

MATHEMATICS OF CRYPTOGRAPHY

PART I

MODULAR ARITHMETIC AND CONGRUENCE

**Book : Cryptography and Network
security by Behrouz A. Forouzan**

Integer Arithmetic

- In integer arithmetic, we use a set and a few operations.
- Reviewed here to create a background for modular arithmetic.

Set of Integers

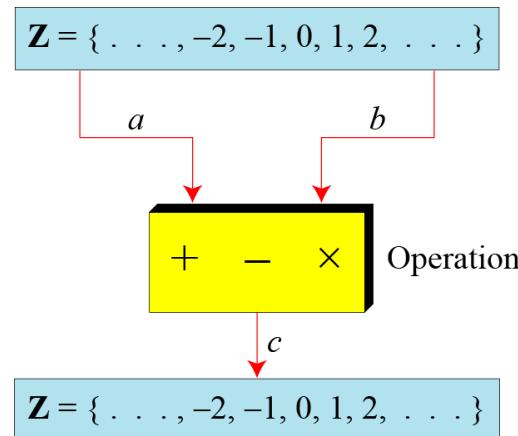
- The set of integers, denoted by \mathbb{Z} , contains all integral numbers (with no fraction) from negative infinity to positive infinity

$$\mathbb{Z} = \{ \dots, -2, -1, 0, 1, 2, \dots \}$$

The set of integers

Binary Operations

- In cryptography, we are interested in three binary operations applied to the set of integers.
- A binary operation takes two inputs and creates one output.



Three binary operations for the set of integers

Integer Division

- In integer arithmetic, if we divide a by n, we can get q and r.
- The relationship between these four integers can be shown as

$$a = q \times n + r$$

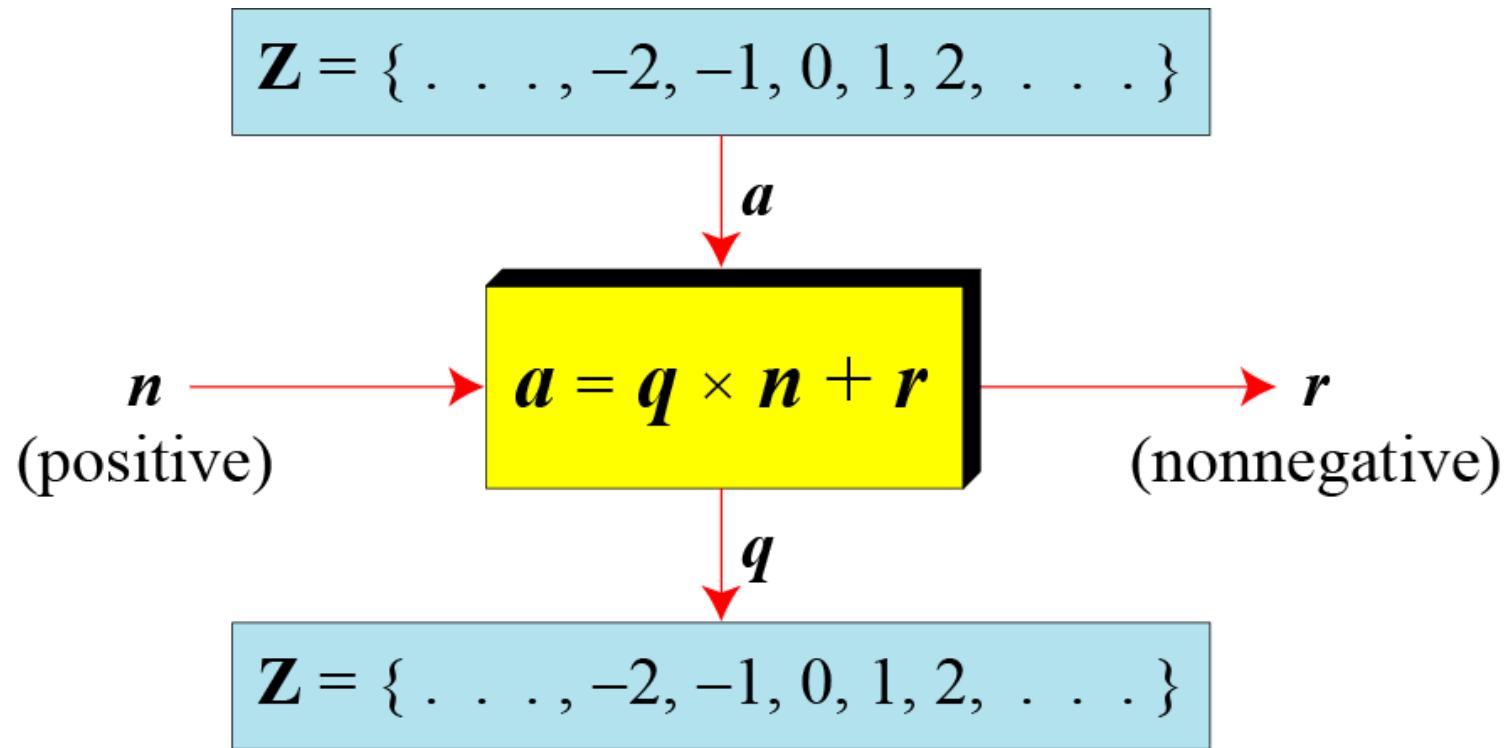
Integer Division(cont.)

- Assume that $a = 255$ and $n = 11$. We can find $q = 23$ and $r = 2$ using the division algorithm.

$$\begin{array}{r} 23 \xleftarrow{\text{q}} \\[-1ex] \overline{)255 \xleftarrow{\text{a}}} \\ 22 \\ \hline 35 \\ 33 \\ \hline 2 \xleftarrow{\text{r}} \end{array}$$

Finding the quotient and the remainder

Integer Division(cont.)



Division algorithm for integers

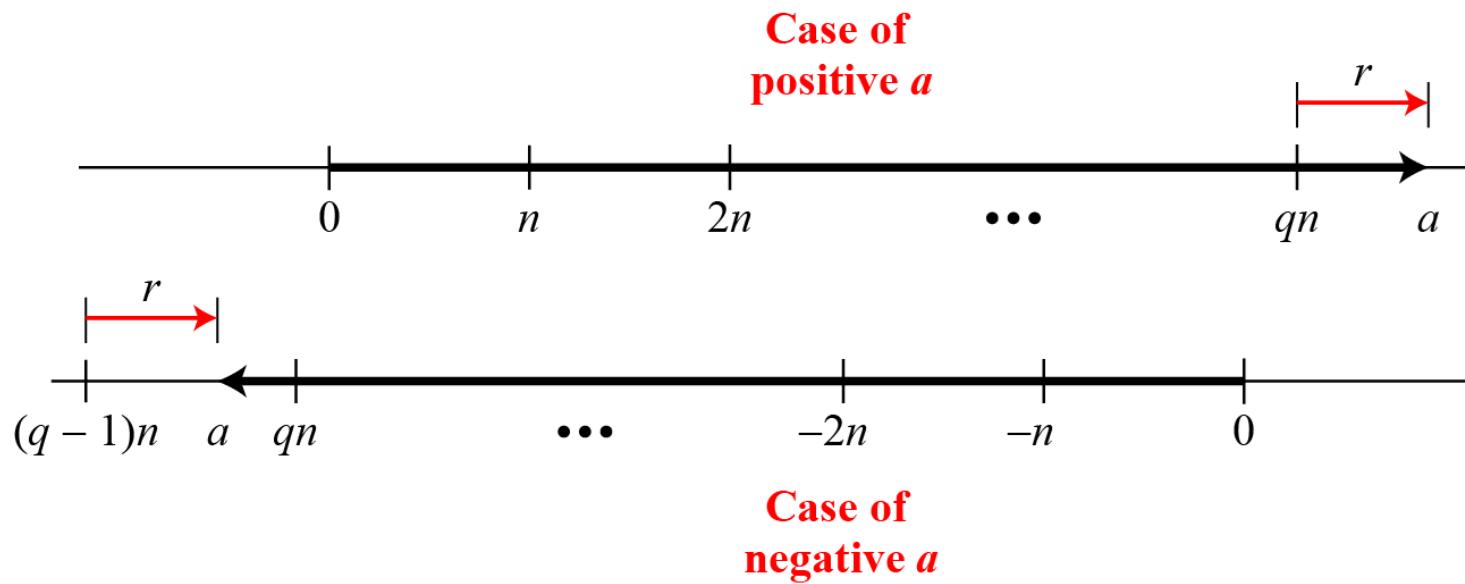
Integer Division(cont.)

- When we use a computer or a calculator, r and q are negative when a is negative.
- How can we apply the restriction that r needs to be positive?
- The solution is simple, we decrement the value of q by 1 and we add the value of n to r to make it positive.

$$-255 = (-23 \times 11) + (-2) \quad \leftrightarrow \quad -255 = (-24 \times 11) + 9$$

Integer Division(cont.)

- Graph of division algorithm



Divisibility

- If a is not zero and we let $r = 0$ in the division relation, we get

$$a = q \times n$$

- If the remainder is zero, $n \mid a$
- If the remainder is not zero, $n \not\mid a$

Divisibility(cont.)

- Properties

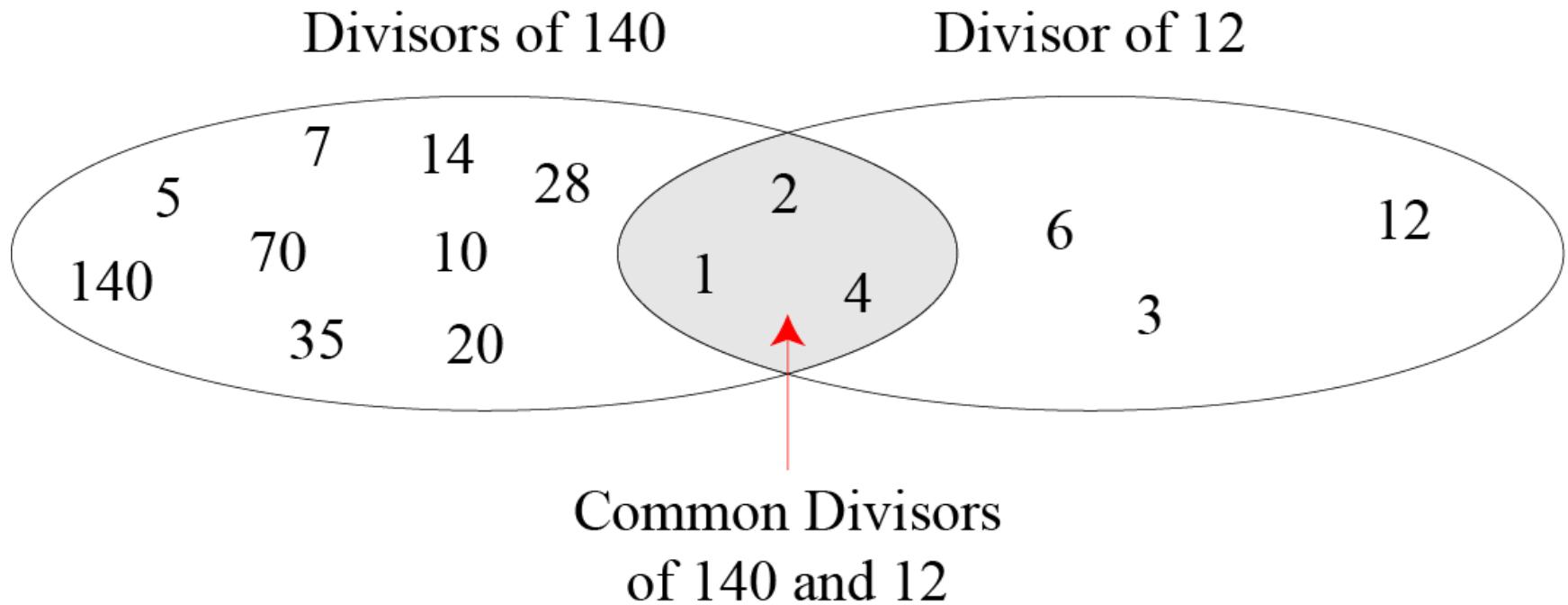
Property 1: if $a|1$, then $a = \pm 1$.

Property 2: if $a|b$ and $b|a$, then $a = \pm b$.

Property 3: if $a|b$ and $b|c$, then $a|c$.

Property 4: if $a|b$ and $a|c$, then
 $a|(m \times b + n \times c)$, where m
and n are arbitrary integers

Divisibility(cont.)



Divisibility(cont.)

Greatest Common Divisor

The greatest common divisor of two positive integers is the largest integer that can divide both integers.

Euclidean Algorithm

Fact 1: $\gcd(a, 0) = a$

Fact 2: $\gcd(a, b) = \gcd(b, r)$, where r is the remainder of dividing a by b

Divisibility(cont.)

- For example, to calculate the $\text{gcd}(36,10)$, we use following steps:

$$\text{gcd}(36, 10) = \text{gcd}(10, 6) \dots \text{by fact 2}$$

$$\text{gcd}(10, 6) = \text{gcd}(6, 4) \dots \text{by fact 2}$$

$$\text{gcd}(6, 4) = \text{gcd}(4, 2) \dots \text{by fact 2}$$

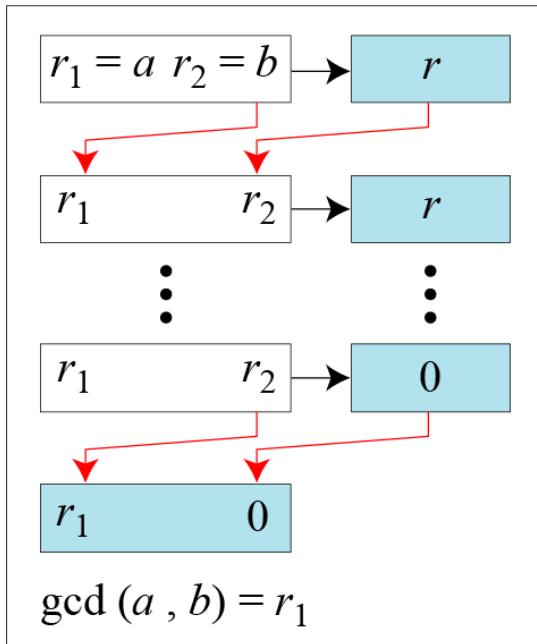
$$\text{gcd}(4, 2) = \text{gcd}(2, 0) \dots \text{by fact 2}$$

$$\text{gcd}(2, 0) = 2 \dots \text{by fact 1}$$

Hence, Answer = 2

Divisibility(cont.)

Euclidean Algorithm



a. Process

```
 $r_1 \leftarrow a; \quad r_2 \leftarrow b;$  (Initialization)  
while ( $r_2 > 0$ )  
{  
     $q \leftarrow r_1 / r_2;$   
     $r \leftarrow r_1 - q \times r_2;$   
     $r_1 \leftarrow r_2; \quad r_2 \leftarrow r;$   
}  
 $\gcd(a, b) \leftarrow r_1$ 
```

b. Algorithm

When $\gcd(a, b) = 1$, we say that a and b are relatively prime.

Divisibility(cont.)

Find the greatest common divisor of 2740 and 1760.

q	r_1	r_2	r
1	2740	1760	980
1	1760	980	780
1	980	780	200
3	780	200	180
1	200	180	20
9	180	20	0
	20	0	

Answer: $\gcd(2740, 1760) = 20$.

Divisibility(cont.)

Find the greatest common divisor of 25 and 60.

Divisibility(cont.)

Extended Euclidean Algorithm

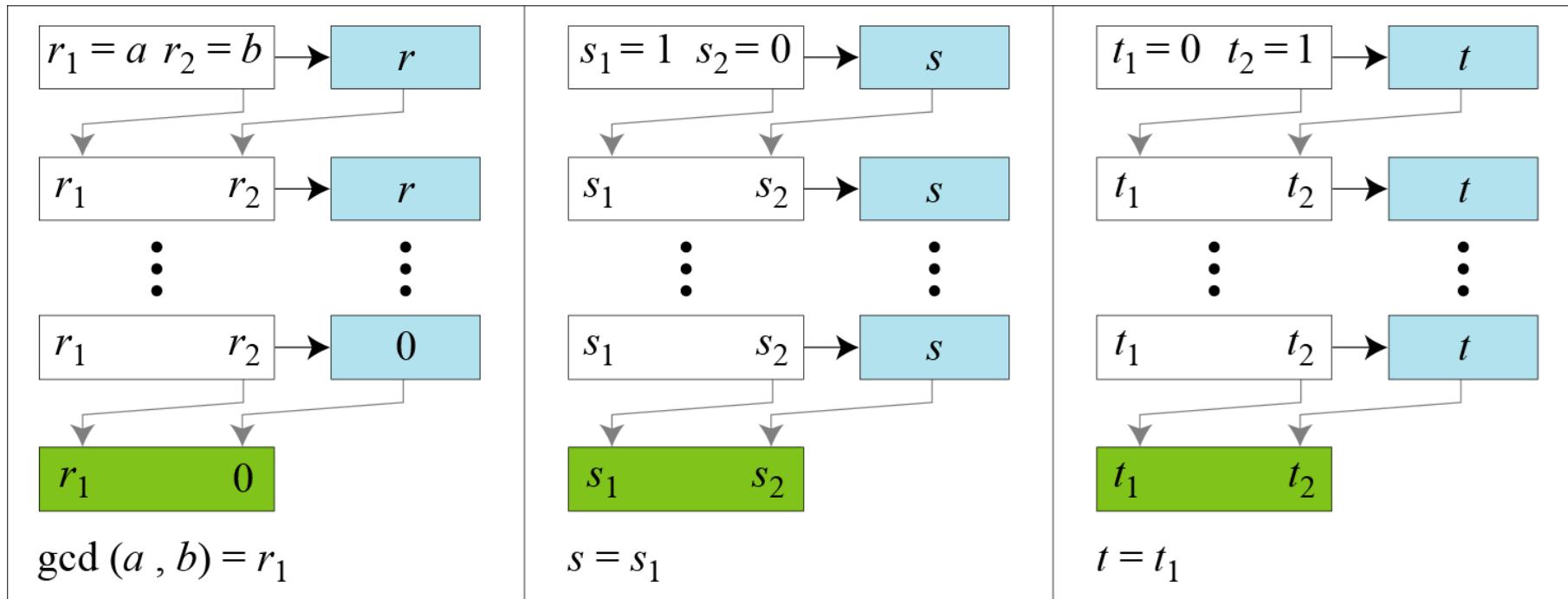
Given two integers a and b , we often need to find other two integers, s and t , such that

$$s \times a + t \times b = \gcd(a, b)$$

The extended Euclidean algorithm can calculate the $\gcd(a, b)$ and at the same time calculate the value of s and t .

Divisibility(cont.)

Extended Euclidean algorithm, part a



a. Process

Divisibility(cont.)

Extended Euclidean algorithm, part b

```
r1 ← a;      r2 ← b;  
s1 ← 1;      s2 ← 0;  
t1 ← 0;      t2 ← 1;
```

(Initialization)

while ($r_2 > 0$)

{

$q \leftarrow r_1 / r_2;$

```
  r ← r1 - q × r2;  
  r1 ← r2; r2 ← r;
```

(Updating r 's)

```
  s ← s1 - q × s2;  
  s1 ← s2; s2 ← s;
```

(Updating s 's)

```
  t ← t1 - q × t2;  
  t1 ← t2; t2 ← t;
```

(Updating t 's)

}

gcd (a , b) ← r₁; s ← s₁; t ← t₁

b. Algorithm

Divisibility(cont.)

Given $a = 161$ and $b = 28$, find $\gcd(a, b)$ and the values of s and t .

Solution

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	-5
1	28	21	7	0	1	-1	1	-5	6
3	21	7	0	1	-1	4	-5	6	-23
	7	0		-1	4		6	-23	

We get $\gcd(161, 28) = 7$, $s = -1$ and $t = 6$.

Divisibility(cont.)

Given $a = 17$ and $b = 0$, find $\gcd(a, b)$ and the values of s and t .

Solution

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
	17	0		1	0		0	1	

We get $\gcd(17, 0) = 17$, $s = 1$, and $t = 0$

Divisibility(cont.)

Given $a = 0$ and $b = 45$, find $\gcd(a, b)$ and the values of s and t .

Solution

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
0	0	45	0	1	0	1	0	1	0
	45	0		0	1		1	0	

We get $\gcd(0, 45) = 45$, $s = 0$, and $t = 1$.

Divisibility(cont.)

Exercise:

Given $a = 84$ and $b = 320$, find $\gcd(a, b)$ and the values of s and t .

Divisibility(cont.)

Exercise:

Given $a = 84$ and $b = 320$, find $\gcd(a, b)$ and the values of s and t .

Solution:

$$\gcd(84, 320) = 4, s = -19, t = 5$$

Modular Arithmetic

Preliminary

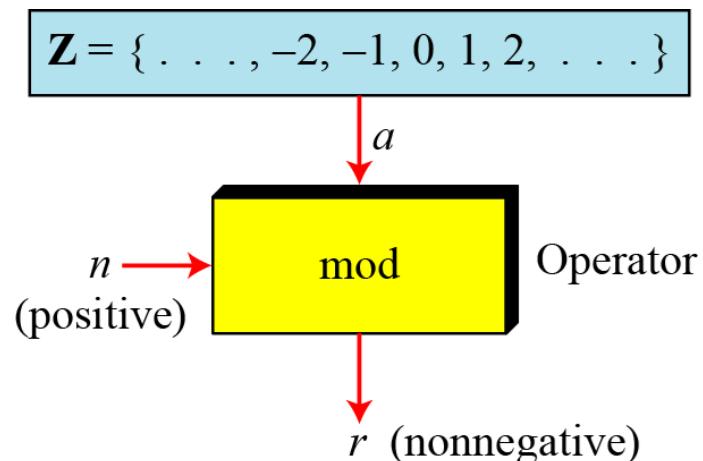
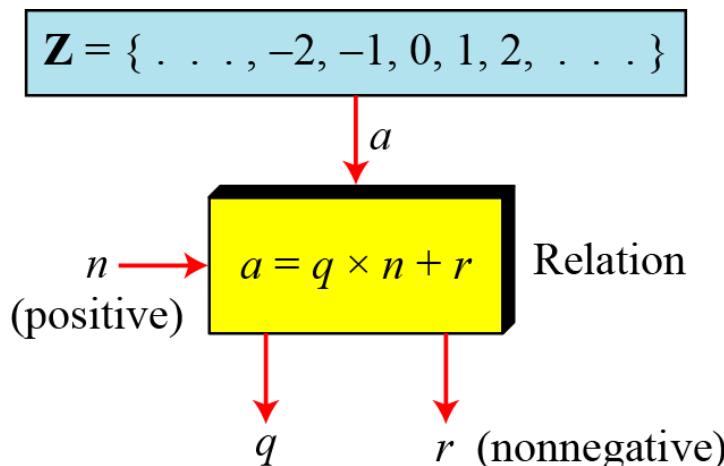
- The division relationship ($a = q \times n + r$) discussed in the previous section has two inputs (a and n) and two outputs (q and r).
- In modular arithmetic, we are interested in only one of the outputs, the remainder r .

Preliminary(cont.)

- We use modular arithmetic in our daily life; for example, we use a clock to measure time. Our clock system uses modulo 12 arithmetic. However, instead of a 0 we use the number 12.

Modulo Operator

- The modulo operator is shown as **mod**. The second input (n) is called the modulus. The output r is called the residue.



Division algorithm and modulo operator

Modulo Operator(cont.)

- Find the result of the following operations:
 - a. $27 \bmod 5$
 - b. $36 \bmod 12$
 - c. $-18 \bmod 14$
 - d. $-7 \bmod 10$

Modulo Operator(cont.)

- Solution
 - a. Dividing 27 by 5 results in $r = 2$
 - b. Dividing 36 by 12 results in $r = 0$
 - c. Dividing -18 by 14 results in $r = -4$. After adding the modulus $r = 10$
 - d. Dividing -7 by 10 results in $r = -7$. After adding the modulus to -7, $r = 3$

Set of Residues

- The modulo operation creates a set, which in modular arithmetic is referred to as **the set of least residues modulo n, or Z_n .**

$$Z_n = \{ 0, 1, 2, 3, \dots, (n - 1) \}$$

$$Z_2 = \{ 0, 1 \}$$

$$Z_6 = \{ 0, 1, 2, 3, 4, 5 \}$$

$$Z_{11} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \}$$

Some Z_n sets

Congruence

- To show that two integers are congruent, we use the congruence operator (\equiv). For example, we write:

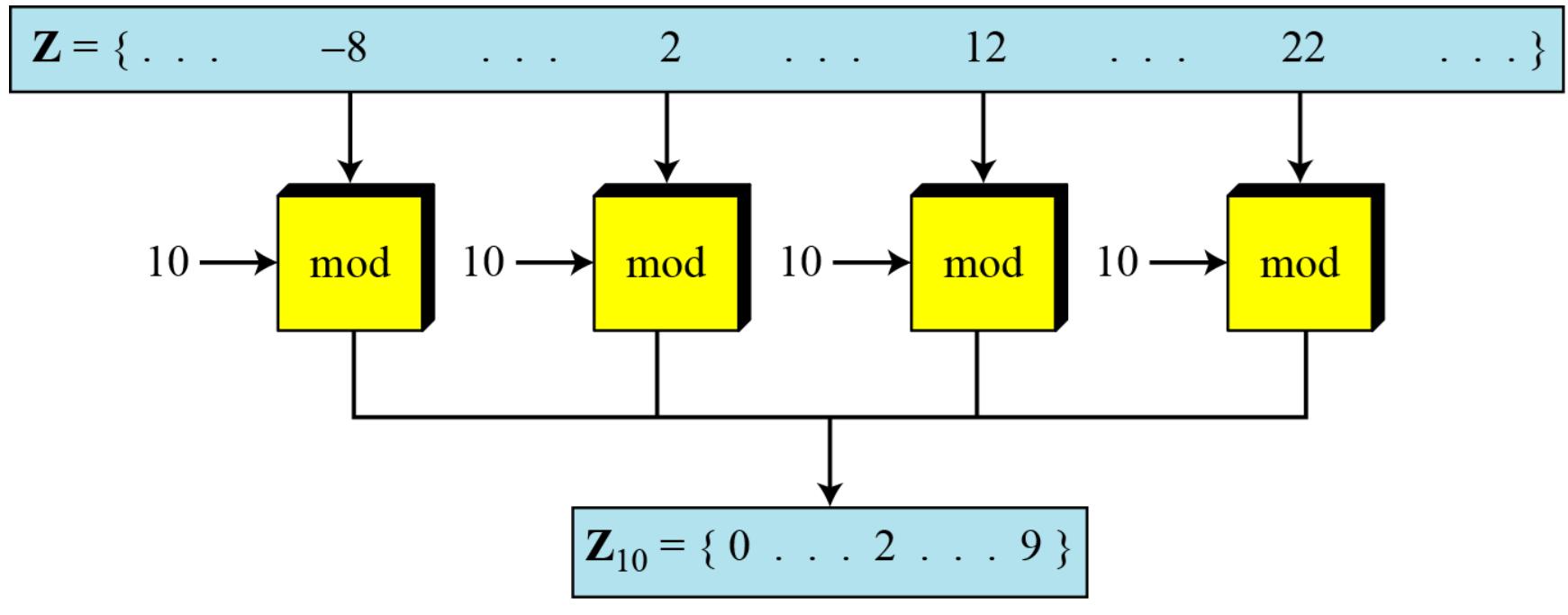
$$2 \equiv 12 \pmod{10}$$

$$3 \equiv 8 \pmod{5}$$

$$13 \equiv 23 \pmod{10}$$

$$8 \equiv 13 \pmod{5}$$

Congruence(cont.)



$$-8 \equiv 2 \equiv 12 \equiv 22 \pmod{10}$$

Congruence Relationship

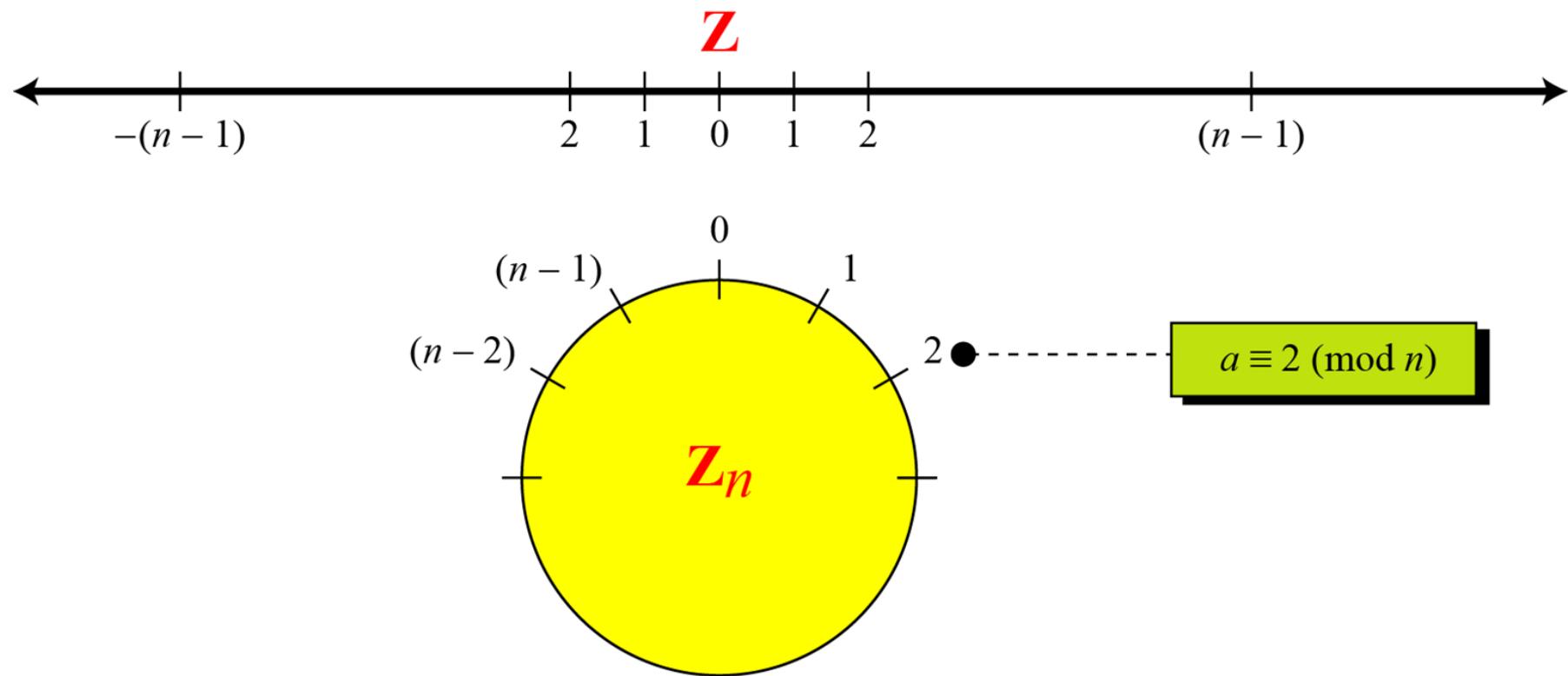
Congruence(cont.)

- Residue Classes
 - A residue class $[a]$ or $[a]_n$ is the set of integers congruent modulo n .
 - It is the set of all integers such that $x=a \pmod{n}$
 - E.g. for $n=5$, we have five sets as shown below:

$$\begin{aligned}[0] &= \{ \dots, -15, -10, -5, 0, 5, 10, 15, \dots \} \\ [1] &= \{ \dots, -14, -9, -4, 1, 6, 11, 16, \dots \} \\ [2] &= \{ \dots, -13, -8, -3, 2, 7, 12, 17, \dots \} \\ [3] &= \{ \dots, -12, -7, -5, 3, 8, 13, 18, \dots \} \\ [4] &= \{ \dots, -11, -6, -1, 4, 9, 14, 19, \dots \}\end{aligned}$$

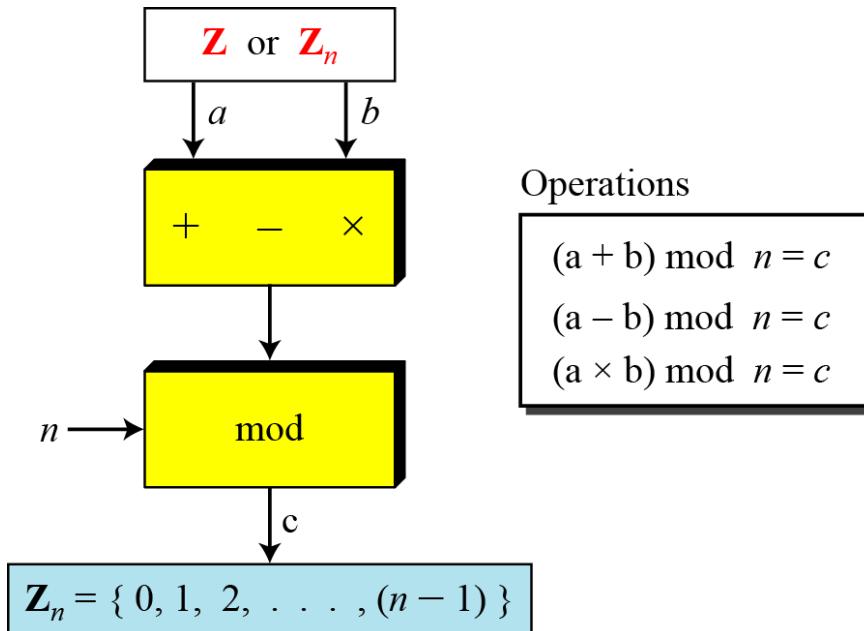
Congruence(cont.)

- Comparison of \mathbb{Z} and \mathbb{Z}_n using graphs



Operation in Z_n

- The three binary operations that we discussed for the set Z can also be defined for the set Z_n . The result may need to be mapped to Z_n using the mod operator.



Operation in Z_n (cont.)

- Perform the following operations (the inputs come from Z_n):
 - a. Add 7 to 14 in Z_{15} .
 - b. Subtract 11 from 7 in Z_{13} .
 - c. Multiply 11 by 7 in Z_{20} .

Operation in Z_n (cont.)

- Solution

$$(14 + 7) \text{ mod } 15 \rightarrow (21) \text{ mod } 15 = 6$$

$$(7 - 11) \text{ mod } 13 \rightarrow (-4) \text{ mod } 13 = 9$$

$$(7 \times 11) \text{ mod } 20 \rightarrow (77) \text{ mod } 20 = 17$$

Operation in Z_n (cont.)

- Perform the following operations (the inputs come from either Z or Z_n):
 - a. Add 17 to 27 in Z_{14} .
 - b. Subtract 43 from 12 in Z_{13} .
 - c. Multiply 123 by -10 in Z_{19} .

Operation in Z_n (cont.)

- Solution
- Add 17 to 27 in Z_{14} .
 - $(17+27)\text{mod } 14 = 2$
- Subtract 43 from 12 in Z_{13} .
 - $(12-43)\text{mod } 13 = 5$
- Multiply 123 by -10 in Z_{19} .
 - $(123 \times (-10)) \text{ mod } 19 = 5$

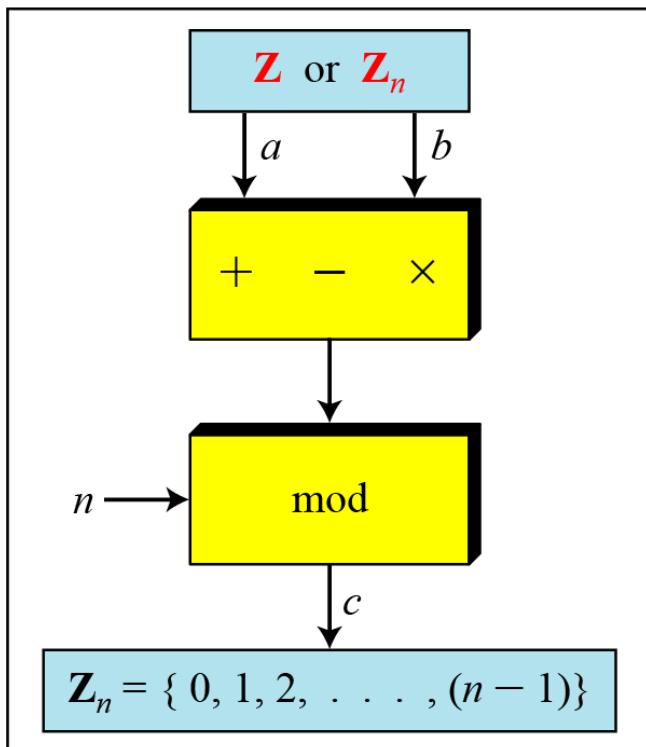
Operation in \mathbb{Z}_n (cont.)

First Property: $(a + b) \bmod n = [(a \bmod n) + (b \bmod n)] \bmod n$

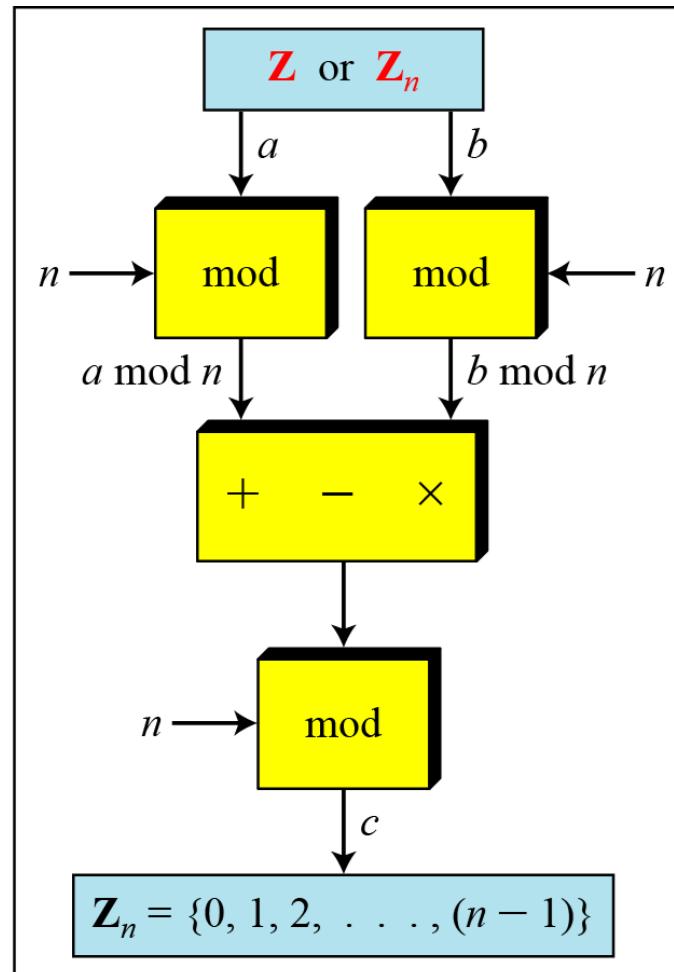
Second Property: $(a - b) \bmod n = [(a \bmod n) - (b \bmod n)] \bmod n$

Third Property: $(a \times b) \bmod n = [(a \bmod n) \times (b \bmod n)] \bmod n$

Operation in \mathbf{Z}_n (cont.)



a. Original process



b. Applying properties

Operation in \mathbb{Z}_n (cont.)

- In arithmetic, we often need to find the remainder of powers of 10 when divided by an integer.

$$10^n \bmod x = (10 \bmod x)^n \quad \text{Applying the third property } n \text{ times.}$$

$$10 \bmod 3 = 1 \rightarrow 10^n \bmod 3 = (10 \bmod 3)^n = 1$$

$$10 \bmod 9 = 1 \rightarrow 10^n \bmod 9 = (10 \bmod 9)^n = 1$$

$$10 \bmod 7 = 3 \rightarrow 10^n \bmod 7 = (10 \bmod 7)^n = 3^n \bmod 7$$

Operation in \mathbb{Z}_n (cont.): Exercise

- We have been told in arithmetic that the remainder of an integer divided by 3 is the same as the remainder of the sum of its decimal digits. In other words, the remainder of dividing 6371 by 3 is same as dividing 17 by 3.

Prove this claim using the properties of the mod operator.

Inverses

- When we are working in modular arithmetic, we often need to find the inverse of a number relative to an operation.
- We are normally looking for an additive inverse (relative to an addition operation) or a multiplicative inverse (relative to a multiplication operation).

Additive Inverses

- In \mathbb{Z}_n , two numbers a and b are additive inverses of each other if

$$a + b \equiv 0 \pmod{n}$$

In modular arithmetic, each integer has an additive inverse. The sum of an integer and its additive inverse is congruent to 0 modulo n.

Additive Inverses

- Find all additive inverse pairs in \mathbb{Z}_{10} .
- Solution
 - The six pairs of additive inverses are $(0, 0)$, $(1, 9)$, $(2, 8)$, $(3, 7)$, $(4, 6)$, and $(5, 5)$.

Multiplicative Inverses

- In \mathbb{Z}_n , two numbers a and b are the multiplicative inverse of each other if,

$$a \times b \equiv 1 \pmod{n}$$

In modular arithmetic, an integer may or may not have a multiplicative inverse.

When it does, the product of the integer and its multiplicative inverse is congruent to 1 modulo n.

Multiplicative Inverses(cont.)

- Find the multiplicative inverse of 8 in \mathbb{Z}_{10} .
 - There is no multiplicative inverse because $\gcd(10, 8) = 2 \neq 1$. In other words, we cannot find any number between 0 and 9 such that when multiplied by 8, the result is congruent to 1.
- Find all multiplicative inverses in \mathbb{Z}_{10} .
 - There are only three pairs: (1, 1), (3, 7) and (9, 9). The numbers 0, 2, 4, 5, 6, and 8 do not have a multiplicative inverse.

Multiplicative Inverses(cont.)

- Find all multiplicative inverse pairs in \mathbb{Z}_{11} .

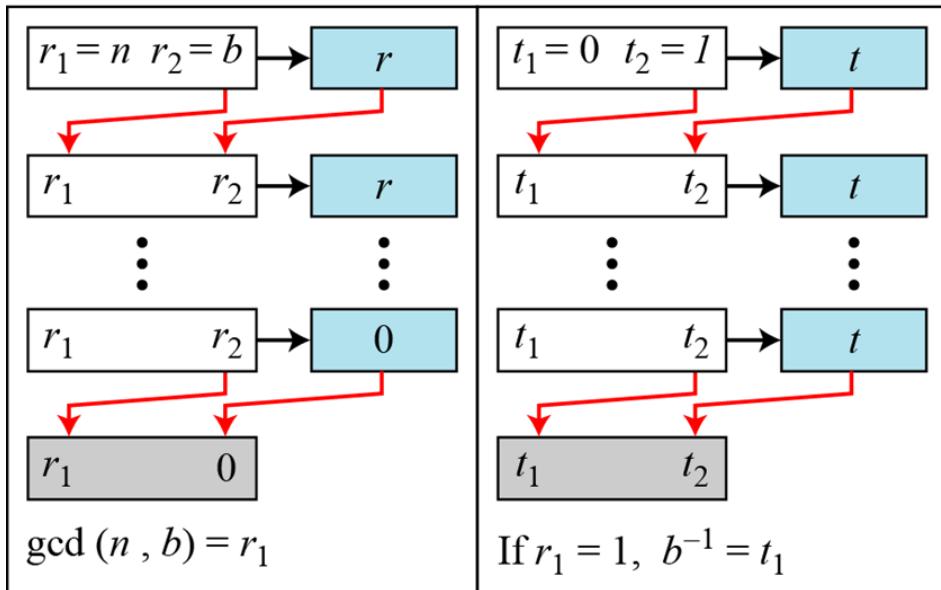
Multiplicative Inverses(cont.)

- Find all multiplicative inverse pairs in \mathbb{Z}_{11} .
 - Solution
 - We have seven pairs: (1, 1), (2, 6), (3, 4), (5, 9), (7, 8), (9, 9), and (10, 10).

Multiplicative Inverses(cont.)

- The extended Euclidean algorithm finds the multiplicative inverses of b in Z_n when n and b are given and $\gcd(n, b) = 1$.
- The multiplicative inverse of b is the value of t after being mapped to Z_n .

Multiplicative Inverses(cont.)



a. Process

```

 $r_1 \leftarrow n; \quad r_2 \leftarrow b;$ 
 $t_1 \leftarrow 0; \quad t_2 \leftarrow 1;$ 

```

while ($r_2 > 0$)

{
 $q \leftarrow r_1 / r_2;$

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

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

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

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

}

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

b. Algorithm

Using extended Euclidean algorithm to find multiplicative inverse

Multiplicative Inverses(cont.)

- Find the multiplicative inverse of 11 in \mathbb{Z}_{26} .

Solution

q	r_1	r_2	r	t_1	t_2	t
2	26	11	4	0	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, 11) is 1; the inverse of 11 is -7 or 19.

Multiplicative Inverses(cont.)

- Find the inverse of 12 in \mathbb{Z}_{26} .

Solution

q	r_1	r_2	r	t_1	t_2	t
2	26	12	2	0	1	-2
6	12	2	0	1	-2	13
	2	0		-2	13	

The gcd (26, 12) is 2; the inverse does not exist.

Multiplicative Inverses(cont.)

- Find the multiplicative inverse of 23 in \mathbb{Z}_{100} .

Multiplicative Inverses(cont.)

- Find the multiplicative inverse of 23 in \mathbb{Z}_{100} .

Solution

q	r_1	r_2	r	t_1	t_2	t
4	100	23	8	0	1	-4
2	23	8	7	1	-4	19
1	8	7	1	-4	9	-13
7	7	1	0	9	-13	100
	1	0		-13	100	

The gcd (100, 23) is 1; the inverse of 23 is -13 or 87.

Addition and Multiplication Tables

- Addition and multiplication table for \mathbb{Z}_{10}

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	0
2	2	3	4	5	6	7	8	9	0	1
3	3	4	5	6	7	8	9	0	1	2
4	4	5	6	7	8	9	0	1	2	3
5	5	6	7	8	9	0	1	2	3	4
6	6	7	8	9	0	1	2	3	4	5
7	7	8	9	0	1	2	3	4	5	6
8	8	9	0	1	2	3	4	5	6	7
9	9	0	1	2	3	4	5	6	7	8

Addition Table in \mathbb{Z}_{10}

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	0	2	4	6	8
3	0	3	6	9	2	5	8	1	4	7
4	0	4	8	2	6	0	4	8	2	6
5	0	5	0	5	0	5	0	5	0	5
6	0	6	2	8	4	0	6	2	8	4
7	0	7	4	1	8	0	2	9	6	3
8	0	8	6	4	2	0	8	6	4	2
9	0	9	8	7	6	5	4	3	2	1

Multiplication Table in \mathbb{Z}_{10}

Different Sets

- Some Z_n and Z_{n^*} sets

$$Z_6 = \{0, 1, 2, 3, 4, 5\}$$

$$Z_6^* = \{1, 5\}$$

$$Z_7 = \{0, 1, 2, 3, 4, 5, 6\}$$

$$Z_7^* = \{1, 2, 3, 4, 5, 6\}$$

$$Z_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$Z_{10}^* = \{1, 3, 7, 9\}$$

We need to use Z_n when additive inverses are needed; we need to use Z_{n^*} when multiplicative inverses are needed.

Two More Sets

- Cryptography often uses two more sets: Z_p and Z_p^* .
- The modulus in these two sets is a prime number.

$$Z_{13} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$$

$$Z_{13}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$$

MATHEMATICS OF CRYPTOGRAPHY

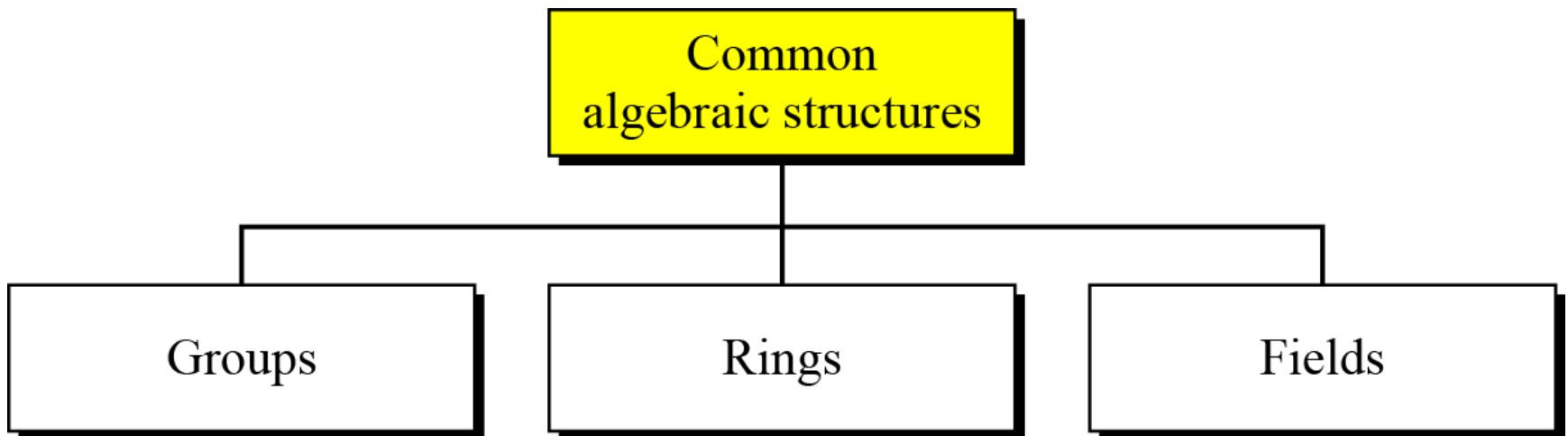
PART II

ALGEBRAIC STRUCTURES

ALGEBRAIC STRUCTURES

- Cryptography requires sets of integers and specific operations that are defined for those sets.
- The combination of the set and the operations that are applied to the elements of the set is called an algebraic structure.
- Three common algebraic structures: groups, rings, and fields.

ALGEBRAIC STRUCTURES(cont.)



Common algebraic structure

Groups

- A group (G) is a set of elements with a binary operation (\bullet) that satisfies four properties (or axioms).
 - Closure
 - Associativity
 - Existence of identity
 - Existence of inverse

Groups(cont.)

- Closure
 - If a and b are elements of G , then $c = a \bullet b$ is also an element of G .
- Associativity
 - If a , b and c are elements of G , then $(a \bullet b) \bullet c = a \bullet (b \bullet c)$
- Existence of identity
 - For all a in G , there exist an element e , called the identity element, such that $e \bullet a = a \bullet e = a$
- Existence of inverse
 - For each a in G , there exists an element a' , called the inverse of a , such that $a \bullet a' = a' \bullet a = e$

Groups(cont.)

- A Commutative group (**Abelian group**) is group in which the operator satisfies four properties plus an extra property that is commutativity.
 - For all a and b in G , we have $a \bullet b = b \bullet a$

Groups(cont.)

- Example

The set of residue integers with the addition operator,

$$G = \langle \mathbb{Z}_n, + \rangle,$$

is a commutative group.

Check the properties....

Groups(cont.)

- Example:
 - The set Z_n^* with the multiplication operator, $G = \langle Z_n^*, \times \rangle$, is also an abelian group.
- Example:
 - Let us define a set $G = \langle \{a, b, c, d\}, \bullet \rangle$ and the operation as shown in Table.

\bullet	a	b	c	d
a	a	b	c	d
b	b	c	d	a
c	c	d	a	b
d	d	a	b	c

Groups(cont.)

- Example:
 - A very interesting group is the permutation group.
 - The set is the set of all permutations, and the operation is composition: applying one permutation after another.
 - Check for properties....
 - Is the group abelian????

Groups(cont.)

- Example(cont.):

\circ	[1 2 3]	[1 3 2]	[2 1 3]	[2 3 1]	[3 1 2]	[3 2 1]
[1 2 3]	[1 2 3]	[1 3 2]	[2 1 3]	[2 3 1]	[3 1 2]	[3 2 1]
[1 3 2]	[1 3 2]	[1 2 3]	[2 3 1]	[2 1 3]	[3 2 1]	[3 1 2]
[2 1 3]	[2 1 3]	[3 1 2]	[1 2 3]	[3 2 1]	[1 3 2]	[2 3 1]
[2 3 1]	[2 3 1]	[3 2 1]	[1 3 2]	[3 1 2]	[1 2 3]	[2 1 3]
[3 1 2]	[3 1 2]	[2 1 3]	[3 2 1]	[1 2 3]	[2 3 1]	[1 3 2]
[3 2 1]	[3 2 1]	[2 3 1]	[3 1 2]	[1 3 2]	[2 1 3]	[1 2 3]

Operation table for permutation group

Groups(cont.)

- In the previous example, we showed that a set of permutations with the composition operation is a group.
- This implies that using two permutations one after another cannot strengthen the security of a cipher.
- Because we can always find a permutation that can do the same job because of the closure property.

Groups(cont.)

- Application
 - Although a group involves a single operation, the properties imposed on the operation allow the use of a pair of operations!!!!
 - How???

Groups(cont.)

- Finite Group
 - If the set has a finite number of elements; otherwise, it is an infinite group.
- Order of a Group $|G|$
 - The number of elements in the group.
 - If the group is finite, its order is finite
- Subgroups
 - A subset H of a group G is a subgroup of G if H itself is a group with respect to the operation on G

Groups(cont.)

- Subgroups(cont.)
 - If $G = \langle S, \bullet \rangle$ is a group, $H = \langle T, \bullet \rangle$ is a group under the same operation, and T is a nonempty subset of S , then H is a subgroup of G
 - If a and b are members of both groups, then $c = a \bullet b$ is also member of both groups
 - The group share the same identity element
 - If a is a member of both groups, the inverse of a is also a member of both groups
 - The group made of the identity element of G , $H = \langle \{e\}, \bullet \rangle$, is a subgroup of G
 - Each group is a subgroup of itself

Groups(cont.)

- Exercise:
 - Is the group $H = \langle \mathbb{Z}_{10}, + \rangle$ a subgroup of the group $G = \langle \mathbb{Z}_{12}, + \rangle$?

Groups(cont.)

- Exercise:
 - Is the group $H = \langle \mathbb{Z}_{10}, + \rangle$ a subgroup of the group $G = \langle \mathbb{Z}_{12}, + \rangle$?
- Solution:
 - The answer is no. Although H is a subset of G , the operations defined for these two groups are different. The operation in H is addition modulo 10; the operation in G is addition modulo 12.

Groups(cont.)

- Cyclic subgroups
 - If a subgroup of a group can be generated using the power of an element, the subgroup is called the **cyclic subgroup**.

$$a^n \rightarrow a \bullet a \bullet \dots \bullet a \quad (n \text{ times})$$

Groups(cont.)

- Four cyclic subgroups can be made from the group $G = \langle \mathbb{Z}_6, + \rangle$.
- They are $H_1 = \langle \{0\}, + \rangle$, $H_2 = \langle \{0, 2, 4\}, + \rangle$, $H_3 = \langle \{0, 3\}, + \rangle$, and $H_4 = G$.

$$0^0 \bmod 6 = 0$$

$$1^0 \bmod 6 = 0$$

$$1^1 \bmod 6 = 1$$

$$1^2 \bmod 6 = (1 + 1) \bmod 6 = 2$$

$$1^3 \bmod 6 = (1 + 1 + 1) \bmod 6 = 3$$

$$1^4 \bmod 6 = (1 + 1 + 1 + 1) \bmod 6 = 4$$

$$1^5 \bmod 6 = (1 + 1 + 1 + 1 + 1) \bmod 6 = 5$$

$$2^0 \bmod 6 = 0$$

$$2^1 \bmod 6 = 2$$

$$2^2 \bmod 6 = (2 + 2) \bmod 6 = 4$$

$$3^0 \bmod 6 = 0$$

$$3^1 \bmod 6 = 3$$

$$4^0 \bmod 6 = 0$$

$$4^1 \bmod 6 = 4$$

$$4^2 \bmod 6 = (4 + 4) \bmod 6 = 2$$

$$5^0 \bmod 6 = 0$$

$$5^1 \bmod 6 = 5$$

$$5^2 \bmod 6 = 4$$

$$5^3 \bmod 6 = 3$$

$$5^4 \bmod 6 = 2$$

$$5^5 \bmod 6 = 1$$

Groups(cont.)

- Exercise:
 - Find out the cyclic subgroups for group $G = \langle \mathbb{Z}_{10}^*, \times \rangle$.

Groups(cont.)

- Three cyclic subgroups can be made from the group $G = \langle \mathbb{Z}_{10}^*, \times \rangle$. G has only four elements: 1, 3, 7, and 9. The cyclic subgroups are $H_1 = \langle \{1\}, \times \rangle$, $H_2 = \langle \{1, 9\}, \times \rangle$, and $H_3 = G$.

$$1^0 \bmod 10 = 1$$

$$3^0 \bmod 10 = 1$$

$$3^1 \bmod 10 = 3$$

$$3^2 \bmod 10 = 9$$

$$3^3 \bmod 10 = 7$$

$$7^0 \bmod 10 = 1$$

$$7^1 \bmod 10 = 7$$

$$7^2 \bmod 10 = 9$$

$$7^3 \bmod 10 = 3$$

$$9^0 \bmod 10 = 1$$

$$9^1 \bmod 10 = 9$$

Groups(cont.)

- Cyclic group
 - A cyclic group is a group that is its own cyclic subgroup.

$$\{e, g, g^2, \dots, g^{n-1}\}, \text{ where } g^n = e$$

Groups(cont.)

- Cyclic group(cont.)
- Example:
 - Three cyclic subgroups can be made from the group $G = \langle Z_{10}^*, \times \rangle$.
 - The cyclic subgroups are $H_1 = \langle \{1\}, \times \rangle$, $H_2 = \langle \{1, 9\}, \times \rangle$, and $H_3 = G$.
 - The group $G = \langle Z_{10}^*, \times \rangle$ is a cyclic group with two generators, $g = 3$ and $g = 7$.
 - The group $G = \langle Z_6, + \rangle$ is a cyclic group with two generators, $g = 1$ and $g = 5$.

Groups(cont.)

- Lagrange's Theorem
 - Assume that G is a group, and H is a subgroup of G . If the order of G and H are $|G|$ and $|H|$, respectively, then, based on this theorem, $|H|$ divides $|G|$.
- Order of an Element
 - The order of an element is the order of the cyclic group it generates.

Groups(cont.)

- Example:
 - In the group $G = \langle \mathbb{Z}_6, + \rangle$, the orders of the elements are:
 $\text{ord}(0) = 1, \text{ord}(1) = 6, \text{ord}(2) = 3, \text{ord}(3) = 2, \text{ord}(4) = 3,$
 $\text{ord}(5) = 6.$
 - In the group $G = \langle \mathbb{Z}_{10}^*, \times \rangle$, the orders of the elements are:
 $\text{ord}(1) = 1, \text{ord}(3) = 4, \text{ord}(7) = 4, \text{ord}(9) = 2.$

Ring

- A ring, $R = \langle \dots, \bullet, \blacksquare \rangle$, is an algebraic structure with two operations.
- First operation must satisfy all five properties
- Second operation must satisfy only the first two
- In addition, second operation must be distributed over first
 - i.e. for all a, b , and c elements of R , we have,
$$a \blacksquare (b \bullet c) = (a \blacksquare b) \bullet (a \blacksquare c) \text{ and}$$
$$(a \bullet b) \blacksquare c = (a \blacksquare c) \bullet (a \blacksquare c)$$

Ring(cont.)

- Commutative Ring

Distribution of \square over \bullet

- 1. Closure \bullet
- 2. Associativity
- 3. Commutativity
- 4. Existence of identity
- 5. Existence of inverse

- 1. Closure \square
- 2. Associativity
- 3. Commutativity

Note:
The third property is
only satisfied for a
commutative ring.

$\{a, b, c, \dots\}$

Set



Operations

Ring

Ring(cont.)

- The set \mathbb{Z} with two operations, addition and multiplication, is a commutative ring.
- We show it by $R = \langle \mathbb{Z}, +, \times \rangle$.
- Addition satisfies all of the five properties; multiplication satisfies only three properties.

Field

- A field, denoted by $F = \langle \{ \dots \}, \bullet, \square \rangle$ is a commutative ring in which the second operation satisfies all five properties defined for the first operation except that the identity of the first operation has no inverse.

Distribution of \square over \bullet

- 1. Closure \bullet
- 2. Associativity
- 3. Commutativity
- 4. Existence of identity
- 5. Existence of inverse

- 1. Closure \square
- 2. Associativity
- 3. Commutativity
- 4. Existence of identity
- 5. Existence of inverse

Note:
The identity element of the first operation has no inverse with respect to the second operation.

$\{a, b, c, \dots\}$
Set

\bullet \square
Operations

Field

Field(cont.)

- Finite Fields
 - Galois showed that for a field to be finite, the number of elements should be p^n , where p is a prime and n is a positive integer.

A Galois field, $GF(p^n)$, is a finite field with p^n elements.

Field(cont.)

- $\text{GF}(p)$ Fields
 - When $n = 1$, we have $\text{GF}(p)$ field.
 - This field can be the set \mathbb{Z}_p , $\{0, 1, \dots, p - 1\}$, with two arithmetic operations.

Field(cont.)

- A very common field in this category is GF(2) with the set {0, 1} and two operations, addition and multiplication.

GF(2)

{0, 1}	+ ×
--------	--------

+	0	1
0	0	1
1	1	0

Addition

×	0	1
0	0	0
1	0	1

Multiplication

a	0	1	a	0	1
-a	1	0	a ⁻¹	—	1

Inverses

GF(2) field

Field(cont.)

- We can define GF(5) on the set \mathbb{Z}_5 (5 is a prime) with addition and multiplication operators.

Field(cont.)

- We can define GF(5) on the set \mathbb{Z}_5 (5 is a prime) with addition and multiplication operators.

GF(5)

$\{0, 1, 2, 3, 4\}$ + ×

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Addition

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Multiplication

Additive inverse

a	0	1	2	3	4
-a	0	4	3	2	1
a	0	1	2	3	4
a^{-1}	-1	3	2	4	

Multiplicative inverse

GF(5) field

- Summary:

<i>Algebraic Structure</i>	<i>Supported Typical Operations</i>	<i>Supported Typical Sets of Integers</i>
Group	(+ −) or (× ÷)	\mathbf{Z}_n or \mathbf{Z}_n^*
Ring	(+ −) and (×)	\mathbf{Z}
Field	(+ −) and (× ÷)	\mathbf{Z}_p

GF(2^n) FIELDS

- In cryptography, we often need to use four operations(addition, subtraction, multiplication and division).
- In other words, we need to use fields.
- However, when we work with computers, the positive integers are stored in the computers as n-bit words in which n is usually 8,16,32 and so on.
- Range of integers is 0 to $2^n - 1$
- Hence modulus is ?????
- What if we want to use field????

GF(2^n) FIELDS (cont.)

- Solution 1
 - Use GF(p), with the set Z_p , where p is the largest prime number less than 2^n
 - But the problem ???
- Solution 2
 - Use GF(2^n)
 - Use a set of 2^n words
 - The elements in this set are n-bit words
 - E.g. for $n=3$, the set is
 $\{000,001,010,011,100,101,110,111\}$

GF(2^n) FIELDS (cont.)

- Solution 2
 - But the problem???

GF(2^n) FIELDS (cont.)

- Solution 2
 - But the problem???
 - 2^n is not prime
 - Need to define operations on the set of elements in GF(2^n)

GF(2^n) FIELDS (cont.)

- Let us define a GF(2^2) field in which the set has four 2-bit words: {00, 01, 10, 11}.
- We can redefine addition and multiplication for this field in such a way that all properties of these operations are satisfied.

		Addition				
		⊕	00	01	10	11
00		⊕	00	01	10	11
01		01	00	11	10	
10		10	11	00	01	
11		11	10	01	00	

Identity: 00

		Multiplication				
		⊗	00	01	10	11
00		⊗	00	00	00	00
01		00	01	10	11	
10		00	10	11	01	
11		00	11	01	10	

Identity: 01

An example of GF(2^2) field

Polynomials

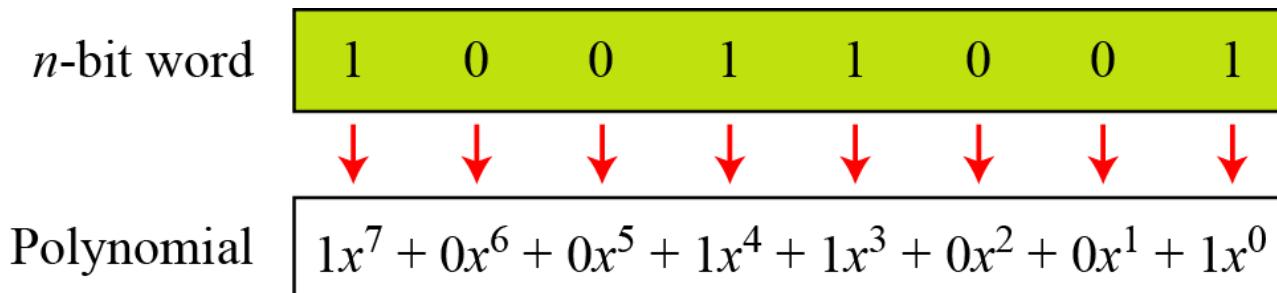
- A polynomial of degree $n - 1$ is an expression of the form

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1x^1 + a_0x^0$$

- where x^i is called the i th term and a_i is called coefficient of the i th term.

Polynomials (cont.)

- We can represent the 8-bit word (10011001) using a polynomial.



First simplification
$$1x^7 + 1x^4 + 1x^3 + 1x^0$$

Second simplification
$$x^7 + x^4 + x^3 + 1$$

Polynomials (cont.)

- Find the 8-bit word related to the polynomial $x^5 + x^2 + x$, we first supply the omitted terms.
- Since $n = 8$, it means the polynomial is of degree 7. The expanded polynomial is,

$$0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x^1 + 0x^0$$

- This is related to the 8-bit word **00100110**.

Polynomials (cont.)

- Operations on polynomials
 - Actually involves two operations
 - Operation on coefficients and operation on polynomials
 - Hence, need to define two fields
 - What for coefficient??
 - What for polynomials???

Polynomials (cont.)

- Operations on polynomials
 - Actually involves two operations
 - Operation on coefficients and operation on polynomials
 - Hence, need to define two fields
 - What for coefficient??
 - What for polynomials???
- GF(2) and GF(2^n) is the answer....

Polynomials (cont.)

- Modulus
 - For the sets of polynomials in $\text{GF}(2^n)$, a group of polynomials of degree n is defined as the modulus.
 - Such polynomials are referred to as **irreducible polynomials**.

Polynomials (cont.)

- **irreducible polynomials.**
 - No polynomial in the set can divide this polynomial
 - Can not be factored into a polynomial with degree of less than n

Degree	Irreducible Polynomials
1	$(x + 1), (x)$
2	$(x^2 + x + 1)$
3	$(x^3 + x^2 + 1), (x^3 + x + 1)$
4	$(x^4 + x^3 + x^2 + x + 1), (x^4 + x^3 + 1), (x^4 + x + 1)$
5	$(x^5 + x^2 + 1), (x^5 + x^3 + x^2 + x + 1), (x^5 + x^4 + x^3 + x + 1),$ $(x^5 + x^4 + x^3 + x^2 + 1), (x^5 + x^4 + x^2 + x + 1)$

Polynomials (cont.)

- Polynomial addition

Addition and subtraction operations on polynomials are the same operation.

Polynomials (cont.)

- Example
- Let us do $(x^5 + x^2 + x) \oplus (x^3 + x^2 + 1)$ in $\text{GF}(2^8)$. We use the symbol \oplus to show that we mean polynomial addition. The following shows the procedure:

$$\begin{array}{r} 0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x^1 + 0x^0 \\ 0x^7 + 0x^6 + 0x^5 + 0x^4 + 1x^3 + 1x^2 + 0x^1 + 1x^0 \\ \hline 0x^7 + 0x^6 + 1x^5 + 0x^4 + 1x^3 + 0x^2 + 1x^1 + 1x^0 \end{array} \rightarrow x^5 + x^3 + x + 1$$

Polynomials (cont.)

- Short cut method
 - Addition in GF(2) means the exclusive-or (XOR) operation.
 - So we can exclusive-or the two words, bits by bits, to get the result.
 - In the previous example, $x^5 + x^2 + x$ is 00100110 and $x^3 + x^2 + 1$ is 00001101.
 - The result is 00101011 or in polynomial notation $x^5 + x^3 + x + 1$.

Polynomials (cont.)

- Multiplication
 - The coefficient multiplication is done in GF(2).
 - The multiplying x^i by x^j results in x^{i+j} .
 - The multiplication may create terms with degree more than $n - 1$, which means the result needs to be reduced using a modulus polynomial.

Polynomials (cont.)

- Example
 - Find the result of $(x^5 + x^2 + x) \otimes (x^7 + x^4 + x^3 + x^2 + x)$ in $\text{GF}(2^8)$ with irreducible polynomial $(x^8 + x^4 + x^3 + x + 1)$.

$$P_1 \otimes P_2 = x^5(x^7 + x^4 + x^3 + x^2 + x) + x^2(x^7 + x^4 + x^3 + x^2 + x) + x(x^7 + x^4 + x^3 + x^2 + x)$$

$$P_1 \otimes P_2 = x^{12} + x^9 + x^8 + x^7 + x^6 + x^9 + x^6 + x^5 + x^4 + x^3 + x^8 + x^5 + x^4 + x^3 + x^2$$

$$P_1 \otimes P_2 = (x^{12} + x^7 + x^2) \bmod (x^8 + x^4 + x^3 + x + 1) = x^5 + x^3 + x^2 + x + 1$$

- To find the final result, divide the polynomial of degree 12 by the polynomial of degree 8 (the modulus) and keep only the remainder.

Polynomials (cont.)

- Polynomial division with coefficients in GF(2)

$$\begin{array}{r} x^4 + 1 \\ \hline x^8 + x^4 + x^3 + x + 1 \quad | \quad x^{12} + x^7 + x^2 \\ \hline x^{12} + x^8 + x^7 + x^5 + x^4 \\ \hline x^8 + x^5 + x^4 + x^2 \\ x^8 + x^4 + x^3 + x + 1 \\ \hline \end{array}$$

Remainder $x^5 + x^3 + x^2 + x + 1$

Polynomials (cont.)

- Example:
 - In GF (2⁴), find the inverse of $(x^2 + 1)$ modulo $(x^4 + x + 1)$.
- Solution
 - The answer is $(x^3 + x + 1)$

q	r_1	r_2	r	t_1	t_2	t
$(x^2 + 1)$	$(x^4 + x + 1)$	$(x^2 + 1)$	(x)	(0)	(1)	$(x^2 + 1)$
(x)	$(x^2 + 1)$	(x)	(1)	(1)	$(x^2 + 1)$	$(x^3 + x + 1)$
(x)	(x)	(1)	(0)	$(x^2 + 1)$	$(x^3 + x + 1)$	(0)
	(1)	(0)		$(x^3 + x + 1)$	(0)	

Polynomials (cont.)

- Example:
 - In $\text{GF}(2^8)$, find the inverse of (x^5) modulo $(x^8 + x^4 + x^3 + x + 1)$.

Polynomials (cont.)

- Example:
 - In $\text{GF}(2^8)$, find the inverse of (x^5) modulo $(x^8 + x^4 + x^3 + x + 1)$.

- Solution

q	r_I	r_2	r	t_I	t_2	t
(x^3)	$(x^8 + x^4 + x^3 + x + 1)$	(x^5)	$(x^4 + x^3 + x + 1)$	(0)	(1)	(x^3)
$(x + 1)$	(x^5)	$(x^4 + x^3 + x + 1)$	$(x^3 + x^2 + 1)$	(1)	(x^3)	$(x^4 + x^3 + 1)$
(x)	$(x^4 + x^3 + x + 1)$	$(x^3 + x^2 + 1)$	(1)	(x^3)	$(x^4 + x^3 + 1)$	$(x^5 + x^4 + x^3 + x)$
$(x^3 + x^2 + 1)$	$(x^3 + x^2 + 1)$	(1)	(0)	$(x^4 + x^3 + 1)$	$(x^5 + x^4 + x^3 + x)$	(0)
	(1)	(0)		$(x^5 + x^4 + x^3 + x)$	(0)	

Polynomials (cont.)

- A better algorithm for polynomial multiplication:
 - Obtain the result by repeatedly multiplying a reduced polynomial by x .
- Example:
 - Find the result of multiplying $P_1 = (x^5 + x^2 + x)$ by $P_2 = (x^7 + x^4 + x^3 + x^2 + x)$ in $\text{GF}(2^8)$ with irreducible polynomial $(x^8 + x^4 + x^3 + x + 1)$

Polynomials (cont.)

- Solution:
 - We first find the partial result of multiplying x^0, x^1, x^2, x^3, x^4 , and x^5 by P_2 . Note that although only three terms are needed, the product of $x^m \otimes P_2$ for m from 0 to 5 because each calculation depends on the previous result.

Powers	Operation	New Result	Reduction
$x^0 \otimes P_2$		$x^7 + x^4 + x^3 + x^2 + x$	No
$x^1 \otimes P_2$	$x \otimes (x^7 + x^4 + x^3 + x^2 + x)$	$x^5 + x^2 + x + 1$	Yes
$x^2 \otimes P_2$	$x \otimes (x^5 + x^2 + x + 1)$	$x^6 + x^3 + x^2 + x$	No
$x^3 \otimes P_2$	$x \otimes (x^6 + x^3 + x^2 + x)$	$x^7 + x^4 + x^3 + x^2$	No
$x^4 \otimes P_2$	$x \otimes (x^7 + x^4 + x^3 + x^2)$	$x^5 + x + 1$	Yes
$x^5 \otimes P_2$	$x \otimes (x^5 + x + 1)$	$x^6 + x^2 + x$	No
$\mathbf{P_1 \times P_2 = (x^6 + x^2 + x) + (x^6 + x^3 + x^2 + x) + (x^5 + x^2 + x + 1) = x^5 + x^3 + x^2 + x + 1}$			

Polynomials (cont.)

- Exercise:

Find the result of multiplying $P_1 = (x^3 + x^2 + x + 1)$ by $P_2 = (x^2 + 1)$ in GF(2⁴) with irreducible polynomial $(x^4 + x^3 + 1)$

Multiplication using computer

We have $P_1 = 000100110$, $P_2 = 10011110$, modulus = 100011010 (nine bits). We show the exclusive or operation by \oplus .

<i>Powers</i>	<i>Shift-Left Operation</i>	<i>Exclusive-Or</i>
$x^0 \otimes P_2$		10011110
$x^1 \otimes P_2$	00111100	$(00111100) \oplus (00011010) = \underline{\underline{00100111}}$
$x^2 \otimes P_2$	01001110	<u>01001110</u>
$x^3 \otimes P_2$	10011100	10011100
$x^4 \otimes P_2$	00111000	$(00111000) \oplus (00011010) = 00100011$
$x^5 \otimes P_2$	01000110	<u>01000110</u>
$P_1 \otimes P_2 = (00100111) \oplus (01001110) \oplus (01000110) = 00101111$		

Polynomials (cont.)

- Exercise:

Find the result of multiplying (10101) by (10000) in $\text{GF}(2^5)$ using $(x^5 + x^2 + 1)$ as modulus.



Chapter 3

Classical Encryption Techniques

Definitions

Plaintext

- An original message

Ciphertext

- The coded message

Enciphering/encryption

- The process of converting from plaintext to ciphertext

Deciphering/decryption

- Restoring the plaintext from the ciphertext

Cryptography

- The area of study of the many schemes used for encryption

Cryptographic system/cipher

- A scheme

Cryptanalysis

- Techniques used for deciphering a message without any knowledge of the enciphering details

Cryptology

- The areas of cryptography and cryptanalysis

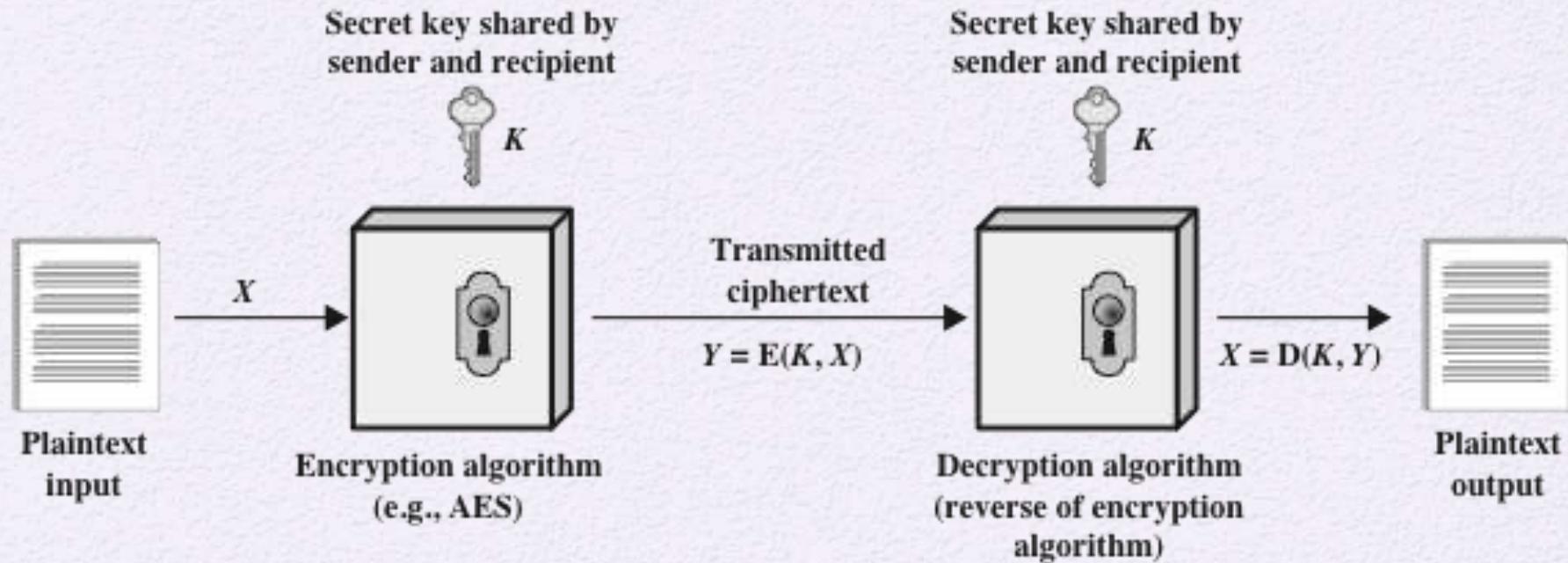


Figure 3.1 Simplified Model of Symmetric Encryption

Symmetric Cipher Model

- There are two requirements for secure use of conventional encryption:
 - A strong encryption algorithm
 - Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure



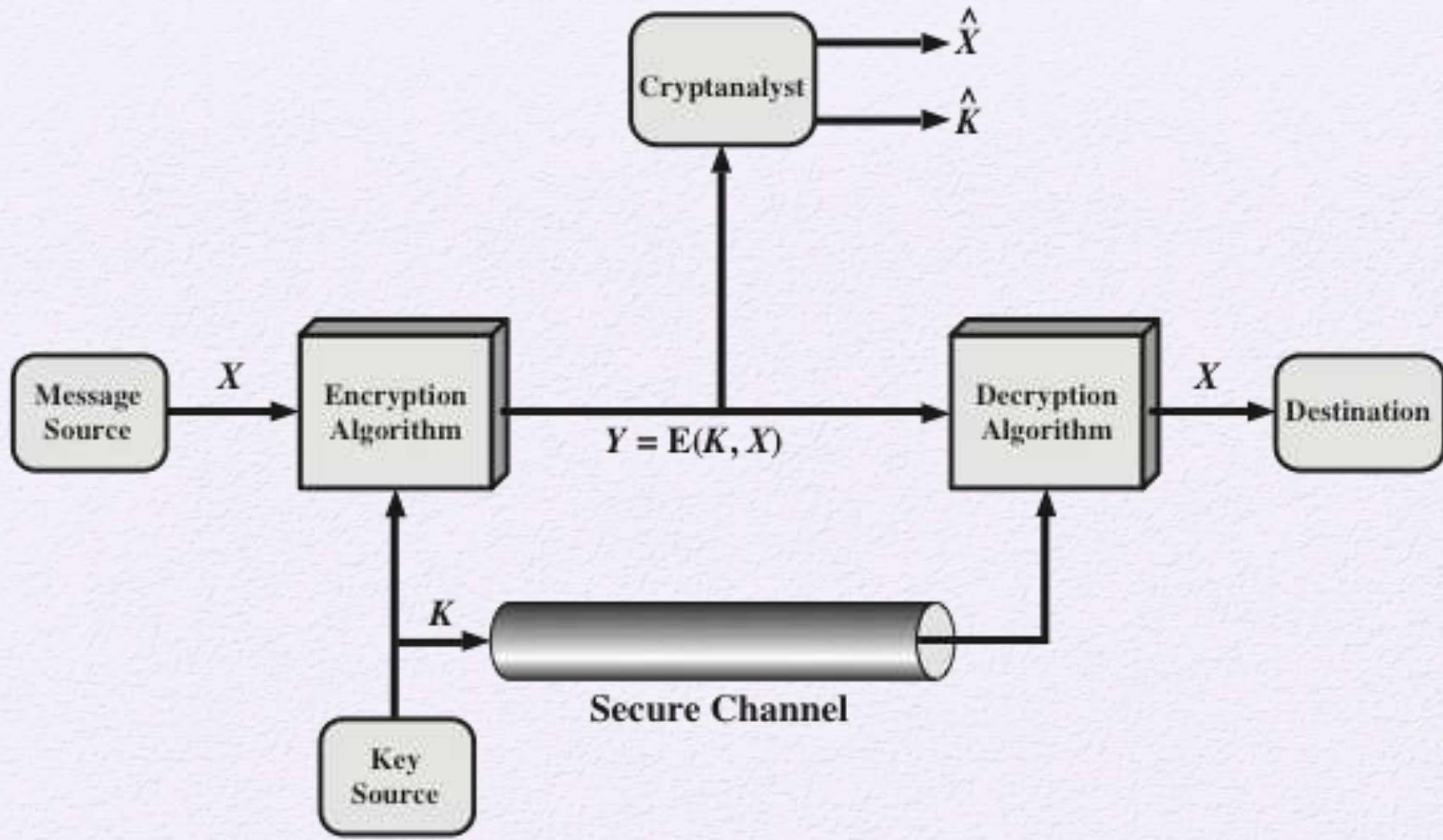


Figure 3.2 Model of Symmetric Cryptosystem

Cryptographic Systems

- Characterized along three independent dimensions:

The type of operations used for transforming plaintext to ciphertext

Substitution

Transposition

The number of keys used

Symmetric,
single-key, secret-key,
conventional
encryption

Asymmetric, two-key,
or public-key
encryption

The way in which the plaintext is processed

Block cipher

Stream cipher

Cryptanalysis and Brute-Force Attack

Cryptanalysis

- Attack relies on the nature of the algorithm plus some knowledge of the general characteristics of the plaintext
- Attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used

Brute-force attack

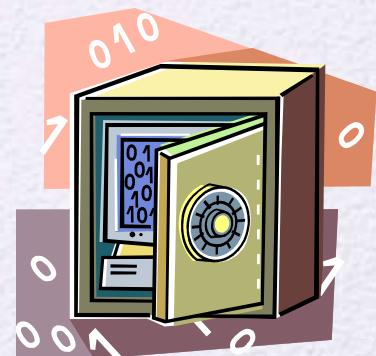
- Attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained
- On average, half of all possible keys must be tried to achieve success

Table 3.1
Types of Attacks on Encrypted Messages

Type of Attack	Known to Cryptanalyst
Ciphertext Only	<ul style="list-style-type: none"> • Encryption algorithm • Ciphertext
Known Plaintext	<ul style="list-style-type: none"> • Encryption algorithm • Ciphertext • One or more plaintext-ciphertext pairs formed with the secret key
Chosen Plaintext	<ul style="list-style-type: none"> • Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	<ul style="list-style-type: none"> • Encryption algorithm • Ciphertext • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	<ul style="list-style-type: none"> • Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

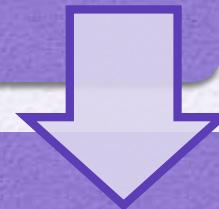
Encryption Scheme Security

- Unconditionally secure
 - No matter how much time an opponent has, it is impossible for him or her to decrypt the ciphertext simply because the required information is not there
- Computationally secure
 - The cost of breaking the cipher exceeds the value of the encrypted information
 - The time required to break the cipher exceeds the useful lifetime of the information

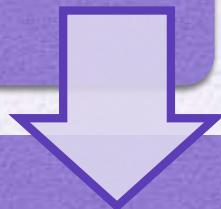


Brute-Force Attack

Involves trying every possible key until an intelligible translation of the ciphertext into plaintext is obtained



On average, half of all possible keys must be tried to achieve success



To supplement the brute-force approach, some degree of knowledge about the expected plaintext is needed, and some means of automatically distinguishing plaintext from garble is also needed

Substitution Technique

- Is one in which the letters of plaintext are replaced by other letters or by numbers or symbols
- If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns





Caesar Cipher

- Simplest and earliest known use of a substitution cipher
- Used by Julius Caesar
- Involves replacing each letter of the alphabet with the letter standing three places further down the alphabet
- Alphabet is wrapped around so that the letter following Z is A

plain: meet me after the toga party

cipher: PHHW PH DIWHU WKH WRJD SDUWB

Caesar Cipher Algorithm

- Can define transformation as:

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
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

- Mathematically give each letter a number

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

- Algorithm can be expressed as:

$$c = E(\beta, p) = (p + \beta) \bmod 26$$

- A shift may be of any amount, so that the general Caesar algorithm is:

$$C = E(k, p) = (p + k) \bmod 26$$

- Where k takes on a value in the range 1 to 25; the decryption algorithm is simply:

$$p = D(k, C) = (C - k) \bmod 26$$

Figure 3.3

Brute-Force Cryptanalysis of Caesar Cipher

(This chart can be found on page 93 in the textbook)

KEY	PHHW PH DIWHU WKH WRJD SDUWB
1	oggv og chvgt vjg vqic rctva
2	nffu nf bgufs uif uphb qbsuz
3	meet me after the toga party
4	ldds ld zesdq sgd snfz ozqsx
5	kccr kc ydrcp rfc rmey nyprw
6	jbbq jb xcqbo qeb qldx mxoqv
7	iaap ia wbpan pda pkcw lwnpu
8	hzzo hz vaozm ocz ojbv kvmot
9	gyyn gy uznyl nby niau julns
10	fxxm fx tymxk max mhzt itkmr
11	ewwl ew sxlwj lzw lgys hsjlq
12	dvvk dv rwkvi kyv kfxr grikp
13	cuuj cu qvjuh jxu jewq fqhjo
14	btti bt puitg iwt idvp epgin
15	assh as othsf hvs hcuo dofhm
16	zrrg zr nsgre gur gbtn cnegl
17	yqqf yq mrfqd ftq fasm bmdfk
18	xppe xp lqepc esp ezrl alcej
19	wood wo kpdob dro dyqk zkbdi
20	vnnnc vn jocna cqn cxpj yjach
21	ummb um inbmz bpm bwoi xizbg
22	tlla tl hmaly aol avnh whyaf
23	skkz sk glzkx znk zumg vgxze
24	rjjy rj fkyjw ymj ytlf ufwyd
25	qiix qi ejxiv xli xske tevxc

Sample of Compressed Text

—+Wμ"— Ω-0)≤4{==‡, ə-Ω%rāu.-† Ø-Z-
Ø≠2Ø#λæð øeq7,Ωn-@3NØÚ Øz'Y-føf [±0_ èΩ,<NO-±«"xÙ λæÈèØ3A
x}øÙk:Å
—yí "ΔÉ] . □ J/*iTø&1 'c<uΩ-
AD(G WÄC~y_IØÄW PØ1<Øtç] , □; "t^uñπ"= "L" 9OgflO" &Ø≤ → Øøg":
"Ø!SGqèvo" ú\,S>h<-*6øt%k" | fiØ#="myk" ≥ñP<, fi Áj AØL"Zù-
Ω"Ø"6øy(% ,Ωøó , i ø+Ái "úO2çSy 'O-
2Äññi /Ø" "ΠK**pøñ, úø^"3Σ"ø"ØZì"Y"ØmY> Ø+øØ/ " «KfL*+~ *≤0~
B ZøK"QøSyøf . :øññzss/] >ØQ a

Monoalphabetic Cipher

- With only 25 possible keys, the Caesar cipher is far from secure.
- A dramatic increase in the key space can be achieved by allowing an arbitrary substitution.
- Permutation
 - Of a finite set of elements S is an ordered sequence of all the elements of S , with each element appearing exactly once
 - For example, if $S = \{a, b, c\}$, there are six permutations of S : abc, acb, bac, bca, cab, cba

Monoalphabetic Cipher

- If the “cipher” line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than 4×10^{26} possible keys
 - This is 10 orders of magnitude greater than the key space for DES
 - Approach is referred to as a *monoalphabetic substitution cipher* because a single cipher alphabet is used per message

Monoalphabetic Cipher

- rather than just shifting the alphabet could shuffle (permute) the letters arbitrarily
- each plaintext letter maps to a different random ciphertext letter
- hence key is 26 letters long

Plain: abcdefghijklmnopqrstuvwxyz

Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN

Plaintext: ifwewishtoreplaceletters

Ciphertext: WIRFRWAJUHYFTSDVFSUUUFYA

Monoalphabetic Cipher

- key size is now 25 characters...
- now have a total of $26! = 4 \times 10^{26}$ keys
- with so many keys, might think is secure
- but would be !!!WRONG!!!
- problem is language characteristics

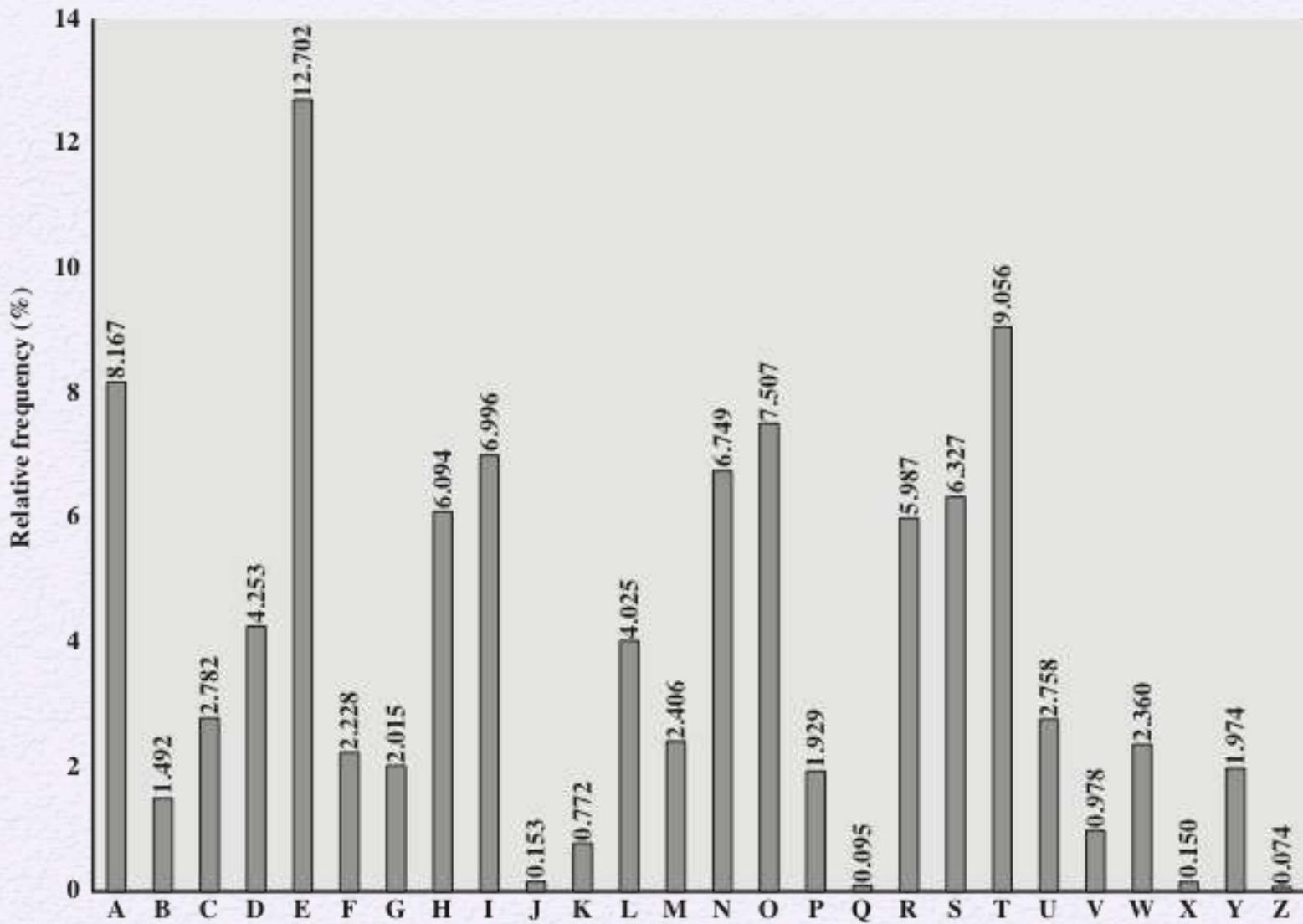


Figure 3.5 Relative Frequency of Letters in English Text

Monoalphabetic Cipher

- given ciphertext:

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPES
XUDBMETSXAIZVUEPHZHMDZSHZOWSFAP
PDTSPQUZWYMXUZUHSXEPLYEPOPDZSZUF
POMBZWPFUPZHMDJUDTMOHMQ

- count relative letter frequencies

Monoalphabetic Cipher

- given ciphertext:

UZQSOVUOHXMOPVGP~~O~~ZPEVSG~~Z~~WS~~Z~~OPF~~P~~PESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSF~~P~~APPDT~~S~~V~~P~~QUZWYMXUZUHSX
EPYEPOPDZSZUF~~P~~OMB~~Z~~WP~~F~~UPZHMDJUDTMOHQ

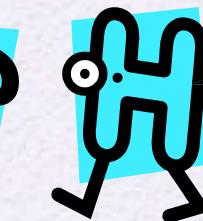
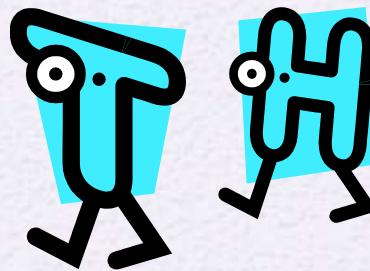
- guess P & Z are e and t
- guess ZW is th and hence ZWP is “the”

- proceeding with trial and error finally get:

it was disclosed yesterday that several
informal but direct contacts have been made
with political representatives of the viet
cong in moscow

Monoalphabetic Ciphers

- Easy to break because they reflect the frequency data of the original alphabet
- Countermeasure is to provide multiple substitutes (homophones) for a single letter
- Digram
 - Two-letter combination
 - Most common is *th*
- Trigram
 - Three-letter combination
 - Most frequent is *the*



Playfair Cipher

- Best-known multiple-letter encryption cipher
- Treats digrams in the plaintext as single units and translates these units into ciphertext digrams
- Based on the use of a 5×5 matrix of letters constructed using a keyword
- Invented by British scientist Sir Charles Wheatstone in 1854
- Used as the standard field system by the British Army in World War I and the U.S. Army and other Allied forces during World War II

Playfair Key Matrix

- Fill in letters of keyword (minus duplicates) from left to right and from top to bottom, then fill in the remainder of the matrix with the remaining letters in alphabetic order
- Using the keyword MONARCHY:

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

Playfair Encryption

- Encryption rules:
- Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x, so that balloon would be treated as ba lx lo on.
- Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, ar is encrypted as RM.

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

Playfair Encryption

- Encryption rules...
- Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last.
- For example, mu is encrypted as CM.

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

Playfair Encryption

- Encryption rules...
- Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter.
- Thus, hs becomes BP and ea becomes IM (or JM, as the encipherer wishes).

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

Playfair Cipher:Security

- Great advance over simple monoalphabetic ciphers.
- Whereas there are only 26 letters, there are $26 * 26 = 676$ digrams, so that identification of individual digrams is more difficult.
- The relative frequencies of individual letters exhibit a much greater range than that of digrams, making frequency analysis much more difficult.
- It is relatively easy to break, because it still leaves much of the structure of the plaintext language intact. A few hundred letters of ciphertext are generally sufficient.

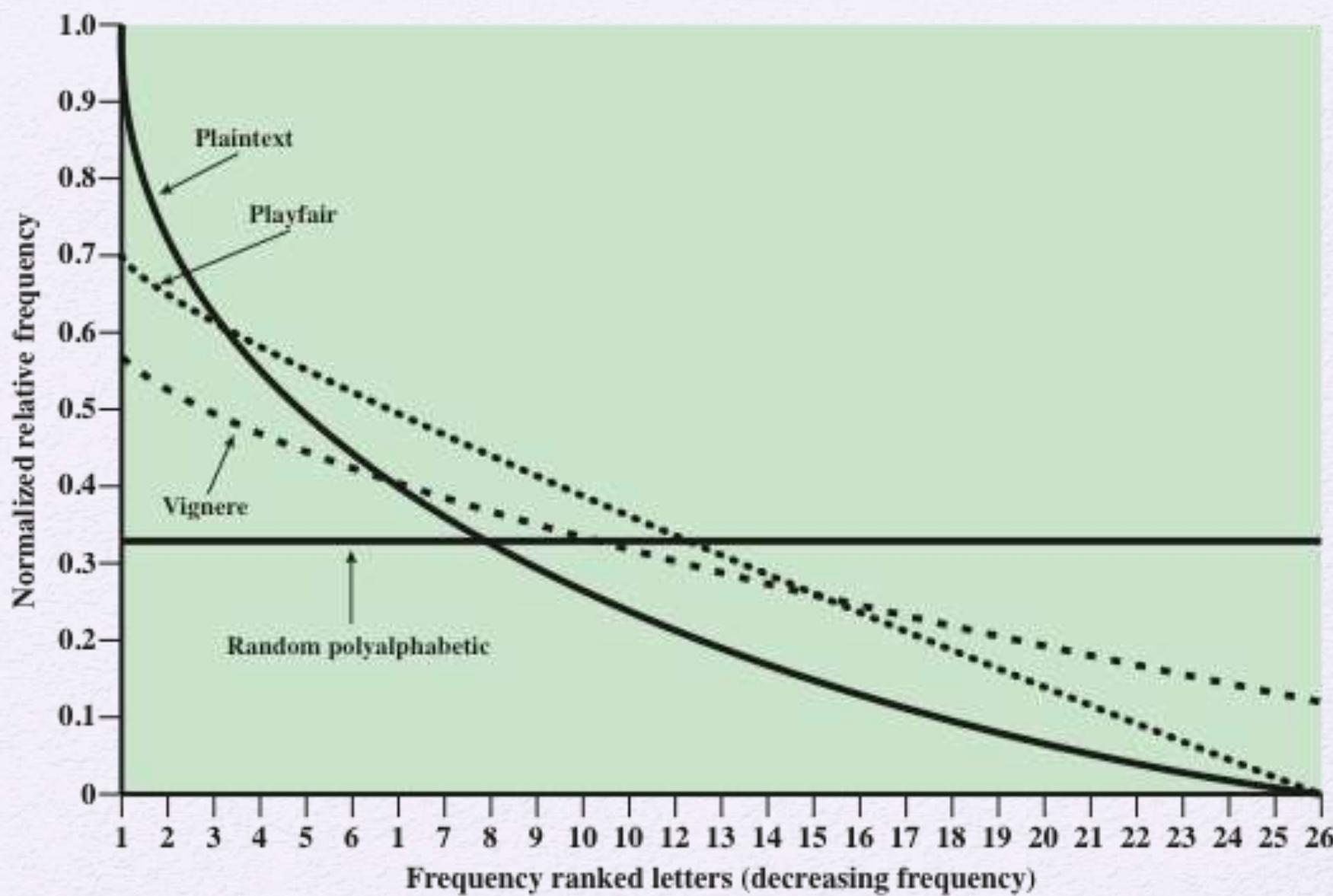


Figure 3.6 Relative Frequency of Occurrence of Letters

Hill Cipher

- Developed by the mathematician Lester Hill in 1929
- Strength is that it completely hides single-letter frequencies
 - The use of a larger matrix hides more frequency information
 - A 3×3 Hill cipher hides not only single-letter but also two-letter frequency information
- Strong against a ciphertext-only attack but easily broken with a known plaintext attack

Polyalphabetic Ciphers

- Polyalphabetic substitution cipher
 - Improves on the simple monoalphabetic technique by using different monoalphabetic substitutions as one proceeds through the plaintext message

All these techniques have the following features in common:

- A set of related monoalphabetic substitution rules is used
- A key determines which particular rule is chosen for a given transformation

Vigenère Cipher

- Best known and one of the simplest polyalphabetic substitution ciphers
- In this scheme the set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers with shifts of 0 through 25
- Each cipher is denoted by a key letter which is the ciphertext letter that substitutes for the plaintext letter a

Example of Vigenère Cipher

- To encrypt a message, a key is needed that is as long as the message
- Usually, the key is a repeating keyword
- For example, if the keyword is *deceptive*, the message “we are discovered save yourself” is encrypted as:

key: deceptive deceptive deceptive

plaintext: wearediscoveredsaveyourself

ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMJ

Vigenère Autokey System

- A keyword is concatenated with the plaintext itself to provide a running key

- Example:

key: deceptivewearediscoveredsav

plaintext: wearediscoveredsaveyourself

ciphertext: ZICVTWQNGKZEIIGASXSTSLVVWLA

- Even this scheme is vulnerable to cryptanalysis
 - Because the key and the plaintext share the same frequency distribution of letters, a statistical technique can be applied

Vernam Cipher

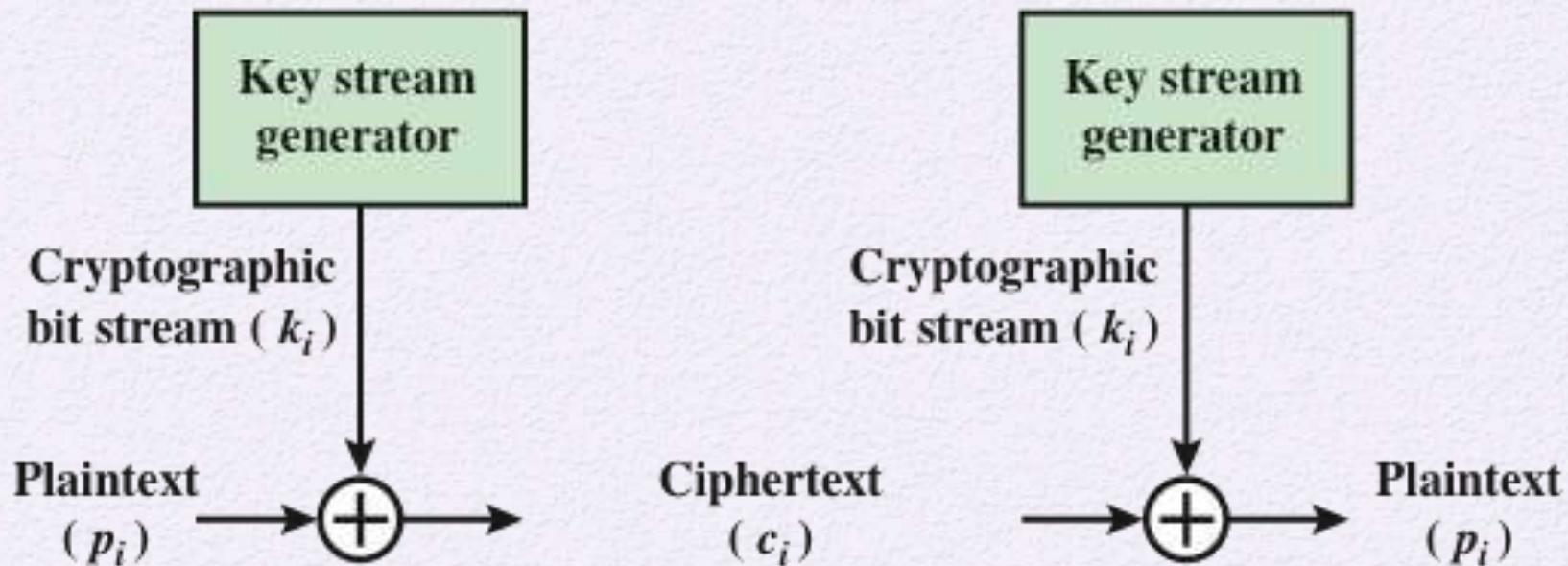


Figure 3.7 Vernam Cipher

One-Time Pad

- Improvement to Vernam cipher proposed by an Army Signal Corp officer, Joseph Mauborgne
- Use a random key that is as long as the message so that the key need not be repeated
- Key is used to encrypt and decrypt a single message and then is discarded
- Each new message requires a new key of the same length as the new message
- Scheme is unbreakable
 - Produces random output that bears no statistical relationship to the plaintext
 - Because the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code



One-Time Pad

ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUFPLUYTS

We now show two different decryptions using two different keys:

ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUFPLUYTS

key: pxlmvmsydoфuyrvzwc tnleбnecvgdupahfzzlmnyih

plaintext: mr mustard with the candlestick in the hall

ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUFPLUYTS

key: pftgpmiydgaхgoufhk111mhsqdqogtewbqfgyovuhwt

plaintext: miss scarlet with the knife in the library

Suppose that a cryptanalyst had managed to find these two keys. Two plausible plaintexts are produced. How is the cryptanalyst to decide which is the correct one? (The answer is that it can't be done in a truly random manner.)



Difficulties

- The one-time pad offers complete security but, in practice, has two fundamental difficulties:
 - There is the practical problem of making large quantities of random keys
 - Any heavily used system might require millions of random characters on a regular basis
 - Mammoth key distribution problem
 - For every message to be sent, a key of equal length is needed by both sender and receiver
- Because of these difficulties, the one-time pad is of limited utility
 - Useful primarily for low-bandwidth channels requiring very high security
- The one-time pad is the only cryptosystem that exhibits perfect secrecy (see Appendix F)

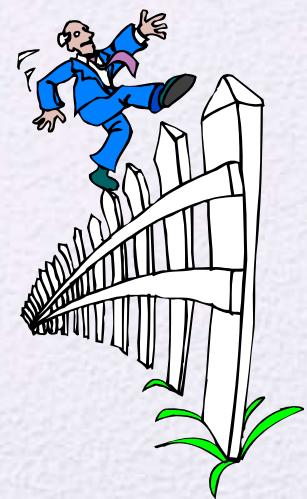
Rail Fence Cipher

- Simplest transposition cipher
- Plaintext is written down as a sequence of diagonals and then read off as a sequence of rows
- To encipher the message “meet me after the toga party” with a rail fence of depth 2, we would write:

m e m a t r h t g p r y
e t e f e t e o a a t

Encrypted message is:

MEMATRHTGPRYETEFETEOAAT



Row Transposition Cipher

- Is a more complex transposition
- Write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns
 - The order of the columns then becomes the key to the algorithm

Key: 4 3 1 2 5 6 7

Plaintext: a t t a c k p

o s t p o n e

d u n t i l t

w o a m x y z

Ciphertext: TTNAAPMTSUOAODWCOIXKNLYPETZ

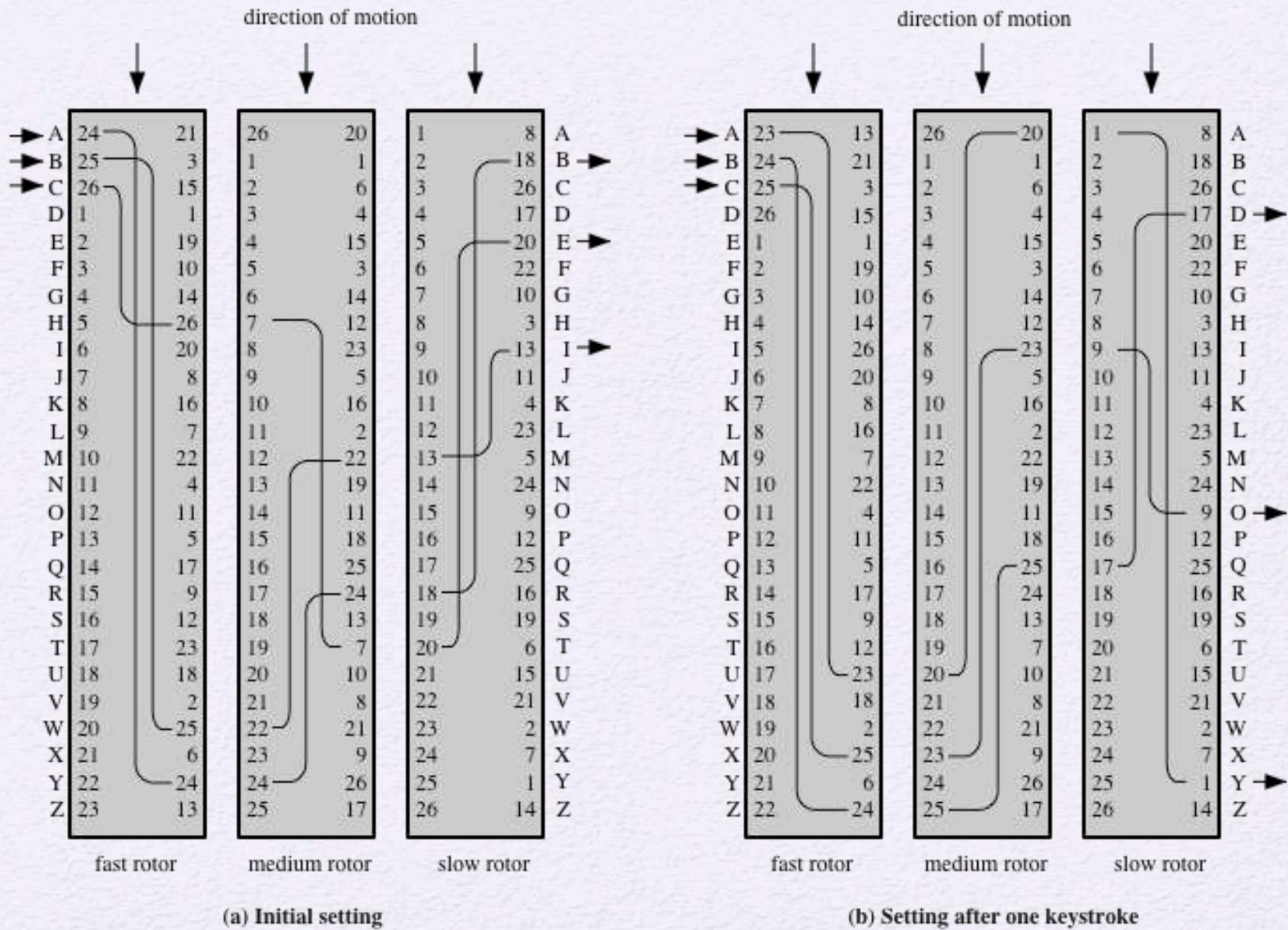


Figure 3.8 Three-Rotor Machine With Wiring Represented by Numbered Contacts

Steganography

3rd March

Dear George,

Greetings to all at Oxford. Many thanks for your letter and for the Summer examination package. All Entry Forms and Fees Forms should be ready for final despatch to the Syndicate by Friday 20th or at the very latest, I'm told, by the 21st. Admin has improved here, though there's room for improvement still; just give us all two or three more years and we'll really show you! Please don't let these wretched 16+ proposals destroy your basic O and A pattern. Certainly this sort of change, if implemented immediately, would bring chaos.

Sincerely yours,

A Puzzle for Inspector Morse
(from *The Silent World of Nicholas Quinn*, by Colin Dexter)

Other Steganography Techniques



- Character marking
 - Selected letters of printed or typewritten text are over-written in pencil
 - The marks are ordinarily not visible unless the paper is held at an angle to bright light
- Invisible ink
 - A number of substances can be used for writing but leave no visible trace until heat or some chemical is applied to the paper
- Pin punctures
 - Small pin punctures on selected letters are ordinarily not visible unless the paper is held up in front of a light
- Typewriter correction ribbon
 - Used between lines typed with a black ribbon, the results of typing with the correction tape are visible only under a strong light

Steganography vs. Encryption

- Steganography has a number of drawbacks when compared to encryption
 - It requires a lot of overhead to hide a relatively few bits of information
 - Once the system is discovered, it becomes virtually worthless

- The advantage of steganography
 - It can be employed by parties who have something to lose should the fact of their secret communication (not necessarily the content) be discovered
 - Encryption flags traffic as important or secret or may identify the sender or receiver as someone with something to hide

Summary

- Symmetric Cipher Model
 - Cryptography
 - Cryptanalysis and Brute-Force Attack
- Transposition techniques
- Rotor machines
- Substitution techniques
 - Caesar cipher
 - Monoalphabetic ciphers
 - Playfair cipher
 - Hill cipher
 - Polyalphabetic ciphers
 - One-time pad
- Steganography



Advanced Encryption Standard

[Slide courtesy: Cryptography and network security by Behrouz Fou�zan]

AES

- Published by NIST (National Institute of Standards and Technology) in December 2001
 - First AES candidate conference
 - 15 out of 21 algorithms selected
 - Second AES candidate conference
 - 5 out of 15 selected as finalists
 - MARS, RC6, Rijndael, Serpent and Twofish
 - Third AES candidate conference
 - NIST announced Rijndael as the AES

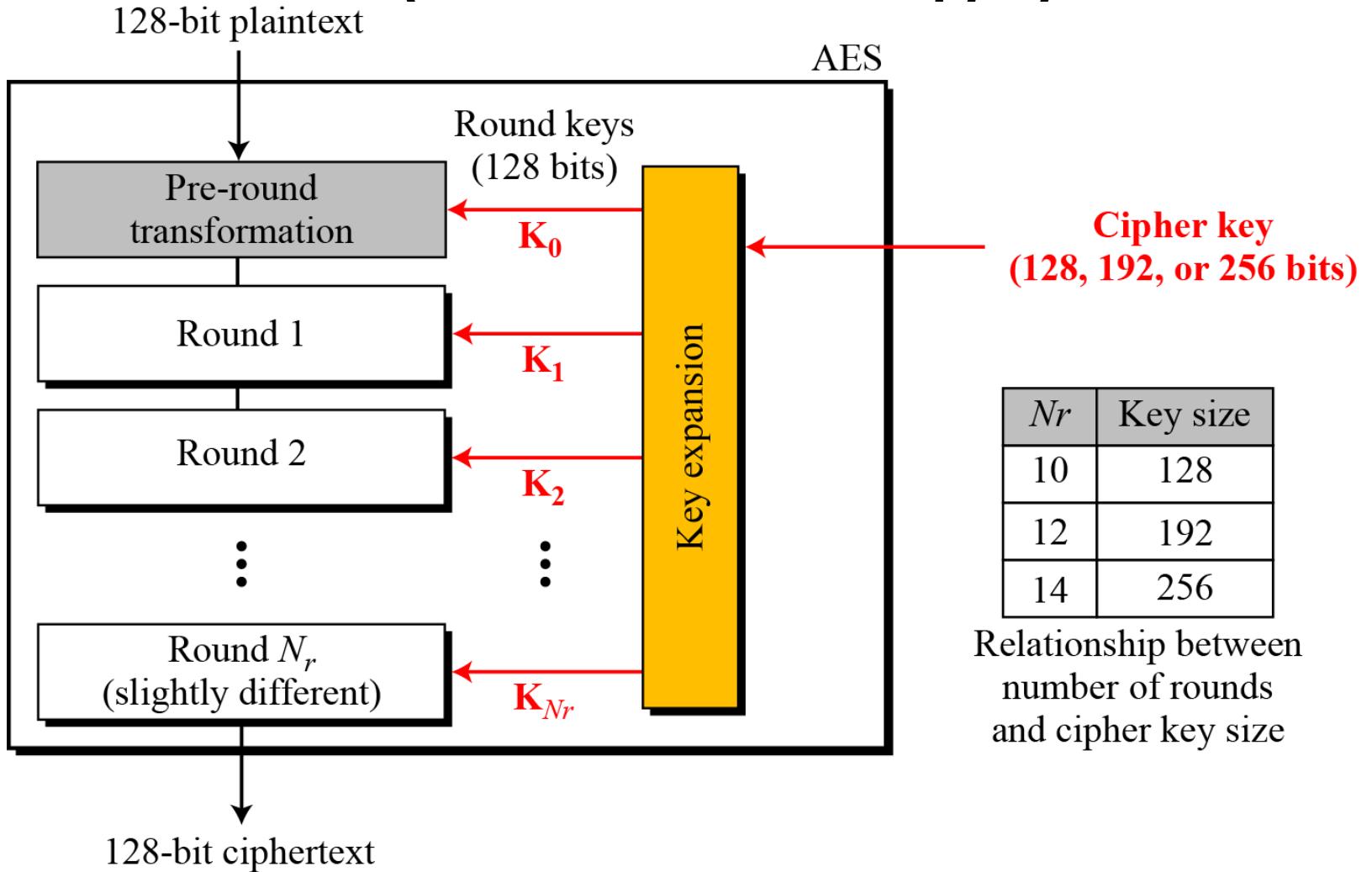
AES...

- The criteria defined by NIST for selecting AES fall into three areas:
 - Security
 - Cost
 - Implementation

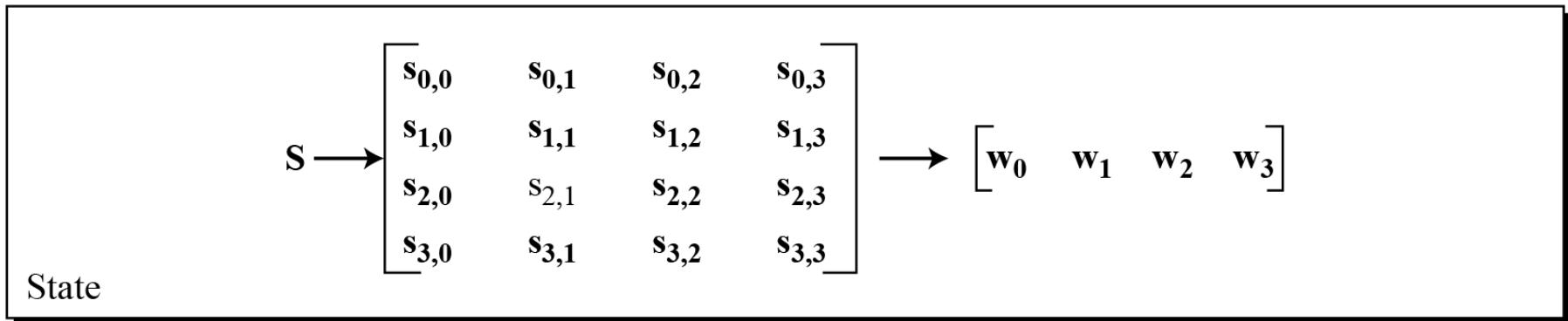
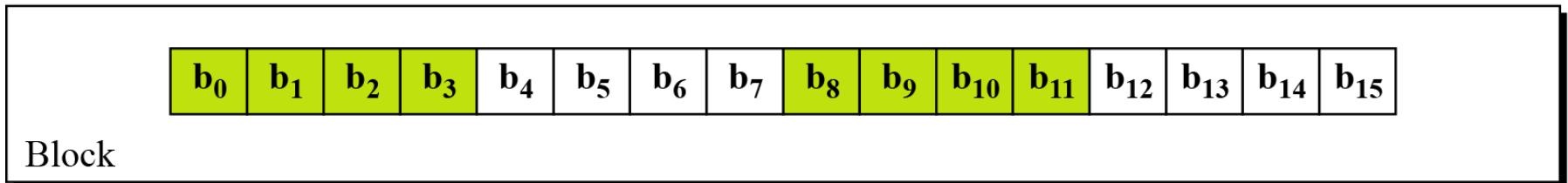
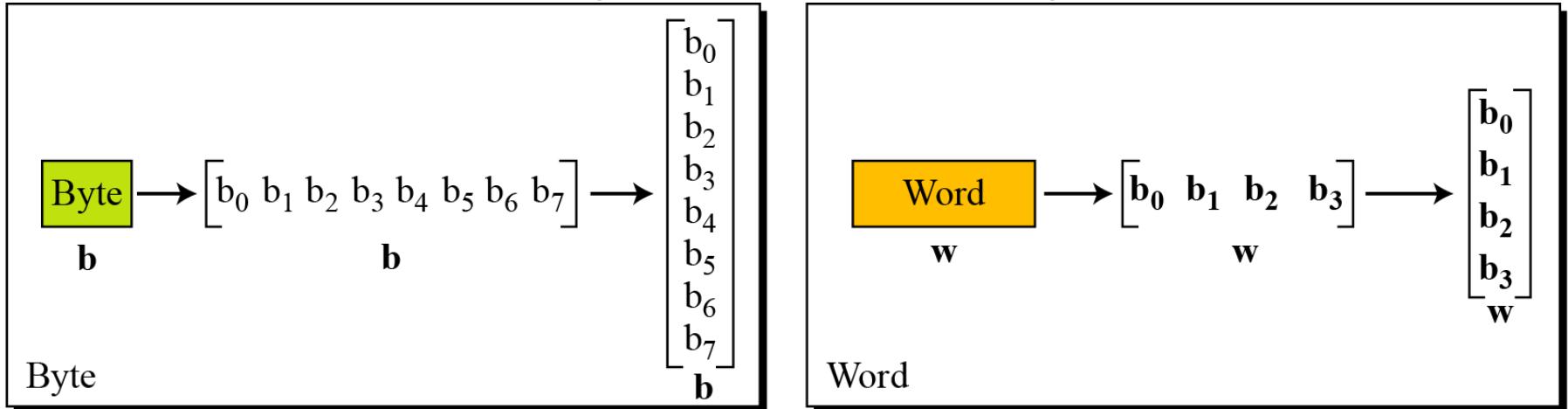
AES...

- AES is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits.
- It uses 10, 12, or 14 rounds.
- The key size, which can be 128, 192, or 256 bits, depends on the number of rounds.
 - But the round keys are always 128 bits

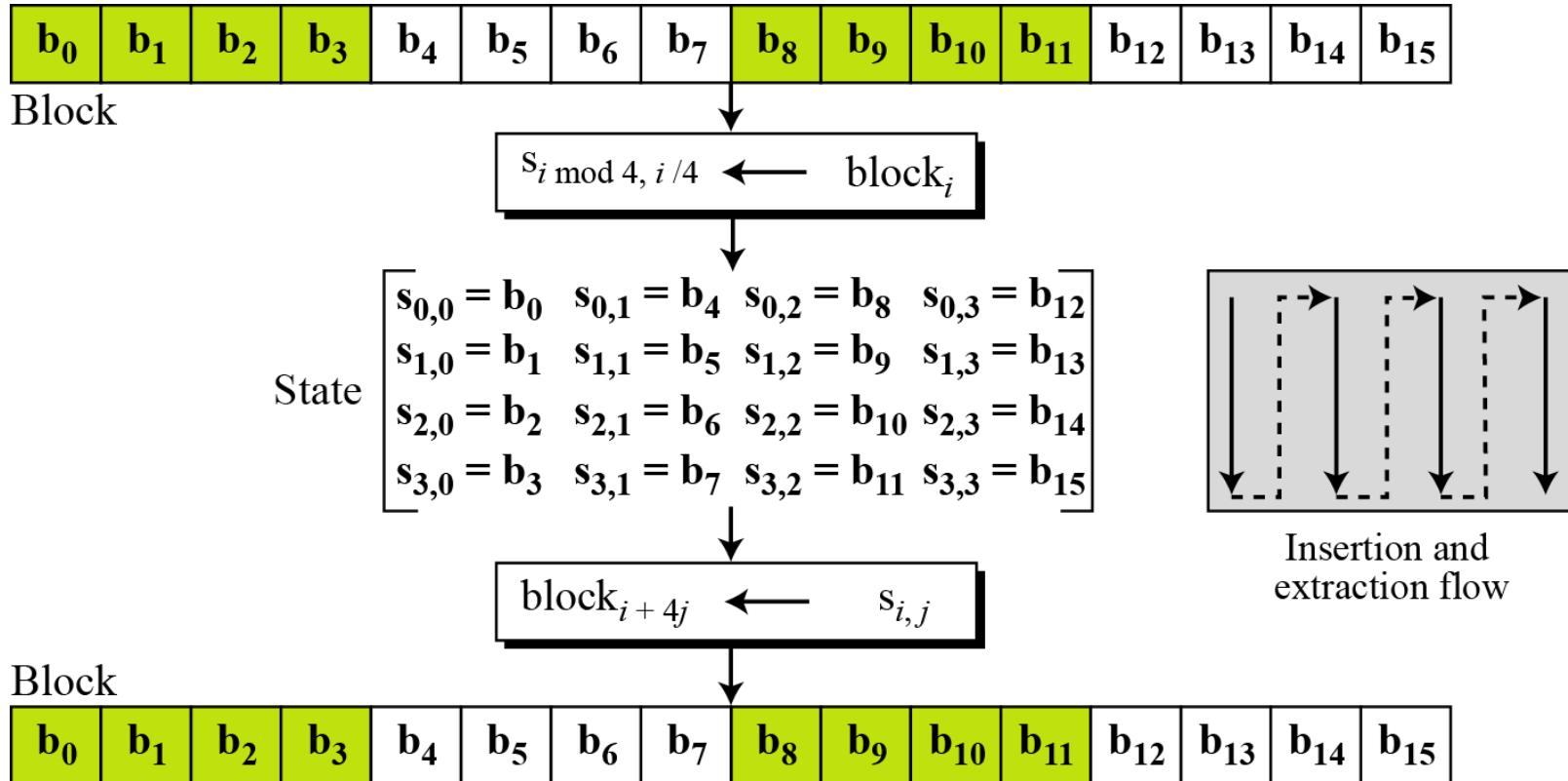
AES (General Design)...



AES (Data Units)...



AES (Data Units)...

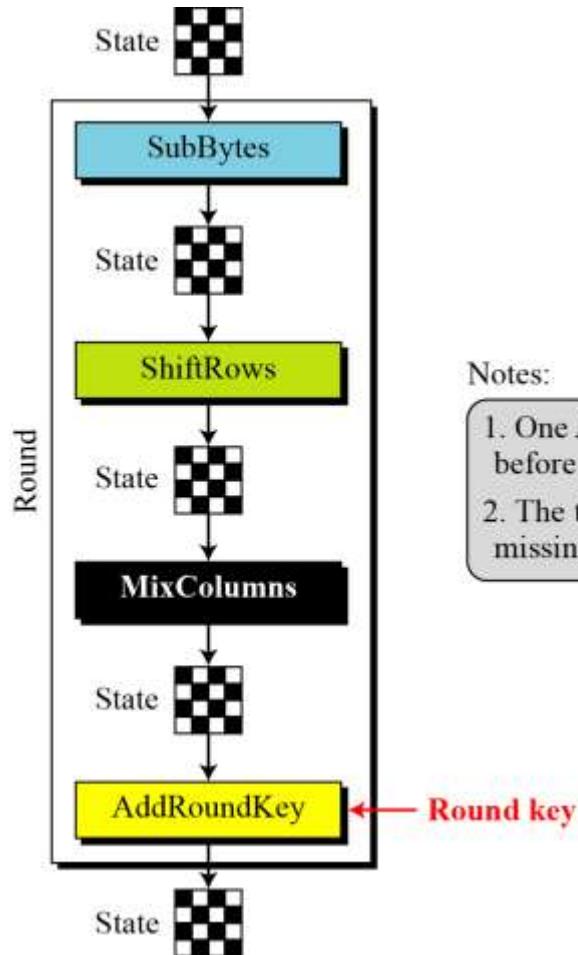


AES (Data Units)...

- Changing plaintext to states

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
$\begin{bmatrix} 00 & 12 & 0C & 08 \\ 04 & 04 & 00 & 23 \\ 12 & 12 & 13 & 19 \\ 14 & 00 & 11 & 19 \end{bmatrix}$ State																

Structure of each round



Notes:

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

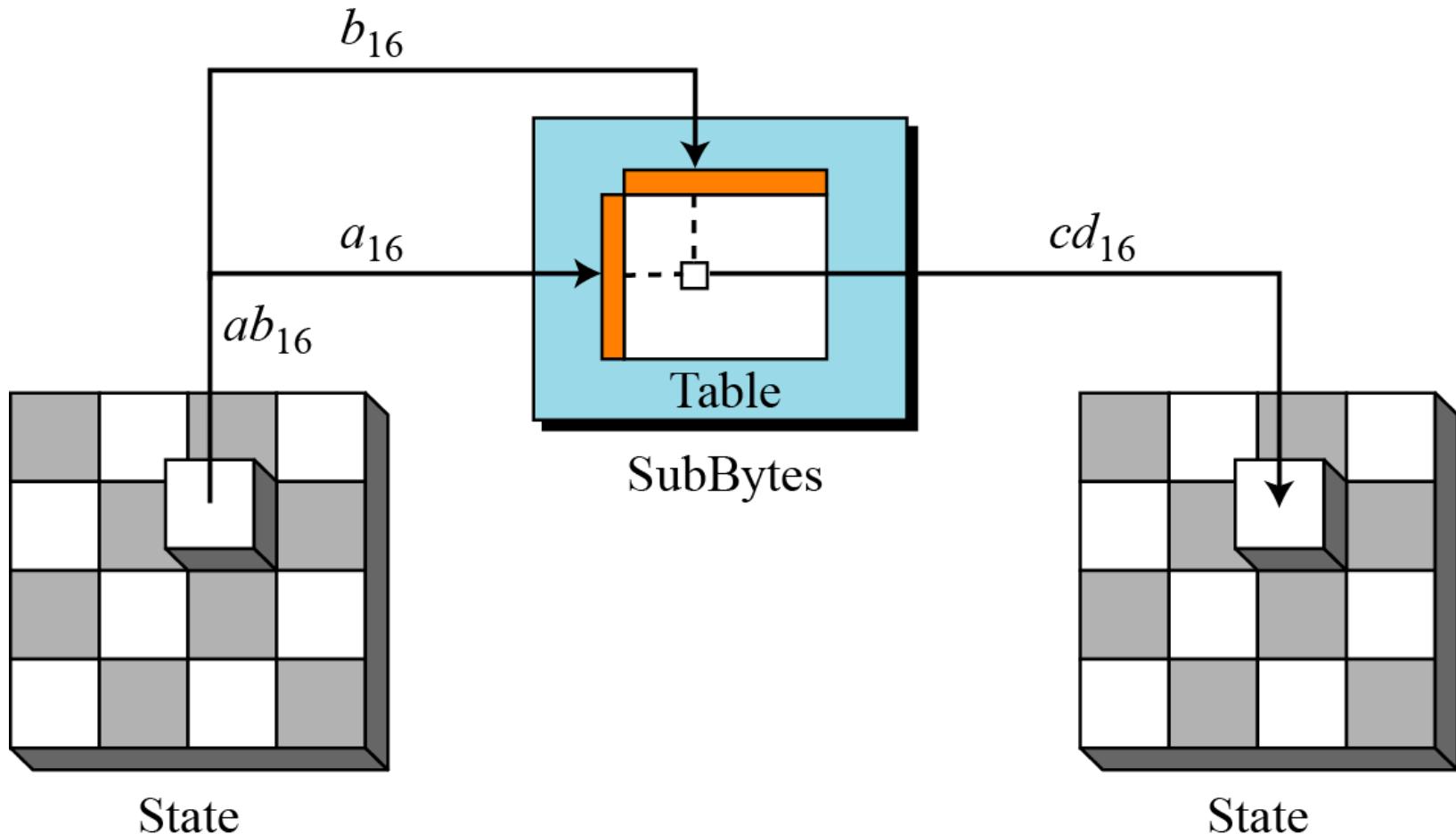
AES Transformations

- Four types
 - Substitution
 - Permutation
 - Mixing
 - Key-adding

Substitution

- AES, like DES, uses substitution. AES uses two invertible transformations.
- SubBytes
 - The first transformation, SubBytes, is used at the encryption site.
 - To substitute a byte, we interpret the byte as two hexadecimal digits.
 - The SubBytes operation involves 16 independent byte-to-byte transformations.

Substitution...



Substitution...

Table 7.1 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

Substitution...

Table 7.1 SubBytes transformation table (continued)

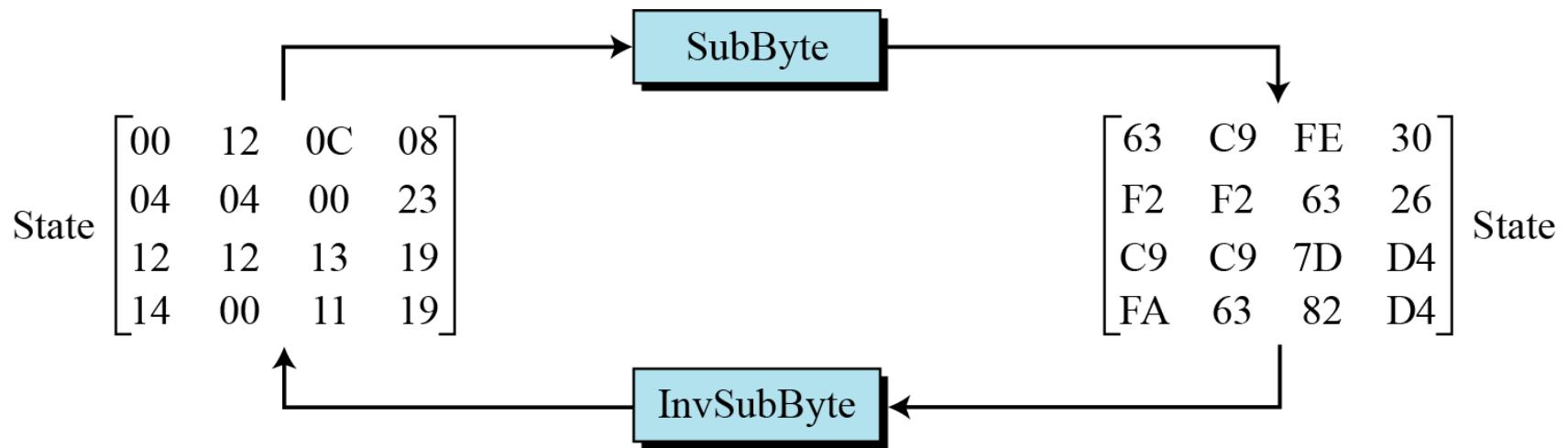
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	CB	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	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

Substitution...

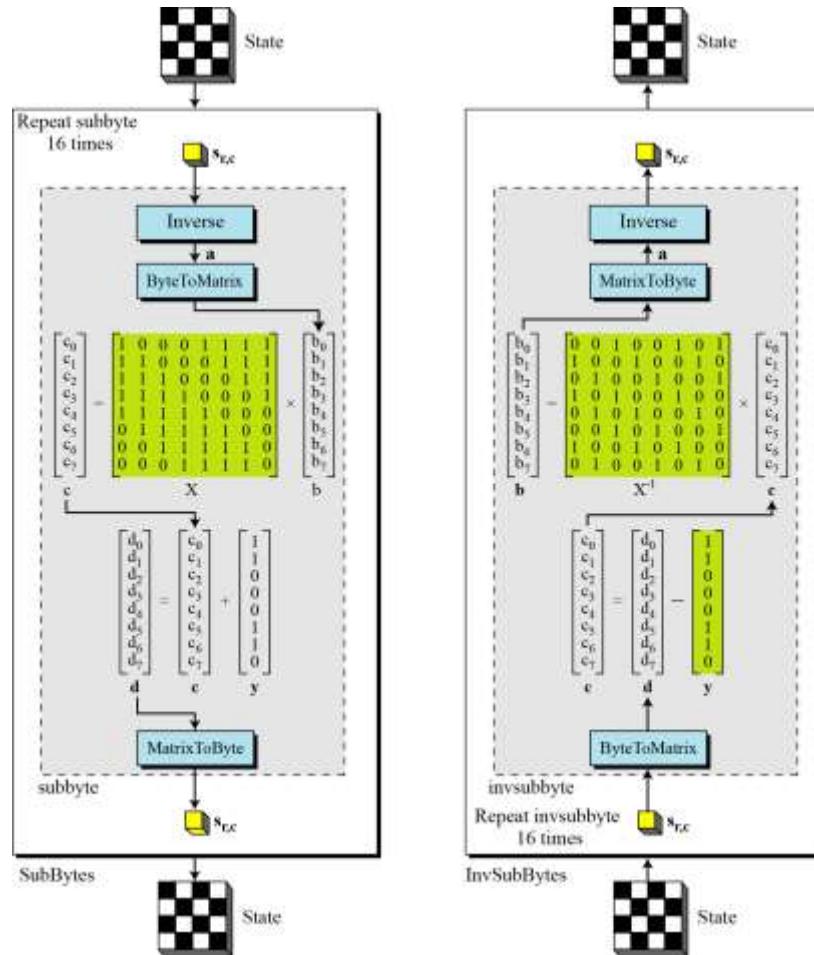
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	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

Substitution...

- Example



SubBytes and InvSubBytes processes



SubBytes and InvSubBytes processes...

- Example
 - Let us show how the byte 0C is transformed to FE by subbyte routine and transformed back to 0C by the invsubbyte routine.

1. *subbyte*:
 - a. The multiplicative inverse of 0C in GF(2^8) field is B0, which means **b** is (10110000).
 - b. Multiplying matrix **X** by this matrix results in **c** = (10011101)
 - c. The result of XOR operation is **d** = (11111110), which is FE in hexadecimal.
2. *invsubbyte*:
 - a. The result of XOR operation is **c** = (10011101)
 - b. The result of multiplying by matrix **X⁻¹** is (11010000) or B0
 - c. The multiplicative inverse of B0 is 0C.

SubBytes and InvSubBytes processes...

Algorithm 7.1 *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
}
```

Substitution...

- Exercise
 1. Can you prove formally that the subbyte and invsubbyte are inverses of each other?
 2. Can you prove that the subbyte transformation is nonlinear?

Substitution...

- Exercise- solution
 1. Can you prove formally that the subbyte and invsubbyte are inverses of each other?

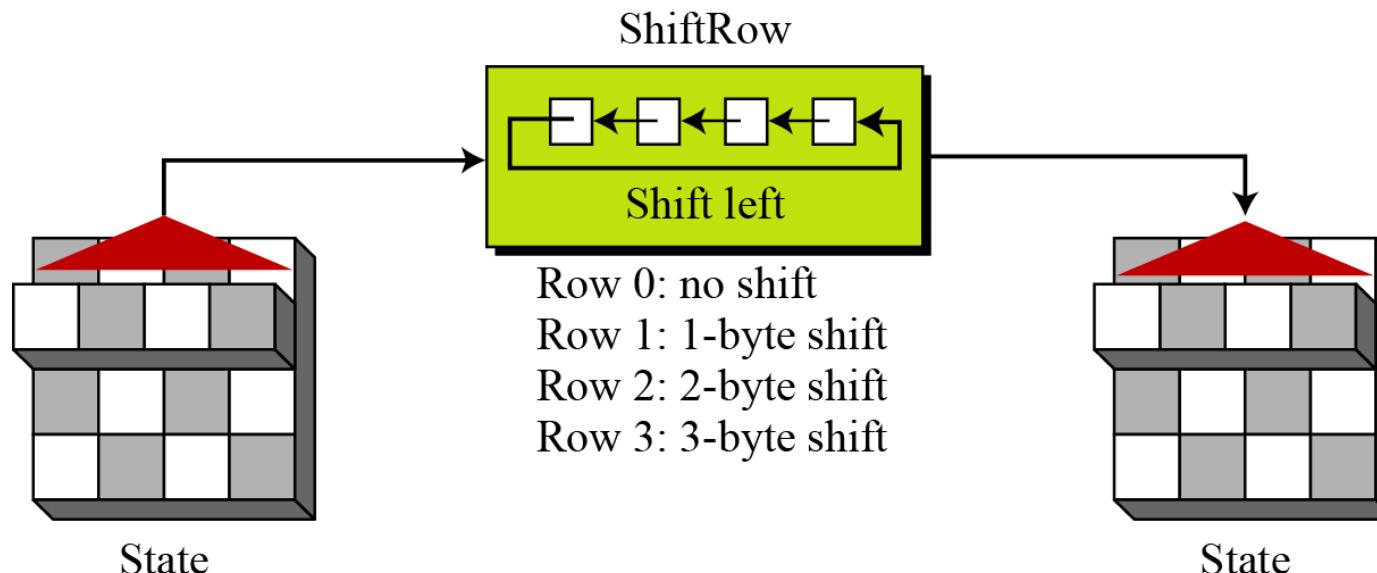
$$\text{subbyte: } \rightarrow \mathbf{d} = \mathbf{X} (s_{r,c})^{-1} \oplus \mathbf{y}$$

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

2. It is the inverse operation, that makes the whole transformation nonlinear

Permutation

- ShiftRows
 - In the encryption, the transformation is called ShiftRows.



Permutation...

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

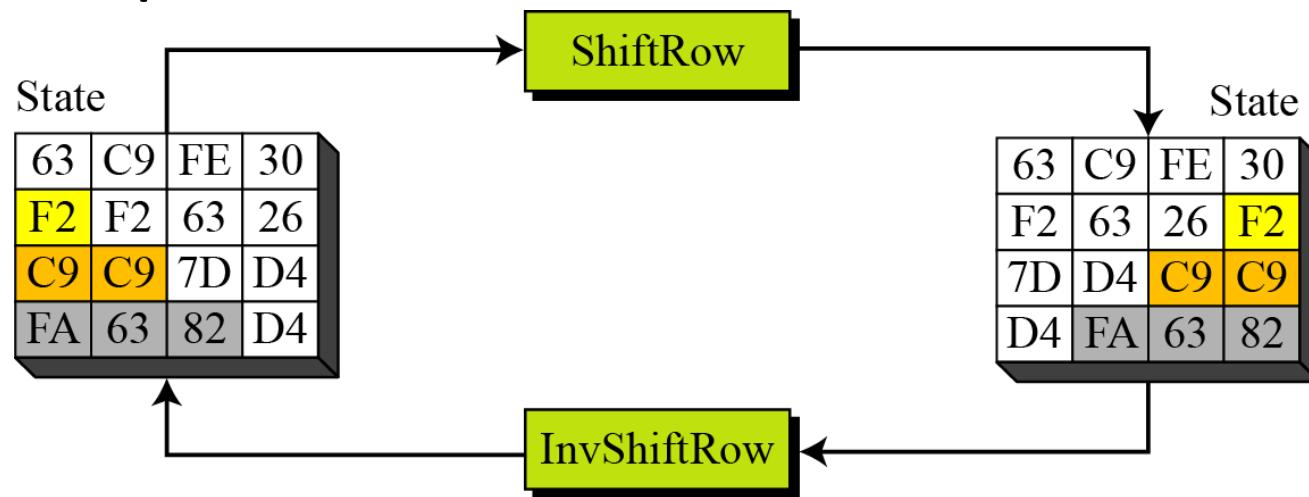
Algorithm 7.2 *Pseudocode for ShiftRows transformation*

```
ShiftRows (S)
{
    for (r = 1 to 3)
        shiftrow (sr, r)           // sr is the rth 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 ← tc
}
```

Permutation...

- Example



Mixing

- An interbyte transformation that changes the bits inside a byte, based on the bits inside the neighboring bytes.
 - We need to mix bytes to provide diffusion at the bit level.

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

New matrix **Constant matrix** Old matrix

Mixing...

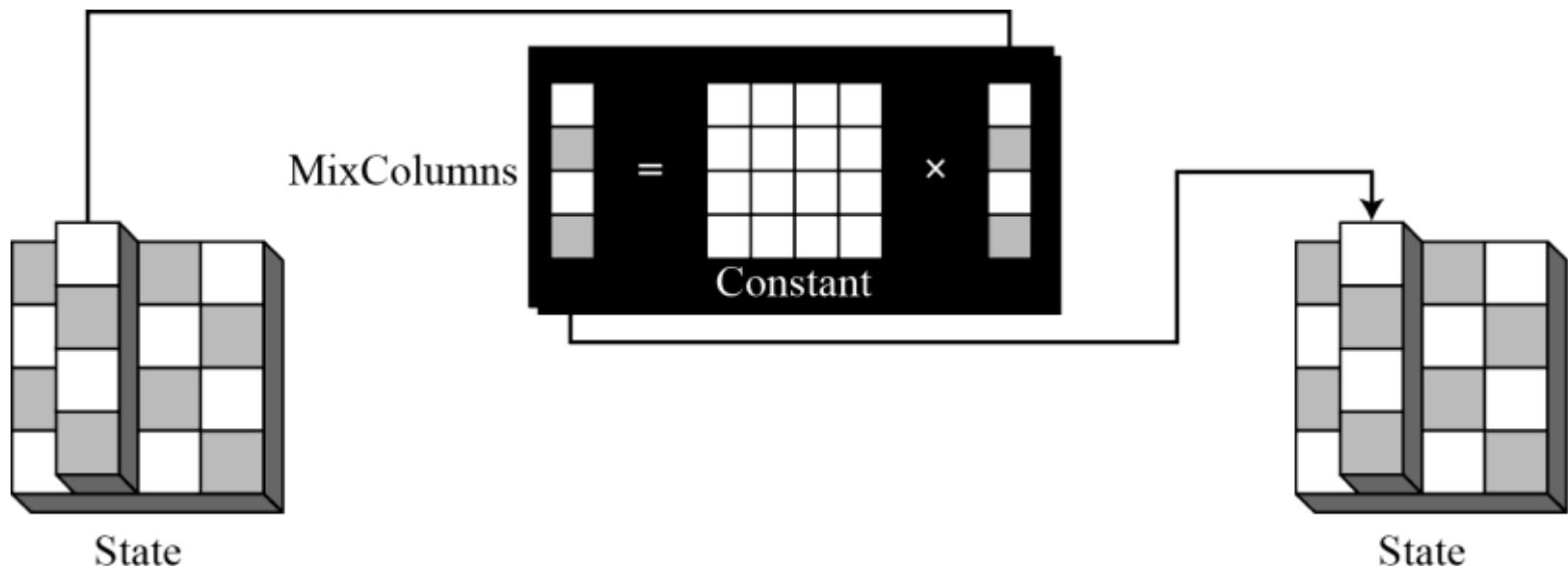
- Constant matrices used by MixColumns and InvMixColumns

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \xleftrightarrow{\text{Inverse}} \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}$$

C C⁻¹

Mixing...

- MixColumns
 - operates at the column level
 - transforms each column of the state to a new column.



Mixing...

- InvMixColumns
 - basically the same as the MixColumns transformation but the inverse

Algorithm 7.3 *Pseudocode for MixColumns transformation*

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

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

    col0  $\leftarrow$  (0x02) • t0  $\oplus$  (0x03 • t1)  $\oplus$  t2  $\oplus$  t3

    col1  $\leftarrow$  t0  $\oplus$  (0x02) • t1  $\oplus$  (0x03) • t2  $\oplus$  t3

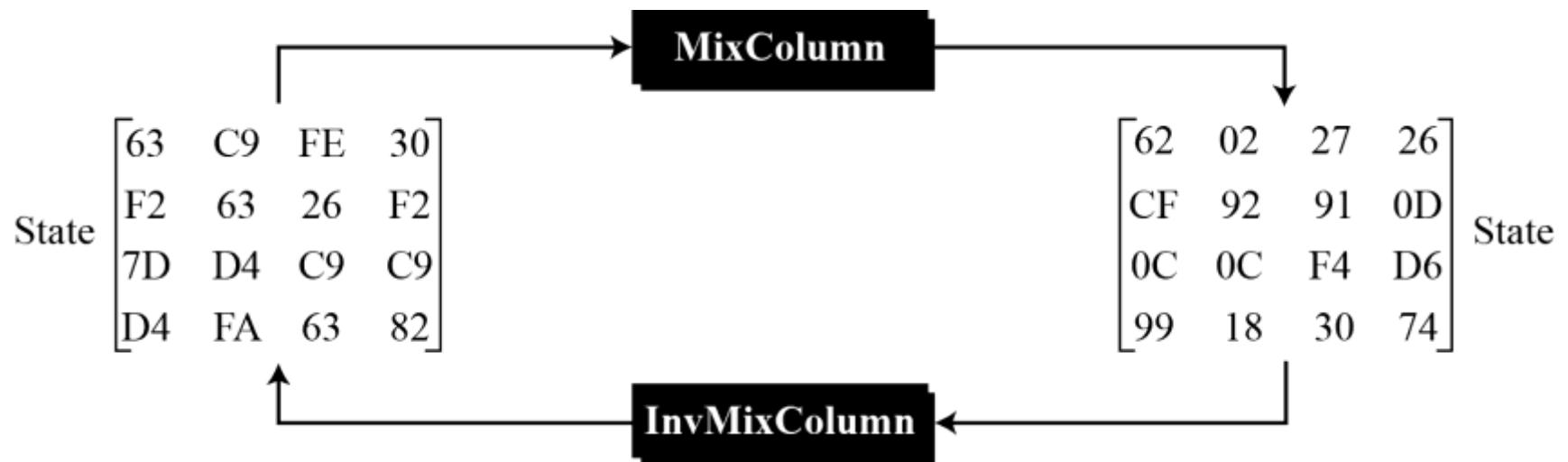
    col2  $\leftarrow$  t0  $\oplus$  t1  $\oplus$  (0x02) • t2  $\oplus$  (0x03) • t3

    col3  $\leftarrow$  (0x03 • t0)  $\oplus$  t1  $\oplus$  t2  $\oplus$  (0x02) • t3
}
```

Mixing...

- Example

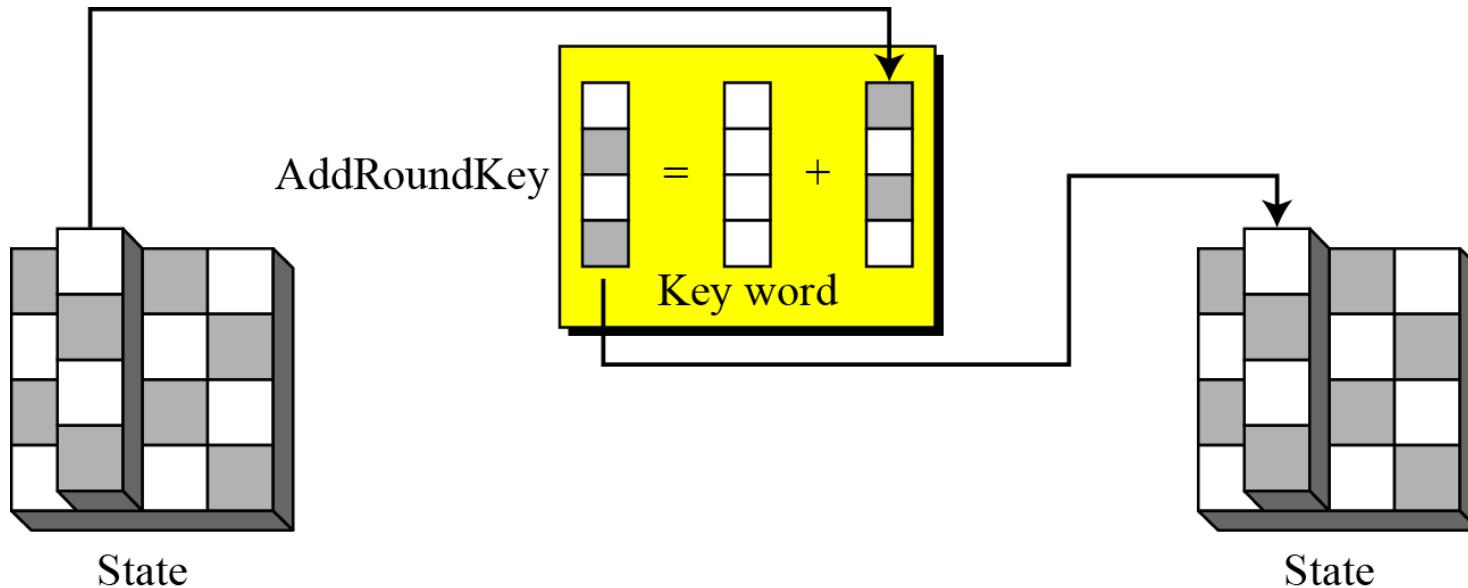
- how a state is transformed using the MixColumns transformation. The figure also shows that the InvMixColumns transformation creates the original one.



Key Adding

- AddRoundKey
 - 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.

Key Adding...



Algorithm 7.4 Pseudocode for AddRoundKey transformation

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

Key Expansion

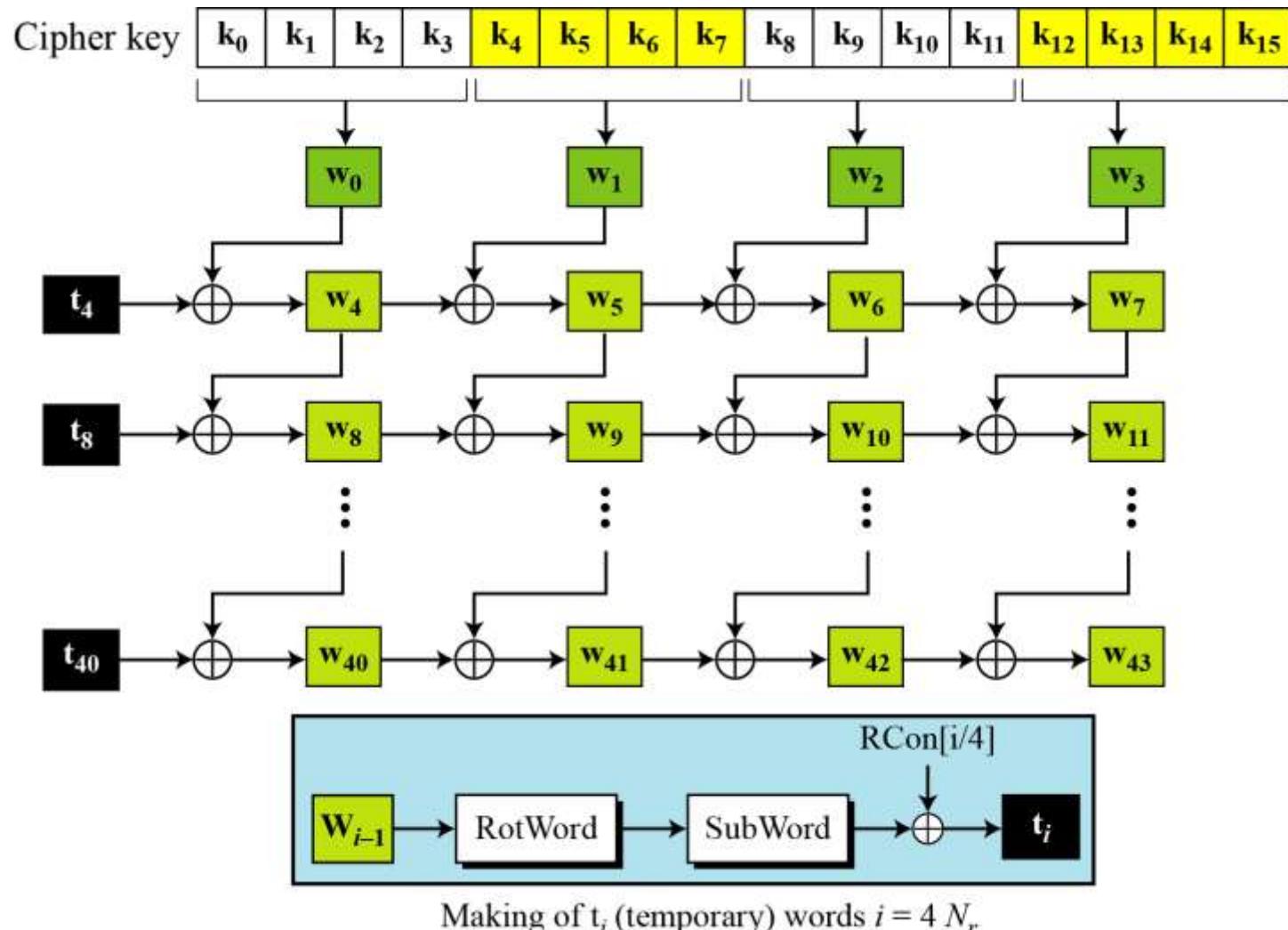
- To create round keys for each round, AES uses a key-expansion process.
- If the number of rounds is N_r , the key-expansion routine creates $N_r + 1$ 128-bit round keys from one single 128-bit cipher key.

Key Expansion...

Table 7.3 *Words for each round*

<i>Round</i>	<i>Words</i>			
Pre-round	\mathbf{w}_0	\mathbf{w}_1	\mathbf{w}_2	\mathbf{w}_3
1	\mathbf{w}_4	\mathbf{w}_5	\mathbf{w}_6	\mathbf{w}_7
2	\mathbf{w}_8	\mathbf{w}_9	\mathbf{w}_{10}	\mathbf{w}_{11}
...	...			
N_r	\mathbf{w}_{4N_r}	\mathbf{w}_{4N_r+1}	\mathbf{w}_{4N_r+2}	\mathbf{w}_{4N_r+3}

Key Expansion in AES-128...



Key Expansion in AES-128...

Table 7.4 *RCon constants*

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

Key Expansion in AES-128...

- The key-expansion routine can either use the above table when calculating the words or use the GF(2⁸) field to calculate the leftmost byte dynamically, as shown below (prime is the irreducible polynomial):

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

Key Expansion in AES-128...

- An illustration

- 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	$w_{32} = EE61ACDE$	$w_{33} = EAFC1F4B$	$w_{34} = 2F62A6E6$	$w_{35} = D8E39D92$
9	0A5E4F61	$w_{36} = E43FE3BF$	$w_{37} = 0EC1FCF4$	$w_{38} = 21A35A12$	$w_{39} = F940C780$
10	3FC6CD99	$w_{40} = DBF92E26$	$w_{41} = D538D2D2$	$w_{42} = F49B88C0$	$w_{43} = 0DDDB4F40$

Key Expansion in AES-128...

- Each round key in AES depends on the previous round key.
- The dependency, however, is nonlinear because of SubWord transformation.
- The addition of the round constants also guarantees that each round key will be different from the previous one.
- An illustration

Cipher Key 1:	12 45 A2 A1 23 31 A4 A3	B2 CC <u>AA</u> 34	C2 BB 77 23
Cipher Key 2:	12 45 A2 A1 23 31 A4 A3	B2 CC <u>AB</u> 34	C2 BB 77 23

Key Expansion in AES-128...

Table 7.6 Comparing two sets of round keys

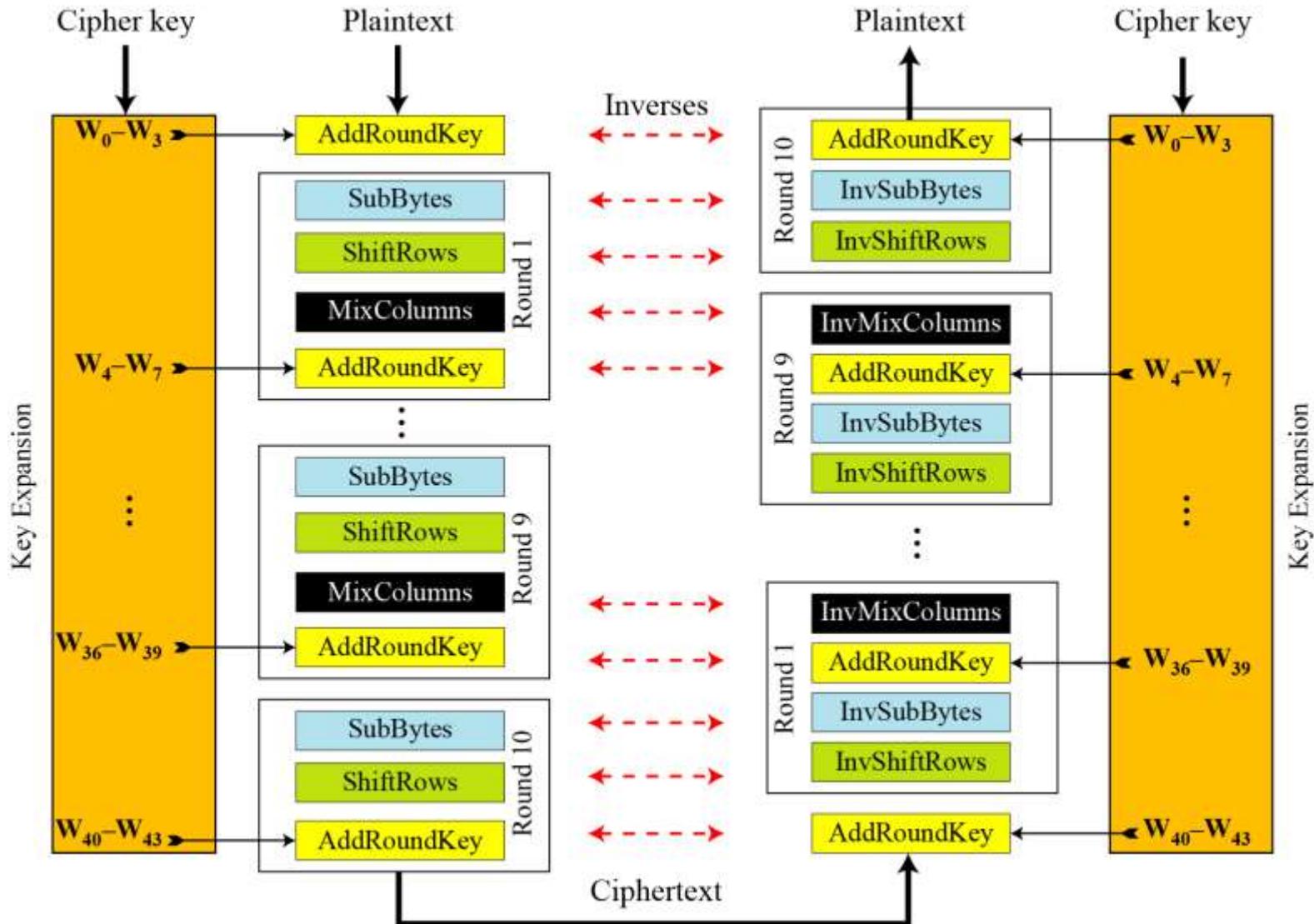
R.	Round keys for set 1	Round keys for set 2	B. D.
—	1245A2A1 2331A4A3 B2CCAA <u>34</u> C2BB7723	1245A2A1 2331A4A3 B2CCAB <u>34</u> C2BB7723	01
1	F9B08484 DA812027 684D8 <u>A1</u> 3 AAF6 <u>FD</u> 30	F9B08484 DA812027 684D8 <u>B1</u> 3 AAF6 <u>FC</u> 30	02
2	B9E48028 6365A00F 0B282A1C A1DED72C	B9008028 6381A00F 0BCC2B1C A13AD72C	17
3	A0EAF11A C38F5115 C8A77B09 6979AC25	3D0EF11A 5E8F5115 55437A09 F479AD25	30
4	1E7BCEE3 DDF49FF6 1553E4FF 7C2A48DA	839BCEA5 DD149FB0 8857E5B9 7C2E489C	31
5	EB2999F3 36DD0605 238EE2FA 5FA4AA20	A2C910B5 7FDD8F05 F78A6ABC 8BA42220	34
6	82852E3C B4582839 97D6CAC3 C87260E3	CB5AA788 B487288D 430D4231 C8A96011	56
7	82553FD4 360D17ED A1DBDD2E 69A9BDCD	588A2560 EC0D0DED AF004FDC 67A92FCD	50
8	D12F822D E72295C0 46F948EE 2F50F523	0B9F98E5 E7929508 4892DAD4 2F3BF519	44
9	99C9A438 7EEB31F8 38127916 17428C35	F2794CF0 15EBD9F8 5D79032C 7242F635	51
10	83AD32C8 FD460330 C5547A26 D216F613	E83BDAB0 FDD00348 A0A90064 D2EBF651	52

Key Expansion in AES-128...

- What about **weak keys** for AES???

Pre-round:	00000000	00000000	00000000	00000000
Round 01:	62636363	62636363	62636363	62636363
Round 02:	9B9898C9	F9FBFBAA	9B9898C9	F9FBFBAA
Round 03:	90973450	696CCFFA	F2F45733	0B0FAC99
...
Round 10:	B4EF5BCB	3E92E211	23E951CF	6F8F188E

The cipher



Analysis

- The result of encryption when the plaintext is made of all 0s.

Plaintext:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Cipher Key:	24	75	A2	B3	34	75	56	88	31	E2	12	00	13	AA	54	87			
Ciphertext:	63	2C	D4	5E	5D	56	ED	B5	62	04	01	A0	AA	9C	2D	8D			

Analysis...

- The avalanche effect.

Plaintext 1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Plaintext 2: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01

Ciphertext 1: 63 2C D4 5E 5D 56 ED B5 62 04 01 A0 AA 9C 2D 8D

Ciphertext 2: 26 F3 9B BC A1 9C 0F B7 C7 2E 7E 30 63 92 73 13

Analysis...

- The effect of using a cipher key in which all bits are 0s.

Plaintext:	00	04	12	14	12	04	12	00	0c	00	13	11	08	23	19	19
Cipher Key:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Ciphertext:	5A	6F	4B	67	57	B7	A5	D2	C4	30	91	ED	64	9A	42	72

Assignment-1

- In a cipher, S-boxes can be either static or dynamic. The parameters in a static S-box do not depend on the key.
 - State some advantages and some disadvantages of static and dynamic S-boxes.
 - Are the S-boxes(substitution tables) in AES static or dynamic?

Assignment-2

- AES has a larger block size than DES. Is this an advantage or disadvantage?
- AES defines different implementation with three different numbers of rounds. DES defines only one implementation with 16 rounds. What are the advantages and disadvantages of AES over DES with respect to this difference?

Assignment-3

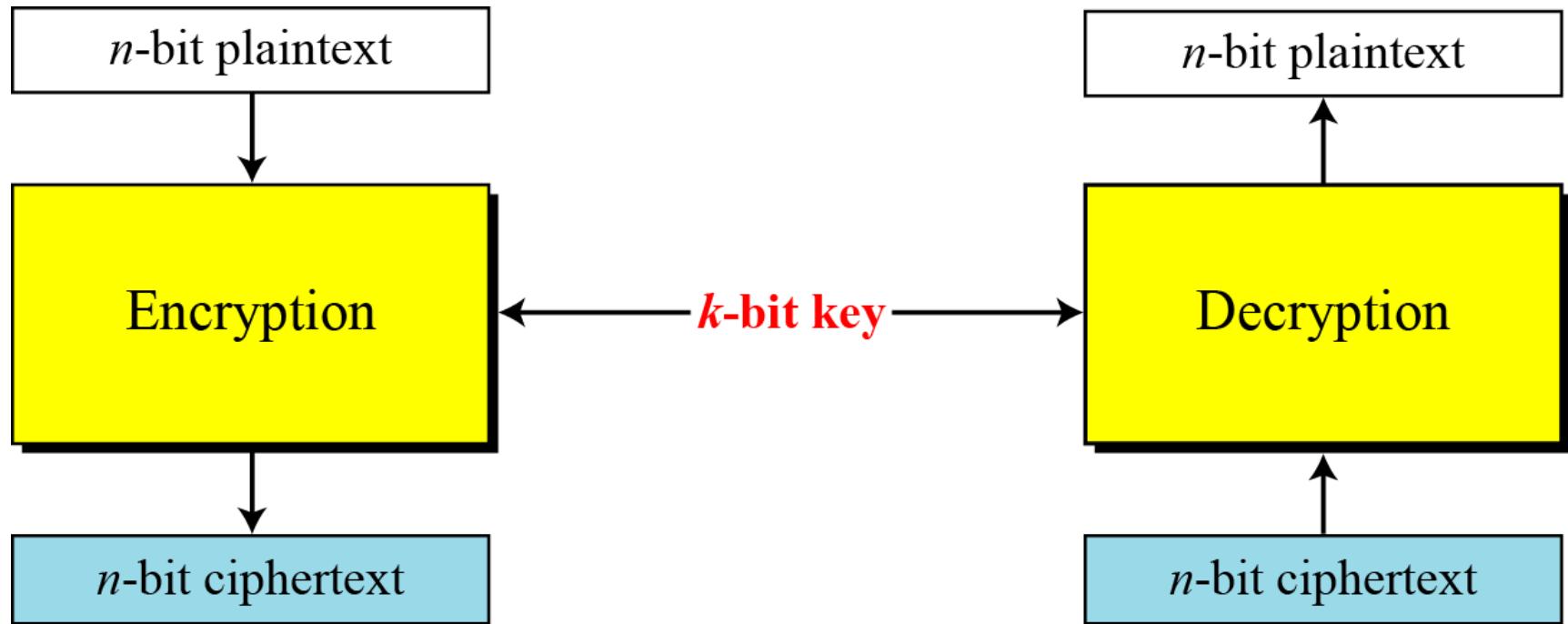
- AES defines three different cipher-key sizes(128, 192 and 256); DES defines only one cipher key size(56). What are the advantages and disadvantages of AES with respect to this difference?

Introduction to Modern Symmetric key ciphers

Modern Block ciphers

A symmetric-key modern block cipher encrypts an **n-bit block** of plaintext or decrypts an **n-bit block** of ciphertext. The encryption or decryption algorithm uses a k-bit key.

Modern Block ciphers...



Modern Block ciphers...

- Example
 - How many padding bits must be added to a message of 100 characters if 8-bit ASCII is used for encoding and the block cipher accepts blocks of 64 bits?

Modern Block ciphers...

- Example
 - How many padding bits must be added to a message of 100 characters if 8-bit ASCII is used for encoding and the block cipher accepts blocks of 64 bits?
- Solution
 - Encoding 100 characters using 8-bit ASCII results in an 800-bit message. The plaintext must be divisible by 64. If $|M|$ and $|Pad|$ are the length of the message and the length of the padding,

$$|M| + |Pad| = 0 \bmod 64 \rightarrow |Pad| = -800 \bmod 64 \rightarrow 32 \bmod 64$$

Substitution or Transposition

- A modern block cipher can be designed to act as a substitution cipher or a transposition cipher.
- Which one do you think is better? Why?

Substitution or Transposition...

- Let us take an example
 - Suppose that we have a block cipher where $n = 64$. If there are 10 1's in the ciphertext, how many trial-and-error tests does Eve need to do to recover the plaintext from the intercepted ciphertext in each of the following cases?
 - a. The cipher is designed as a substitution cipher.
 - b. The cipher is designed as a transposition cipher.

Substitution or Transposition...

- Solution
 - In the first case, Eve has no idea how many 1's are in the plaintext. Eve needs to try all possible 2^{64} 64-bit blocks to find one that makes sense.
 - Would take hundreds of years if Eve would try 1 billion blocks per second !!!
 - In the second case, Eve knows that there are exactly 10 1's in the plaintext. Eve can launch an exhaustive-search attack using only those 64-bit blocks that have exactly 10 1's.
 - Would need to try $(64!)/[(10!)(54!)]$ possibilities
 - Can be successful in less than 3 minutes

Substitution or Transposition

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

Block Ciphers as Permutation Groups

- Is a modern block cipher a group?
 - Full-Size Key Transposition Block Ciphers
 - In a full-size key transposition cipher we need to have $n!$ possible keys, so the key should have $[\log_2(n!)]$ bits.

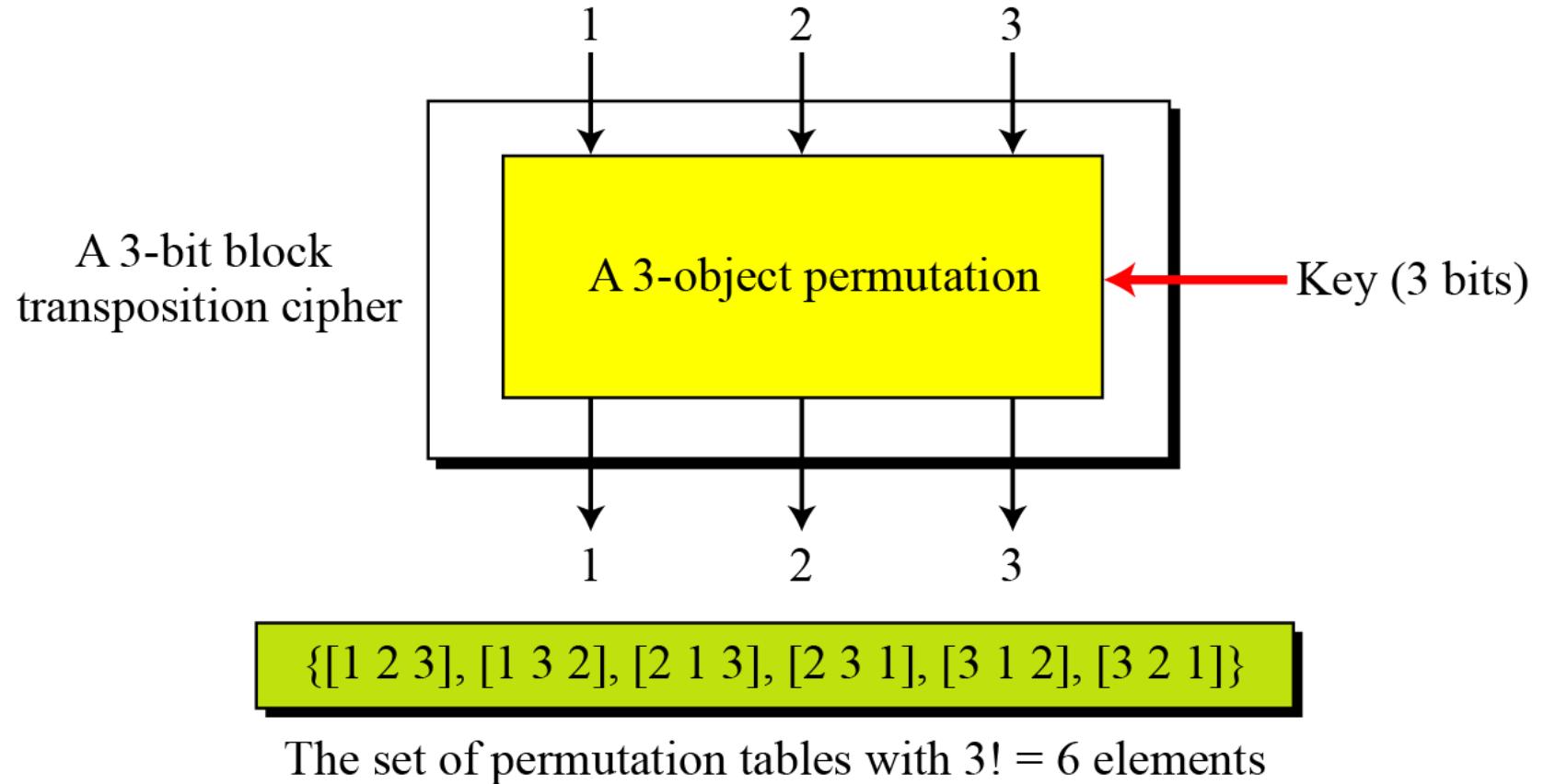
Example

Show the model and the set of permutation tables for a 3-bit block transposition cipher where the block size is 3 bits.

Solution

The set of permutation tables has $3! = 6$ elements

Block Ciphers as Permutation Groups...



Block Ciphers as Permutation Groups...

– Full-Size Key substitution Block Ciphers

- A full-size key substitution cipher does not transpose bits; it substitutes bits.
- We can model the substitution cipher as a permutation if we can decode the input and encode the output.

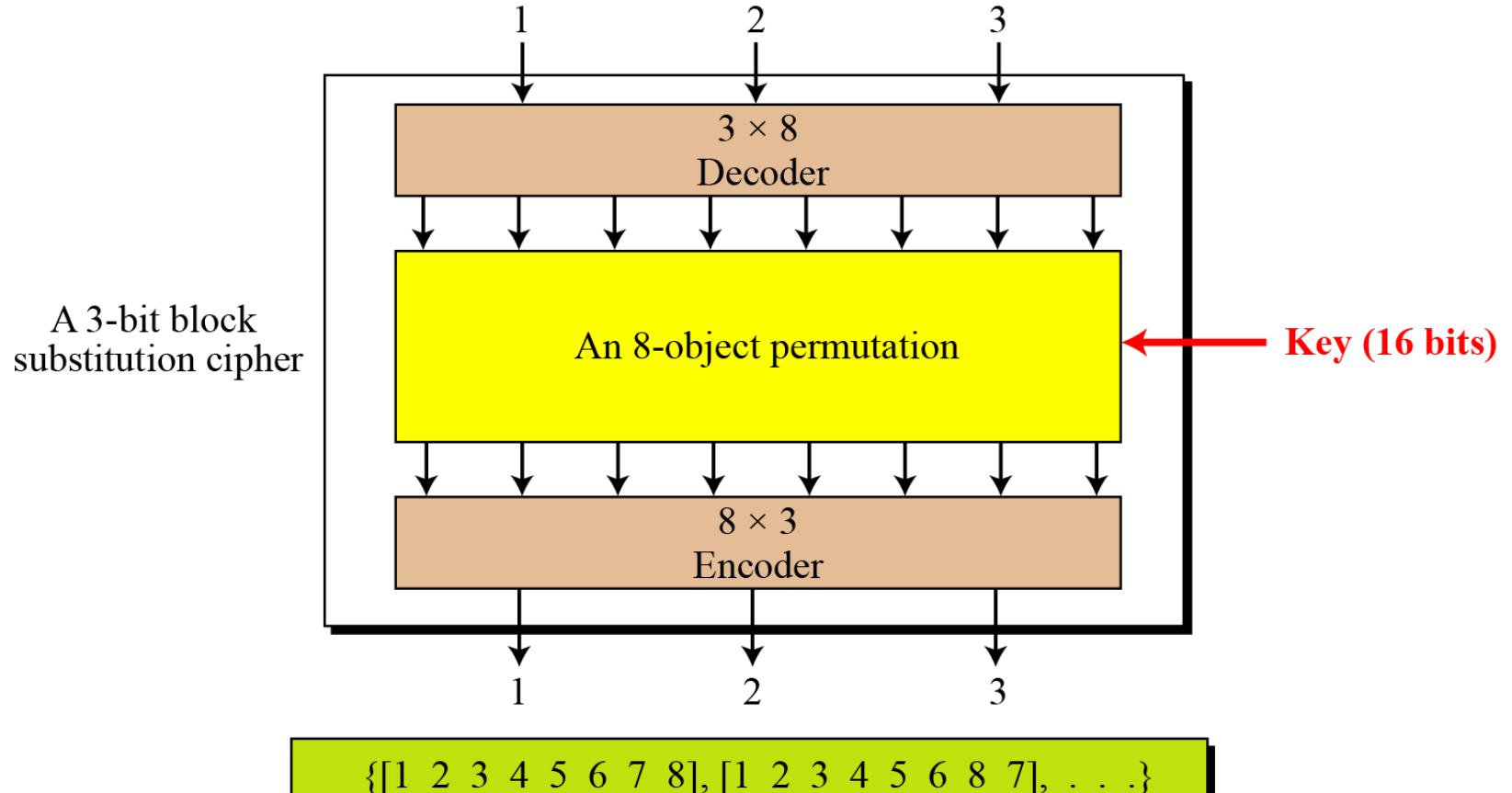
Example:

Show the model and the set of permutation tables for a 3-bit block substitution cipher.

Solution

Leads to a much longer key of $[\log_2 (40320)] = 16$ bits

Block Ciphers as Permutation Groups...



The set of permutation tables with $8! = 40,320$ elements

Block Ciphers as Permutation Groups...

- A full-size key n-bit transposition cipher or a substitution block cipher can be modeled as a permutation, but their key sizes are different:
 - Transposition: the key is ???
 - Substitution: the key is ???

Block Ciphers as Permutation Groups...

- A full-size key n-bit transposition cipher or a substitution block cipher can be modeled as a permutation, but their key sizes are different:
 - Transposition: the key is $\log_2(n!)$
 - Substitution: the key is $\log_2(2^n!)$

Block Ciphers as Permutation Groups...

- It is useless to have more than one stage of full size key ciphers, because the effect is the same as having a single stage
- Can you justify this ???

Block Ciphers as Permutation Groups...

- Partial size key ciphers
 - Actual ciphers can not use full-size keys because the size of the key becomes large
 - E.g. DES is a common substitution cipher with 64-bit block
 - If the designers of DES would have used full-size key cipher, what will be the size of a key?

Block Ciphers as Permutation Groups...

- Analysis of full-size key ciphers
 - Actual ciphers can not use full-size keys because the size of the key becomes large
 - E.g. DES is a common substitution cipher with 64-bit block
 - If the designers of DES would have used full-size key cipher, what will be the size of a key?
 - The answer is $\log_2(2^{64}!)$ $\approx 2^{70}$ bits !!!!
 - DES uses only 56-bit keys

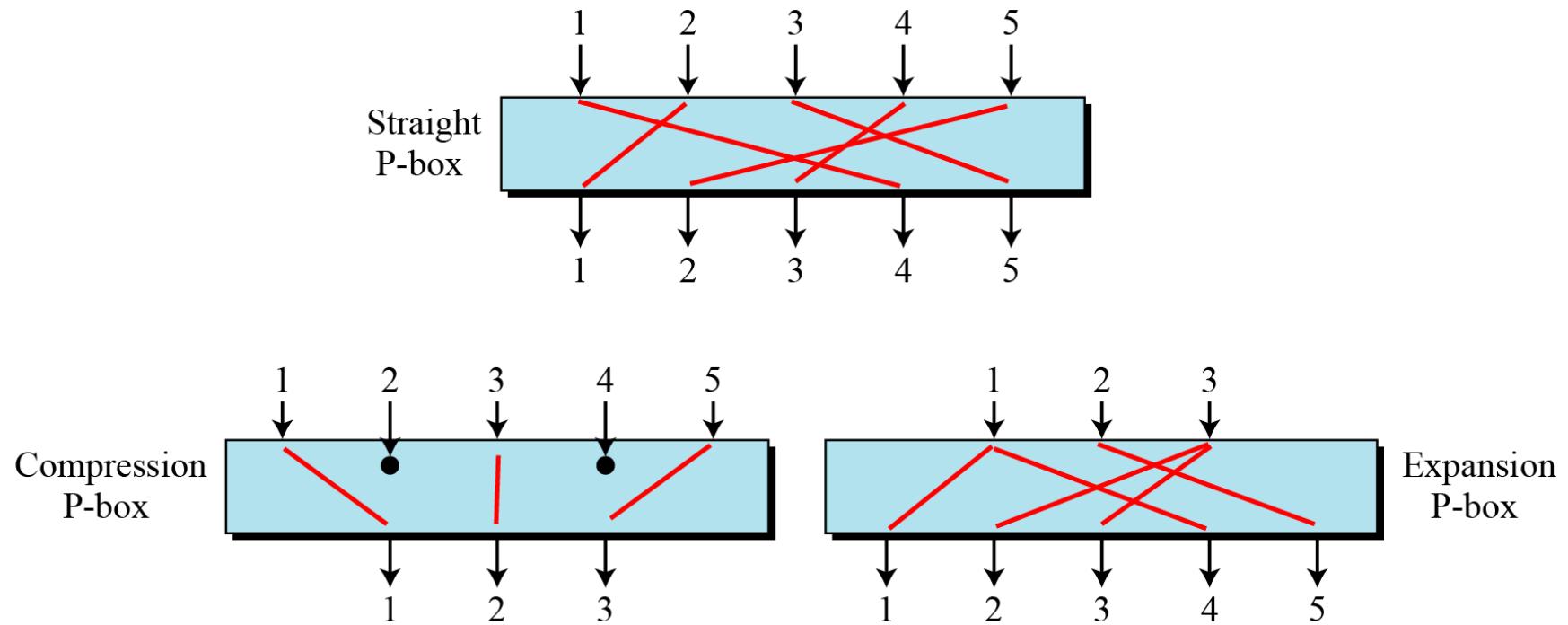
Block Ciphers as Permutation Groups...

- Partial-size key ciphers
 - Is a group under the composition operation if it is a subgroup of the corresponding full-size cipher
 - Can a multistage version of a partial-size key cipher be made to achieve more security?

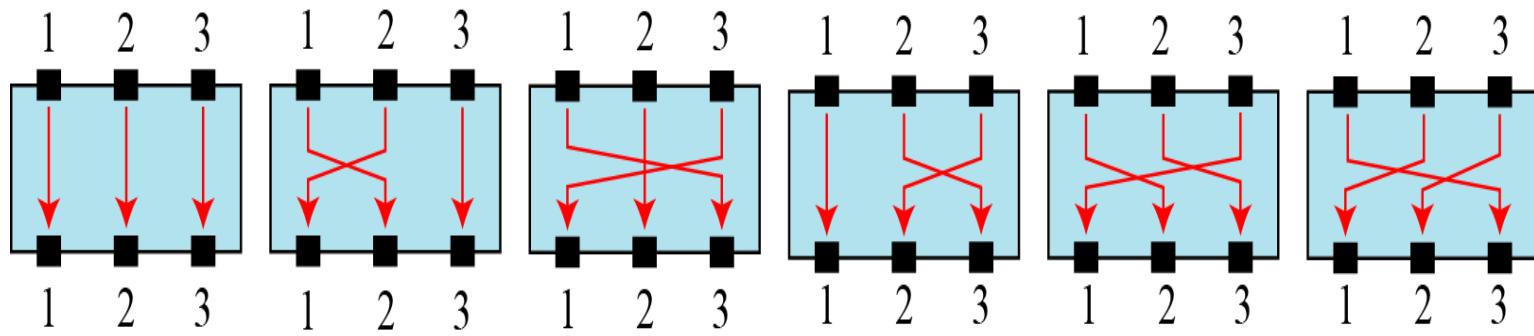
Components of a Modern Block Cipher

- Modern block ciphers normally are **keyed substitution ciphers** in which the key allows only **partial mappings** from the possible inputs to the possible outputs.
- P-Boxes
 - A P-box (permutation box) parallels the traditional transposition cipher for characters.
 - It transposes bits.

Components of a Modern Block Cipher...



- Possible mappings of a P-Box



- Straight P-Box

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

- Example
 - Design an 8×8 permutation table for a straight P-box that moves the two middle bits (bits 4 and 5) in the input word to the two ends (bits 1 and 8) in the output words. Relative positions of other bits should not be changed.

- Example
 - Design an 8×8 permutation table for a straight P-box that moves the two middle bits (bits 4 and 5) in the input word to the two ends (bits 1 and 8) in the output words. Relative positions of other bits should not be changed.
- Solution
 - We need a straight P-box with the table [4 1 2 3 6 7 8 5].

- Compression P-Boxes
 - A compression P-box is a P-box with n inputs and m outputs where m < n.
 - Example of a 32×24 permutation table

01	02	03	21	22	26	27	28	29	13	14	17
18	19	20	04	05	06	10	11	12	30	31	32

- Expansion P-Boxes
 - An expansion P-box is a P-box with n inputs and m outputs where m > n.
 - Example of a 12 X 16 P-Box

01	09	10	11	12	01	02	03	03	04	05	06	07	08	09	12
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

- P-Box invertibility
 - What can you say about this???
 - A straight P-box is invertible, but compression and expansion P-boxes are not.
- Inverting a permutation table represented as a one-dimensional table.

1. Original table

6	3	4	5	2	1
---	---	---	---	---	---

2. Add indices

6	3	4	5	2	1
1	2	3	4	5	6

3. Swap contents
and indices

1	2	3	4	5	6
6	3	4	5	2	1

4. Sort based
on indices

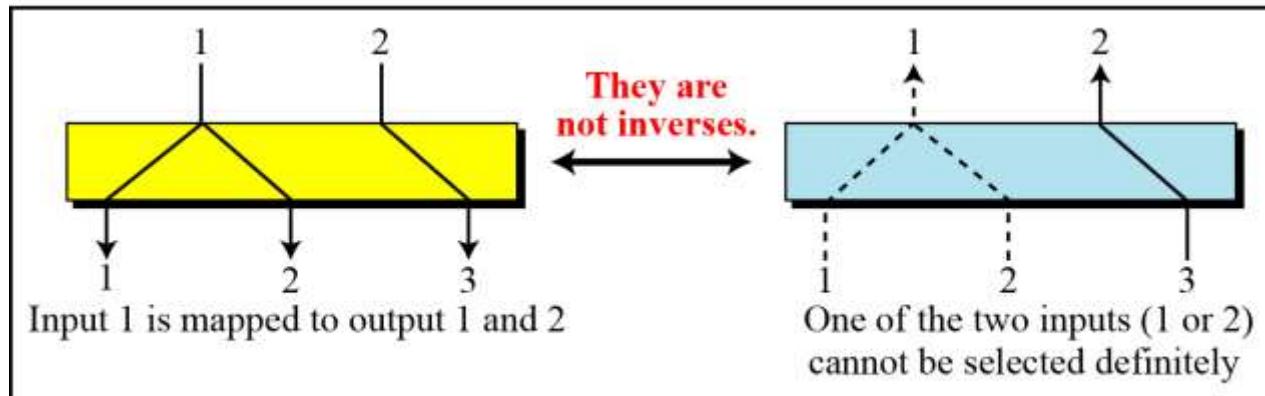
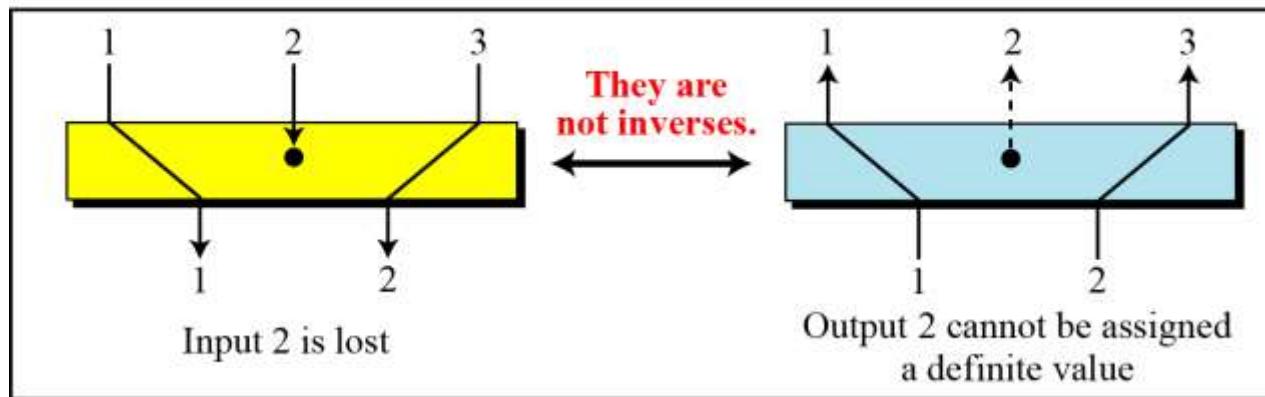
6	5	2	3	4	1
1	2	3	4	5	6

6	5	2	3	4	1
---	---	---	---	---	---

5. Inverted table

- Compression and expansion P-boxes are non-invertible

Compression P-box



Expansion P-box

- **S-Box**
 - An S-box (substitution box) can be thought of as a miniature substitution cipher.
 - An S-box is an $m \times n$ substitution unit, where m and n are not necessarily the same.
 - Example
 - In an S-box with three inputs and two outputs, we have,

$$y_1 = x_1 \oplus x_2 \oplus x_3 \quad y_2 = x_1$$

- S-Box
 - The S-box is linear because $a_{1,1} = a_{1,2} = a_{1,3} = a_{2,1} = 1$ and $a_{2,2} = a_{2,3} = 0$. The relationship can be represented by matrices, as shown below:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

- S-Box
 - In an S-box with three inputs and two outputs, we have

$$y_1 = (x_1)^3 + x_2 \quad y_2 = (x_1)^2 + x_1 x_2 + x_3$$

- where multiplication and addition is in GF(2).
The S-box is nonlinear because there is no linear relationship between the inputs and the outputs.

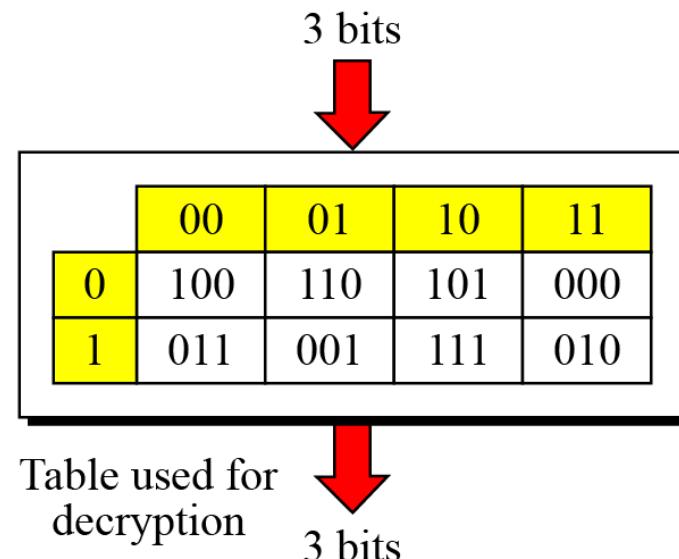
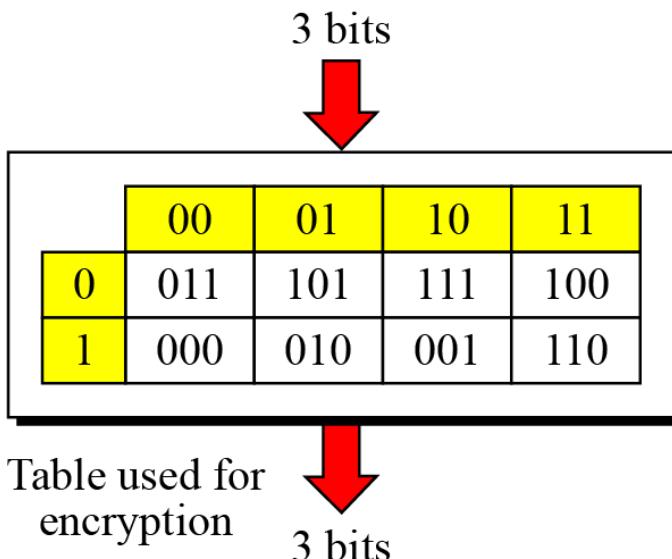
- Example

- The following table defines the input/output relationship for an S-box of size 3×2 . The leftmost bit of the input defines the row; the two rightmost bits of the input define the column. The two output bits are values on the cross section of the selected row and column.

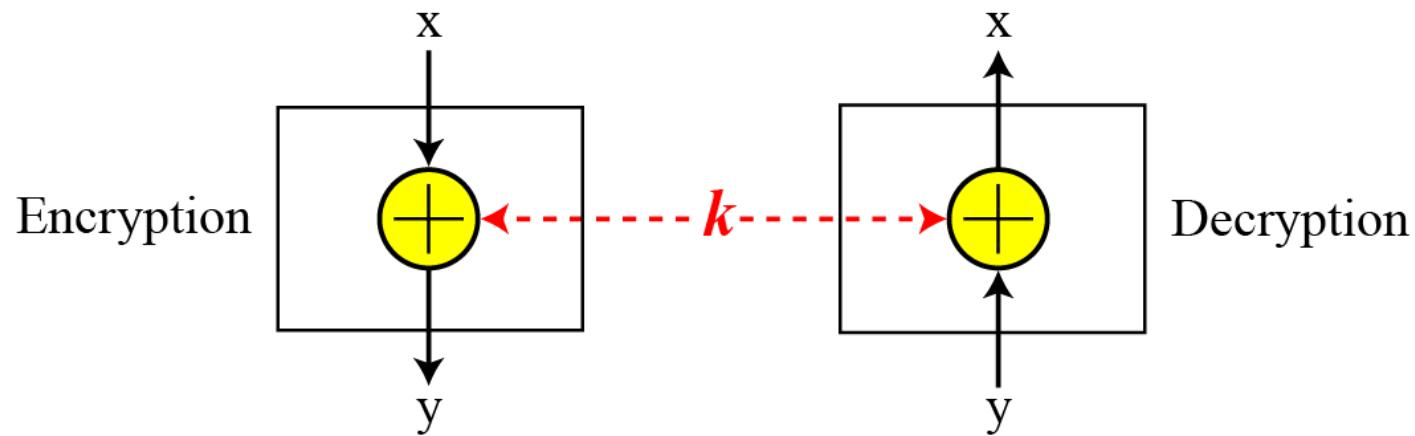
		Leftmost bit					
		00	01	10	11	Rightmost bits	
0	00	10	01	11			
	10	00	11	01			
		Output bits					

- Based on the table, an input of 010 yields the output 01. An input of 101 yields the output of 00.

- S-Box Invertibility
 - An S-box may or may not be invertible.
 - In an invertible S-box, the number of input bits should be the same as the number of output bits.



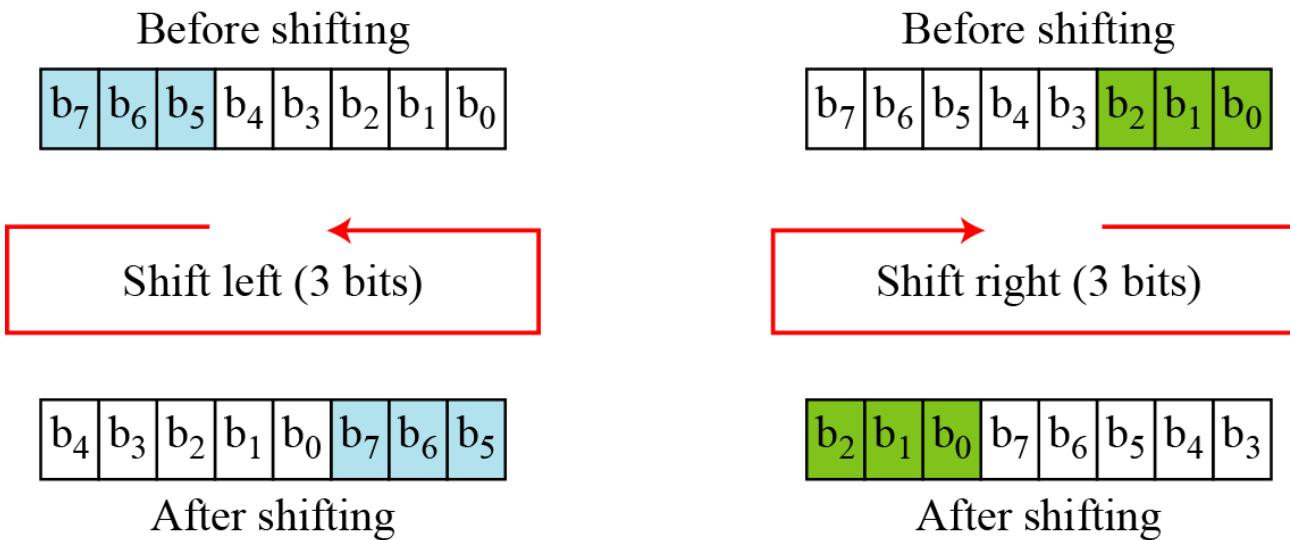
- Exclusive-Or



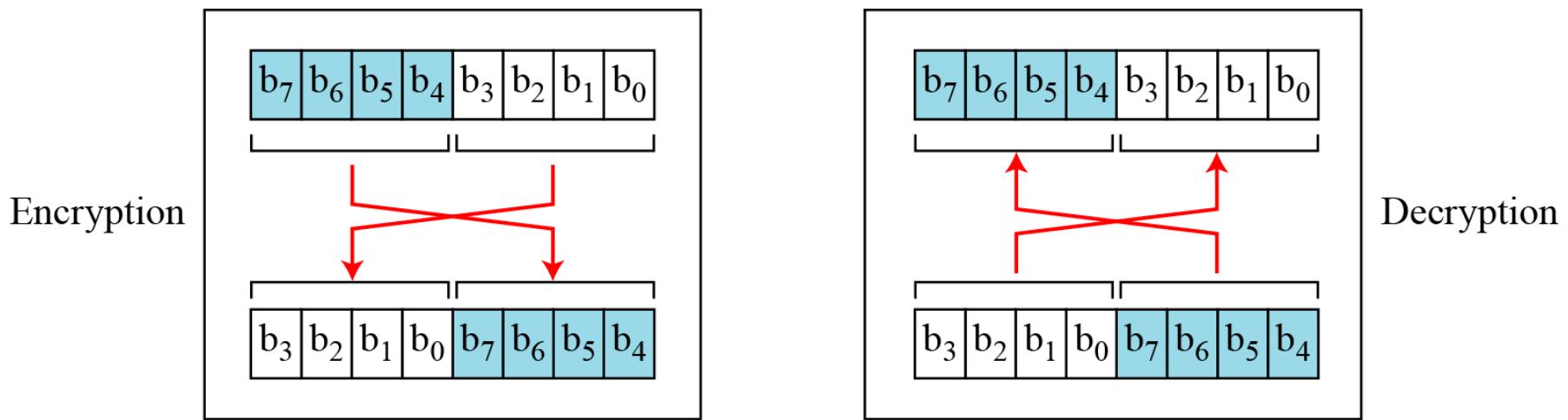
- Exclusive-Or...
 - An important component in most block ciphers is the exclusive-or operation.
 - Addition and subtraction operations in the $GF(2^n)$ field are performed by a single operation called the exclusive-or (XOR).
 - The five properties of the exclusive-or operation in the $GF(2^n)$ field makes this operation a very interesting component for use in a block cipher: closure, associativity, commutativity, existence of identity, and existence of inverse.

- Exclusive-Or...
 - The inverse of a component in a cipher makes sense if the component represents a unary operation (one input and one output).
 - For example, a keyless P-box or a keyless S-box can be made invertible because they have one input and one output.
 - An exclusive operation is a binary operation. The inverse of an exclusive-or operation can make sense only if one of the inputs is fixed (is the same in encryption and decryption).
 - For example, if one of the inputs is the key, which normally is the same in encryption and decryption, then an exclusive-or operation is self-invertible

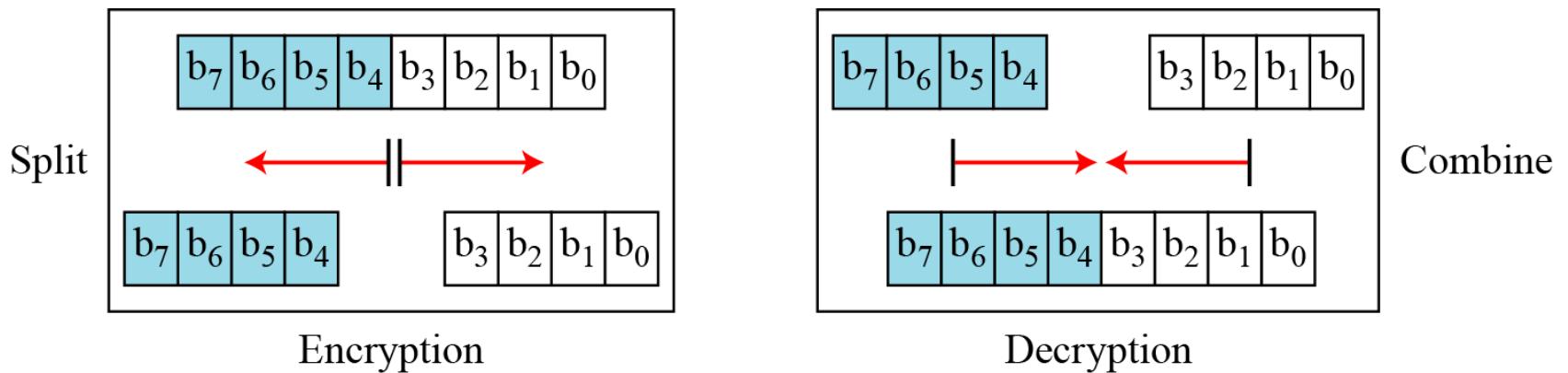
- Circular Shift operation



- Swap
 - The swap operation is a special case of the circular shift operation where $k = n/2$.



- Split and Combine
 - Two other operations found in some block ciphers are split and combine.



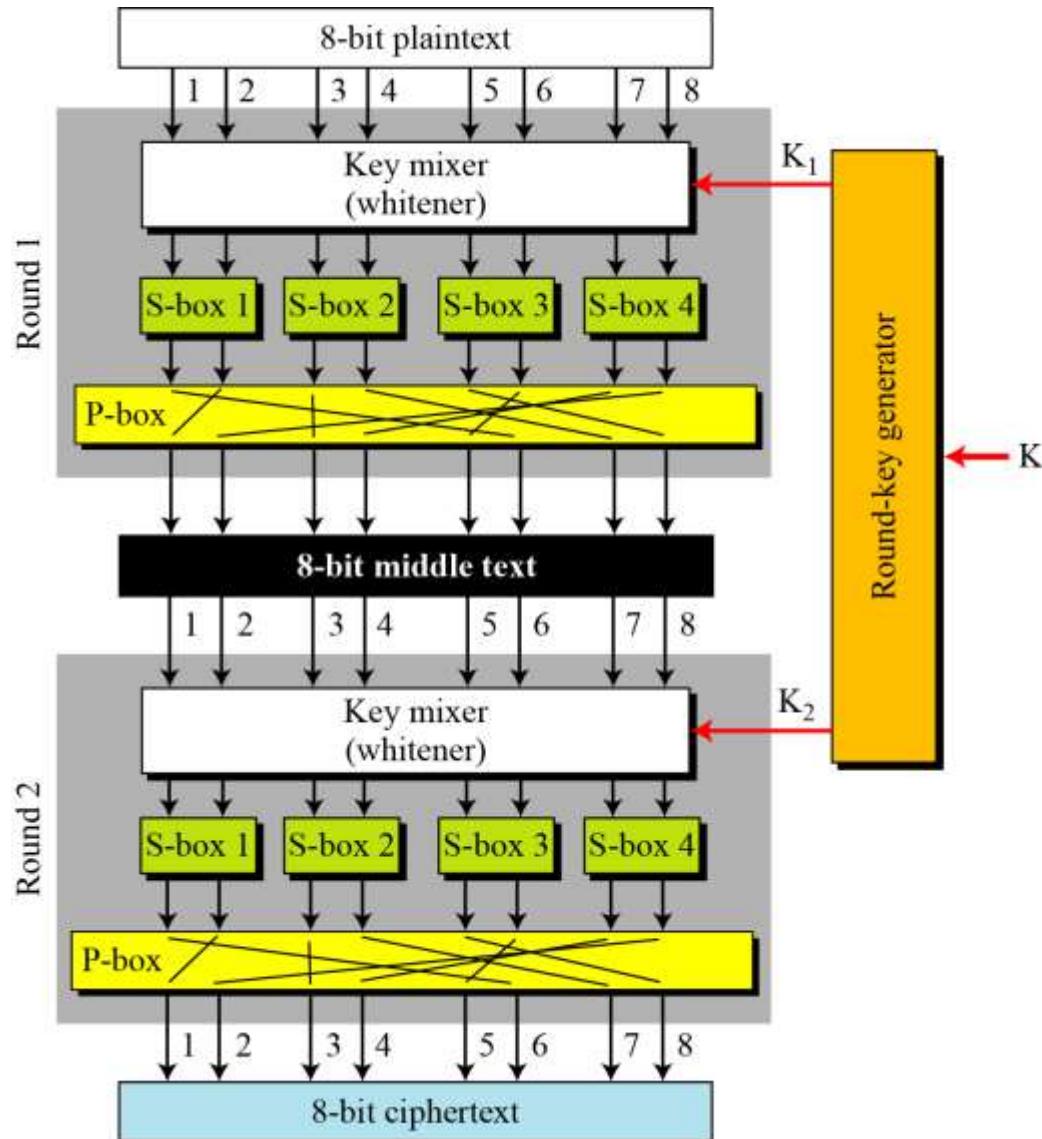
Product cipher

- Shannon introduced the concept of a product cipher.
- A product cipher is a complex cipher combining substitution, permutation, and other components

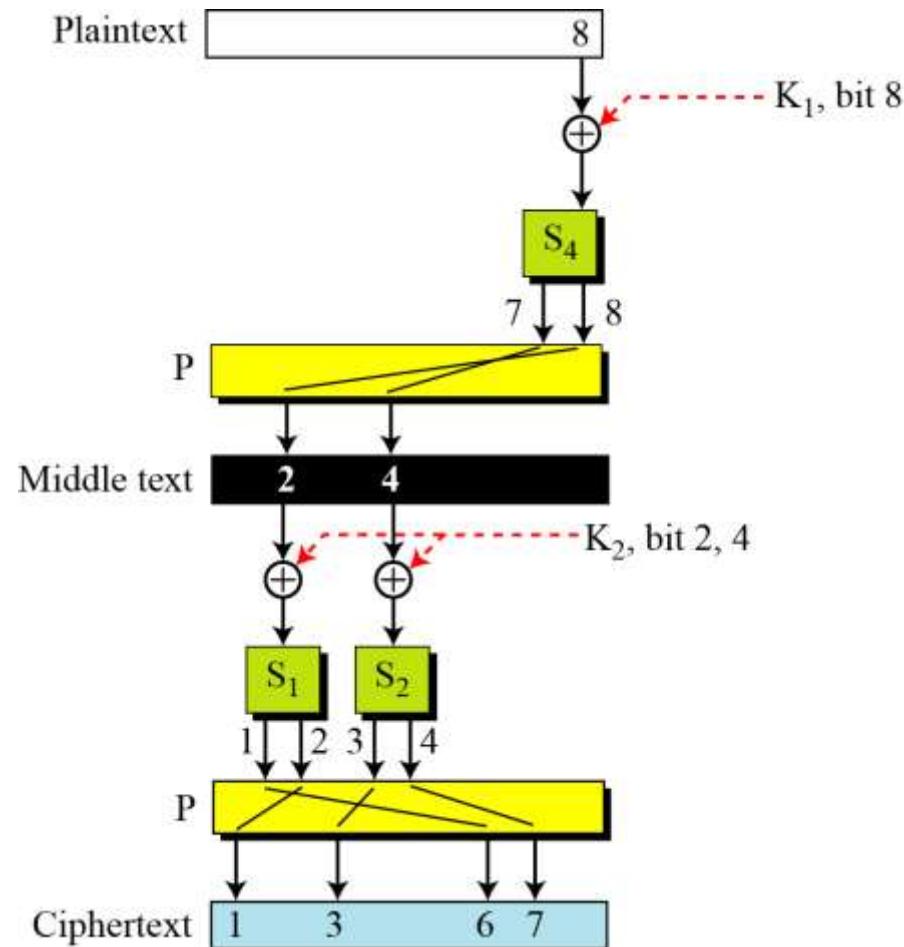
Product cipher...

- Diffusion
 - The idea of diffusion is to hide the relationship between the ciphertext and the plaintext.
- Confusion
 - The idea of confusion is to hide the relationship between the ciphertext and the key.
- Diffusion and confusion can be achieved using iterated product ciphers where each iteration is a combination of S-boxes, P-boxes, and other components.

Product cipher...



Product cipher...



Two classes of product ciphers

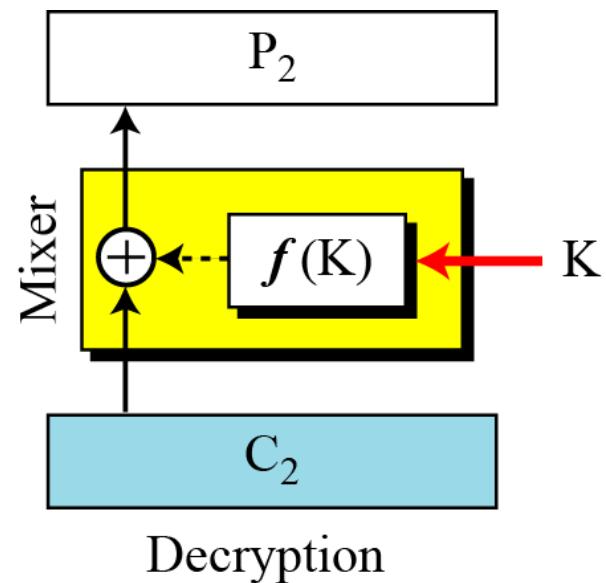
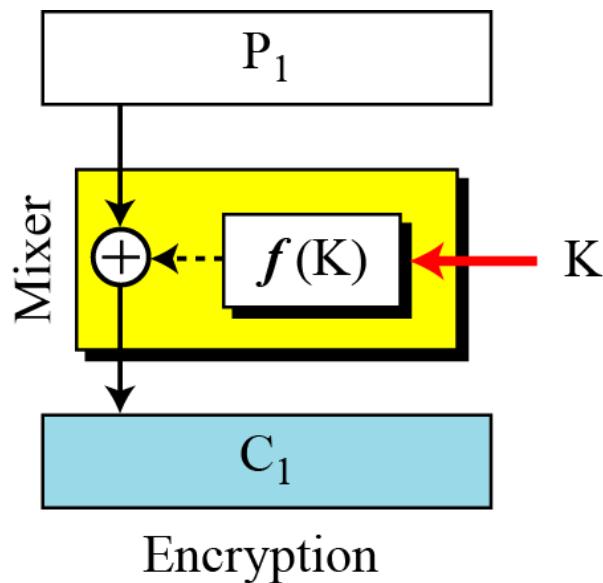
- Modern block ciphers are all product ciphers, but they are divided into two classes.
 - Feistel ciphers
 - Non-Feistel ciphers

Two classes of product ciphers...

- Feistel Ciphers
 - Feistel designed a very intelligent and interesting cipher that has been used for decades.
 - A Feistel cipher can have three types of components: self-invertible, invertible, and noninvertible.

Two classes of product ciphers...

- The first thought in Feistel Cipher design



Two classes of product ciphers...

- Example
 - The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern. Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.

Two classes of product ciphers...

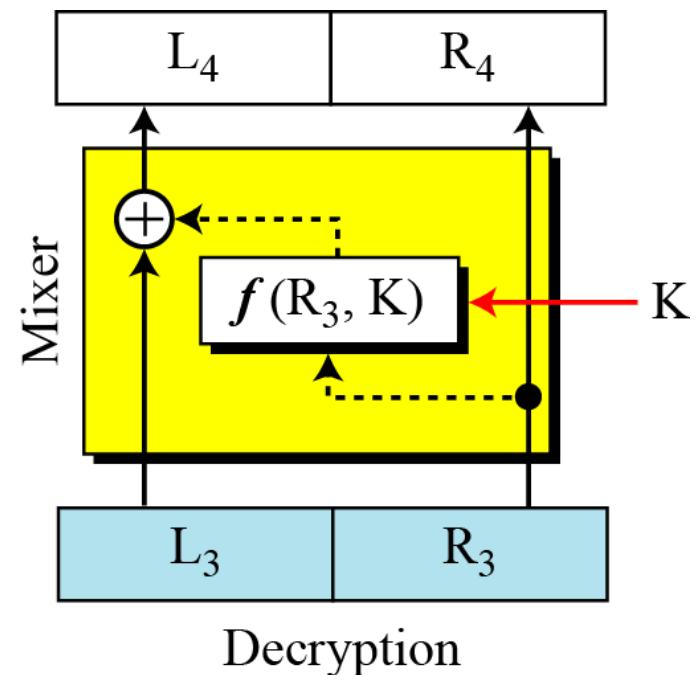
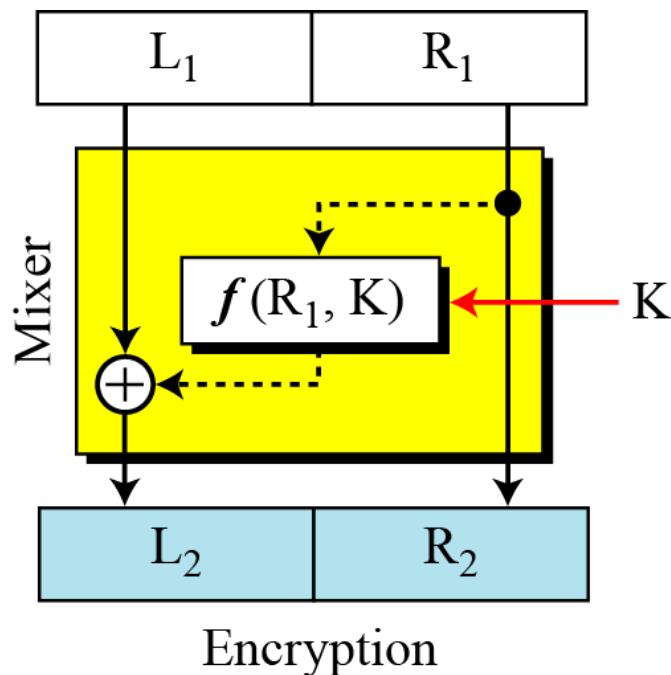
- Solution
 - The function extracts the first and second bits to get 11 in binary or 3 in decimal. The result of squaring is 9, which is 1001 in binary.

Encryption: $C = P \oplus f(K) = 0111 \oplus 1001 = 1110$

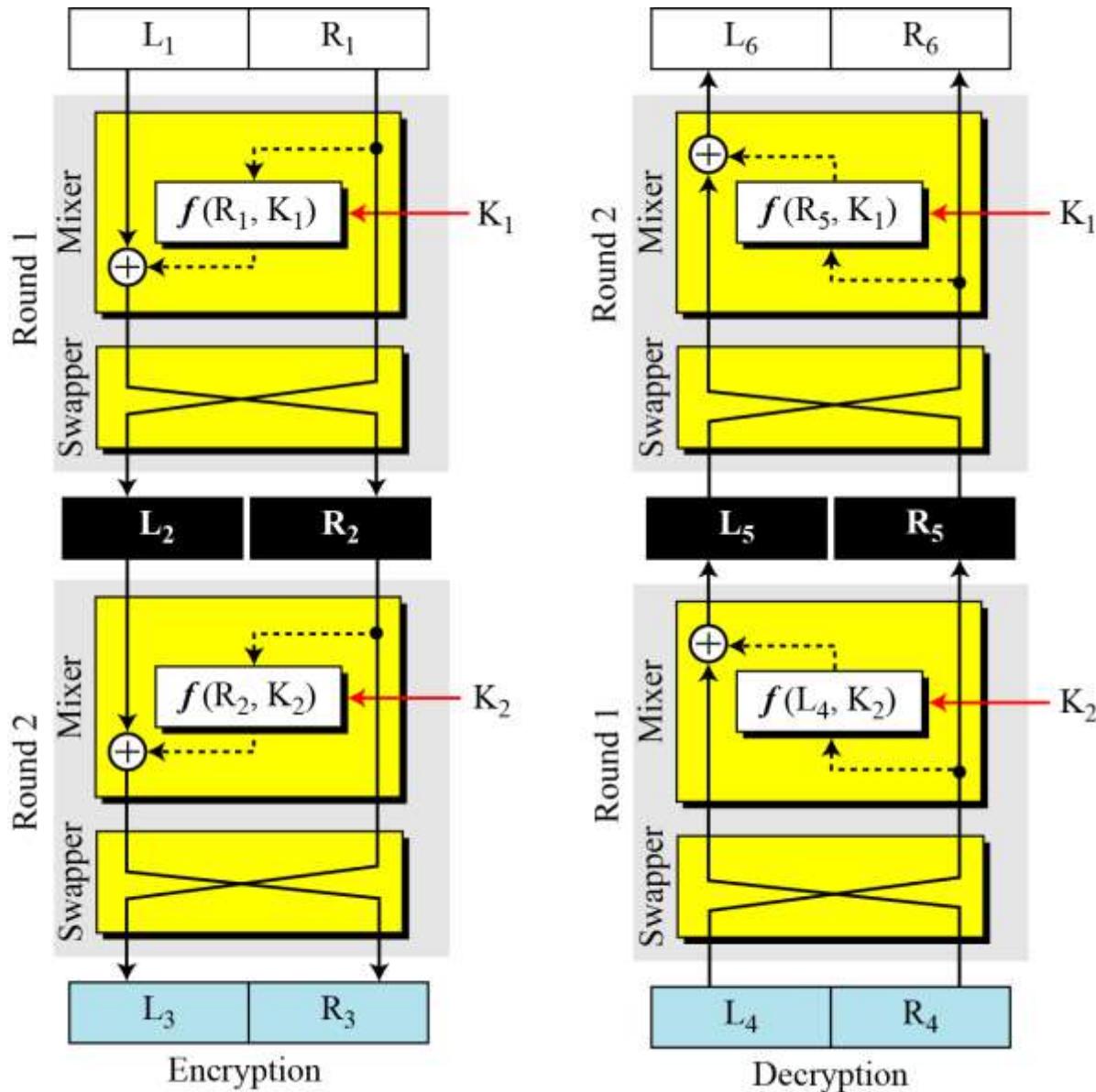
Decryption: $P = C \oplus f(K) = 1110 \oplus 1001 = 0111$

Two classes of product ciphers...

- Improvement in previous design



Final design of a fiestel cipher...



Final design of a fiestel cipher...

- Non-fiestel cipher
 - Use only invertible components
 - A component in the encryption cipher has the corresponding component in the decryption cipher.

Attacks on modern ciphers

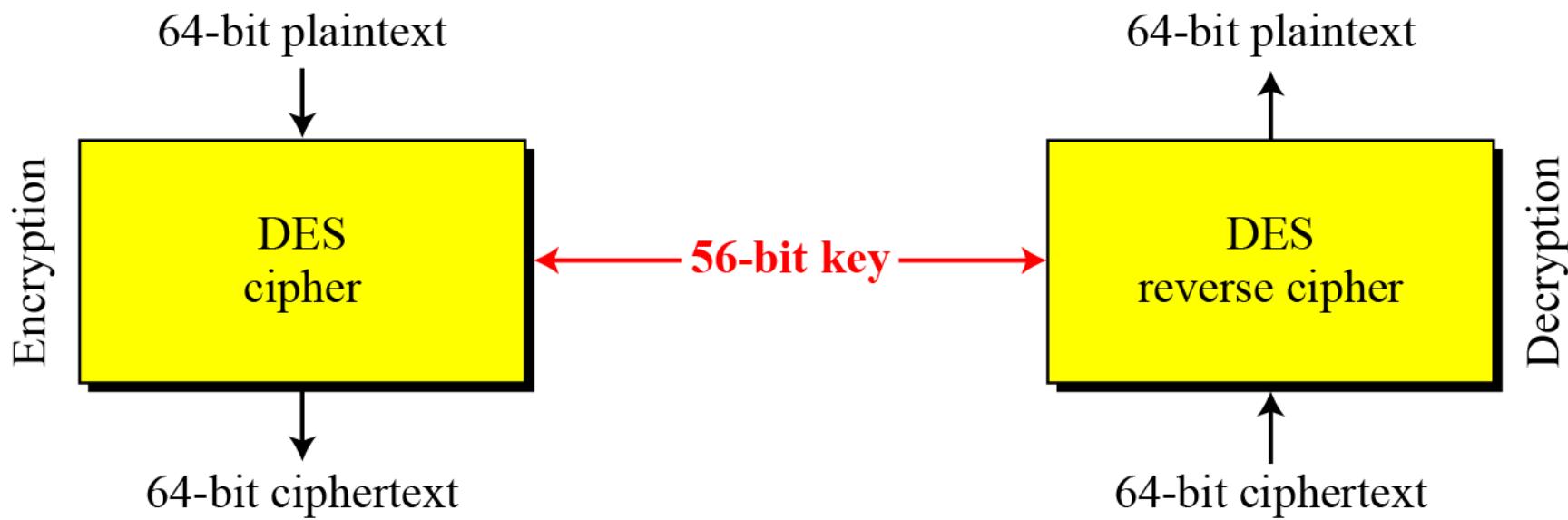
- Attacks on traditional ciphers can also be used on modern block ciphers, but today's block ciphers resist most of the attacks that are possible in classical ciphers

Data Encryption Standard (DES)

Introduction

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

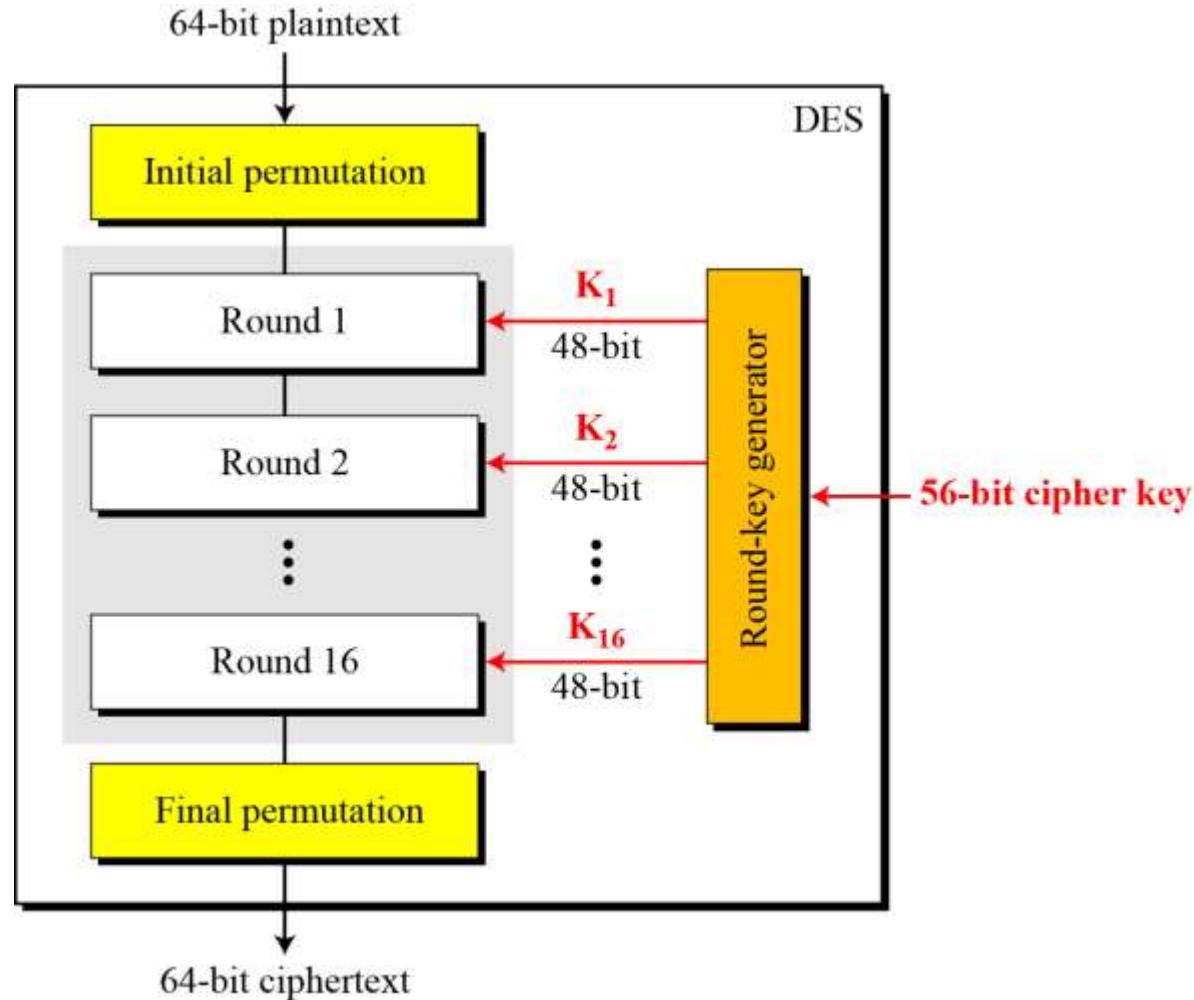
Overview



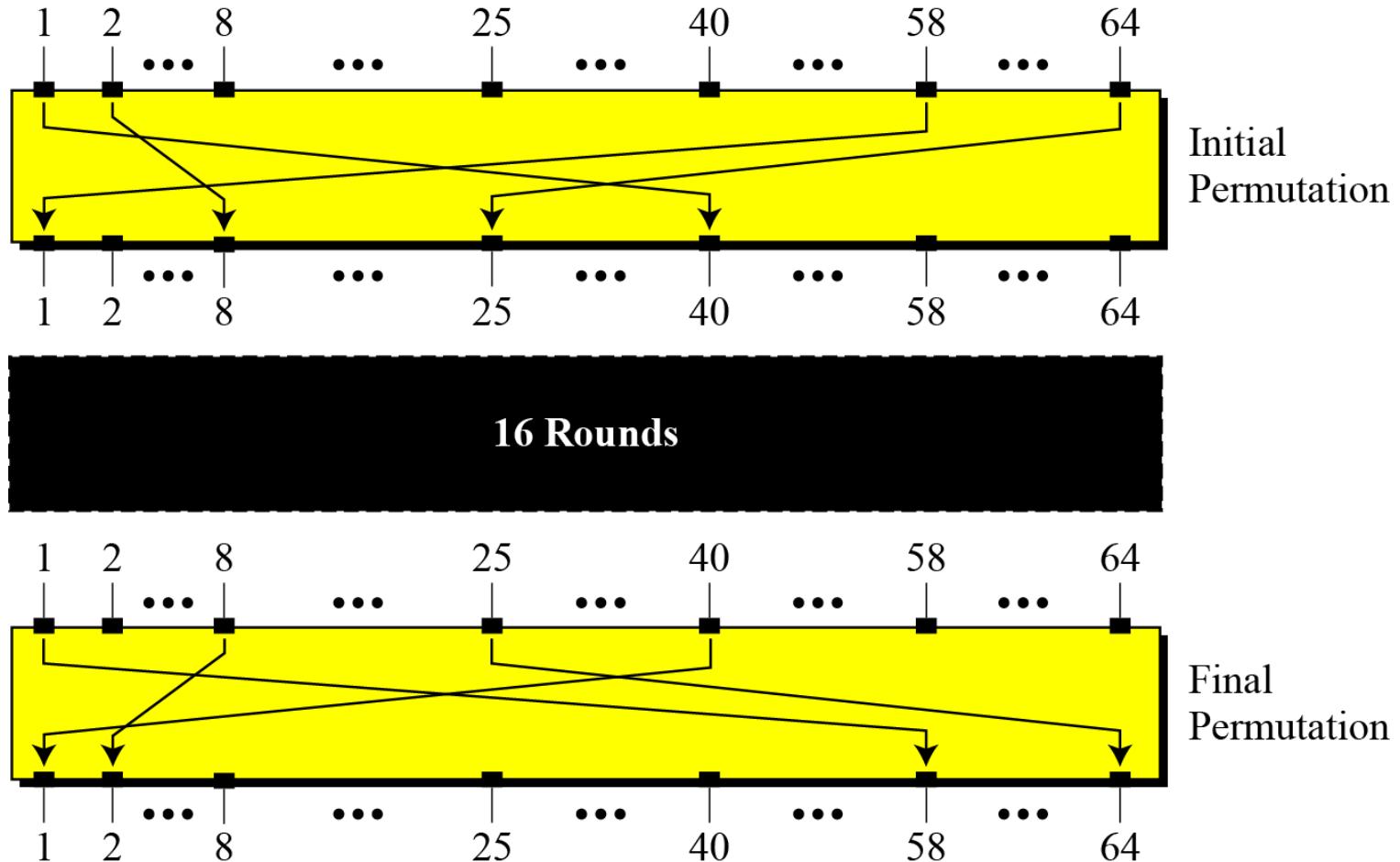
Structure of DES

- The encryption process is made of two permutations (P-boxes), which we call initial and final permutations, and sixteen Feistel rounds.

Structure of DES...



Initial and Final permutations



Initial and Final permutations...

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

Initial and Final permutations...

- Find the output of the initial permutation box when the input is given in hexadecimal as:

0x0000 0080 0000 0002

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

Initial and Final permutations...

- Find the output of the initial permutation box when the input is given in hexadecimal as:

```
0x0000 0080 0000 0002
```

- Solution:

Assignment questions...

- Prove that the initial and final permutations are the inverse of each other by finding the output of the final permutation if the input is

0x0002 0000 0000 0001

Assignment questions...

- Prove that the initial and final permutations are the inverse of each other by finding the output of the final permutation if the input is

```
0x0002 0000 0000 0001
```

Assignment questions...

- The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern. Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.

$$C = P \text{ XOR } f(K) \quad \text{and} \quad P = C \text{ XOR } f(K)$$

Assignment questions...

- The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern. Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.

$$C = P \text{ XOR } f(K) \quad \text{and} \quad P = C \text{ XOR } f(K)$$

- Solution
 - The function extracts the first and second bits to get 11 in binary or 3 in decimal. The result of squaring is 9 which is

Encryption: $C = P \oplus f(K) = 0111 \oplus 1001 = 1110$

Decryption: $P = C \oplus f(K) = 1110 \oplus 1001 = 0111$

Assignment questions...

- A transposition block has 10 inputs and 10 outputs. What is the order of the permutation group? What is the key size?
- A substitution block has 10 inputs and 10 outputs. What is the order of the permutation group? What is the key size?
- The input/output relation in a 2x2 S-box is shown by the following table. Show the table for the inverse S-box.

Input left bit/input right bit	0	1
0	01	11
1	10	00

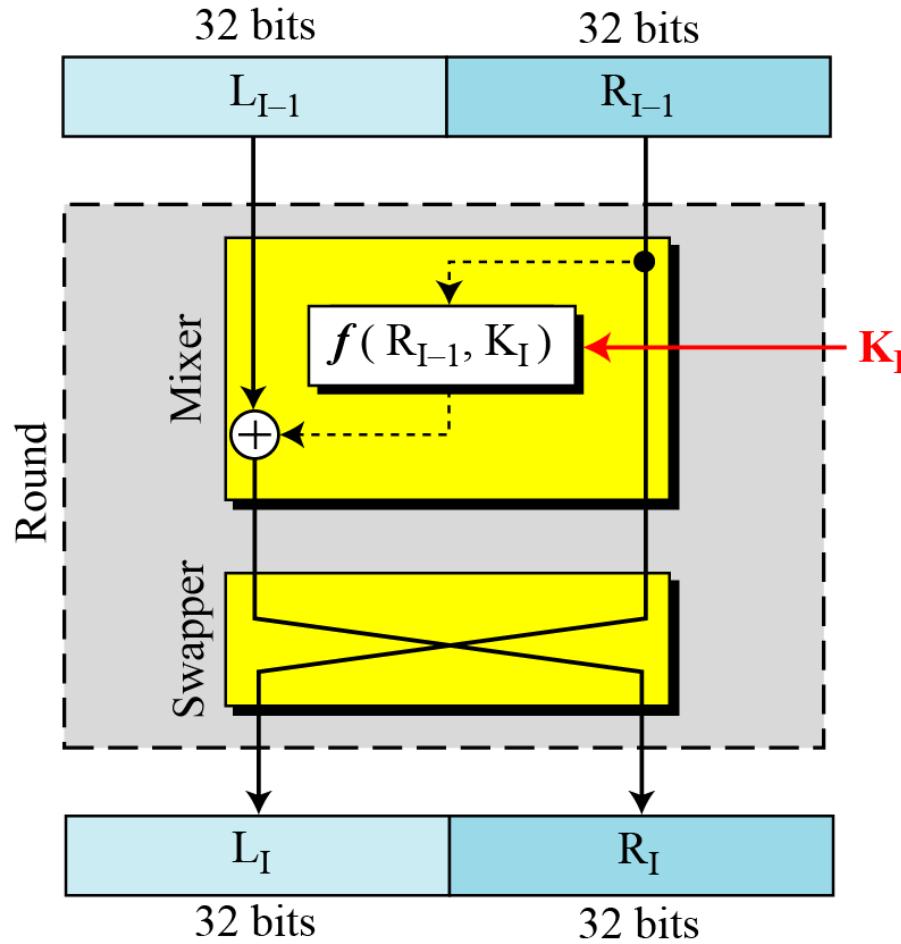
Assignment questions...

- Show the D-box defined by the following table:

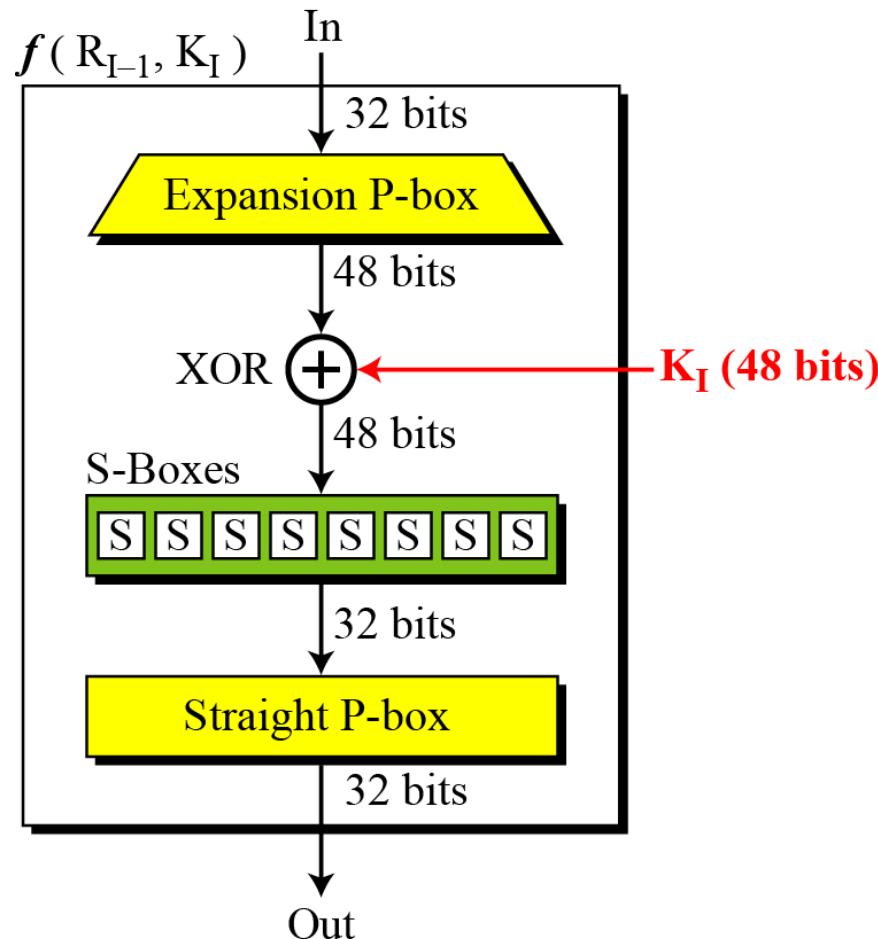
8	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Rounds

- 16 rounds of Fiestel cipher

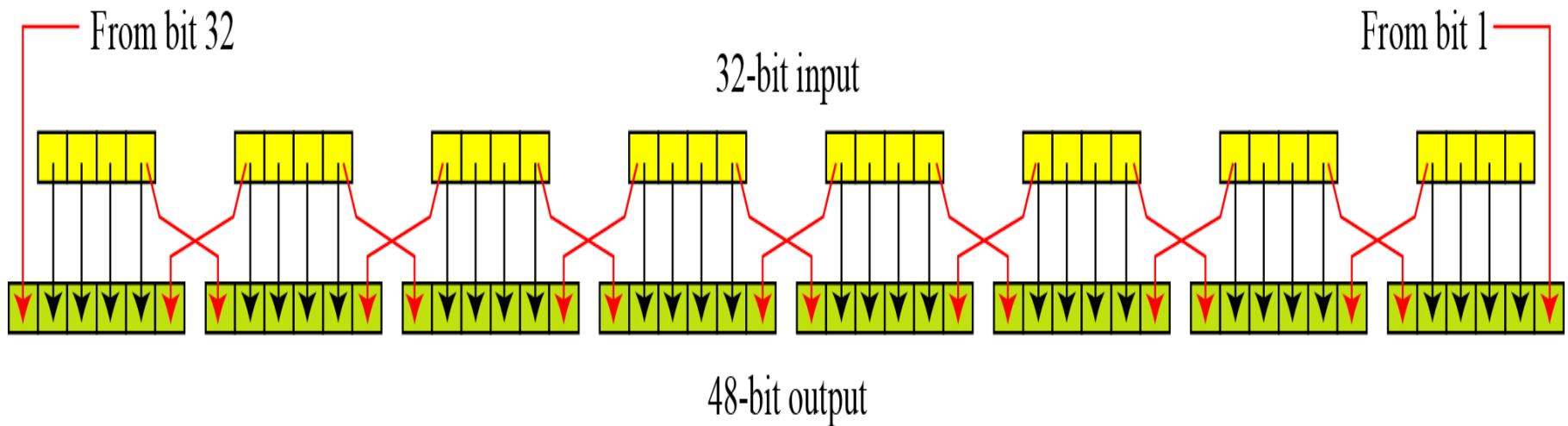


DES function



DES function...

- Expansion P-box
 - Since R_{I-1} is a 32-bit input and K_I is a 48-bit key, we first need to expand R_{I-1} to 48 bits.



DES function...

- Expansion P-box
 - Since R_{l-1} is a 32-bit input and K_l is a 48-bit key, we first need to expand R_{l-1} to 48 bits.

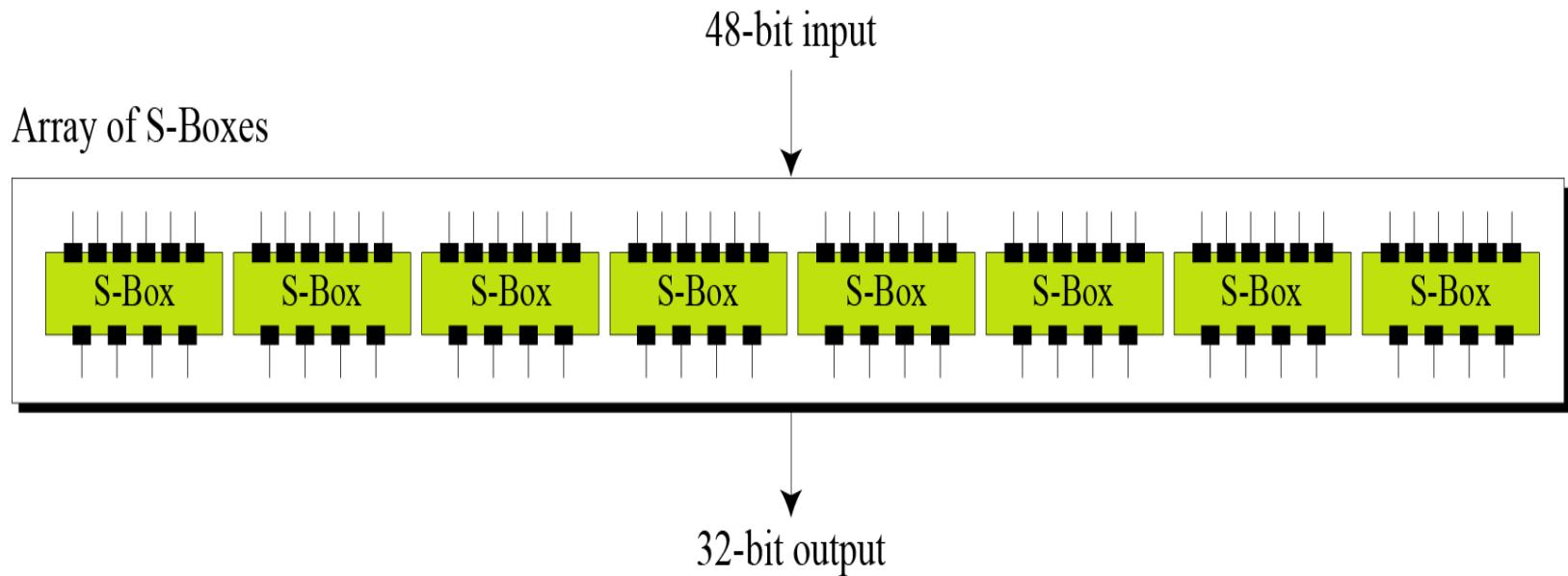
32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

DES function...

- Whitener
 - After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key. Note that both the right section and the key are 48-bits in length. Also note that the round key is used only in this operation.

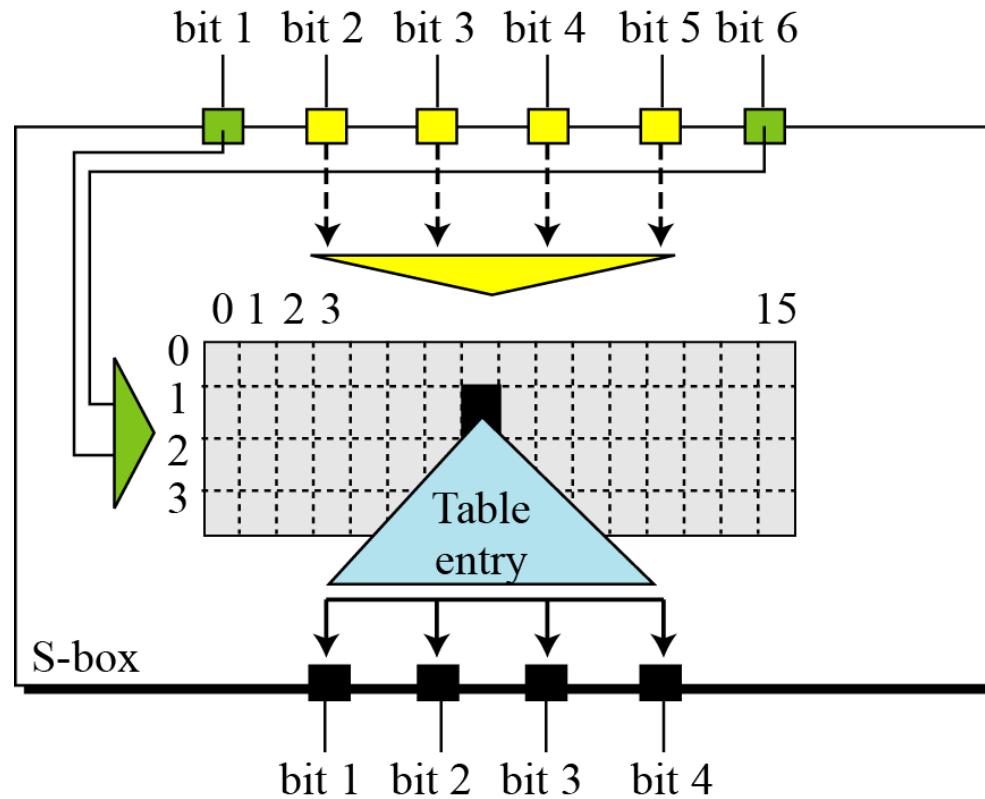
DES function...

- S-Boxes



DES function...

- S-Boxes



DES function...

- Permutations for S-Box-I

	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 function...

- Example
 - The input to S-box 1 is 100011. What is the output?

DES function...

- Example
 - The input to S-box 1 is 100011. What is the output?
- Solution
 - If we write the first and the sixth bits together, we get 11 in binary, which is 3 in decimal. The remaining bits are 0001 in binary, which is 1 in decimal. We look for the value in row 3, column 1, in Table of S-box 1. The result is 12 in decimal, which in binary is 1100. So the input 100011 yields the output 1100.

DES function...

- Straight P-Box

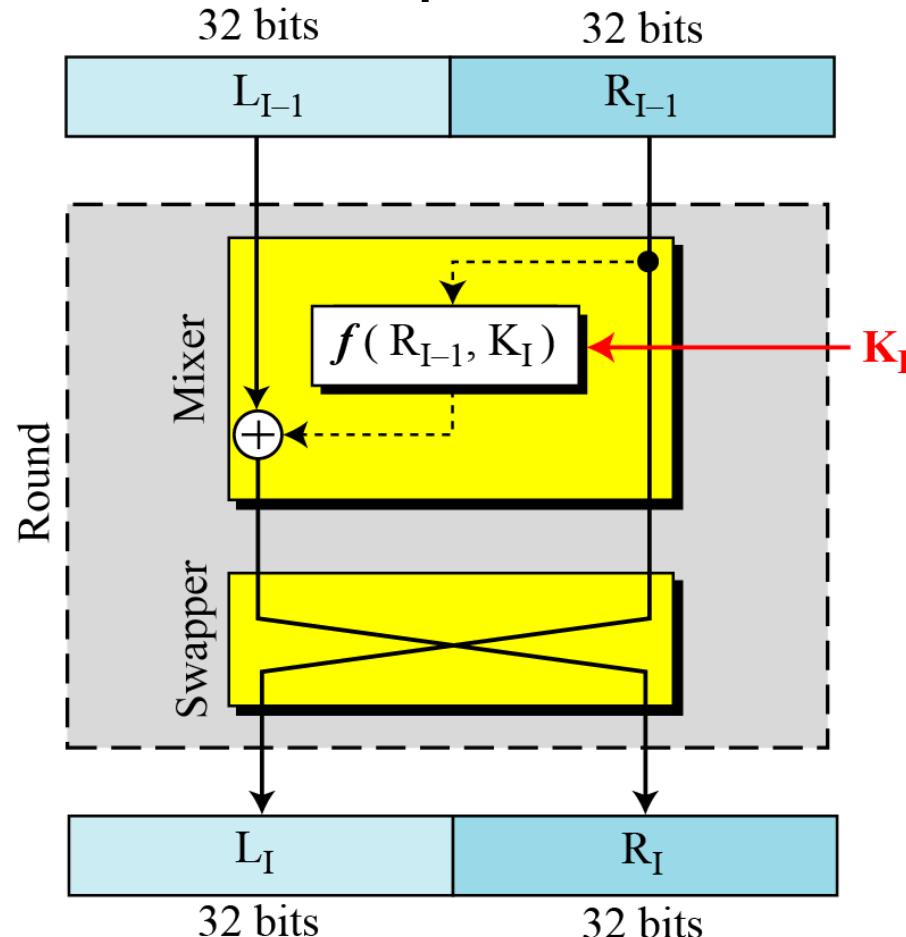
16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Cipher and reverse cipher

- Using mixers and swappers, we can create the cipher and reverse cipher, each having 16 rounds.
- First Approach
 - To achieve this goal, one approach is to make the last round (round 16) different from the others; it has only a mixer and no swapper.

Cipher and reverse cipher...

- 16 rounds of Fiestel cipher



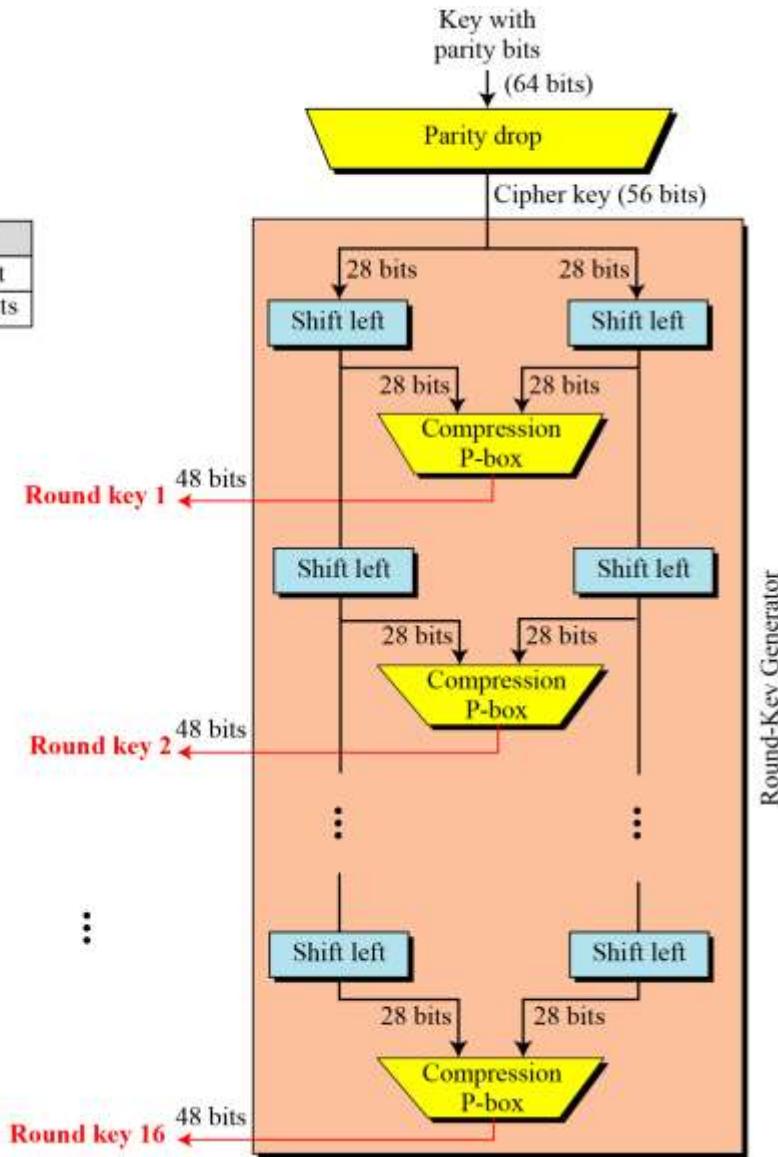
Cipher and reverse cipher...

- Alternative approach
 - We can make all 16 rounds the same by including one swapper to the 16th round and add an extra swapper after that.

Key generation

- The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key.

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



Key generation

Parity bit drop table

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

Number of shift bits

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

Key generation

Key compression table

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

DES: an example of encipherment

Plaintext: 123456ABCD132536

Key: AABB09182736CCDD

CipherText: C0B7A8D05F3A829C

Plaintext: 123456ABCD132536			
<i>After initial permutation:</i> 14A7D67818CA18AD After splitting: L ₀ =14A7D678 R ₀ =18CA18AD			
Round	Left	Right	Round Key
Round 1	18CA18AD	5A78E394	194CD072DE8C
Round 2	5A78E394	4A1210F6	4568581ABCCE
Round 3	4A1210F6	B8089591	06EDA4ACF5B5
Round 4	B8089591	236779C2	DA2D032B6EE3

DES: an example of encipherment...

<i>Round 5</i>	236779C2	A15A4B87	69A629FEC913
<i>Round 6</i>	A15A4B87	2E8F9C65	C1948E87475E
<i>Round 7</i>	2E8F9C65	A9FC20A3	708AD2DDB3C0
<i>Round 8</i>	A9FC20A3	308BEE97	34F822F0C66D
<i>Round 9</i>	308BEE97	10AF9D37	84BB4473DCCC
<i>Round 10</i>	10AF9D37	6CA6CB20	02765708B5BF
<i>Round 11</i>	6CA6CB20	FF3C485F	6D5560AF7CA5
<i>Round 12</i>	FF3C485F	22A5963B	C2C1E96A4BF3
<i>Round 13</i>	22A5963B	387CCDAA	99C31397C91F
<i>Round 14</i>	387CCDAA	BD2DD2AB	251B8BC717D0
<i>Round 15</i>	BD2DD2AB	CF26B472	3330C5D9A36D
<i>Round 16</i>	19BA9212	CF26B472	181C5D75C66D
<i>After combination:</i> 19BA9212CF26B472			
<i>Ciphertext:</i> C0B7A8D05F3A829C		<i>(after final permutation)</i>	

DES: an example of decipherment

Ciphertext: C0B7A8D05F3A829C			
After initial permutation: 19BA9212CF26B472			
After splitting: $L_0=19BA9212$ $R_0=CF26B472$			
Round	Left	Right	Round Key
Round 1	CF26B472	BD2DD2AB	181C5D75C66D
Round 2	BD2DD2AB	387CCDAA	3330C5D9A36D
...
Round 15	5A78E394	18CA18AD	4568581ABCCE
Round 16	14A7D678	18CA18AD	194CD072DE8C
After combination: 14A7D67818CA18AD			
Plaintext: 123456ABCD132536		(after final permutation)	

DES: Analysis

- Two desirable properties
 - Avalanche effect
 - Completeness

Avalanche effect

- To check the avalanche effect in DES, let us encrypt two plaintext blocks (with the same key) that differ only in one bit and observe the differences in the number of bits in each round.

Plaintext: 0000000000000000

Key: 22234512987ABB23

Ciphertext: 4789FD476E82A5F1

Plaintext: 0000000000000001

Key: 22234512987ABB23

Ciphertext: 0A4ED5C15A63FEA3

Avalanche effect...

- Ciphertext blocks differ in 29 bits.
 - i.e. changing approximately 1.5 percent of the plaintext creates a change of approximately 45 percent in the ciphertext.

Rounds	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bit differences	1	6	20	29	30	33	32	29	32	39	33	28	30	31	30	29

Completeness

- Completeness effect means that each bit of the ciphertext needs to depend on many bits on the plaintext.
 - S-Box
 - P-Box
 - 16 rounds of fiestel blocks

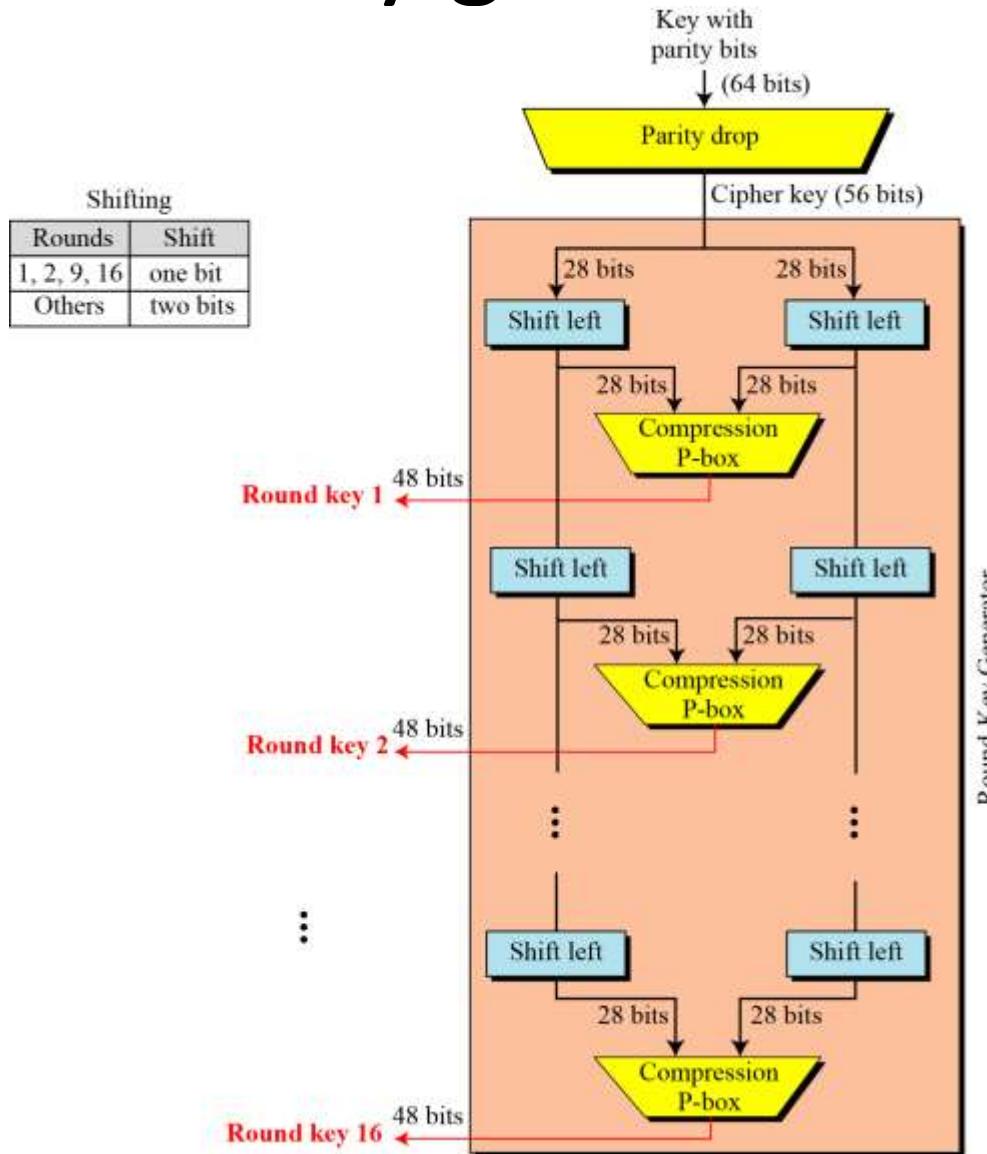
DES Weaknesses

- Weakness in 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

Key generation



DES Weaknesses...

- Example
 - 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

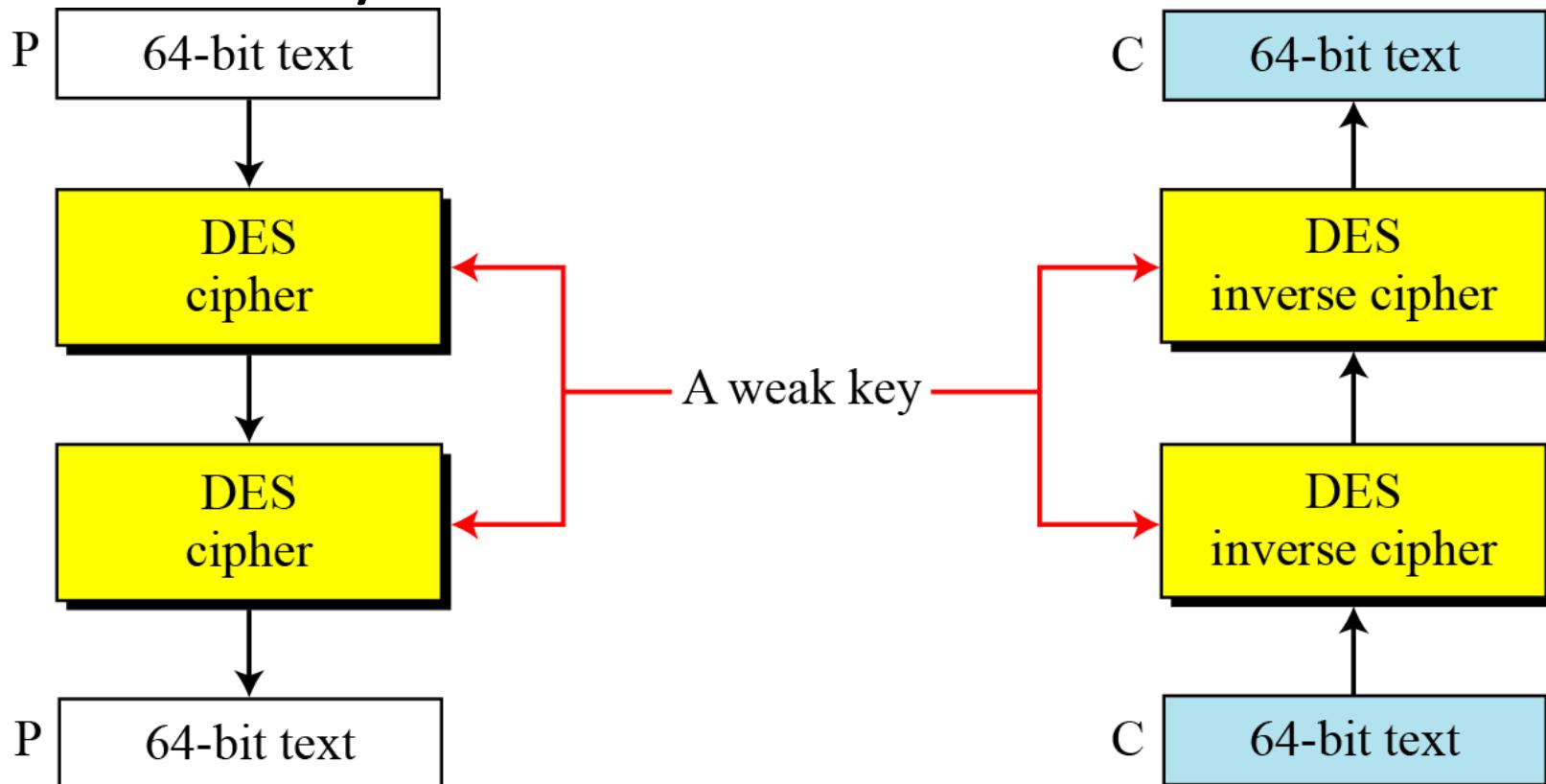
Key: 0x0101010101010101

Plaintext: 0x814FE938589154F7

Ciphertext: 0x1234567887654321

DES Weaknesses...

- Double encryption and decryption with a weak key



DES Weaknesses...

- Semi weak keys

Table 6.19 *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
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

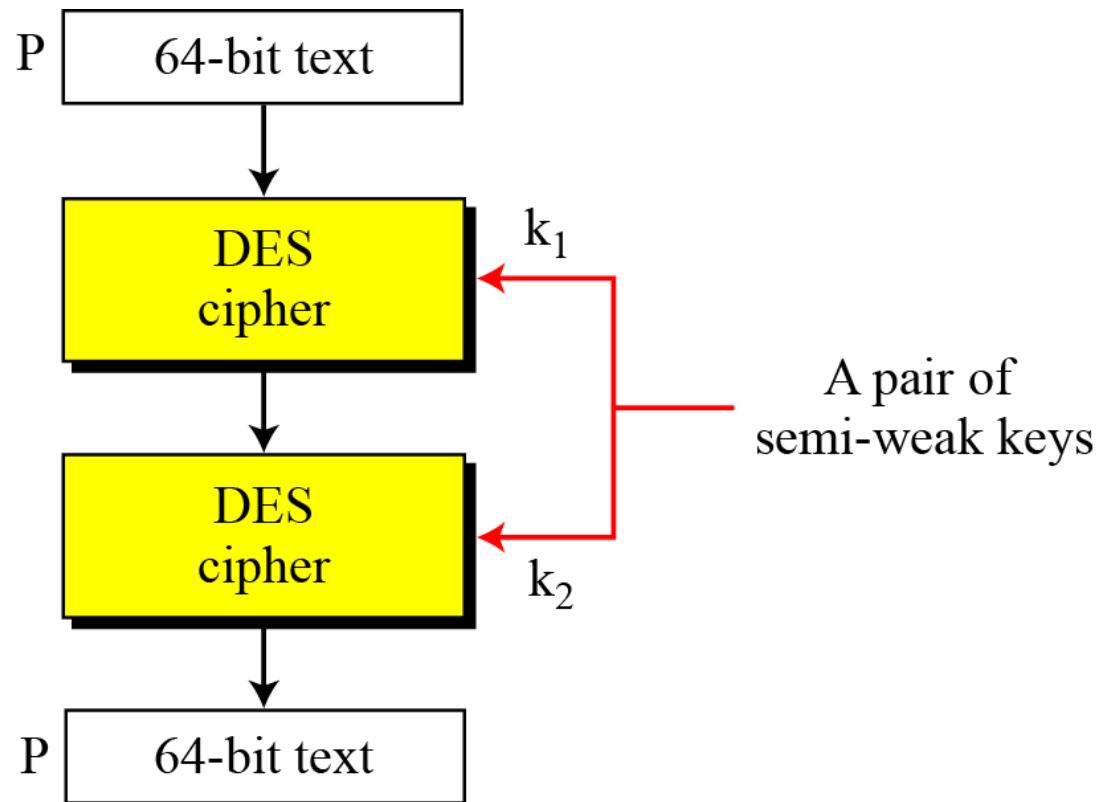
DES Weaknesses...

- Semi weak keys...

<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

DES Weaknesses...

- A pair of semi-weak keys in encryption and decryption



DES Weaknesses...

- What is the probability of randomly selecting a weak, a semi-weak, or a possible weak key?
 - DES has a key domain of 256. The total number of the above keys are 64 ($4 + 12 + 48$). The probability of choosing one of these keys is 8.8×10^{-16} , almost impossible.

DES Weaknesses...

- Key complement

Key Complement In the key domain (2^{56}), definitely half of the keys are *complement* of the other half. A **key complement** can be made by inverting (changing 0 to 1 or 1 to 0) each bit in the key. Does a key complement simplify the job of the cryptanalysis? It happens that it does. Eve can use only half of the possible keys (2^{55}) to perform brute-force attack. This is because

$$C = E(K, P) \rightarrow \bar{C} = E(\bar{K}, \bar{P})$$

In other words, if we encrypt the complement of plaintext with the complement of the key, we get the complement of the ciphertext. Eve does not have to test all 2^{56} possible keys, she can test only half of them and then complement the result.

DES Weaknesses...

- Key complement example

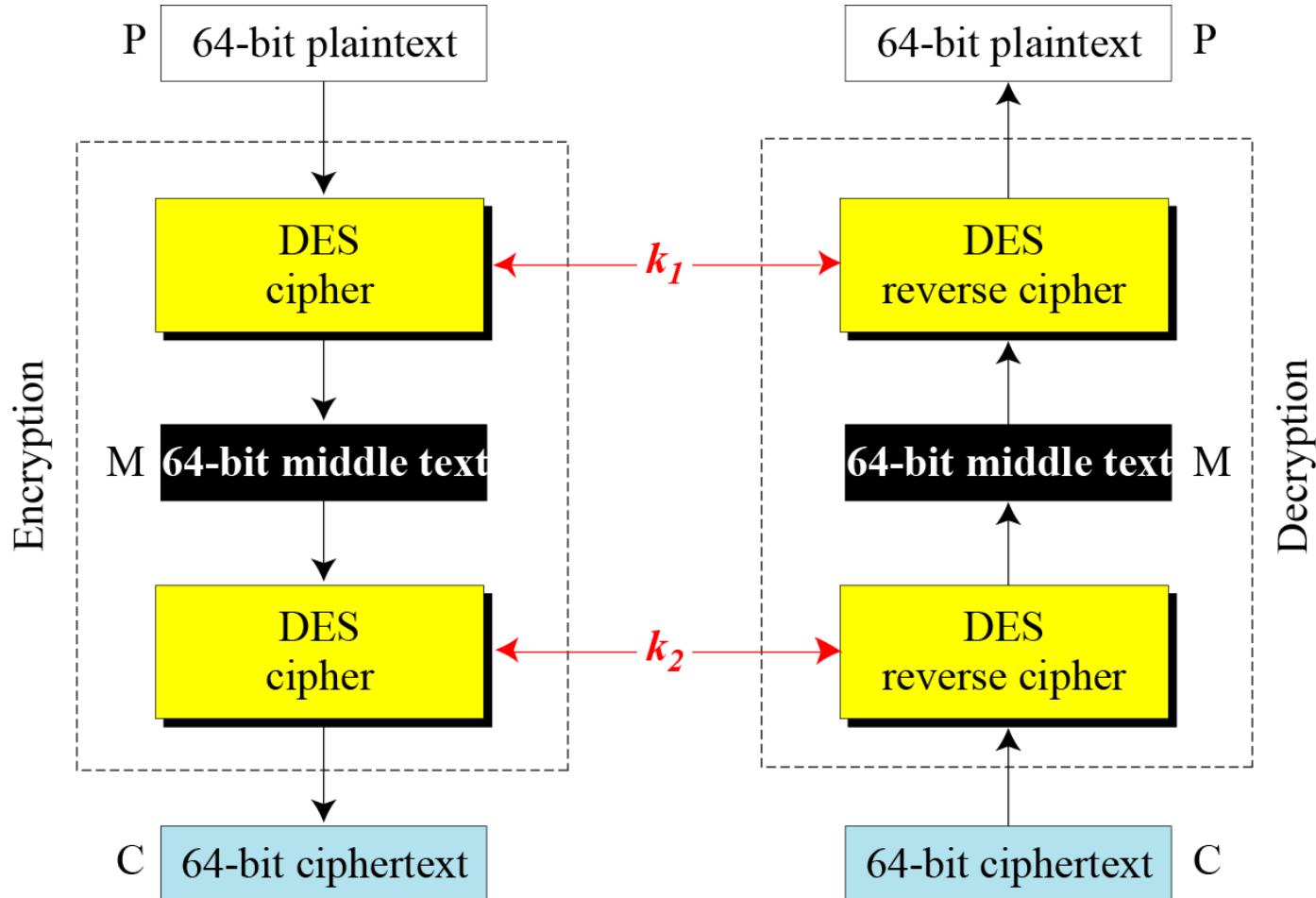
Table 6.20 *Results for Example 6.10*

	<i>Original</i>	<i>Complement</i>
Key	1234123412341234	EDCBEDCBEDCBEDCB
Plaintext	12345678ABCDEF12	EDCBA987543210ED
Ciphertext	E112BE1DEFC7A367	1EED41E210385C98

Multiple DES

- The major criticism of DES
 - its key length.
- Fortunately DES is not a group.
 - This means that we can use double or triple DES to increase the key size.

Double DES



Triple DES

