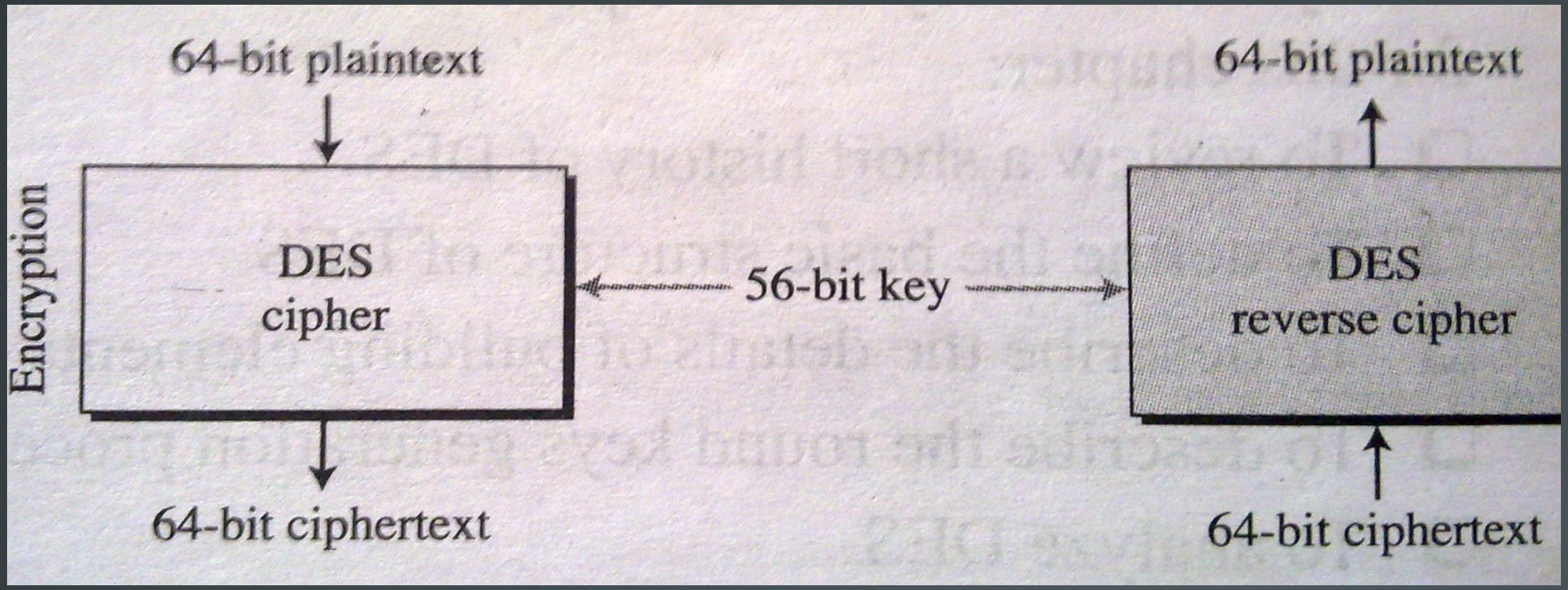
DATA ENCRYPTION STANDARD

- DES, in short, is a symmetric key block cipher published by National Institute of Standards and Technology (NIST)
- Modified version of IBM's ***Lucipher***
- 64 bits long PT blocks => 64 bits long CT
- 56 bits long key

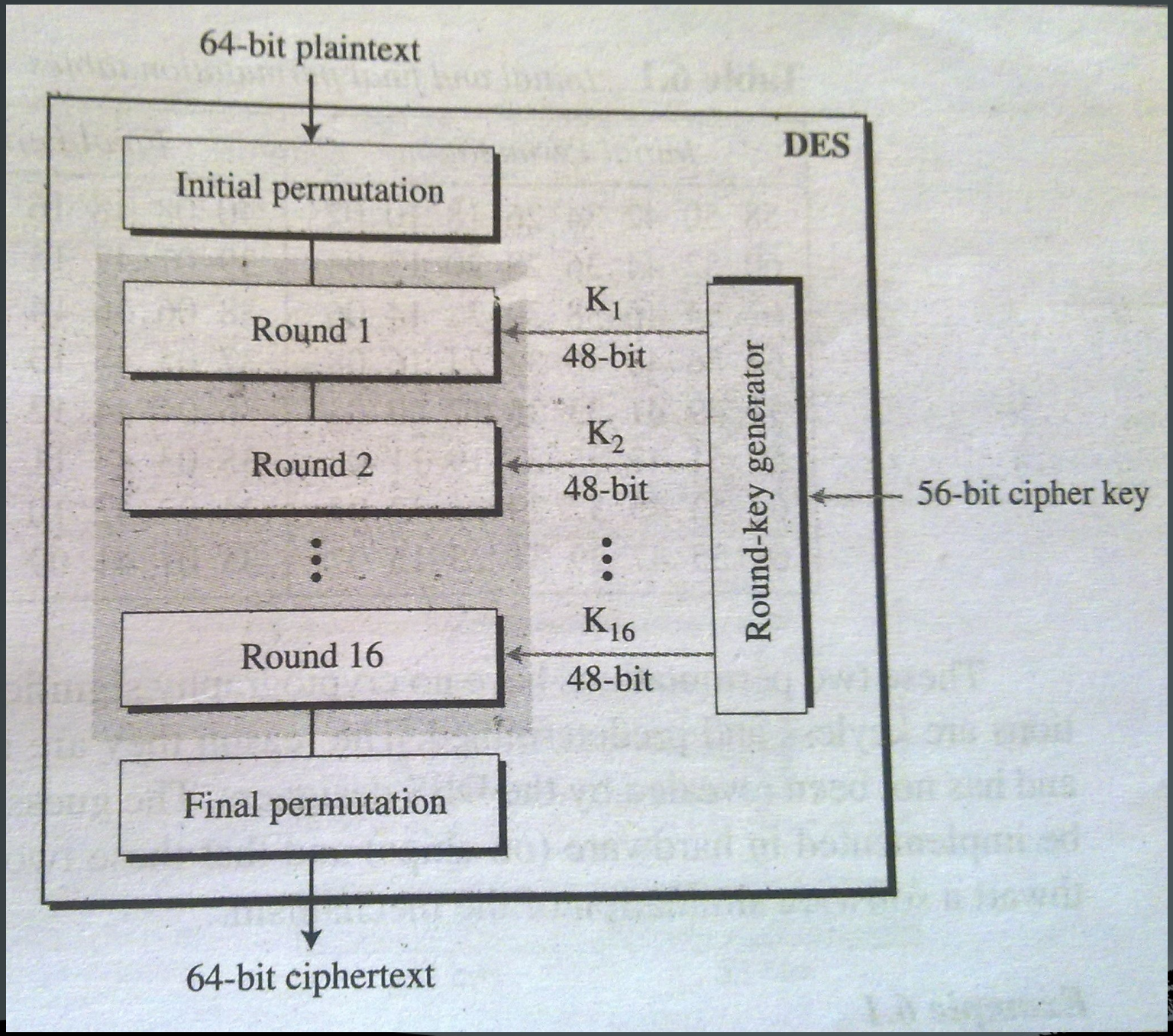
DATA ENCRYPTION STANDARD



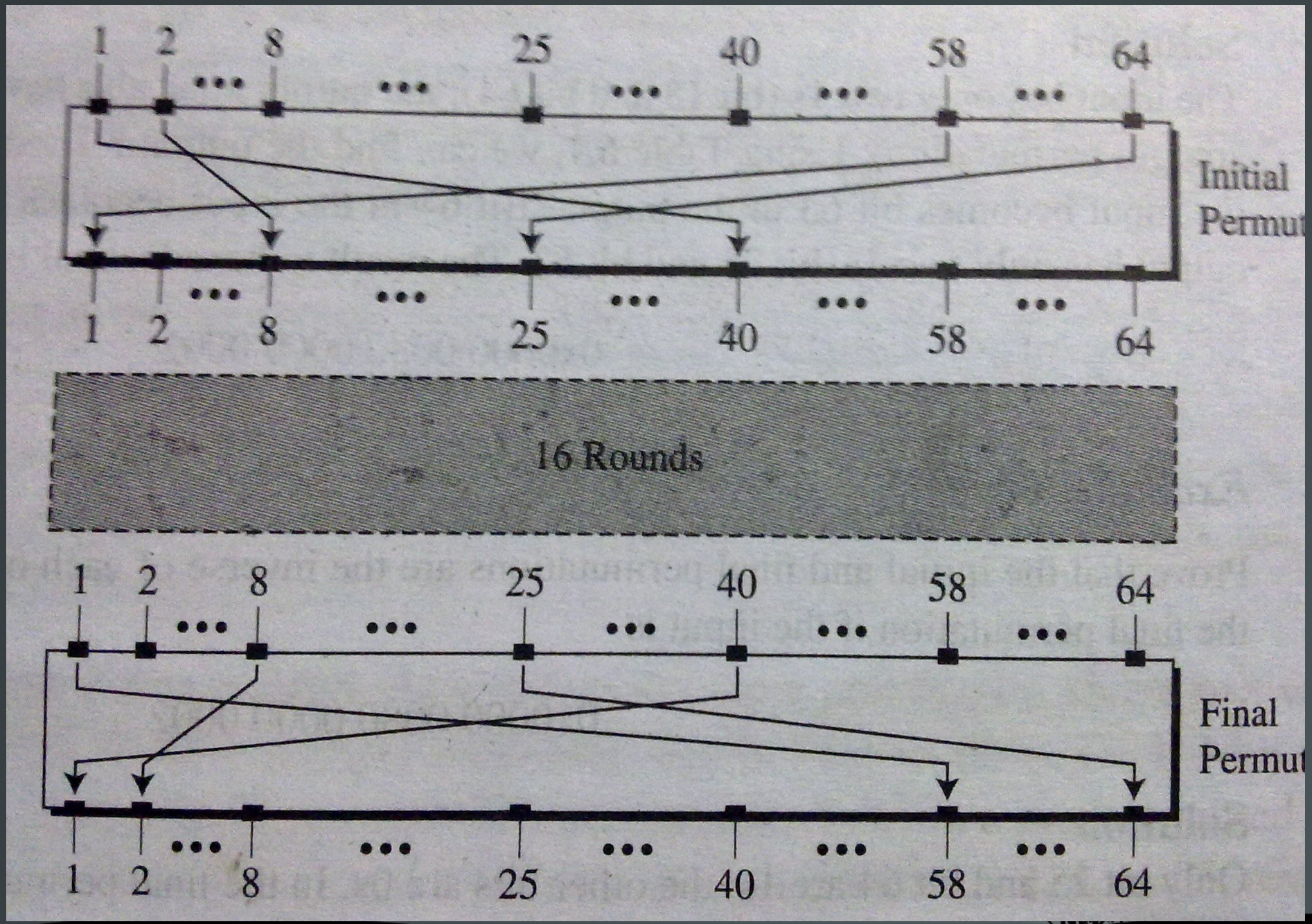
DES STRUCTURE

- Encryption process
  - 2 P-Boxes (for initial permutation & final permutation)
  - 16 Fiestel rounds
  - Each round has a unique 48 bit 'round key' which is generated from the cipher key

DES STRUCTURE

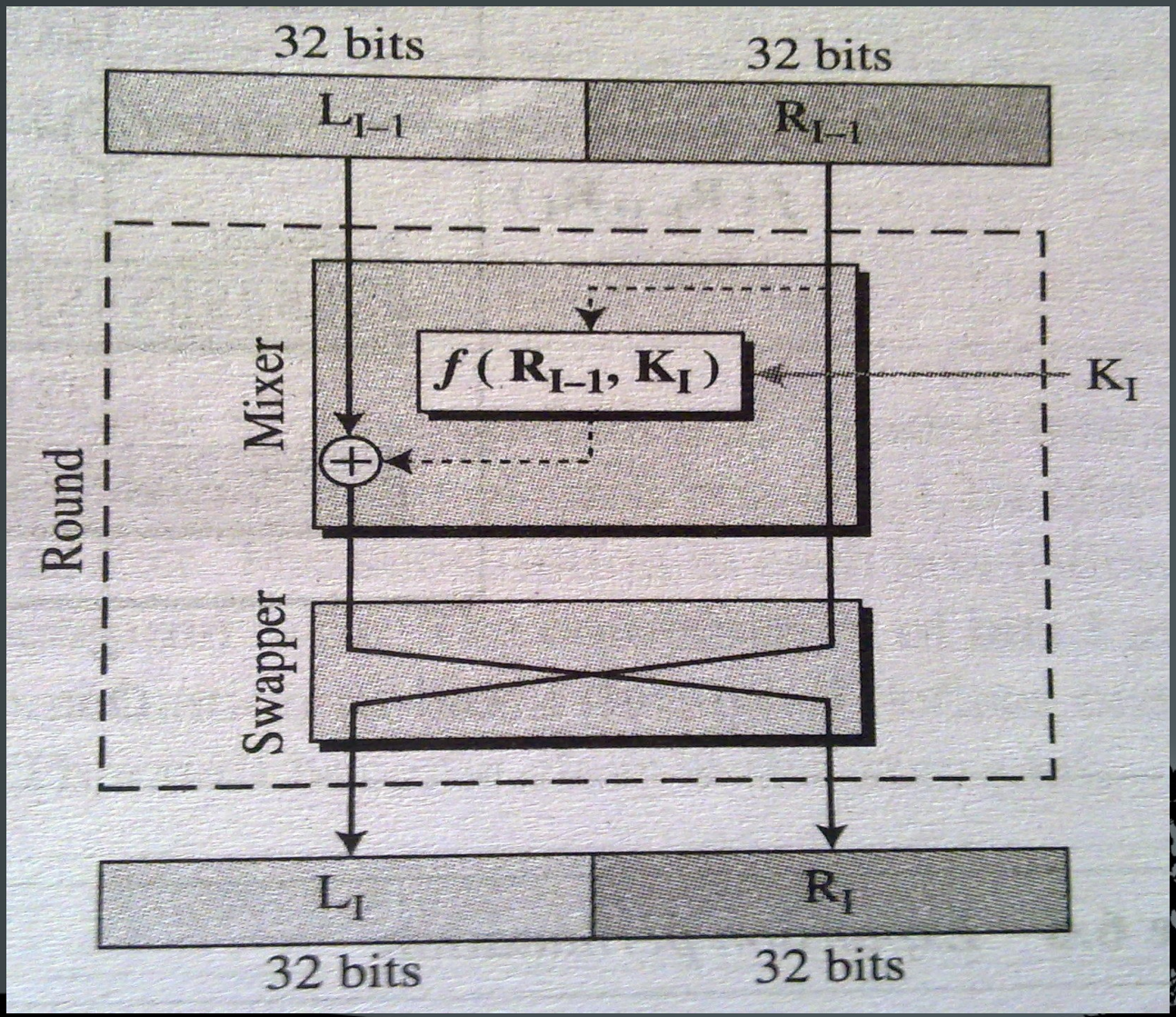


Initial & Final Permutation



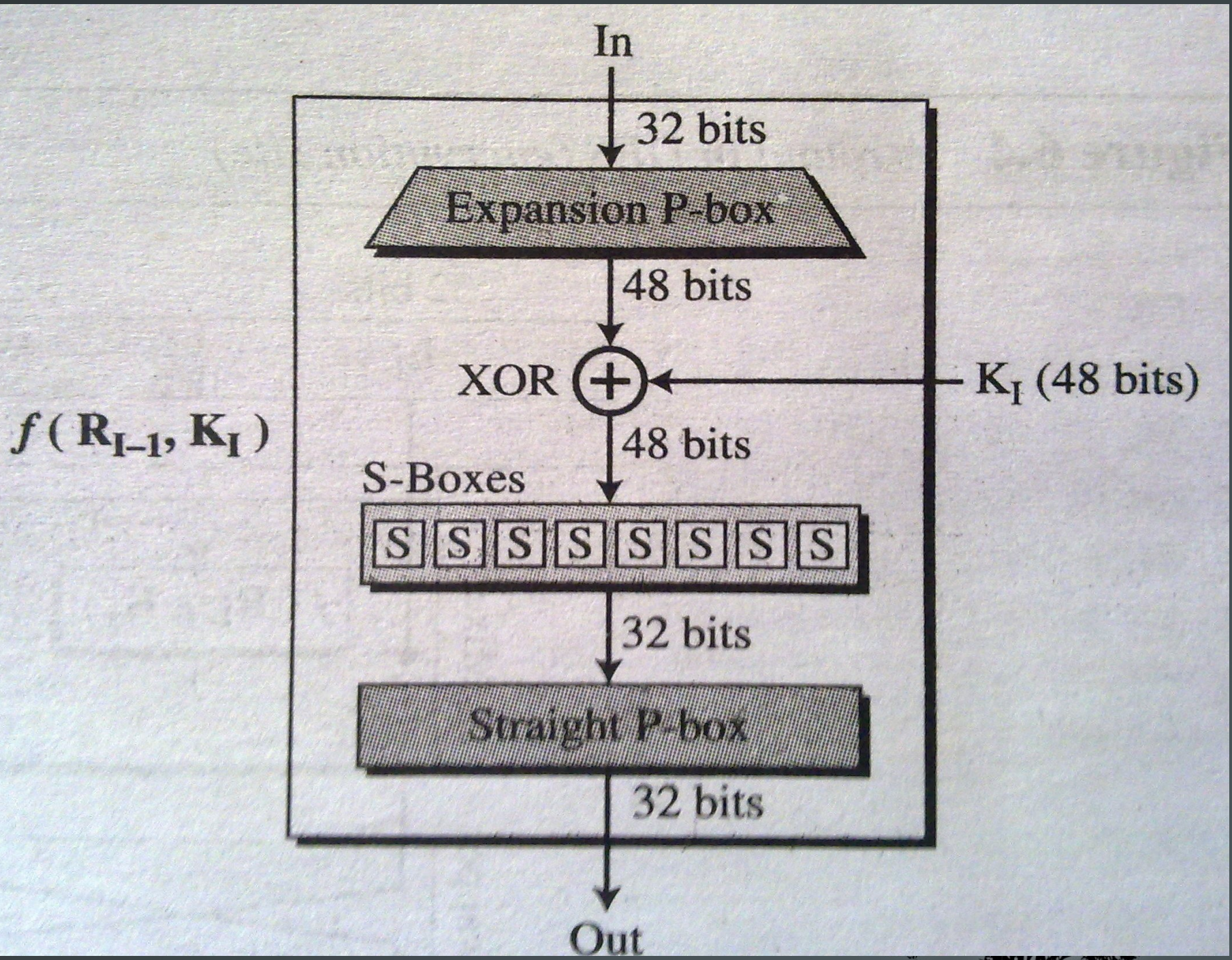
Rounds

- Each round is a Feistel cipher round



DES function $f(R_{i-1}, K_i)$

- Applies a 48 bit key to the rightmost 32 bit PT to produce a 32 bit output
- Made of an expansion P box, a whitener, a group of S boxes, and a stright P box


In
32 bits
Expansion P-box
48 bits
XOR
K_I (48 bits)
$f(R_{I-1}, K_I)$
48 bits
S-Boxes
S S S S S S S S
32 bits
Straight P-box
32 bits
Out

Expansion P-Box

- Need:
  - R_{i-1} is 32 bits long & K_i is 48 bits long
  - R_{i-1} is to be expanded to 48 bits
- Following figure illustrates . . .

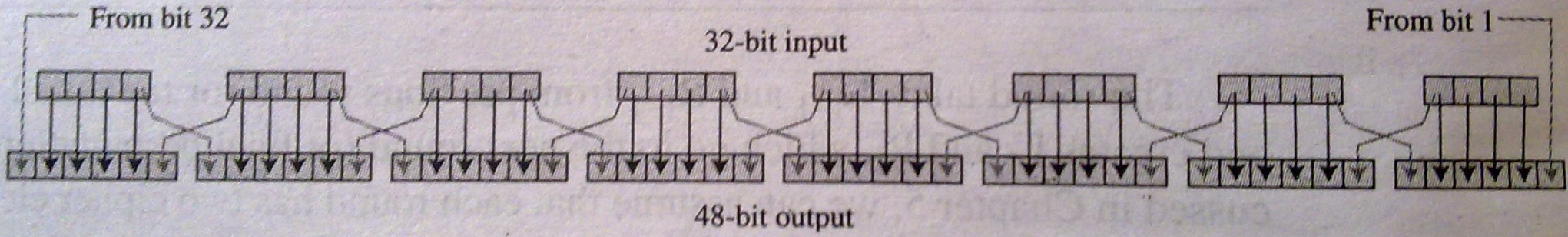


Table 6.2 *Expansion P-box table*

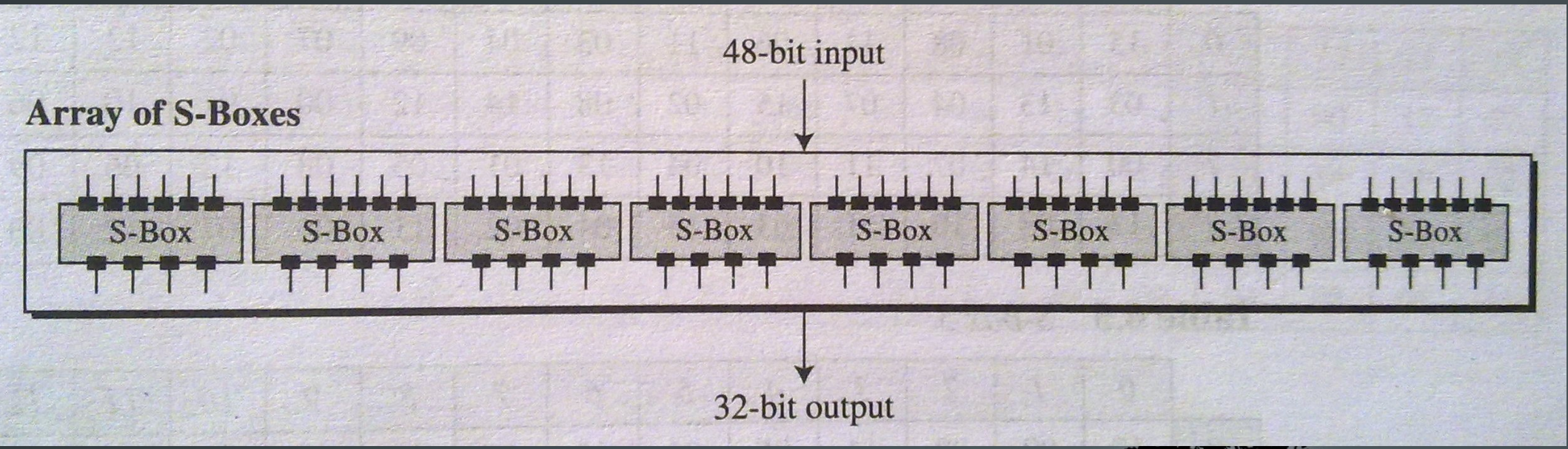
32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

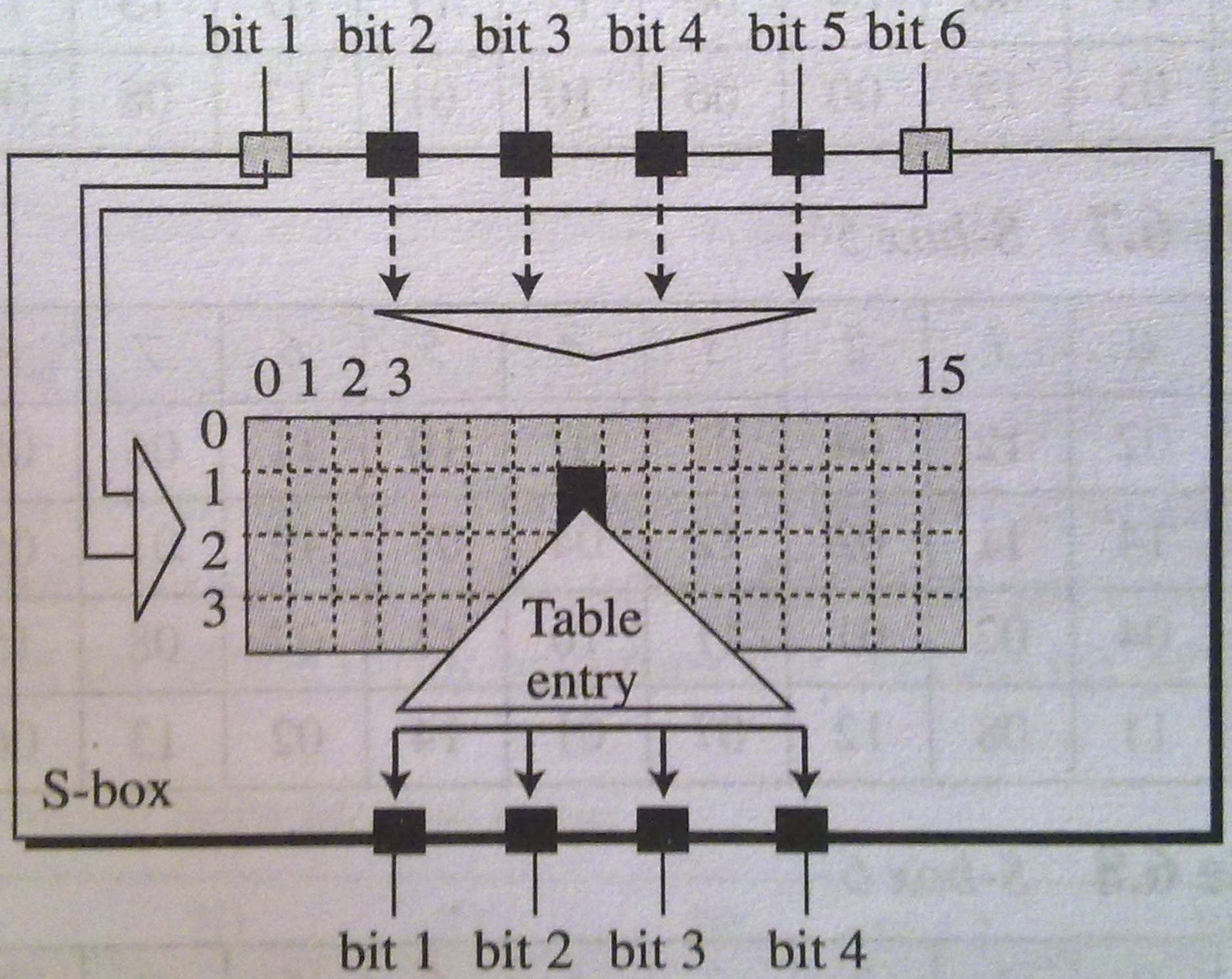
Whitener (XOR)

- Expanded right section is XORed with the Round Key
- Round key is used only in this operation

S-Boxes

- Do the real mixing (confusion & diffusion)
- 8 different (non-invertible) S-Boxes used
- Each has 6 bit input and 4 bit output




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

The 8 S-Boxes

Table 6.3 *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

Table 6.4 *S-box 2*

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

The 8 S-Boxes

Table 6.5 *S-box 3*

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

Table 6.6 *S-box 4*

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

Straight permutation

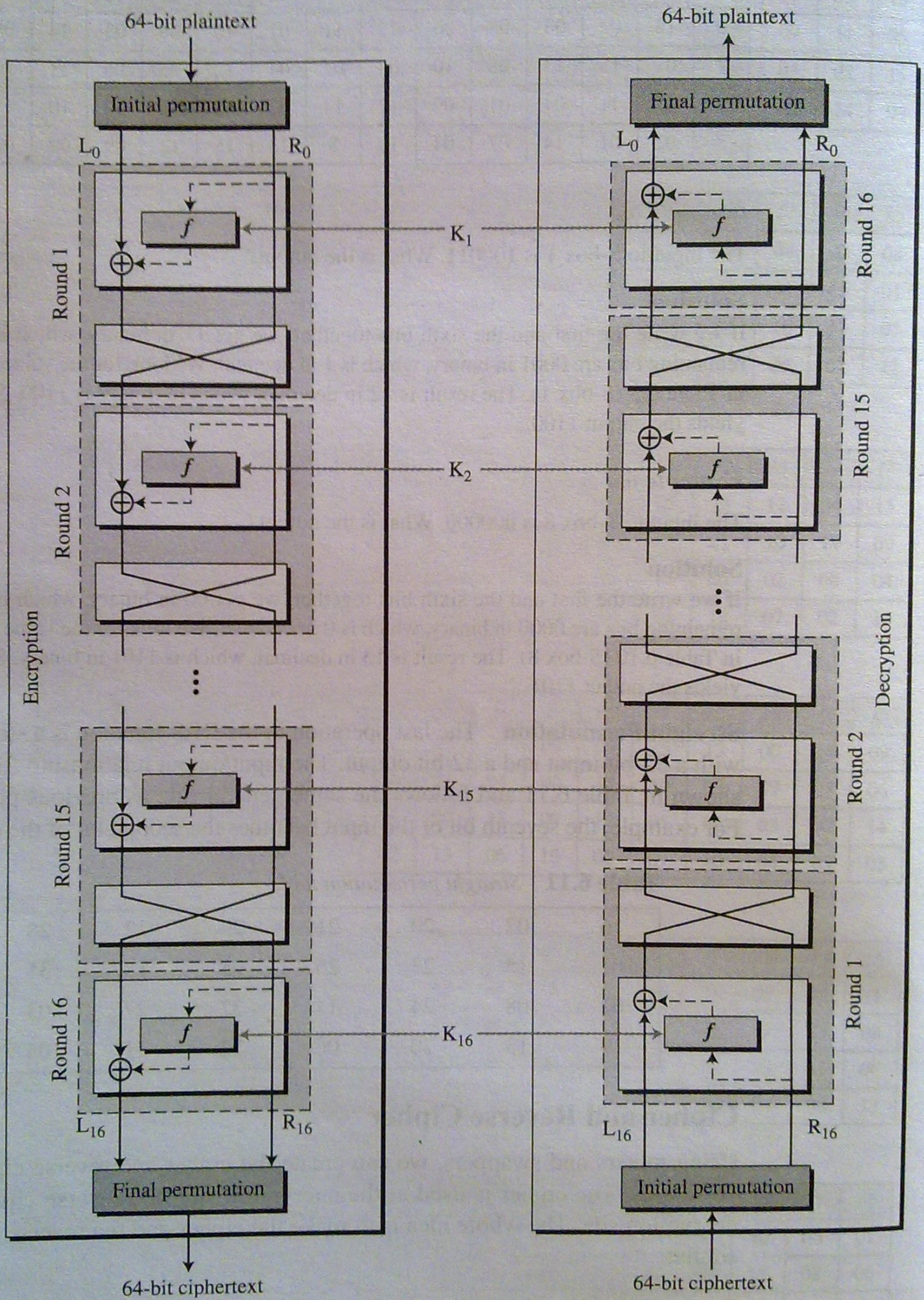
- 32 bit input & 32 bit output

Table 6.11 *Straight permutation table*

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

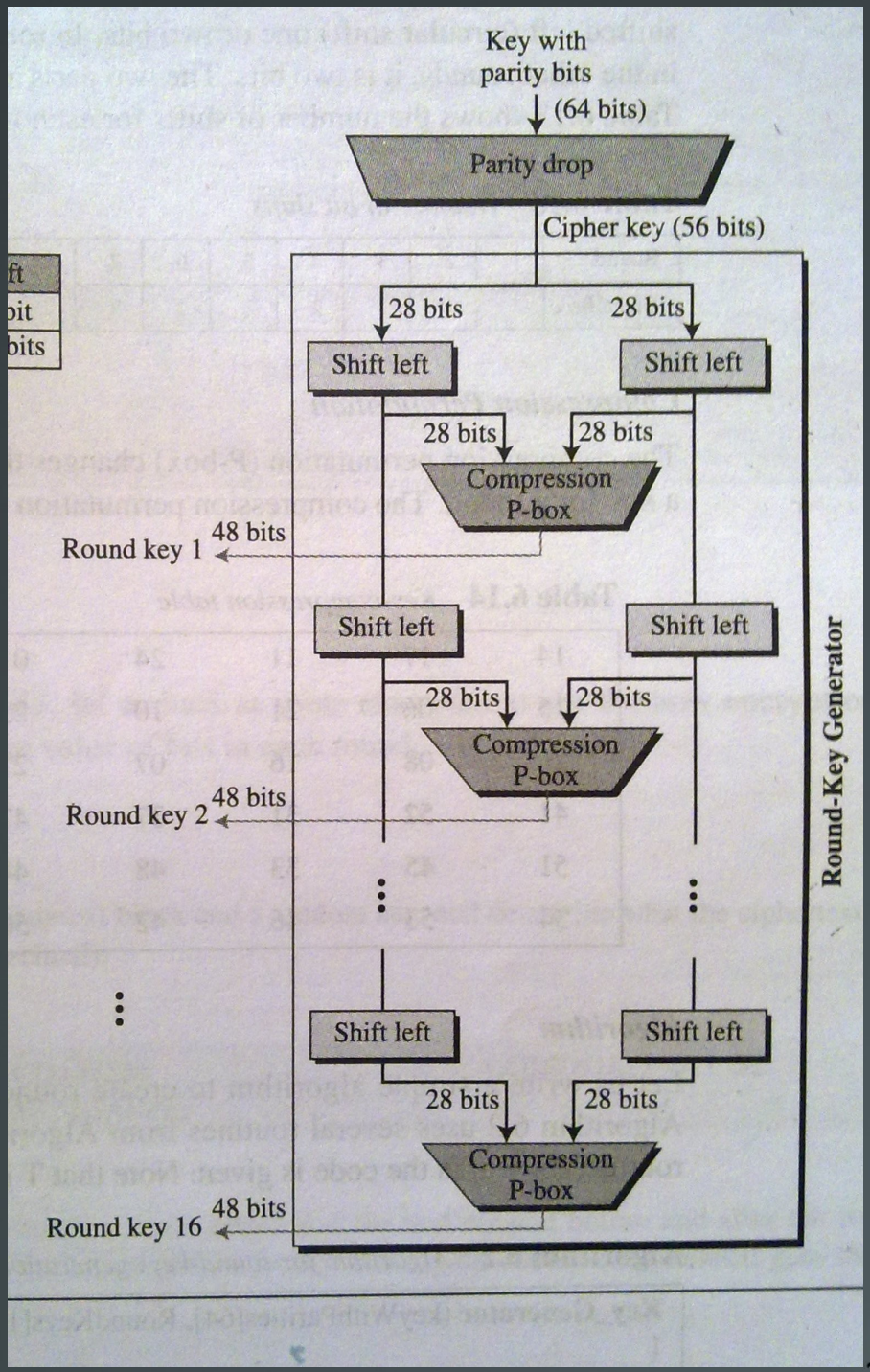
The Enc. & Dec. ciphers

- Each having 16 rounds
- Rounds are not aligned
- Round keys (k1, . . . , k16) must be applied in reverse order


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

Round key generation

- Creates sixteen 48 bit keys based on the 56 bit cipher key
- Following figure illustrates the process


Key with
parity bits
(64 bits)
Parity drop
Cipher key (56 bits)
28 bits
28 bits
Shift left
Shift left
28 bits
28 bits
Compression
P-box
Round key 1
48 bits
Shift left
Shift left
28 bits
28 bits
Compression
P-box
Round key 2
48 bits
Shift left
Shift left
28 bits
28 bits
Compression
P-box
Round key 16
48 bits
Round-Key Generator

Parity drop

- Actual 56 bit key is given as a 64 bit key to the cipher
- 56 bit key is transformed to 64 bit by adding a parity bit for every 7 bit group
- These parity bits have to be dropped before the round key generation
- Together with parity drop, a permutation is also done (with the help of a compression P-Box)

Parity drop compression P-Box

- Bits 8, 16, 24, . . . , 64 are blocked

Table 6.12 *Parity-bit drop table*

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

Shift left

- In rounds 1, 2, 9 & 16 shifting is 1 bit, & in other rounds it is 2 bits
- Ref. figure

Per round compression permutation

- Changes 56 bits to 48 bits
- Results of left shifting are combined and given as input to the compression P-Box

Table 6.14 *Key-compression table*

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

Two desirable properties

- Two desirable properties of a block cipher are
- Avalanche effect: *A small change in the PT or key should create a significant change in the CT*
- DES have been proved to be strong w.r.t. this prop.

Plaintext: 0000000000000000 Key: 22234512987ABB23
Ciphertext: 4789FD476E82A5F1

Plaintext: 0000000000000001 Key: 22234512987ABB23
Ciphertext: 0A4ED5C15A63FEA3

- 1 bit difference in PT => 29 bit differences in CT

Two desirable properties

- Completeness effect: Each bit of the CT needs to depend on many bits of the PT
- The P-Boxes and S-boxes of DES ensures that it is complete

DES weaknesses

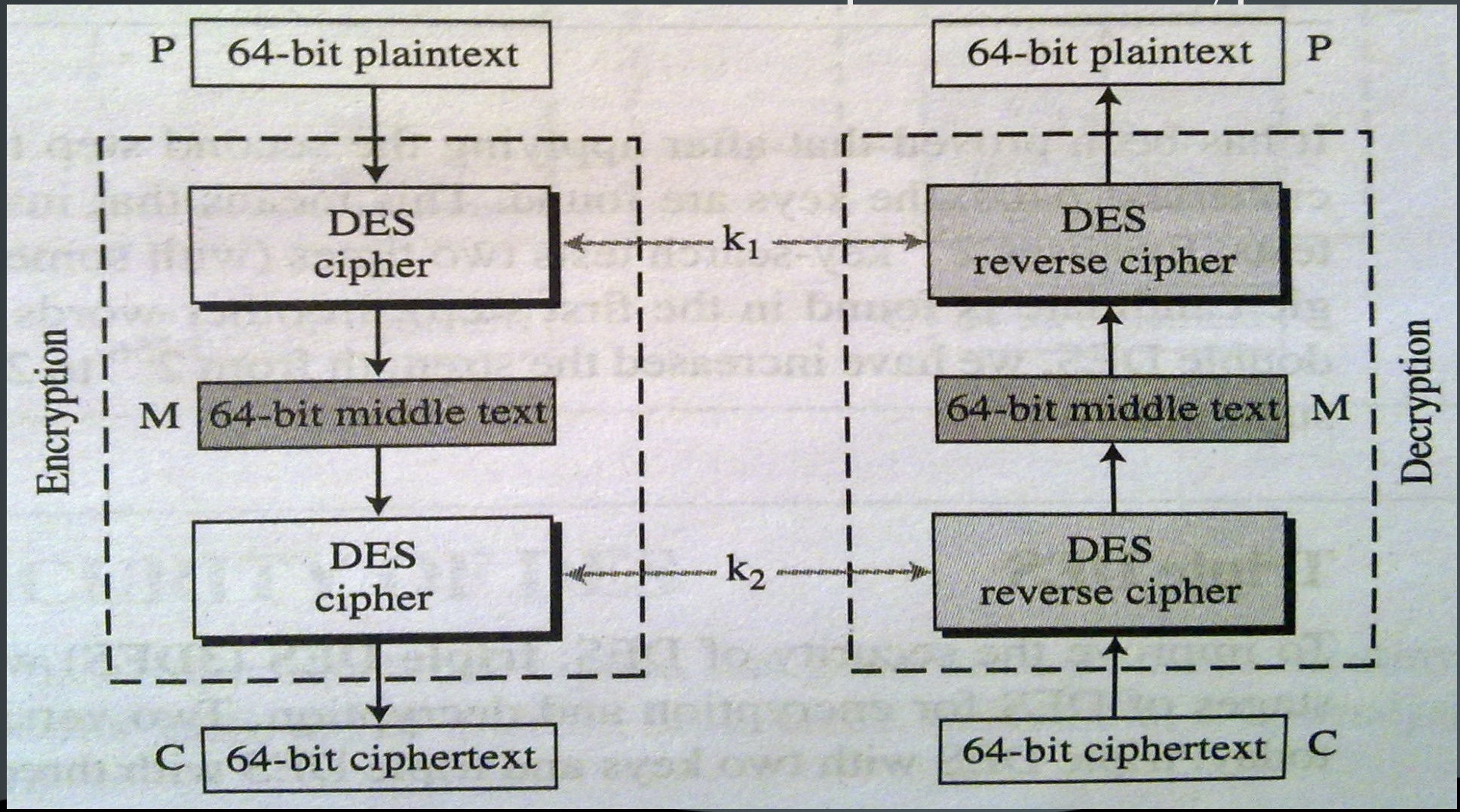
- Later

MULTIPLE DES

- With 56 bit key, a bruteforce attack is feasile on DES
- Solution: use mulitple (cascaded) instance of DES with different keys

Double DES (2DES)

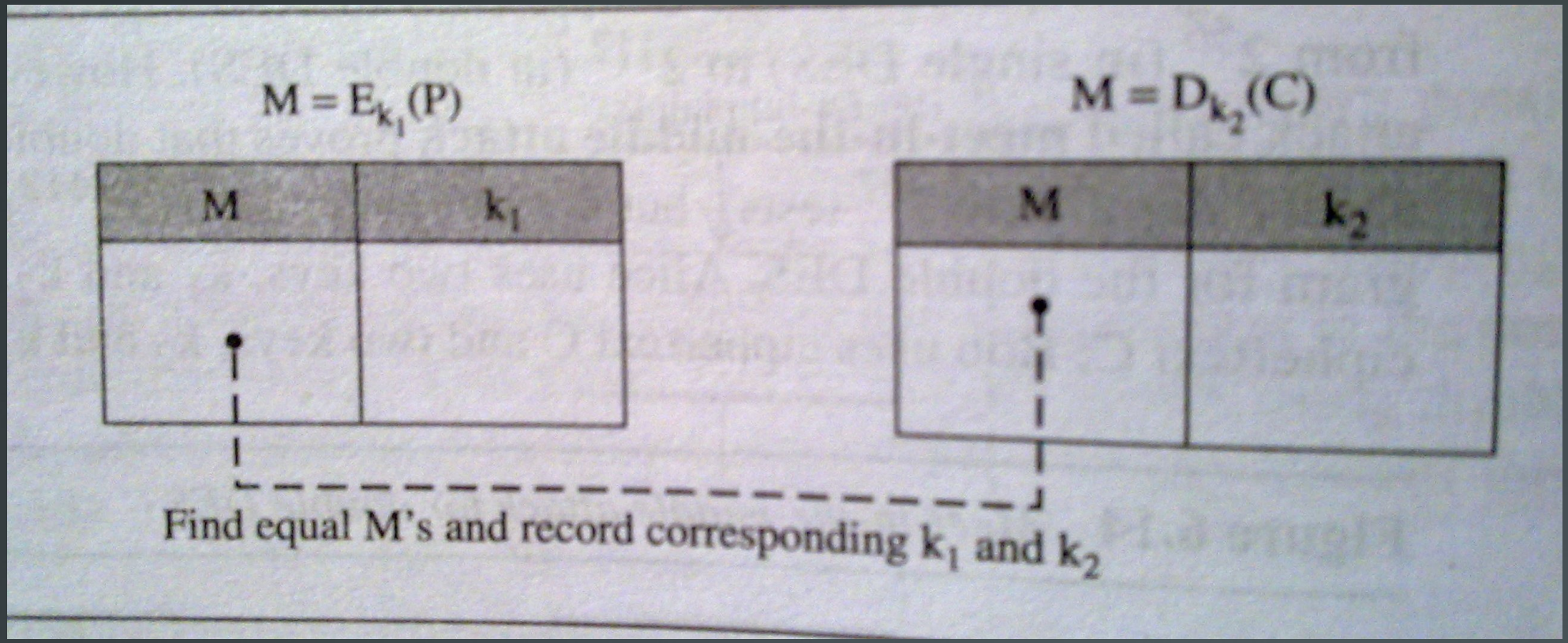
- 2 instances of DES ciphers for encryption & 2 instances of DES reverse ciphers for decryption



Meet-in-the-Middle attack

- Actually 2DES is vulnerable to 2^{57} bruteforce tests
- We have $\mathbf{M = E_{k1}(P)}$ and $\mathbf{M = D_{k2}(P)}$
- A known PT attack
  - Eve encrypts P using all 2^{56} values of k_1 and records all values M in a table
  - Eve decrypts C using all 2^{56} values of k_2 and records all values M in a table
  - Then compares

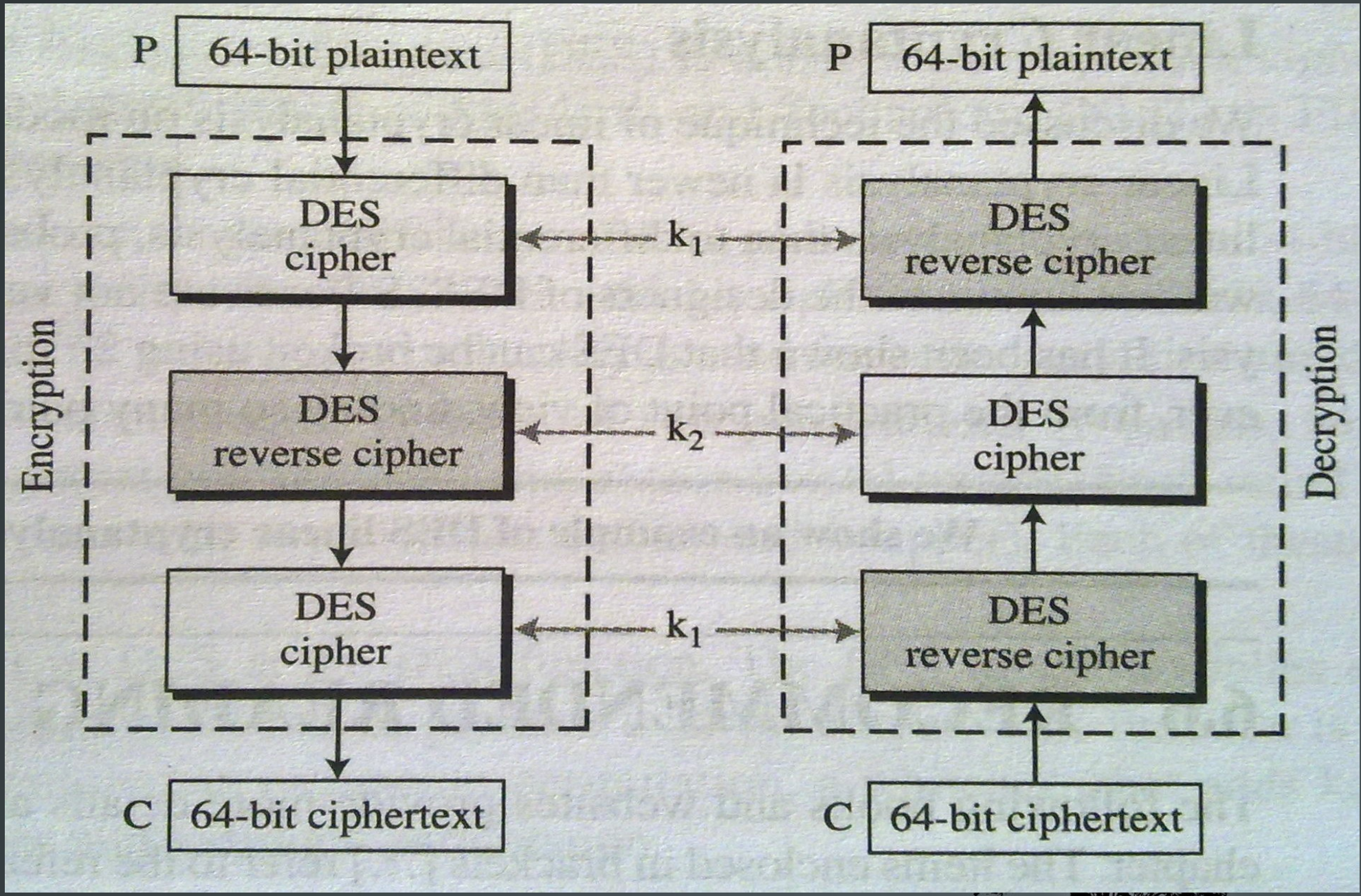
Meet-in-the-Middle attack



Triple DES (3DES)

- Two versions – with 2 keys & with 3 keys
- DES cipher used 3 times
- **3DES with 2 keys**
  - First and Third stages use key k_1
  - Second stage uses key k_2
  - At the encryption site, middle (second) stage uses DES decryption cipher and the other two stages use DES encryption cipher

Triple DES with 2 keys



3DES with 2 keys

- A msg encrypted with DES cipher can be decrypted with "3DES cipher with 2 keys" by assuming the keys $k_1 = k_2 = k$ (k is the key used for DES encr.)
- Was adopted by banking industry
- Effective key size = 112 bits

3DES with 3 keys

- Three "DES cipher stages" & a separate key each for each stage
- For compatibility with DES the encryption site uses EDE and the decryption site uses DED
- Used by many applications such as PGP (Pretty Good Privacy)
- Effective key size = 168 bits