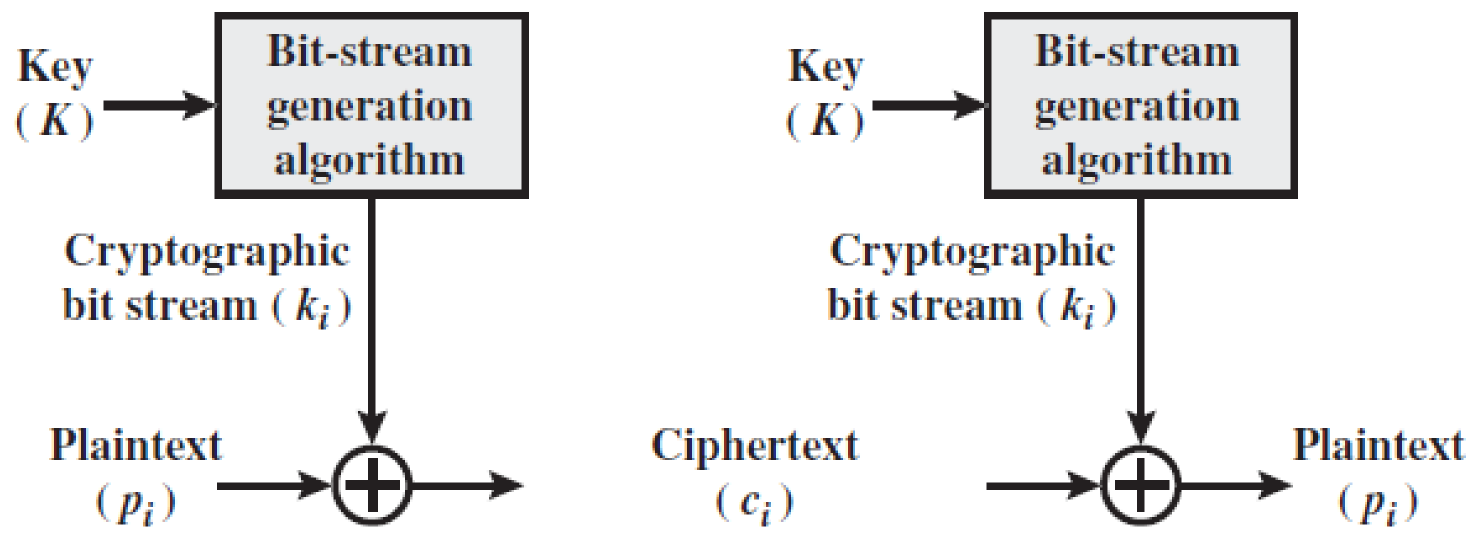
COMPUTER SECURITY (CSE 4105)

Block Cipher and DES

Stream Cipher

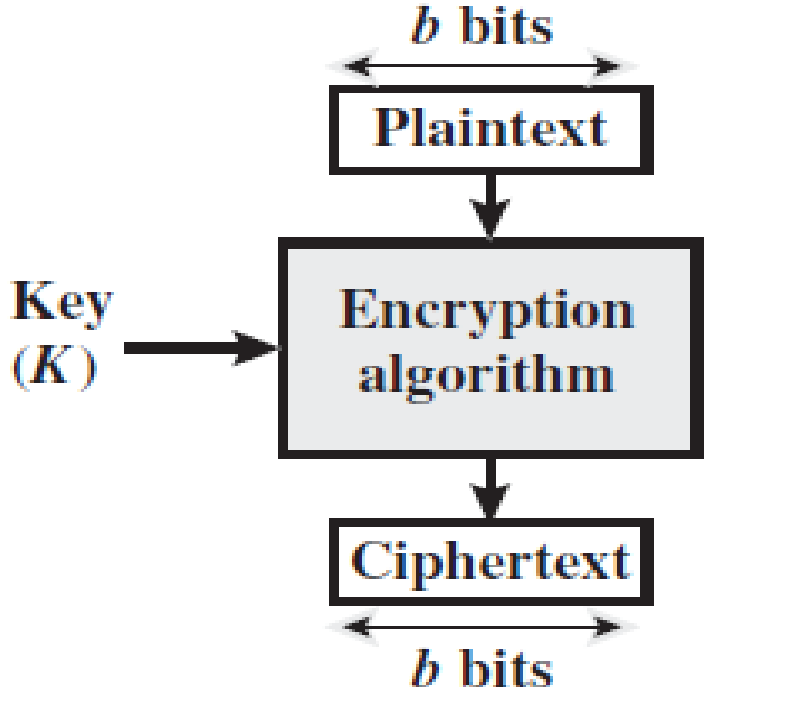
- A stream cipher is one that encrypts a digital data stream one bit or one byte at a time.
- Examples of classical stream ciphers are the autokeyed Vigenère cipher and the Vernam cipher.



(a) Stream cipher using algorithmic bit-stream generator

Block Cipher

- A block cipher is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.
- Typically, a block size of 64 or 128 bits is used.



(b) Block cipher

Block Cipher

- A block cipher operates on a plaintext block of n bits to produce a ciphertext block of n bits.
- There are 2^n possible different plaintext blocks and, for the encryption to be reversible (i.e., for decryption to be possible), each must produce a unique ciphertext block.
- Such a transformation is called reversible, or nonsingular.

Reversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	00
11	01

Irreversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	01
11	01

Block Cipher

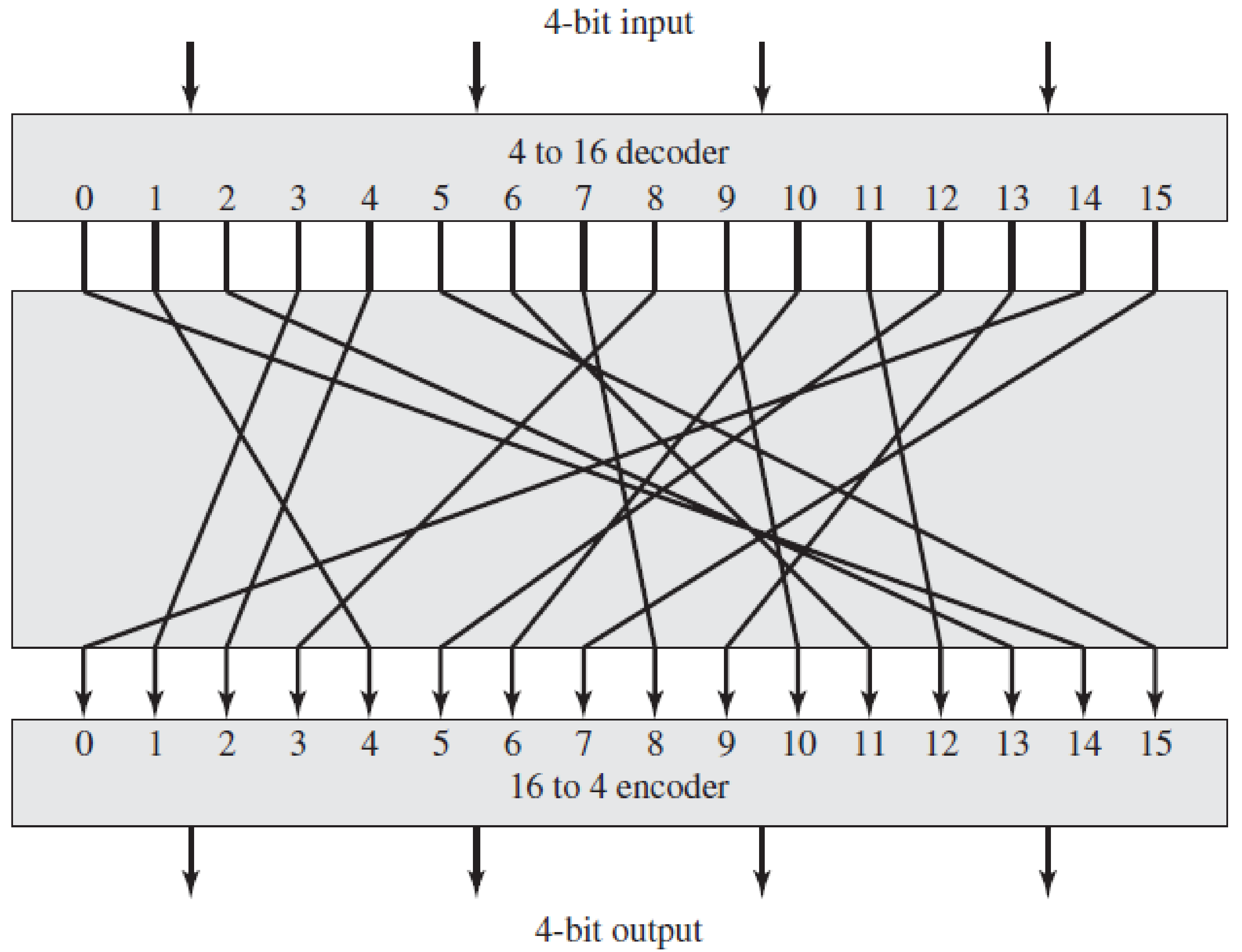


Figure 3.2 General n-bit-n-bit Block Substitution (shown with $n = 4$)

Block Cipher

- But there is a practical problem with the ideal block cipher.
- If a small block size, such as $n=4$, is used, then the system is equivalent to a classical substitution cipher.
- Such systems, as we have seen, are vulnerable to a statistical analysis of the plaintext.
- This weakness is not inherent in the use of a substitution cipher but rather results from the use of a small block size.

Block Cipher

- An arbitrary reversible substitution cipher (the ideal block cipher) for a large block size is not practical, however, from an implementation and performance point of view.
- For such a transformation, the mapping itself constitutes the key.
- In general, for an *n-bit* ideal block cipher, the length of the key defined in this fashion is $n \times 2^n$ bits.
- For a *64-bit* block, which is a desirable length to thwart statistical attacks, the required key length is $64 \times 2^{64} = 2^{70} \approx 10^{21}$ bits.
- In considering these difficulties, Feistel points out that what is needed is an approximation to the ideal block cipher system for large *n*

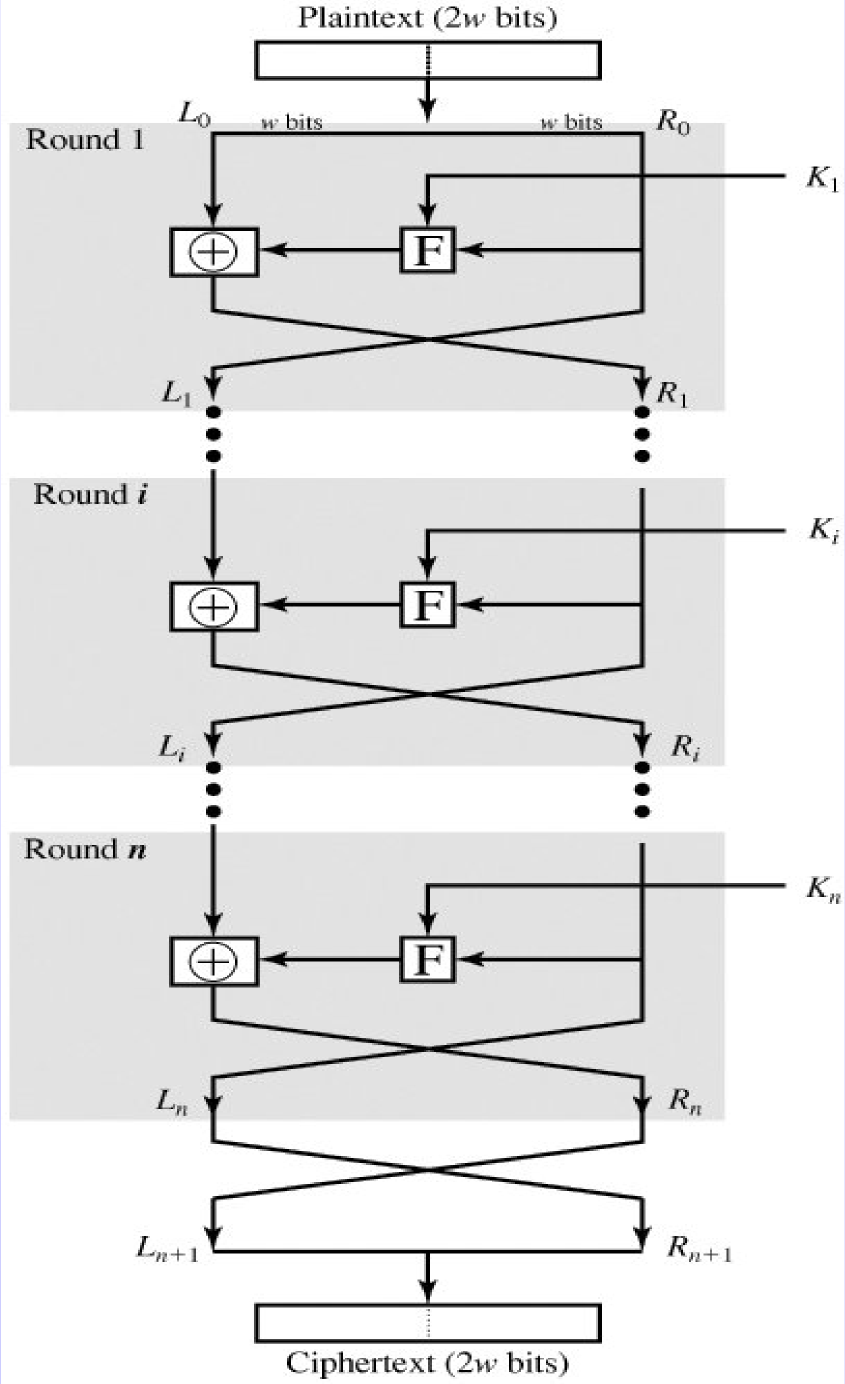
Feistel Cipher

- Feistel proposed the use of a cipher that alternates substitutions and permutations
  - Substitution: Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements.
  - Permutation: A sequence of plaintext elements is replaced by a permutation of that sequence.

Feistel Cipher

- The inputs to the encryption algorithm are a plaintext block of length *2w* bits and a key *K*.
- The plaintext block is divided into two halves, L_0 and R_0.
- The two halves of the data pass through *n* rounds of processing and then combine to produce the ciphertext bock.
- Each round *i* has as inputs L_{i-1} and R_{i-1} derived from previous round as well as a subkey K_i.

Continue…


Plaintext (2w bits)
L0
w bits
w bits
R0
Round 1
K1
F
L1
R1
Round i
Ki
F
Li
Ri
Round n
Kn
F
Ln
Rn
Ln+1
Rn+1
Ciphertext (2w bits)

Continue…

- A substitution is performed on the left half of the data.
- This is done by applying a round function *F* to the right half of the data and then taking the X-OR of the output of that function and the left half data.
- After substitution, a permutation is performed that consists of the interchange of the two halves of the data.

Continue…

- Feistel network depends on the following choices:
  - Block size: Larger block means greater security but reduced encryption/decryption speed for a given algorithm.
  - Key size: Larger key size means greater security but may decrease encryption/ decryption speed.
  - Number of rounds: Single round offers inadequate security but that multiple rounds offer increasing security.

Continue…

- Feistel network depends on the following choices:
  - Subkey generation algorithm: Greater complexity in this algorithm should lead to greater difficulty of cyptanalysis.
  - Round function: Greater complexity means greater resistance to cryptanalysis.

Data Encryption Standard (DES)

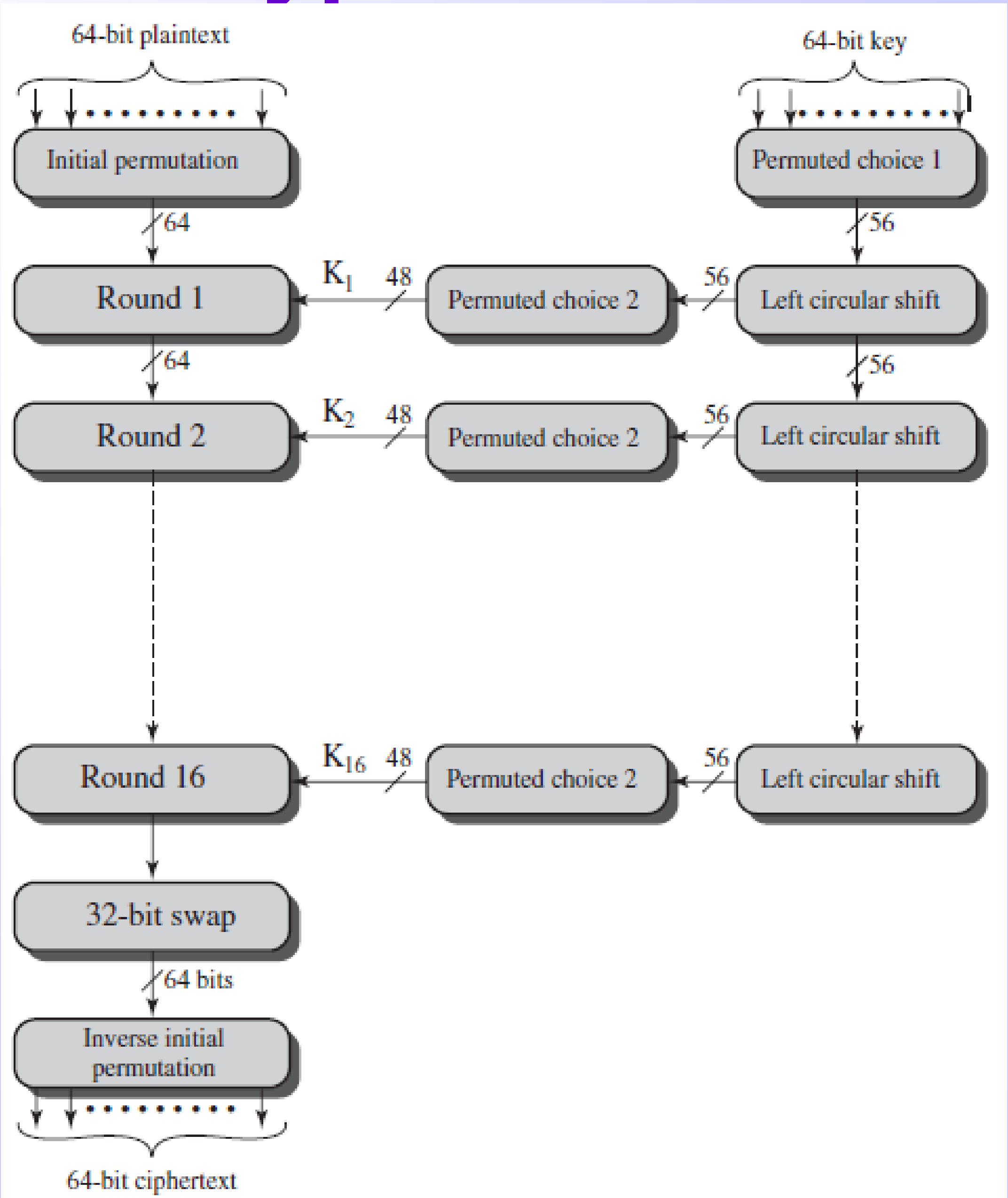



- The most widely used encryption scheme is based on the DES adopted in 1977 by NBS (now NIST).
- Late 1960, IBM set up a research project in computer cryptography led by Horst Feistel.
- The project developed an algorithm in 1971 LUCIFER, which was sold to Lyoyd Bank of London for cash dispensing system.
- LUCIFER is a Feistel block cipher that operates on 64 bits and key size of 128 bits.
- For marketable commercial encryption product, IBM and other technical consultants reduce the key size to 56 bits to fit in a single chip (headed by Walter Tuchman and Carl Meyer).

Data Encryption Standard (DES)

- In 1973, the NBS issued a request for proposals for a national cipher standard.
- IBM submitted the results of its Tuchman-Meyer project.
- This was by far the best algorithm proposed and was adopted in 1977 as the Data Encryption Standard.
- **Criticisms on DES**:
  - The key length of IBM's original project was 128 bits but that was the proposed system was only 56 which will be too short for brute-force attack.
  - The design criteria for the internal structure of DES, S-boxes were classified.

Data Encryption Standard (DES)



Continue...

- Initial Permutation (IP)

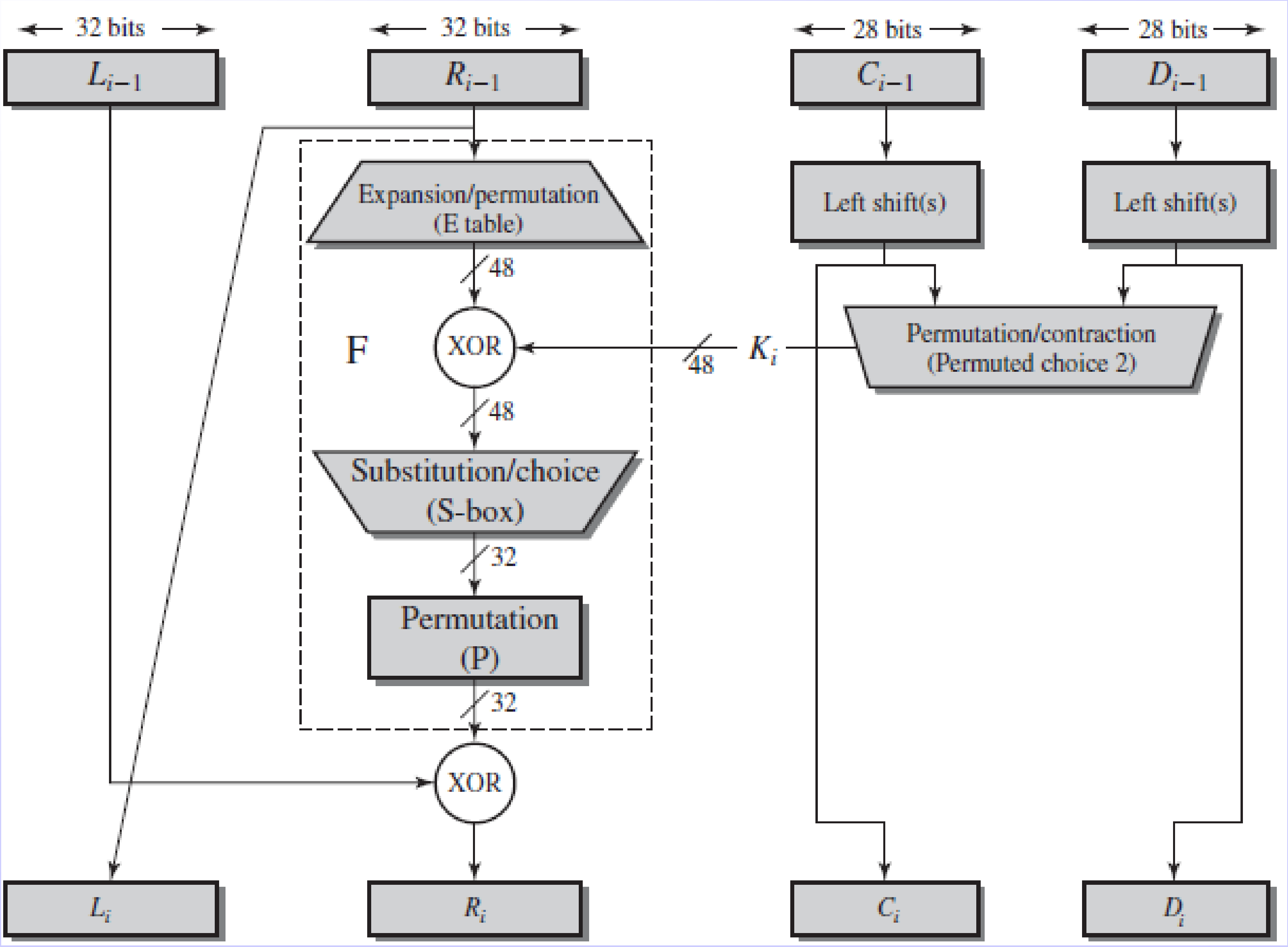
M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}
M_{17}	M_{18}	M_{19}	M_{20}	M_{21}	M_{22}	M_{23}	M_{24}
M_{25}	M_{26}	M_{27}	M_{28}	M_{29}	M_{30}	M_{31}	M_{32}
M_{33}	M_{34}	M_{35}	M_{36}	M_{37}	M_{38}	M_{39}	M_{40}
M_{41}	M_{42}	M_{43}	M_{44}	M_{45}	M_{46}	M_{47}	M_{48}
M_{49}	M_{50}	M_{51}	M_{52}	M_{53}	M_{54}	M_{55}	M_{56}
M_{57}	M_{58}	M_{59}	M_{60}	M_{61}	M_{62}	M_{63}	M_{64}

Original Plaintext (64 bits)

IP

M_{58}	M_{50}	M_{42}	M_{34}	M_{26}	M_{18}	M_{10}	M_2
M_{60}	M_{52}	M_{44}	M_{36}	M_{28}	M_{20}	M_{12}	M_4
M_{62}	M_{54}	M_{46}	M_{38}	M_{30}	M_{22}	M_{14}	M_6
M_{64}	M_{56}	M_{48}	M_{40}	M_{32}	M_{24}	M_{16}	M_8
M_{57}	M_{49}	M_{41}	M_{33}	M_{25}	M_{17}	M_9	M_1
M_{59}	M_{51}	M_{43}	M_{35}	M_{27}	M_{19}	M_{11}	M_3
M_{61}	M_{53}	M_{45}	M_{37}	M_{29}	M_{21}	M_{13}	M_5
M_{63}	M_{55}	M_{47}	M_{39}	M_{31}	M_{23}	M_{15}	M_7

Details of a Single Round (DES)



Continue…

- The left and right halves of each 64-bit intermediate value are treated separate 32-bit quantities, labeled L and R.
- The overall processing at each round can be summarized in the following formula:
  - $L_i = R_{i-1}$
  - $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
- The key K_i is 48 bits.
- The R input is 32 bits.
- R input is first expanded to 48 bits by using expansion permutation table.
- The resulting 48 bits are X-Ored with K_i.

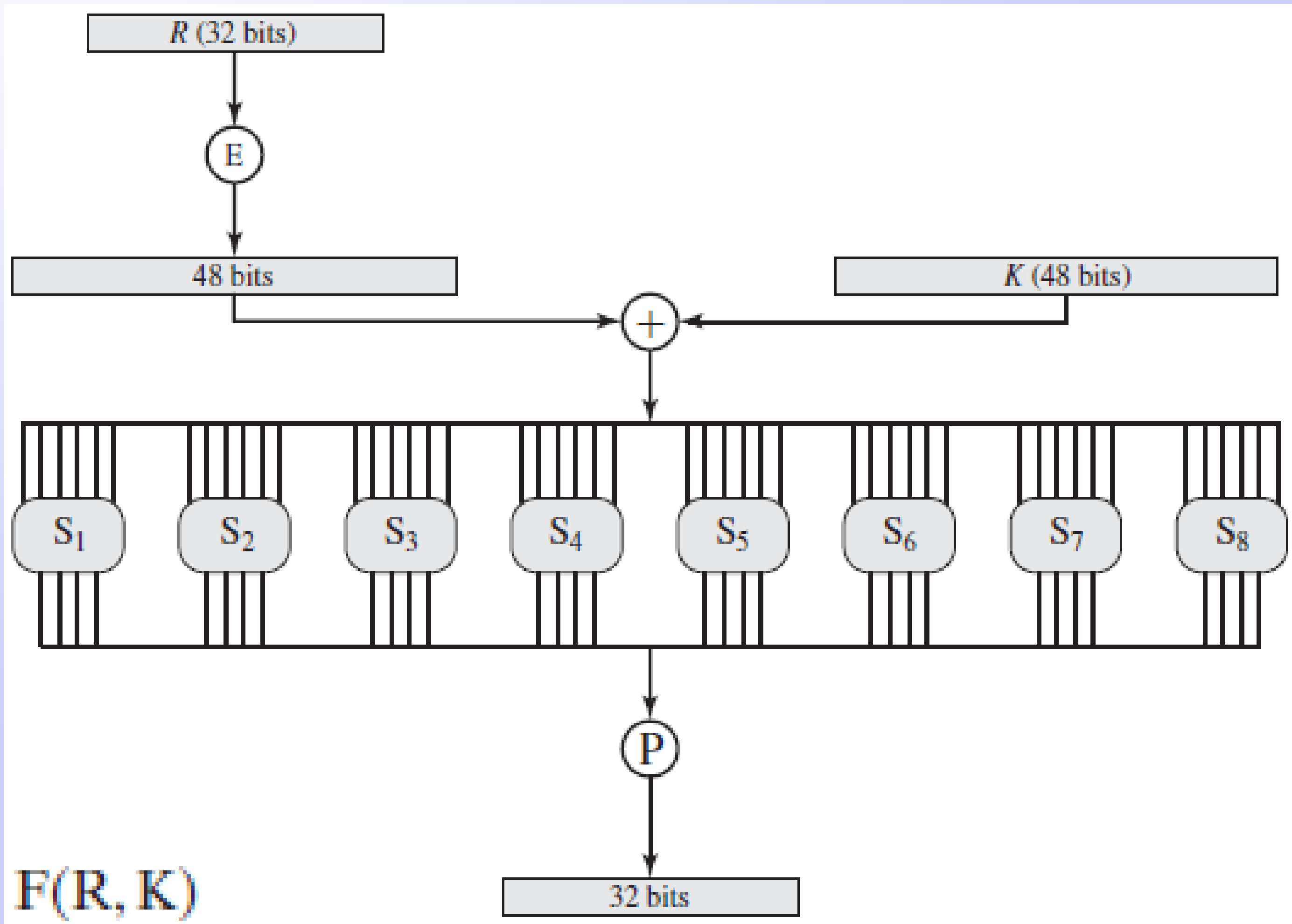
Cont. (Expansion Permutation)

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

- The resulting 48 bits are X-Ored with K_i.

Continue...

- The 48 bit result passes through a substitution function that produce a 32 bit output



Continue...

- The substitution consists of a set of eight S-boxes, each of which accepts 6 bits as input and produces 4 bit as output.
- The first and last bits of the input to box S_i form a 2-bit binary number to select one of the four rows and middle four bits selects one of the 16 columns.
- The decimal value in the cell is then converted to its 4 bit representation to produce the output.
- The 32 bit output from the eight S-boxes is then permuted, so that on the next round the output from each S-box immediately affects as many others as possible.

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

S_4	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Cont.(Permutation Function)

- The 32-bit output is permuted by permutation function table.

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

- The permuted data is then X-ored with L_{i-1} that produce R_i.
- R_{i-1} come directly to produce L_i.

Key Generation

- The bits of the key are numbered from 1 through 64; every eighth bit is ignored.
- The key is first subjected to a permutation governed by the PC-1.

(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

- The resulting 56-bit key is then treated as two 28-bit quantities, labeled C_0 and D_0.
- At each round, C_{i-1} and D_{i-1} are separately subjected to a circular left shift or ration of 1 or 2 bits as directed.
- These shifted values serve as input to the next round.
- They also serve as input to PC-2, which produce a 48-bit output that serves as input to the function $F(R_{i-1}, K_i)$

Continue…

(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

Thanks for your Attention