Parul® University

203105381-Cryptography

Prof. Ashish Patel , Assistant Professor
Computer Science & Engineering

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER-3

BLOCK CIPHERS AND THE DATAENCRYPTION STANDARD

Parul®
University

DIGITAL LEARNING CONTENT

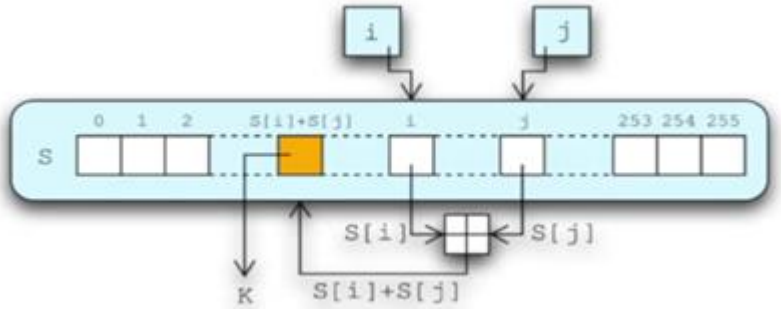
Syllabus

Block Ciphers and the Data Encryption Standard:: Block Cipher Principles, Data Encryption Standard (DES), Deferential and Linear Cryptanalysis, Block Cipher Design Principles, Block Cipher Operation, RC4

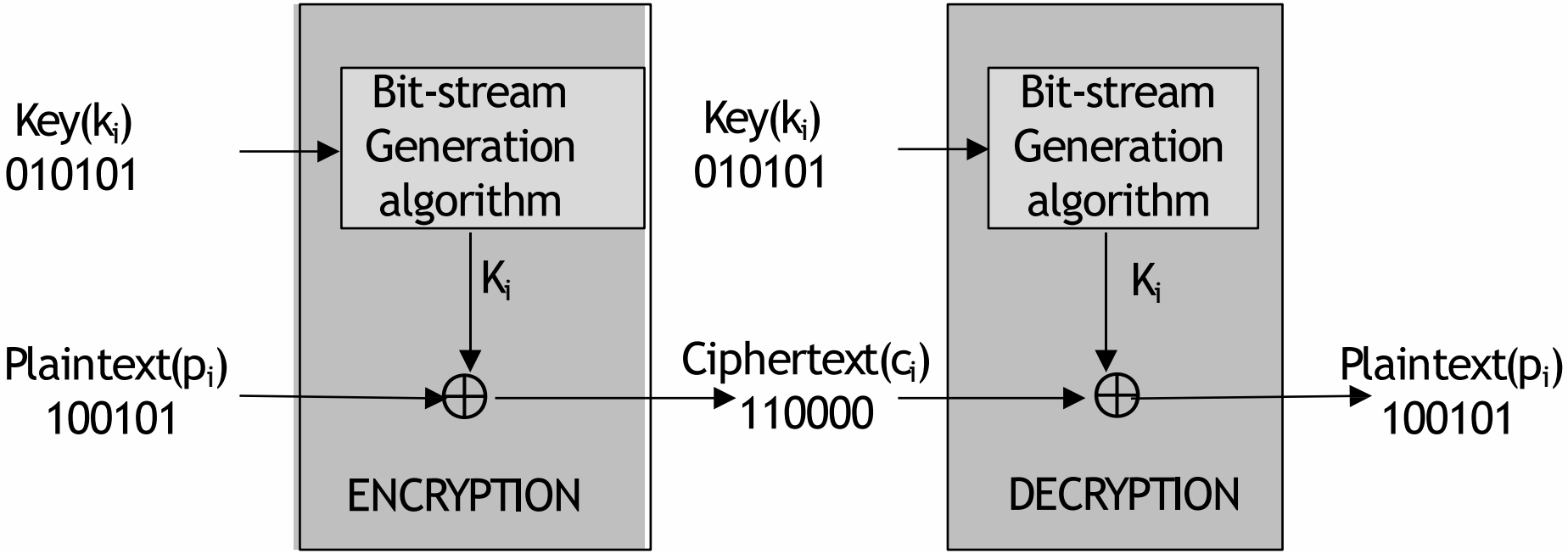
Stream Cipher

- A stream cipher is one that encrypts a digital data stream one bit or one byte at a time.

- Examples:
  - Vigenère cipher
  - Ceasar Cipher
  - Vernam cipher.



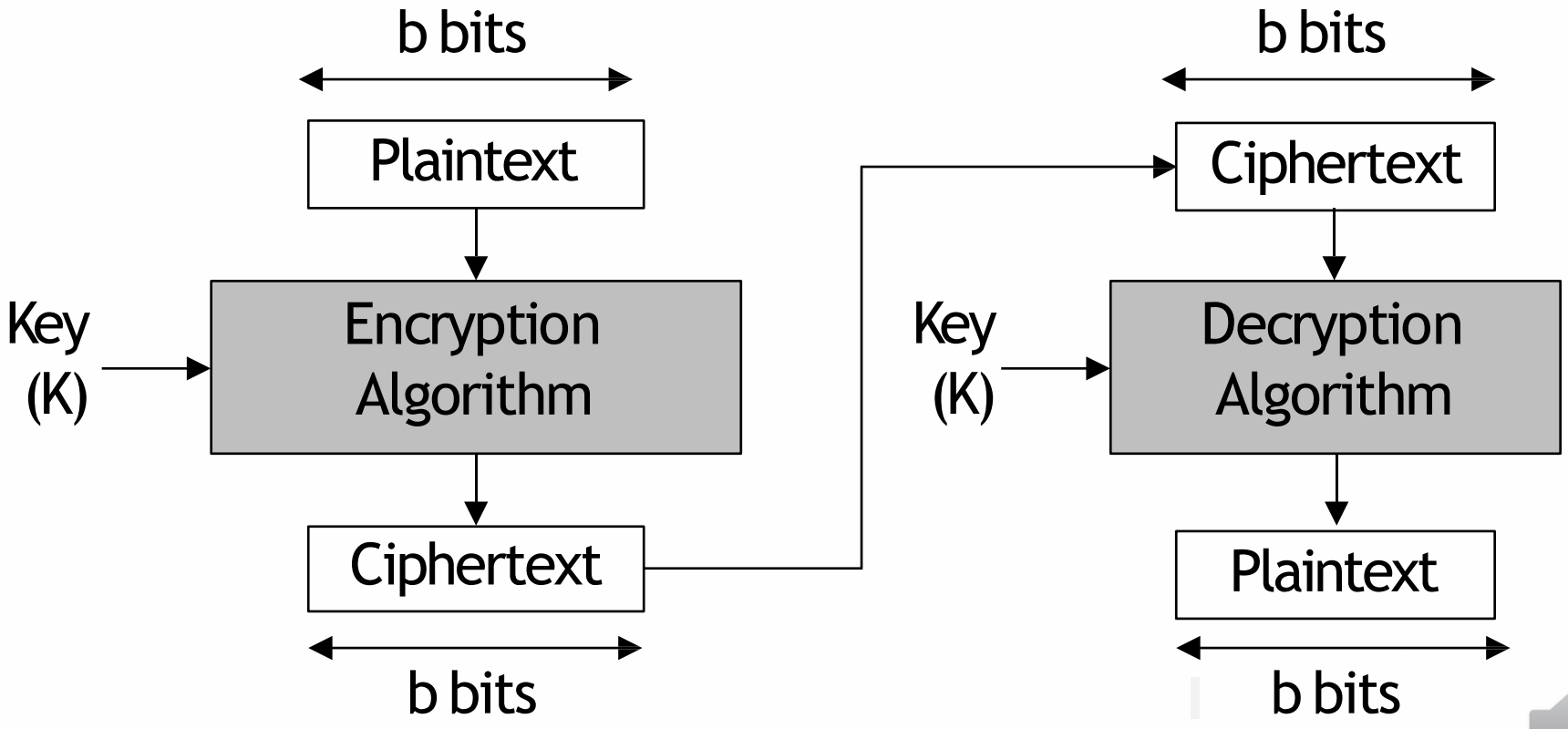
Stream Cipher- Example



Block Cipher

- A block size of 64 or 128 bits is used.
- Symmetric encryption
- Confusion: Key to ciphertext relationship shoud be very complicated.
- Diffusion: Output should depend on the input in a complex way.
- Examples:
  - Feistel cipher, DES, Triple DES,AES etc.

BlockCipher-Example



Diffusion and Confusion

- The terms diffusion and confusion were introduced by claude Shannon.
- Shannon's concern was to prevent cryptanalysis based on .
- "Diffusion" = the statistical structure of the plaintext is dissipated into long-range statistics of the cipher text.
- This is achieved by having each plaintext digit affect the value of many cipher text digits;
- The mechanism of diffusion look for to make the statistical relationship between the plaintext and cipher text as complex as possible in order to prevent attempts to deduce the key.

Diffusion and Confusion

- “Confusion” = confusion seeks to make the relationship between the statistics of the cipher text and the value of the encryption key as complex as possible, again to thwart attempts to discover the key.
- Thus, even if the attacker can get some handle on the statistics of the cipher text, the way in which the key was used to produce that cipher text is so complex as to make it difficult to deduce the key.
- This is achieved by the use of a complex substitution algorithm.

DES- Data Encryption Standard

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

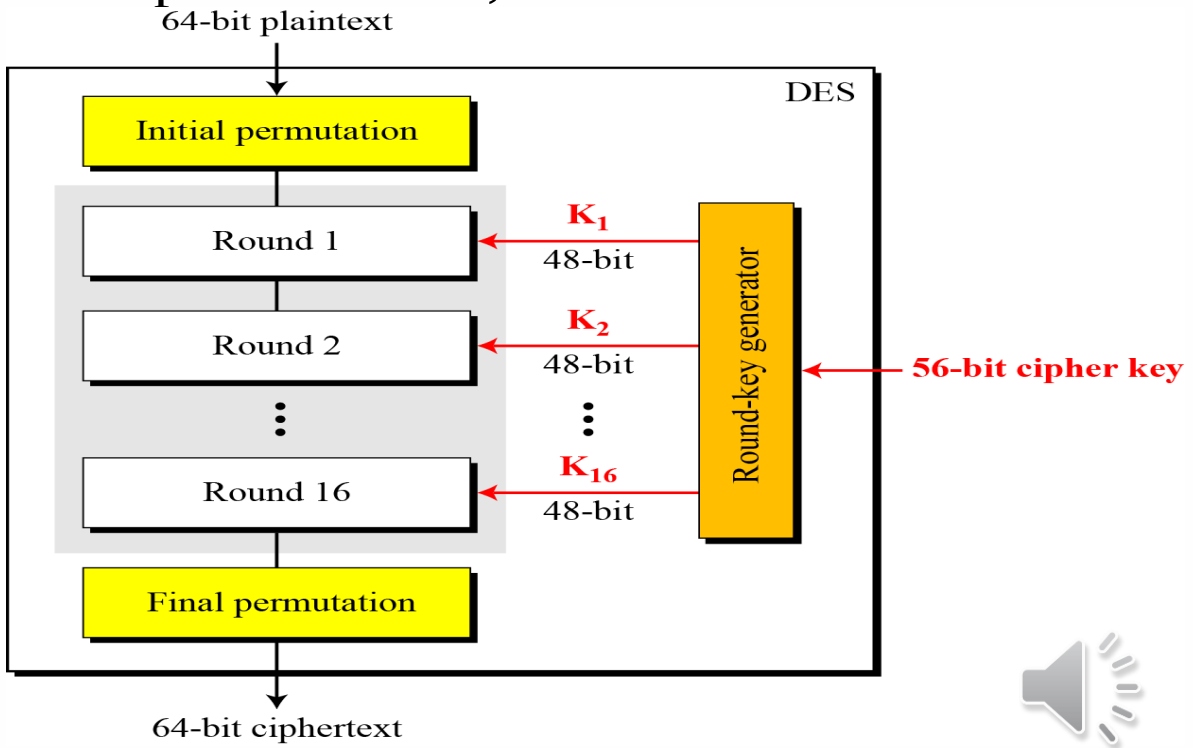
- In 1973, NIST published a request for proposals for a national symmetric-key cryptosystem.
- A proposal from IBM, a modification of a project called Lucifer, was accepted as DES.
- DES was published in the Federal Register in March 1975 as a draft of the Federal Information Processing Standard (FIPS).

DES- Data Encryption Standard

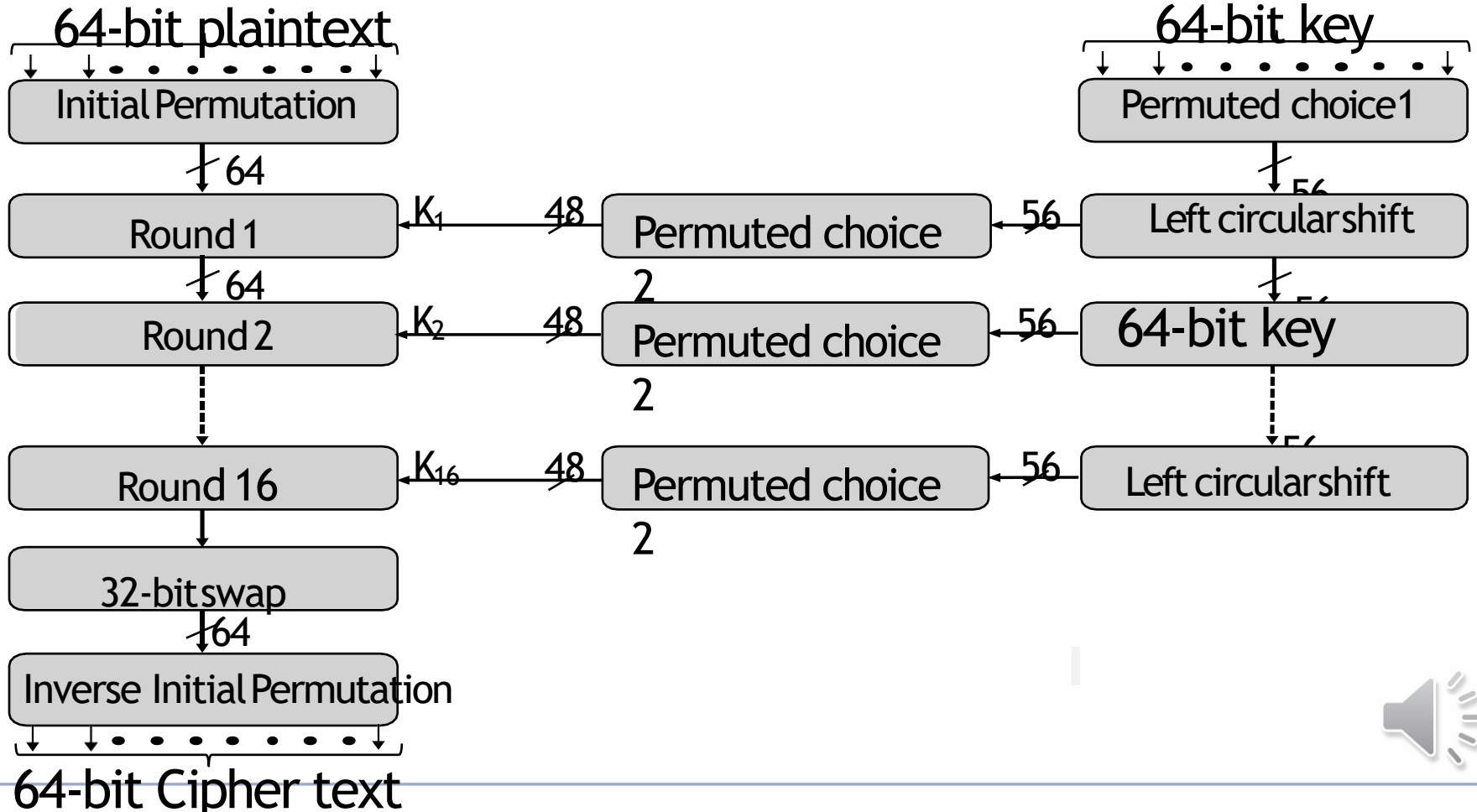
- Type: *DES is a block cipher*
- Block Size : 64-bit
- Key Size: 64-bit, with only 56-bit effective
- Number of Rounds: 16

DES STRUCTURE

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



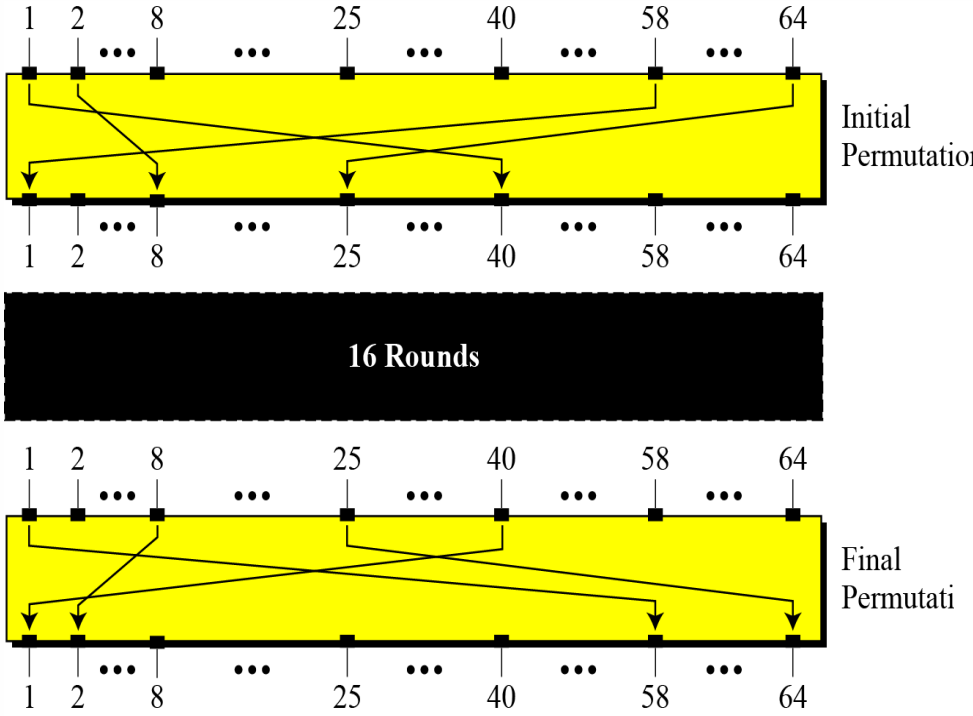
DES Encryption Algorithm



Parul® University

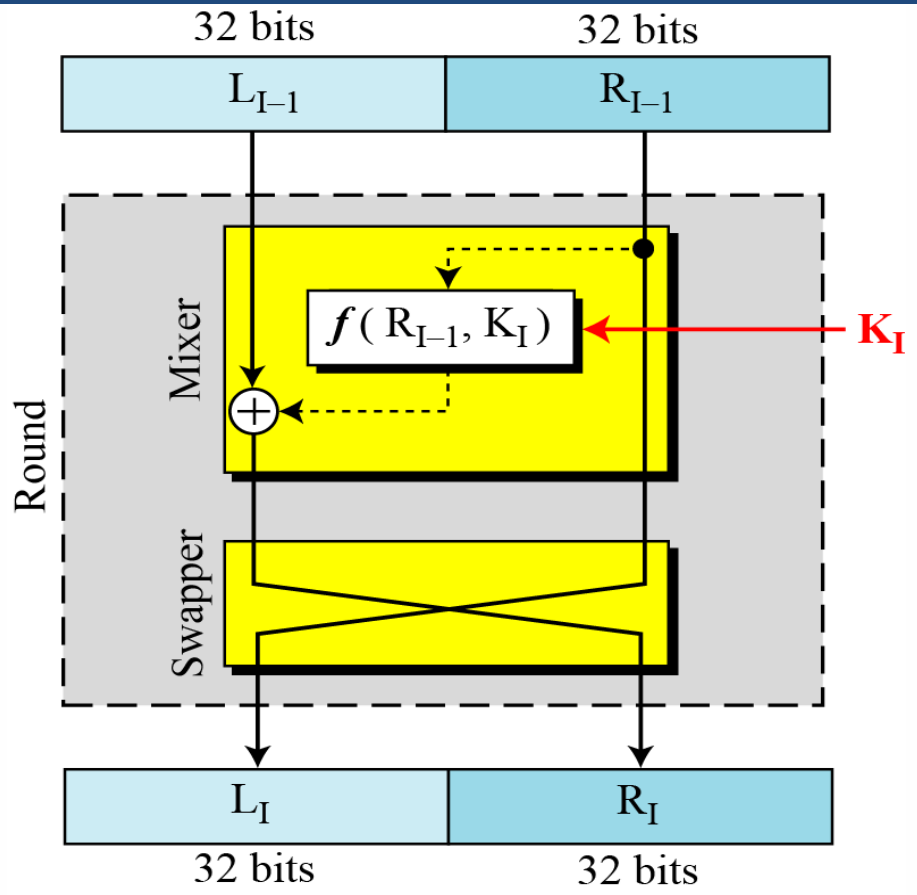
DIGITAL LEARNING CONTENT

Initial and Final Permutations in DES



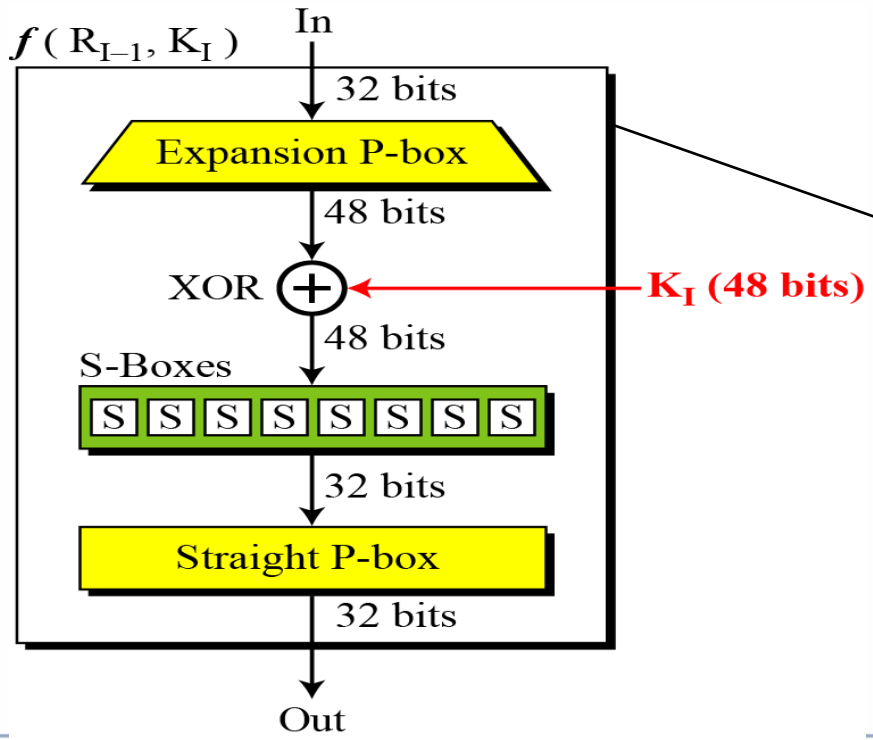
Initial Permutation	*Final Permutation*
58 50 42 34 26 18 10 02	40 08 48 16 56 24 64 32
60 52 44 36 28 20 12 04	39 07 47 15 55 23 63 31
62 54 46 38 30 22 14 06	38 06 46 14 54 22 62 30
64 56 48 40 32 24 16 08	37 05 45 13 53 21 61 29
57 49 41 33 25 17 09 01	36 04 44 12 52 20 60 28
59 51 43 35 27 19 11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

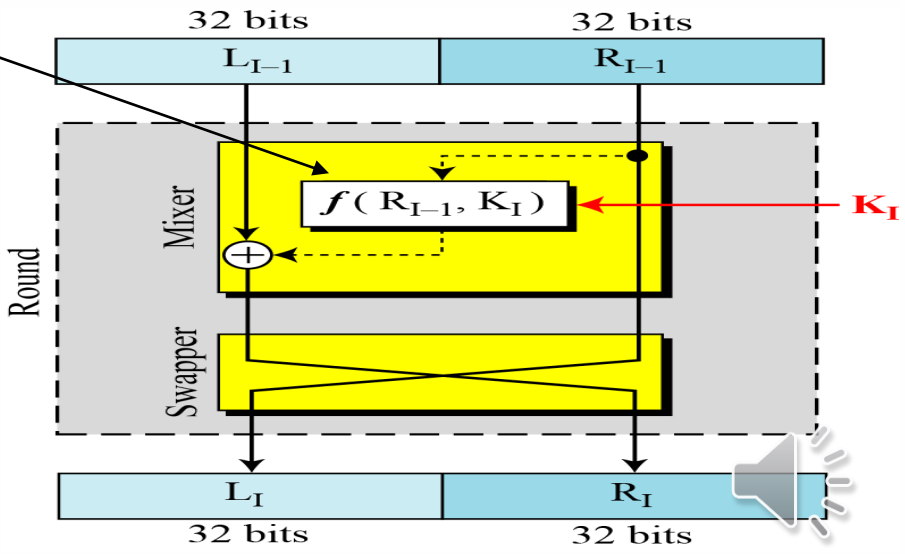
DES uses 16 rounds. Each round of DES is a Feistel cipher.



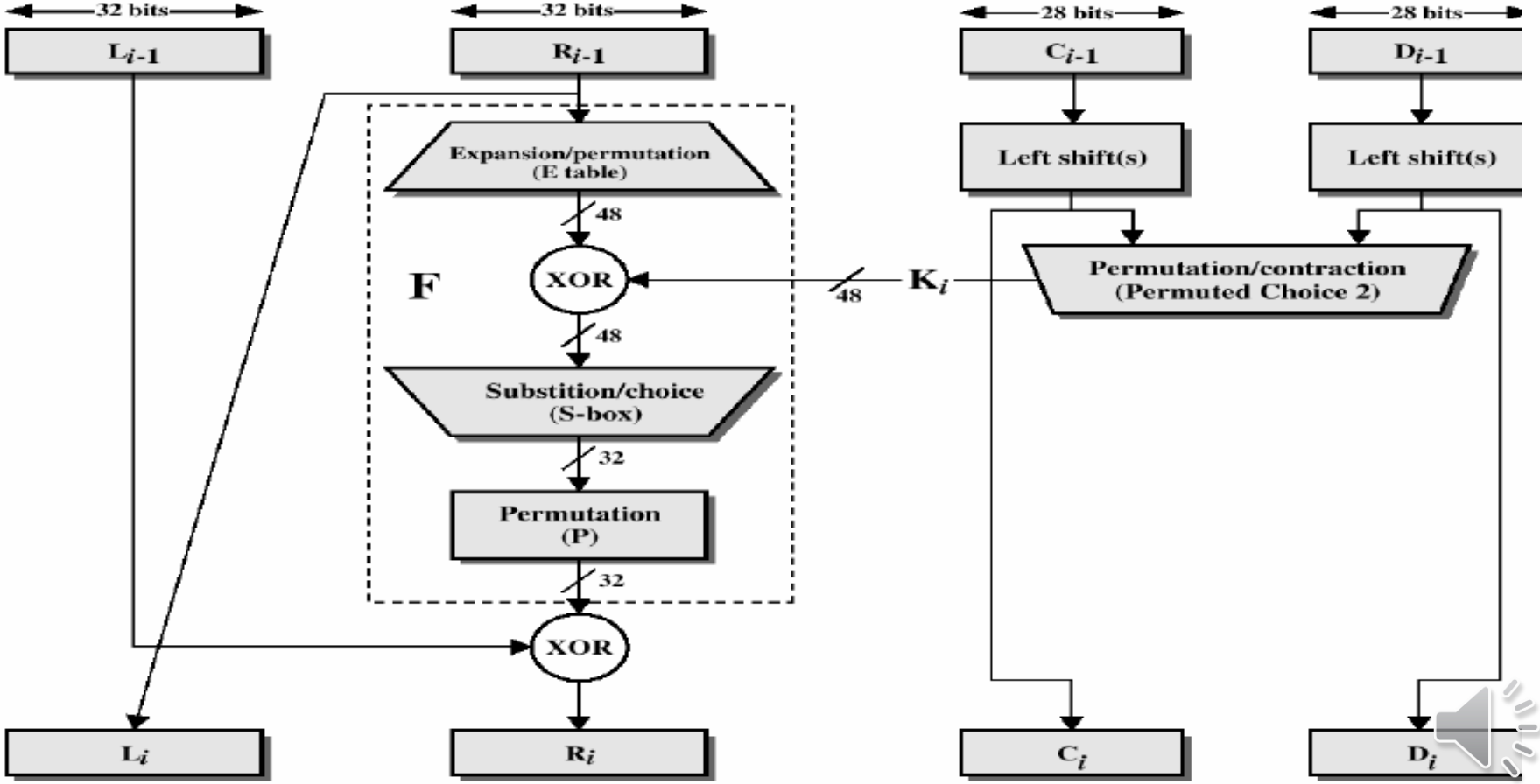
DES Function

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.





Details of Single Round of DES Algorithm

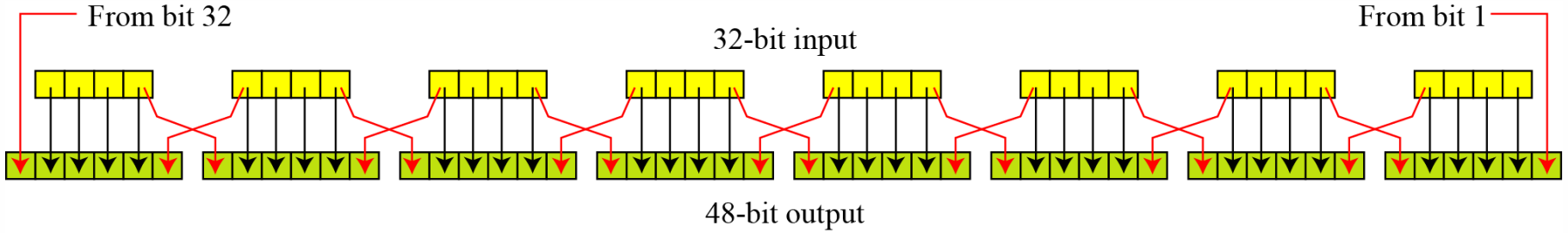


DES Single Round (Cont…)

1. Key Transformation
   - Permutation of selection of sub-key from original key
2. Expansion Permutation (E-table)
   - Right half is expanded from 32-bits to 48-bits
3. S-box Substitution
   - Accepts 48-bits from XOR operation and produce 32-bits using 8 substitution boxes (each S-boxes has a 6-bit i/p and 4-bit o/p).
4. P-Box Permutation
5. XOR and Swap

Expansion permutation

Expansion P-box –E(R):Since RI−1 is a 32-bit input and KI is a 48-bit key, we first need to expand RI−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

Expansion P-box table

Although the relationship between the input and output can be defined mathematically, DES uses Table 2 to define this P-box.

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

Block Cipher Components

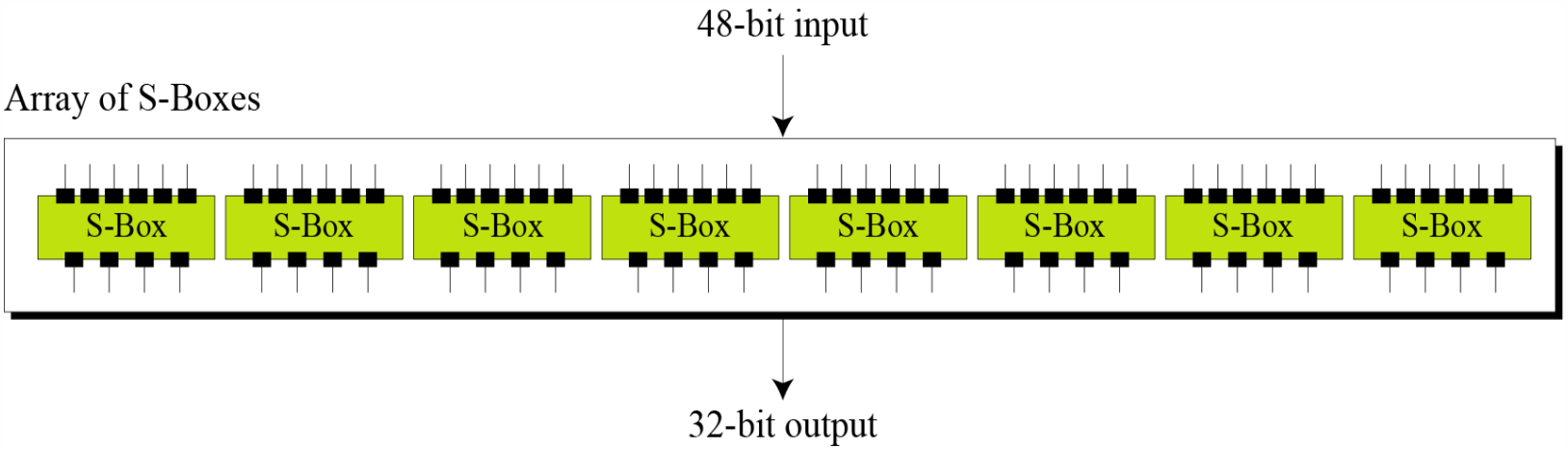
- **P Box**
  - key less fixed transposition cipher.
  - used to provide diffusion
  - Input bits are permuted to produce output bits
  - Classifies in three types:

Whitener (XOR)

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-Boxes

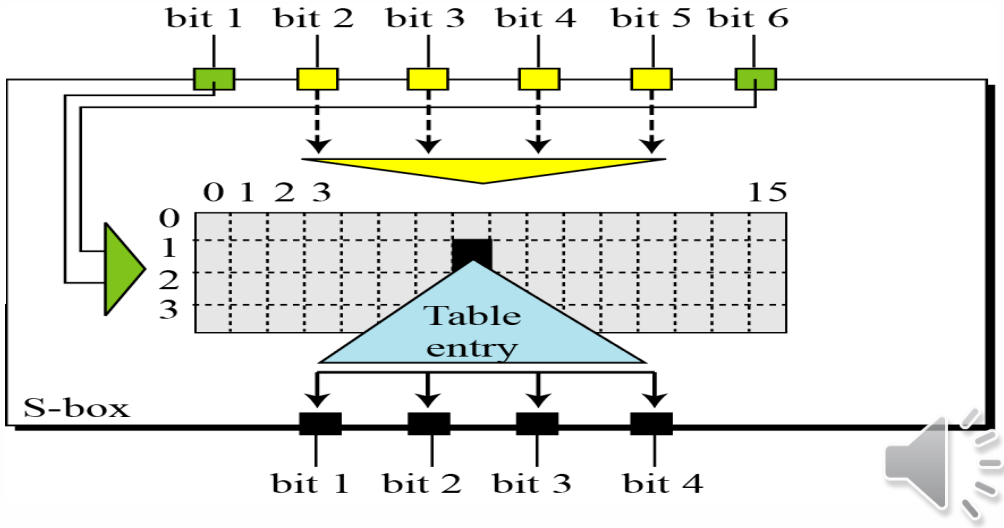
The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.



S-Boxes(Continue)

	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

Table shows the permutation for S-box 1. For the rest of the boxes see the textbook.



Block Cipher Components

- S Box
  - key less fixed substitution cipher.
  - used to provide confusion
  - dependent on unknown key
  - take n bit of plaintext as a input and produce m bit of cipher text as output, where value of n & m may be same or different
  - mapping is predetermined

Example-1

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 6.3 (S-box 1). The result is 12 in decimal, which in binary is 1100. So the input 100011 yields the output 1100.

Example-2

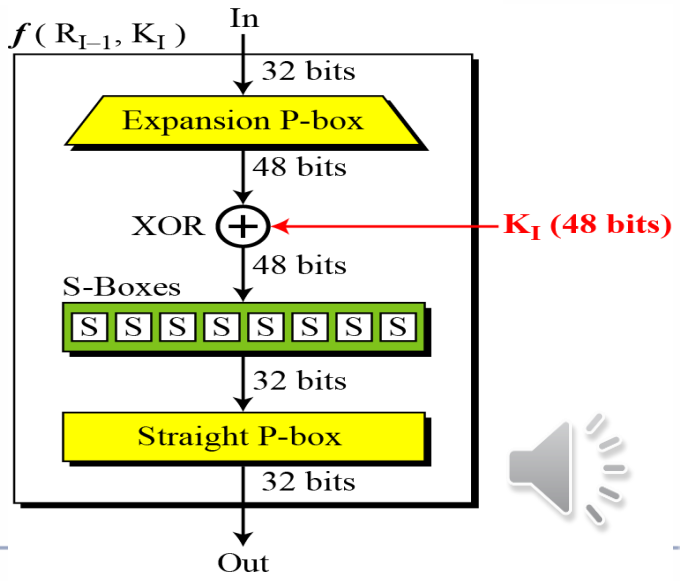
The input to S-box 8 is 000000. What is the output?

Solution: If we write the first and the sixth bits together, we get 00 in binary, which is 0 in decimal. The remaining bits are 0000 in binary, which is 0 in decimal. We look for the value in row 0, column 0, in Table 6.10 (S-box 8). The result is 13 in decimal, which is 1101 in binary. So the input 000000 yields the output 1101.

Straight Permutation

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

- Straight P box
  - takes n bits as i/p permutes n bits as o/p
  - so n! of ways to map i/p to o/p



Cipher and Reverse Cipher

Using mixers and swappers, we can create the cipher and reverse cipher, each having 16 rounds.

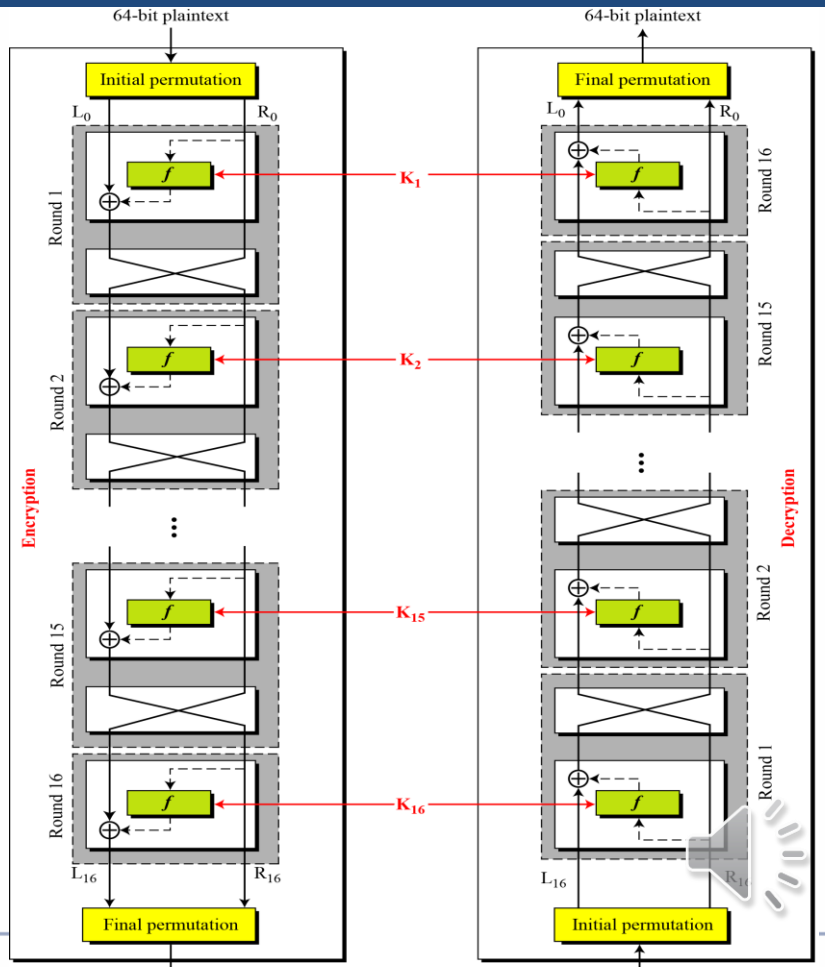
First Approach
To achieve this goal, one approach is to make the last round (round 16) different from the others; it has only a mixer and no swapper.

Note: In the first approach, there is no swapper in the last round.

Parul®
University

DIGITAL LEARNING CONTENT

DES cipher and reverse cipher for the first approach

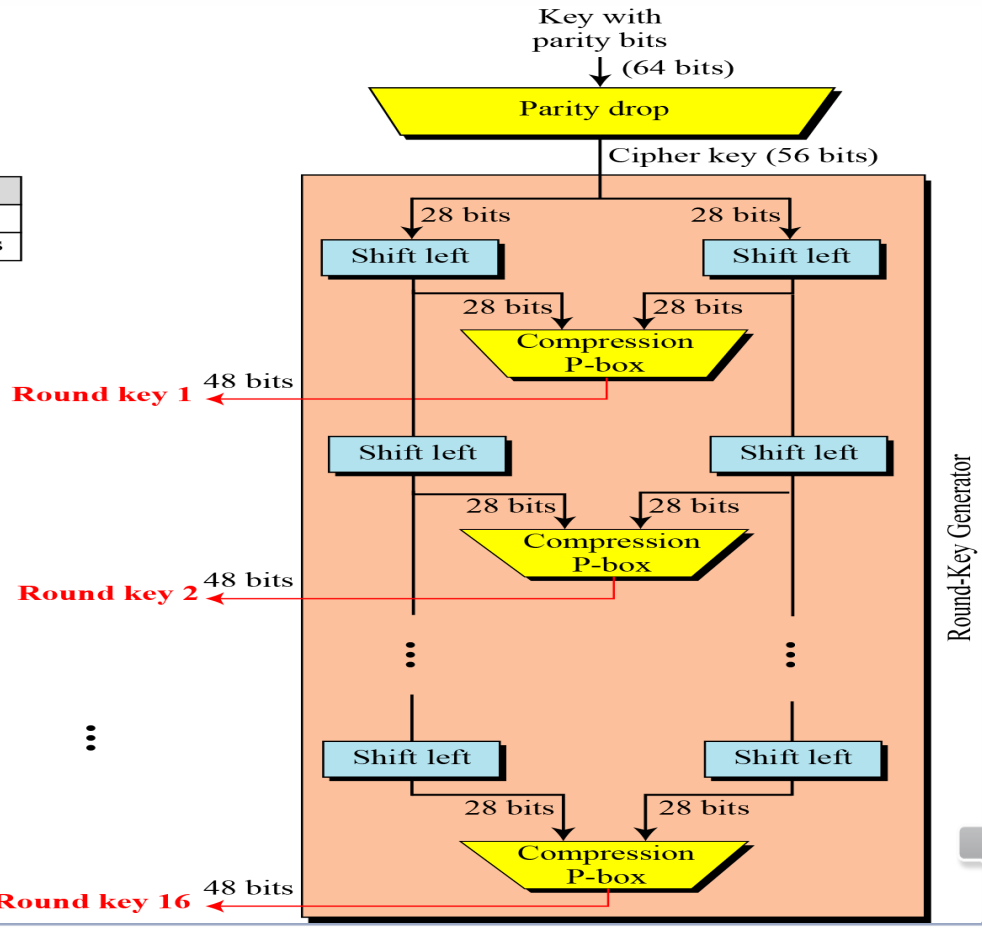


Key Generation

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



Parity-bit drop table PC-1

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

Block Cipher Components

- Circular shift
  - Classifies in two ways:
    - Circular left shift
      - every bit of word is shifted by specific number of positions in left direction
      - n number of leftmost bits are removed and placed at rightmost

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

Number of bits shifts

Key-compression table (PC-2)

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

Example-3

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

Plaintext: 123456ABCD132536 Key: AABB09182736CCDD
CipherText: C0B7A8D05F3A829C

Plaintext: 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

Example-3(Continue)

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

Example-4

Let us see how Bob, at the destination, can decipher the ciphertext received from Alice using the same key. Table 6.16 shows some interesting points.

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

Critics have used a strong magnifier to analyze DES. Tests have been done to measure the strength of some desired properties in a block cipher.

Topics discussed in this section:

- Properties
- Design Criteria
- DES Weaknesses

Properties

Two desired properties of a block cipher are the avalanche effect and the completeness.

The Avalanche Effect

- A desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce significant change in the ciphertext – avalanche effect.

- In particular, a change in one bit of the plaintext or one bit of the key should produce a change in many bits of the ciphertext.

The Avalanche Effect

Table 3.6 Avalanche Effect in DES: Change in Plaintext

Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32

Round		δ
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32

The Avalanche Effect

Table 3.7 Avalanche Effect in DES: Change in Key

Round		δ
	02468aceeca86420 02468aceeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2	27

Round		δ
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeeaa	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	33
16	75e8fd8f25896490	30

Completeness effect

Completeness effect means that each bit of the ciphertext needs to depend on many bits on the plaintext.

Design Criteria

S-Boxe:
The design provides confusion and diffusion of bits from each round to the next.

P-Boxes:
They provide diffusion of bits.

Number of Rounds:
DES uses sixteen rounds of Feistel ciphers. the ciphertext is thoroughly a random function of plaintext and ciphertext.

DES Weaknesses

During the last few years critics have found some weaknesses in DES.

Weaknesses in Cipher Design

1. Weaknesses in S-boxes
2. Weaknesses in P-boxes
3. Weaknesses in Key

Security of DES

DES, as the first important block cipher, has gone through much scrutiny. Among the attempted attacks, three are of interest: brute-force, differential cryptanalysis, and linear cryptanalysis.

Topics discussed in this section:

- Brute-Force Attack
  Differential Cryptanalysis
- Linear Cryptanalysis

Brute-Force Attack

We have discussed the weakness of short cipher key in DES. Combining this weakness with the key complement weakness, it is clear that DES can be broken using 255 encryptions.

Differential Cryptanalysis

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

Linear Cryptanalysis

Linear cryptanalysis is newer than differential cryptanalysis. DES is more vulnerable to linear cryptanalysis than to differential cryptanalysis. S-boxes are not very resistant to linear cryptanalysis. It has been shown that DES can be broken using 243 pairs of known plaintexts. However, from the practical point of view, finding so many pairs is very unlikely.

The Strength of DES

- The use of 56-bit keys
- With the key length of 56 bits, there are 256 possible keys, which is approximately 7.2x1016 keys. Thus, on the face of it, brute-force attack appears impractical.
- DES finally and definitively proved insecure in july 1998, when the Electronic Frontier Foundation (EFF) announced that it had broken a DES encryption using a special-purpose “DES Crackers” machine that was built for less than $250000.
- Fortunately, there are a number of alternatives to DES, the most important of which are AES and triple DES

The Strength of DES (Conti…)

- The nature of the DES Algorithm.
- Another concern is the possibility that cryptanalysis is possible by exploiting the characteristics of the DES algorithm.
- The focus of concern has been on the eight substitution tables, or S-boxes, that are used in each iteration.
- Because the design criteria for these boxes, and indeed for the entire algorithm, were not made public, there is a mistrust that the boxes were constructed in such a way that cryptanalysis is possible for an adversary who knows the weakness in the S-boxes.
- This statement is attracting and over the years a number of regularities and unexpected behaviors of the S-boxes have been discovered.
- Even if this, no one has so far succeeded in discovering the supposed fatal weaknesses in the S-boxes.

The Strength of DES (Conti…)

- Timing Attacks
- Timing attack is one in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertext.
- A timing attack exploits the fact that an encryption or decryption algorithm often takes slightly different amounts of time on different inputs.
- DES appears to be fairly resistant to a successful timing attack.

Block cipher design principles

- DES design criteria
  - Focus on design of S box and P function
- Number of rounds
  - The greater the number of rounds the more difficult it is to perform cryptanalysis, even for a relatively weak F.
- Design of function F
  - Algorithm should have good avalanche property
  - Bit independence criterion (BIC) that states that output bits j and k should change independently when any single bit I is inverted.
- Key schedule Algorithm
  - Select the subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key.

Differential and Linear Cryptanalysis

- Differential Cryptanalysis attack
- It found in 1990.
- It required 247 chosen plain text.
- Differential cryptanalysis is the first published attack that is capable of breaking DES in less than 255 encryptions.
- The idea behind differential cryptanalysis is to observe the behavior of pairs of text blocks evolving along each round of the cipher, instead of observing the evolution of a single text block.

Differential and Linear Cryptanalysis

- Differential Cryptanalysis attack
- Consider the original plaintext block m to consist of two halves m0,m1.
- Each round of DES maps the right-hand input into the left-hand output and sets the right-hand output to be a function of the left-hand input and the subkey for this round. So, at each round, only one new 32-bit block is created.
- If we label each new block mi (2 <= i<=17) then the intermediate message halves are related as follows:

$$m_{i+1} = m_{i-1} \oplus f(m_i, K_i), \qquad i = 1, 2, \ldots, 16$$

Differential Cryptanalysis attack

- In differential cryptanalysis, we start two messages, m and m', with a known XOR difference Δm=m XOR m' and consider the difference between the intermediate message halves: Δmi=mi XOR mi' then we have.

$$\begin{aligned}\Delta m_{i+1} &= m_{i+1} \oplus m'_{i+1} \\ &= [m_{i-1} \oplus \mathrm{f}(m_i, K_i)] \oplus [m'_{i-1} \oplus \mathrm{f}(m'_i, K_i)] \\ &= \Delta m_{i-1} \oplus [\mathrm{f}(m_i, K_i) \oplus \mathrm{f}(m'_i, K_i)]\end{aligned}$$

Therefore, if we know and with high probability, then we know with high probability. Furthermore, if a number of such differences are determined, it is feasible to determine the subkey used in the function F.

Linear Cryptanalysis

- It is a minor improvement over differential cryptanalysis
- This attack is based on finding linear approximations to describe the transformations performed in DES.
- This method can find a DES key given 243 known plaintexts, as compared to 247 chosen plaintexts for differential cryptanalysis.
- Although this is a minor improvement, it still leaves linear cryptanalysis infeasible as an attack on DES.

Linear Cryptanalysis

We now give a brief summary of the principle on which linear cryptanalysis is based. For a cipher with n-bit plaintext and ciphertext blocks and an m-bit key, let the plaintext block be labeled P[1], ... P[n], the cipher text block C[1], ... C[n], and the key K[1], ..., K[m]. Then define

$$A[i, j, \ldots, k] = A[i] \oplus A[j] \oplus \ldots \oplus A[k]$$

The objective of linear cryptanalysis is to find an effective *linear* equation of the form:

$$P[\alpha_1, \alpha_2, \ldots, \alpha_a] \oplus C[\beta_1, \beta_2, \ldots, \beta_b] = K[\gamma_1, \gamma_2, \ldots, \gamma_c]$$

- Once a proposed relation is determined, the procedure is to compute the results of the left-hand side of the preceding equation for a large number of plaintext–ciphertext pairs. If the result is 0 more than half the time,

assume $K[\gamma_1, \gamma_2, \ldots, \gamma_c] = 0$. If it is 1 most of the time, assume $K[\gamma_1, \gamma_2, \ldots, \gamma_c] = 1$

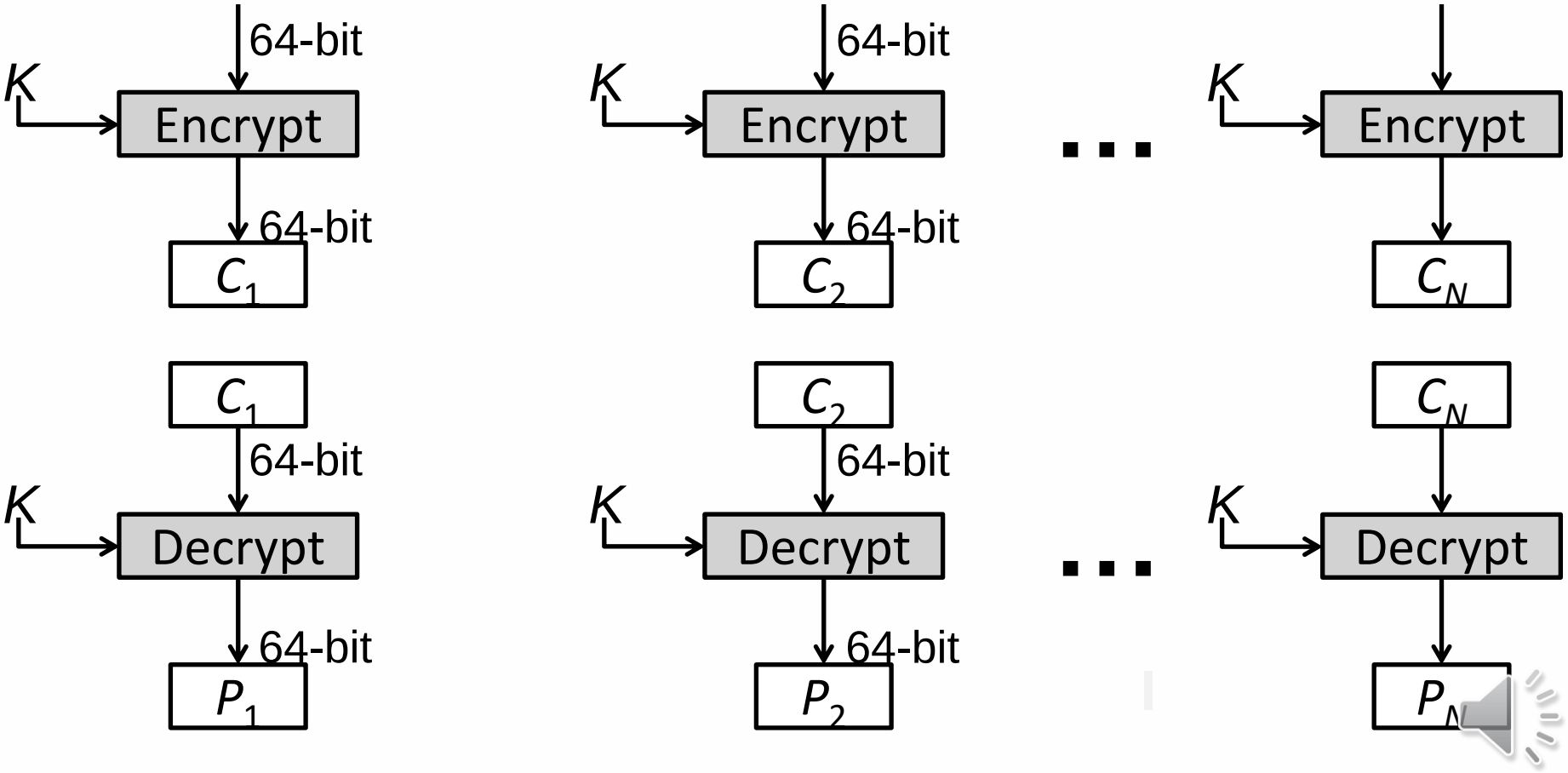
Block Cipher Modes of Operations

- To apply a block cipher in a variety of applications, five "modes of operation" have been defined.
- The five modes are intended to cover a wide variety of applications of encryption for which a block cipher could be used.
- These modes are intended for use with any symmetric block cipher, including triple DES and AES.
- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter (CTR)

1. Electronic Code Book (ECB)

- In ECB Mode Plaintext handled one block at a time and each block of plaintext is encrypted using the same key.
- The term codebook is used because, for a given key, there is a unique ciphertext for every b-bit block of plaintext.

ECB Encryption & Decryption



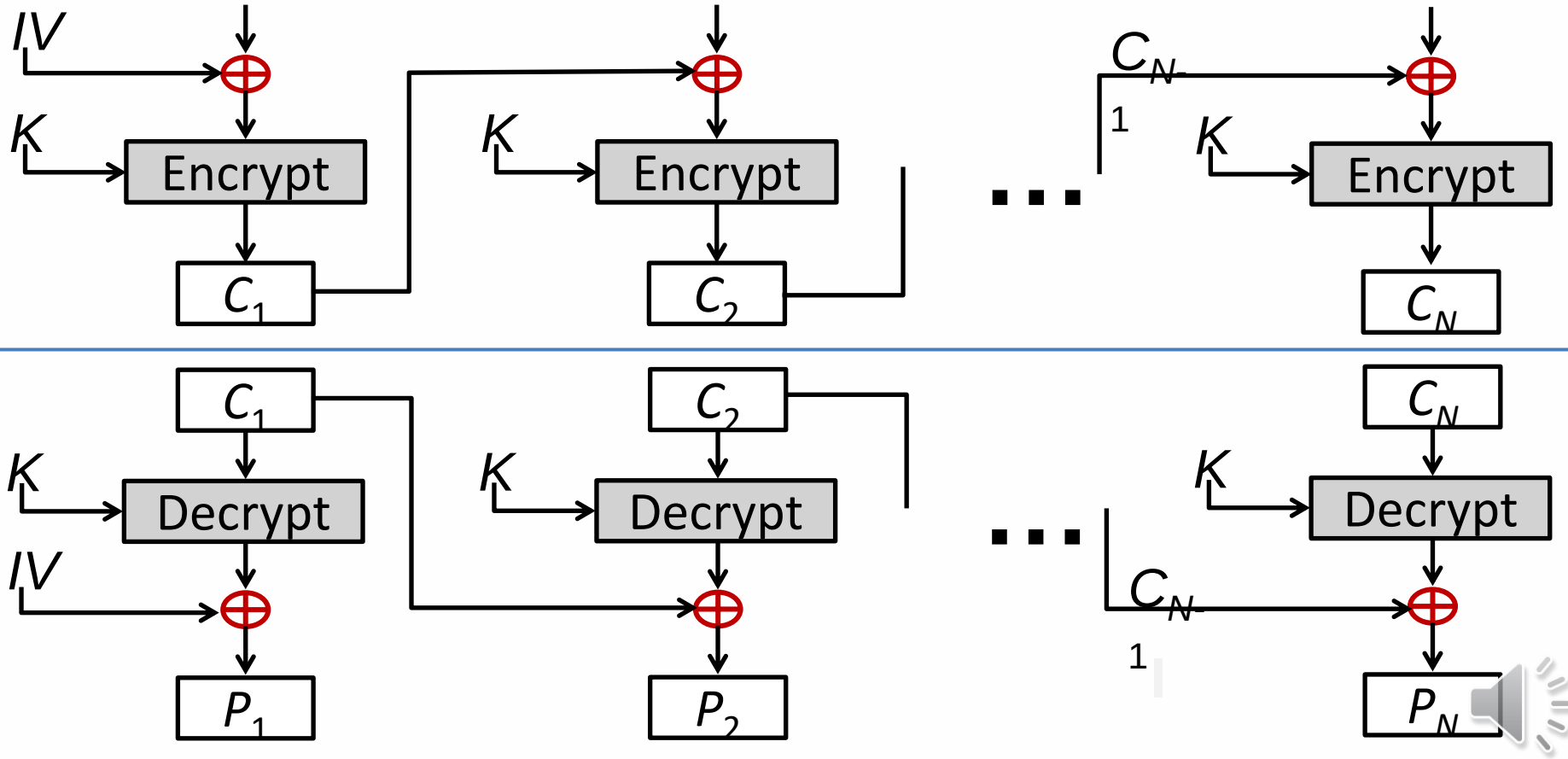
Electronic Code Book (Continue)

- Strength: it’s simple.
- Weakness:
- Repetitive information contained in the plaintext may show in the ciphertext, if aligned with blocks.
- If the message has repetitive elements with a period of repetition a multiple of b bits, then these elements can be identified by the analyst.
- Typical application:
- Secure transmission of short pieces of information (e.g. a temporary encryption key)

2. Cipher Block Chaining (CBC)

- CBC is a technique in which the same plaintext block, if repeated, produces different ciphertext blocks.
- In this scheme, the input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block; the same key is used for each block.
- To produce the first block of ciphertext, an initialization vector (IV) is XORed with the first block of plaintext.
- On decryption, the IV is XORed with the output of the decryption algorithm to recover the first block of plaintext.

CBC - Encryption & Decryption



Cipher Block Chaining (CBC) – Continue…

- Strength: because of the chaining mechanism of CBC, it is an appropriate mode for encrypting messages of length greater than b bits
- Typical application:
- General-purpose block oriented transmission
- Authentication

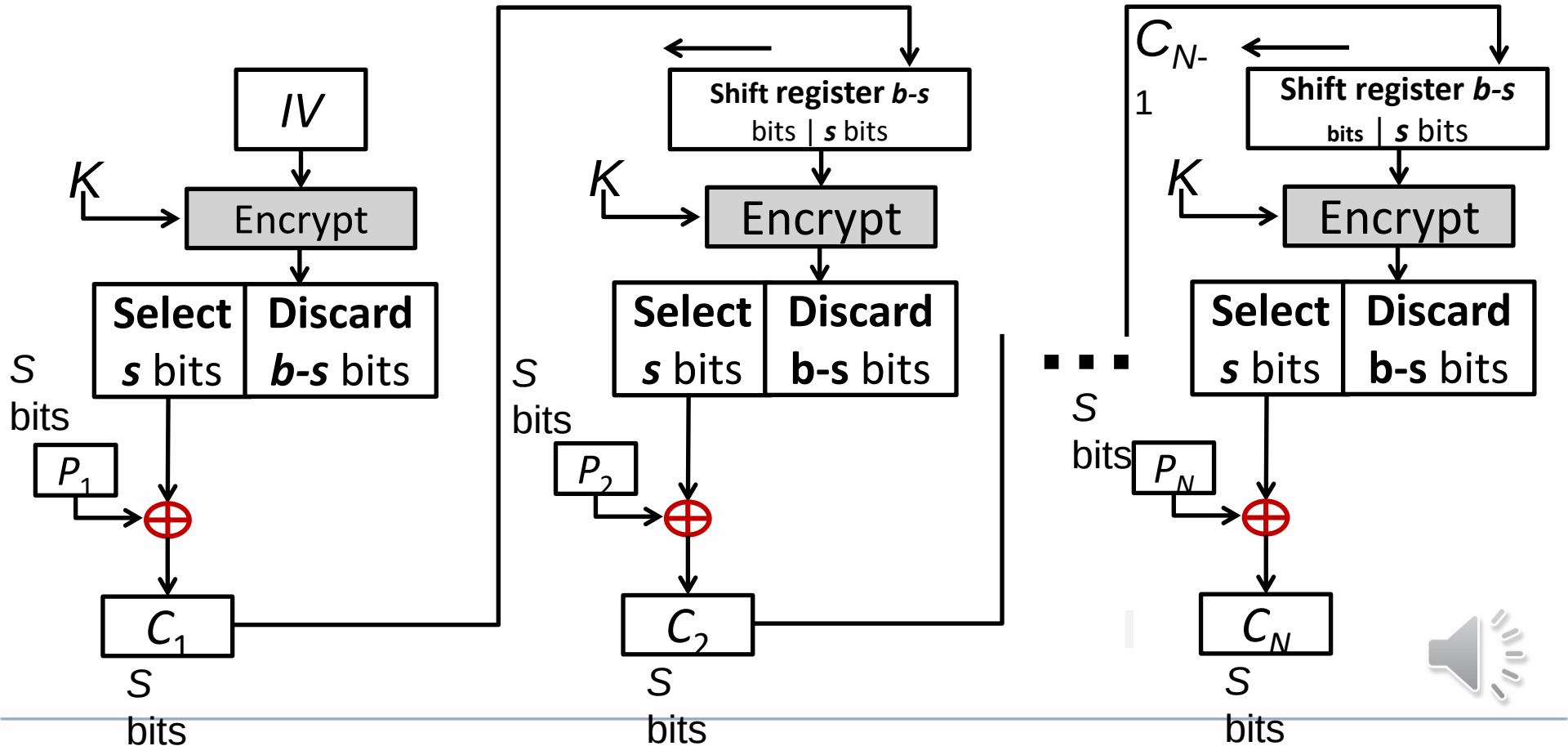
3. Cipher Feedback Mode (CFB)

- For AES, DES, or any block cipher, encryption is performed on a block of b bits. In DES, b = 64 and in AES, b = 128.

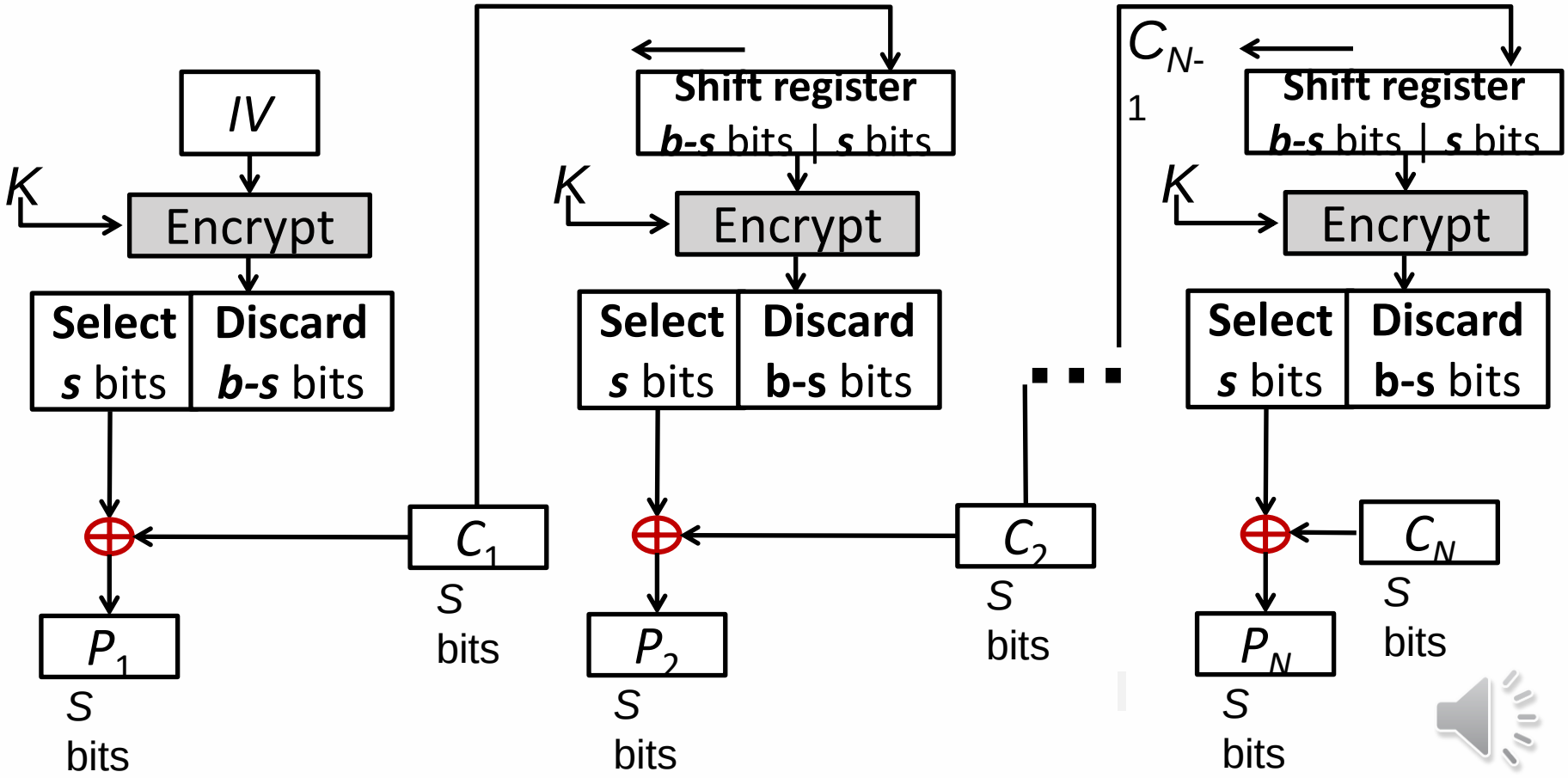
- However, it is possible to convert a block cipher into a stream cipher, using cipher feedback (CFB) mode, output feedback (OFB) mode, and counter (CTR) mode.

- A stream cipher eliminates the need to pad a message to be an integral number of blocks.

CFB Encryption



CFB Decryption



CFB Mode

- The input to the encryption function is a b-bit shift register that is initially set to some initialization vector (IV).
- The leftmost (most significant) s bits of the output of the encryption function are XORed with the first segment of plaintext P1 to produce the first unit of ciphertext C1 , which is then transmitted.
- In addition, the contents of the shift register are shifted left by s bits, and C1 is placed in the rightmost (least significant) s bits of the shift register.
- For decryption, the same scheme is used, except that the received ciphertext unit is XORed with the output of the encryption function to produce the plaintext unit.

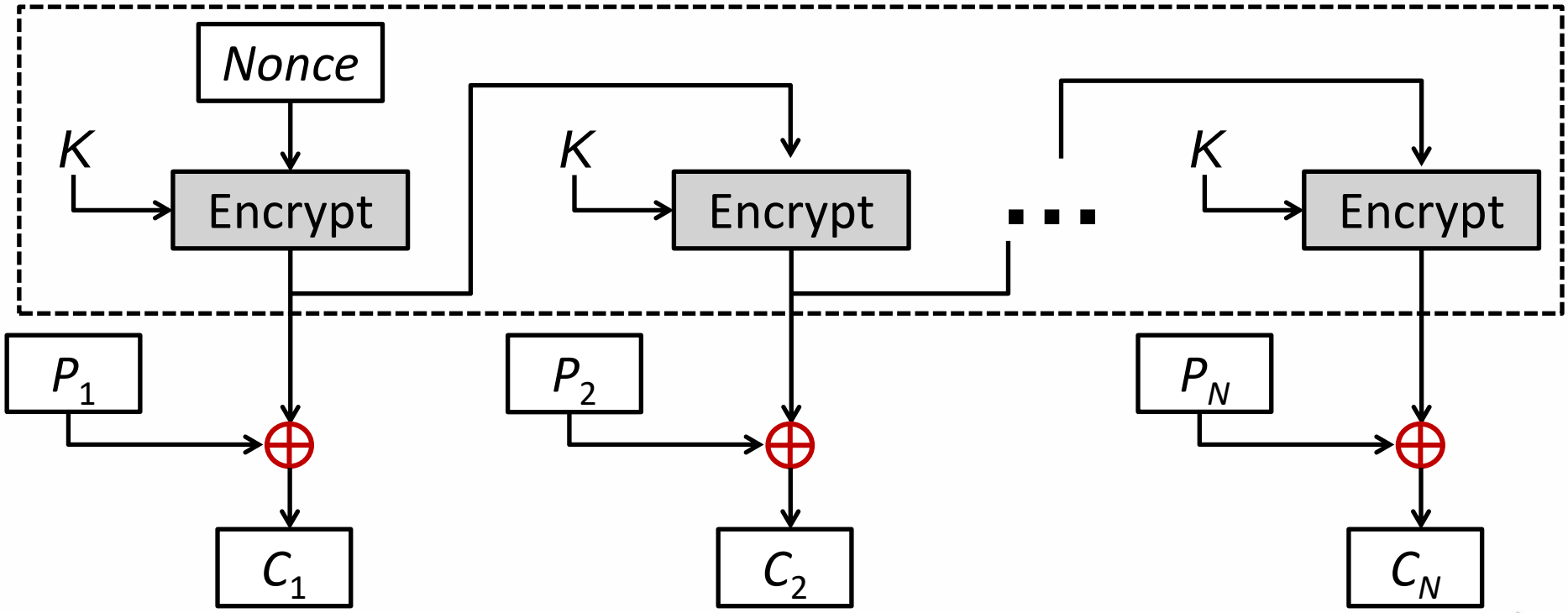
CFB Mode – Cont…

CFB	$I_1 = IV$ $I_j = \text{LSB}_{b-s}(I_{j-1}) \| C_{j-1}$ $j = 2, \ldots, N$ $O_j = \text{E}(K, I_j)$ $j = 1, \ldots, N$ $C_j = P_j \oplus \text{MSB}_s(O_j)$ $j = 1, \ldots, N$	$I_1 = IV$ $I_j = \text{LSB}_{b-s}(I_{j-1}) \| C_{j-1}$ $j = 2, \ldots, N$ $O_j = \text{E}(K, I_j)$ $j = 1, \ldots, N$ $P_j = C_j \oplus \text{MSB}_s(O_j)$ $j = 1, \ldots, N$

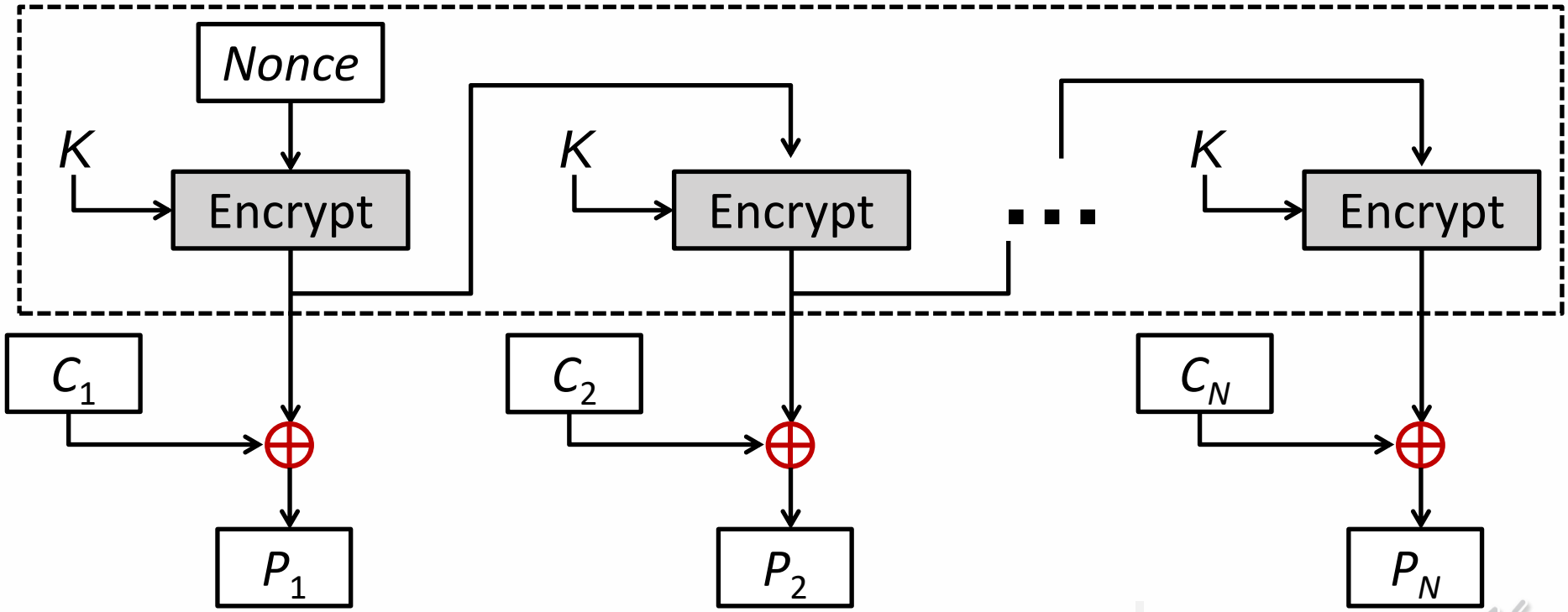
4. Output Feedback Mode (OFB)

- The output feedback (OFB) mode is similar in structure to that of CFB.
- For OFB, the output of the encryption function is fed back to become the input for encrypting the next block of plaintext.
- In CFB, the output of the XOR unit is fed back to become input for encrypting the next block.
- The other difference is that the OFB mode operates on full blocks of plaintext and ciphertext, whereas CFB operates on an s-bit subset.
- Nonce: A time-varying value that has at most a negligible chance of repeating, for example, a random value that is generated anew for each use, a timestamp, a sequence number, or some combination of these.

OFB Encryption



OFB Decryption



OFB Mode

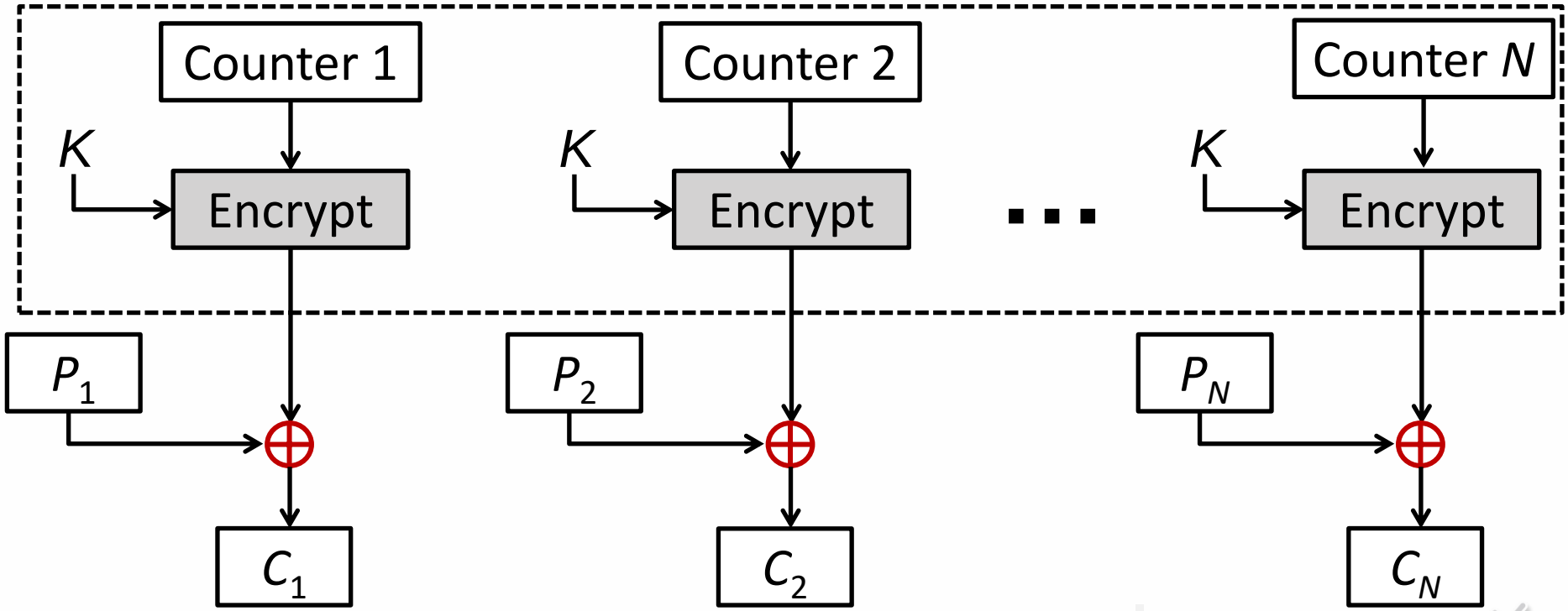
- Each bit in the ciphertext is independent of the previous bit or bits.
- This avoids error propagation
- Pre-compute of forward cipher is possible

OFB	$I_1 = Nonce$ $I_j = O_{j-1} \quad j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j) \quad j = 1, \ldots, N$ $C_j = P_j \oplus O_j \quad j = 1, \ldots, N-1$ $C_N^* = P_N^* \oplus \mathrm{MSB}_u(O_N)$	$I_1 = Nonce$ $I_j = O_{j-1} \quad j = 2, \ldots, N$ $O_j = \mathrm{E}(K, I_j) \quad j = 1, \ldots, N$ $P_j = C_j \oplus O_j \quad j = 1, \ldots, N-1$ $P_N^* = C_N^* \oplus \mathrm{MSB}_u(O_N)$

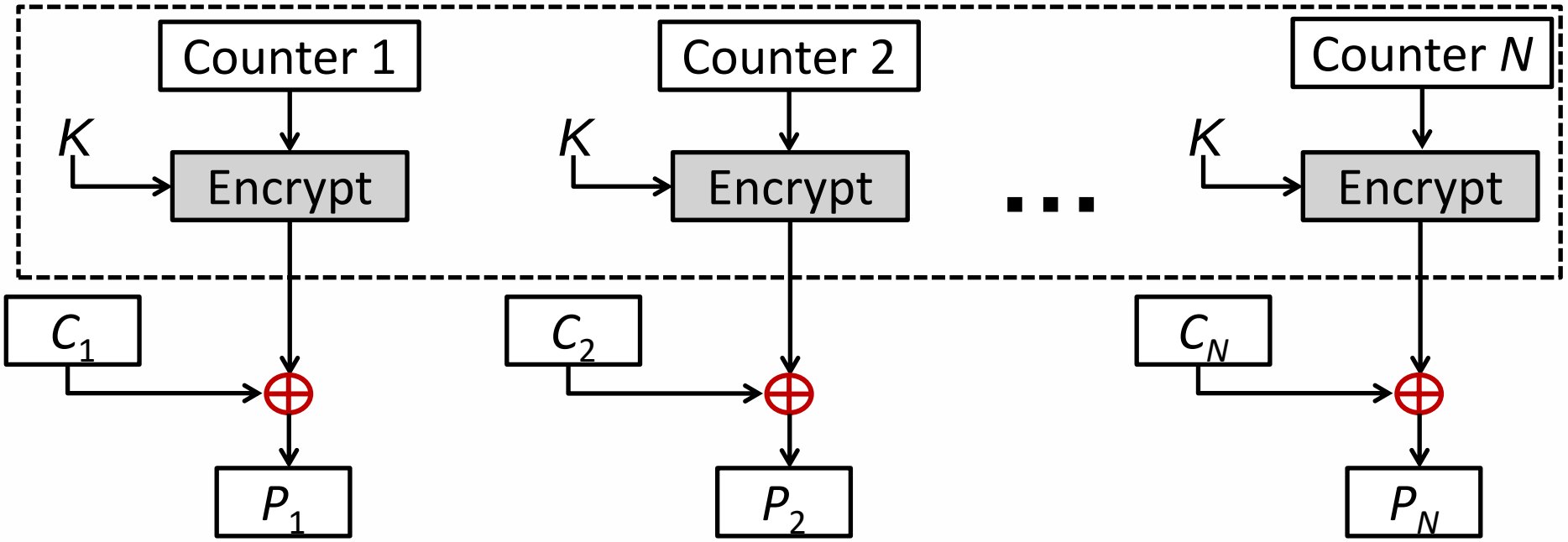
5. Counter Mode (CTR)

- Counter (CTR) mode has increased recently with applications to ATM (asynchronous transfer mode) network security and IP sec (IP security).
- A counter equal to the plaintext block size is used.
- The counter value must be different for each plaintext block that is encrypted.
- Typically, the counter is initialized to some value and then incremented by 1 for each subsequent block

CTR Encryption



CTR Decryption



CTR	$C_j = P_j \oplus \mathrm{E}(K, T_j) \quad j = 1, \ldots, N-1$ $C_N^* = P_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$	$P_j = C_j \oplus \mathrm{E}(K, T_j) \quad j = 1, \ldots, N-1$ $P_N^* = C_N^* \oplus \mathrm{MSB}_u[\mathrm{E}(K, T_N)]$

Parul®
University

Advantages of the CTR Mode

- Strengths:
- Needs only the encryption algorithm
- Random access to encrypted data blocks
- blocks can be processed (encrypted or decrypted) in parallel
- Simple; fast encryption/decryption

- Counter must be
- Must be unknown and unpredictable
- pseudo-randomness in the key stream is a goal

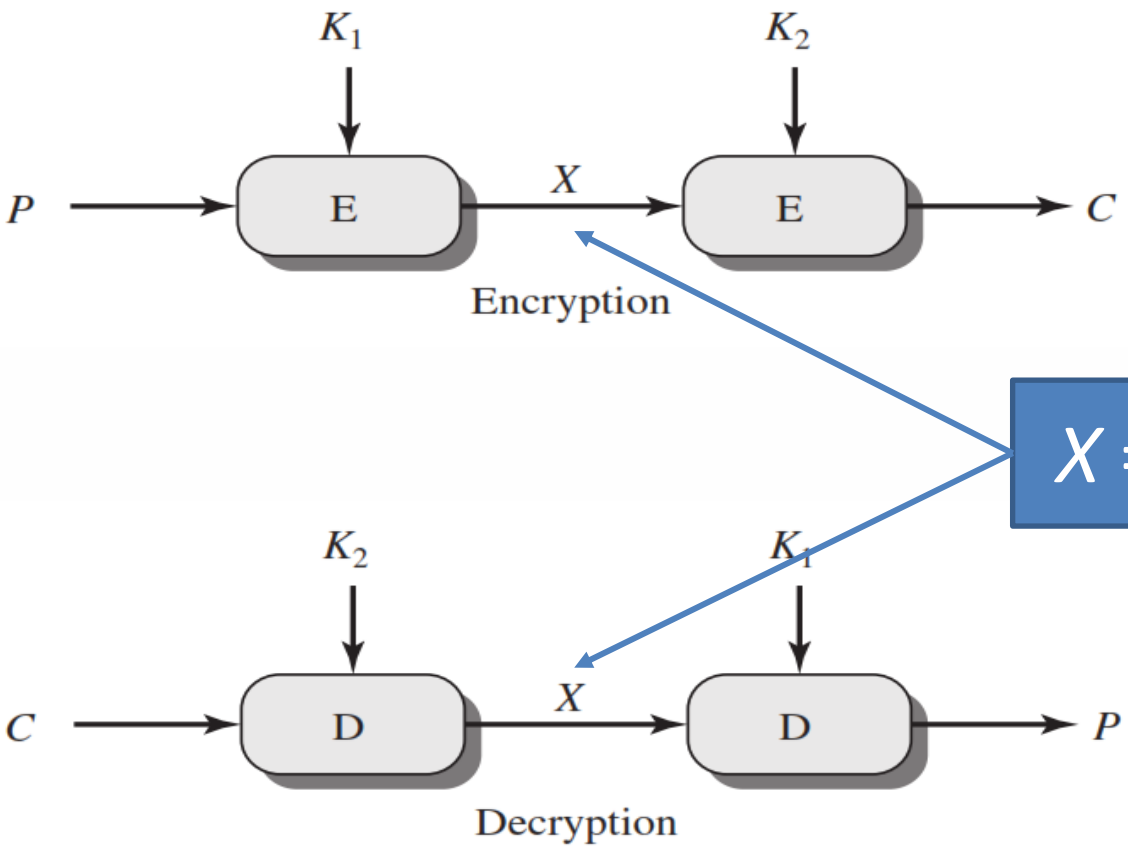
Summary of all modes

Operation Mode	Description	Type of Result
ECB	Each n-bit block is encrypted independently with same key	Block Cipher
CBC	Same as ECB, but each block is XORed with previous cipher text	Block Cipher
CFB	Each s-bit block is XORed with s-bit key which is part of previous cipher text	Stream Cipher
OFB	Same as CFB, but the shift register is updated by the previous s-bit key	Stream Cipher
CTR	Same as OFB, but a counter is used instead of nonce	Stream Cipher

Multiple Encryption

- Given the potential vulnerability of DES to a brute-force attack, there has been considerable interest in finding an alternative.
- One approach is to design a completely new algorithm, of which AES is a prime example.
- Another alternative, which would preserve the existing investment in software and equipment, is to use multiple encryption with DES and multiple keys.

Double DES



$$C=E(K_2,E(K_1, P))$$

$$X = E(K_1, P) = D(K_2, C)$$

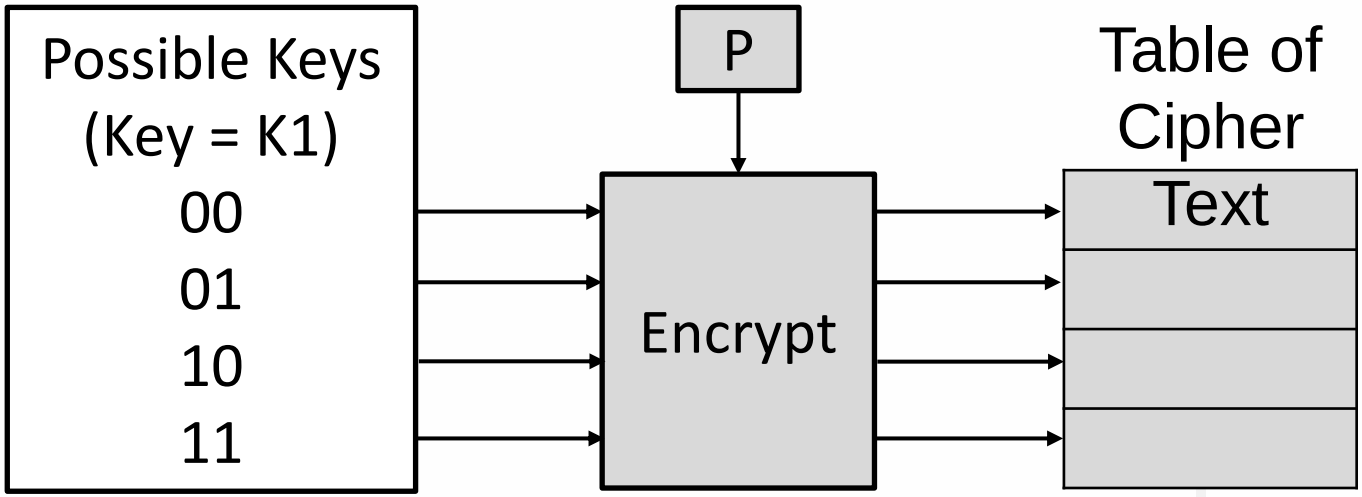
$$P=D(K_1,D(K_2, C))$$

Meet in the Middle Attack

- This attack involves encryption from one end, decryption from the other and matching the results in the middle.
- Suppose cryptanalyst knows P and corresponding C.
- Now, the aim is to obtain the values of K1 and K2.

Meet in the Middle Attack Step-1

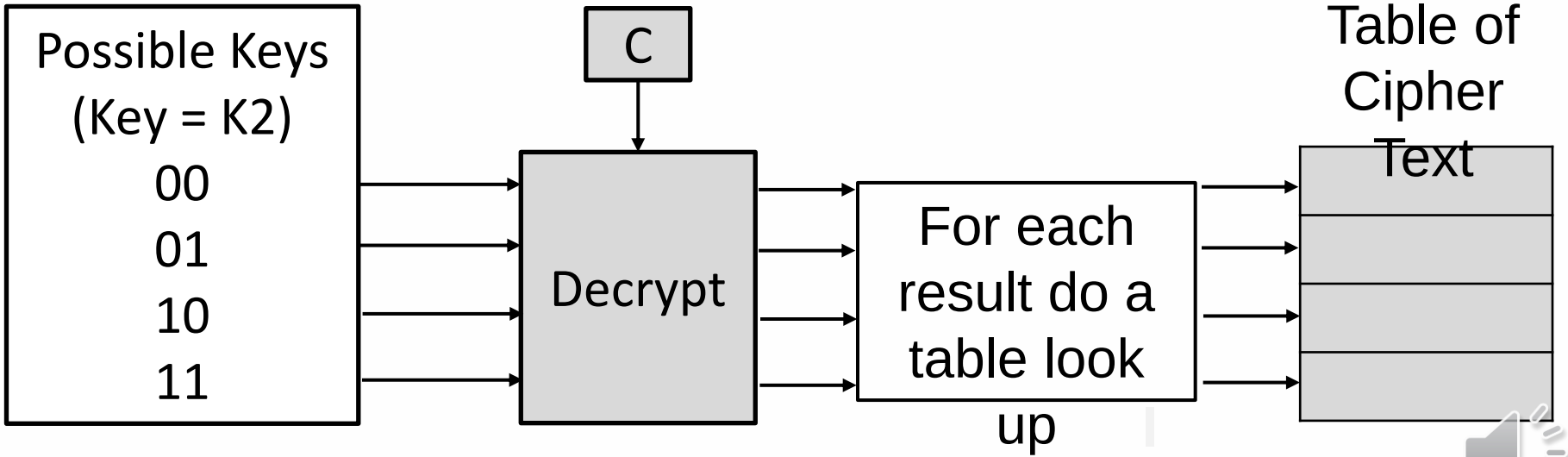
- For all possible values (256) of key K1, the cryptanalyst would encrypt the P by performing E(K1,P).
- The cryptanalyst would store output in a table.



- Cryptanalyst encryption operation

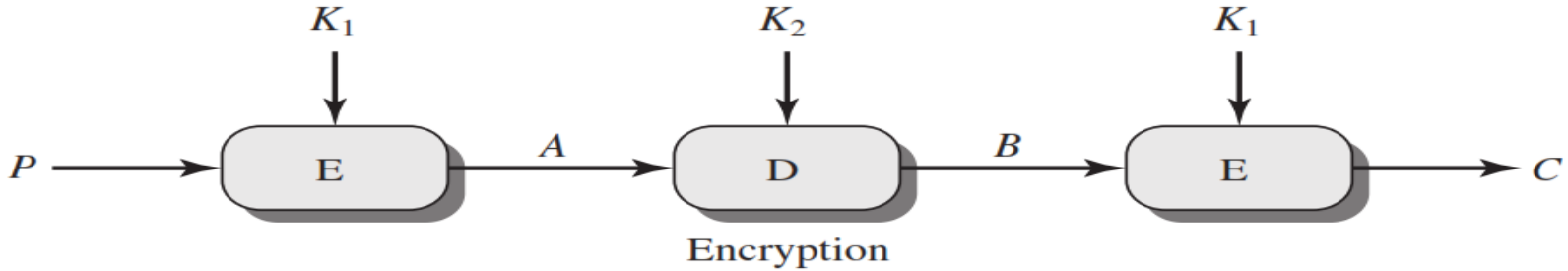
Meet in the Middle Attack Step-2

- Cryptanalyst decrypt the known C with all possible values of K2.
- In each case cryptanalyst will compare the resulting value with the all values in the table of ciphertext.

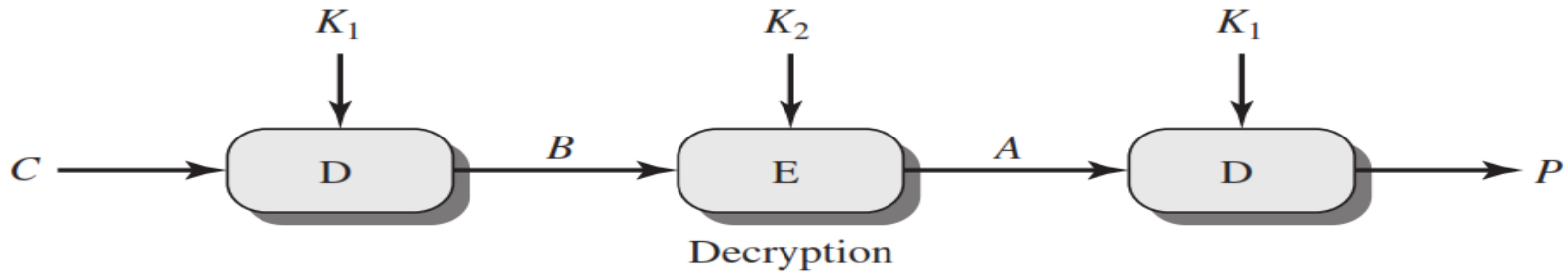


- Cryptanalyst decryption operation

Triple DES



$$C=E(K_1,D(K_2,\ E(K_1,P)))$$



$$P=D(K_1,E(K_2,\ D(K_1,C)))$$

What is RC4 Encryption?

RC4 means Rivest Cipher 4 invented by Ron Rivest in 1987 for RSA Security. It is a Stream Ciphers. Stream Ciphers operate on a stream of data byte by byte. RC4 stream cipher is one of the most widely used stream ciphers because of its simplicity and speed of operation. It is a variable key-size stream cipher with byte-oriented operations. It uses either 64 bit or 128-bit key sizes. It is generally used in applications such as Secure Socket Layer (SSL), Transport Layer Security (TSL), and also used in IEEE 802.11 wireless LAN std.

Why Encryption Is Important?

Unauthorized data access can be prevented by encryption. If we perform encryption then third parties can not have access to data which we share or receive. The encryption is done by using a secret key, or we can say that by using a public key and private key. Both sender and receiver are having their public key and private key through which encryption of plain text and decryption of ciphertext is performed.

Types of RC4

There are various types of RC4 such as Spritz, RC4A, VMPC, and RC4A.

SPRITZ: Spritz can be used to build a cryptographic hash function, a deterministic random bit generator (DRBG), n an encryption algorithm that supports authenticated encryption with associated data (AEAD).
RC4A: Souraduyti Paul and Bart Preneel have proposed an RC4 variant, which they call RC4A, which is stronger than RC4.
VMPC: VMPC is another variant of RC4 which stands for Variably Modified Permutation Composition.
RC4A+: RC4A+ is a modified version of RC4 with a more complex three-phase key schedule which takes about three times as long as RC4 and a more complex output function which performs four additional lookups in the S array for each byte output, taking approximately 1.7 times as long as basic RC4.

Algorithm

The algorithm operates on a user-selected variable-length key(K) of 1 to 256 bytes (8 to 2048 bits), typically between 5 and 16 bytes. To generate a 256-byte state vector S, the master key is used.
The first step is the array initialization. It is a character array of size 256 i.e. S[256]. After that, for every element of the array, we initialize S[i] to i.

Code for array initialization: Char S[256]; int i; for(i=0;i<256;i++) S[i] = i The array will look like - S[] = {0, 1, 2, 3, ------, 254, 255}
After this, we will run the KSA algorithm-
KSA is going to use the secret key to scramble this array. KSA is a simple loop, in which we are having two variable i and j. We are using these variables to rearrange the array. Rearranging the array is done by using a secret key.

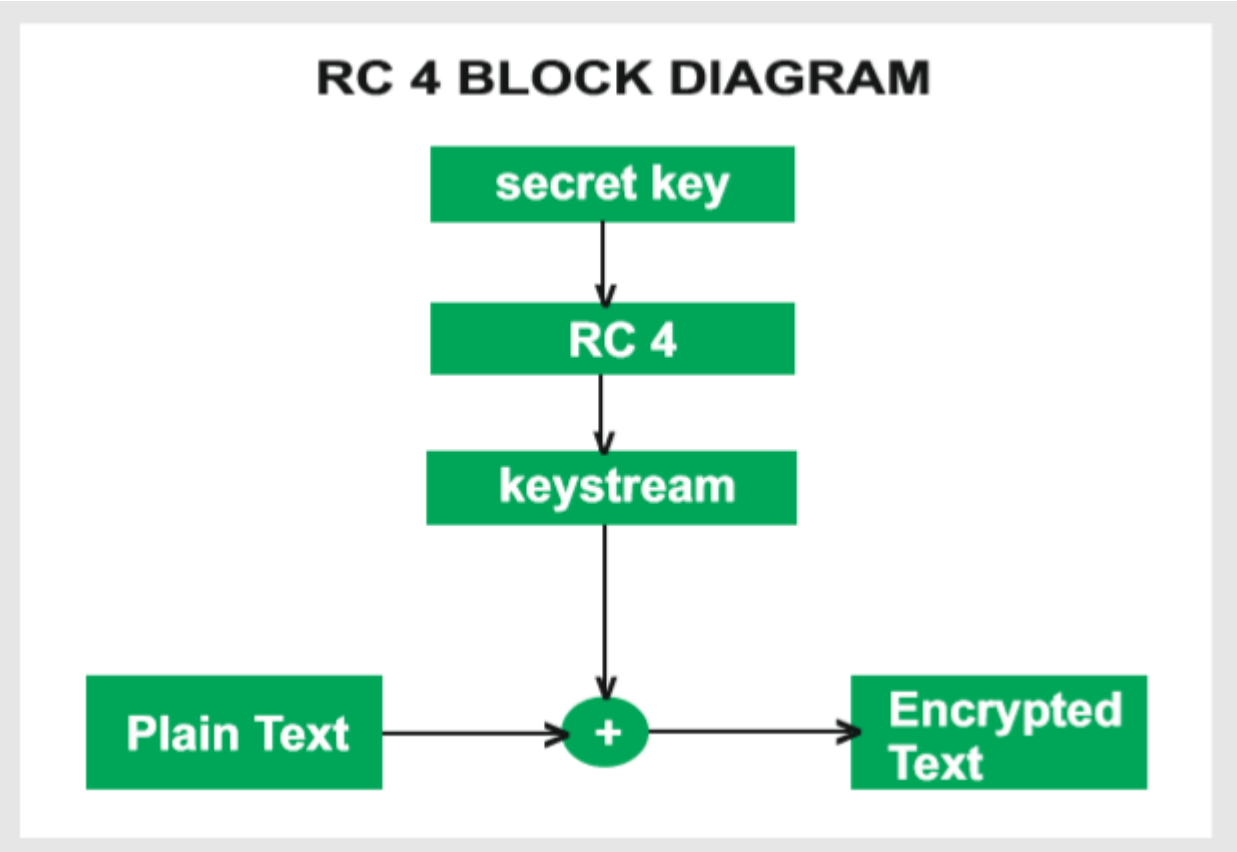
Algorithm

Code for KSA (Key Scheduling Algorithm) : int i, j=0; for(i=0;i<256;i++) { j=(j + S[i] + T[i]) mod 256; Swap(S[i], S[j]); }

KSA has been scrambled, S[256] array is used to generate the PRGA(Pseudo Random Generation Algorithm). This is the actual Keystream.

Code for PRGA (Pseudo Random Generation Algorithm): i=j=0; while(true) { i = (i + 1) mod 256; j = (j + S[i]) mod 256; Swap(S[i], S[j]); t = (S[i] + S[j]) mod 256 ; k = S[t]; } This is the next step of scrambling.

RC4 Block Diagram



Working of RC4

Encryption Procedure:

The user inputs a plain text file and a secret key.

The encryption engine then generates the keystream by using KSA and PRGA Algorithm.

This keystream is now XOR with the plain text, this XORing is done byte by byte to produce the encrypted text.

The encrypted text is then sent to the intended receiver, the intended receiver will then decrypted the text and after decryption, the receiver will get the original plain text.

Decryption Procedure:

Decryption is achieved by doing the same byte-wise X-OR operation on the Ciphertext.

Example: Let A be the plain text and B be the keystream (A xor B) xor B = A

Advantages and Disadvantages

Advantages:
RC4 stream ciphers are simple to use.
The speed of operation in RC4 is fast as compared to other ciphers.
RC4 stream ciphers are strong in coding and easy to implement.
RC4 stream ciphers do not require more memory.
RC4 stream ciphers are implemented on large streams of data.
Disadvantages:
If RC4 is not used with strong MAC then encryption is vulnerable to a bit-flipping attack.
RC4 stream ciphers do not provide authentication.
RC4 algorithm requires additional analysis before including new systems.
RC4 stream ciphers cannot be implemented on small streams of data.
RC4 fails to discard the beginning of output keystream or fails to use non-random or related keys for the algorithm.

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in