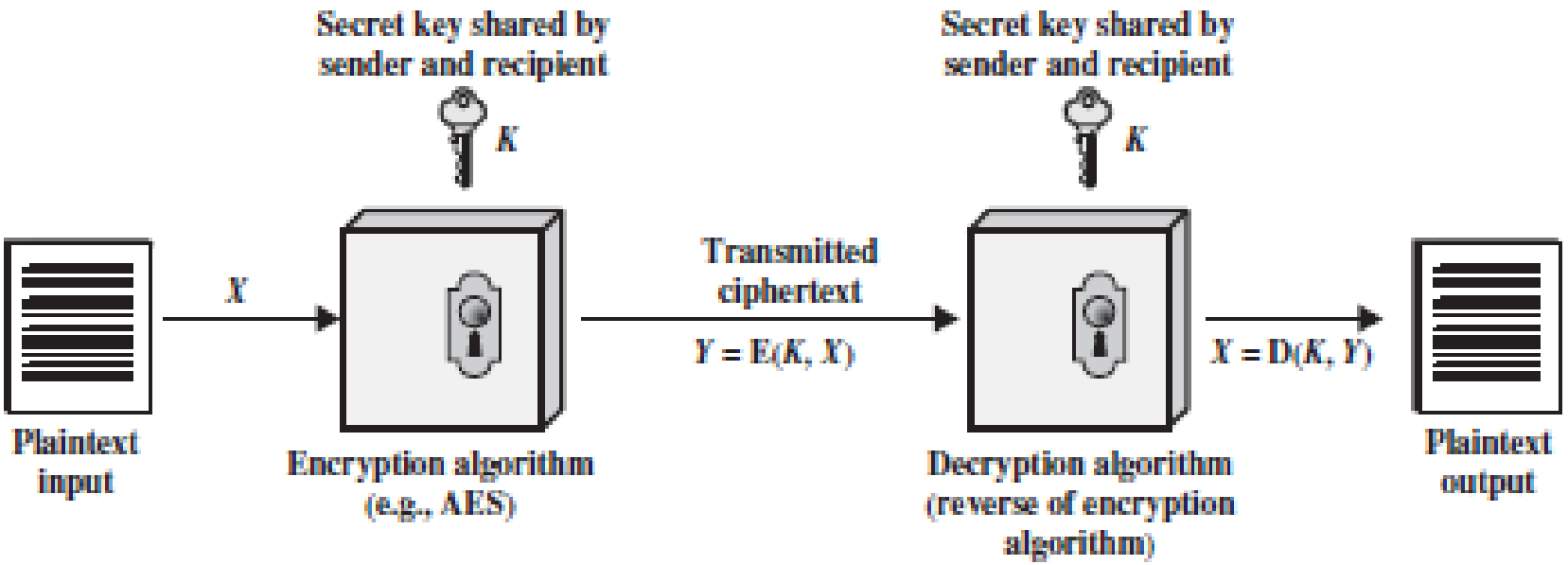
Data Encryption Standards (DES)

Dr. Shubhangi Sapkal

Symmetric cipher model

- **Plaintext:** This is the original intelligible message or data that is fed into the algorithm as input.
- **Encryption algorithm:** The encryption algorithm performs various substitutions and transformations on the plaintext.
- **Secret key:** The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.
- **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.
- **Decryption algorithm:** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.



Simplified Model of Symmetric Encryption

Cryptography

Cryptographic systems are characterized along three independent dimensions:

1. The type of operations used for transforming plaintext to ciphertext. All encryption algorithms are based on two general principles: substitution, in which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element, and transposition, in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost (i.e., that all operations are reversible). Most systems, referred to as *product systems*, involve multiple stages of substitutions and transpositions.

2. The number of keys used. If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption. If the sender and receiver use different keys, the system is referred to as asymmetric, two-key, or public-key encryption.

3. The way in which the plaintext is processed. A *block cipher* processes the input one block of elements at a time, producing an output block for each input block. A *stream cipher* processes the input elements continuously, producing output one element at a time, as it goes along.

Data Encryption Standard (DES)

- DES is a symmetric block cipher (shared secret key), with a key length of 56-bits. Published as the Federal Information Processing Standards (FIPS) 46 standard in 1977, DES was officially withdrawn in 2005

Data Encryption Standard (DES)

- The federal government originally developed DES encryption over 35 years ago to provide cryptographic security for all government communications. The idea was to ensure government systems all used the same, secure standard to facilitate interconnectivity.

Data Encryption Standard (DES)

To show that the DES was inadequate and should not be used in important systems anymore, a series of challenges were sponsored to see how long it would take to decrypt a message. Two organizations played key roles in breaking DES: distributed.net and the Electronic Frontier Foundation (EFF).

- The DES I contest (1997) took 84 days to use a brute force attack to break the encrypted message.
- In 1998, there were two DES II challenges issued. The first challenge took just over a month and the decrypted text was *"The unknown message is: Many hands make light work".* The second challenge took less than three days, with the plaintext message *"It's time for those 128-, 192-, and 256-bit keys".*
- The final DES III challenge in early 1999 only took 22 hours and 15 minutes. Electronic Frontier Foundation's Deep Crack computer (built for less than $250,000) and distributed.net's computing network found the 56-bit DES key, deciphered the message, and they (EFF & distributed.net) won the contest. The decrypted message read *"See you in Rome (Second AES Candidate Conference, March 22-23, 1999)",* and was found after checking about 30 percent of the key space...Finally proving that DES belonged to the past.

Data Encryption Standard (DES)

- The Data Encryption Standard (DES) is a secret key encryption scheme adopted as standard in the USA in 1977. It uses a 56-bit key, which is today considered by many to be insufficient as it can with moderate effort be cracked by brute force.
- A variant called Triple-DES (TDES or 3DES) uses a longer key and is more secure, but has never become popular. The Advanced Encryption Standard (AES) is expected to supersede DES (and 3DES) as the standard encryption algorithm.

Security of DES

- This secret key encryption algorithm uses a key that is 56 bits, or seven characters long.
- At the time it was believed that trying out all 72,057,594,037,927,936 possible keys (a seven with 16 zeros) would be impossible because computers could not possibly ever become fast enough. In 1998 the Electronic Frontier Foundation (EFF) built a special-purpose machine that could decrypt a message by trying out all possible keys in less than three days. The machine cost less than $250,000 and searched over 88 billion keys per second.

How DES works

- Encryption of a block of the message takes place in 16 stages or rounds.
- From the input key, sixteen 48 bit keys are generated, one for each round. In each round, eight so-called S-boxes are used. These S-boxes are fixed in the specification of the standard. Using the S-boxes, groups of six bits are mapped to groups of four bits. The contents of these S-boxes has been determined by the U.S. National Security Agency (NSA). The S-boxes appear to be randomly filled, but this is not the case. Recently it has been discovered that these S-boxes, determined in the 1970s, are resistant against an attack called differential cryptanalysis which was first known in the 1990s.

How DES works

- The block of the message is divided into two halves. The right half is expanded from 32 to 48 bits using another fixed table.
- The result is combined with the subkey for that round using the XOR operation. Using the S-boxes the 48 resulting bits are then transformed again to 32 bits, which are subsequently permutated again using yet another fixed table.
- This by now thoroughly shuffled right half is now combined with the left half using the XOR operation.
- In the next round, this combination is used as the new left half.

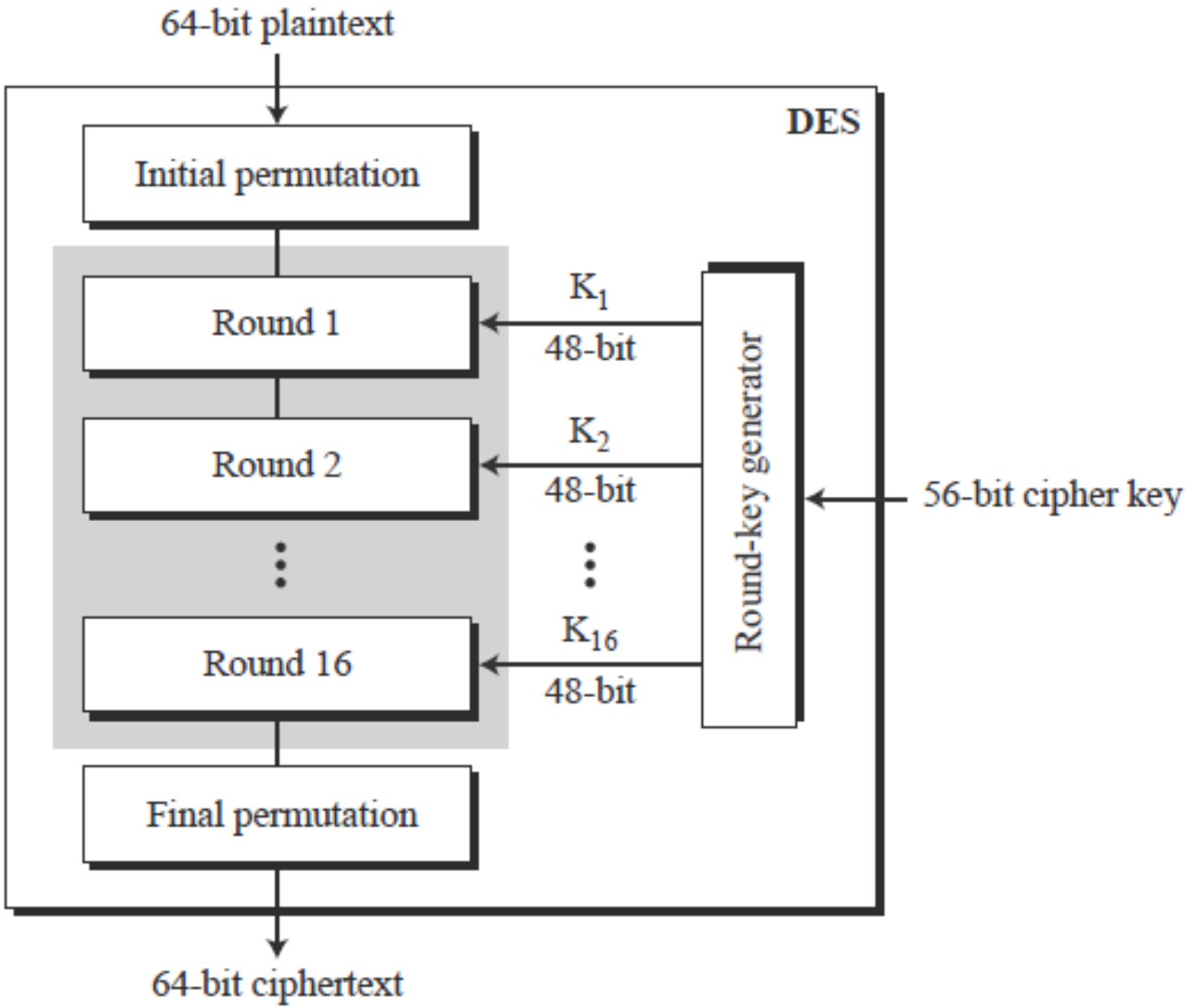
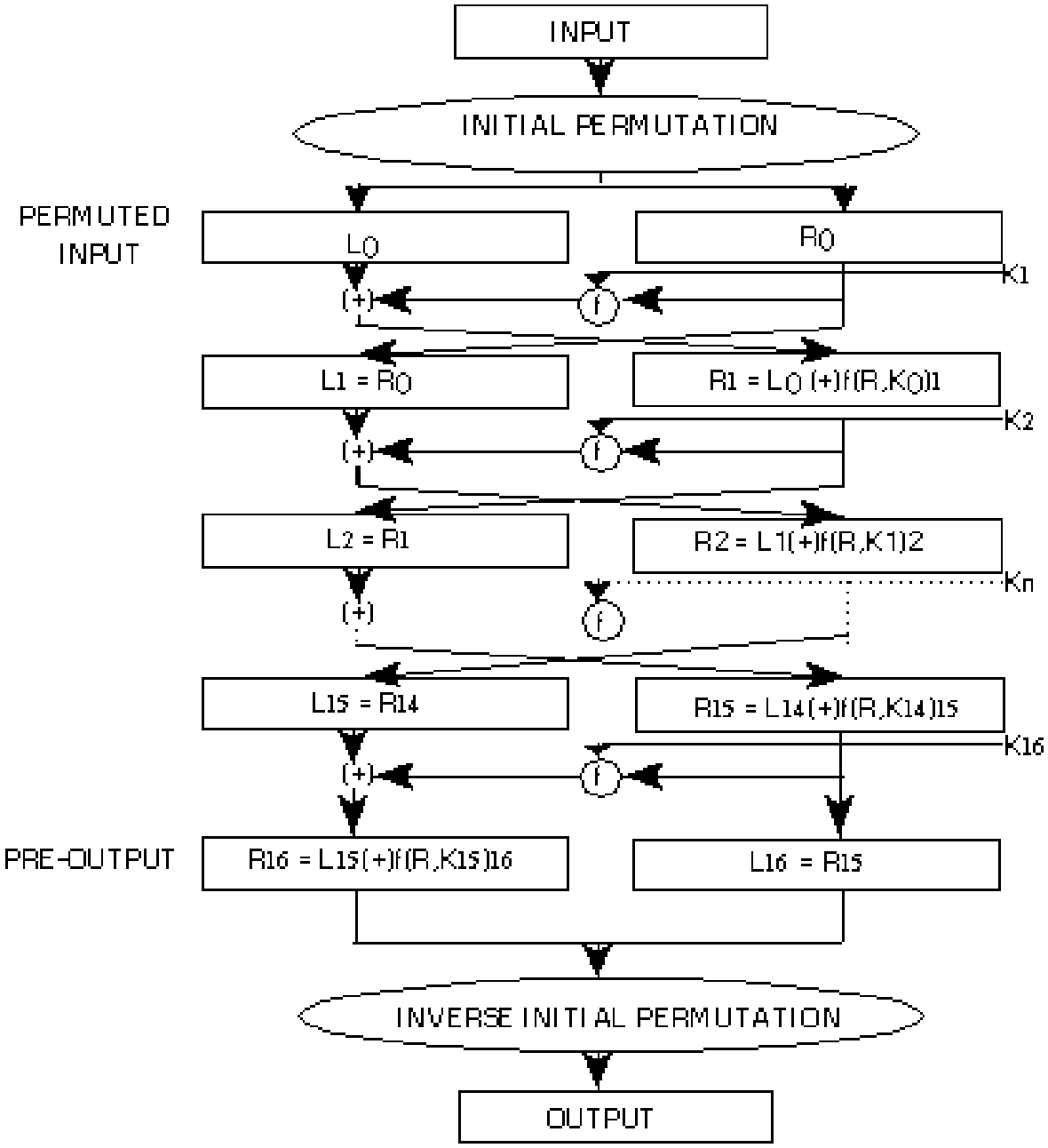


Figure: General structure of DES


INPUT
INITIAL PERMUTATION
PERMUTED INPUT
L0
R0
K1
(+)
f
L1 = R0
R1 = L0 (+)f(R,K0)1
K2
L2 = R1
R2 = L1(+)f(R,K1)2
Kn
L15 = R14
R15 = L14(+)f(R,K14)15
K16
PRE-OUTPUT
R16 = L15(+)f(R,K15)16
L16 = R15
INVERSE INITIAL PERMUTATION
OUTPUT

How DES works

- Plaintext is broken into blocks of length 64 bits. Encryption is blockwise.
- A message block is first gone through an initial permutation IP, then divided into two parts L_0,R0 where L_0 is the left part of 32 bits and R_0 is the right part of the 32 bits

How DES works

Round i has input L_{i-1},R_{i-1} and output L_i,R_i

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

and K_i is the subkey for the 'i'th

where $1 \le i \le 16$

$$L_1 = R_0, \quad R_1 = L_0 \oplus f(R_0,K_1)$$

$$L_2 = R_1, \quad R_2 = L_1 \oplus f(R_1,K_2)$$

$$L_3 = R_2, \quad R_3 = L_2 \oplus f(R_2,K_3)$$

....

....

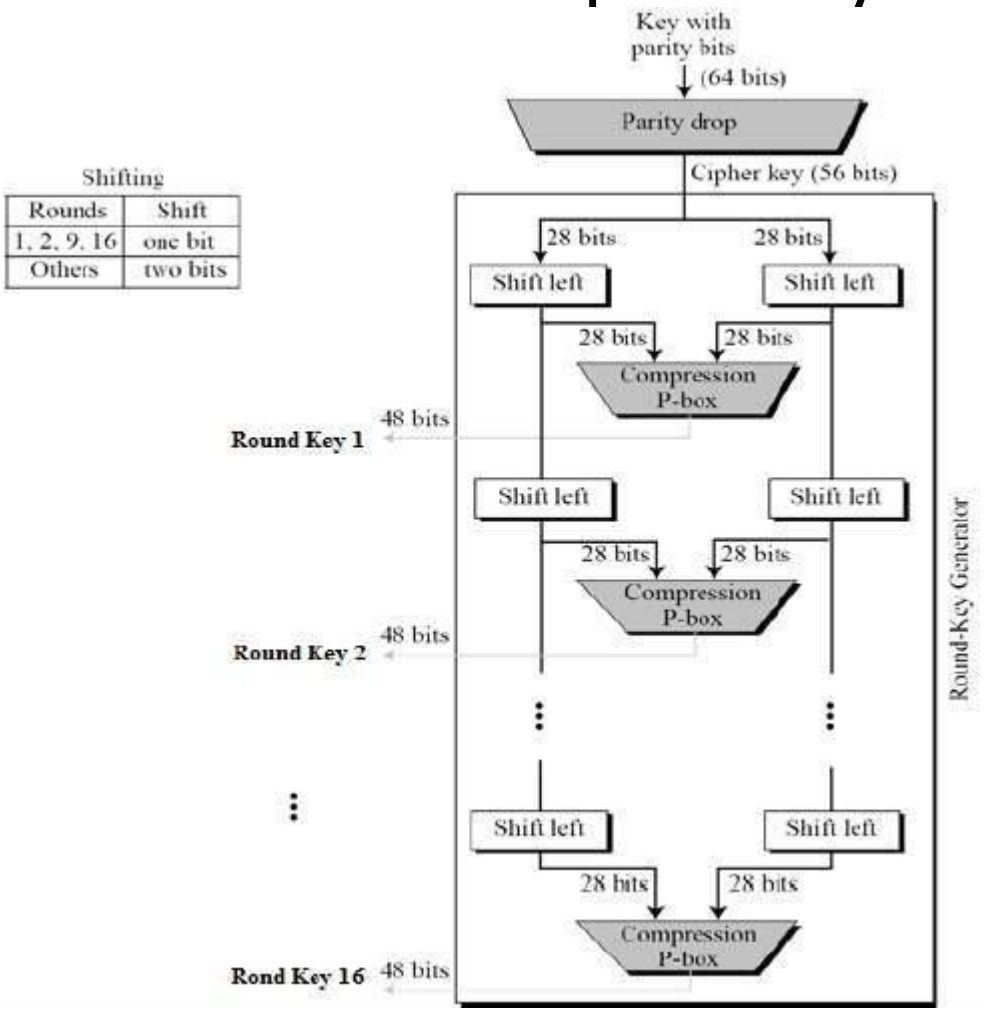
$$L_{16} = R_{15}, \quad R_{16} = L_{15} \oplus f(R_{15},K_{16})$$

How DES works

- After round 16,L_{16} and R_{16} are swapped, so that the decryption algorithm has the same structure as the encryption algorithm.
- Finally, the block is gone through the inverse the permutation IP^{-1} and then output
- One round of DES in very simple way during encryption

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

Initial and Final Permutations

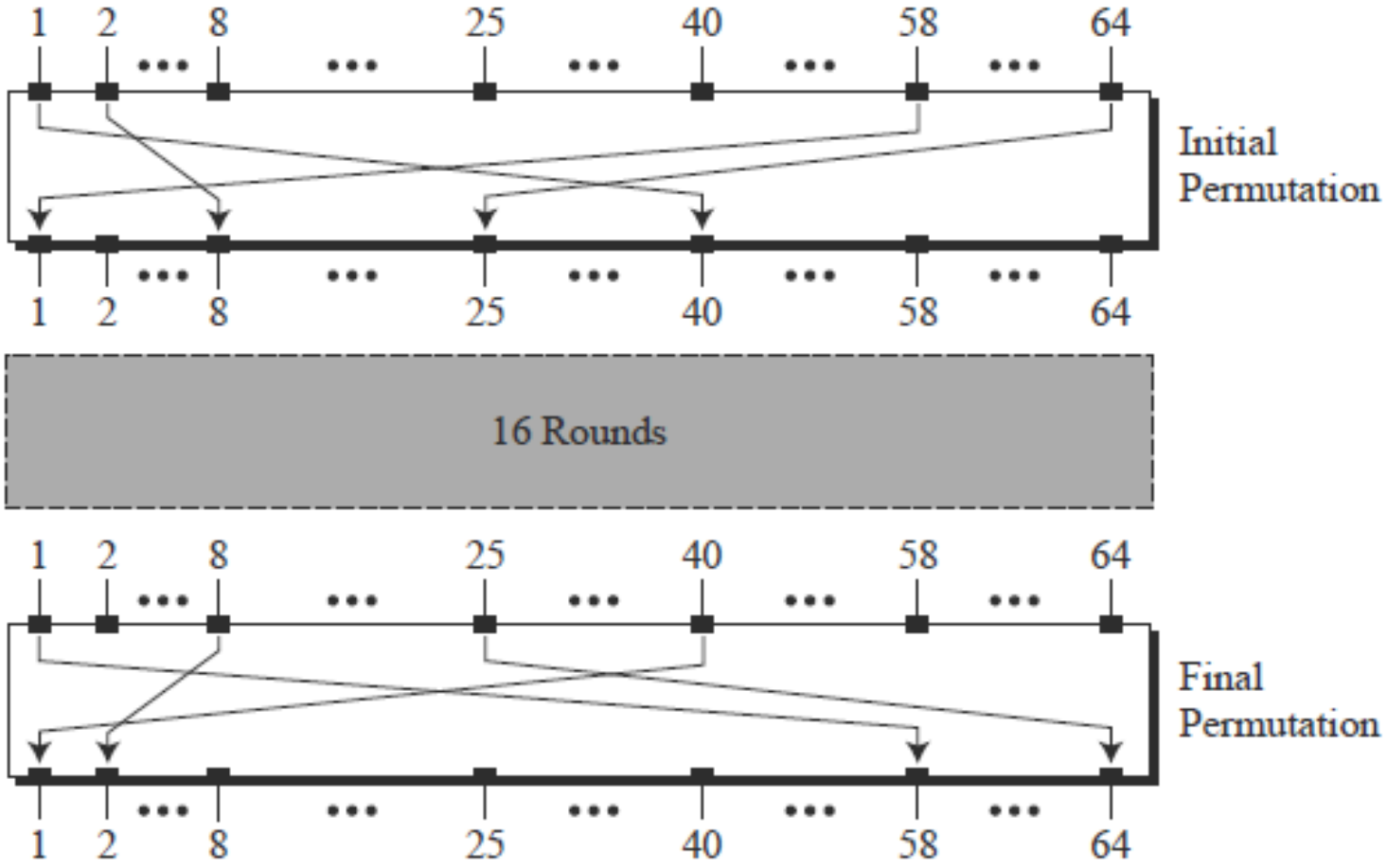


Figure: Initial and final permutation steps in DES

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

Figure: Initial and final permutation tables

Rounds

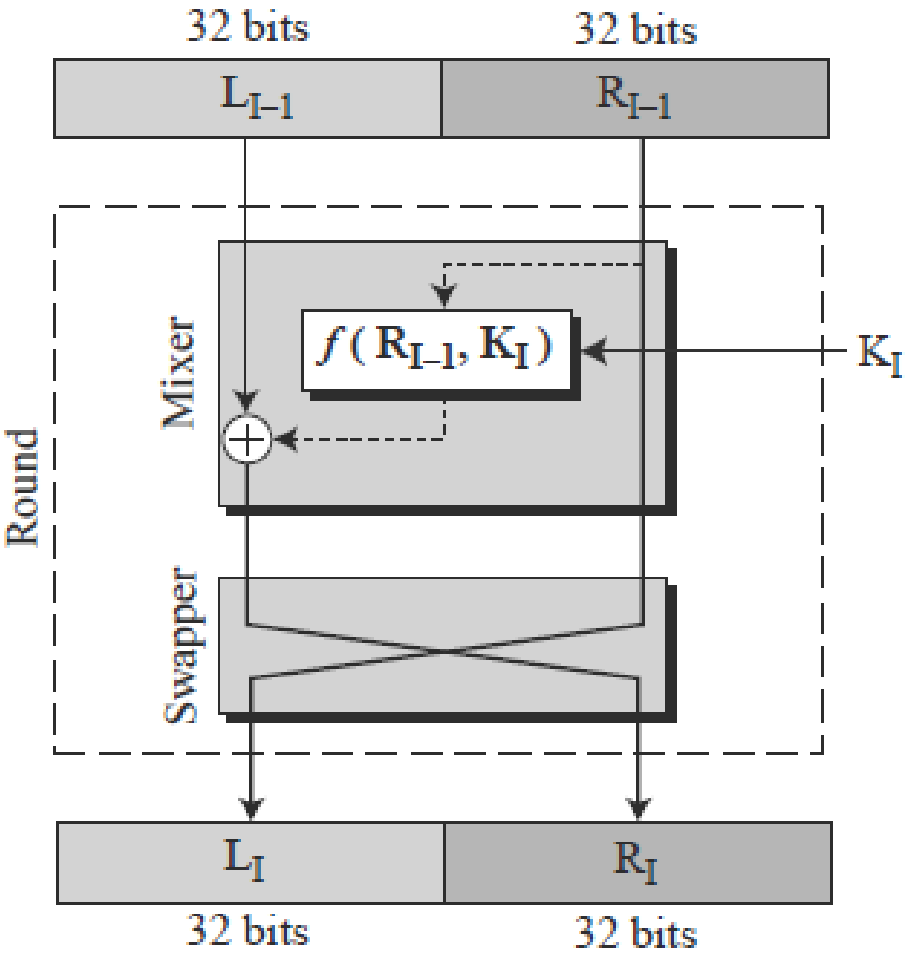


Figure: A round in DES

DES Function

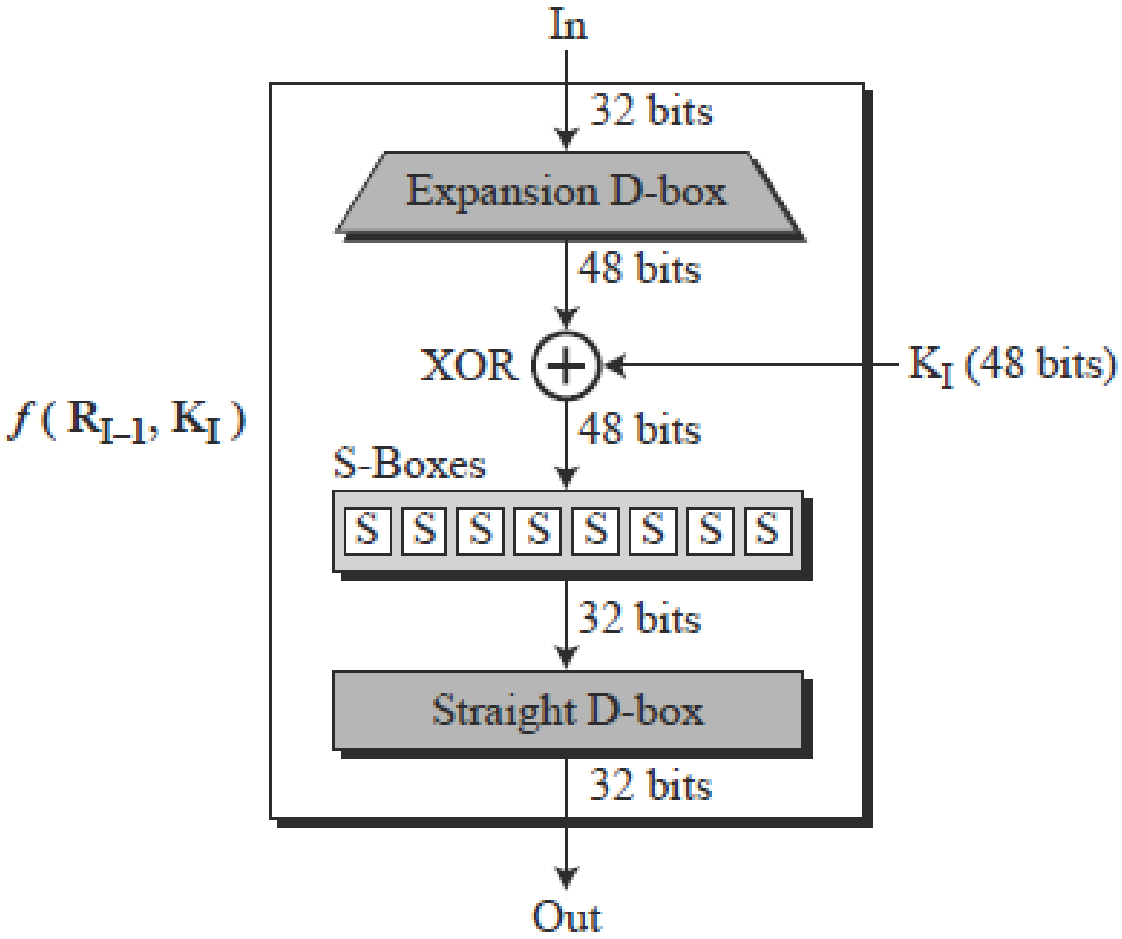


Figure: DES function

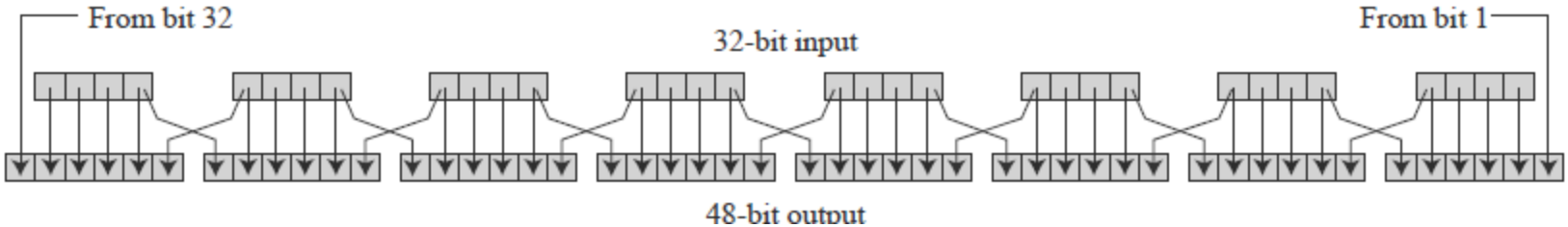


Figure: expansion permutation

How DES works

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

Figure: Expansion D-Box table

S-Boxes

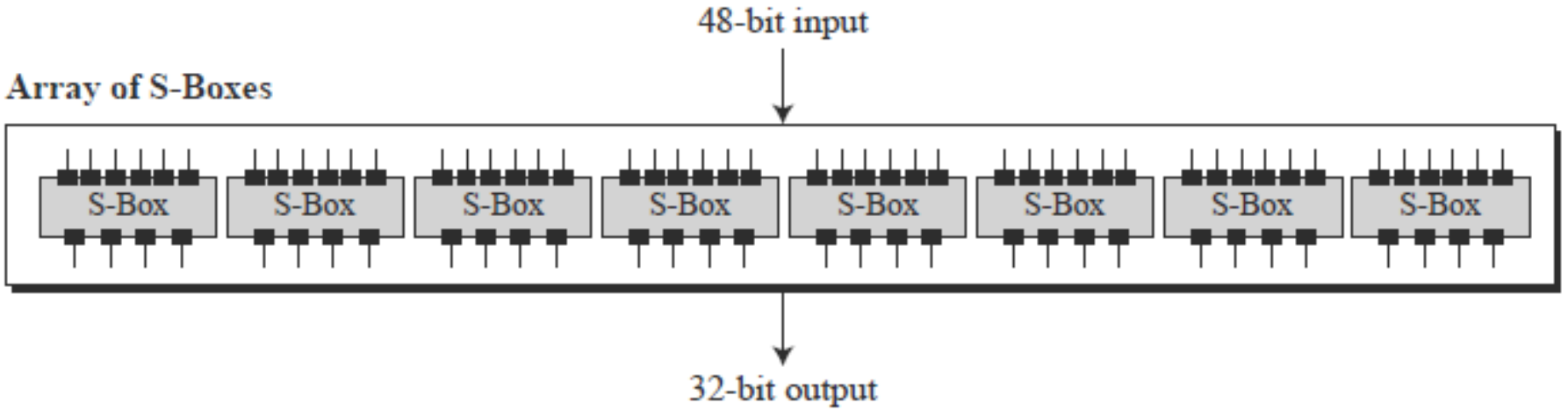


Figure: S-boxes

How DES works

- The 48-bit data from the second operation is divided into eight 6-bit chunks, and each chunk is fed into a box.
- The result of each box is a 4-bit chunk; when these are combined the result is a 32-bit text.
- The combination of bits 1 and 6 of the input defines one of four rows; the combination of bits 2 through 5 defines one of the sixteen columns

How DES works

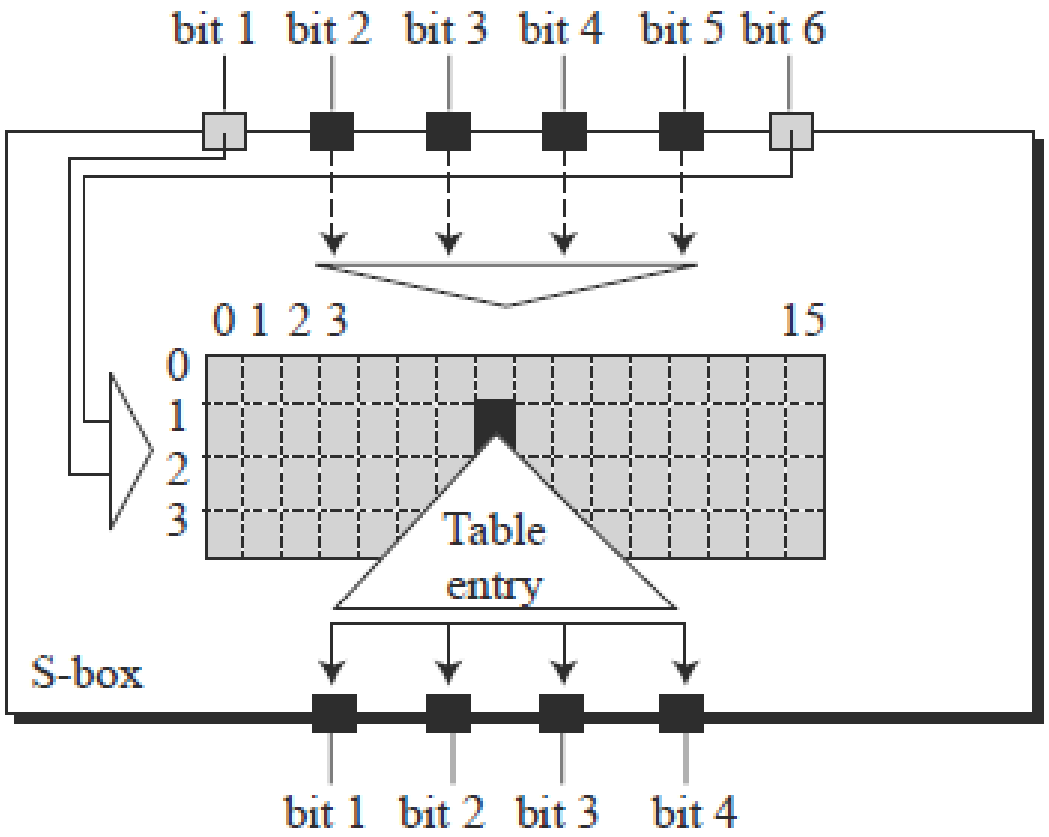


Figure: S-box rule

How DES works

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

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

Q1: The input to S-box 1 is **1**00001**1**. What is the output?

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

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

S-box 5

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

S-box 6

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

S-box 7

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

S-box 8

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

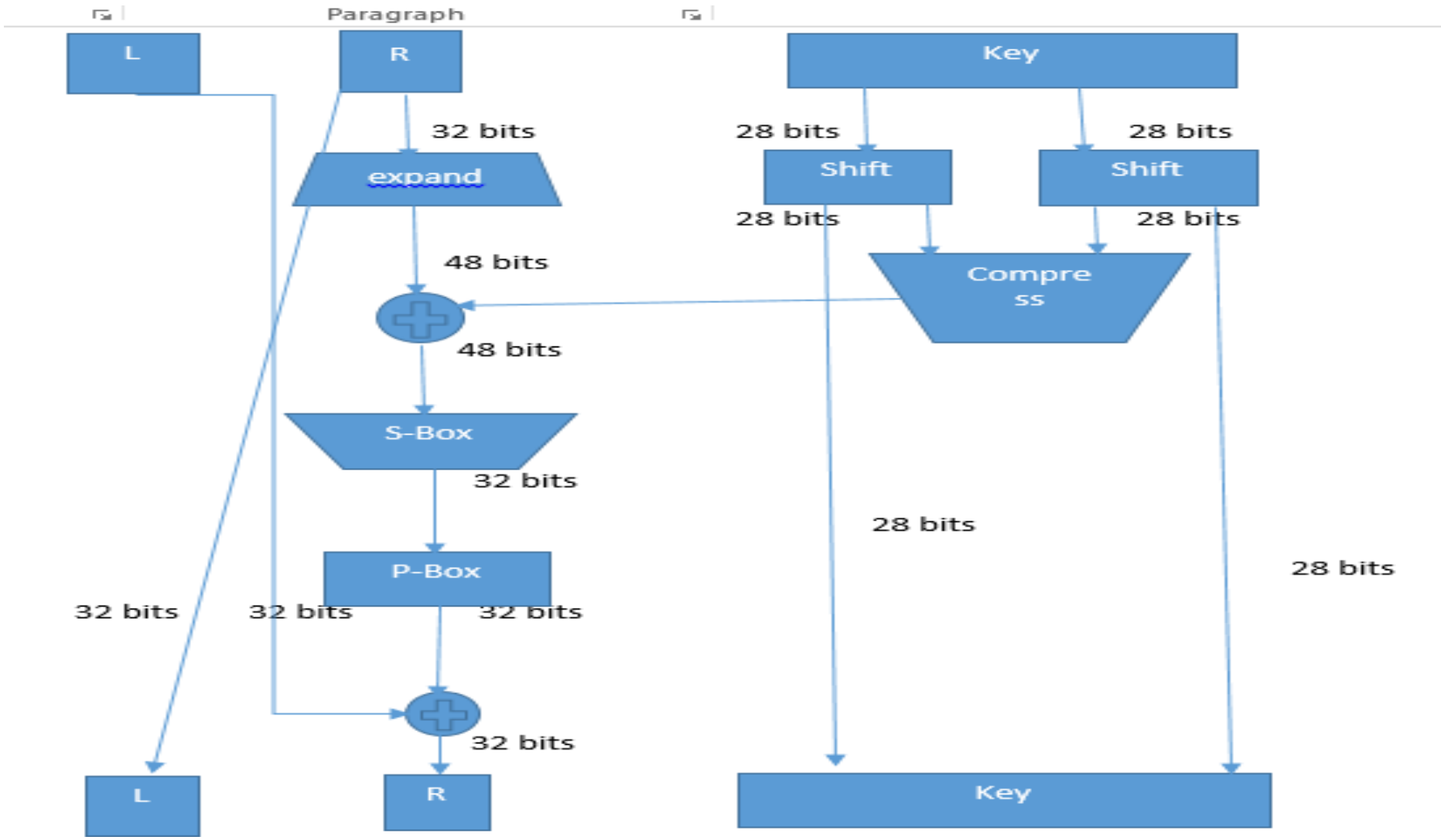
Final Permutation

- The last operation in the DES function is a permutation with a 32-bit input and a 32-bit output.
- For example, the seventh bit of the input becomes the second bit of the output.

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

One round of the DES



DES Modes of Operation

- The DES algorithm turns a 64-bit message block **M** into a 64-bit cipher block **C**. If each 64-bit block is encrypted individually, then the mode of encryption is called ***Electronic Code Book*** (ECB) mode.
- There are two other modes of DES encryption, namely ***Chain Block Coding*** (CBC) and ***Cipher Feedback*** (CFB), which make each cipher block dependent on all the previous messages blocks through an initial XOR operation.

Table 3.5 Average Time Required for Exhaustive Key Search

Key Size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 Decryptions/s	Time Required at 10^{13} Decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years
26 characters (permutation)	Monoalphabetic	$2! = 4 \times 10^{26}$	2×10^{26} ns = 6.3×10^{9} years	6.3×10^{6} years

Triple-DES

- Triple-DES is just DES with two 56-bit keys applied. Given a plaintext message, the first key is used to DES-encrypt the message. The second key is used to DES-decrypt the encrypted message. (Since the second key is not the right key, this decryption just scrambles the data further.) The twice-scrambled message is then encrypted again with the first key to yield the final ciphertext. This three-step procedure is called triple-DES.
- Triple-DES is just DES done three times with two keys used in a particular order. (Triple-DES can also be done with three separate keys instead of only two. In either case the resultant key space is about 2^112.)