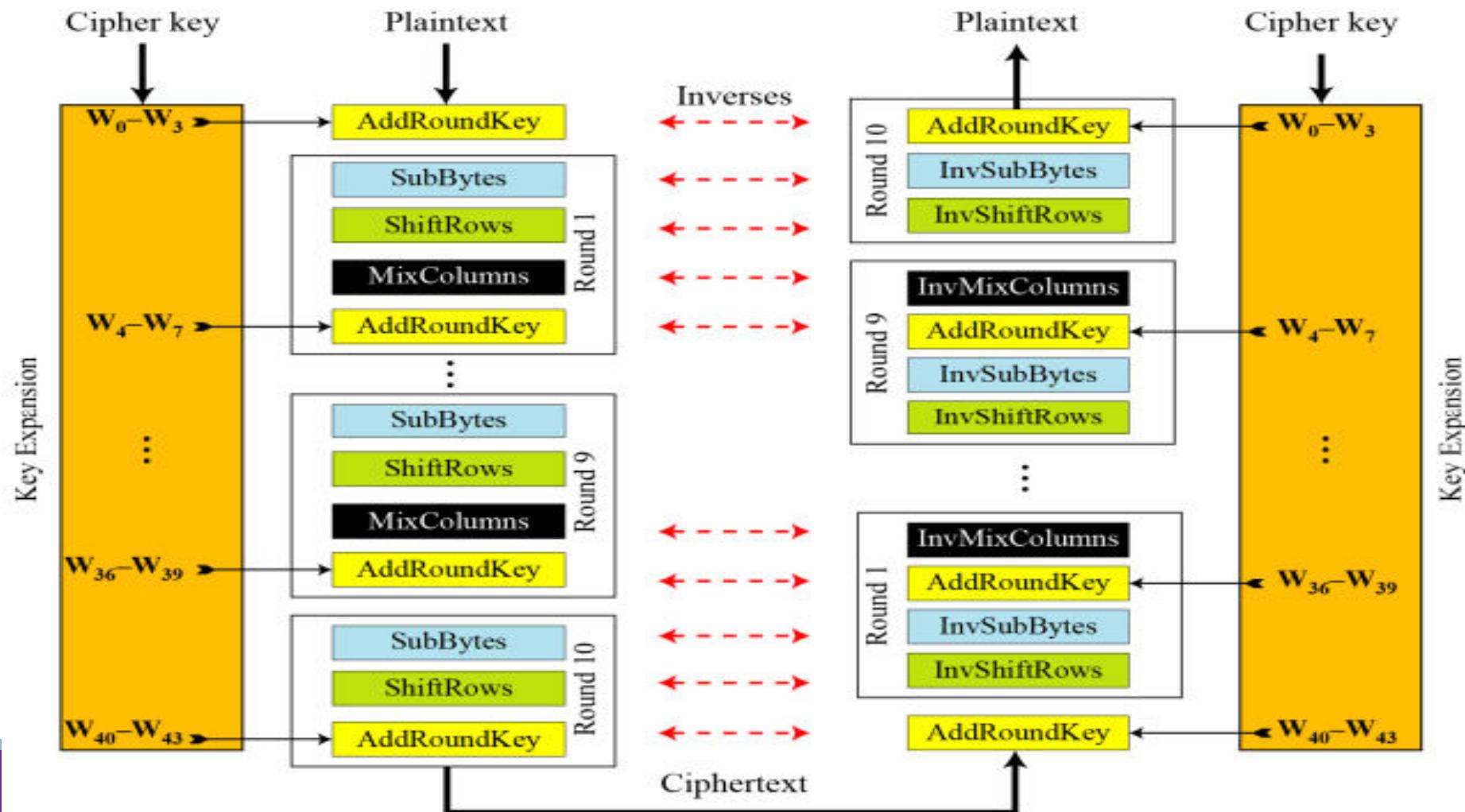
1. Even if Eve knows only part of the cipher key or the values of the words in some round keys, she still needs to find the rest of the cipher key before she can find all round keys. This is because of the nonlinearity produced by SubWord transformation in the key-expansion process.
2. Two different cipher keys, no matter how similar to each other, produce two expansions that differ in at least a few rounds.
3. Each bit of the cipher key is diffused into several rounds. For example, changing a single bit in the cipher key, will change some bits in several rounds.
4. The use of the constants, the RCons, removes any symmetry that may have been created by the other transformations.
5. There are no serious weak keys in AES, unlike in DES.
6. The key-expansion process can be easily implemented on all platforms.
7. The key-expansion routine can be implemented without storing a single table; all calculations can be done using the  $GF(2^8)$  and  $FG(2)$  fields.

# The AES Ciphers

---

- AES uses four types of transformations for encryption and decryption.
- Encryption algorithm is referred to as the cipher
- Decryption algorithm as the inverse cipher.
- Two different design for implementation
  - Original Design
  - Alternative Design

# The AES Ciphers – Original design



# The AES Ciphers –Original design

---

*The code for the AES-128 version of this design is shown in Algorithm*

*Pseudocode for cipher in the original design*

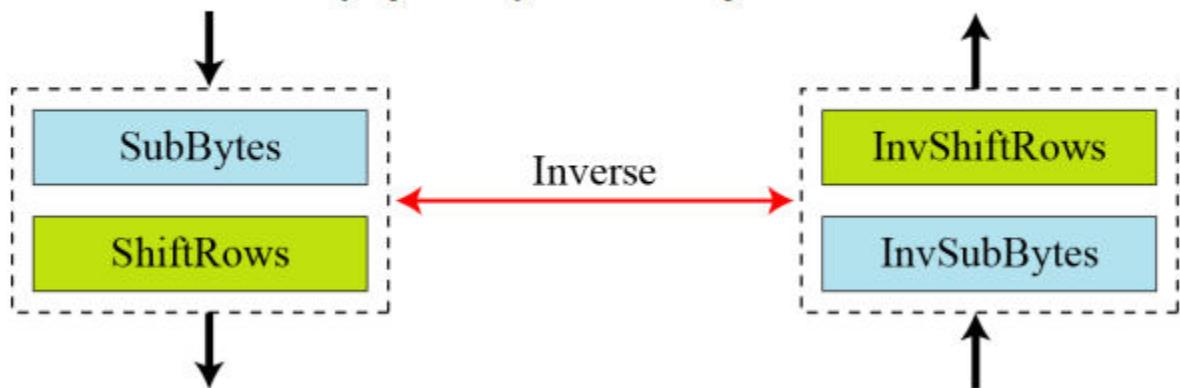
```
Cipher (InBlock [16], OutBlock[16], w[0 ... 43])
{
    BlockToState (InBlock, S)

    S ← AddRoundKey (S, w[0...3])
    for (round = 1 to 10)
    {
        S ← SubBytes (S)
        S ← ShiftRows (S)
        if (round ≠ 10) S ← MixColumns (S)
        S ← AddRoundKey (S, w[4 × round, 4 × round + 3])
    }

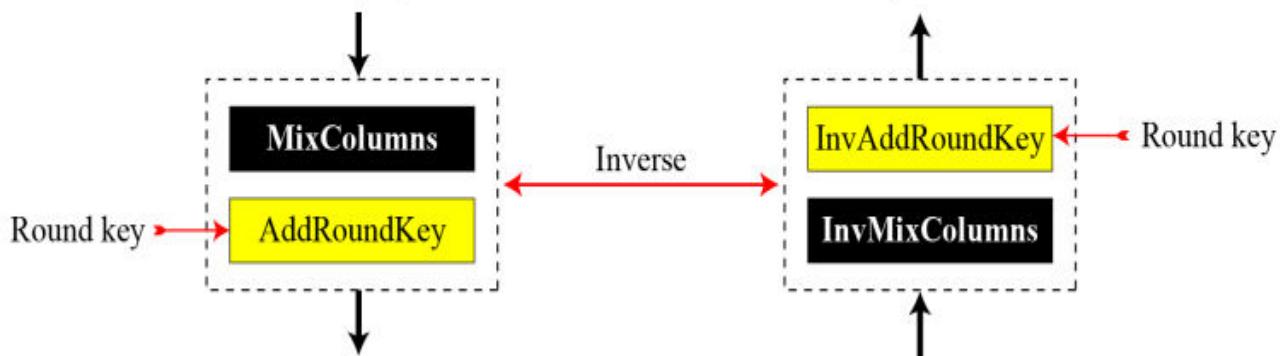
    StateToBlock (S, OutBlock);
}
```

# The AES Ciphers –Alternative Design

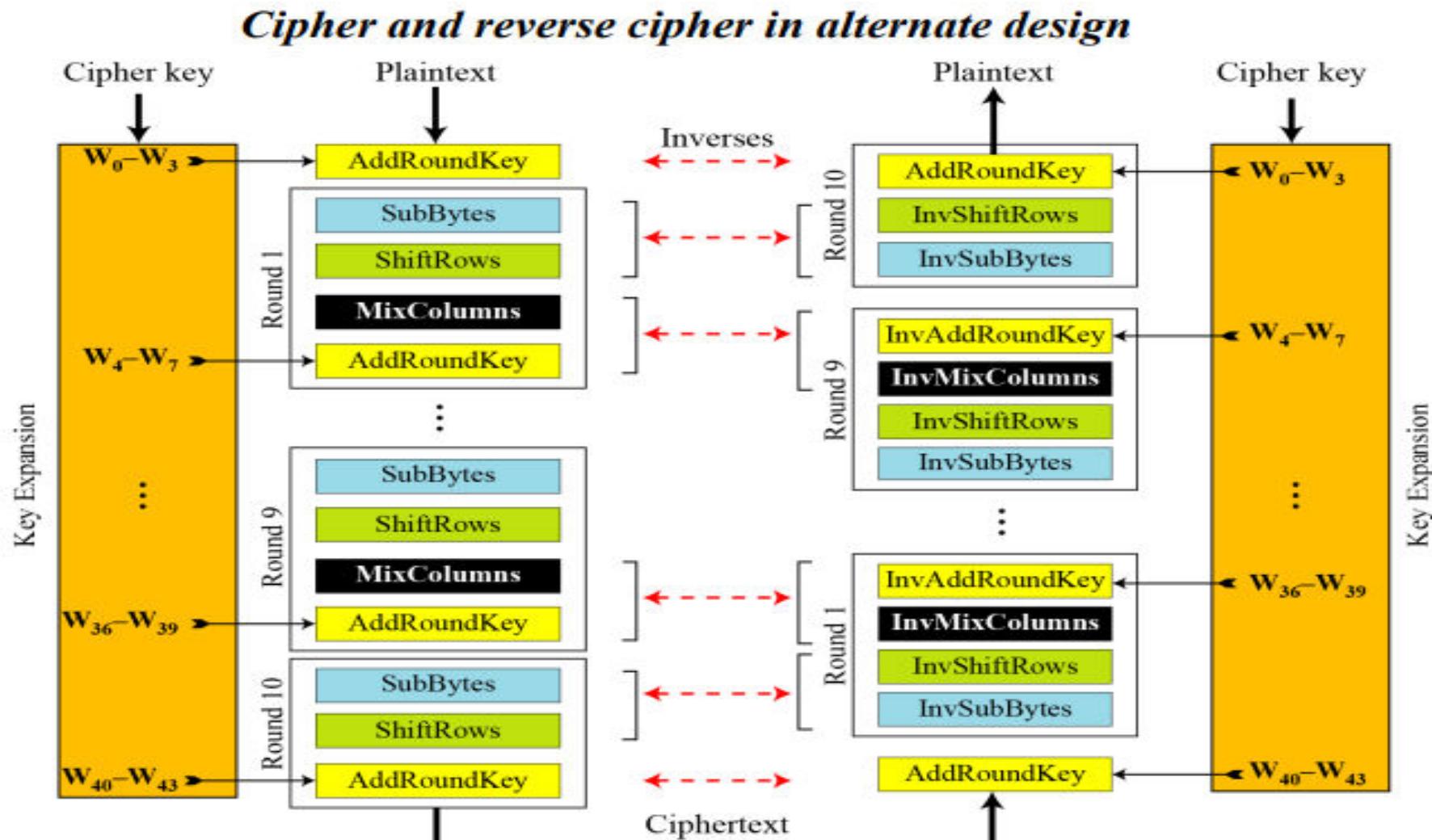
*Invertibility of SubBytes and ShiftRows combinations*



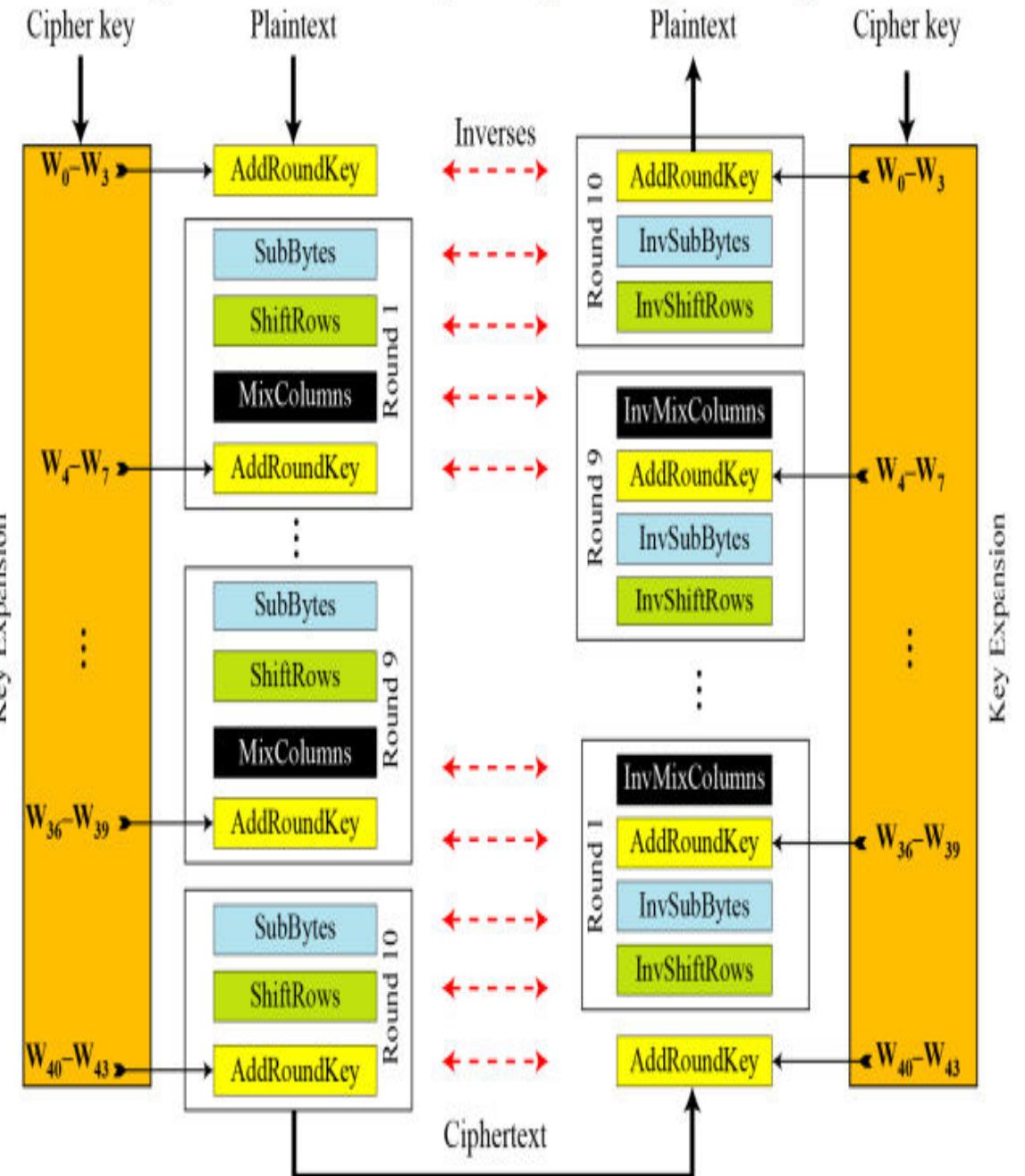
*Invertibility of MixColumns and AddRoundKey combination*



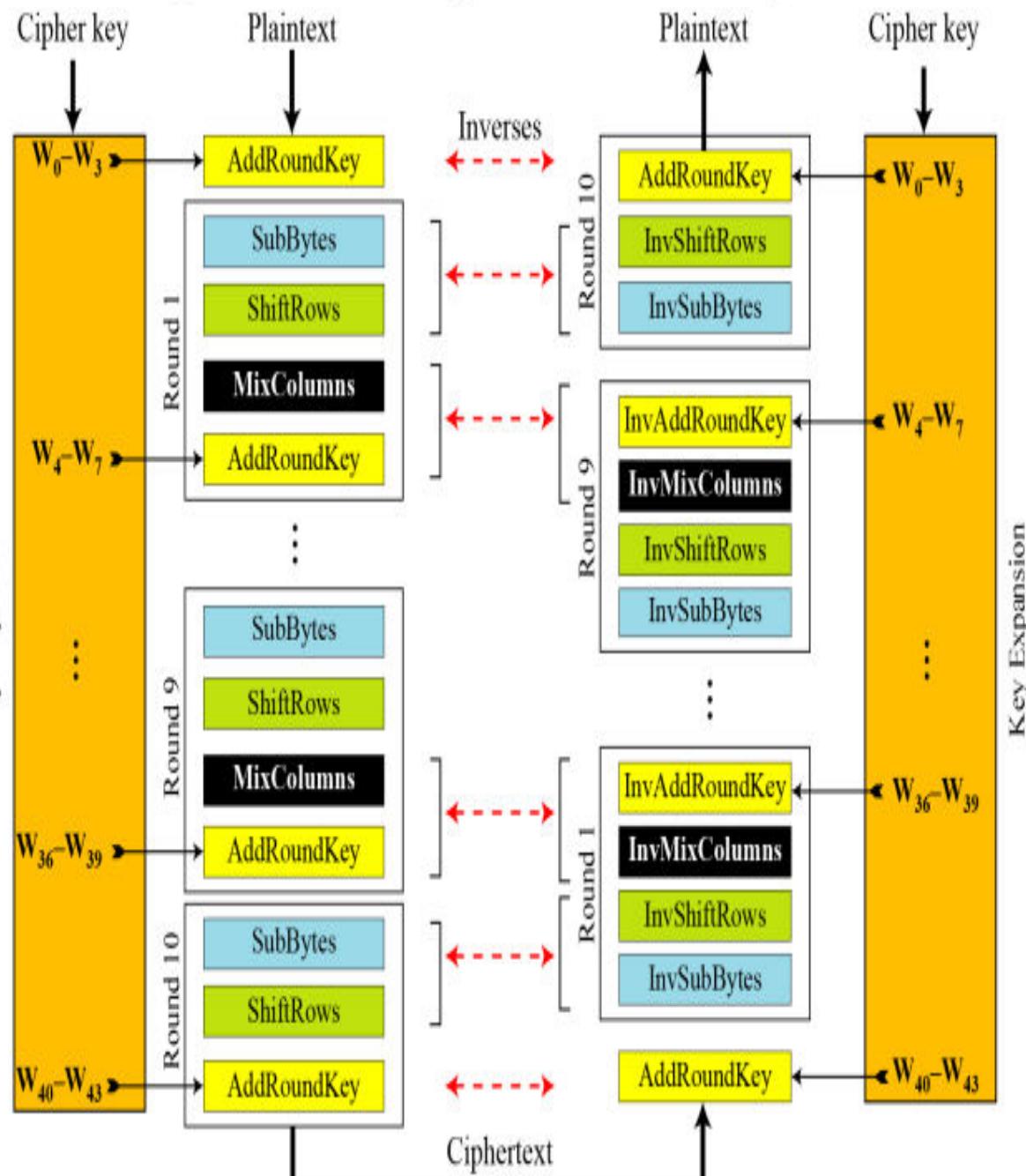
# The AES Ciphers –Alternative Design



## Ciphers and inverse ciphers of the original design



## Cipher and reverse cipher in alternate design



# Analysis of AES

---

This section is a brief review of the three characteristics of AES.

- Security
- Implementation
- Simplicity and Cost

# Analysis of AES

---

## Security

- Brute-Force Attack :AES is definitely more secure than DES due to the larger-size key.
- Statistical Attacks :Numerous tests have failed to do statistical analysis of the ciphertext.
- Differential and Linear Attacks :There are no differential and linear attacks on AES as yet.

# Analysis of AES

---

## Implementation

AES can be implemented in software, hardware, and firmware.

The implementation can use table lookup process or routines that use a well-defined algebraic structure.

# Analysis of AES

---

## Simplicity and Cost

The algorithms used in AES are so simple that they can be easily implemented using cheap processors and a minimum amount of memory.

# THANK YOU