

# Network and Information Security

## Lecture 9

B.Tech. Computer Engineering  
Sem. VI.

Prof. Mrudang T. Mehta  
Associate Professor  
Computer Engineering Department  
Faculty of Technology,  
Dharmsinh Desai University, Nadiad

# Cryptanalysis

Two parts:

1. Finding the length of the key
2. Finding the key itself
  - For 1<sup>st</sup>, there are several methods, one such method is 'kasiski test'
    - Cryptanalyst searches for repeated text segments, of at least three characters in the cipher text.
    - Suppose that two of these segments are found and the distance between them is  $d$ .

- The cryptanalyst assumes that  $d \mid m$ , ie.  $d$  divides  $m$
- Where  $m$  = key length
- If more repeated segments are found with distances  $(d_1, d_2, \dots, d_n)$ , then take,  $\gcd(d_1, d_2, \dots, d_n) \mid m$
- This assumption is logical because if the two characters are same and are  $(k \times m)$  ( $k=1,2,\dots$ ) characters apart in the plaintext, they are same and  $(k \times m)$  characters apart from the ciphertext.
- Cryptanalyst uses segments of at least three characters to avoid the cases where the characters in the key are not distinct.

- The index of coincidence (IC) method is used to confirm the  $m$  value determined by the kasiski test.
- Definition:
- The index of coincidence of  $x = x_1, x_2, \dots, x_n$ , which is a string of length  $n$  formed by the alphabets  $A, B, \dots, Z$  is defined as probability that the random elements of  $x$  are the same.
- Frequencies of  $A, B, C, \dots, Z$  in  $x$  are denoted by the  $f_0, f_1, \dots, f_{25}$
- $$I_c(x) = \frac{\sum f_i C_2}{n C_2}$$
$$= \frac{\sum f_i \times (f_i - 1)}{n \times (n - 1)} = \sum (f_i/n)^2$$

- The index of coincidence (IC) is an invariant for any shift cipher.
- This is because in a shift cipher, the individual probabilities will get permuted but the sum of the squares of the probabilities will remain constant.
- For standard english language text, the value of IC is approximately (0.065).
- However, if all the letters are equally likely then the IC value is 0.038.

$n = \text{length}$  {there are 26 alphabets and each is appearing nearly equal number of times.}

$$P_i = (n/26)/n = 1/26$$

$$IC(x) = \sum P_i^2$$

$$i=25$$

$$= \sum_{i=0} (1/26)^2$$

$$= 26 (1/26)^2$$

$$= 1/26 = 0.038$$

- Since, these two values are quite far apart, the IC serves as an important tool to “distinguish” between English text and a random string of English alphabets.
- How to verify the value of  $m$ ?
- Arrange the given alphabetic string  $Y = Y_1 \dots Y_n$ , into  $m$  substring as follows:

$$Y_1 = Y_1 Y_{m+1} Y_{2m+1} \dots$$

$$Y_2 = Y_2 Y_{m+2} Y_{2m+2} \dots$$

$$Y_m = Y_m Y_{2m} Y_{3m} \dots$$

- If the value of  $m$  reported by Kasiski test is correct, each substring  $Y_i$ ,  $1 \leq i \leq m$  is a shift cipher which has been shifted by a key  $K_i$ .
- Hence, the expected value of  $I_c(Y_i)$  is about 0.065.
- However, if the guess of  $m$  is incorrect, each substring is a random string and thus the IC value is about 0.038.
- Thus we can confirm the value of  $m$  reported by the Kasiski test.



- Next we investigate a method to actually determine the key  $K = (k_1, k_2, \dots, k_m)$

Mutual Index of Coincidence (MI) between two alphabetic strings  $x$  and  $y$ .

Definition:

Suppose,  $x = x_1x_2\dots x_n$  and  $y = y_1y_2\dots y_n$ , are two alphabetic strings

- The mutual index of coincidence between  $x$  and  $y$  is the probability that a random element of  $x$  is equal to that of  $y$ .

- Thus if the probabilities of A, B.....are  $f_0, f_1, \dots, f_{25}$  and  $f'_0, f'_1, \dots, f'_{25}$  respectively in x and y, then  
Length of x =n, Length of y=n'

$$i=25$$

- $MI_c(x, y) = \sum_{i=0}^{i=25} f_i f'_i / nn'$

- For string  $x$  (shift by  $K_i$ )

Letter	A	B	C		Z
Probability	$p_0$	$p_1$	$p_2$		$p_{25}$

- Shift by key  $k_i$

$A + k_i$     $B + k_i$                        $Z + k_i$

$p_0$                        $p_1$                        $p_{25}$

- To find which out of  $A + k_i$     $B + k_i$                        $Z + k_i$  is mapped to A.

- Consider that a letter denoted by a number  $j$  between 0 to 25 in the unencrypted text thus becomes

$$j + k_i = 0 \pmod{26}$$

$$j = -k_i \pmod{26}$$

- Hence, corresponding probability of  $A$  in the encrypted text is  $p_j = p_{-k_i}$

Suffix values are modulo 26 (e.g.  $p_3 \equiv p_{-23}$ )

- Thus if we consider two strings  $x$  and  $y$ , which have been shifted by  $k_i$  and  $k_j$  respectively, the probability that both characters in  $x$  and  $y$  are  $A$  is  $p_{-k_i}p_{-k_j}$
- Similarly for  $B$ ,  

$$(j + k_i) = 1 \pmod{26}$$

$$j = (1 - k_i) \pmod{26}$$
- Likewise, the probability that both the characters are  $B$  is  $p_{1-k_i}p_{1-k_j}$  and so on.

- Total probability that randomly selected characters are same from X and Y

= sum of all such probabilities

= Both are A or Both are B or ...Both are Z

Since all the events are Mutually Exclusive, It can be written as sum.

= P(both are A's) + P(both are B's) + ... + P(both are Z's)

=  $p_{-ki} p_{-kj} + p_{1-ki} p_{1-kj} + \dots + p_{25-ki} p_{25-kj}$

$h=25$

$$MI_c(x,y) = \sum_{h=0} p_{h-ki} p_{h-kj}$$

- $h' = h - k_i \Rightarrow h = h' + k_i$

$$h=25$$

$$MI_c(x,y) = \sum_{h=0} p_{h-k_i} p_{h-k_j}$$

$$h=25-k_i$$

$$MI_c(x,y) = \sum_{h=-k_i} p_{h'} p_{h'+k_i-k_j}$$

- $h' = -k_i$  to  $(25 - k_i)$  is equivalent to  $h' = 0$  to 25

$$h'=25$$

$$MI_c(x,y) = \sum_{h'=0} p_{h'} p_{h'+k_i-k_j}$$

$$h=25$$

$$MI_c(x,y) = \sum_{h=0} p_h p_{h+k_i-k_j}$$

$$\text{If } k_i = k_j \text{ then } MI(X,Y) = \sum_{h=0}^{h=25} p_h^2 = 0.065$$

# Network and Information Security

## Lecture 10

B.Tech. Computer Engineering  
Sem. VI.

Prof. Mrudang T. Mehta  
Associate Professor  
Computer Engineering Department  
Faculty of Technology,  
Dharmsinh Desai University, Nadiad



# Cryptanalysis of Vigenere Cipher (Continue...)

1. For two strings  $x$  and  $y$  ciphered using keys  $k_i$  and  $k_j$  the value of  $MI_c(x, y)$  depends on the difference  $k_i - k_j \pmod{26}$ .
2. A relative shift of  $s$  yields the same value as  $26 - s$ .

When  $k_i - k_j = 0$ , the value of  $MI_c$  is maximum and is equal to 0.065. However, for other values, the estimate is comparatively less and ranges from 0.032 to 0.045 on an average.

- So in order to find the actual key, we divide the given string of encrypted characters into  $m$  rows.
- Each row is a shift cipher, which has been shifted by a key,  $k_i$ .
- Thus for each row we find the Mutual Index of Coincidence with respect to an unencrypted english text.
- We compute the MI values by varying the keys,  $k_i$  from 0 to 25.
- The values for which the MI values become close to 0.065 will indicate the correct key,  $k_j$ .
- This process is repeated for the  $m$  rows to obtain the entire key.

# Example

- Let us assume we have intercepted the following cipher text:

LIOMWGFEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTHVTS  
GXQOVGCSVETQLTJSUMY.WVEUVLXEWSLGFZMVVWLGYHCUSWXQHKVGSHEEV  
FLCFDGVSUMPHKIRZDMPHHBVVWVWJWIXWIXGFWLTSHGJOUEEHHVUCFVGOW  
ICQLTJSUXGLW.

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

String	First Index	Second Index	Difference
QLT	65	165	100
LTJ	66	166	100
TJS	67	167	100
JSU	68	168	100
SUM	69	117	48
VWV	72	132	60

- The greatest common divisor of differences is 4, which means that the key length is multiple of 4.
- We try confirm this guess by the Index of Coincidence test.
- We divide the ciphertext into 4 rows.
- We also mention the corresponding index of coincidence values.
- The high values of the IC confirms the key length reported by the kasiski test.

1 <sup>st</sup> String IC=0.067 677	LWGWCR AOKTEPGTQCTJVUEGVGUQGECVPRPVJGTJEUGCJG
2 <sup>nd</sup> String IC=0.074 747	IGGGQHGWGKVCTSOSQSWVWFVYSHSVFSHZHWWFSOHCOQSL
3 <sup>rd</sup> String IC=0.070 707	OFDHURWQZKLZHGVVLUVLSZWHWKHF DUKDHVIWHUHF WL UW
4 <sup>th</sup> string IC=0.076 768	MEVHCWILEMWV VXGETMEXLMLCXVELGMIMBWXLGEVVITX

First line is made up of characters 1,5,9,....,Second line is 2,6,10,....and so on.

- Thus we perform the Mutual index of coincidence to obtain the actual key value. Running the test, we obtain that the key value is CODE and the corresponding plain text is:
- Julius Caesar used a cryptosystem in his wars, which is now referred to as Caesar cipher. It is an additive cipher with the key set to three. Each character in the plain text is shifted three characters to create ciphertext.

# Network and Information Security

## Lecture 11

B.Tech. Computer Engineering  
Sem. VI.

Prof. Mrudang T. Mehta  
Associate Professor  
Computer Engineering Department  
Faculty of Technology,  
Dharmsinh Desai University, Nadiad

# Hill Cipher

- Polyalphabetic cipher
- Invented by Lester S. Hill
- The plain text is divided into equal-size blocks.
- The blocks are encrypted one at a time in such a way that each character in the block contributes to the encryption of other characters in the block.
- For this reason, the Hill cipher belongs to a category of ciphers called block ciphers.



- In a Hill cipher, the key is a square matrix of size  $m \times m$  in which  $m$  is the size of the block.
- If we call the key matrix  $K$ , each element of the matrix is  $K_{i,j}$

$$K = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix}$$

- How one block of the ciphertext is encrypted.
- If we call the  $m$  characters in the plaintext block  $P_1, P_2, \dots, P_m$ , the corresponding characters in the cipher text block are  $C_1, C_2, \dots, C_m$ .

$$C_1 = P_1 K_{11} + P_2 K_{21} + \dots + P_m K_{m1}$$

$$C_2 = P_1 K_{12} + P_2 K_{22} + \dots + P_m K_{m2}$$

$$C_m = P_1 K_{1m} + P_2 K_{2m} + \dots + P_m K_{mm}$$

- Note- Not all square matrices have multiplicative inverse in  $Z_{26}$
- Bob will not be able to decrypt the cipher text sent by Alice if the matrix does not have a multiplicative inverse.

# Example

Plain text: code is ready

Matrix representation of plain text can make 3 x 4 matrix when adding extra bogus character z to the last block and removing the spaces.

$$\begin{pmatrix} c & o & d & e \\ i & s & r & e \\ a & d & y & z \end{pmatrix} \quad \begin{pmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{pmatrix}$$

P

$$\begin{pmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{pmatrix}$$

K

$$\begin{pmatrix} 09 & 07 & 11 & 13 \\ 04 & 07 & 05 & 06 \\ 02 & 21 & 14 & 09 \\ 03 & 23 & 21 & 08 \end{pmatrix}$$

C

$$\begin{pmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 07 \\ 05 & 08 & 18 & 18 \end{pmatrix}$$

$$C_1 = P_1 K_{11} + P_2 K_{21} + P_3 K_{31} + P_4 K_{41}$$

$$C_1 = (2)(9) + (14)(4) + (3)(2) + (4)(3)$$

$$= 18 + 56 + 6 + 12$$

$$= 92 \bmod 26$$

$$= 14$$

$C_1$	$2*9 + 14*4 + 3*2 + 4*3 = 92 \% 26 = 14$
$C_2$	$2*7 + 14*7 + 3*21 + 4*23 = 267 \% 26 = 7$
$C_3$	$2 * 11 + 14*5 + 3*14 + 4*21 = 10$
$C_4$	$2*13 + 14*6 + 3*9 + 4*8 = 13$
$C_5$	$8*9 + 18*4 + 17*2 + 4*3 = 8$
$C_6$	$8*7 + 18*7 + 17*21 + 4*23 = 7$
$C_7$	$8*11 + 18*5 + 17*14 + 4*21 = 6$
$C_8$	$8*13 + 18*6 + 17*9 + 4*8 = 7$
$C_9$	$0*9 + 3*4 + 24*2 + 25*3 = 5$
$C_{10}$	$0*7 + 3*7 + 24 * 21 + 25*23 = 8$
$C_{11}$	$0*11 + 3*5 + 24*14 + 25*21 = 18$
$C_{12}$	$0*13 + 3*6 + 24*9 + 25*8 = 18$

- Decryption

$$\begin{array}{cccc}
 \left( \begin{array}{cccc} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{array} \right) & = & \left( \begin{array}{cccc} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 07 \\ 05 & 08 & 18 & 18 \end{array} \right) & \left( \begin{array}{cccc} 02 & 15 & 22 & 03 \\ 15 & 00 & 19 & 03 \\ 09 & 09 & 03 & 11 \\ 03 & 23 & 21 & 08 \end{array} \right) \\
 P & & C & K^{-1}
 \end{array}$$



- $A^{-1} = 1/|A| * \text{adj}(A)$

- Example

- $K = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \quad 2 \times 2$

- $\text{Det}(K) = 2 \times 2 - 1 \times 1 = 4 - 1 = 3$

- $K^{-1} = 1/\text{Det}(K) \text{ adj}(K)$

- $= (3)^{-1} \text{ adj}(K)$

- $(3)^{-1} \bmod 26$

q	r1	r2	r	t1	t2	t
8	26	3	2	0	1	-8
1	3	2	1	1	-8	9
2	2	1	0	-8	9	-26
	1	0		9	-26	

$$(3)^{-1} \bmod 26 = 9$$

$$K^{-1} = 9 * \text{adj}(K)$$

- Cofactor matrix
- Cofactor of  $K_{11}$  [ 2] =  $(-1)^{1+1} \times 2 = 2$
- Cofactor of  $K_{12}$  [ 1] =  $(-1)^{1+2} \times 1 = -1$
- Cofactor of  $K_{21}$  [ 1] =  $(-1)^{2+1} \times 1 = -1$
- Cofactor of  $K_{22}$  [ 2] =  $(-1)^{2+2} \times 2 = 2$

- Cofactor matrix = 
$$\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$$

- Adjoint = transpose of cofactor matrix

- $\text{Adj}(K) = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$
- $K^{-1} = 9 * \text{adj}(K) = \begin{pmatrix} 18 & -9 \\ -9 & 18 \end{pmatrix} \pmod{26}$

- $K^{-1} = \begin{pmatrix} 18 & 17 \\ 17 & 18 \end{pmatrix}$

- Encryption

- $C = (P \times K) \bmod 26$

- Plain text = abcd

- Plain text block =  $\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}$

- $C = (P \times K) \bmod 26$

- $\begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix} \times \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 7 & 8 \end{pmatrix}$

- Decryption

- $P = (C \times K^{-1}) \bmod 26$

$$= \begin{pmatrix} 1 & 2 \\ 7 & 8 \end{pmatrix} \times \begin{pmatrix} 18 & 17 \\ 17 & 18 \end{pmatrix} \bmod 26$$

$$= \begin{pmatrix} 52 & 53 \\ 262 & 263 \end{pmatrix} \bmod 26$$

$$= \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}$$

$$= P = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$



# Cryptanalysis

- Bruteforce is not possible
- Each entry in the matrix can have one of the possible 26 values
- (at first glance)
- Number of keys =  $26^{m * m} = 26^{m^2}$
- Not all of the matrices have multiplicative inverses (Smaller key domain but huge)
- Statistical attack is not possible as one Cipher Text depends on many plain text characters

- Possible attacks
  - Known plain text attack
  - Chosen plain text attack
- $K = (C \times P^{-1}) \bmod 26$
- Eve can choose Invertible  $P$  and can obtain  $C$  using chosen plain text attack
- Using received  $C$  and  $P^{-1}$  , Eve can guess the key.
- Difficulty: Value of  $m$  not known
- Chosen plain text attack is difficult to launch

# One Time Pad Cipher

- Goal of Cryptography is perfect secrecy.
- Shannon - It can be achieved if each plaintext symbol is encrypted with a key randomly chosen from a key domain
- Additive cipher can become a perfect cipher if the key that is used to encrypt each character is chosen randomly from the key domain i.e.  $\{0,1,2,\dots,25\}$
- If the first character is encrypted with key 4, the second character is encrypted with key 2, the third character is encrypted with key 21.

- Invented by Vernam
- The key has the same length as the plain text and is chosen completely in random.
- Difficulty:
- It is a perfect cipher but it is impossible to implement commercially.
- If the key must be newly generated each time, how can Alice tell the new key each time she has a message to send?

# Network and Information Security

## Lecture 12

B.Tech. Computer Engineering  
Sem. VI.

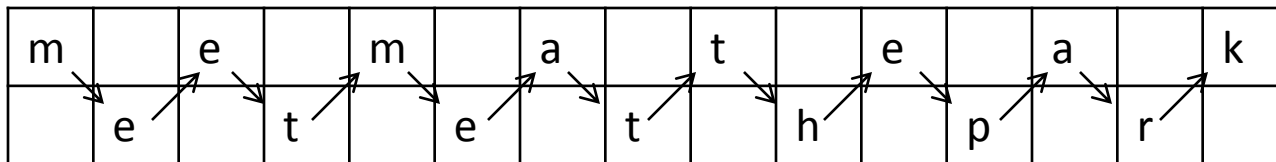
Prof. Mrudang T. Mehta  
Associate Professor  
Computer Engineering Department  
Faculty of Technology,  
Dharmsinh Desai University, Nadiad

# Transposition Ciphers

- A transposition cipher does not substitute one symbol for another, instead it changes the location of the symbols.
- A symbol in the first position of the plaintext may appear in the tenth position of the cipher text.
- A symbol in the eighth position in the plaintext may appear in the first position of the cipher text.
- A transposition cipher reorders (transposes) the symbols.

# Keyless transposition ciphers

- There are two methods
- One: the text is written into a table column by column and then transmitted row by row.
- Second: the text is written into the table row by row and then transmitted column by column
- Example 1: Rail fence cipher
- Plain text is arranged in two lines as a zig zag pattern



- Plain text: Meet me at the park
- Cipher text: MEMATEAKETETHPR
- Bob receives the cipher text and divides it into half.
- First half forms the first row,
- Second half forms the second row
- Bob reads the result in zig zag.
- Cryptanalysis is easy. No key is used.



- Example 2: Alice and Bob can agree on the number of columns and use the second method. Alice writes the same plaintext, row by row, in a table of four columns.

	1	2	3	4
1	m	e	e	t
2	m	e	a	t
3	t	h	e	p
4	a	r	k	

- She then creates the cipher text MMTAEEHREAEKTTP by transmitting the characters column by column.
- Bob receives the cipher text and follows the reverse process.
- He writes the received message, column by column, and reads it row by row as the plaintext.
- Eve can easily decipher the message if she knows the number of columns.

- Length of the plaintext = 15
- Number of columns = 4 ( known to both alice and bob)
- Number of rows =  $\lceil \text{length} / \text{number of column} \rceil$   

$$= 15/4 = 4$$
- Write text column by column
- Read it row by row

- Example 3
- The following shows the permutation of each character in the plaintext into the cipher text based on the position. (Example2)

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

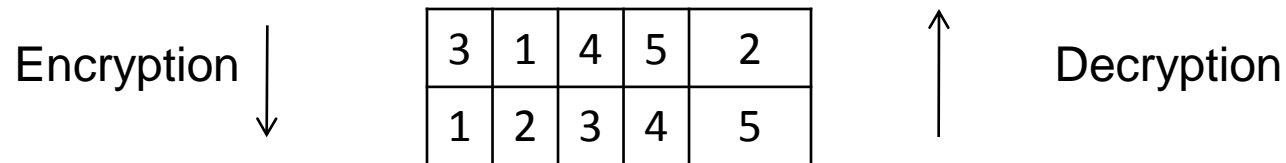
- The second character in the plaintext has moved to the fifth position in the cipher text
- The third character has moved to the ninth position, and so on.
- The pattern in the permutation (1,5,9,13), (2,6,10,14), (3,7,11,15), (8,12)
- In each section, the difference between the two adjacent numbers is 4.

# Keyed transposition ciphers

- Divide the plaintext into groups of predetermined size, blocks, and then use a key to permute the characters in each block separately.
- Example 4
- Alice and Bob have agreed to divide the text into groups of five characters and then permute the characters in each group .

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

- The key used for encryption and decryption is a permutation key,



- The third character in the plain text block becomes the first character in the cipher text block;
- The first character in the plain text block becomes the second character in the cipher text block

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

- Bob divides the cipher text into 5- character groups and, using the key in the reverse order, finds the plain text.

Encryption Key: 3,1,4,5,2

e	n	e	m	y
E	E	M	Y	N

a	t	t	a	c
T	A	A	C	T

k	s	t	o	n
T	K	O	N	S

i	g	h	t	z
H	I	T	Z	G

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

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

1	2	3	4	5
1	2	3	4	5

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



Decryption Key: 2,5,1,3,4

E	E	M	Y	N
e	n	e	m	y

T	A	A	C	T
a	t	t	a	c

T	K	O	N	S
k	s	t	o	n

H	I	T	Z	G
i	g	h	t	z

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

- Combining Two Approaches:

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

Encryption

Write Row by Row (Step1)

(Assume key = number of columns = 5)

	1	2	3	4	5
1	e	n	e	m	y
2	a	t	t	a	c
3	k	s	t	o	N
4	i	g	h	t	z

- Step 2 Use key to permute columns
- 3 1 4 5 2

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

- Step 3 Read column by column

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

- At decryption side, number of columns are known
- Length=20, Number of columns are 5
- Number of rows =  $20/5 = 4$
- Step 1 Write column by column

	1	2	3	4	5
1	E	E	M	Y	N
2	T	A	A	C	T
3	T	K	O	N	S
4	H	I	T	Z	G

- Step 2: Use key 2,5,1,3,4 to permute the columns

	1	2	3	4	5
1	e	n	e	m	y
2	a	t	t	a	c
3	k	s	t	o	N
4	i	g	h	y	z

- Step 3: Read row by row
- e n e m y a t t a c k s t o n i g h t z

# Transposition Cipher Cryptanalysis

- Assume cipher text length is L
- We don't know the length of the key
- So, we can assume key length and then proceed
- Plain text: `enigma`
- Key 312
- PT: e n i   g m a
- 3 1 2    3 1 2
- CT: I E N    A G M

- Given, IENAGM
- We want to apply Bruteforce:
- Assume key length is 1
  - It is not possible because character is permuted by itself
  - No permutation
- Key length is 2
  - (1,2), (2,1) {two possibilities }
  - (1,2) does not do any permutation
  - Apply (2,1) after dividing CT into blocks of 2 characters each, E I A N M G <= Does not mean anything

- Next guess is Key length is 3.

- $3! = 6$  Try 1, 3, 2

- 1,2,3                    i e n            a g m
- 1,3,2                    1 3 2            1 3 2

- 1,3,2                    Try
- 2,3,1                    2, 3, 1
- 2,1,3                    e n i g m a

- 3,1,2

- 3,2,1                    So, (2,3,1) is correct one for decryption.  
Note (3,1,2) is the encryption key but (2,3,1) is used for decryption because both can be obtained easily provided one is given.



- Brute force trials required
- $2! + 3! + 4! + 5! + \dots + L!$
- Where  $L$  = cipher text length
- $\sum i!$  (for  $i=2$  to  $L$ )

# Network and Information Security

## Lecture 13

B.Tech. Computer Engineering  
Sem. VI.

Prof. Mrudang T. Mehta  
Associate Professor  
Computer Engineering Department  
Faculty of Technology,  
Dharmsinh Desai University, Nadiad

# Transposition cipher Cryptanalysis (continue..)

- Better Approach
- Example
- Suppose that Eve has intercepted the ciphertext “EEMYNTAACTTKONSHITZG”
- The message length  $L=20$  means that the number of columns can be 1,2,4,5,10 or 20.
- Eve ignores the first value because it means only one column and no permutation.

- If the number of column is 2, the only two permutations are (1,2), (2,1).
- (1,2) – no permutation
- (2,1) –
  - EE MY NT AA CT TK ON SH IT ZG
  - ee ym tn aa tc kt no hs ti gz (does not make sense)
  - Therefore  $(2! - 1)$  trials

- Next,  $(4! - 1)$  trials  $(24 - 1)$  [ first one is  $(1\ 2\ 3\ 4)$ ]
- Next,  $(5! - 1)$  trials  $(120 - 1)$  [  $(1\ 2\ 3\ 4\ 5)$  does not make permutation]
- This has to be done till we find proper guess
- Worst case

$$= (2! - 1) + (4! - 1) + (5! - 1) + (10! - 1) + (20! - 1)$$

Number of trials are required which are better than the brute force

- Pattern attack
- The cipher text created from the keyed transposition cipher has some repeated pattern

e	n	e	m	y	<b>a</b>	<b>t</b>	<b>t</b>	<b>a</b>	<b>c</b>	k	s	t	o	n	<b>i</b>	<b>g</b>	<b>h</b>	<b>t</b>	<b>z</b>
3	1	4	5	2	<b>3</b>	<b>1</b>	<b>4</b>	<b>5</b>	<b>2</b>	3	1	4	5	2	<b>3</b>	<b>1</b>	<b>4</b>	<b>5</b>	<b>2</b>
e	e	m	y	n	<b>t</b>	<b>a</b>	<b>a</b>	<b>c</b>	<b>t</b>	<b>t</b>	k	o	n	s	<b>h</b>	<b>i</b>	<b>t</b>	<b>z</b>	<b>g</b>

e n e m y

a t t a c

k s t o n

i g h t z

3 1 4 5 2

e e m y n

t a a c t

t k o n s

h i t z g

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

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

Difference between 2 adjacent is 5 in all the groups

- If Eve knows/ can guess the number of columns (which is 5 in this case) she can organize the ciphertext into groups of 4 characters.
- Permuting the groups can provide clue to find the plaintext.
- In the above example
- $L = 20$  , number of rows =  $L/\text{number of columns}$
- Number of rows =  $20/5 = 4$



- Fill the cipher text

Row 1    e   e   m   y   n

Row 2    t   a   a   c   t

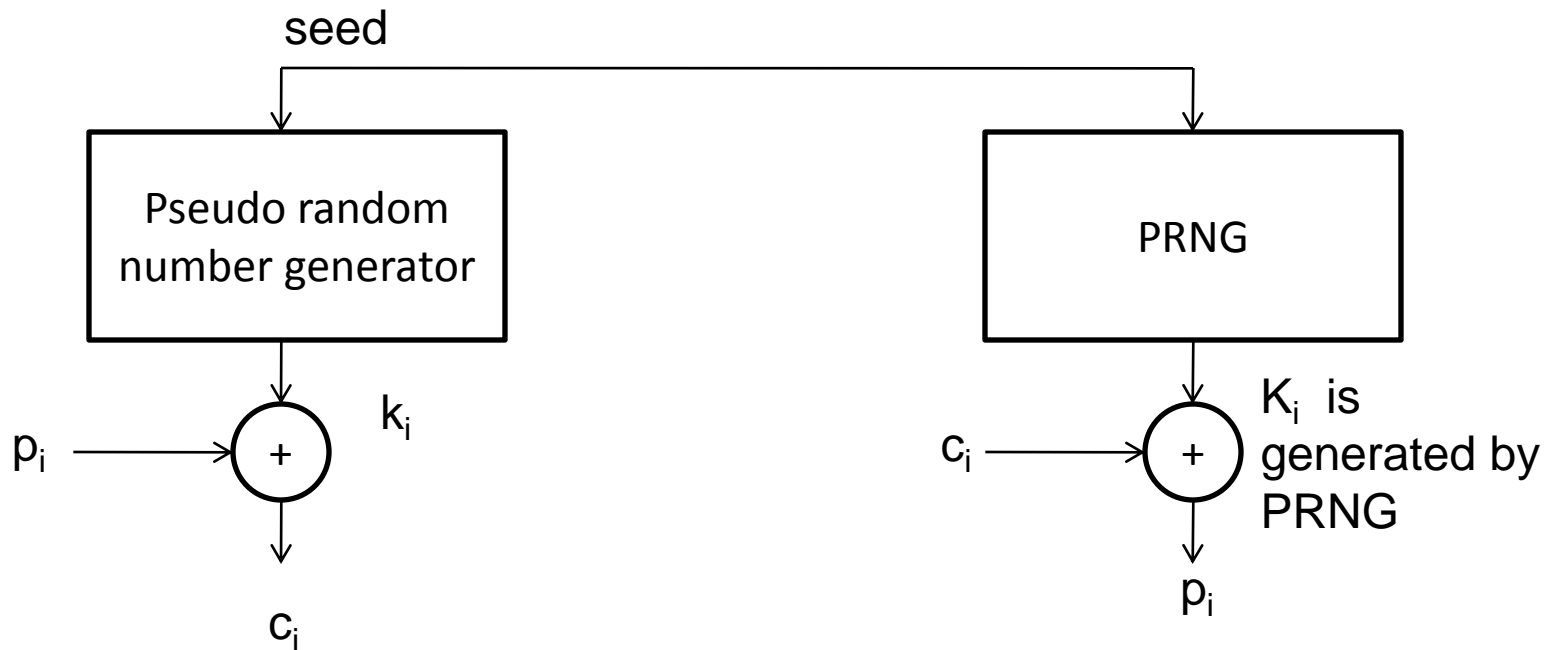
Row 3    t   k   o   n   s

Row 4    h   i   t   z   g

Find permutation which gives meaningful answer when  
read row by row

- Traditional ciphers
  - Block cipher (encrypts more than 1 characters at a time)
    - Hill cipher, playfair cipher
  - Stream cipher (encrypts one characters at a time)
    - Shift cipher

- How to approximate one time pad cipher?



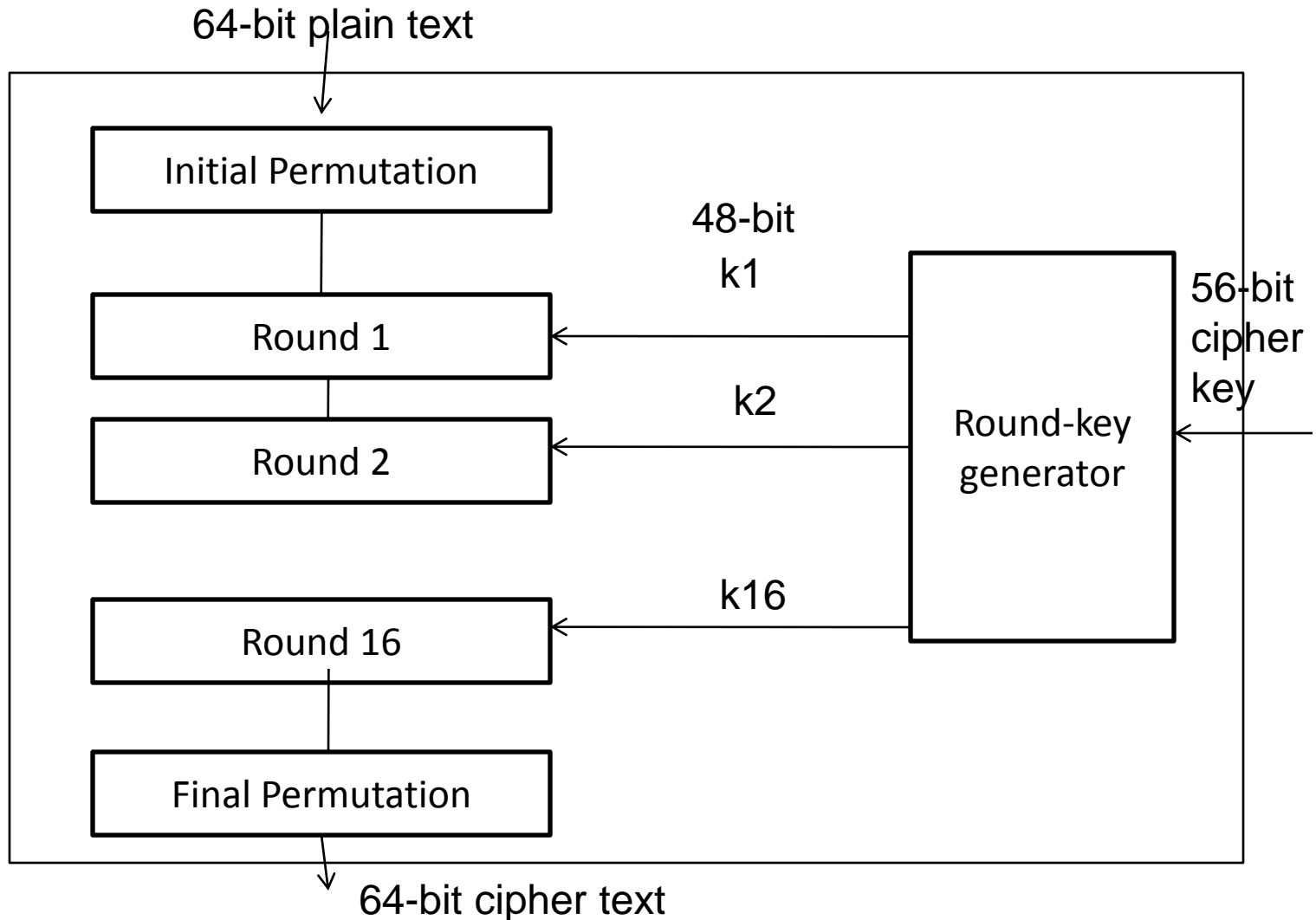
$$C_i = p_i \oplus k_i$$

$$p_i = C_i \oplus k_i$$

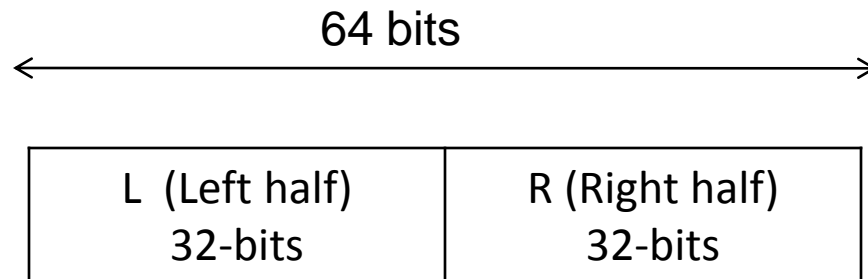
# Data Encryption Standard (DES)

- It is symmetric key cipher
- It is block cipher published by NIST (National Institute of Standard and Technology)
- Block size = 64 bits
- i.e. It encrypts 64 bits at a time
- It has 16 rounds
- Structure of each round is same

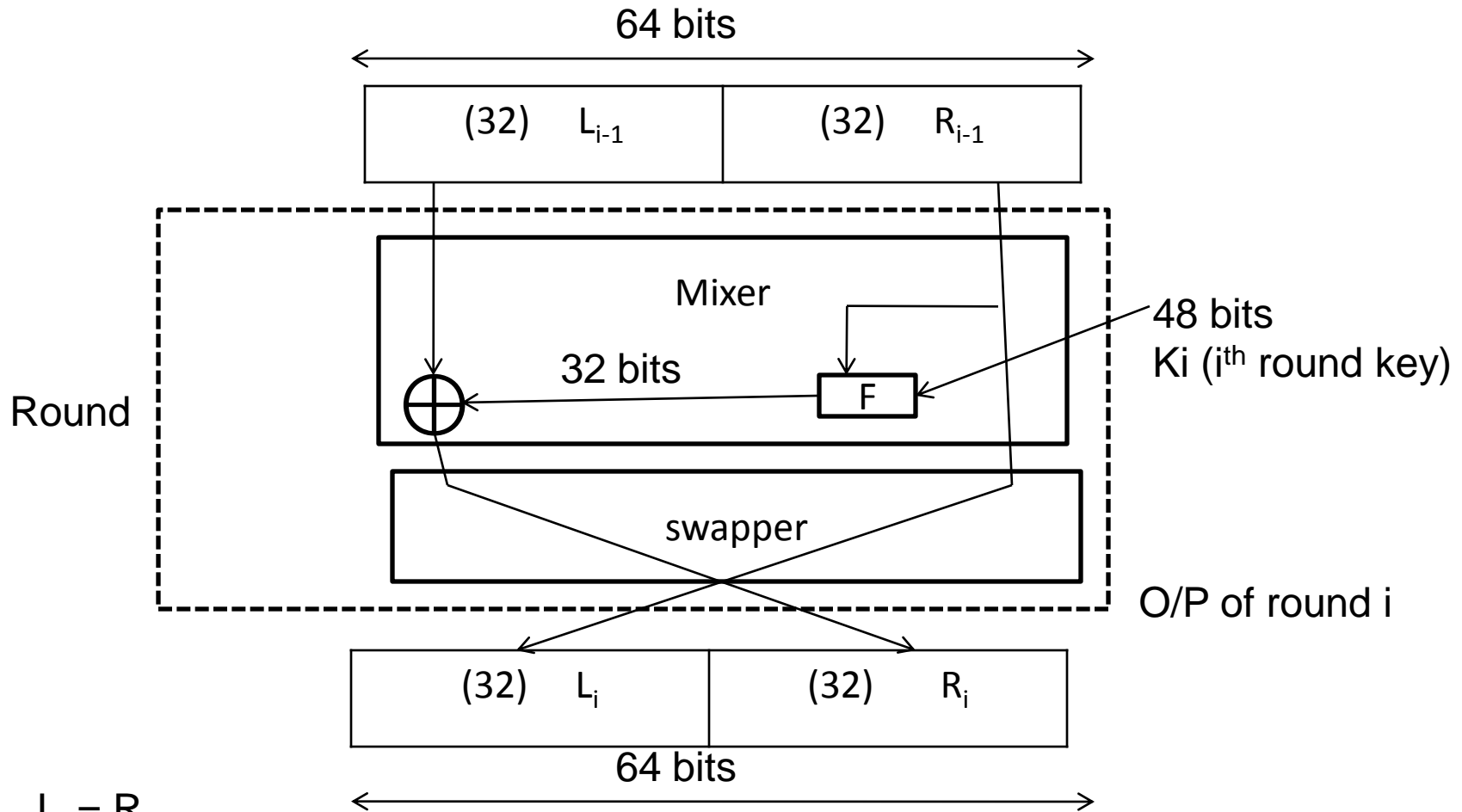
# DES Structure



- Round structure
- For round number  $i$
- 64 bits input to round  $i$  is divided into two halves each of 32 bits



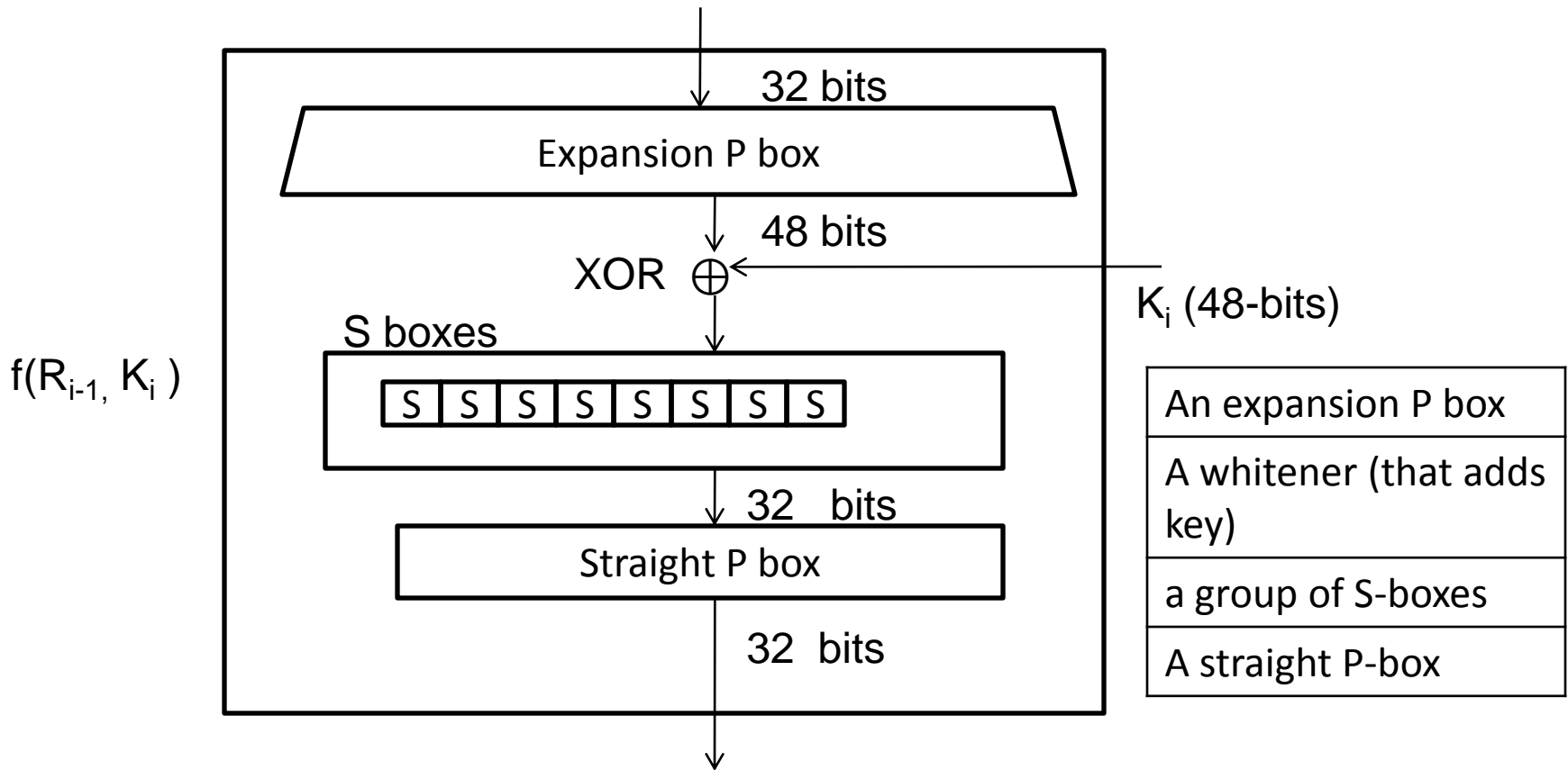
# A Round in DES



$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, k_i) \quad F \text{ is Round function}$$

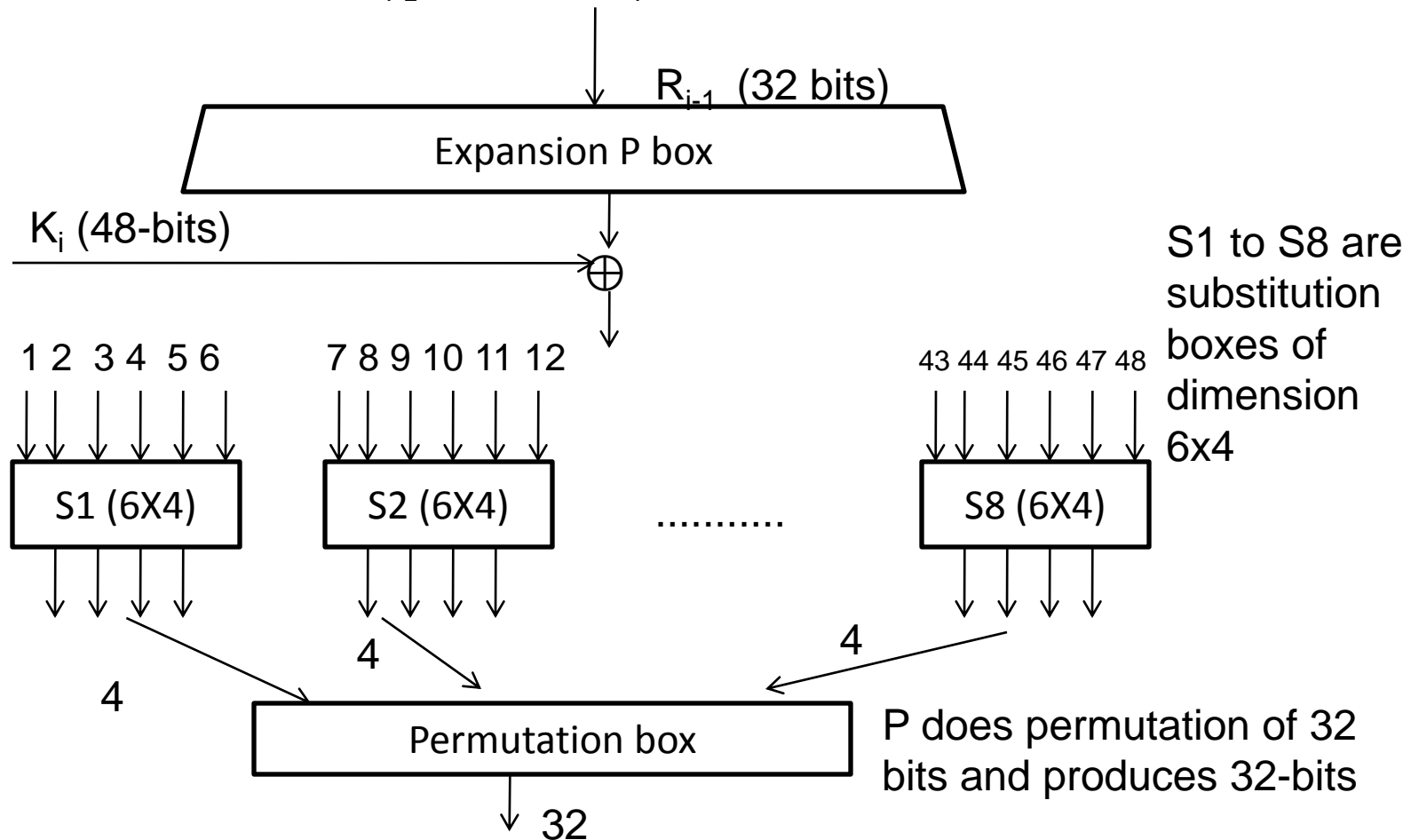
# DES Function





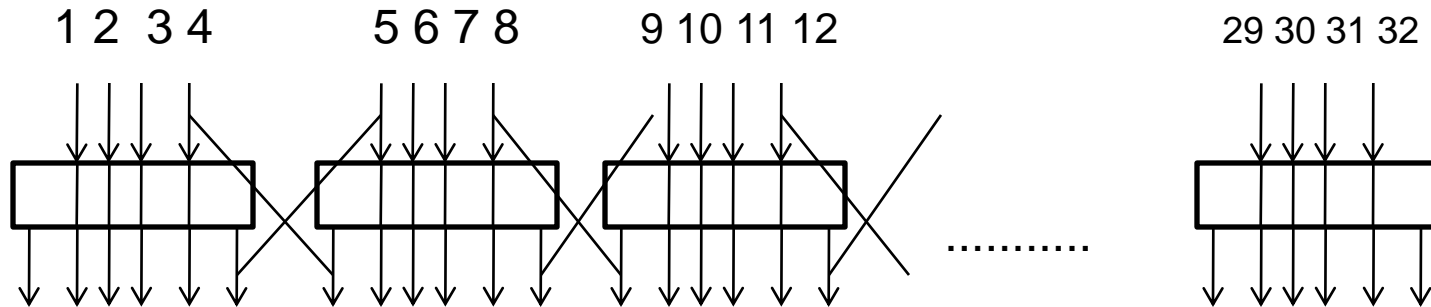
# S-boxes

Round function, Input:  $R_{i-1}$  (32 bits),  $K_i$  (48 bits), Output: 32 bits



- Thus output of rand function  $F$  is 32 bits
  1.  $R_{i-1}$  is expanded to 48- bits
  2. 48-bits are ex-ored with  $K_i$  (ith round key of 48 bits)
  3. Then 8 groups each of 6 bits are given to respective S-boxes to produce 8 groups of 4 bits
  4. Output 32 bits are permuted

# Expansion permutation



First bit is copied  
from last 32<sup>nd</sup> bit  
of input

last bit is copied  
from first 1<sup>st</sup> bit of  
input

# Expansion P-box table

32	1	2	3	4	5
4	5	6	7	8	9
8	9	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	30	31	32	1

- First entry (1<sup>st</sup> row , 1<sup>st</sup> column) is 32 which indicates the index of bit from where we need to copy i.e. 1<sup>st</sup> bit of output is 32<sup>nd</sup> bit of input
- Thus by copying bits, we are able to generate 48 bits
- Basically we are adding redundancy i.e. We are creating 16 more bits by copying bits from certain position

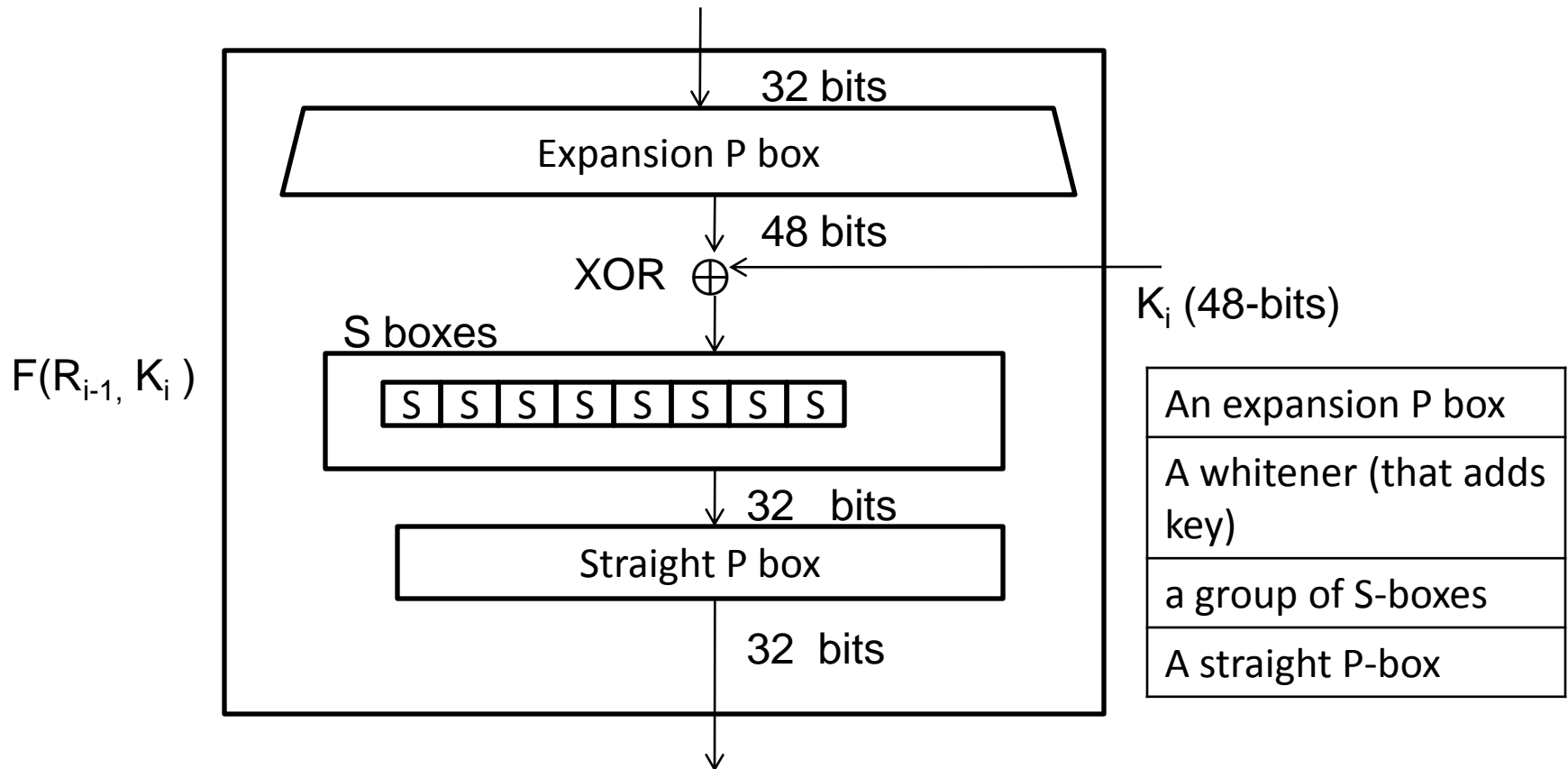
# Network and Information Security

## Lecture 14

B.Tech. Computer Engineering  
Sem. VI.

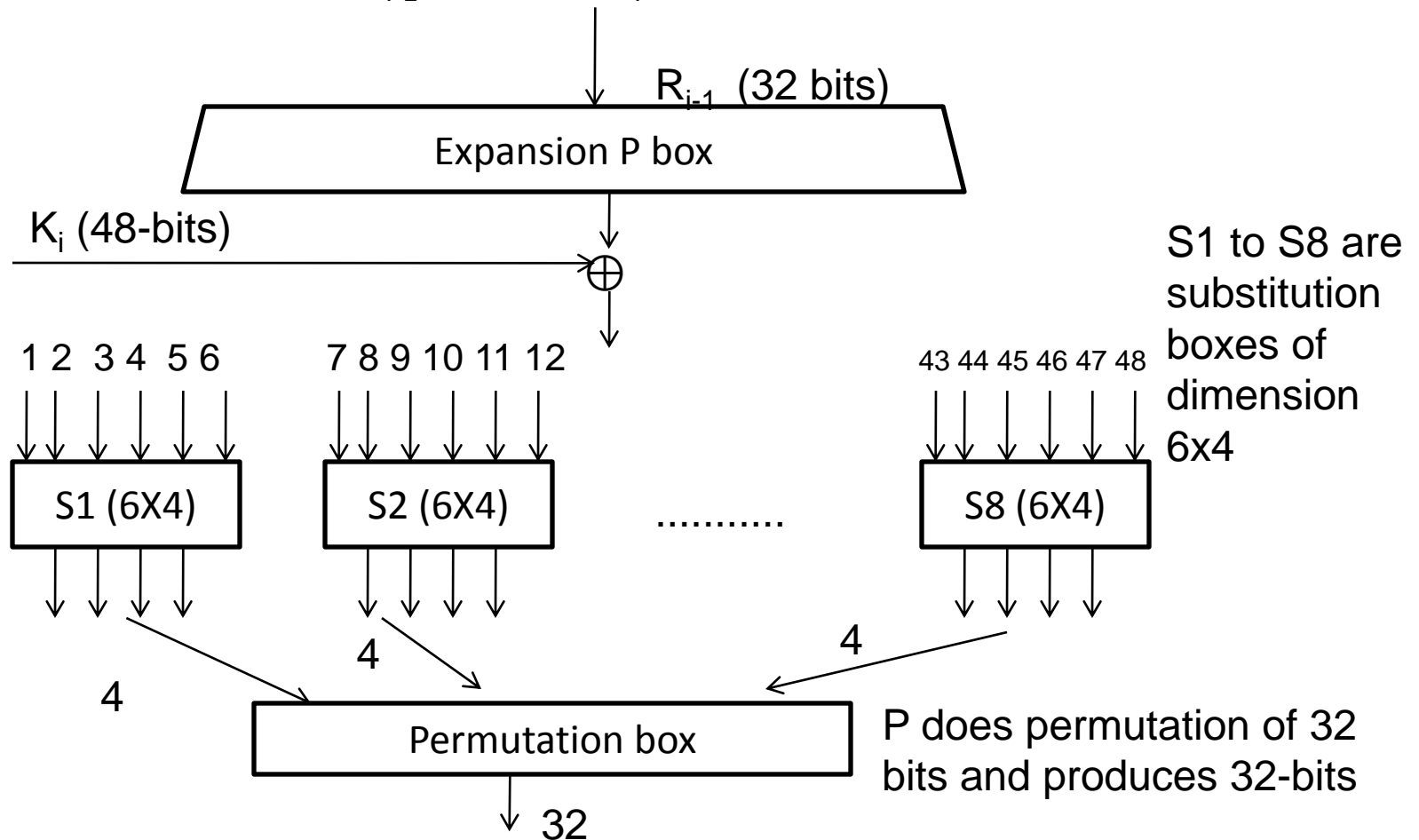
Prof. Mrudang T. Mehta  
Associate Professor  
Computer Engineering Department  
Faculty of Technology,  
Dharmsinh Desai University, Nadiad

# DES Function f



# S-boxes

Round function, Input:  $R_{i-1}$  (32 bits),  $K_i$  (48 bits), Output: 32 bits



# Expansion P-box table

## Step 1

32	1	2	3	4	5
4	5	6	7	8	9
8	9	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	30	31	32	1

- First entry (1<sup>st</sup> row , 1<sup>st</sup> column) is 32 which indicates the index of bit from where we need to copy i.e. 1<sup>st</sup> bit of output is 32<sup>nd</sup> bit of input
- Thus by copying bits, we are able to generate 48 bits
- Basically we are adding redundancy i.e. We are creating 16 more bits by copying bits from certain position



- Step 2 (48 bits output of step 1) is ex-ored with 48-bits of round key  $K_i$

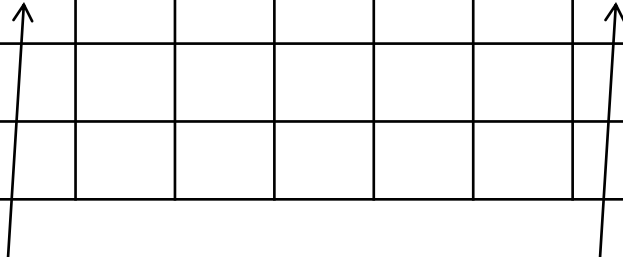
a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

# S-box

Step 3 S-box is of dimension of 6x4 which means it maps or substitutes 4 bits for 6 bits of input

6-bits  $\Rightarrow 2^6 = 64$  values are arranged in the following manner (4(rows) \* 16(columns)=64 , 4 bits entries)

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



Value from 0 to 15

Each output entry is of 4-bits (0 to 15)

- DES uses 8 S-boxes , each with a 6-bit input and a 4-bit output
- The 48-bit data from the second operation is divided into eight 6-bit chunks, and each chunk is fed into a box.
- The result of each box is a 4-bit chunk; when these are combined the result is a 32-bit text.
- The substitution in each box follows a pre-determined rule based on a 4-row by 16-column table.

- The combination of bits 1 and 6 of the input defines one of four rows;
- the combination of bits 2 through 5 defines one of the sixteen columns
- Each S-box has its own table (8-tables)

	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

S-  
box1

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

S-  
box2

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

S-  
box3

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

S-  
box4

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

S-  
box5

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

S-  
box6

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

S-  
box7

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

S-  
box8

## Example 1

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

If we write first and sixth bit together we get  $(11)_2$  in binary.

11 in binary is 3 in decimal.

The remaining bits are  $(0001)_2$  which is 1 in decimal.

Hence, we look in Row 3, and Column 1 in S-box1 table.

$(3,1) = 12 = (1100)_2 \Rightarrow$  Output is 1100.



## Example 2

The input to S-box 8 is 000000. What is the output?

If we write first and sixth bit together, we get 00 in binary, which is 0 in decimal.

The remaining bits are  $(0000)_2$ , which is 0 in decimal.

Hence, we look in Row 0, and Column 0 in S-box8 table.

$(0,0) = 13 = (1101)_2 \Rightarrow$  Output is 1101

Step 4: 32 bit output from step 3 is permuted to create 32 bits.

- P-box or permutation box is also given which simply does permutation of input bits.
- Straight permutation table

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

- Thus,  $F$  (function) creates 32 bits.
- The output 32 bits from  $F$  are ex-ored with  $L$  to create the right half of output.
  - $L_i = R_{i-1}$ , Left output of Round  $i$  is same as right input to round  $i-1$ .
  - $R_i = L_{i-1} \oplus F(R_{i-1}, k_i)$

- This process is repeated 16 times so that resultant cipher text can't be crypt-analyzed easily.
- In round structure IP: Initial permutation, FP: Final permutation are just 64 bit permutations of the input 64 bits.
- FP and IP are inverses of each other. ( $FP = IP^{-1}$ )
- FP and IP have no cryptography significance in DES.

# Initial and Final permutation tables

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

## Example 1

Find the output of the final permutation box when the input is given in hexadecimal as: 0x000000080000000002

Represent hex in binary and find 1s

0000 0000 0000 0000 0000 0000 1000 00000000000000000000 0000 0000 0000 0010

Only bit 25 and bit 63 are 1s; the other bits are 0s.

In the final permutation, bit 25 becomes bit 64 and bit 63 becomes bit 15.

The result is 0x0002 0000 0000 0001

## Example 2

Prove that the initial and final permutations are the inverse of each other by finding the output of the initial permutation if the input is 0x0002 0000 0000 0001.

0000 0000 0000 0010 0000 0000 0000 0000 0000 0000  
0000 0000 0000 0000 0000 0001

- The input has only two 1s; the output must also have only two 1s.
- Using table, we can find the output related to these two bits.
- Bit 15 in the input becomes bit 63 in the output.
- Bit 64 in the input becomes bit 25 in the output.
- So the output has only two 1s, bit 25 and bit 63.
- The result in hexadecimal is `0x0000008000000002`



# Network and Information Security

## Lecture 15

B.Tech. Computer Engineering  
Sem. VI.

Prof. Mrudang T. Mehta  
Associate Professor  
Computer Engineering Department  
Faculty of Technology,  
Dharmsinh Desai University, Nadiad

DES round contains 3 types of elements

- Self invertible like  $\oplus$

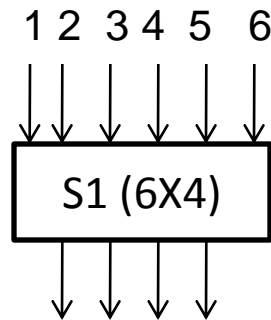
$$z = x \oplus y$$

$$x = z \oplus y$$

$$y = z \oplus x$$

- Invertible like P-Box
- Non-invertible like S-box (S1 to s8)

# S-boxes are non-invertible



	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

S-box1

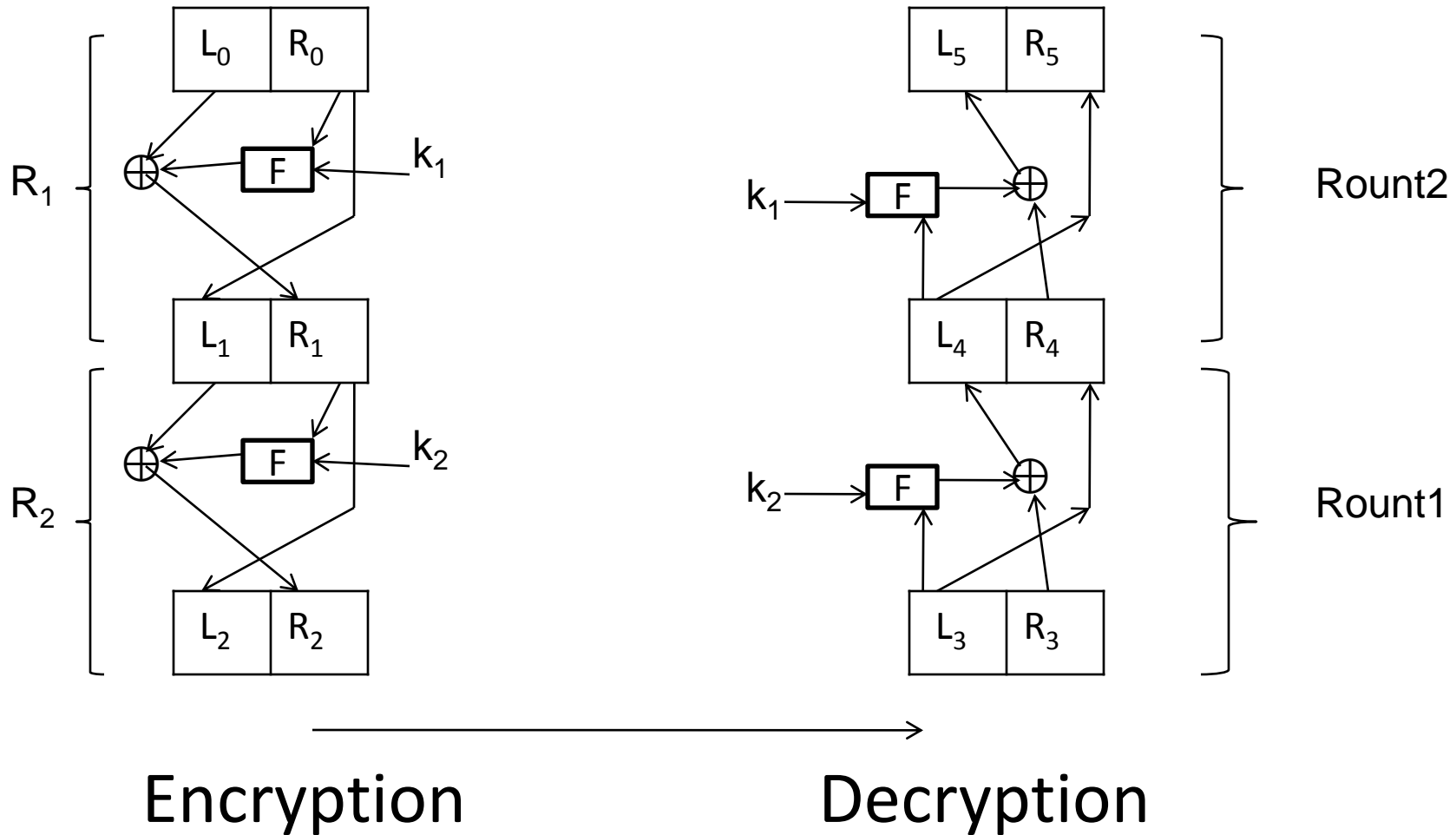
010110  $\longrightarrow (12)_d$  [For 12 we can get multiple input values]

110010  $\longrightarrow (12)_d$

- Round function contains element/components
  - Self invertible
  - Invertible
  - Non invertible
- Feistel structure cipher
  - If there are invertible and non-invertible elements present in the structure (with decryption possible)
  - Example- DES (in Function F, both invertible and non-invertible elements present)

$L_i = R_{i-1},$   
 $R_i = L_{i-1} \oplus F(R_{i-1}, k_i)$   
 $F$  is Round function

How decryption is possible with non-invertible element present?



Show: If  $L_2 = L_3$  and  $R_2 = R_3$  then

(1)  $L_4 = L_1$ ,  $R_4 = R_1$

(2)  $L_5 = L_0$ ,  $R_5 = R_0$

If  $L_2, R_2$  is received without error, then using above decryption we can decrypt.

Proof:

From the encryption part, we can write the following equations.

$$L_1 = R_0 \text{ -----(1)}$$

$$R_1 = L_0 \oplus F(R_0, k_1) \text{ -----(2)}$$

$$L_2 = R_1 \text{ -----(3)}$$

$$R_2 = L_1 \oplus F(R_1, k_2) \text{ -----(4)}$$

- From the decryption design we can write following:

$$R_4 = L_3 \text{ -----(5)}$$

$$L_4 = R_3 \oplus F(L_3, k_2) \text{ -----(6)}$$

$$R_5 = L_4 \text{ -----(7)}$$

$$L_5 = R_4 \oplus F(L_4, k_1) \text{ -----(8)}$$



If  $L_2 = L_3$  and  $R_2 = R_3$  then  $R_4 = R_1$  ,  $L_4 = L_1$

$$\begin{aligned}
 \text{L H S} &= R_4 \\
 &= L_3 \text{ ( ....from (5) )} \\
 &= L_2 \text{ (Given)} \\
 &= R_1 \text{ (....from (3) )} \\
 &= \text{R H S}
 \end{aligned}$$

$$R_4 = R_1 \text{ .....(9)}$$

$$L_4 = L_1 \text{ .....(10)}$$

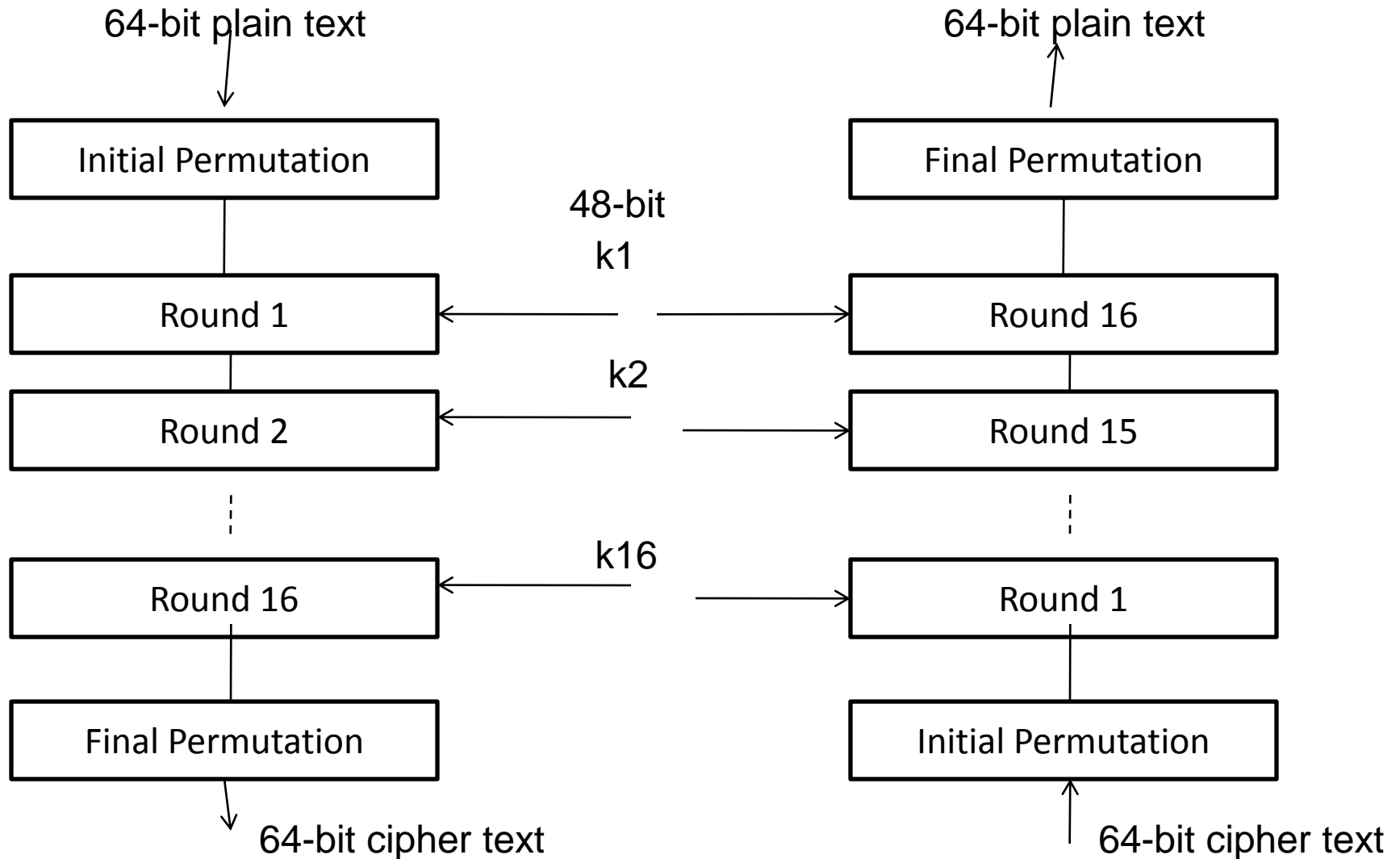
$$\begin{aligned}
 \text{L H S} &= L_4 \\
 &= R_3 \oplus F(L_3, k_2) \text{ (.. from (6) )} \\
 &= R_2 \oplus F(L_2, k_2) \text{ (Given)} \\
 &= L_1 \oplus F(R_1, k_2) \oplus F(L_2, k_2) \\
 &\quad \text{(...from(4))} \\
 &= L_1 \oplus F(R_1, k_2) \oplus F(R_1, k_2) \\
 &\quad \text{(...from(3))} \\
 &= L_1 \oplus 0 \quad (a \oplus a = 0) \\
 &= L_1 \quad (a \oplus 0 = a) \\
 &= \text{R.H.S.}
 \end{aligned}$$

If  $L_2 = L_3$  and  $R_2 = R_3$  then  $R_5 = R_0$  ,  $L_5 = L_0$

$$\begin{aligned}
 LHS &= R_5 \\
 &= L_4 \text{ ( ....from (7) )} \\
 &= L_1 \text{ (...from 10)} \\
 &= R_0 \text{ (....from (1) )} \\
 &= RHS
 \end{aligned}$$

$$\begin{aligned}
 LHS &= L_5 \\
 &= R_4 \oplus F(L_4, k_1) \text{ (.. from (8) )} \\
 &= R_1 \oplus F(L_1, k_1) \text{ (from 9 ,10)} \\
 &= L_0 \oplus F(R_0, k_1) \oplus F(L_1, k_1) \\
 &\quad \text{(...from(2))} \\
 &= L_0 \oplus F(R_0, k_1) \oplus F(R_0, k_1) \\
 &\quad \text{(...from(1))} \\
 &= L_0 \oplus 0 \quad (a \oplus a = 0) \\
 &= L_0 \quad (a \oplus 0 = a) \\
 &= R.H.S.
 \end{aligned}$$

# DES cipher and reverse cipher



# Alternative approach

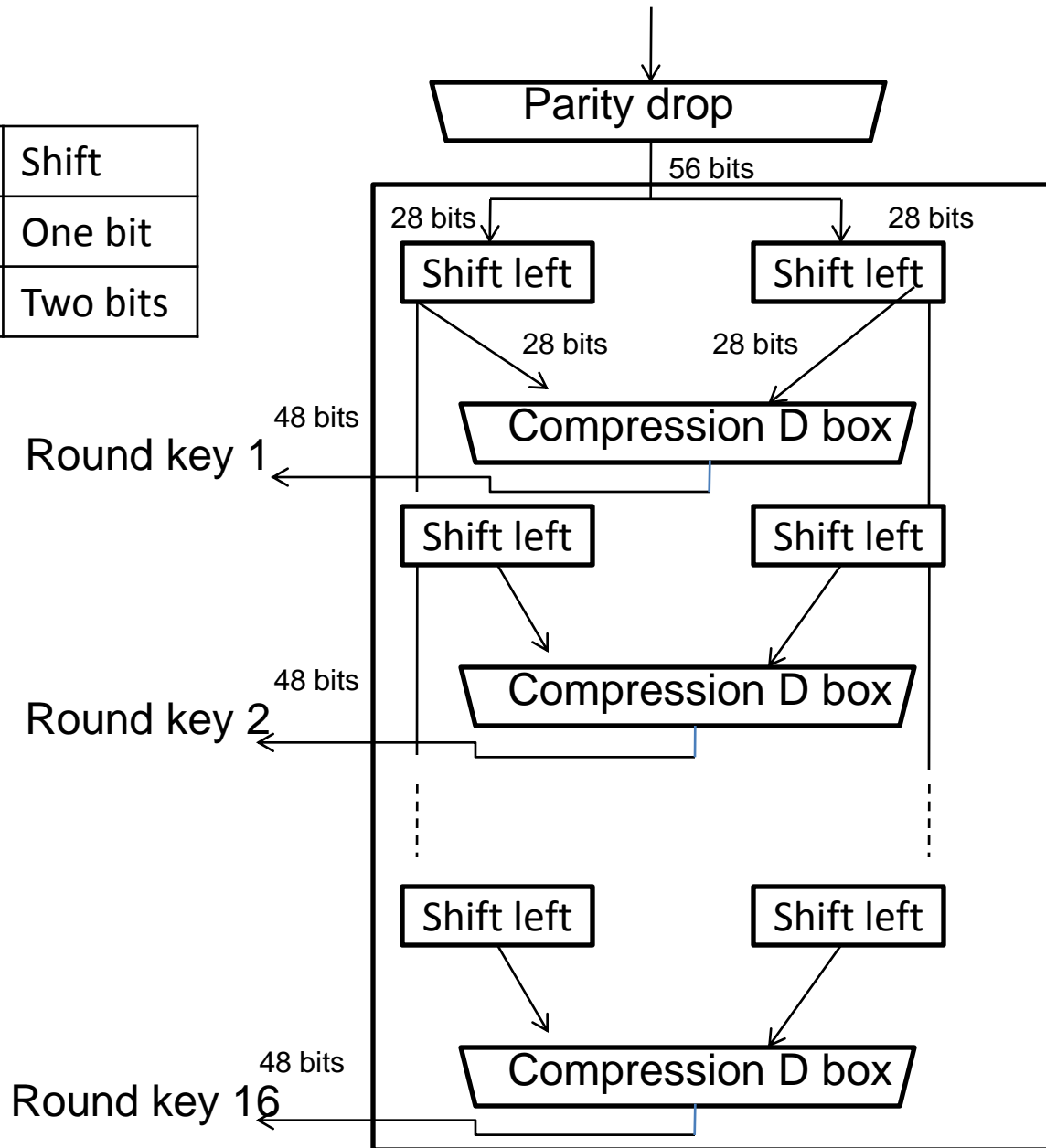
- In the first approach round 16 is different from other rounds, there is no swapper in this round.
- This is needed to make the last mixer in the cipher and the first mixer in the reverse cipher aligned.
- We can make all 16 rounds the same by one swapper to the 16<sup>th</sup> round and add an extra swapper after that (two swapper cancels the effect of each other).

# Key Generation

- The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key.
- However, the cipher key is normally given as 64-bit in which 8 extra bits are the parity bits, which are dropped before the actual key-generation process.

Key with parity bits (64 bits)

Rounds	Shift
1,2,9,16	One bit
Others	Two bits



- Parity drop
- The preprocess before key expansion is a compression transposition step that we call parity bit drop.
- It drops the parity bits (8,16,24,32,40,48,56,64) from the 64-bit key and permutes the rest of the bits according to table.
- The remaining 56-bit value is the actual cipher key which is used to generate round keys.

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

Parity bit drop table



- Shift left

- After the straight permutation, the key is divided into two 28-bit parts.
- Each part is shifted left (circular shift) one or two bits.
- In round 1,2,9 and 16, shifting is one bit; in the other rounds it is two bits.
- The two parts are then combined to form a 56-bit part.

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

- Compression Permutation (Compression D-box)

The compression D-box changes the 58-bits to 48-bits, which are used as a key for a round.

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

Note: (In a book, P-box is named as D-box, Both are same)

- Example 1 We choose a random plaintext block and a random key, and determine what the ciphertext block would be (all in hexadecimal):

Plain text: 123456ABCD132536
Cipher text: C0B7A8D05F3A829C

Key: AABB09182736CCDD
-----------------------

Plain text: 123456ABCD132536			
After initial permutation: 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

Round 5	236779C2	A15A4B87	69A629FEC913
Round 6	A15A4B87	2E8F9C65	C1948E87475E
Round 7	2E8F9C65	A9FC20A3	708AD2DDB3C0
Round 8	A9FC20A3	308BEE97	34F822F0C66D
Round9	308BEE97	10AF9037	84BB4473DCCC
Round10	10AF9037	6CA6CB20	02765708B5BF
Round11	6CA6CB20	FF3C485F	6D5560AF7CA5
Round12	FF3C485F	22A5963B	C2C1E96A4BF3
Round13	22A5963B	387CCDAA	99C31397C91F
Round14	387CCDAA	BD2DD2AB	251B8BC717D0
Round15	BD2DD2AB	CF26B472	3330C5D9A36D
Round16	19BA9212	CF26B472	181C5D75C66D
After combination: 19BA9212CF26B472			
Cipher text: C0B7A8D05F3A829C (after final permutation)			