Data Encryption Standard (DES)

G.P. Biswas

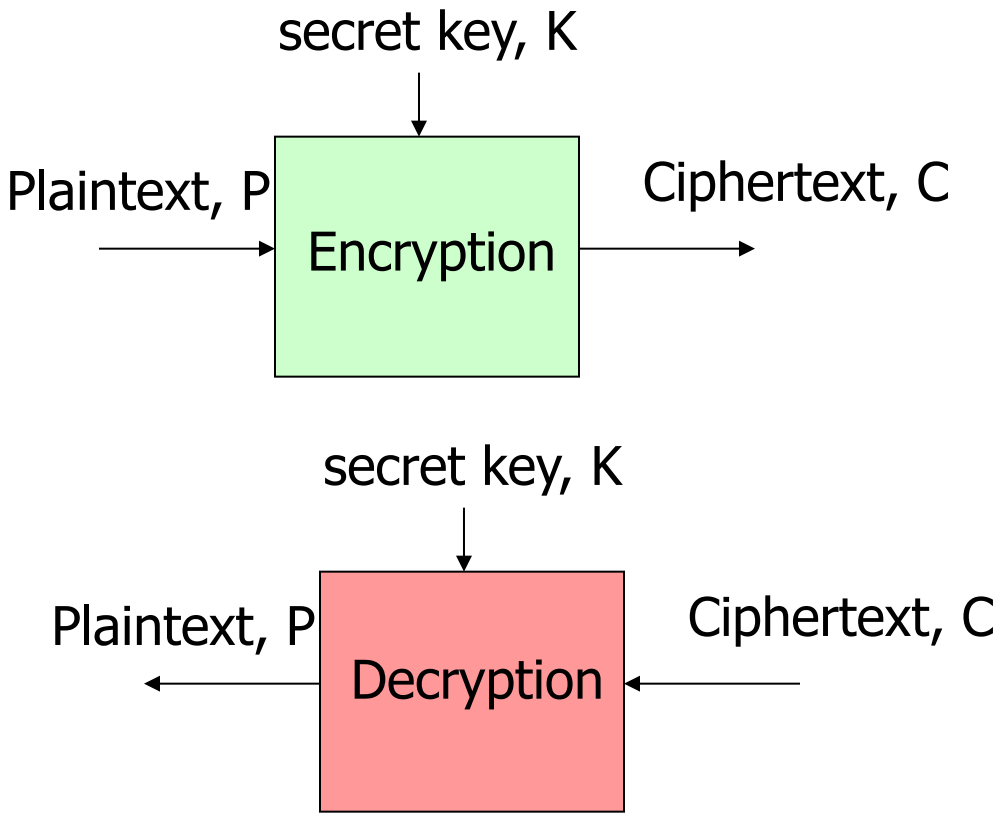
Prof/CSE, IIT-ISM, Dhanbad

Outline

- Basic idea of block ciphers
- Structure of Feistel block Ciphers
- Design process of DES
- Differential Cryptanalysis
- Alternatives to DES

Block Ciphers

- Block cipher encrypts an *n*-bits block of plaint text and/or decrypts an *n*-bits block of ciphertext.



Block Cipher Principles

- It encrypts an entire block of plain text bits at a time.
- The encryption of any plaintext bit in a given block depends on every other plain text bit in the same block.
- To allow unique decryption, the encryption function must be one-to-one (i.e., invertible).
- In reversible mapping the number of different transformations is $2^n!$ for n-bits plain text.

Contd…

- A block ciphers of *n*-bits is too small may be vulnerable to attacks based on statistical analysis.
- Even choosing too large a value for block cipher of *n*-bits may create difficulties as the complexity of implementation of many ciphers grows rapidly with block size.
- The most of modern block ciphers like DES is based on **Feistel Cipher Structure**

Contd.

- Feistel cipher is the class of product ciphers.
- It consists of both invertible and non-invertible components.
- It also provides *confusion* and *diffusion* of message .
- The non-Feistel ciphers only considers invertible components. (Example AES)

Diffusion and Confusion

- Ciphertext will hide the statistical properties of original message. An one-time pad does this.
- The idea of diffusion is to hide the statistical relationship between the plaintext and ciphertext. This is achieved by having each plaintext bit/digit affects the value of many ciphertext bits/digits.
- An example of diffusion is to encrypt a message M = m1, m2, m3, … of characters with averaging operation

$$y_n = \sum_{i=1}^{k} m_{n+i} (mod\ 26)$$

Adding k successive letters to get a ciphertex letter y_n.

- The aim of this process is to make the system as complex as possible in order to thwart attempts to deduce the key.

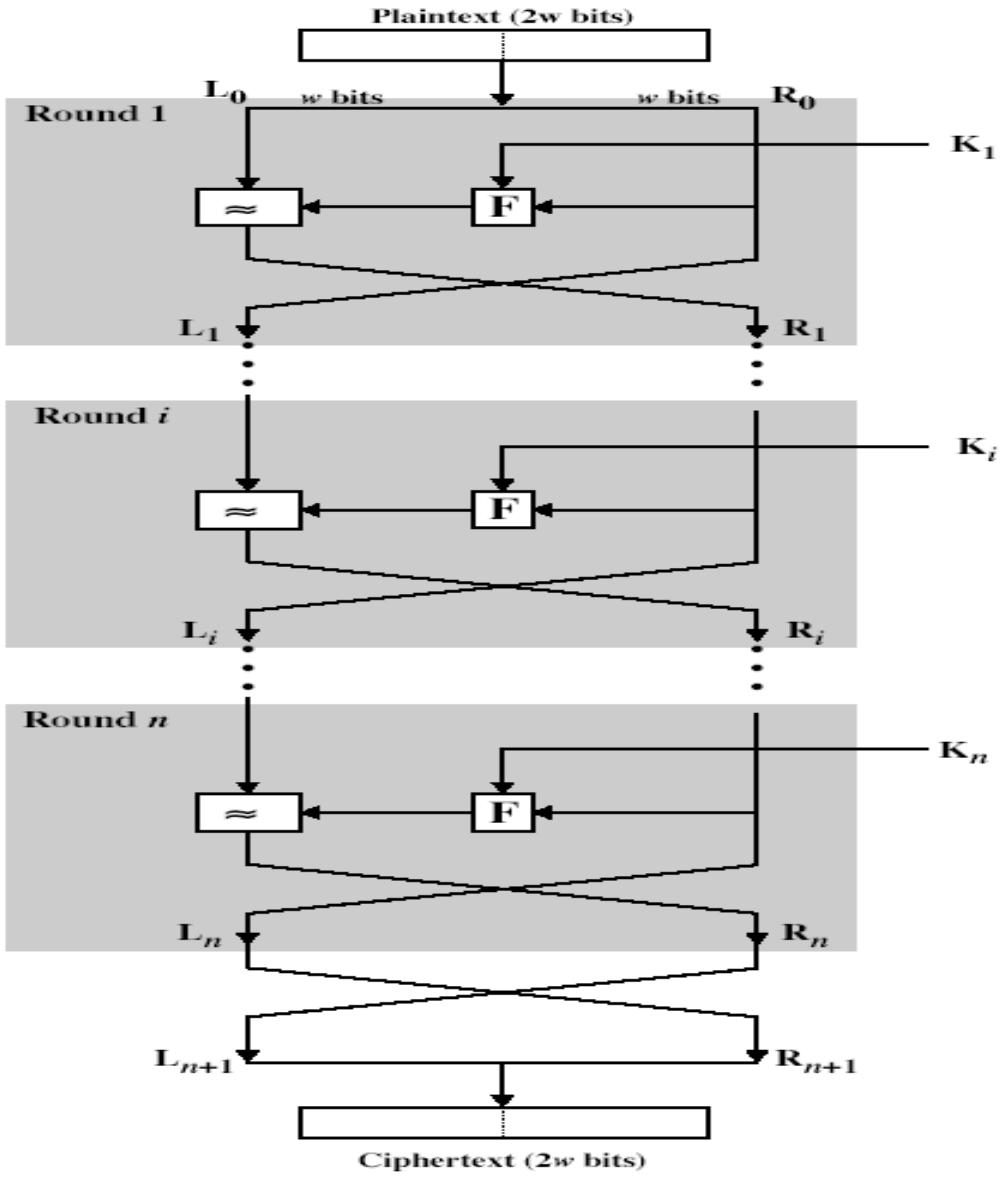
Contd…

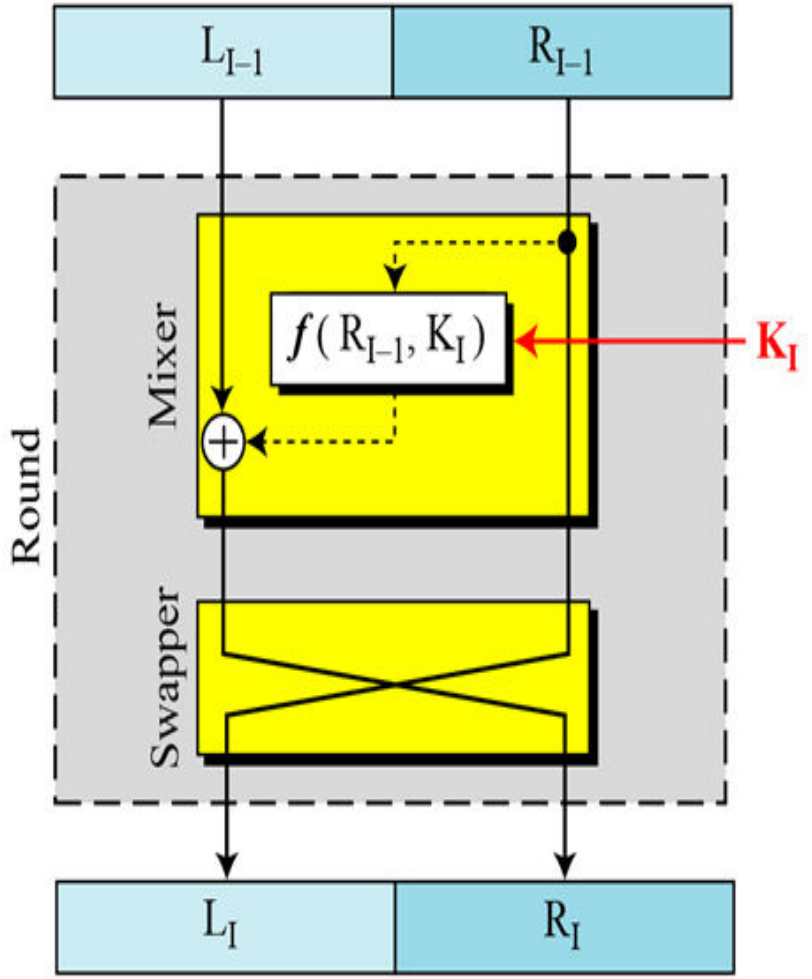
- The confusion seeks to make the relationship between the statistics of the ciphertext and the value of the encryption key as complex as possible, again to thwart attempts to discover the key.
- This is achieved by the use of a complex substitution algorithm.

Feistel Cipher Structure

- Feistel proposed a suitable structure which is based on Shannon's S-P network structure.
- Essentially the same hardware or software is used for both encryption and decryption, with just a slight change in how the keys are used.
- It uses both invertible and non-invertible components.

Feistel Cipher Structure





- The encryption and decryption processes are:
  - $LE_i = RE_{i-1}$
  - $RE_i = LE_{i-1} \oplus F(RE_{i-1}, K_i)$

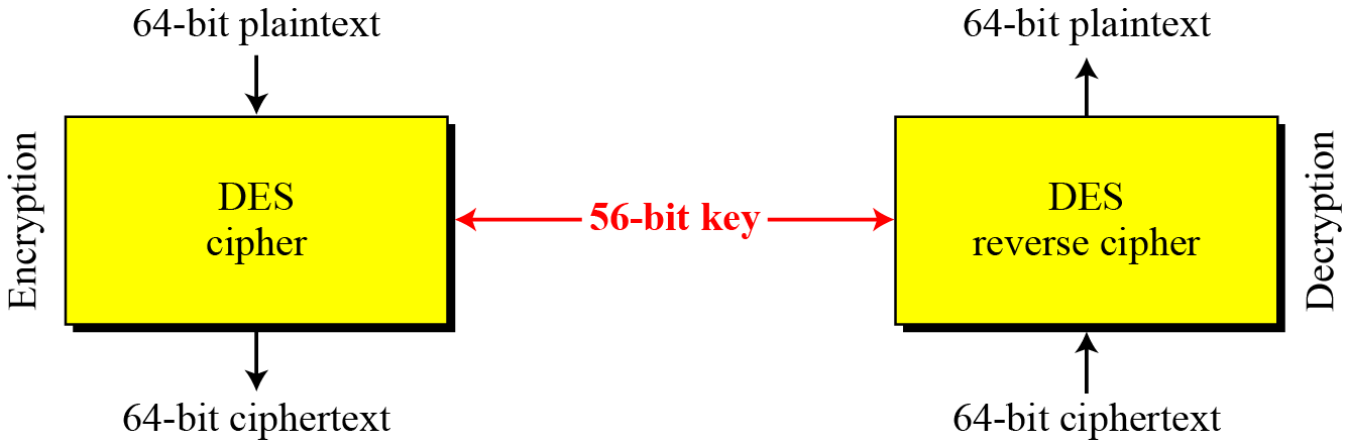
  - $RE_{i-1} = LE_i$
  - $LE_{i-1} = RE_i \oplus F(RE_{i-1}, K_i) = RE_i \oplus F(LE_i, K_i)$

Feistel Cipher Design Principles

- **The exact realization of Feistel cipher depends on the choice of the following parameters and design parameters:**
- **Block size**
  - increasing size improves security, but reduces the speed of encryption/decryption process.
- **Key size**
  - increasing size improves security, makes exhaustive key searching harder, but may decrease the speed of encryption/decryption process.
- **Number of rounds**
  - Multiple rounds offer increasing security.
- **Subkey generation**
  - Greater complexity can make analysis harder, but slows cipher
- **Round function**
  - Greater complexity can make analysis harder, but slows cipher
- **Fast software en/decryption**
  - are more recent concerns for practical use and testing
- Easy of analysis: Although E/D algorithms are complex, there is great benefit in making algorithm easy to analyse as cryptanalytic analysis develop a higher level assurance of its strength, DES does not have easily analysed functionality.

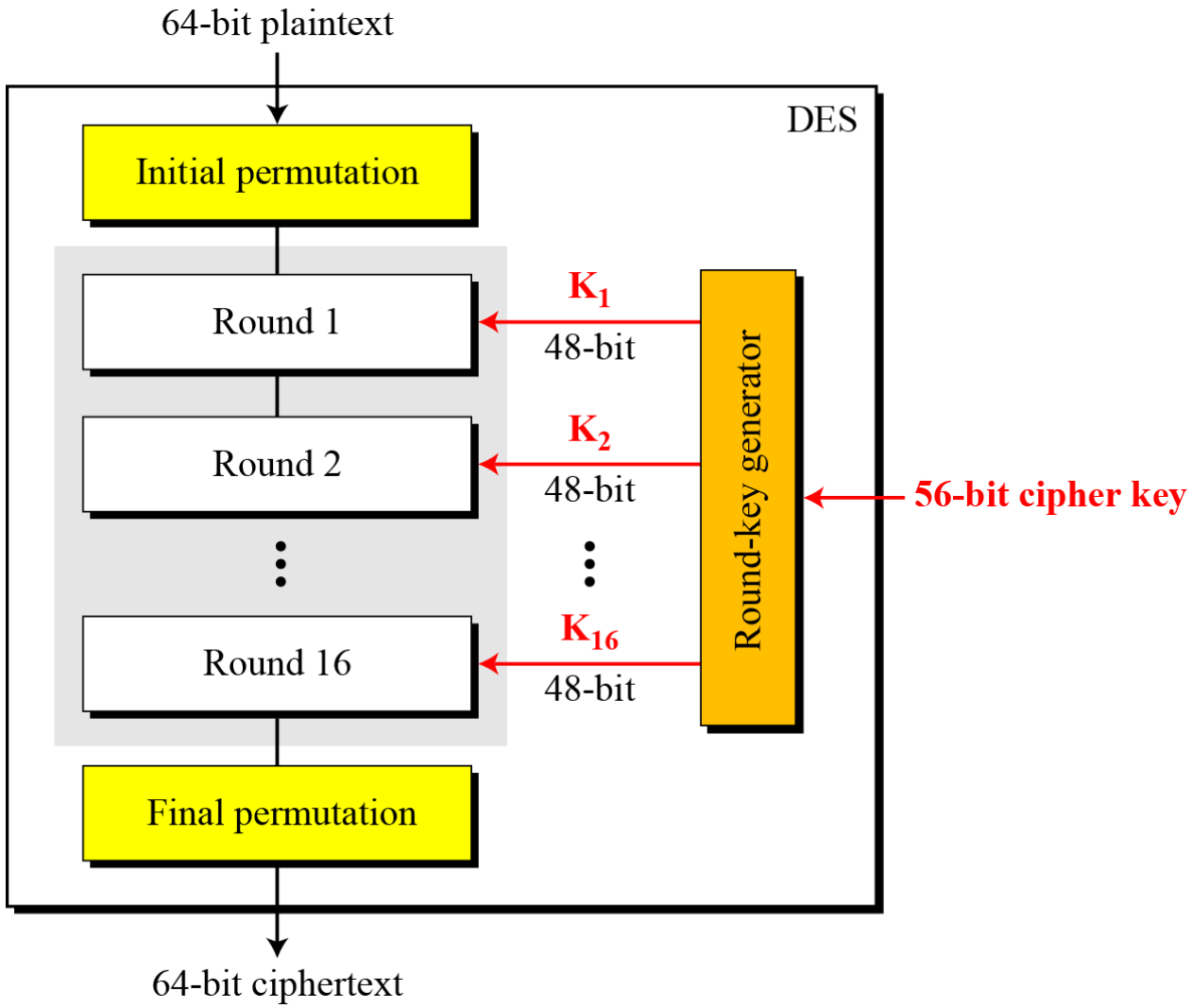
DES

- Nowadays, DES is not secure algorithm since the key space is too small (56-bit length)



- Its design principles have inspired many current ciphers.
- Studying of DES helps us to understand many other symmetric algorithms.

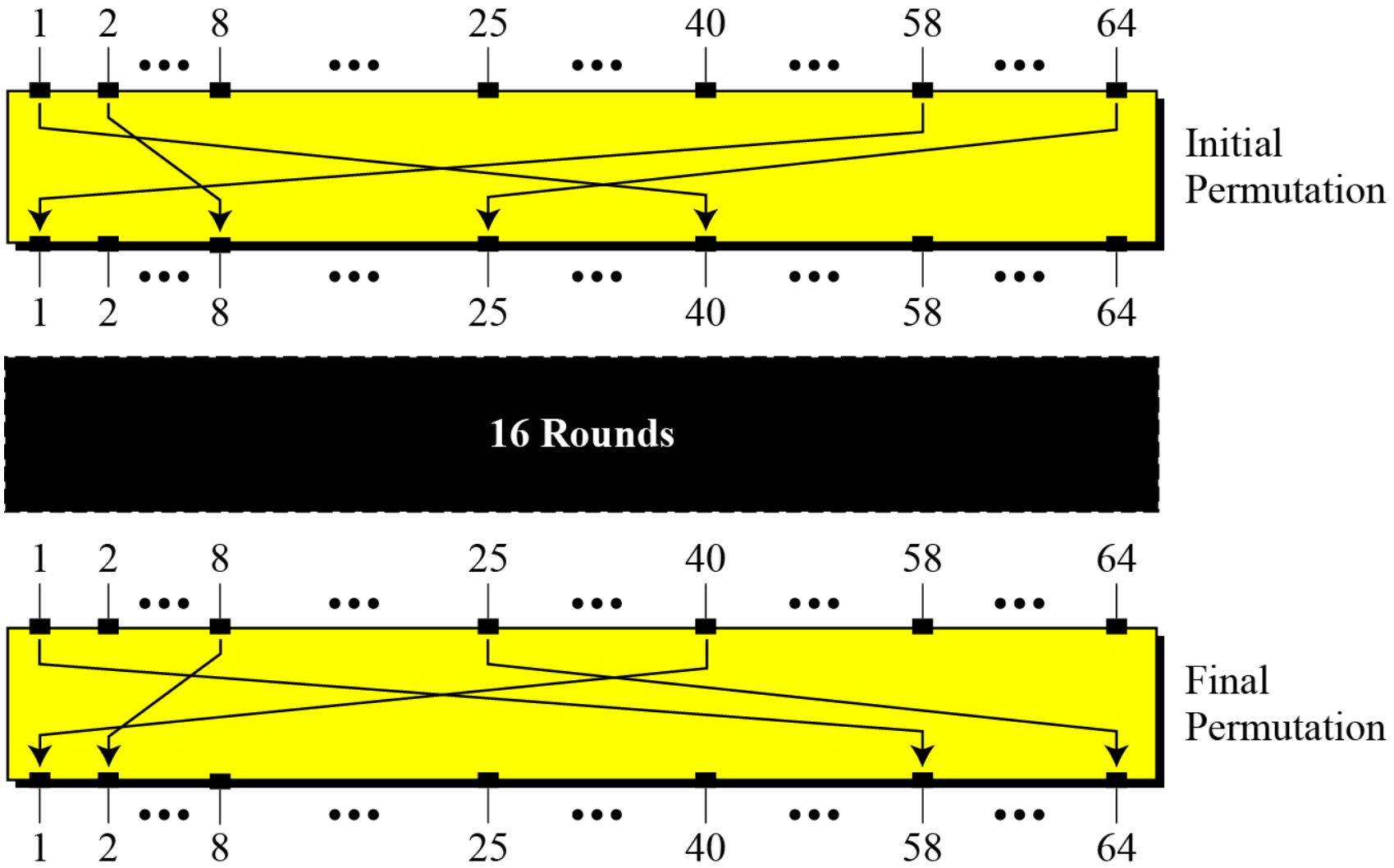
General Structures of DES



Details of each Steps

- **Initial and Final Permutations**
- Rounds
- DES Function
  - Expansion D-box
  - S-box Operation
- Key Schedule

Initial and Final Permutations



Initial and final permutation tables

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

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

- Each entry in the permutation table indicates the position of a numbered input bit in the output.

Contd.

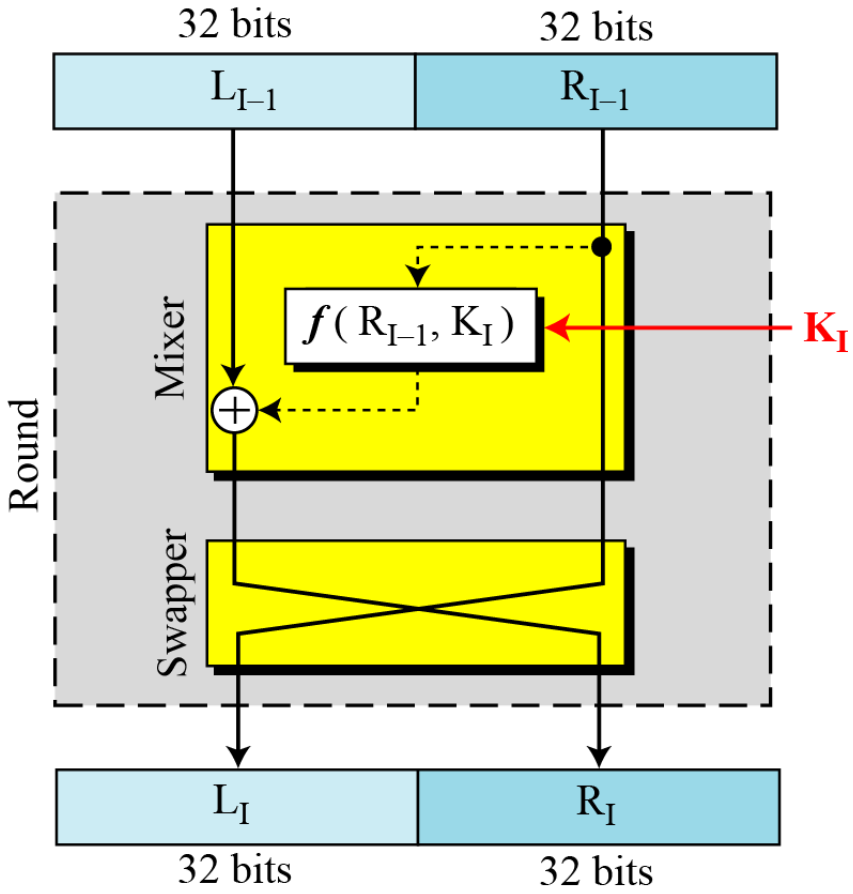
- These two permutations have no cryptographic significance in DES other than to prevent S/W attack.
- Both permutations are keyless and predetermined.

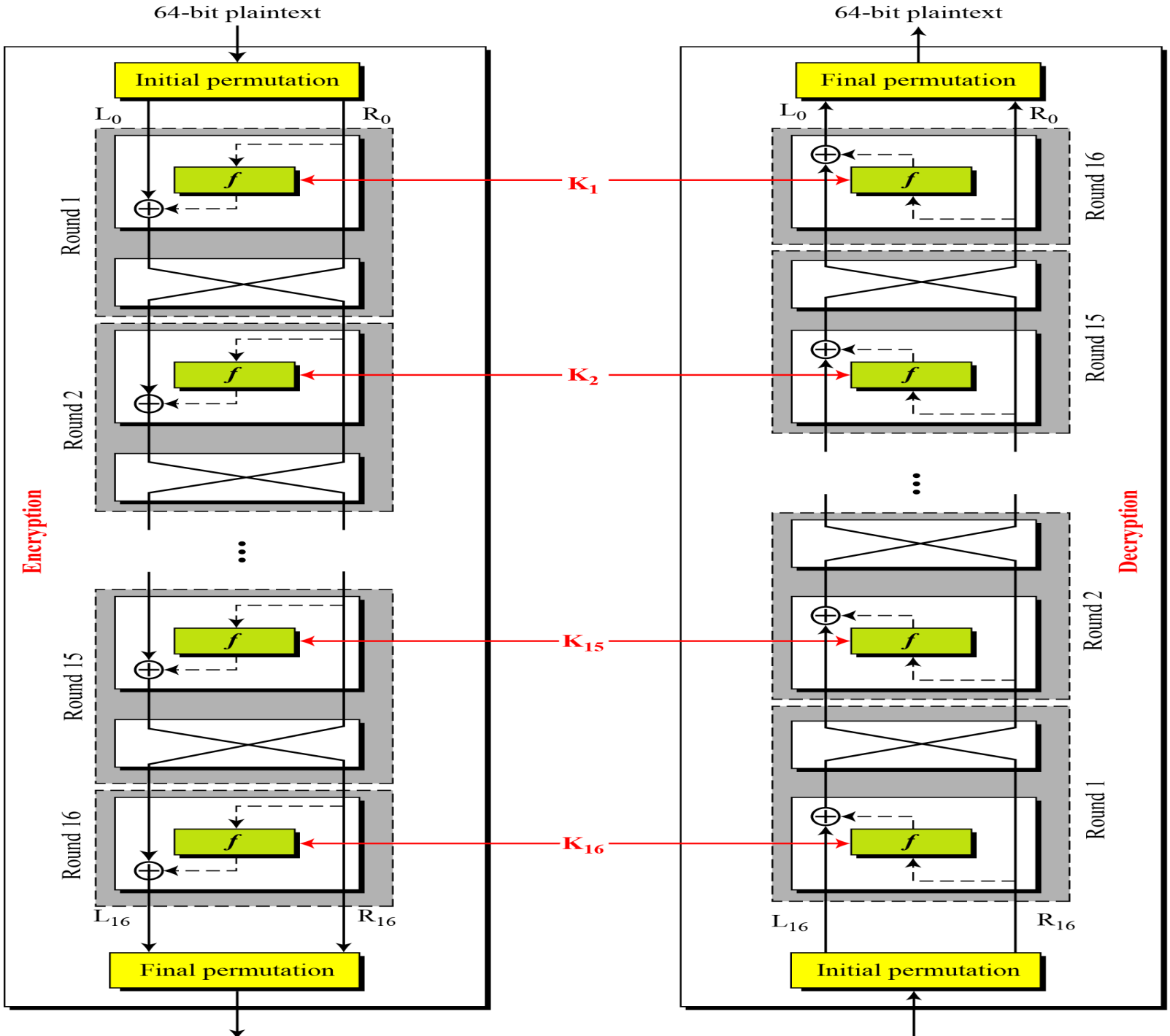
Details of each Steps

- Initial and Final Permutations
- **Rounds**
- DES Function
  - Expansion D-box
  - S-box operation
- Key Schedule

Rounds

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



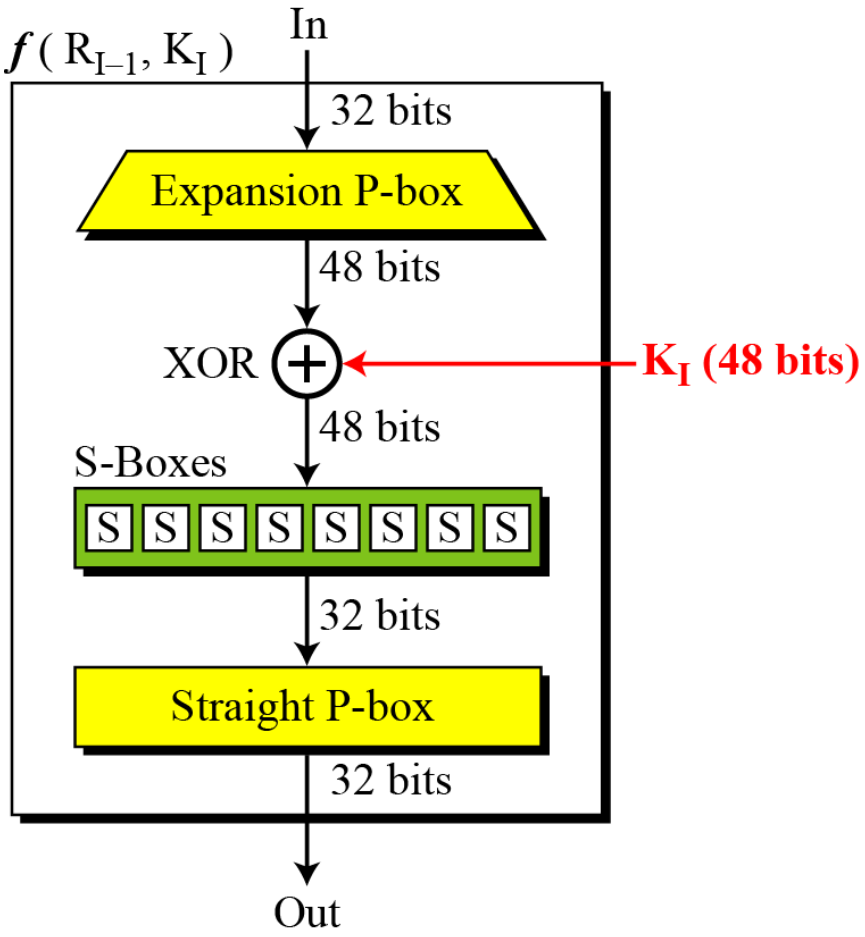

64-bit plaintext
Initial permutation
L_0
R_0
Round 1
Round 2
Round 15
Round 16
L_{16}
R_{16}
Final permutation
Encryption
K_1
K_2
K_{15}
K_{16}
64-bit plaintext
Final permutation
L_0
R_0
Round 16
Round 15
Round 2
Round 1
L_{16}
R_{16}
Initial permutation
Decryption

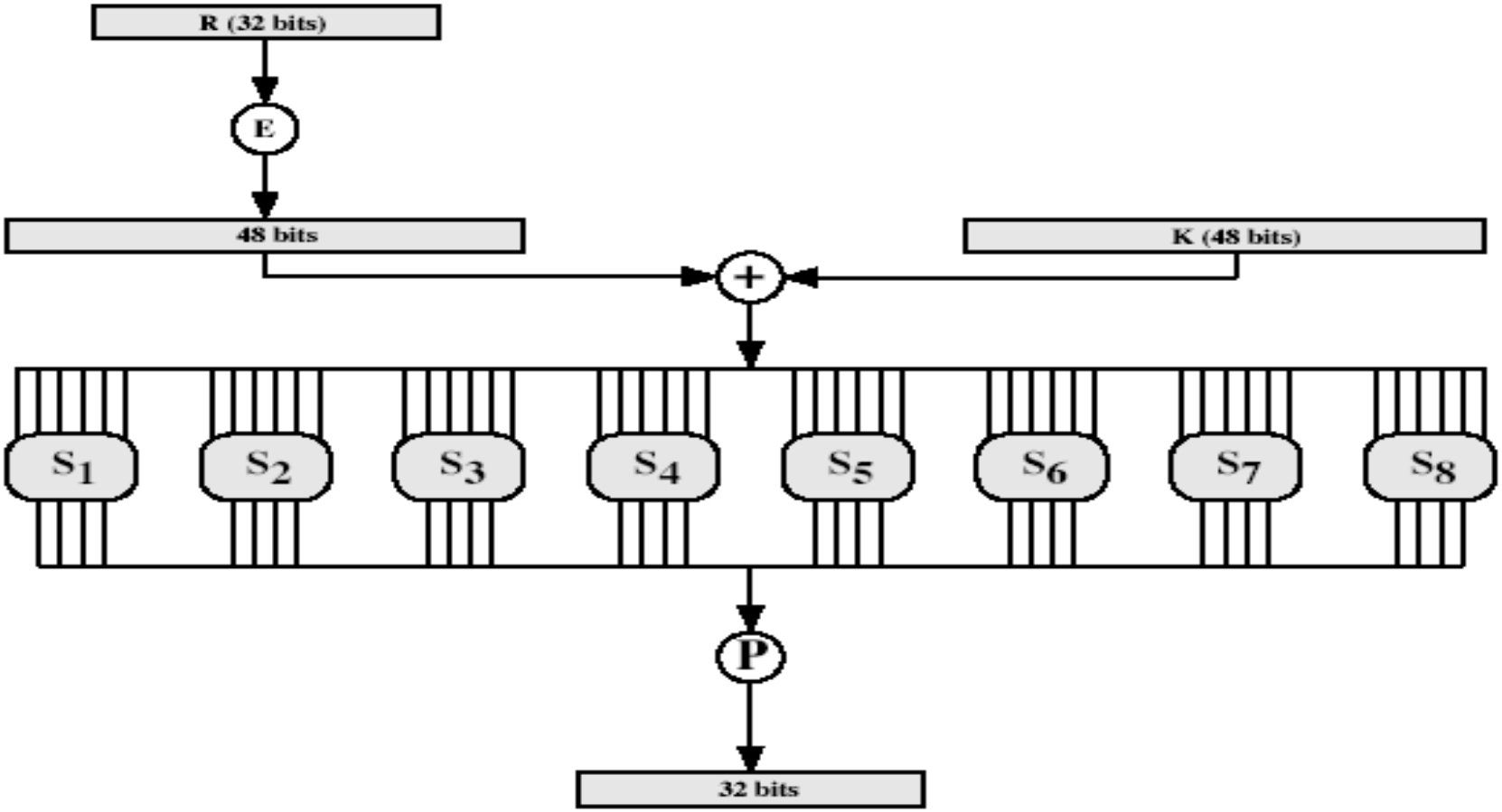
Details of each Steps

- Initial and Final Permutations
- Rounds
- **DES Function**
  - **Expansion D-box**
  - **S-box operation**
- Key Schedule

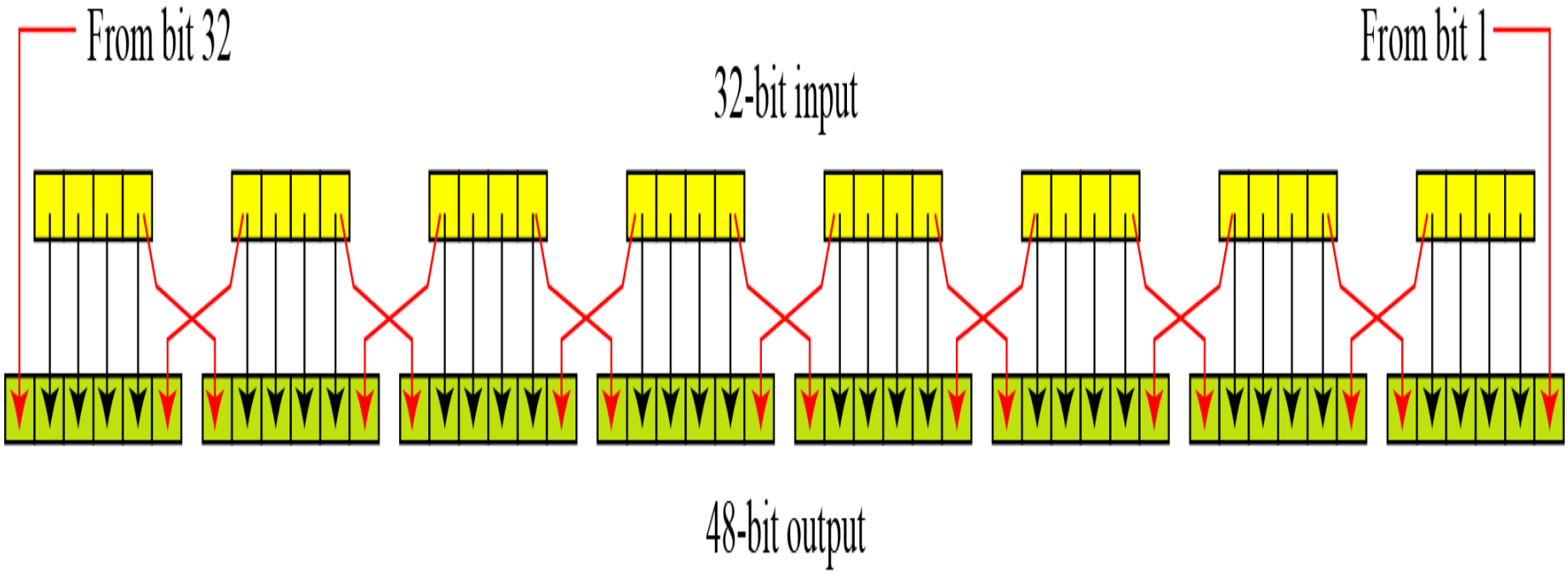
DES Function

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




R (32 bits)
E
48 bits
K (48 bits)
+
S1
S2
S3
S4
S5
S6
S7
S8
P
32 bits

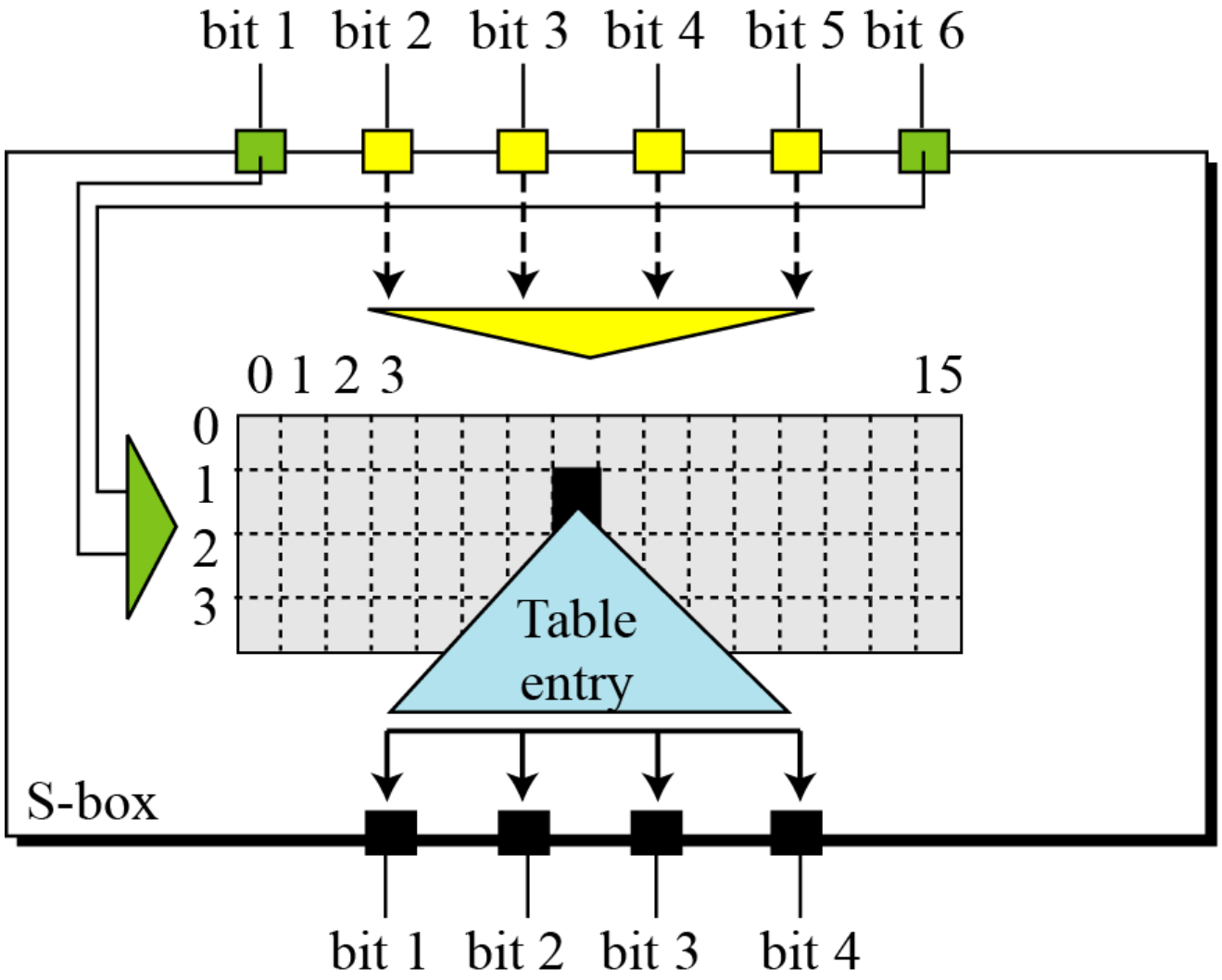
Expansion permutation



XOR-operation

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.

S-box


bit 1
bit 2
bit 3
bit 4
bit 5
bit 6
0 1 2 3
15
0
1
2
3
Table
entry
S-box
bit 1
bit 2
bit 3
bit 4

S box-1

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

Contd…

- S-box is a lookup table that maps a 6-bits input to a 4-bits output.
- Entry of each table represents the decimal notation of a 4-bit value.
- The S-boxes are the core of DES in terms of cryptographic strength.
- These are the nonlinear element in the algorithm and provides confusion.

Details of each Steps

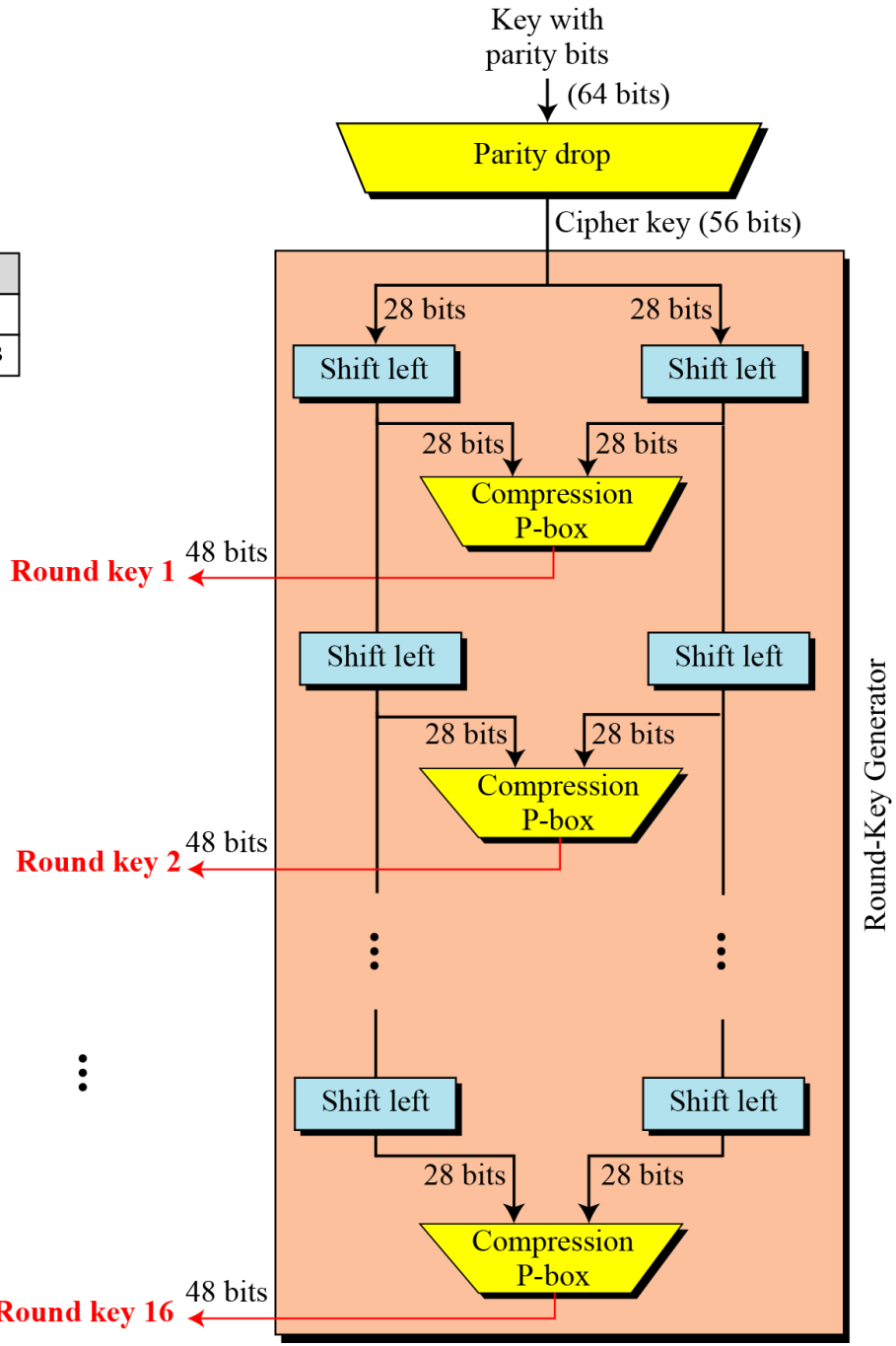
- Initial and Final Permutations
- Rounds
- DES Function
  - Expansion D-box
  - S-box operation
- **Key Schedule**

Key Schedule

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

Shifting

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



Key-compression table

- P-Box

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 Analysis

- Two desired properties of a block cipher are:
  - Avalanche effect: A small change in plaintext should create a significant change in ciphertext. DES has this property: Highest change in 6^{th} round 1 bit to 33 bits.
  - Completeness effect: Each bit in ciphertext needs to depend on many bits in the plaintext
- Design criteria: DES was designed by IBM in 1994 and some of its design criteria are given below:
- S-Boxes:
  - Entries of each row are permutation of values 0 to 15
  - S-boxes are non-linear
  - A single bit change in input, two or more bits change in output.
  - If two inputs differ by bits 3 and 4, output must differ at least by two bits.
  - If two inputs differ by bits 1 and 2, and are the same in bits 5 and 6, two output must be different.
  - If a single bit is held constant (0 or 1) and other bits are changed randomly, the differences between number of 0's and 1's are minimized.

- Number of rounds: DES uses 16 round of Feistel cipher. It has been proved that after 8 rounds, each ciphertext is a function of every plaintext bit and every key-bit (ciphertext is a random function of plaintext and key). So 8 rounds should enough, however, experiments have found that that DES with less than 16 rounds are more vulnerable to known plaintext attack than brute-force attack.

DES Weaknesses

- S-boxes:
  - In S-box 4, the last 3 output bits can be derived in the same way as the first output bit by complementing some of the input bits.
  - Two specifically chosen inputs to an S-box can create the same output.
  - It is possible to obtain the same output in a single round by changing the bits in only 3 neighboring S-boxes.

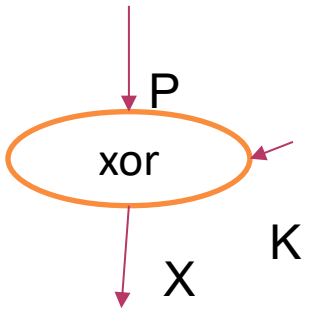
- Weakness in cipher key:
  - Key size: A brute-force attack requires to check 2^56 keys, which is insufficient in present-day technology that can check 10^6 keys per sec.
  - Weak keys: Four out of 2^56 are weak as they comprise all 0's, all 1's, half 0,s and half 1's and half 1's and half 0's. Double encryptions or decryption using weak keys result original plaintext.
  - Semi weak keys: there are 6 key pairs called semi weak keys. A semi weak key creates only two different round keys and each of the repeated 8 times. In addition, the round keys created from each pair are the same with different orders. The semi weak keys inverses each other as E (K2, E(k1, P) = P
  - Possible weak keys: There are 48 such keys. A possible weak key that creates only 4 different round keys (16 round keys are divided into 4 groups having 4 equal round keys).
  - Key complement: Out of 2^56, half of them are complement of other half. An attacker only uses half of these keys to make a brute-force attack.
  - Key clustering: It refers to the situation in which two or more different keys create the same ciphertext from the same plaintext.

Differential Cryptanalysis

- Differential Cryptanalysis is a kind of **Chosen-plaintext** attack introduced by **Eli Biham and Adi Shamir.**
- Differential Cryptanalysis would be more efficient than exhaustively searching all possible keys if the algorithm used at most 15 rounds.
- This indicate that perhaps the designers of DES had been aware of this type of attack.

Idea of Differential Cryptanalysis

- Let P1 and P2 are two inputs to DES and the corresponding ciphertexts are C1 and C2 with the same key K. If we XORed (take difference) them, we have
  - C1 xor C2 = (P1 xor K) xor (P2 xor K)= P1 xor P2
  - The relation of C1 xor C2 = P1 xor P2 without key K is the basic idea of this attack
- For simplistic, consider 3-bit plaintext, and thus 3-bit key and consider a single S-box table as shown below: where P xor K = X
  - X: 000 001 010 011 100 101 110 111
  - C: 11 00 10 10 01 00 11 00



- If we take two inputs P1 and P2, so P1 xor P2 = X1 xor X2 or P1 xor P2 = C1 xor C2: 3-bit each.
- Thus, we form a table with P1 xor P2 as rows and C1 xor C2 as columns.
- There are 8 cases for each xor in the input and there are 4 cases for C1 xor C2 as 00, 01, 10 and 11

P1 xor P2 ↓	C1 xor C2			
	00	01	10	11
000	8 (1)	-	-	-
001	2 (.25)	2 (.25)	-	4 (.5)
010	2 (.25)	2 (.25)	4 (.5)	-
011	-	4 (.5)	2 (.25)	2 (.25)
100	2 (.25)	2 (.25)	4 (.5)	-
101	-	4 (.5)	2 (.25)	2 (.25)
110	4 (.5)	-	2 (.25)	2 (.25)
111	-	-	4 (.50)	4 (.50)

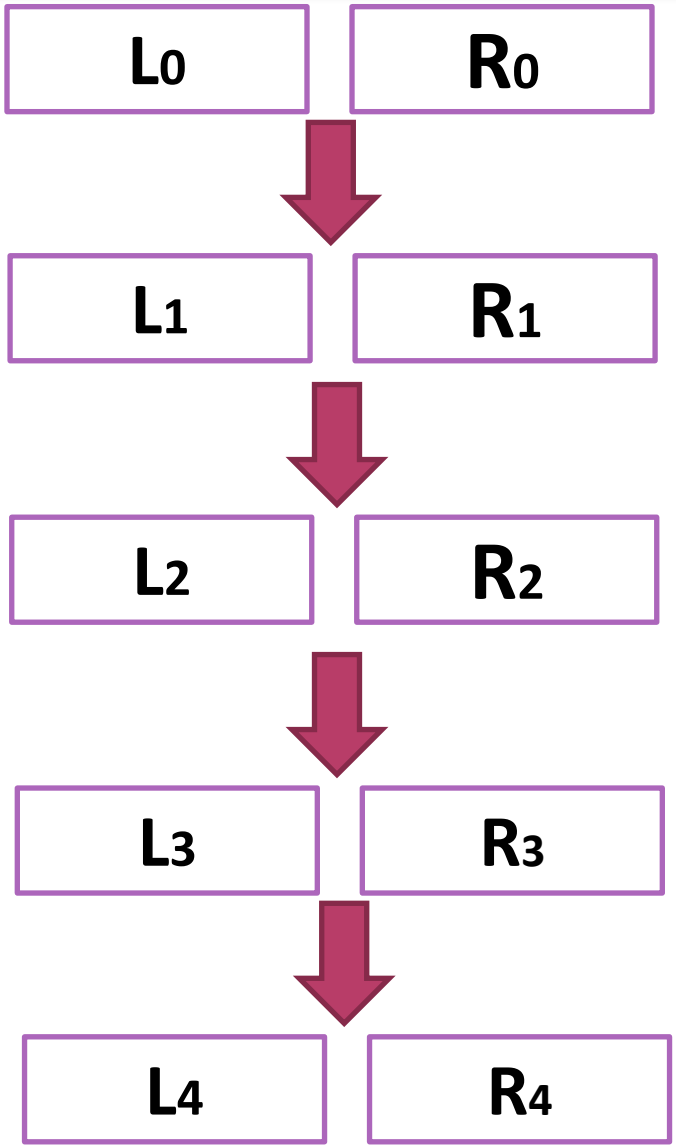
- Table shows that the probabilities are not uniformly distributed, because of weakness of S-box and hence, the chosen plaintext attack (and Differential cryptanalysis) is made.
- An attacker chooses plaintexts that have highest probability in the table.

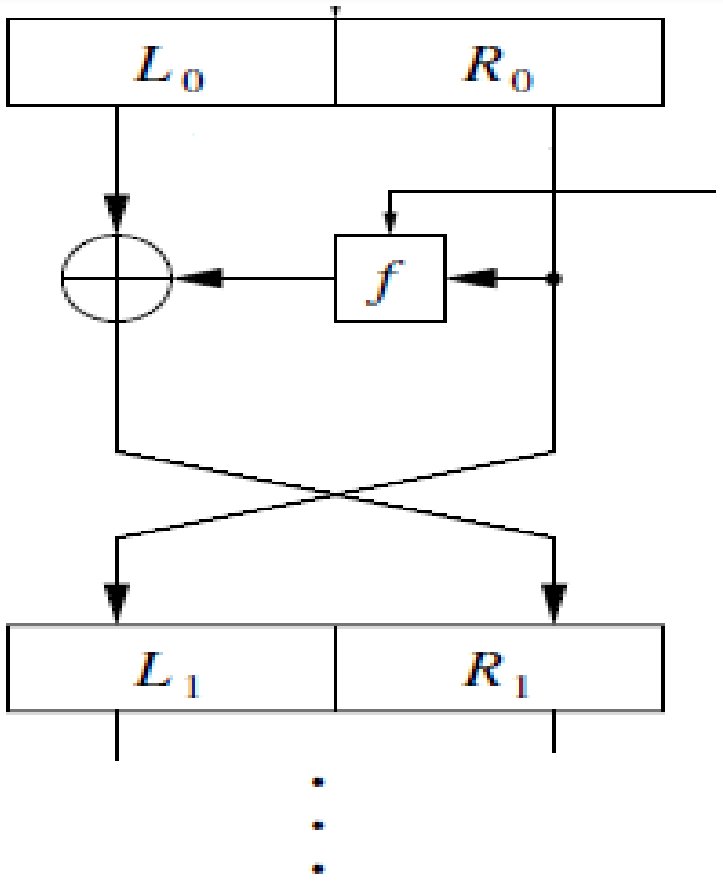
- Consider the case P1 xor P2 = 001 and C1 xor C2 = 11 with probability 0.50.
  - Thus we may have, P1 = 010 & P2 = 011, and C1 = 00 & C2 = 11. Thus, using P1 and C1:
  - For C1 = 00, X1 = 001 or 111 (from S-box)
  - We have K = X1 xor P1 = 001 xor 010 = 011
  - Or K = X1 xor P1 = 111 xor 010 = 101
- Using P2 and C2:
  - For C2= 11, X2 = 000 or 110
  - We have K = X2 xor P2 = 000 xor 011 = 011
  - Or, K = X2 xor P2 = 110 xor 011 = 101
- For key, it is true that the rightmost bit of key is ***1***

Differential Cryptanalysis on Simplified DES

- In this context, consider Four-round simplified DES cipher and analyse the attack
- First of all, we see the working process of Four-round simplified DES algorithm.

Simplified DES algorithm





$$R_i = L_{i-1} \oplus f\left(R_{i-1}, K_i\right)$$

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

Contd…

- The messages are *12*-bit sizes and input plaintext is written as L0R0, where L0 consists of the left *6* bits and R0 consists of the right *6* bits.
- The key K has *9* bits (it is *12* bits, however *3* bits corresponding to each 4^{th} bit is deleted)
- Subkeys *Ki* is 8 bits *obtained from 9 bits, Ex. K=010011001, then K4=01100101*
- *Expander function:*

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

- *For example: 011001→01010101*

- Two S-boxes used are:
- S1:

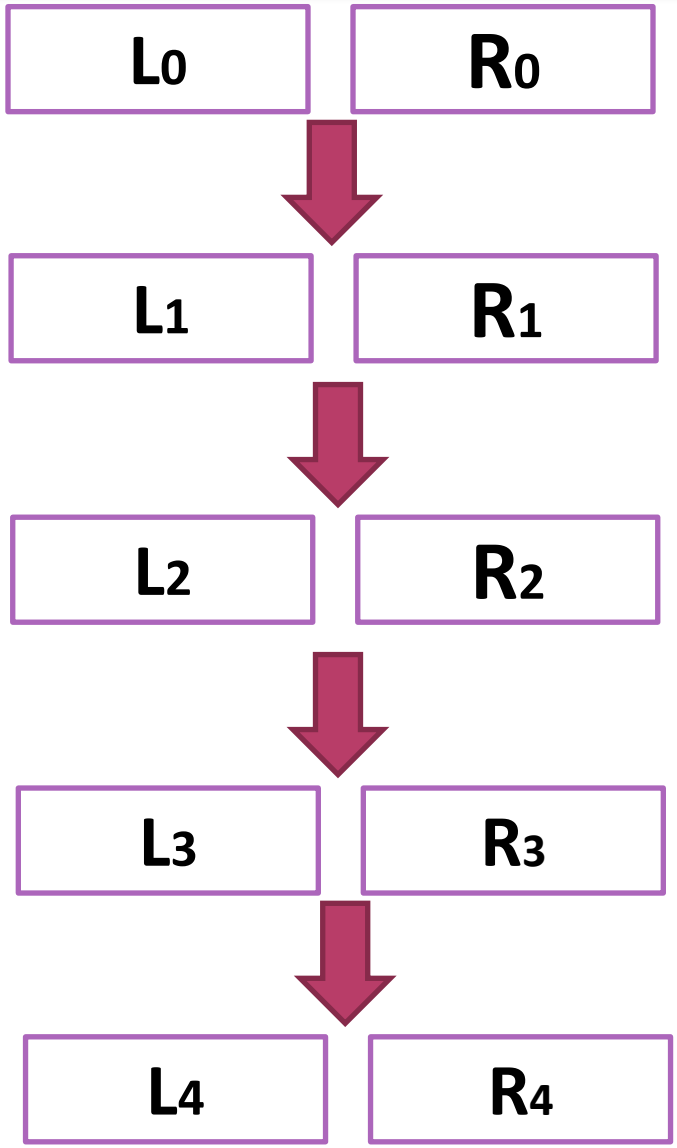
101	010	001	110	011	100	111	000
001	100	110	010	000	111	101	011

- S2:

100	000	110	101	111	001	011	010
101	011	000	111	110	010	001	100

- The input for an S-box has 4 bits.
- First bit denotes a row, and other 3 bits represent a column

Differential Cryptanalysis for 4 Rounds



- We start the analysis with

  R1=L0 xor f (R0, K1)

  L1 = R0
- Assumption: K is unknown

$$R_2 = L_1 \oplus f\left(R_1, K_2\right)$$

$$L_2 = R_1$$

$$R_3 = L_2 \oplus f\left(R_2, K_3\right)$$

$$L_3 = R_2 = L_1 \oplus f\left(R_1, K_2\right)$$

$$L_4 = R_3$$

$$R_4 = L_3 \oplus f\left(R_3, K_4\right)$$

$$= L_1 \oplus f\left(R_1, K_2\right) \oplus f\left(R_3, K_4\right)$$

- Differential Cryptanalysis is a kind of **Chosen-plaintext** attack.
- We have considered two messages:

$$L_1 R_1, \text{and } L_1^* R_1^* \text{ with } R_1 = R_1^*$$

$$\text{Then, } R_4 = L_1 \oplus f\left(R_1, K_2\right) \oplus f\left(R_3, K_4\right)$$

$$\text{and } R_4^* = L_1^* \oplus f\left(R_1, K_2\right) \oplus f\left(R_3^*, K_4\right)$$

$$\text{For each i, let } R_i' = R_i \oplus R_i^* \text{ and } L_i' = L_i \oplus L_i^*$$

$$\text{Then, } R_4' = L_1' \oplus f\left(R_3, K_4\right) \oplus f\left(R_3^*, K_4\right)$$

This may be rearranged to

$$R_4' \oplus L_1' = f\left(R_3, K_4\right) \oplus f\left(R_3^*, K_4\right)$$

$$\text{Since, } R_3 = L_4 \text{ and } R_3^* = L_4^*$$

$$R_4' \oplus L_1' = f\left(L_4, K_4\right) \oplus f\left(L_4^*, K_4\right)$$

- We know everything in this last equation except K_4.
- This is non-linear equation.

$$R_4' \oplus L_1' = f\left(L_4, K_4\right) \oplus f\left(L_4^*, K_4\right)$$

$$\text{Now, } f\left(L_4, K_4\right) = S(E\left(L_4\right) \oplus K_4)$$

$$\text{Similarly, } f\left(L_4^*, K_4\right) = S(E\left(L_4^*\right) \oplus K_4)$$

$$E\left(L_4\right) \oplus E\left(L_4^*\right) = E\left(L_4 \oplus L_4^*\right) = E\left(L_4'\right)$$

$$f\left(L_4, K_4\right) \oplus f\left(L_4^*, K_4\right)$$

$$= S(E\left(L_4\right) \oplus E\left(L_4^*\right)) = S(E\left(L_4'\right))$$

- So we have,
  - $R_4' \oplus L_1' = S(E(L_4'))$
- Compare above and guess the key used

Example

- Let's restrict our attention to S_1 (Analysis for S_2 will be similar)
- Suppose XOR input=1011 and output XOR=100.
  - Two inputs are 1010 and 0001
  - Two outputs are 110 and 010 (from S1 box)

Contd…

- We know, L_4=101110 and L_4^*=000010

E(L_4)=10111110 and E(L_4^*)=00000010

S1: Output XOR is 100, then

$$\left(1011 \oplus K_4^L, 0000 \oplus K_4^L\right) \approx \left(1010, 0001\right) or \left(0001, 1010\right)$$

$$\Rightarrow K_4^L = 0001 or 1010$$

Alternatives to DES

- Key space is the main constraint of DES.
- Not enough to resist exhaustive key search attack with 56-bit key (DES does not satisfy group properties).
- As a result, one possible way of effectively increasing the key space of DES is to perform encryption process multiple times.
- Double DES (2DES), Triple DES (3DES3, 3DES2) are some variants.

Contd.

- However, it is true that for all 2^56 key values, given any two keys K1 and K2, it would be possible to find a third key K3 such that E(K2, E(K1, P) = E(K3, P). So, 2DES, 3DES (or any no, multiple DES encryptions) would be useless. But it is not true as explained below:
- Consider that DES encryption is a mapping of 64-bit blocks to 64-bit blocks (which can be viewed as permutation).
- Consider all possible 64-bit inputs, DES will map each input block with a specific key into a unique 64-bit block (two different inputs cannot map to same output block- if so decryption is infeasible).
- With 2^64 possible inputs, the total possible mapping (or permutation) is $(2^{64})! > 10^{10^{20}}$,

On the other hand, DES defines one mapping for each different key, total mappings are $2^{56} < 10^{17}$

- Thus, it is reasonable if DES is used twice with different keys, it will produce one of the many mappings that is not defined by a single application of DES.

Double DES

Encryption:

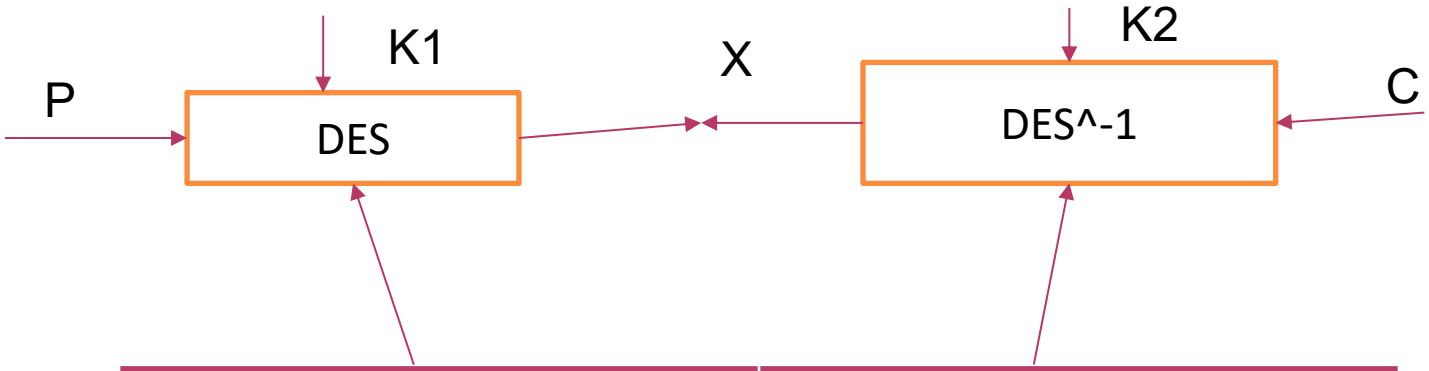
$$2DES(K_1 \| K_2, P) = DES(K_2, DES(K_1, P))$$

Decryption:

$$2DES^{-1}(K_1 \| K_2, C) = DES^{-1}(K_1, DES^{-1}(K_2, C))$$

- Although, 2DES has a key-length of 112, but it can be broken using about 2^{57} DES/DES^{-1} computations.

An attack known as **Meet-in-the-middle** is possible $DES^{-1}(K_2, C) = DES(K_1, P)$



$DES(K_1, P)$	$DES^{-1}(K_2, C)$

2DES

- For a given plaintext, there are possible 2^64 ciphertext values.
- Since 2DES uses 112-bit key, so for a given plaintext P, the number of different 112-bit keys that will produce a given ciphertext C is 2^112/2^64 = 2^48.
- Thus, it produces 2^48 false alarms on the first (P, C) pair.
- If another known pair (P, C) is used and 2^48 keys are used, then, we have 2^48/2^64 = 2^-16 false alarms, and thus, the correct key can be obtained with 1-2^16 probability.

Triple DES (3DES)

3DES3 Encryption: $3DES3(K_1 \| K_2 \| K_3, P)$

$$= DES\left(K_3, DES^{-1}\left(K_2, DES\left(K_1, P\right)\right)\right)$$

- The effective key length is 112 due to Meet-in-the-middle attack .
- 3DES2 Encryption: $3DES2(K_1 \| K_2, P)$

$$= DES\left(K_1, DES^{-1}\left(K_2, DES\left(K_1, P\right)\right)\right)$$

- The effective key length is 112 .

DESX

- Although 2DES, 3DES3, and 3DES2 appear to provide adequate security, they are slow.
- DESX encryption:

$$DESX\left(K_1 \| K_2 \| K_3, P\right)$$
$$= K_3 \oplus DES\left(K_2, K_1 \oplus P\right)$$

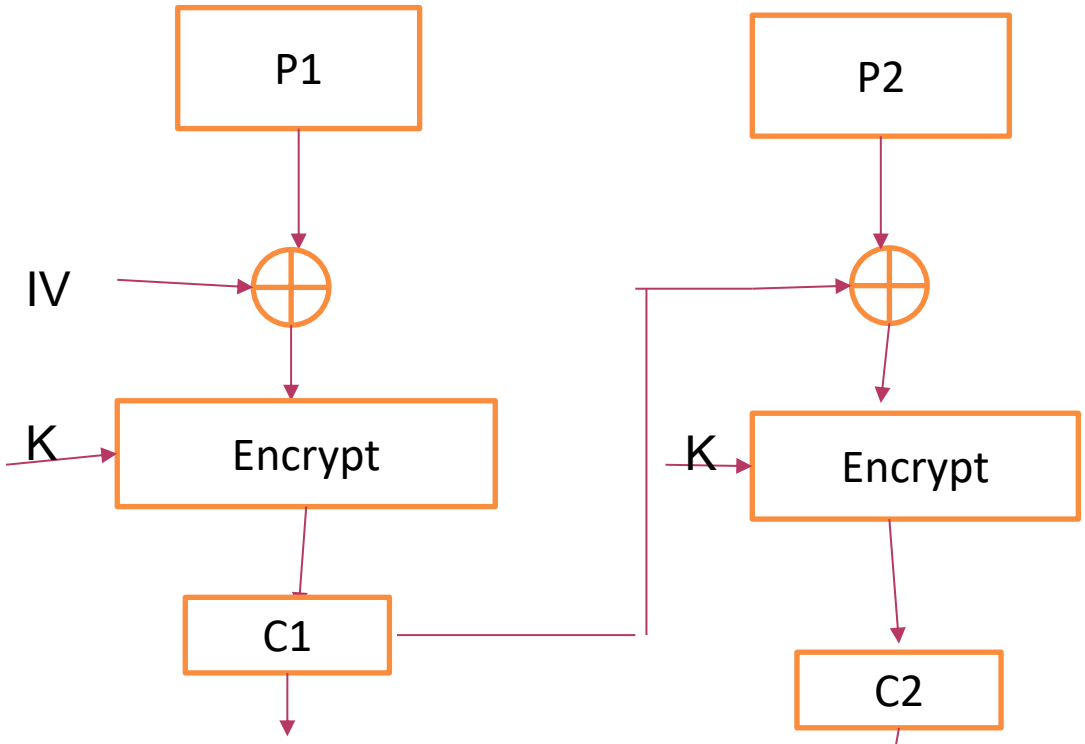
- The effective key length is 120, which is large enough for security.
- DESX is popular because it is much cheaper than 2DES, and 3DES while providing adequate security.

Block Cipher Mode of Operation

- To apply DES, four modes of operations have been defined and they are :
  - Electronic Code Book Mode (ECB)
  - Cipher Block Chaining Mode (CBC)
  - Cipher Feedback Mode (CFB)
  - Output Feedback Mode (OFB)
- ECB: Codebook is used here, i.e., for a given key, there is a unique ciphertext for every 64-bit input plaintext. So, we can imagine a gigantic codebook in which an entry for every possible 64-bit plaintext showing its corresponding ciphertext.
- In ECB, if the same plaintext appears more than one, it always produces the same ciphertext.- its weakness

CBC Mode

- To overcome security weakness of ECB, CBC is used, which produces different ciphertext blocks for the same plaintext block.
- For encryption in CBC, input to the encryption algorithm is the XOR of current plaintext block and preceding ciphertext block (and the encrypted using key) and to produce first ciphertext block, an Initialization Vector (IV) is XORed with the first plaintext block.
- For decryption, after decryption of ciphertext using key, the result is XORed with the preceding ciphertext block to produce plaintext block.

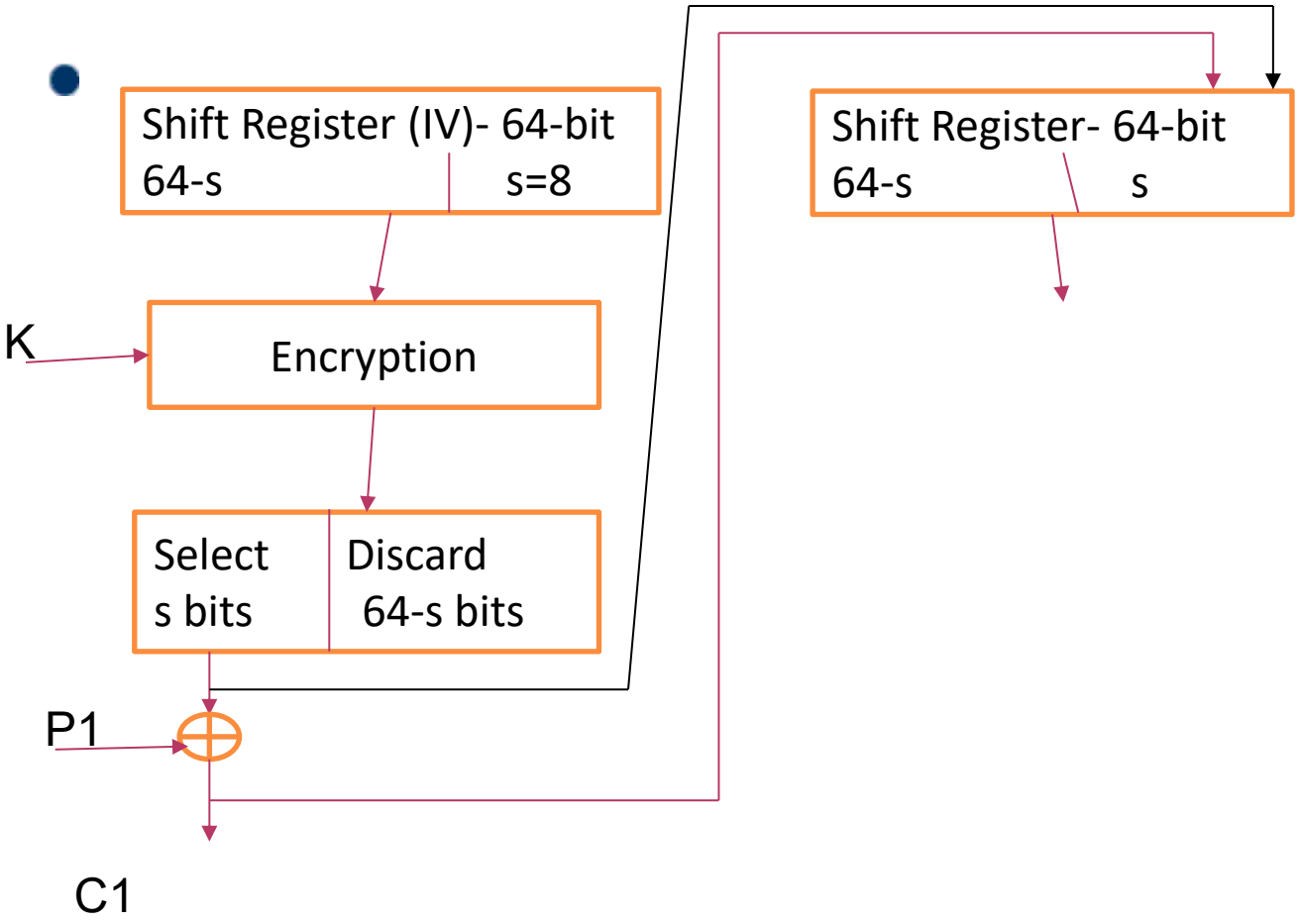

P1
P2
IV
K
Encrypt
K
Encrypt
C1
C2

CBC

- The encryption and decryption in CBC are:
  - $C_j = E_K[C_{j-1} \oplus P_j]$
  - $D_K[C_j]) = D_K[E_K[C_{j-1} \oplus P_j]] = C_{j-1} \oplus P_j$
  - $P_j = C_{j-1} \oplus C_{j-1} \oplus P_j$
- IV must be known to both sender and receiver, and it should be protected as well as the key (it can be done by sending IV using ECB).
- DES block cipher can be converted into stream cipher and it is done using CFB or OFB modes.
  - RC4 WLAN

CFB Mode

- In CFB, input to encryption algorithm is a 64-bit shift register, which is initially set to some initialization vector (IV).
- It is encrypted using key *K* and the leftmost *s* bit (let *s=8* for byte stream) are XORed with the 8-bit plaintext to produce first unit of ciphertext C1.
- The contents of shift register is left shifted by s bits and C1 is placed in the rightmost s-bit position of shift register, and the processes is continued.
  - $C_1 = P_1 \oplus S_s(E_K(IV))$ Encryption
  - $P_1 = C_1 \oplus S_s(D_K(IV))$ Decryption

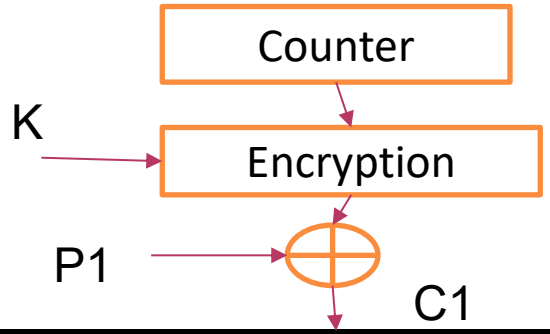

Shift Register (IV)- 64-bit
64-s
s=8
Shift Register- 64-bit
64-s
s
K
Encryption
Select
s bits
Discard
64-s bits
P1
C1

OFB Mode

- It is similar to CFB mode, in OFB, the output of encryption function is fed back to the shift register (instead of ciphertext).
- One advantage is that the bit error in transmission do not propagate, for ex. if error in C1, only the P1 value is affected in OFB mode, and subsequent plaintext units are not corrupted.
- With CFB, C1 serves as input to shift register and therefore causes additional corruption downstream.

Counter Mode

- A counter, equal to plaintext block size is used and its value must be different for each plaintext block.
- Initially, the counter is initialized to some value and incremented by 1 for each subsequent block (counter operated as mod 2^b, where b is the size of counter)



Conclusions

- DES was the dominant symmetric encryption algorithm from the mid-1970 to mid-1990s.
- DES with 56 bits key can be broken relatively easily nowadays.
- No practical attack is currently known on 3DES, so it is used in various applications.
- AES is considered as standard and suited for resource constrained applications.