Information Security

CS3002

Lecture 4

4th September 2023

Dr. Rana Asif Rehman

Email: r.asif@nu.edu.pk

Why need bit-oriented ciphers?

- In previous chapters we discussed *character-oriented ciphers the problem is*
  - Keep language statistics
- Need bit-oriented ciphers
  - Numbers, graphics, audio and video data
  - Mixing at larger number of symbols increases security

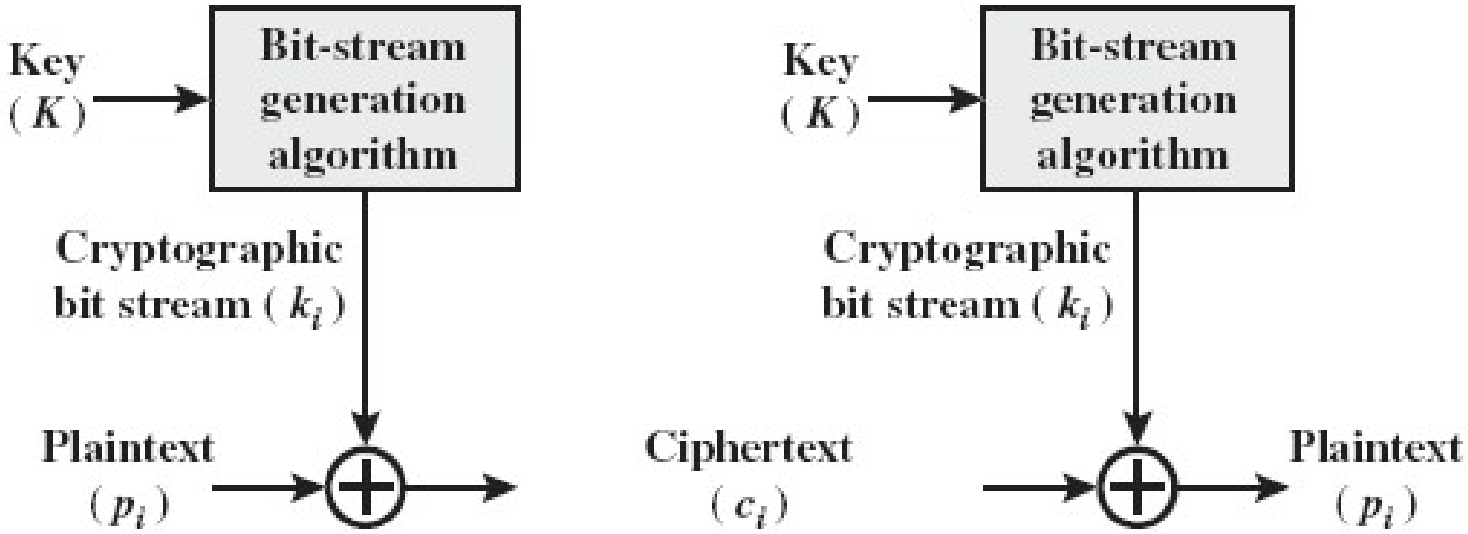
Stream and Block Ciphers

- Way in which plaintext is processed
  - Blockwise
  - Streamwise

Stream Ciphers

- Stream ciphers process messages a *bit or byte* at a time when en/decrypting
  - Vernam Cipher
- If the cryptographic keystream is random, cipher is unbreakable.
- Bit-stream generator is implemented as an algorithm, can be produced by both users.
- In this approach the bit-stream generator is a key-controlled algorithm and must produce a bit stream that is cryptographically strong.

Stream Ciphers

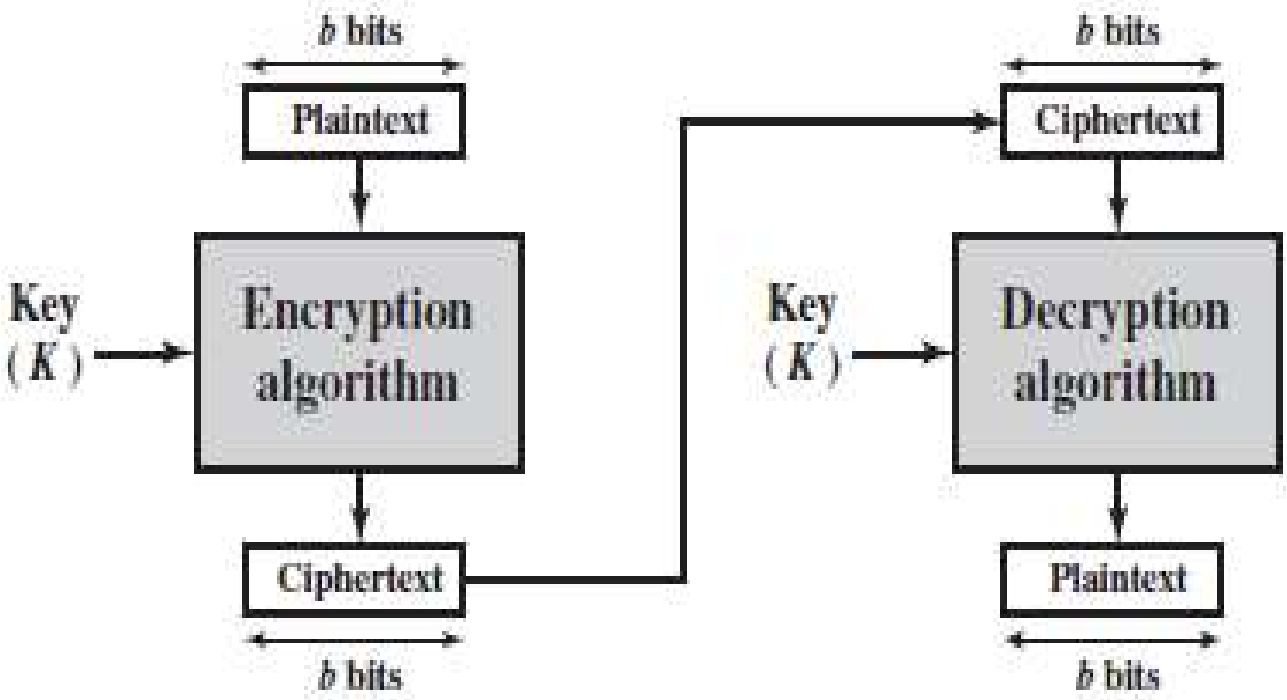


(a) Stream cipher using algorithmic bit-stream generator

Block Ciphers

- Block ciphers process messages in blocks, each of which is then en/decrypted
- Like a substitution on very big characters
  - 64-bits or 128
- Many current ciphers are block ciphers.
- Broader range of applications.

Block Ciphers



Feistel Proposal

- Feistel proposed that we can approximate the ideal block cipher by utilizing the concept of a product cipher.
- The essence of the approach is to develop a block cipher with a key length of k bits and a block length of n bits, allowing a total of 2^k possible transformations, rather than the $2^n!$ transformations available with the ideal block cipher.
- He proposed the use of a cipher that alternates substitutions and permutations.
- Feistel's is a practical application of a proposal by Claude Shannon.

Claude Shannon and Substitution-Permutation Ciphers

- Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper
- Form basis of modern block ciphers
- S-P nets are based on the two primitive cryptographic operations :
  - *substitution* (S-box)
  - *permutation* (P-box)
- provide *confusion* & *diffusion* of message & key

Confusion and Diffusion

- Confusion: An encryption operation where the relationship between key and ciphertext is obscured.
  - Today, a common element for achieving confusion is substitution, which is found in both AES and DES
- Diffusion: An encryption operation where the influence of one plaintext symbol is spread over many ciphertext symbols with the goal of hiding statistical properties of the plaintext.
  - A simple diffusion element is the bit permutation, which is frequently use

Credit: Chapter 3 of Understanding Cryptography by Christof Paar and Jan Pelzl

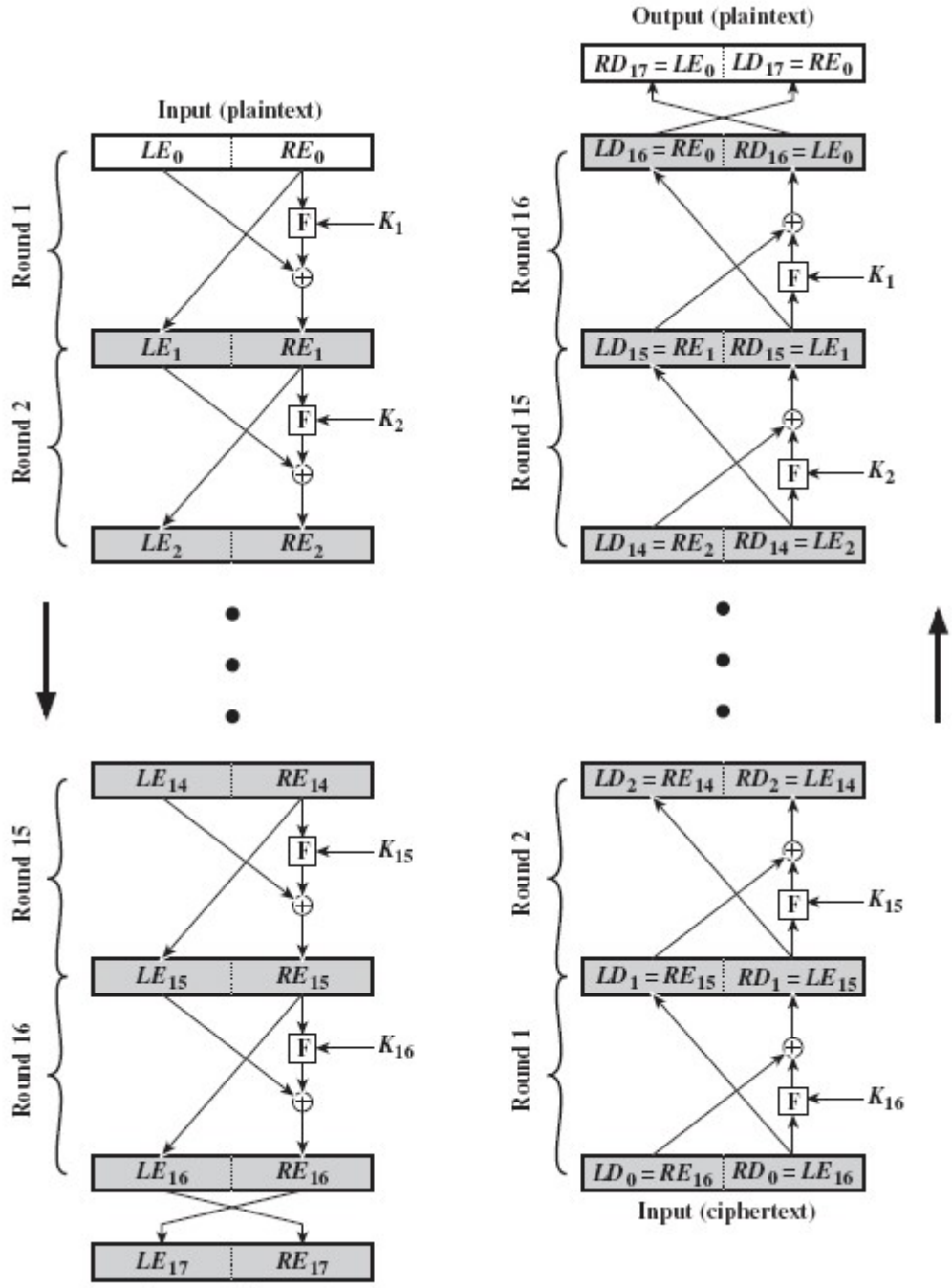
Feistel Cipher

- Most algorithms based on a structure referred to as Feistel block cipher.
- Build strong cipher from a product of multiple simple ciphers.
- Alternate substitutions & permutations.
- Key length *k*, block length *n.*
- Limit to 2^k possible transformations, rather than $2^n!$.

Feistel Cipher Structure

- Input
  - plaintext block of length $2w$
  - key K
- Plaintext block divided to L_0, R_0
- Pass through n rounds of processing
- Each round i has
  - L_{i-1}, R_{i-1} derived from previous round
  - subkey K_i derived from overall K

Feistel encryption and decryption



Feistel Cipher Structure

- Substitution
  - Apply round function F to right half.
  - Take XOR of output with left half.
  - F is parameterized by round subkey K_i.
- Permutation
  - Interchange left and right halves.

Design Parameters (1/3)

- Block size
  - Larger: greater security (diffusion)
  - Smaller: faster encryption, decryption
  - Typical: 64 bit, 128 bit AES
- Key size
  - Larger: greater security (brute-force resist)
  - Smaller: faster encryption, decryption
  - Typical: 128 bit

Design Parameters (2/3)

- Number of rounds
  - Multiple rounds increase security
  - Typical: 16
- Subkey generation algorithm
  - Complexity makes cryptanalysis difficult
- Round function
  - Complexity makes cryptanalysis difficult

Design Parameters (3/3)

- Speed of execution
  - Required for embedded systems
- Ease of analysis
  - Algorithm easy to understand is easy to identify vulnerabilities
  - DES isn't easy to analyze

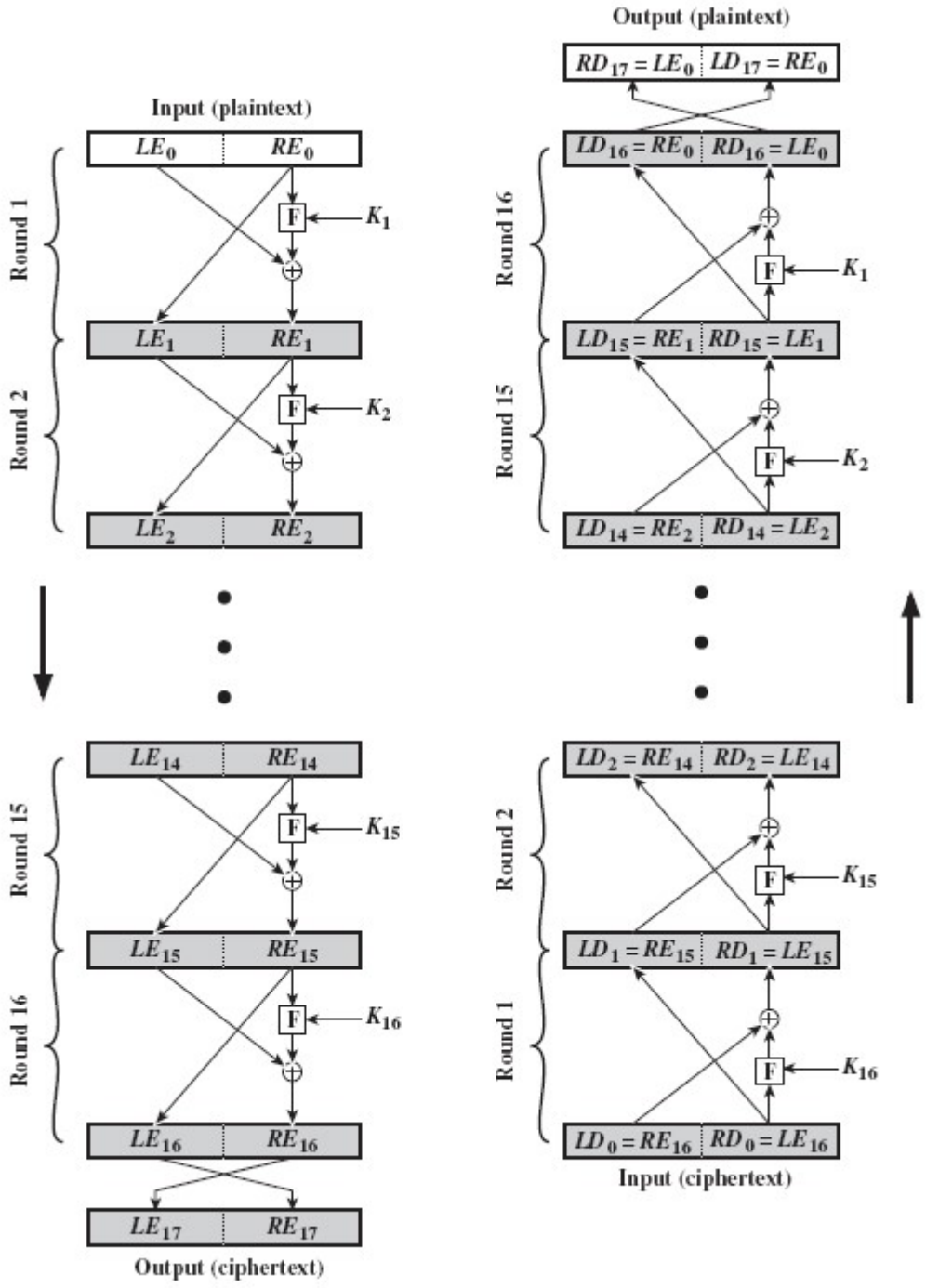
Feistel Cipher Decryption

- Ciphertext is used as input
- Use subkeys K_i in reverse order
- Same algorithm is used
- Notation
  - LE_i : left half in encryption algorithm
  - RE_i : right half in encryption algorithm
  - LD_i : left half in decryption algorithm
  - RD_i : right half in decryption algorithm

Feistel Cipher Decryption

- Output of i^{th} encryption round → input to i^{th} decryption round swapped
- $LE_i || RE_i \equiv RD_{16-i} || LD_{16-i}$

Feistel Cipher Decryption



Decryption Proof (1/2)

- Encryption side
  - $LE_{16} = RE_{15}$
  - $RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16})$
- Decryption side
  - $LD_1 = RD_0 = LE_{16} = RE_{15}$
  - $LD_0 = RE_{16}$
  - $RD_1 = LD_0 \oplus F(RD_0, K_{16})$
    $= RE_{16} \oplus F(RE_{15}, K_{16})$
    $= [LE_{15} \oplus F(RE_{15}, K_{16})] \oplus F(RE_{15}, K_{16})$
    $RD_1 = LE_{15}$

Decryption Proof (2/2)

- XOR properties
  - [A $\oplus$ B] $\oplus$C = A $\oplus$ [B $\oplus$ C]
  - D $\oplus$ D = 0
  - E $\oplus$ 0 = E
- Thus
- RD1 = [LE15 $\oplus$ F(RE15, K16)] $\oplus$ F(RE15, K16)

  = LE15
- F does not have to be reversible

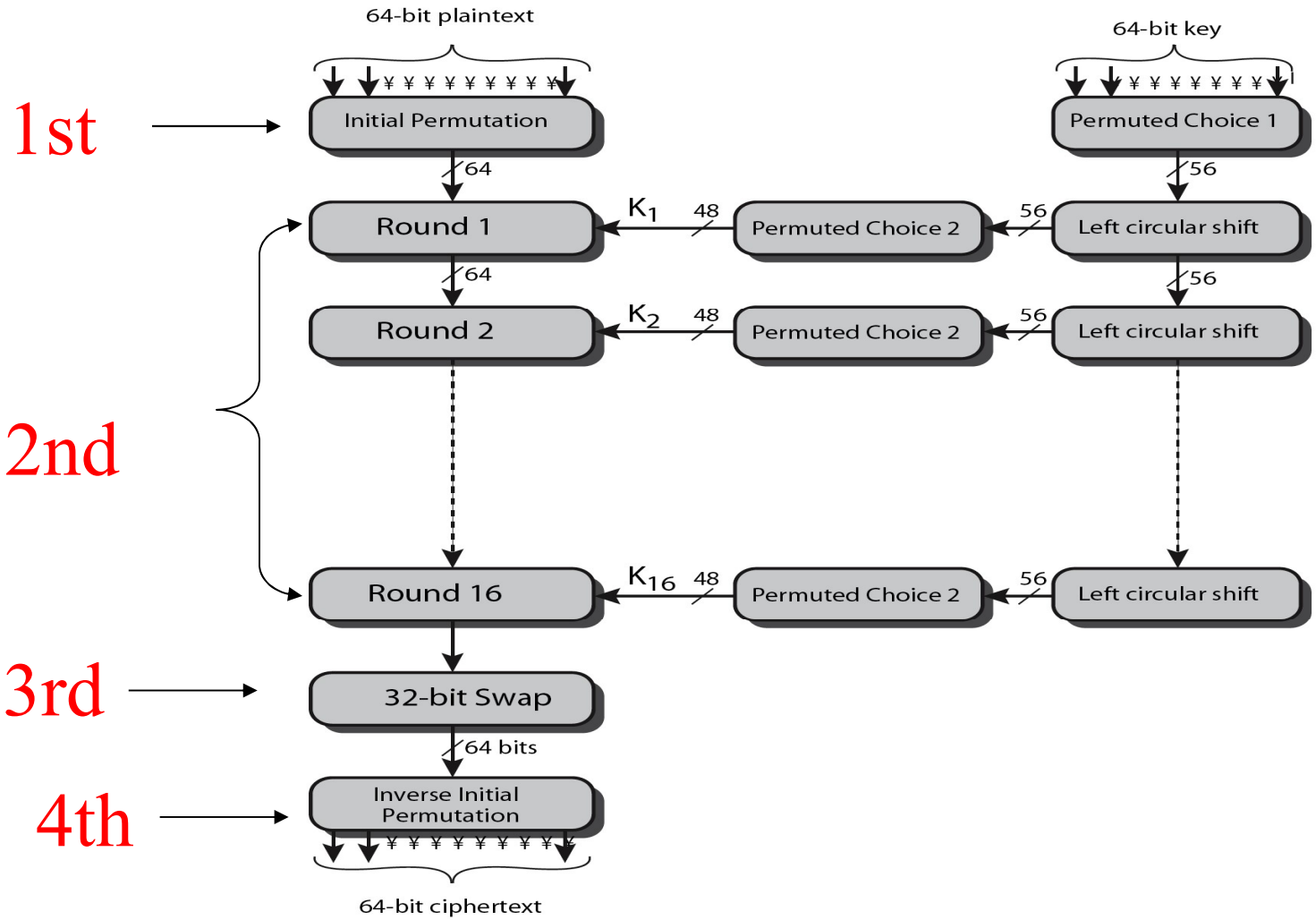
Data Encryption Standard (DES)

Data Encryption Standard (DES)

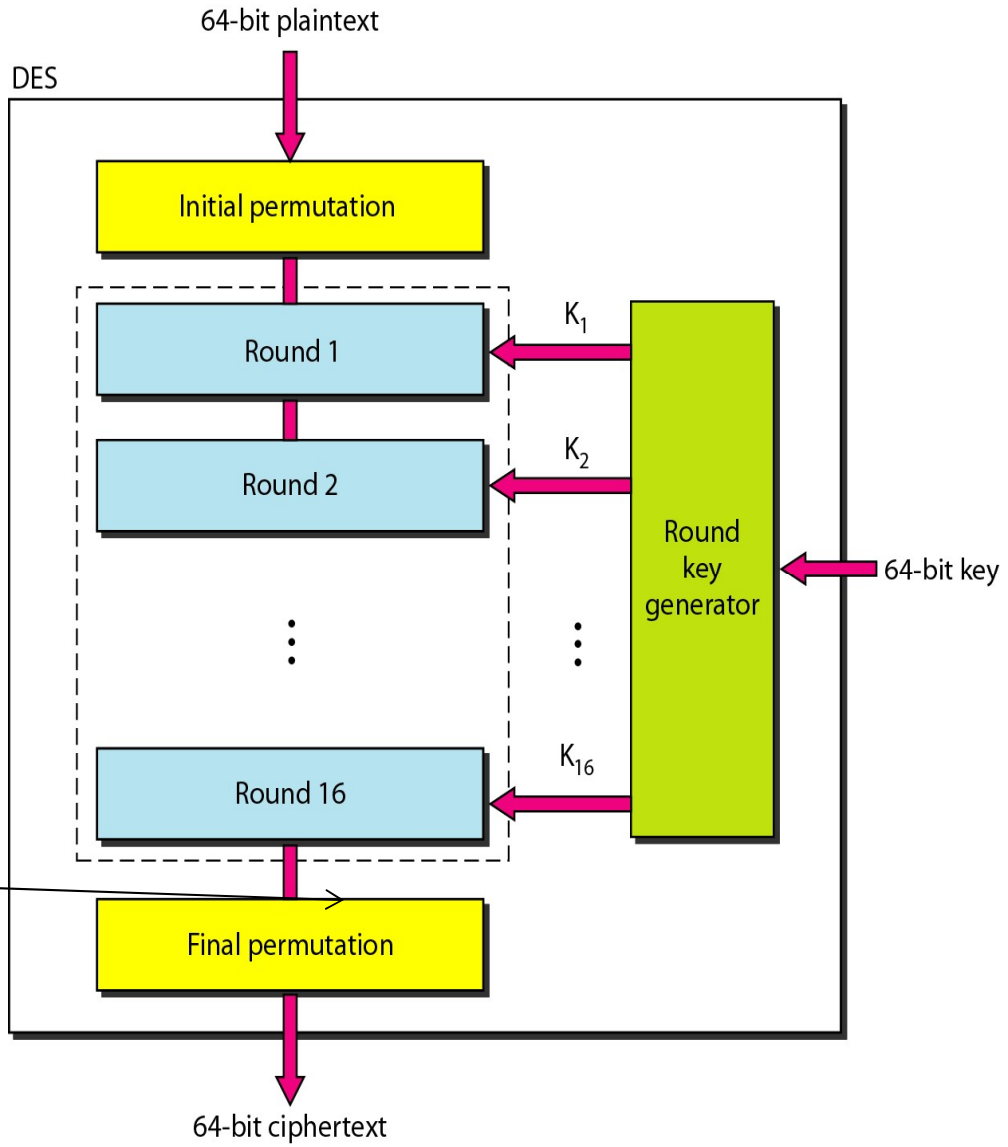
- Most widely used block cipher in world
- Adopted in 1977 by NBS (now NIST)
  - as FIPS PUB 46
- Encrypts 64-bit data using 56-bit key
- Exact structure as Feistel except
  - Initial permutation of plaintext
  - Final permutation of last round's out

- http://des.online-domain-tools.com/

DES Encryption Overview



Another View

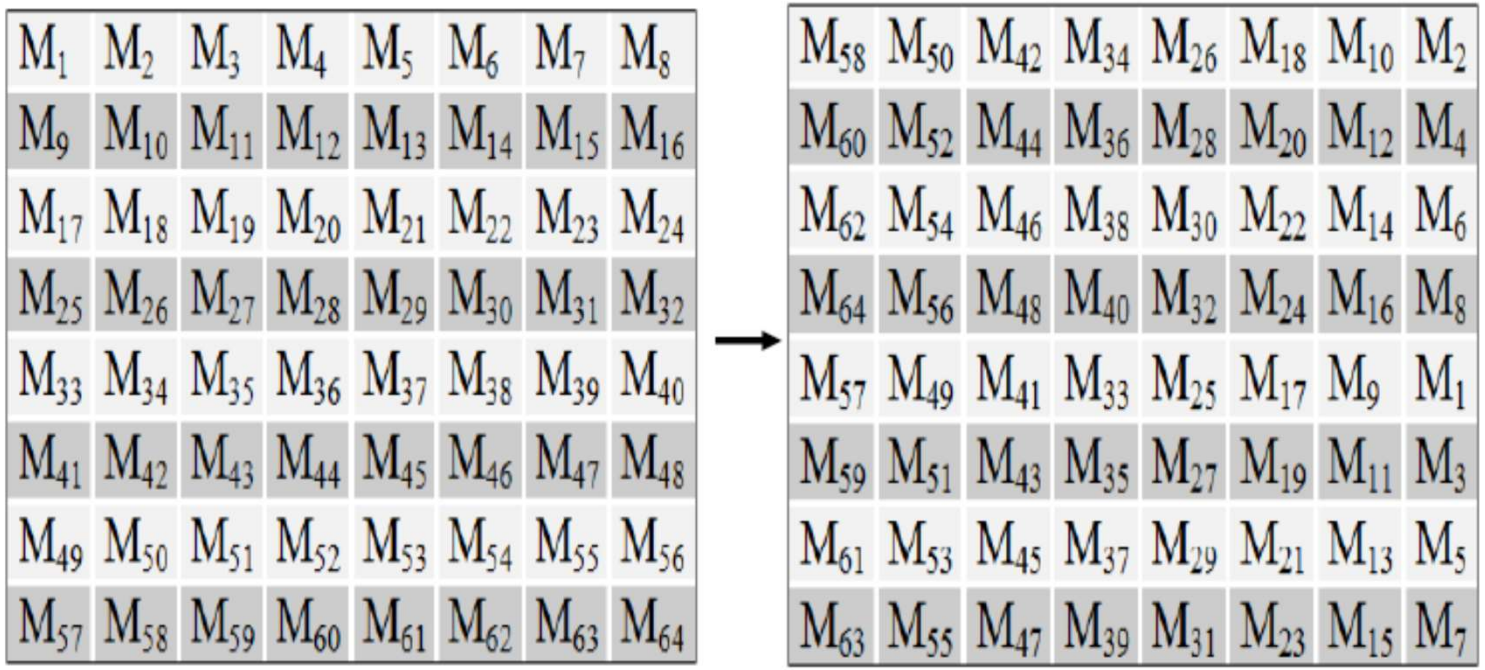


Missing Swap block

DES Encryption Four Phases

- 64 bit plaintext pass thru initial permutation (IP).
  - Rearrange bit to produce permuted input.
- Followed by 16 rounds of same function.
  - Involve permutation & substitution functions.
- Output of last round swapped (LH, RH) to produce preoutput.
- Preoutput pass thru a permutation (IP^{-1}).
  - Inverse of IP to produce 64 bit ciphertext.

Initial Permutation (IP) 1st



example:

IP(675a6967 5e5a6b5a) = (ffb2194d 004df6fb)

verify (?)

Inverse Initial Permutation (IP^{-1})

1st

(a) Initial Permutation (IP)

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

(b) Inverse Initial Permutation (IP^{-1})

4th

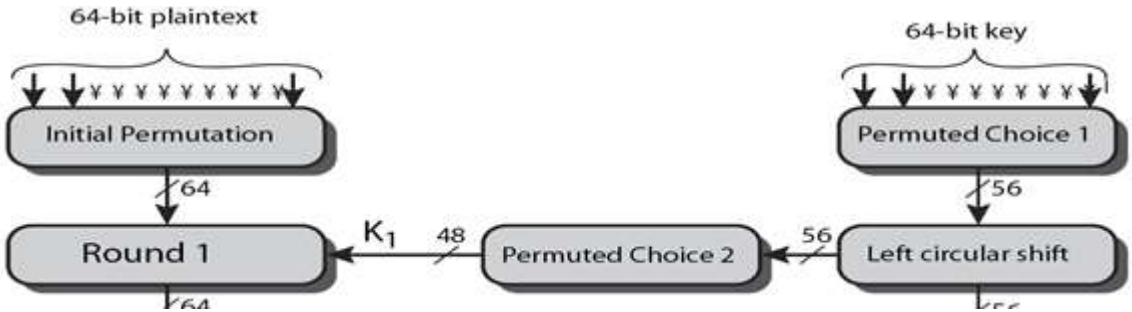
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Details of Single Round 2nd

- Left and right halves of 64 bits are separated into two 32-bit parts L,R

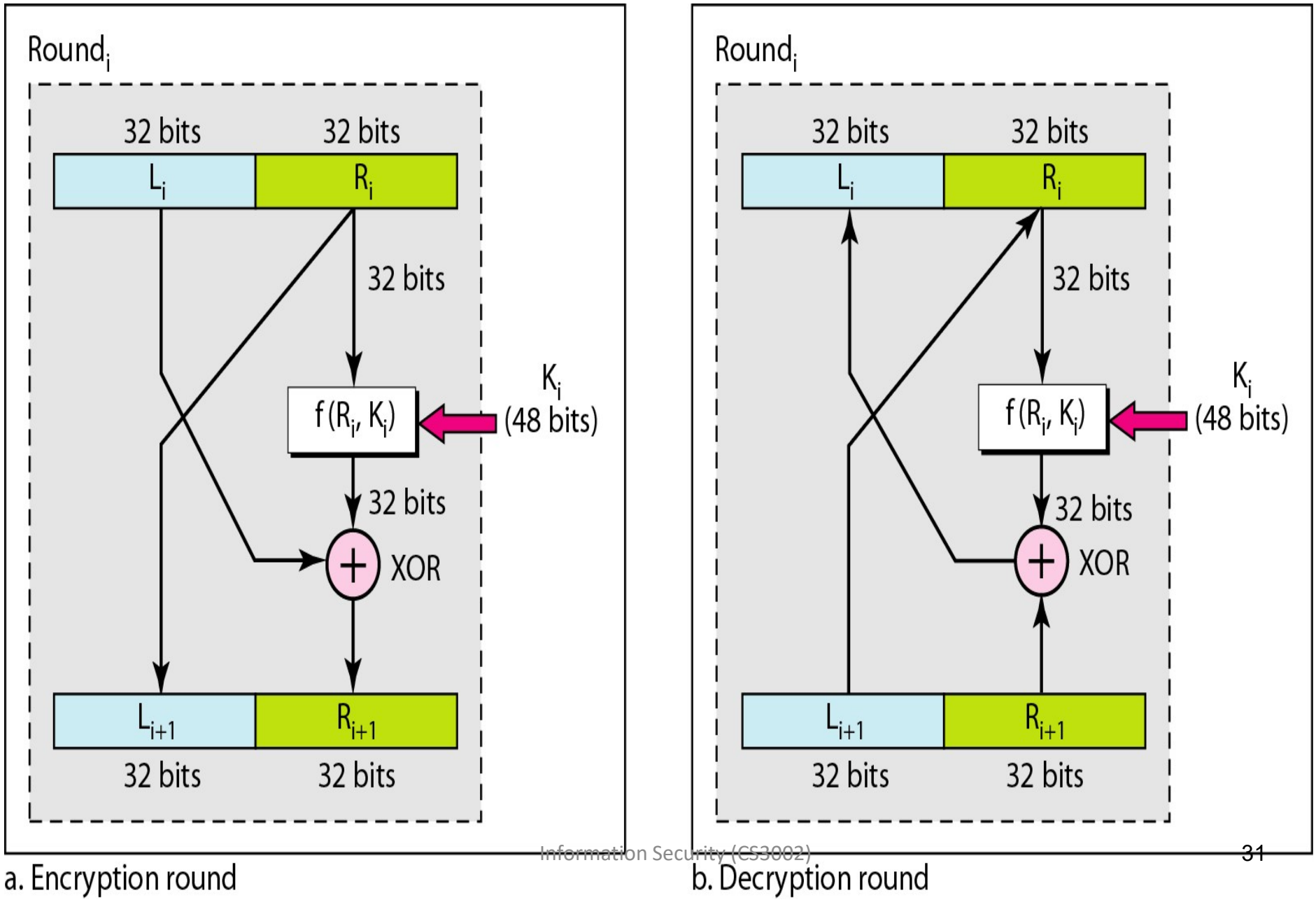
  $L_i = R_{i-1}$

  $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
- R is expanded to 48 bits using permutation E
- Resulting 48 bits are XORed with K_i
- 48 bit result passes thru substitution function F
  - (8 S-boxes) producing 32-bit output
- Output is permuted using permutation function P



One round in DES

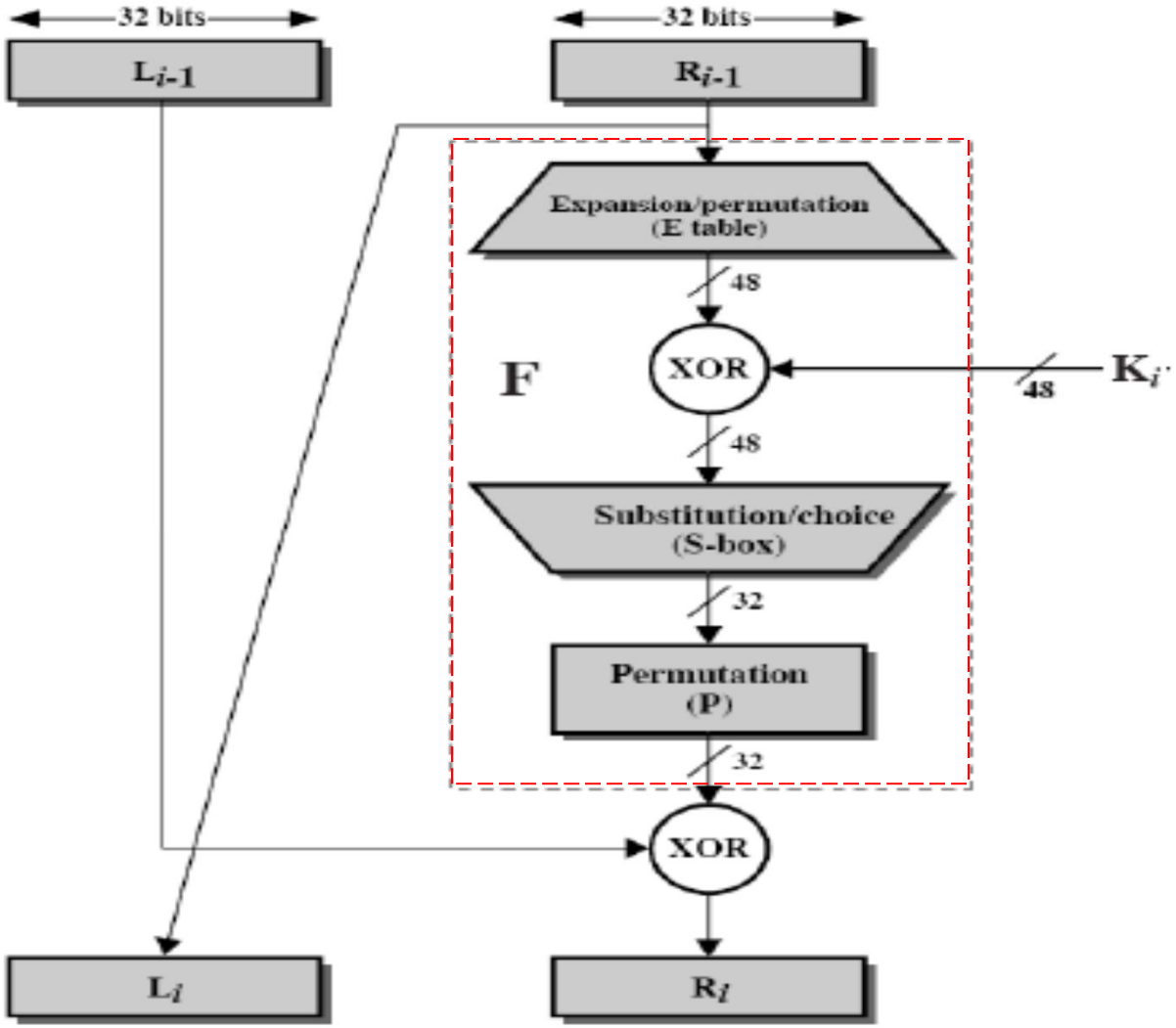
DataComm&nw/4e,
Forouzanbook



a. Encryption round

b. Decryption round

Details of Single Round



Permutation Tables E, P

(c) Expansion Permutation (E)

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

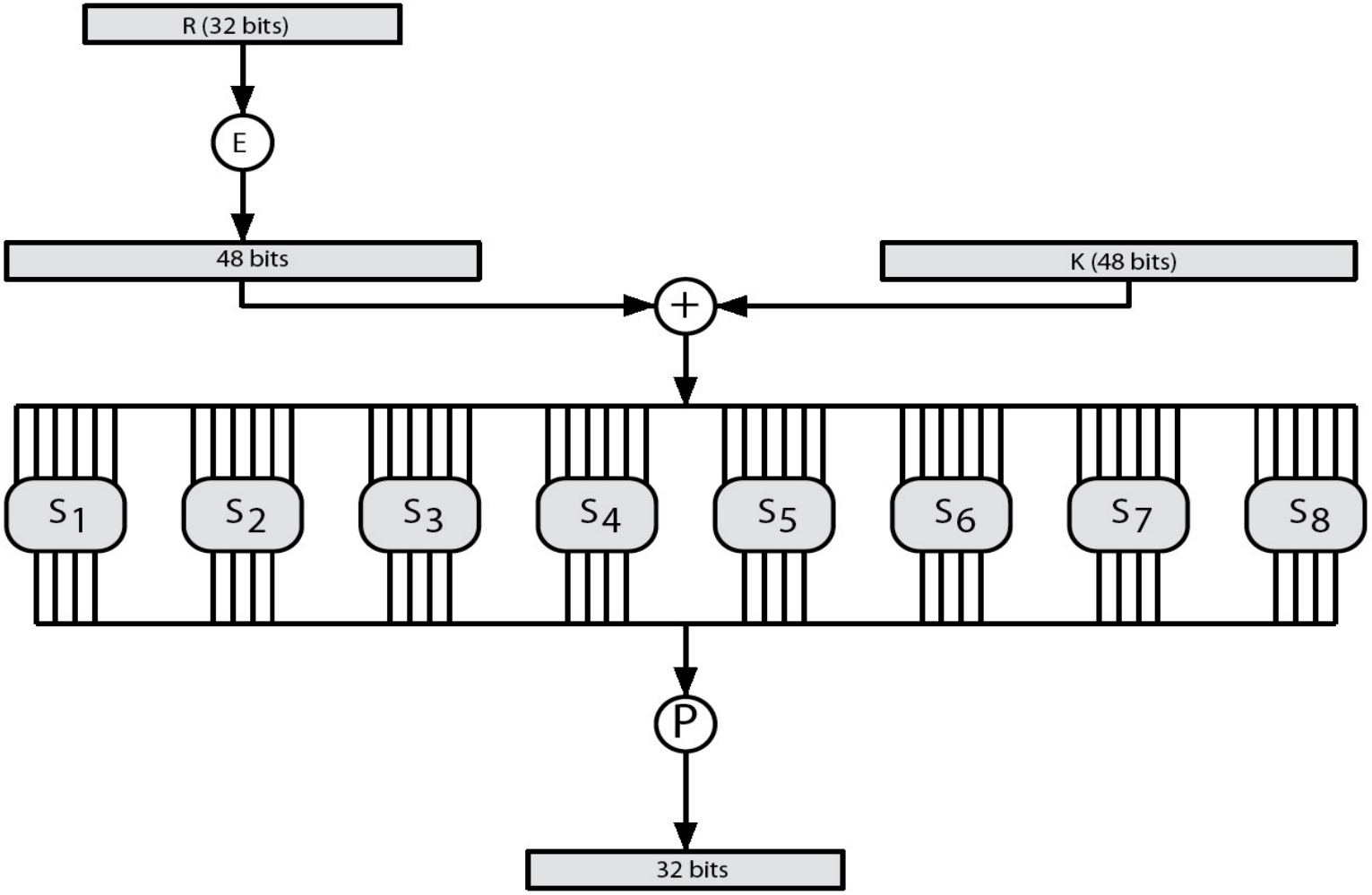
(d) Permutation Function (P)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Roles of S-boxes

- 8 s-boxes, each has 6 bits input, 4 bits out.
- Outer 2 bits (1,6) used to select row.
- Inner 4 bits (2-5) used to select column.
- Decimal value of cell converted to 4 bits out.
  - Note that decimal values are [0-15].
- 8 4-bit groups produce 32 bit output.
- Row selection depends on both data & key.
  - Feature known as autoclaving (auto-keying).

Role of S-Boxes (cont.)



Role of S-Boxes

Appendix S of reference book.

S_1	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

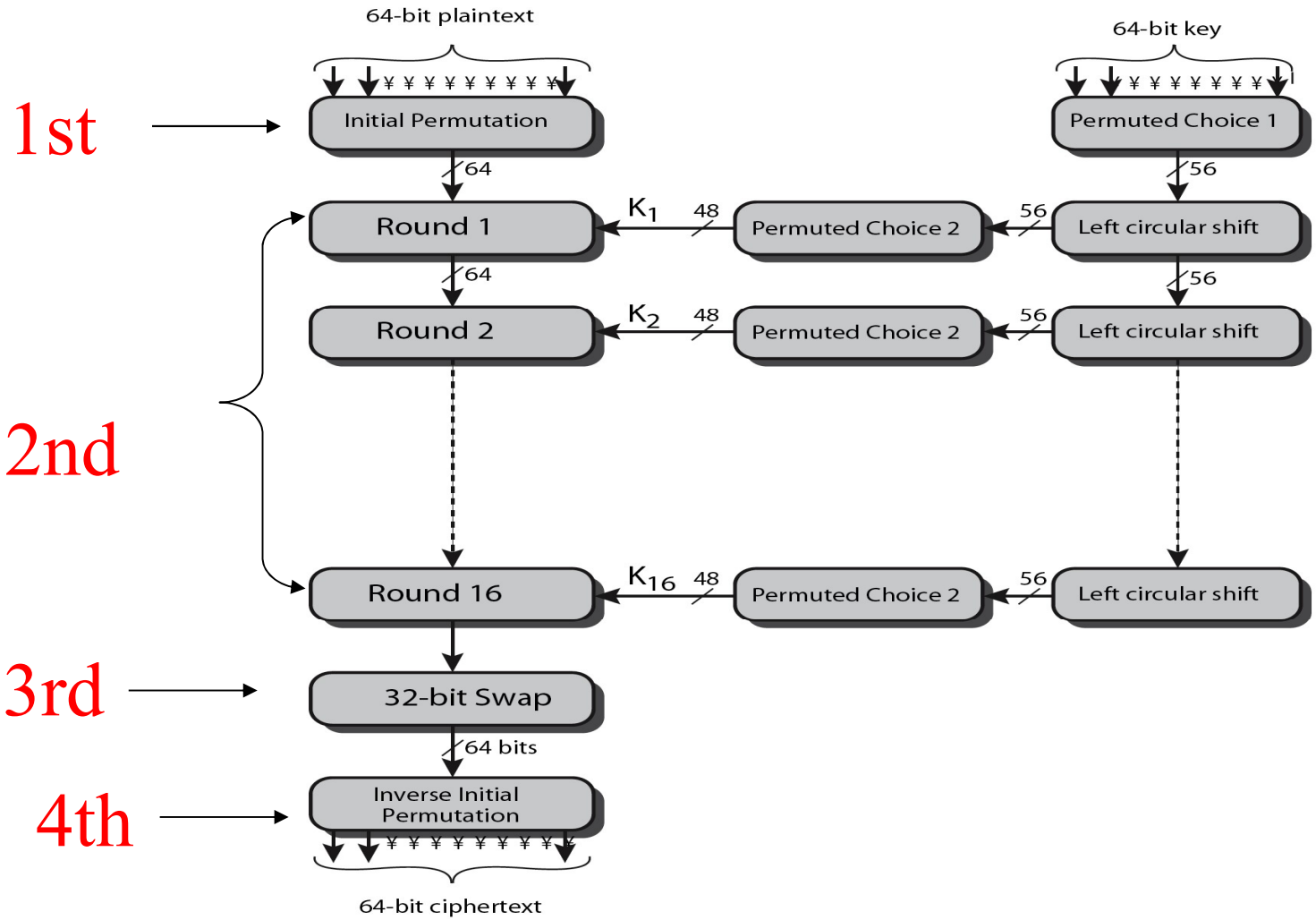
S_3	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

Example

- Using S1
- Input: 011001
- Row is 01: (1)
- Column 1100: (12)
- Value of row 1, column 12 is 9
- Output is 1001

Key Generation in DES

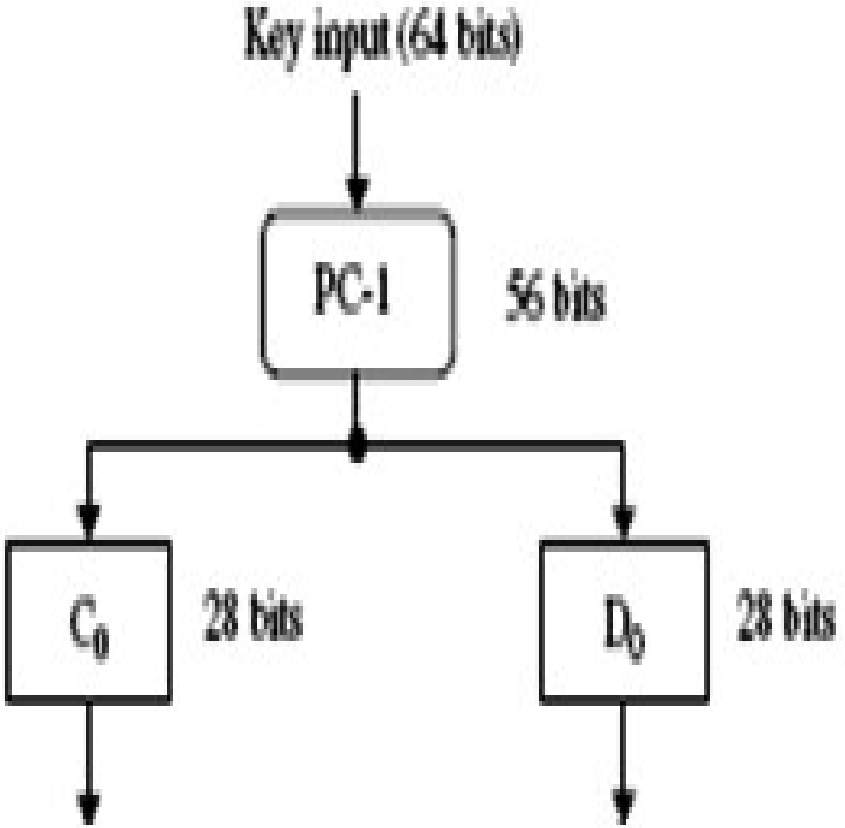
DES Encryption Overview



Key Generation (1/6)

- 64-bit key used as input (8 × 8 table)
- 8th bit in each row is ignored $\rightarrow$ 56 bits
- Key is permuted using table PC-1
- Resulting 56 bits separated into two 28-bit parts C_0, D_0

Key Generation (2/6)



Internet Security: Cryptographic Principles, Algorithms and Protocols, P-61

Key Generation (3/6)

(a) Input Key

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

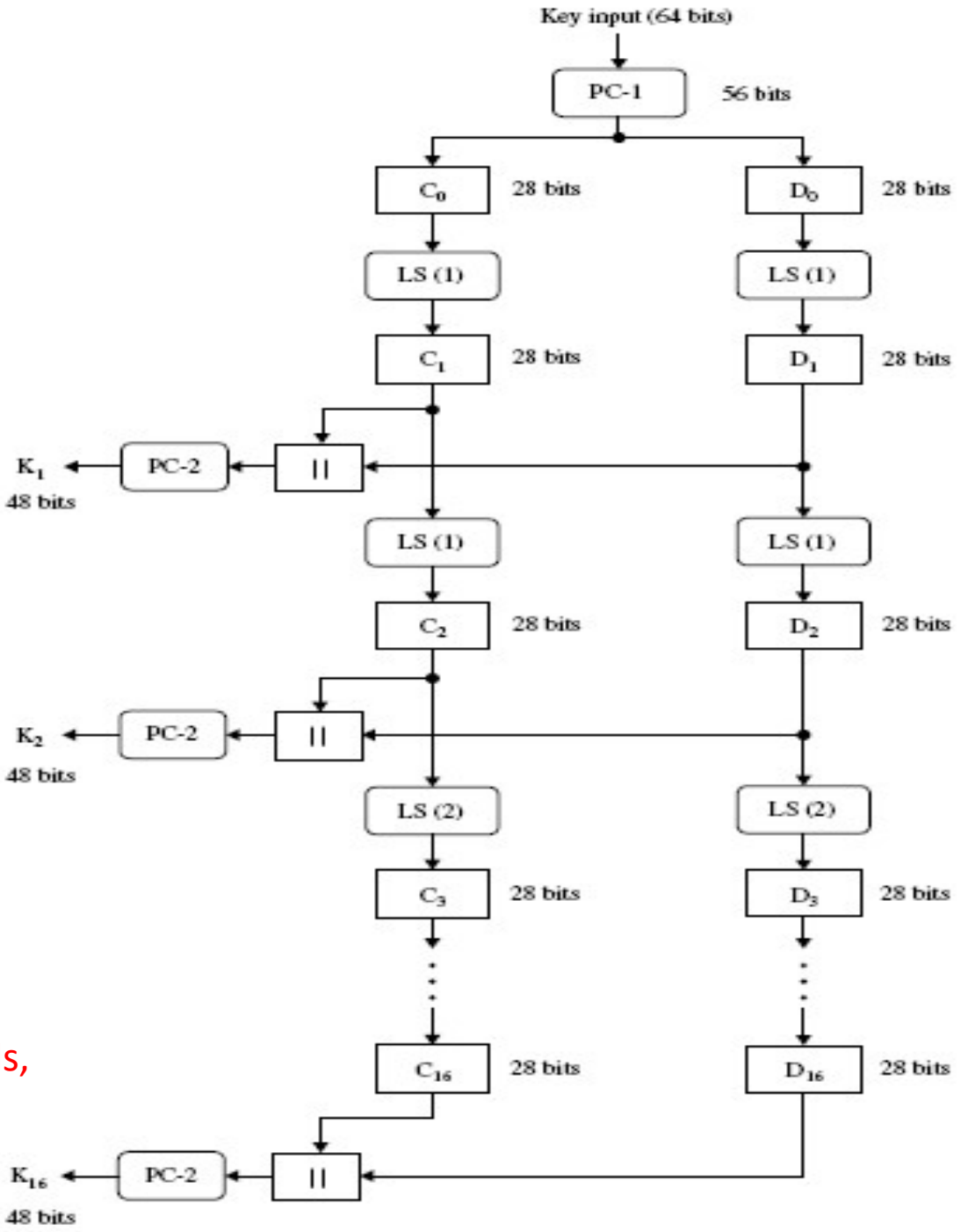
(b) Permuted Choice One (PC-1)

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

Key Generation (4/6)

- Each round
  - Circular left shift c_{i-1}, d_{i-1} of 1 or 2 bits (table)
  - Shifted values go to next round
  - Also used as input to table PC-2
  - PC-2 produce 48-bit output k_i used in $f(r_{i-1}, k_i)$

Key Generation (5/6)



Internet Security: Cryptographic Principles, Algorithms and Protocols, P-61

Key Generation (6/6)

(c) Permuted Choice Two (PC-2)

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

(d) Schedule of Left Shifts

Round number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

DES Decryption

- Decrypt must unwind steps of data computation.
- With Feistel design, do encryption steps again using subkeys in reverse order (SK_{16} … SK_1).
  - IP undoes final FP step of encryption
  - 1st round with SK_{16} undoes 16th encrypt round
  - ….
  - 16th round with SK_1 undoes 1st encrypt round
  - then final FP undoes initial encryption IP
  - thus recovering original data value

Strength of DES – Key Size

- 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values
- Brute force search looks hard.
- Recent advances have shown is possible
  - in 1997 on Internet in a few months .
  - in 1998 on dedicated h/w (EFF) in a few days .
  - in 1999 above combined in 22hrs!
- Still must be able to recognize plaintext
- Must now consider alternatives to DES