Information Security
Fall 2022

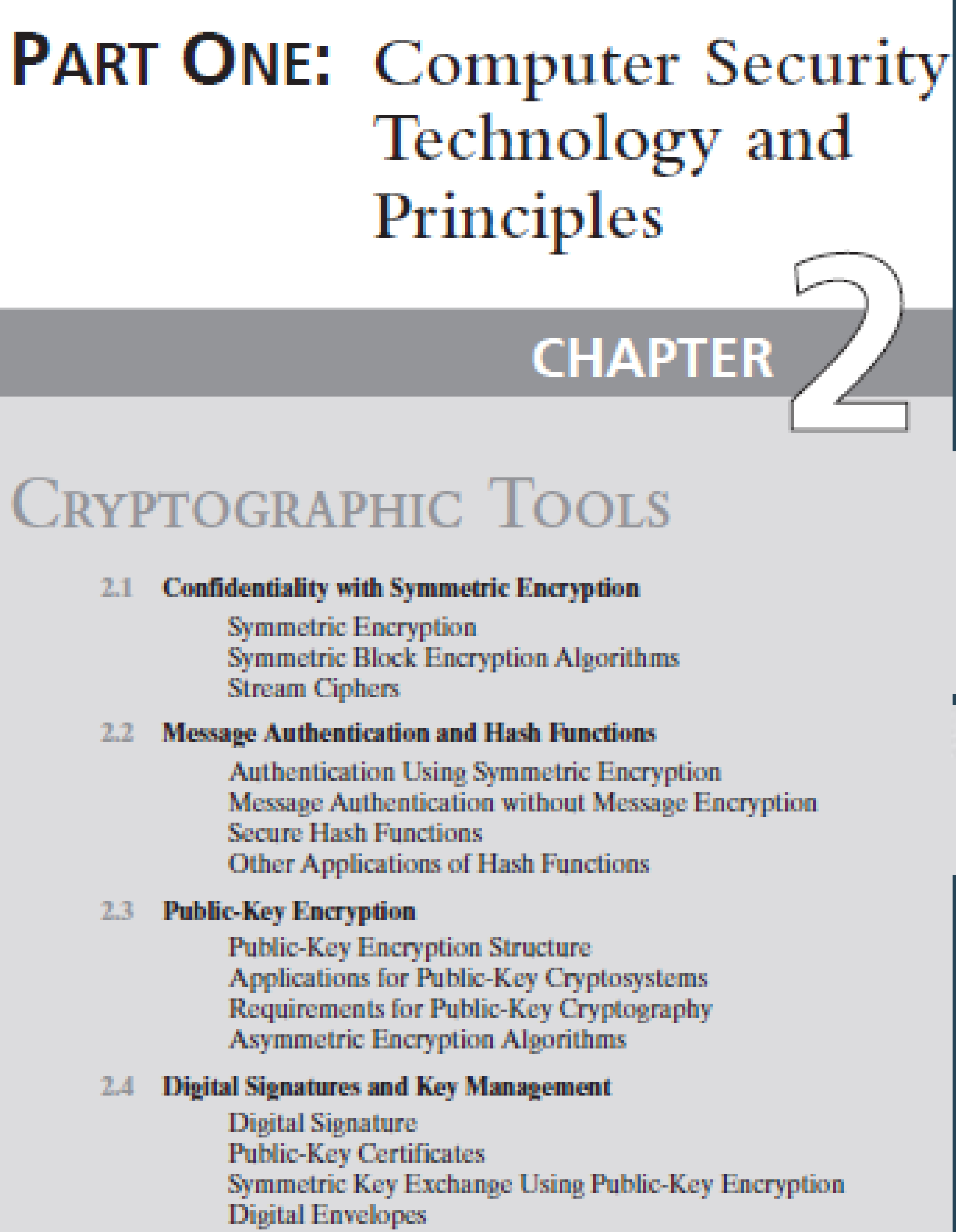
Week # 2

Lecture # 4, 5 and 6

Dr. Aqsa Aslam

Topics from text book:

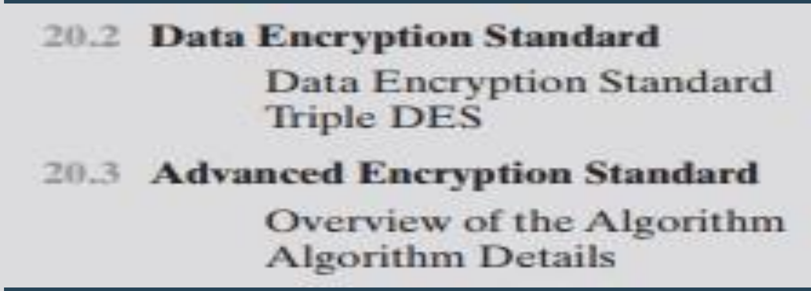
- Chapter # 2
- Chapter # 20
- Chapter # 21

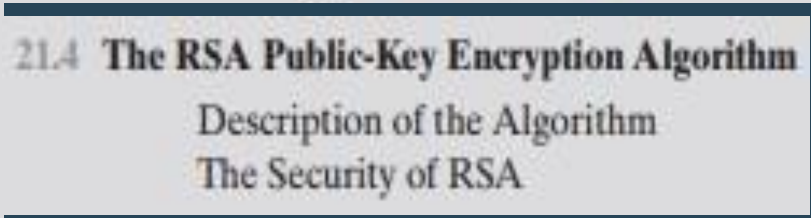


PART ONE: Computer Security Technology and Principles

CHAPTER 2

CRYPTOGRAPHIC TOOLS





Symmetric Encryption

- The universal technique for providing confidentiality for transmitted or stored data
  - Also referred to as conventional encryption or single-key encryption
- Two requirements for secure use:
  - Need a strong encryption algorithm
  - Sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure
- Types of Symmetric Encryption
  - ***Block Ciphers***
  - ***Stream Ciphers***
- Symmetric Algorithms
  - ***DES and AES***

Symmetric Encryption

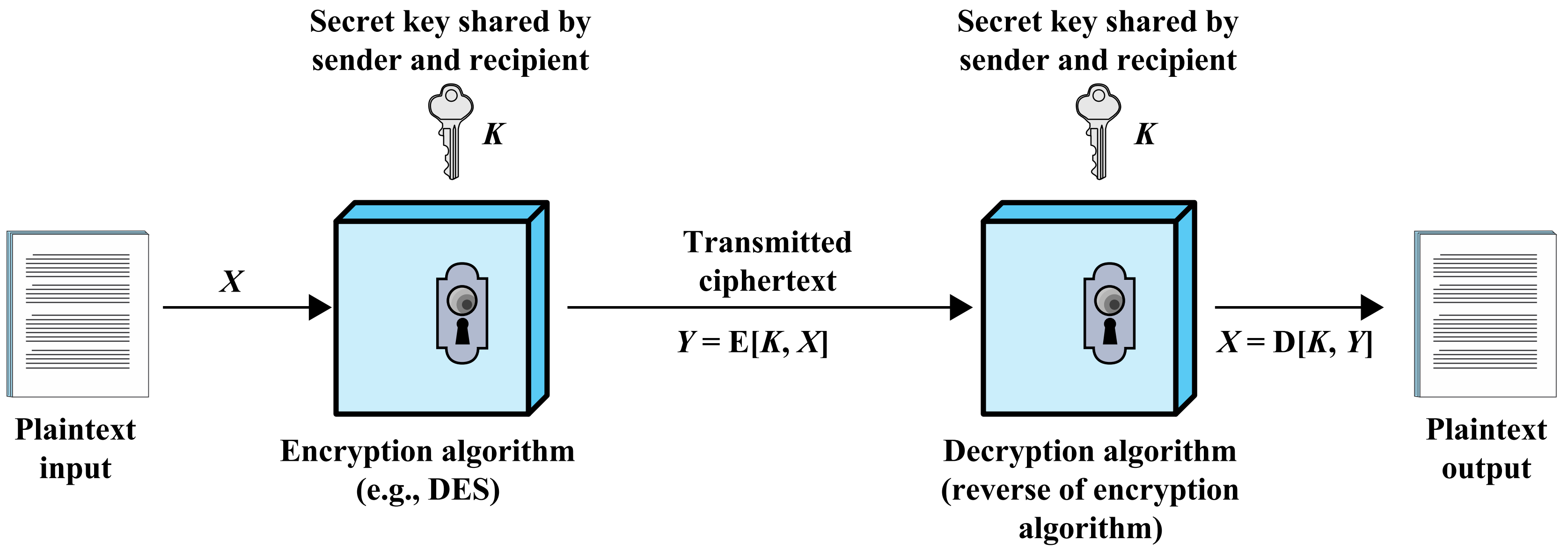

Secret key shared by sender and recipient
K
Secret key shared by sender and recipient
K
X
Transmitted ciphertext
Y = E[K, X]
X = D[K, Y]
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Figure 2.1 Simplified Model of Symmetric Encryption

Symmetric Encryption

- A symmetric encryption scheme has five ingredients
  1. ***Plaintext:*** This is the original message or data that is fed into the algorithm as input.
  2. ***Encryption algorithm:*** The encryption algorithm performs various substitutions and transformations on the plaintext.
  3. ***Secret key:*** The secret key is also input to the encryption algorithm. The exact substitutions and transformations performed by the algorithm depend on the key.
  4. ***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.
  5. ***Decryption algorithm:*** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

Attacking Symmetric Encryption

Cryptanalytic Attacks

- Rely on:
  - Nature of the algorithm
  - Some knowledge of the general characteristics of the plaintext
  - Some sample plaintext-ciphertext pairs
- Exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or the key being used
  - If successful all future and past messages encrypted with that key are compromised

Brute-Force Attacks

- Try all possible keys on some ciphertext until an intelligible translation into plaintext is obtained
  - On average half of all possible keys must be tried to achieve success

Symmetric Block Encryption Algorithms

- ***Block Ciphers***
  - The most commonly used symmetric encryption algorithms are block ciphers.
  - A block cipher processes the plaintext input in fixed-size blocks and produces a block of ciphertext of equal size for each plaintext block.
  - The algorithm processes longer plaintext amounts as a series of fixed-size blocks.
  - The most important symmetric algorithms, all of which are block ciphers, are the:
    - Data Encryption Standard (DES),
    - Triple DES (3DES)
    - The Advanced Encryption Standard (AES)

Data Encryption Standard (DES)

- Until recently was the most widely used encryption scheme
  - FIPS PUB 46
  - Referred to as the Data Encryption Algorithm (DEA)
  - Uses 64 bit plaintext block and 56 bit key to produce a 64 bit ciphertext block

- Strength concerns:
  - Concerns about the algorithm itself

    DES is the most studied encryption algorithm in existence
  - Concerns about the use of a 56-bit key

    The speed of commercial off-the-shelf processors makes this key length woefully inadequate

Triple DES (3DES)

- Repeats basic DES algorithm three times using either two or three unique keys
- First standardized for use in financial applications in ANSI standard X9.17 in 1985
- ***Attractions:***
  - 168-bit key length overcomes the vulnerability to brute-force attack of DES
  - Underlying encryption algorithm is the same as in DES
- ***Drawbacks:***
  - Algorithm is sluggish in software
  - Uses a 64-bit block size

Advanced Encryption Standard (AES)

Needed a replacement for 3DES

3DES was not reasonable for long term use

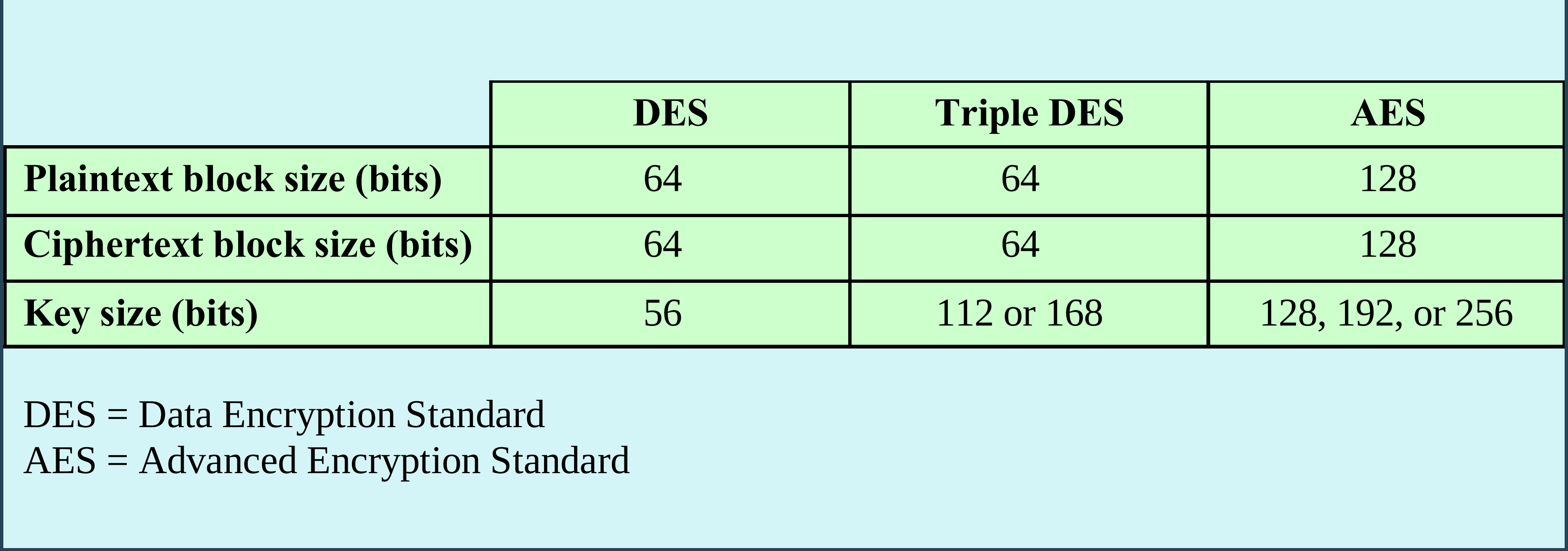
NIST called for proposals for a new AES in 1997

- Should have a security strength equal to or better than 3DES
- Significantly improved efficiency
- Symmetric block cipher
- 128 bit data and 128/192/256 bit keys

Selected Rijndael in November 2001

Published as FIPS 197

Table 2.1



	DES	Triple DES	AES
Plaintext block size (bits)	64	64	128
Ciphertext block size (bits)	64	64	128
Key size (bits)	56	112 or 168	128, 192, or 256

DES = Data Encryption Standard
AES = Advanced Encryption Standard

Comparison of Three Popular Symmetric Encryption Algorithms

Table 2.2

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

Average Time Required for Exhaustive Key Search

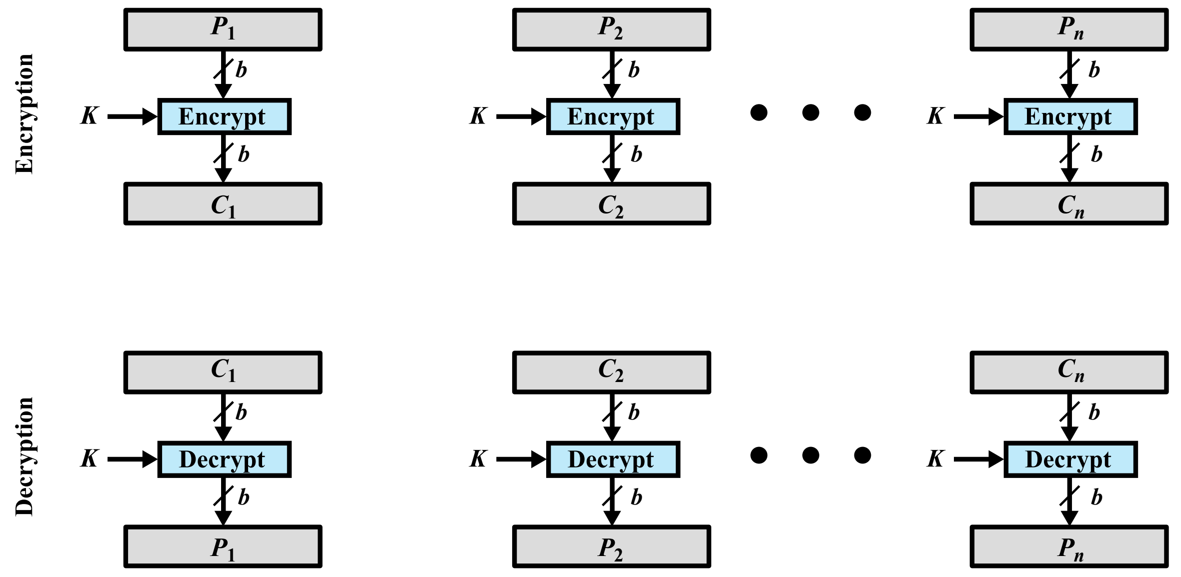
Practical Security Issues

- Typically symmetric encryption is applied to a unit of data larger than a single 64-bit or 128-bit block
- Different modes of operations on block cipher
  - means how data can be encrypted and decrypted
  - **ECB,** CBC, CTR
- The simplest approach to multiple-block encryption is known as ***electronic codebook (ECB) mode***
  - plaintext is handled **b bits** at a time and each block of plaintext is encrypted using the same key.
    - b=64 or b=128
  - Each block size encrypt and decrypt process is independently.

- Cryptanalysts may be able to exploit regularities in the plaintext
- Modes of operation
  - Alternative techniques developed to increase the security of symmetric block encryption for large sequences
  - Overcomes the weaknesses of ECB

Electronic codebook (ECB)

- Encryption and decryption in blocks (e.g., 64 or 128 bit)



(a) Block cipher encryption (electronic codebook mode)

Computationally Secure Encryption Schemes

- Encryption is computationally secure if:
  - Cost of breaking cipher exceeds value of information
  - Time required to break cipher exceeds the useful lifetime of the information
- Usually very difficult to estimate the amount of effort required to break
- Can estimate time/cost of a brute-force attack

Feistel Cipher Structure

- The Feistel structure is a particular example of the more general structure used by all symmetric block ciphers
- Symmetric block cipher consists of:
  - ***A sequence of rounds***
  - ***With substitutions and permutations controlled by key***
- A symmetric block cipher depends on the choice of the following parameters and design features:
  - Block size
  - Key size
  - Number of rounds
  - Subkey generation algorithm
  - Round function
  - Fast software encryption/decryption
  - Ease of analysis

Feistel Cipher Structure

Steps:

- ❑ The inputs to the encryption algorithm are:
  - Plaintext block of length **2w** bits
  - A key **K**
- ❑ The plaintext block is divided into two halves, **L0** and **R0**
- ❑ The two halves of the data pass through **n** rounds of processing and then combine to produce the ciphertext block.
- ❑ Each round **i** has as inputs **Li-1** and **Ri-1**, derived from the previous round, as well as a subkey **Ki** , derived from the overall **K**
- ❑ In general, the subkeys **Ki** are different from **K** and from each other, and are generated from the key by a subkey generation algorithm
- ❑ All rounds have the same structure
- ❑ 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 exclusive-OR (XOR) of the output of that function and the left half of the data.
- ❑ The round function has the same general structure for each round but is parameterized by the round subkey Ki .
- ❑ Following this substitution, a permutation is performed that consists of the interchange of the two halves of the data.

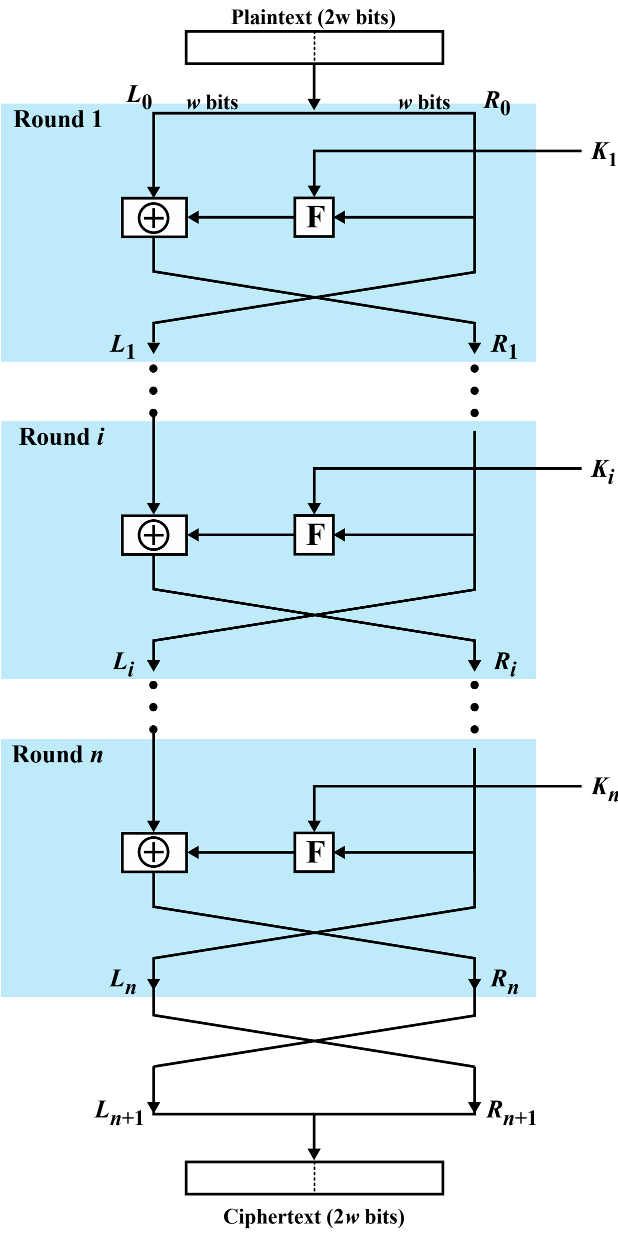


Figure 20.1 Classical Feistel Network

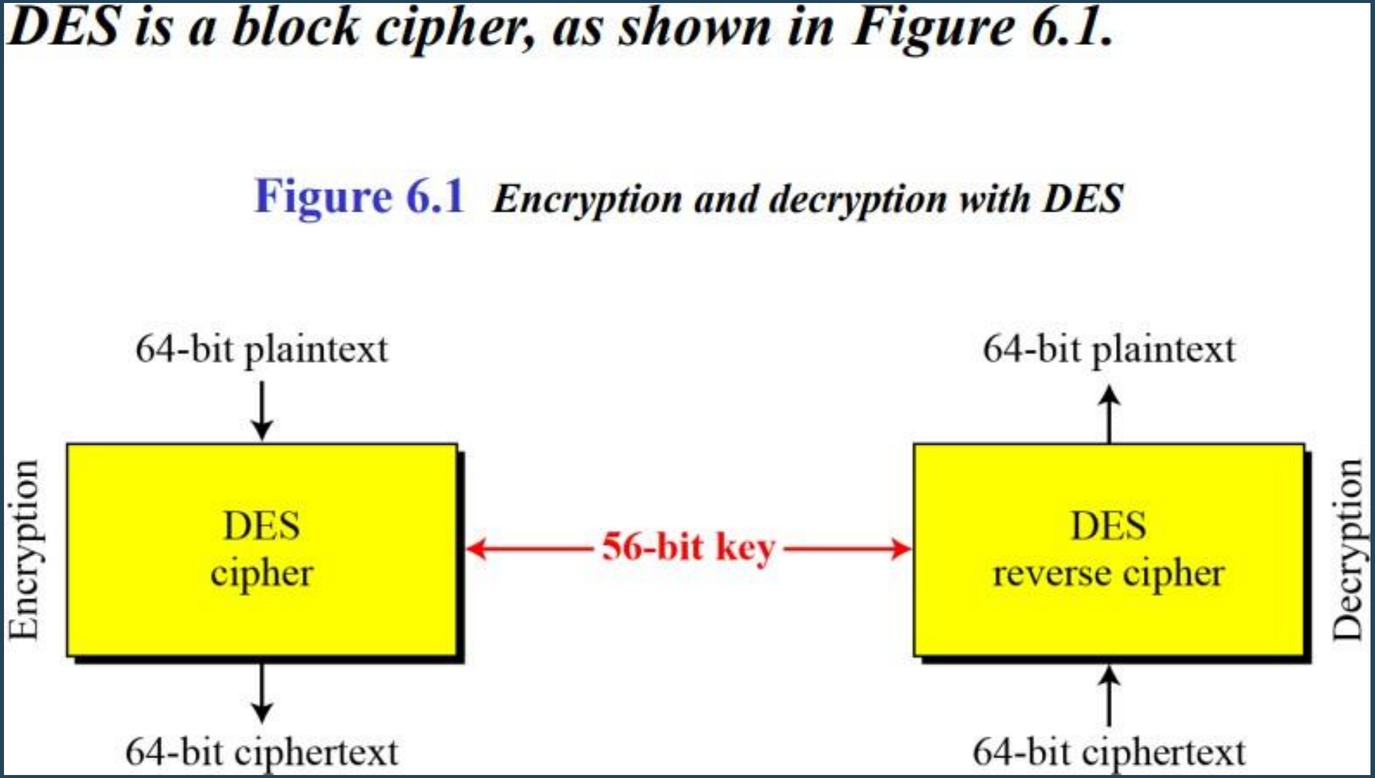
Data Encryption Standard (DES)

- The DES algorithm can be described as follows.
  - The plaintext is 64 bits in length
  - The key is 56 bits in length
  - There are 16 rounds of processing.
  - From the original 56-bit key, 16subkeys are generated (ki= 48 bit)
    - one of which is used for each round.

- **Encryption process:**
  1. Permutation performed on the input block
  2. Generation of round keys
  3. Performing 16 identical rounds
     a. Substitution
     b. Permutation
  4. Inverse permutation to step 1

- **Decryption process:**
  - Use the ciphertext as input to the DES algorithm, but use the sub-keys Ki in reverse order.
  - That is, use K16 on the first iteration, K15 on the second iteration, and so on until K1 is used on the sixteenth and last iteration.

DES block cipher



DES is a block cipher, as shown in Figure 6.1.

Figure 6.1 ***Encryption and decryption with DES***

Encryption

64-bit plaintext

DES
cipher

64-bit ciphertext

56-bit key

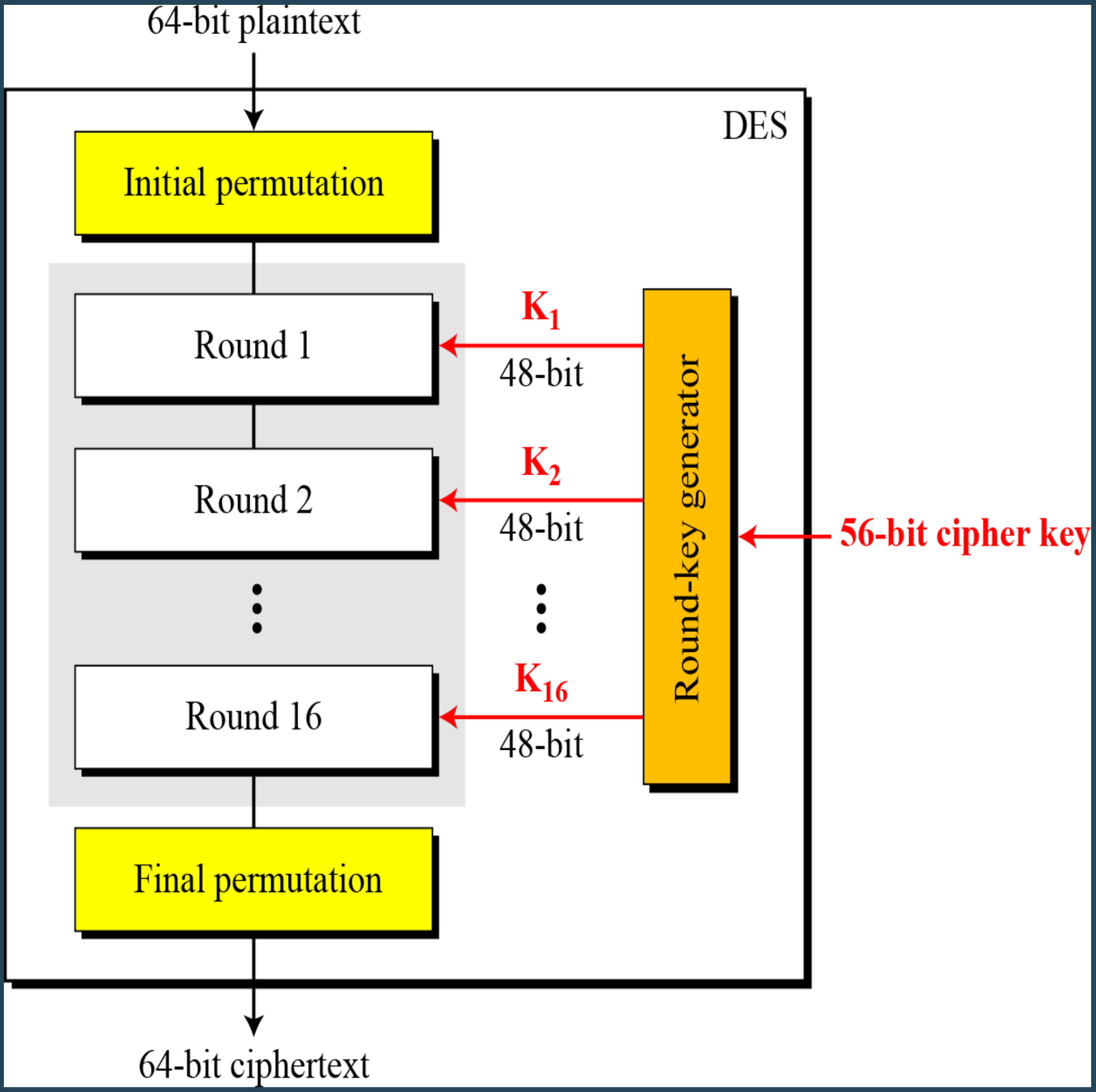
Decryption

64-bit plaintext

DES
reverse cipher

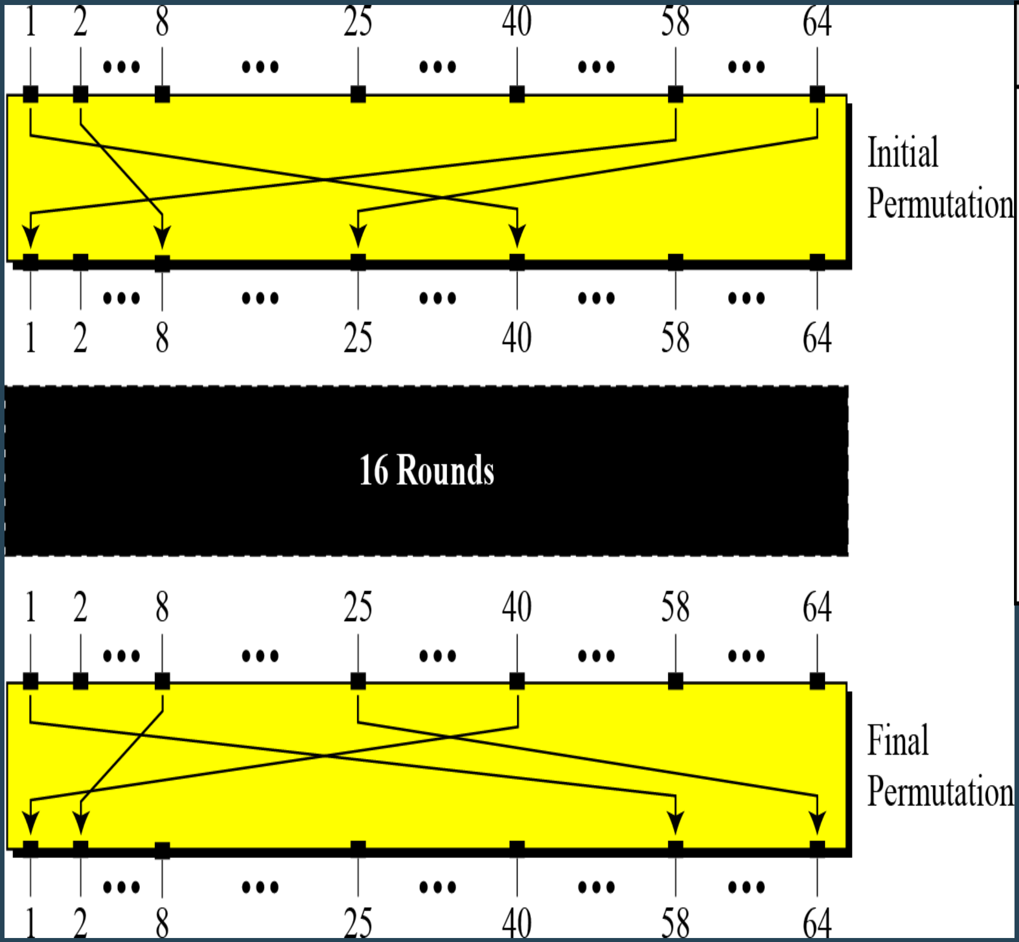
64-bit ciphertext

Figure 6.2 *General structure of DES*



Note: The initial and final permutations are *straight P-boxes* that are inverses of each other. They have no cryptography significance in DES. Both permutations are keyless and predetermined.

Figure 6.3 *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

Rule:
In the initial permutation, the 58th bit in the input becomes the first bit in the output. Similarly, in the final permutation, the first bit in the input becomes the 58th bit in the output.

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

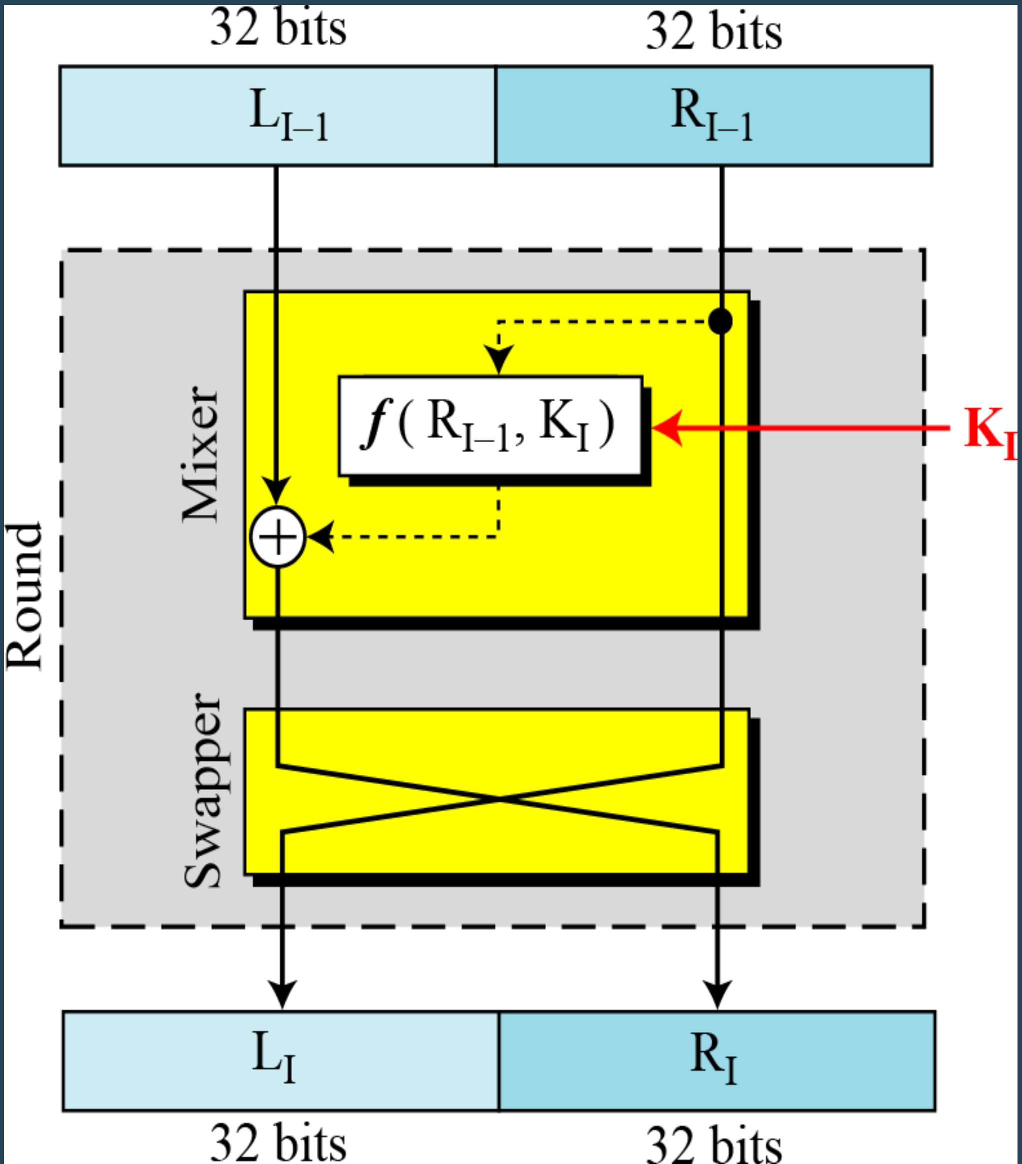
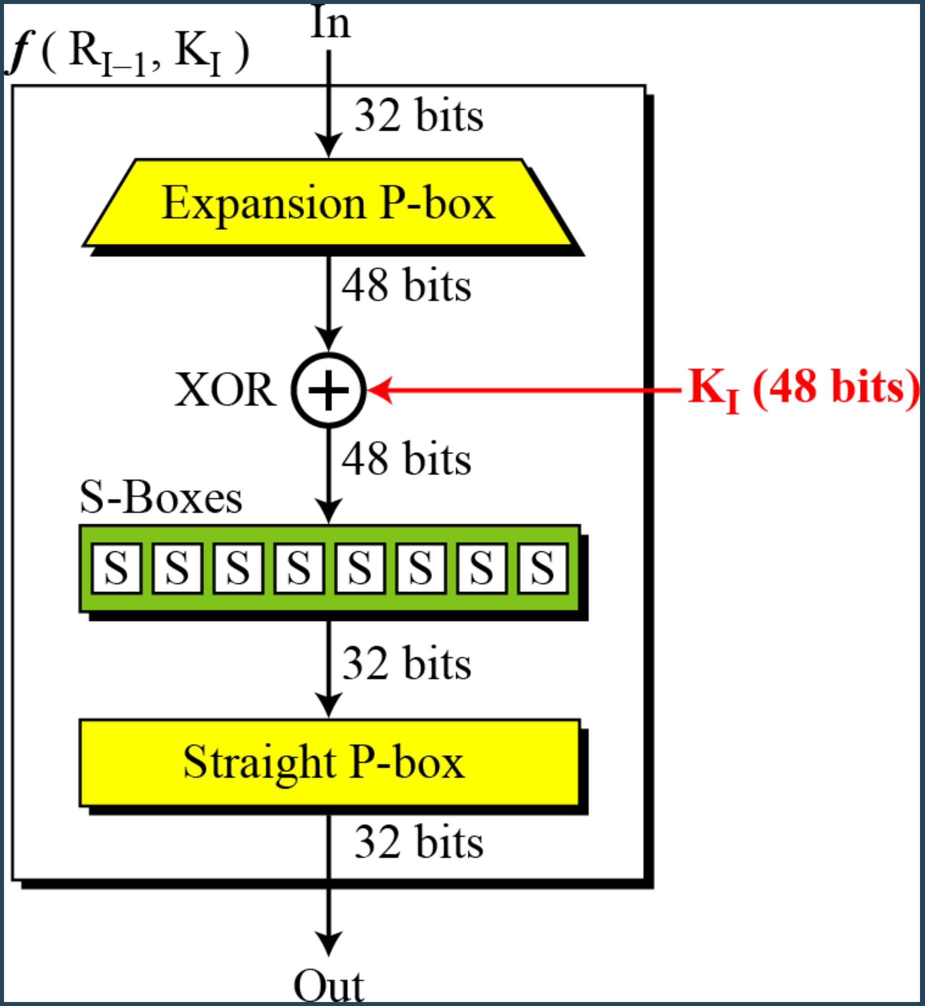


Figure 6.4 ***A round in DES (encryption site)***

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.

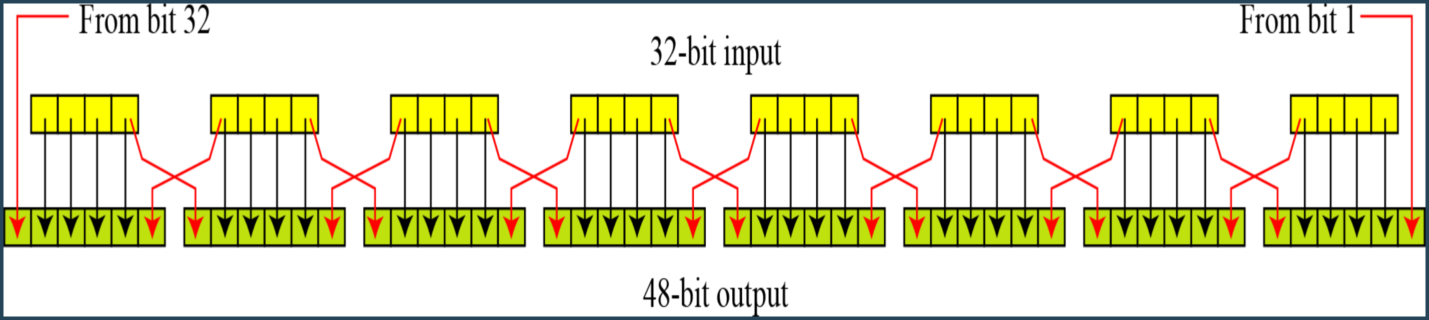
Figure 6.5: ***DES function***



Expansion P-box

Since R_{I-1} is a 32-bit input and K_I is a 48-bit key, we first need to expand R_{I-1} to 48 bits.

Figure 6.6 ***Expansion permutation***



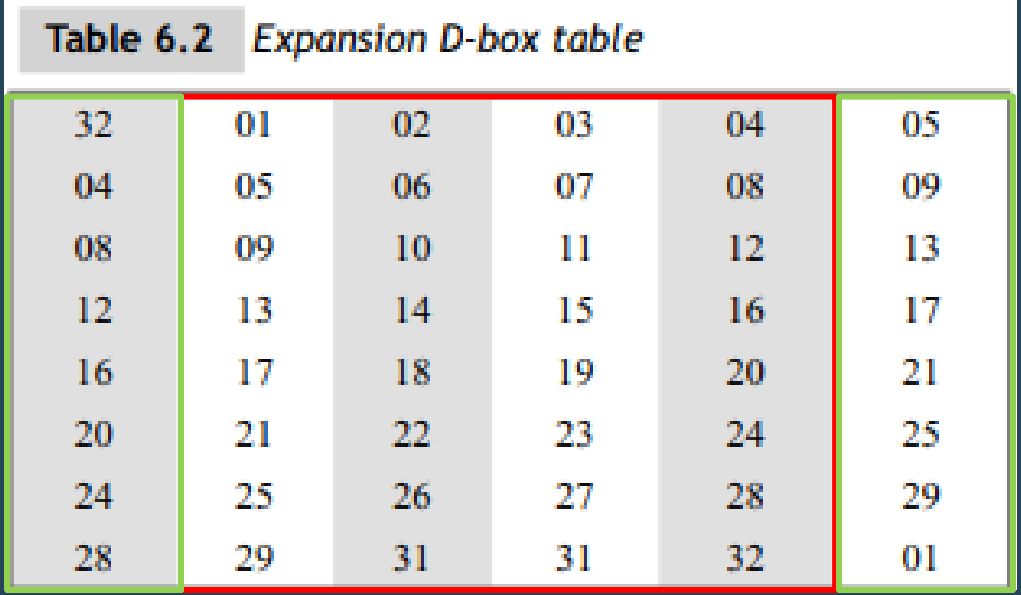


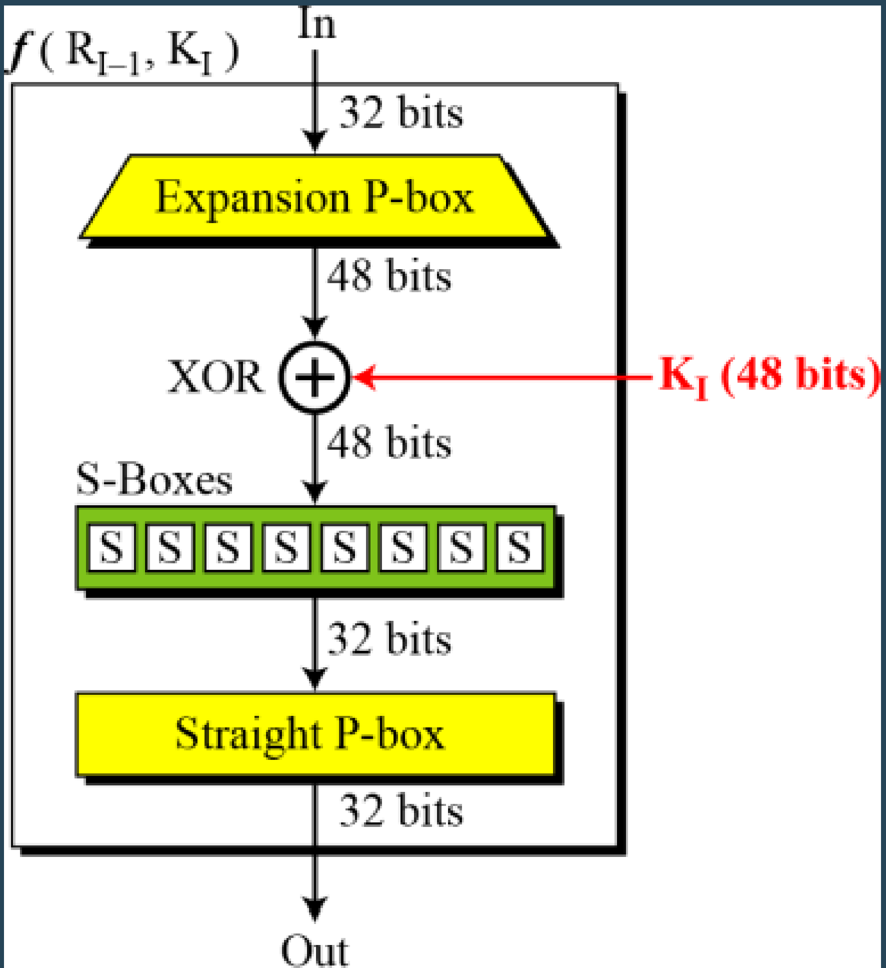
Table 6.2 *Expansion D-box table*

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

Note: The number of output ports is 48, but the value range is only 1 to 32. Some of the inputs go to more than one output.

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 (Substitution 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.

Figure 6.7 *S-boxes*

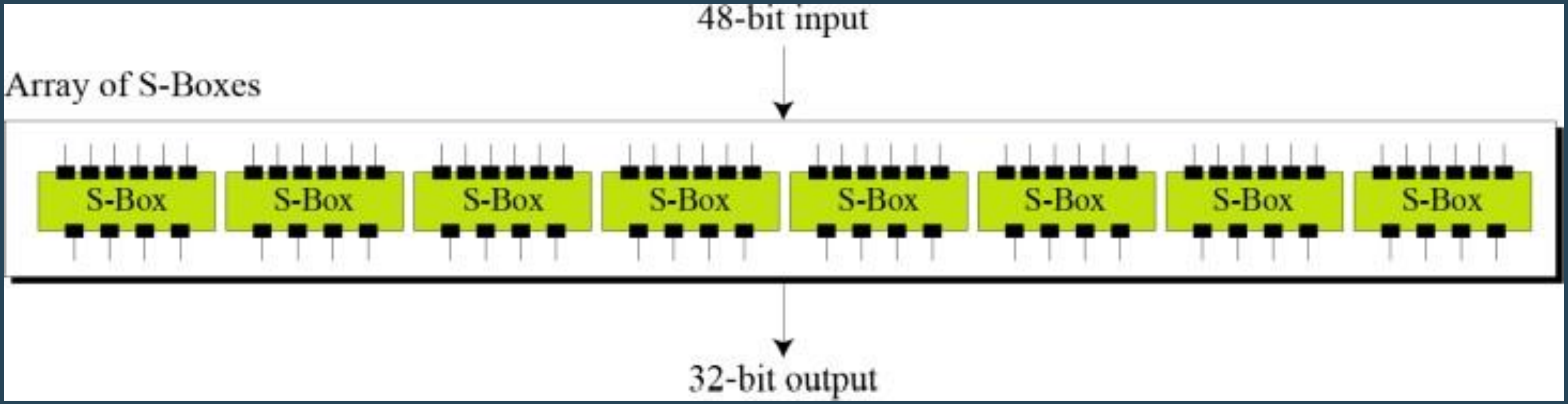


Table 6.3 shows the permutation for S-box 1. Each s-box has its own table

- Each 8-S-boxes has a 6-bit input and a 4-bit output.
- The values of the inputs (row number and column number)
- **Each 6-bit piece uses as an address in the S-boxes where the first and last bits are used to address the ith row and the middle four bits to address the jth column in the S-boxes.**
- **The output of each S-box is 4-bit length piece. The output of all eight S-boxes is then combined into 32 bit section**

Figure 6.8 ***S-box rule***

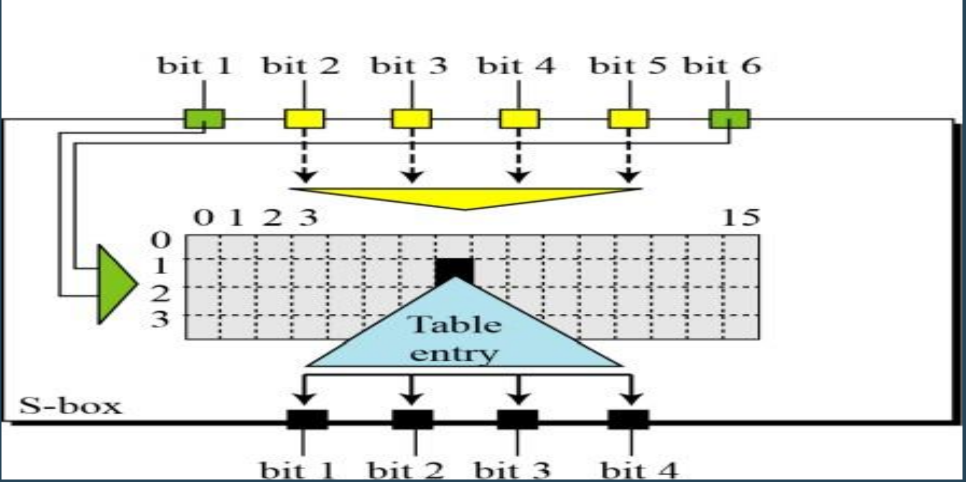
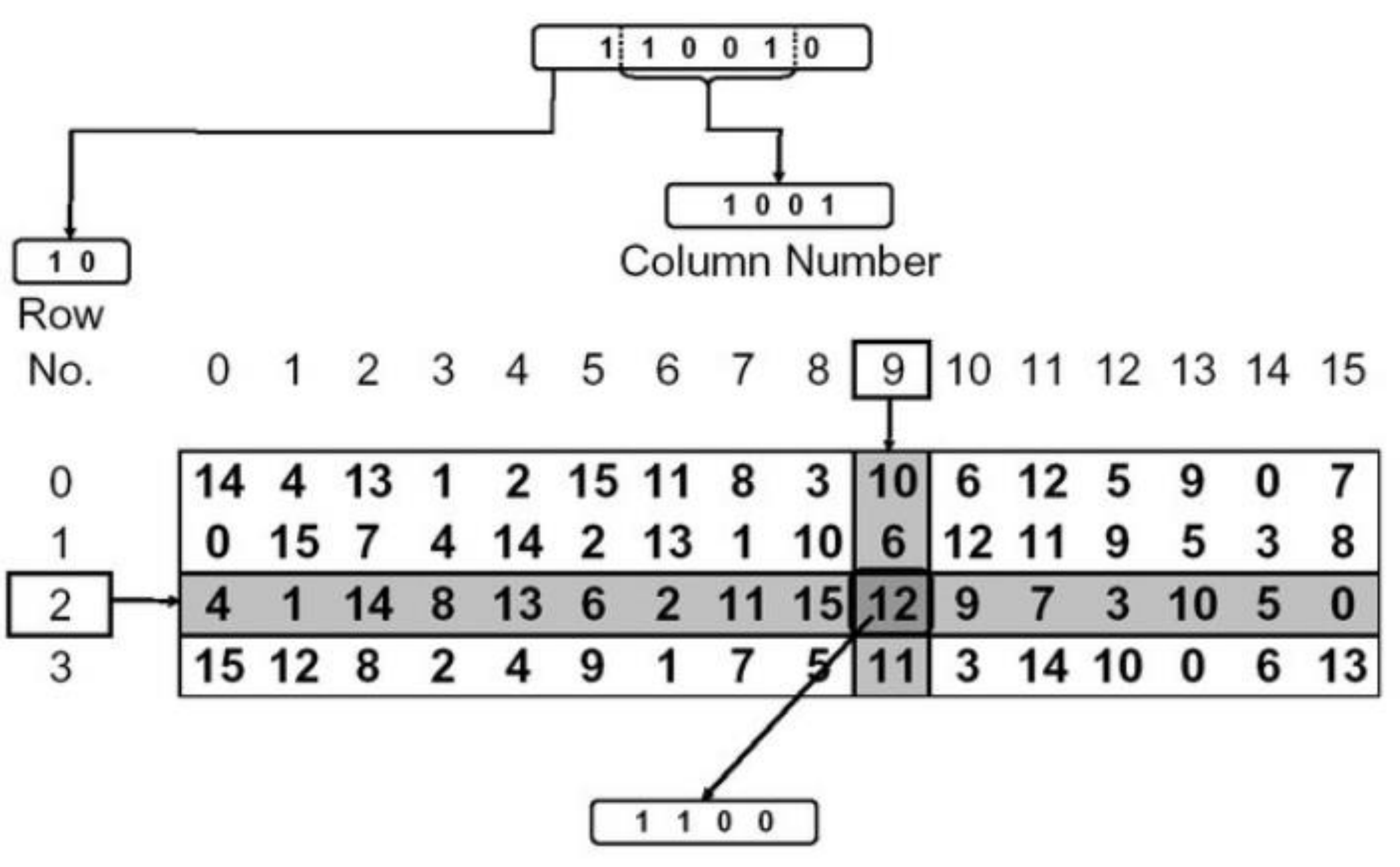


Table 6.3 *S-box 1*

	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



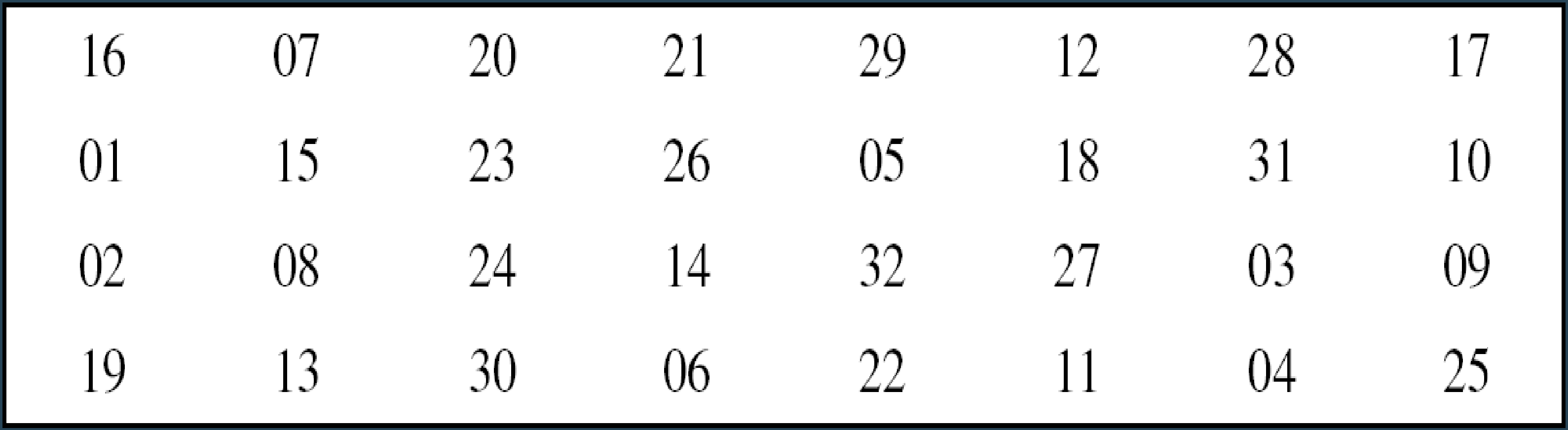
Row No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Straight Permutation

The last operation in the DES function is a permutation with a 32-bit input and a 32-bit output. The input/output relationship for this operation is shown in Table 6.11. Follows the same general rule as previous tables.

For example, the seventh bit of the input becomes the second bit of the output.

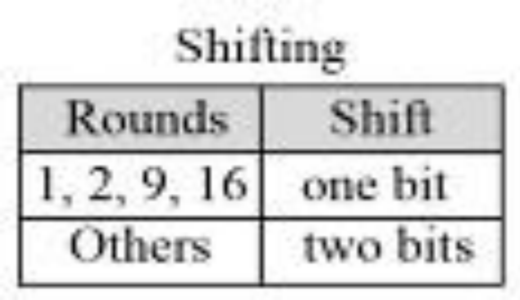
Figure 6.11 ***Straight Permutation table***



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

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

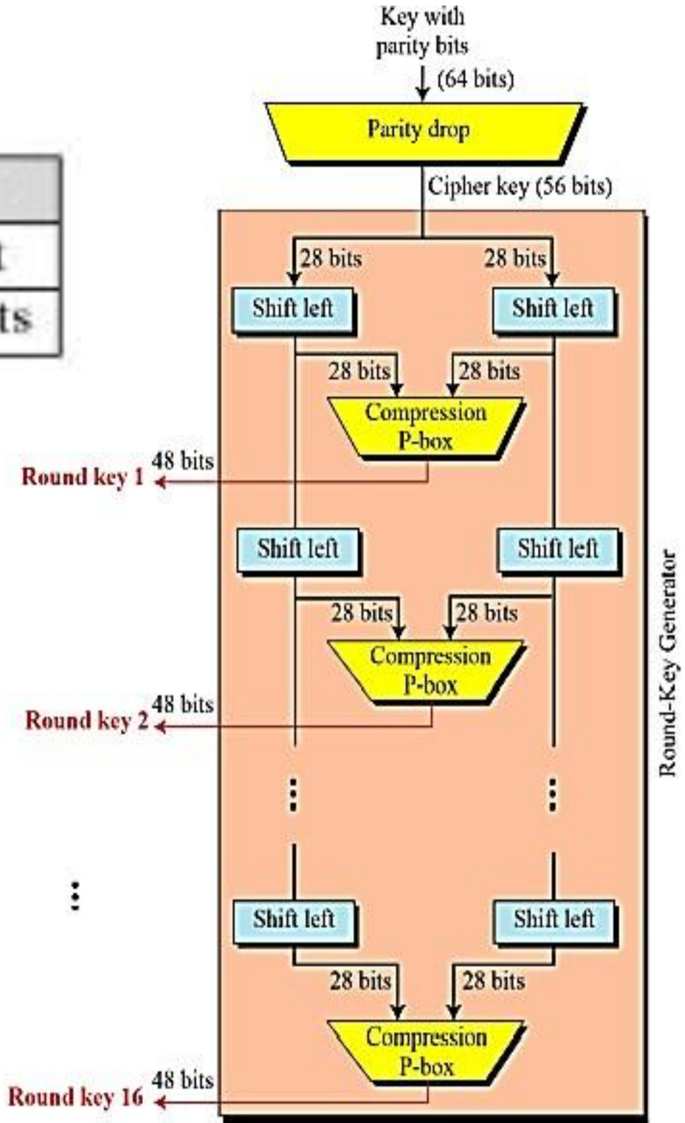
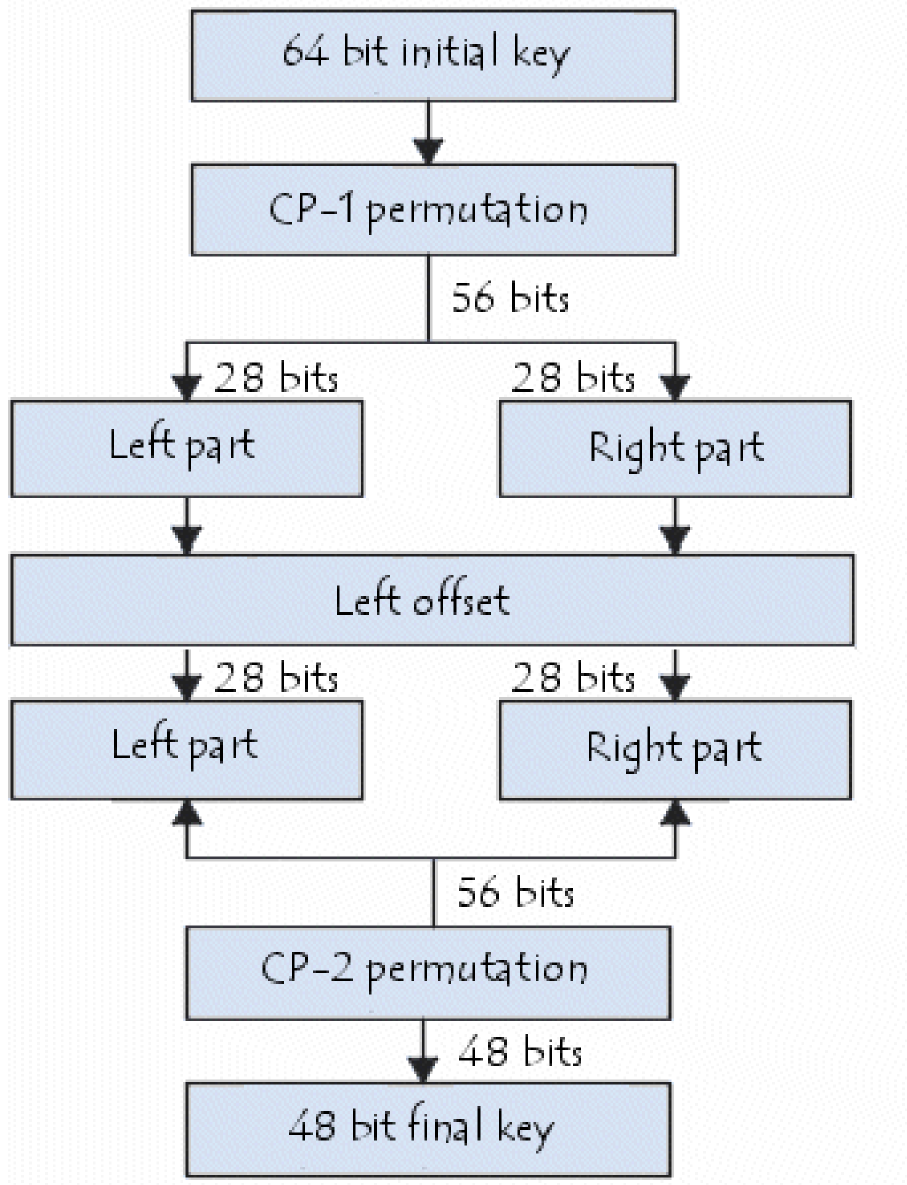


Figure 6.10
Key generation

Key Generation

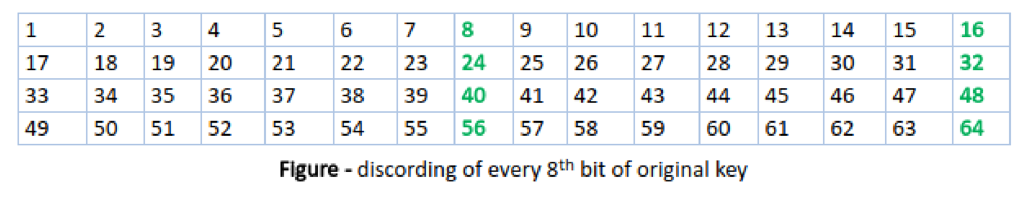


Steps:-

1. It drops the parity bits (bits 8, 16, 24, 32, 40,48,56, & 64) from the 64-bit key and permutes the rest of the bits according to Table 6.12
2. From this 56-bit key, a different 48-bit Sub Key is generated during each round using a process called key transformation
3. For this, the 56-bit key is divided into two halves, each of 28 bits.
4. These halves are circularly shifted left by one or two positions, depending on the round.
5. For example, if the round numbers 1, 2, 9, or 16 the shift is done by **only one position** for other rounds, **the circular shift is done by two positions.**

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



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

Figure - discording of every 8th bit of original key

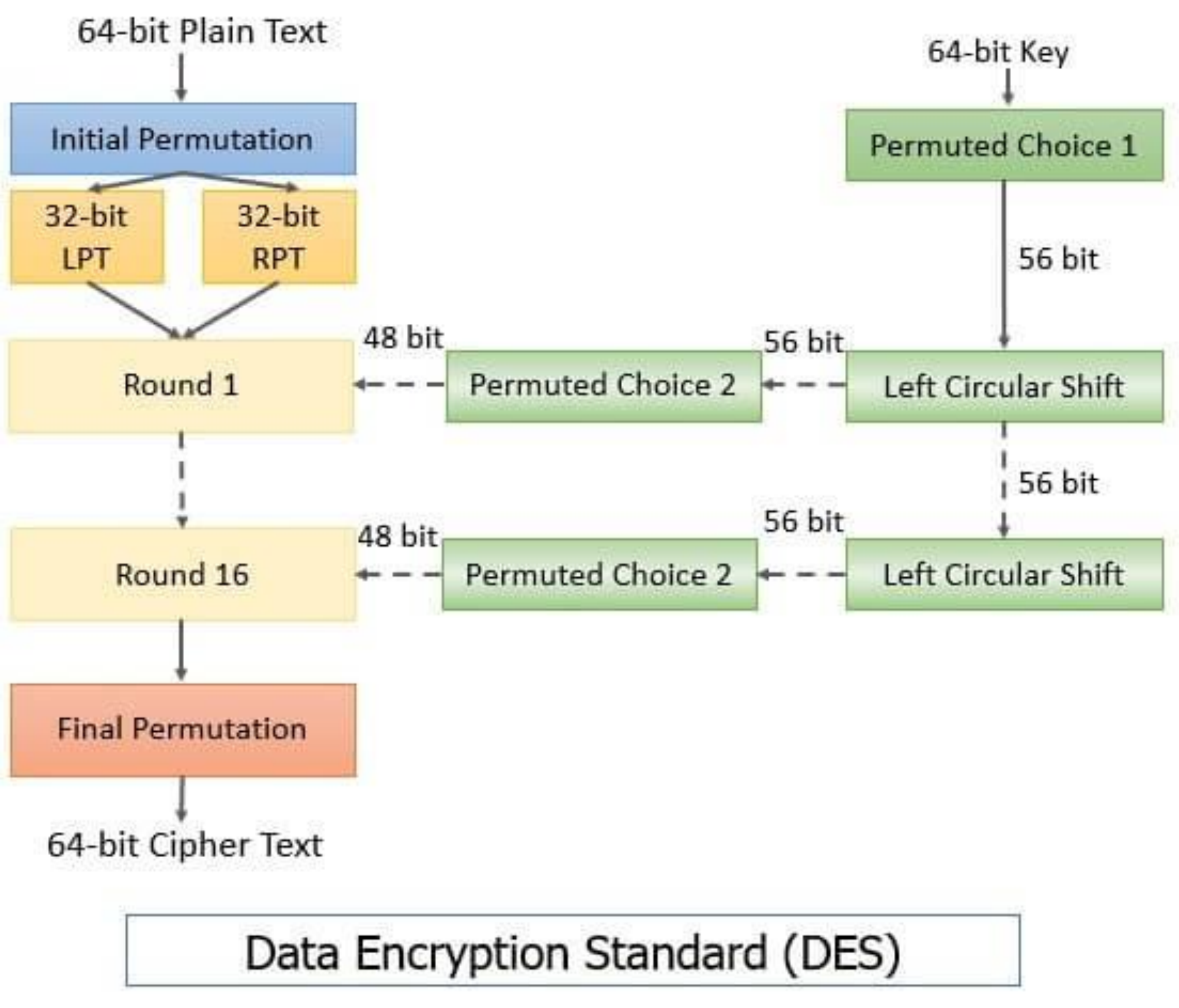
Table 6.13 *Number of bits shifts*

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

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

Complete DES Algorithm



Triple DES (3DES)

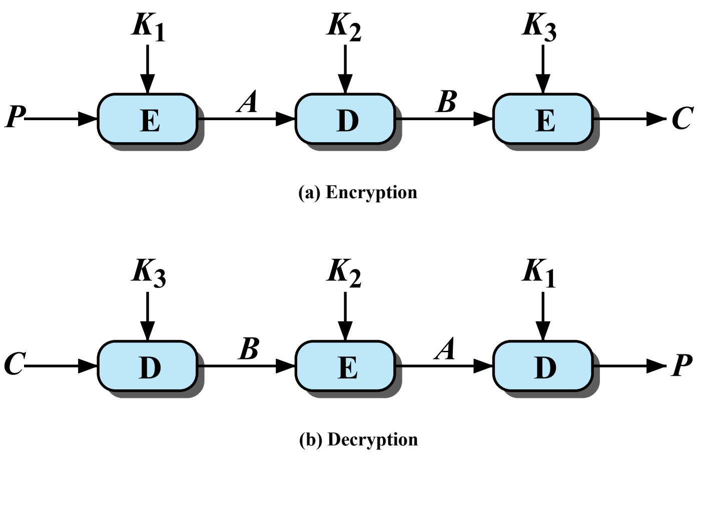
- 3DES uses three keys and three executions of the DES algorithm. The function follows an encrypt-decrypt-encrypt (EDE) sequence

$$C = \mathrm{E}(K_3, \mathrm{D}(K_2, \mathrm{E}(K_1, P)))$$

- where:
  - ○ C = ciphertext;
  - ○ P = plaintext; E[K, X] = encryption of X using key K

- Decryption is simply the same operation with the keys reversed

$$P = \mathrm{D}(K_1, \mathrm{E}(K_2, \mathrm{D}(K_3, C)))$$



(a) Encryption

(b) Decryption

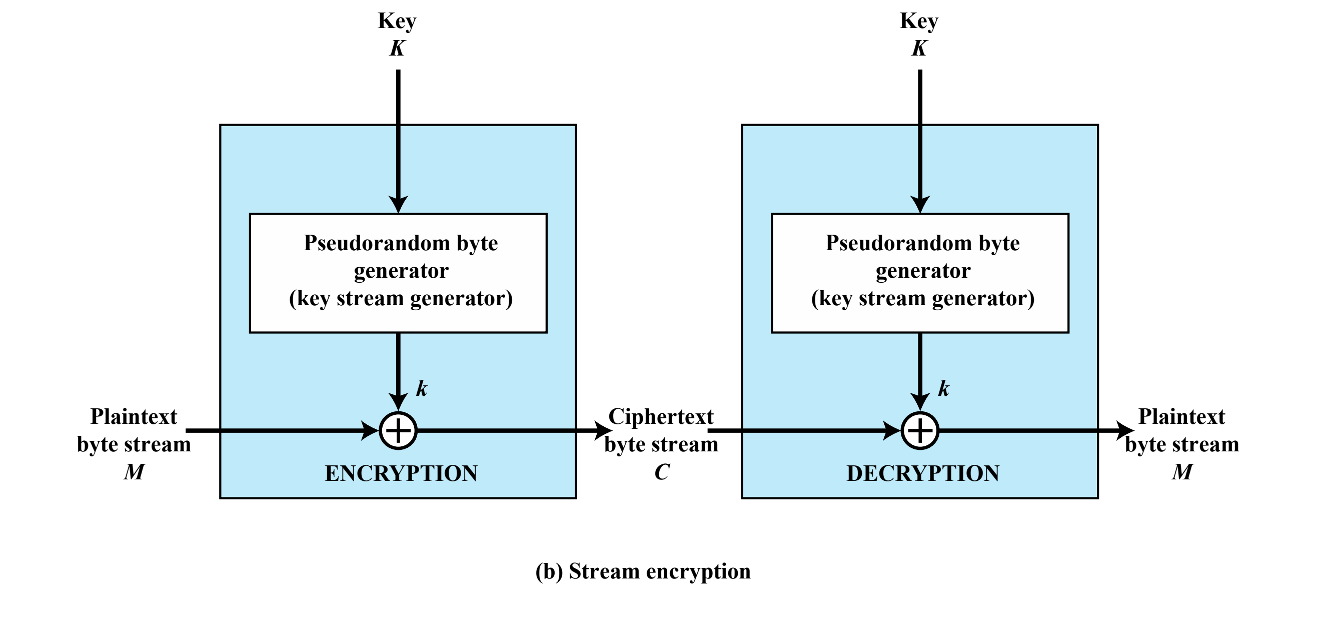
Figure 20.2 Triple DES

Stream Ciphers

- A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along.
  - A block cipher processes the input one block of elements at a time, producing an output block for each input block.
    - Although block ciphers are far more common, there are certain applications in which a stream cipher is more appropriate.
- A typical stream cipher encrypts plaintext one byte at a time, although a stream cipher may be designed to operate on one bit at a time or on units larger than a byte at a time.
  - In this structure, a key is input to a pseudorandom bit generator that produces a stream of 8-bit numbers that are apparently random.
- A pseudorandom stream is one that is unpredictable without knowledge of the input key and which has an apparently random character.
- The output of the generator, called a ***keystream***,
  - is combined one byte at a time with the plaintext stream using the bitwise exclusive OR (XOR) operation.

Stream Ciphers

- For applications that require encryption/decryption of a stream of data, such as
  - over a data communications channel or a browser/Web link, a stream cipher might be the better alternative.
  - For applications that deal with blocks of data, such as file transfer, e-mail, and database, block ciphers may be more appropriate. However, either type of cipher can be used in virtually any application


Key
K
Pseudorandom byte generator (key stream generator)
k
Plaintext byte stream *M*
ENCRYPTION
Ciphertext byte stream *C*
Key
K
Pseudorandom byte generator (key stream generator)
k
DECRYPTION
Plaintext byte stream *M*

(b) Stream encryption

Block & Stream Ciphers

Block Cipher

- Processes the input one block of elements at a time
- Produces an output block for each input block
- Can reuse keys
- More common

Stream Cipher

- Processes the input elements continuously
- Produces output one element at a time
- Primary advantage is that they are almost always faster and use far less code
- Encrypts plaintext one byte at a time
- Pseudorandom stream is one that is unpredictable without knowledge of the input key