Unit II

2

Data Encryption Techniques and Standards

Syllabus

Introduction, Encryption Methods : Symmetric, Asymmetric, Cryptography, Substitution Ciphers, Transposition Ciphers, Steganography applications and limitations, Block Ciphers and methods of operations, Feistal Cipher, Data Encryption Standard (DES), Triple DES, Weak Keys in DES Algorithms, Advance Encryption Standard (AES).

Contents

2.1 Introduction

Important Terms :

1. **Plaintext** : Original message
2. **Ciphertext** : Coded message
3. **Enciphering or encryption** : The process of converting from plaintext to ciphertext
4. **Deciphering or decryption** : The process of restoring the plaintext from the ciphertext

- Many schemes used for encryption constitute the area of study known as **cryptography**. Such a scheme is known as a **cryptographic system** (cryptosystem) or a **cipher.**
- Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of **cryptanalysis.**
- Cryptanalysis is what the layperson calls "breaking the code". The areas of cryptography and cryptanalysis together are called **cryptology.**

2.2 Encryption Methods

2.2.1 Symmetric Encryption

- A symmetric encryption model has five ingredients.
  1. Plaintext
  2. Encryption algorithm
  3. Secret key
  4. Ciphertext
  5. Decryption algorithm
- Fig. 2.2.1 shows the conventional encryption model.
- **Plaintext** is the original message or data that is fed into the algorithm as input.
- **Encryption algorithm** performs various substitutions and transformations on the plaintext.
- **Secret key** is a value independent of the plaintext and of the algorithm. The exact substitutions and transformations performed by the algorithm depend on the key.
- **Ciphertext** is the scrambled message produced as output. It depends on the plaintext and the secret key.
- **Decryption algorithm** takes the ciphertext and the secret key and produces the original plaintext.
- The original intelligible message, referred to as plaintext is converted into random nonsense, referred to as ciphertext. The science and art of manipulating messages to make them secure is called **cryptography**
- An original message to be transformed is called the plaintext, and the resulting message after the transformation is called the ciphertext.
- The process of converting the plaintext into ciphertext is called encryption. The reverse process is called decryption. The encryption process consists of an algorithm and a key. The key controls the algorithm.
- The objective is to design an encryption technique so that it would be very difficult or impossible for an unauthorized party to understand the contents of the ciphertext.
- A user can recover the original message only by decrypting the ciphertext using the secret key. Depending upon the secret key used, the algorithm will produce a different output. If the secret key changes, the output of the algorithm also changes.

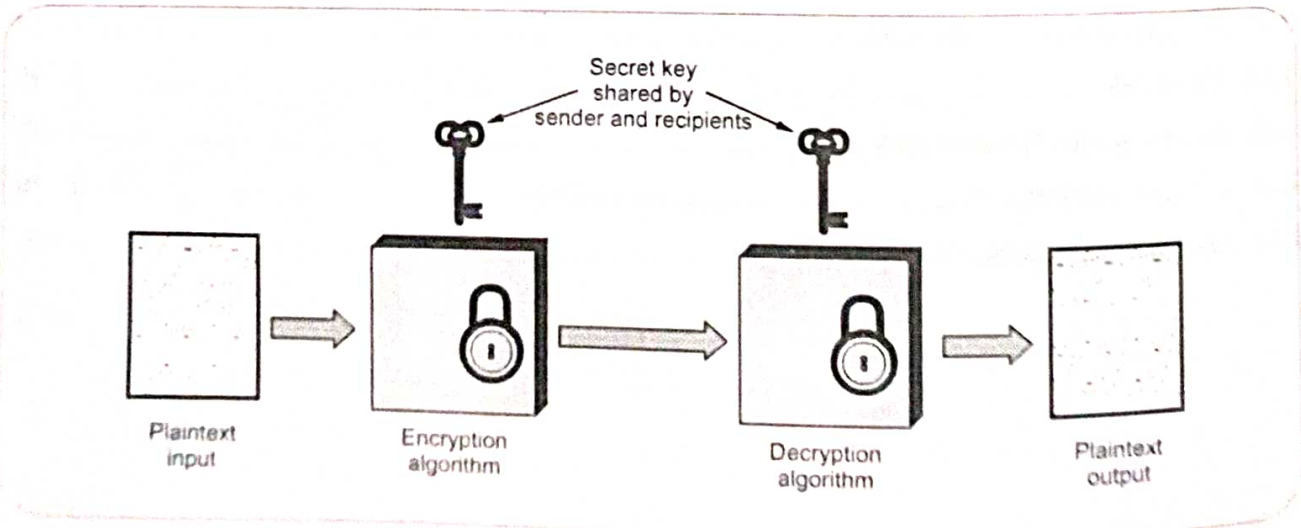

Fig. 2.2.1 Conventional encryption model

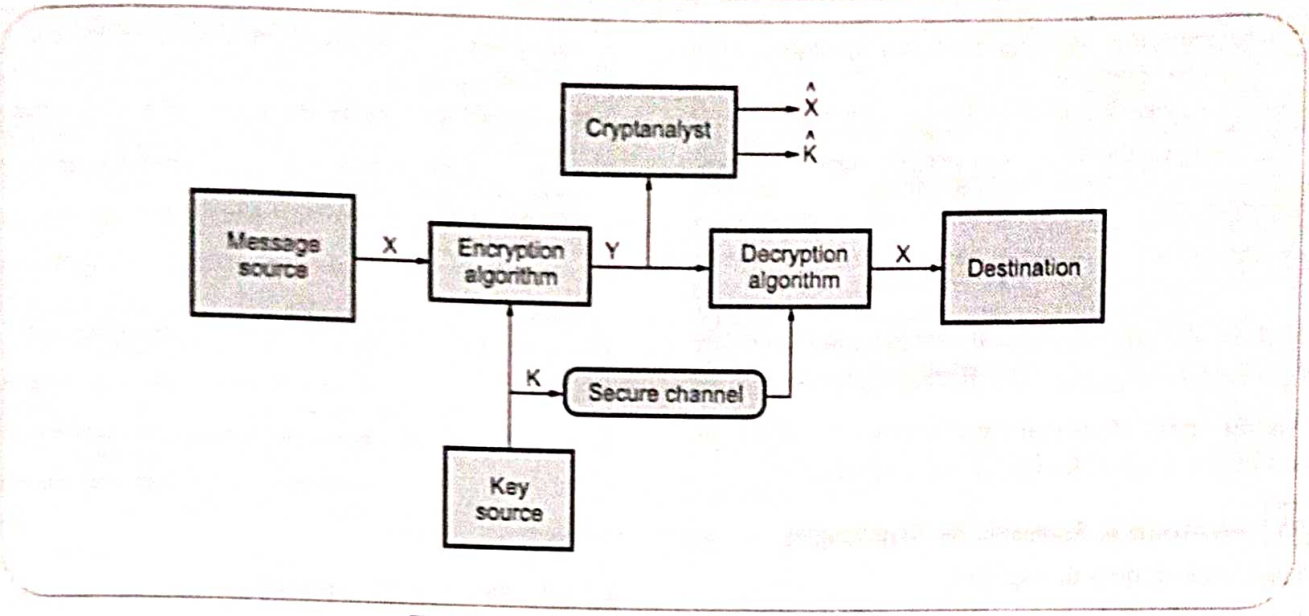

Fig. 2.2.2 Model of conventional cryptosystem

- Fig. 2.2.2 shows the conventional encryption process.
- The security of the conventional encryption depends on the several factors. The encryption algorithm must be powerful. Decryption message must be difficult. The algorithm depend on the secrecy of the key only. The algorithm is upon to all but only key is to keep secret. As shown in the diagram, the message source is the plaintext. i.e. X with the message X and encryption key K as input and ciphertext Y, we can write this as,

$$Y = E(K, X) \quad \ldots(2.2.1)$$

- Using equation (2.2.1) Y is to be produced by using encryption algorithm E as a function of the plaintext X. The intended receiver in possession of the key, is able to invert the transformation.

$$X = D(K, Y) \quad \ldots(2.2.2)$$

- An opponent, observing Y but not having access to K or X, must attempt to recover X and K or both X and K. It is assumed that the opponent does have knowledge of the encryption (E) and decryption (D) algorithms.

Characteristics of cryptography

1. The type of operations used for transforming plaintext to ciphertext.
2. The number of keys used.
3. The way in which the plaintext is processed.

Cryptanalysis

- The process of trying to break any cipher text message to obtain the original plain text message itself is called as **cryptanalysis**.
- Cryptanalysis is the breaking of codes. The person attempting a cryptanalysis is called as a **cryptanalyst**.
- **Brute force attack :** The attacker tries every possible key on a piece of cipher text until an intelligible translation into plaintext is obtained.

Types of attacks on encrypted messages

Sr. No.	Type of attack	Known to cryptanalyst
1.	Ciphertext only	1. Encryption algorithm 2. Ciphertext
2.	Known plaintext	1. Encryption algorithm 2. Ciphertext 3. One or more plaintext ciphertext pairs formed with the secret key.
3.	Chosen plaintext	1. Encryption algorithm 2. Ciphertext 3. Plaintext message chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key.
4.	Chosen ciphertext	1. Encryption algorithm 2. Ciphertext 3. Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key.

Breakable encryption

- An encryption algorithm may be breakable, meaning that given enough time and data, an analyst could determine the algorithm.
- Practicality is an issue
  1. For a given cipher scheme, there may be 10^{30} possible decipherments, so the task is to select the right one out of the 10^{30}.
  2. Cryptanalyst cannot be expected to try just the hard, long way but within this time another efficient algorithm may exist.
  3. Estimates of breakability are based on current technology so it is budget dependent.

2.4 Substitution Ciphers

- A substitution cipher changes characters in the plaintext to produce to ciphertext. A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols.
- If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns.

2.4.1 Caesar Cipher

- Caesar cipher is a special case of substitution techniques wherein each alphabet in a message is replaced by an alphabet three places down the line.
- Caesar cipher is susceptible to a statistical ciphertext only attack.
- For example,

Plaintext	hellow world
Ciphertext	KHOOR ZRUOG

- List of all possible combination of letters.

Plain	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
Cipher	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V

Plain	t	u	v	w	x	y	z
Cipher	W	X	Y	Z	A	B	C

- Numerical equivalent to each letter is given below.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	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

- The algorithm can be expressed as follows. For each plaintext letter P, substitute the ciphertext letter C :

$$C = E(3, P) = (P + 3) \bmod 26$$

- A shift may be of any amount, so that the general Caesar algorithm is

$$C = E(K, P) = (P + K) \bmod 26$$

where K = Values from 1 to 25

- The decryption algorithm is simply

$$P = D(K, C) = (C - K) \bmod 26$$

- If it is known that a given ciphertext is a Caesar cipher, then a brute force cryptanalysis is easily performed : Simply try all the 25 possible keys.

- **Demerits :**
  1. The encryption and decryption algorithms are known.
  2. There are only 25 keys to try.
  3. The language of the plaintext is known and easily recognizable.

2.4.2 Monoalphabetic Cipher

- Monoalphabetic cipher substitutes one letter of the alphabet with another letter of the alphabet. However, rather than substituting according to a regular pattern, any letter can be substituted for any other letter, as long as each letter has a unique substitute left and vice versa.

Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m
Ciphertext	m	n	b	v	c	x	z	a	s	d	f	g	h

Plaintext	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext	j	k	l	p	o	i	u	y	t	r	e	w	q

For example

Plaintext message : hello how are you

Ciphertext message : acggk akr moc wky

- Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet.

Homophonic substitution cipher

- It provides multiple substitutes for a single letter. For example, A can be replaced by D, H, P, R; B can be replaced by E, Q, S, T etc.

2.4.3 Playfair Cipher

- The playfair algorithm is based on the use of a 5 × 5 matrix of letters constructed using a keyword.
- **For example :** Monarchy is the keyword.

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

- The matrix is constructed by filling in the letters of the keyword from left to right and from top to bottom and then filling in the remainder of the matrix with the remaining letters in alphabetic order.
- The letters I and J count as one letter.

2.4.4 Hill Cipher

- The encryption algorithm takes m successive plaintext letters and substitutor for them m ciphertext letters.
- The substitution is determined by m linear equations in which each character is assigned a numerical value (a = 0, b = 1, c = 2, ... z = 25), the system can be described as follows :

$$C_1 = (K_{11} P_1 + K_{12} P_2 + K_{13} P_3) \bmod 26$$

$$C_2 = (K_{21} P_1 + K_{22} P_2 + K_{23} P_3) \bmod 26$$

$$C_3 = (K_{31} P_1 + K_{32} P_2 + K_{33} P_3) \bmod 26$$

- This can be expressed in term of column vectors and matrices :

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} \bmod 26$$

or $C = KP \bmod 26$

Where C and P are column vectors of length 3, representing the plaintext and ciphertext.

- K is a 3 × 3 matrix, representing the encrypting key.
- **For example :**

Plaintext = Paymoremoney

$$\text{Key (K)} = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

The first three letters of the plaintext are represented by the vector.

$$C = KP \bmod 26$$

$$= \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix} \bmod 26$$

$$= \begin{pmatrix} 375 \\ 819 \\ 486 \end{pmatrix} \bmod 26 = \begin{pmatrix} 11 \\ 13 \\ 18 \end{pmatrix} = \text{LNS}$$

For plaintext pay, ciphertext is LNS.

The entire ciphertext is **LNSHDLEWMTRW**

- Decryption requires using the inverse of the matrix K.
- The general terms in Hill cipher is

Cipher $C = E(K, P) = KP \bmod 26$

Plaintext $P = D(K, P) = K^{-1} C \bmod 26 = K^{-1} KP = P$

Advantages

1. It completely hides single letter frequency.
2. Hill cipher is strong against a ciphertext only attack.
3. By using larger matrix, more frequency information hiding is possible.

Disadvantage

1. Easily broken with a known plaintext attack.

2.4.5 Polyalphabetic Substitution

- In polyalphabetic substitution, each occurrence of a character can have a different substitute. The relationship between a character in the plaintext to a character in the ciphertext is one to many.
- An example of polyalphabetic substitution is the **Vigenere cipher.**
- The Vigenere cipher chooses a sequence of keys, represented by a string. The key letters are applied to successive plaintext characters, and when the end of the key is reached, the key start over.
- Fig. 2.4.1 shows a tableall or table to implement this cipher efficiently,

		Plaintext																								
		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Key	a	A	B	C	D	E	F	G	H	I	J	K	I	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
	c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
	d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
	e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
	f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
	g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
	h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
	i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
	j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
	k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
	l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
	m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
	n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
	o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	N	N	O	P	Q
	s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
	x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
	y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
	z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Fig. 2.4.1

- **For example :** Let the message be THE BOY HAS THE BAG and let the key be VIG.

Key = VIG VIG VIG VIG VIG

Plaintext = THE BOY HAS THE BAG

Ciphertext = OPKWWECIYOPKWIM

- The strength of this cipher is that there are multiple ciphertext letters for each plaintext letter, one for each unique letter of the keyword.

2.4.6 One Time Pad

- The key string is chosen at random and at least as long as the message, so it does not repeat.
- Each new message requires a new key of the same length as the new message. It produces random output that bears no statistical relationship to the plaintext.
- **Vernam cipher** uses a one time pad, which is discarded after a single use, and therefore is suitable only for short messages.
- **For example :**

Plaintext :	c	o	m	e	t	o	d	a	y
	2	14	12	4	19	14	3	0	24
Key	N	C	B	T	Z	Q	A	R	X
	13	2	1	19	25	16	0	17	23
Total	15	16	13	23	44	30	3	17	47
Subtract 26 if > 25	15	16	13	23	18	04	3	17	21
Ciphertext	P	Q	N	X	S	E	D	R	V

- The one time pad offers complete security but, in practice, has two fundamental difficulties.
  1. There is the practical problem of making large quantities of random keys.
  2. Key distribution and protection is also major problem with one time pad.
  3. Only possible attack to such a cipher is a brute force attack.

2.4.7 Feistel Cipher

- Fig. 2.4.2 shows the classical Feistel network. 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 i.e. Left (L_0) and Right (R_0).

(See Fig. 2.4.2 on next page)

Parameters and design features

Following parameters are considered :

1. Block size
2. Key size
3. Number of rounds
4. Subkey generation algorithms
5. Round function
6. Fast software encryption / decryption.
7. Ease of analysis

1. Security depends upon the block size. Larger **block size** gives greater security but encryption / decryption speed is reduced normal. Block size is 64-bit and AES uses 128-bit block size.
2. Greater security is achieved by using longer **key size**. Because of longer key size, again speed of algorithm decreases. Key sizes of 64 bits or less are now widely considered to be inadequate and 128 bits have become a common size.
3. **Number of rounds** are 16 in most of the algorithm. In Feistel cipher, single round offers insufficient security and multiple rounds offer greater security.
4. In **subkey generation algorithm**, greater complexity leads to greater difficulty of cryptanalysis.
5. **Round function** is again greater complexity for greater resistance to cryptanalysis.
6. **Fast software encryption / decryption** : The speed of execution of the algorithm becomes a concern.
7. **Ease of analysis** : There is great benefit in making the algorithm easy to analysis.

Decryption algorithm

- Use the ciphertext as input to the algorithm, but use the subkeys K_i in reverse order.
- The output of the first round of the decryption process is equal to a 32 bit swap of the input to the 16^{th} round of the encryption process.
- Consider the encryption process :

$$LE_{16} = RE_{15}$$

$$RE_{16} = LE_{15} \times F(RE_{15}, K_{16})$$

- On the decryption side

$$LD_1 = RD_0 = LE_{16} = RE_{15}$$

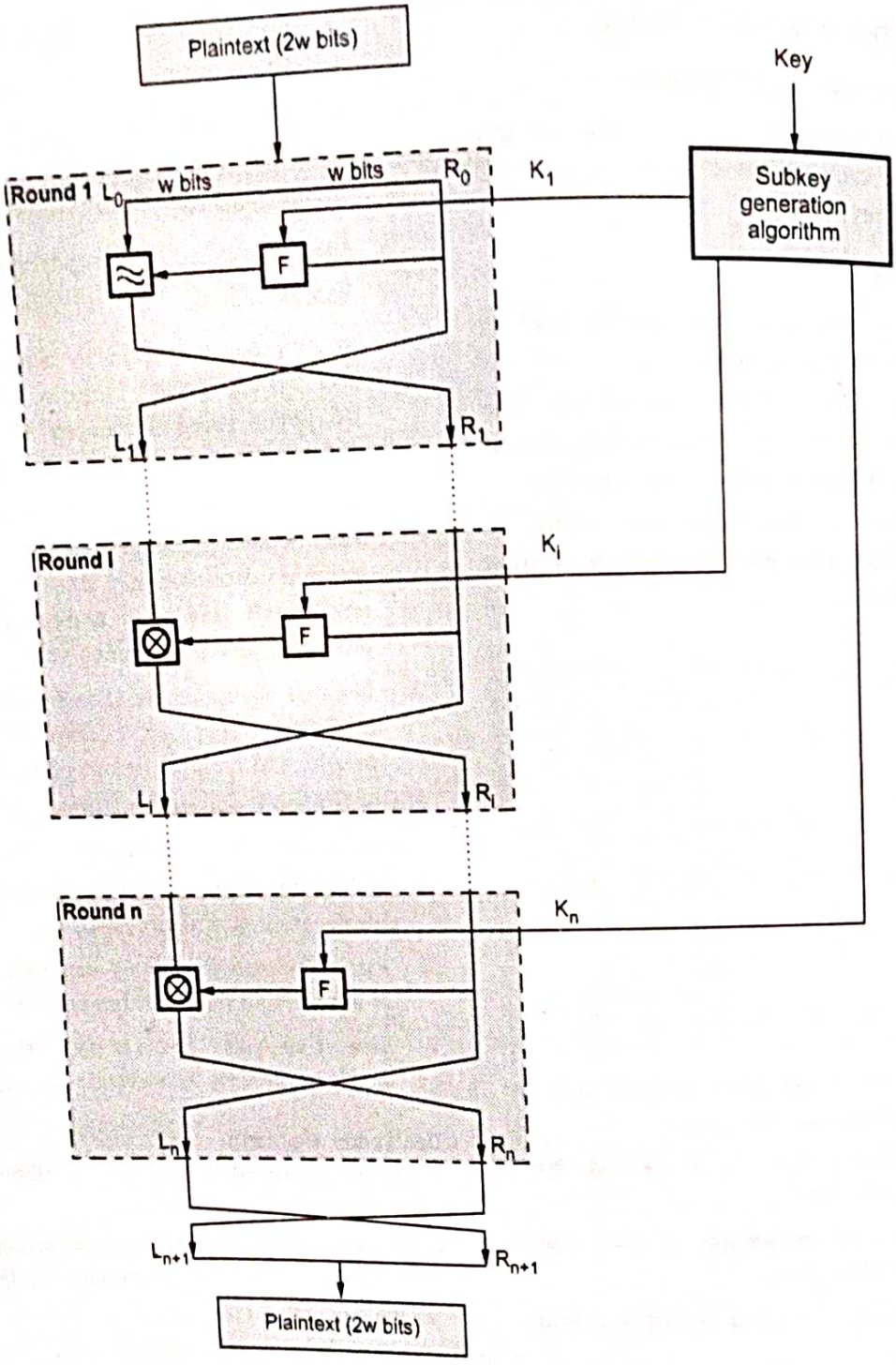

Fig. 2.4.2 Classical feistel network

$$RD_1 = LD_0 \times F(RD_0, K_{16})$$

$$= RE_{16} \times F(RE_{15}, K_{16})$$

$$= [(LE_{15} \times F(RE_{15}, K_{16})] \times F(RE_{15}, K_{16})$$

$\therefore$ We have $LD_1 = RE_{15}$ and $RD_1 = LE_{15}$

- For the i^{th} iteration of the encryption algorithm,

$$LE_i = RE_{i-1}$$

$$RE_i = LE_{i-1} \times F(RE_{i-1}, K_i)$$

Finally, the output of the last round of the decryption process is $RE_0 \| LE_0$. A 32 bit swap recovers the original plaintext, demonstrating the validity of the Feistel decryption process.

2.4.8 Comparison between Monoalphabetic and Polyalphabetic Cipher

Sr. No.	Monoalphabetic cipher	Polyalphabetic cipher
1.	Once a key is chosen, each alphabetic character of a plaintext is mapped onto a **unique** alphabetic character of a ciphertext.	Each alphabetic character of a plaintext can be mapped onto "m" alphabetic characters of a ciphertext.
2.	The relationship between a character in the plaintext and the characters in the ciphertext is one-to-one.	The relationship between a character in the plaintext and the characters in the ciphertext is one-to-many.
3.	A stream cipher is a monoalphabetic cipher if the value of k_i does not depend on the position of the plaintext character in the plaintext stream	A stream cipher is a polyalphabetic cipher if the value of k_i does depend on the position of the plaintext character in the plaintext stream.
4.	Monoalphabetic cipher includes additive, multiplicative, affine and monoalphabetic substitution cipher.	Polyalphabetic cipher includes autokey, Playfair, Vigenere, Hill, one-time pad, rotor , and Enigma cipher.

Ex. 2.4.1 *Encrypt the message "PAY" using Hill cipher with the following key matrix and show the decryption to get the original plain text.*

$$K = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

Sol. : $$K = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

The letters PAY of the plaintext are represented by the vector :

$$\begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix} \begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix} = \begin{pmatrix} 375 \\ 819 \\ 486 \end{pmatrix} \bmod 26 = \begin{pmatrix} 11 \\ 13 \\ 18 \end{pmatrix} = \text{LNS}$$

Ciphertext = LNS

Ex. 2.4.2 : *Use play fair cipher to encrypt the following message "This is a columnar transposition" use key - APPLE.*

Sol. : Message = This is a columnar transposition

Key = APPLE

Encryption

A	P	L	E	B
C	D	F	G	H
I/J	K	M	N	O
Q	R	S	T	U
V	W	X	Y	Z

Message = Th is is ac ol um na rt ra ns po si ti on

Ciphertext = UG MQ MQ BH MB SO IE SU MT BK QM NQ KN

Ex. 2.4.3 : *Using hill cipher encrypt plain text "COE" use key "ANOTHERBZ".*

Sol. : Plain text = COE

Key = ANOTHERBZ

For plaintext COE, here C = 2 O = 14 E = 4

Therefore $P = \begin{pmatrix} 2 \\ 14 \\ 4 \end{pmatrix}$

For key ANOTHERBZ the numbers are 0, 13, 14, 19, 6, 4, 17, 1, 25

The numbers in the matrix form :

$$K = \begin{pmatrix} 0 & 13 & 14 \\ 19 & 6 & 4 \\ 17 & 1 & 25 \end{pmatrix}$$

Ciphertext = (Key X Plaintext) Mod 26

Encryption is as follows :

$$C = \begin{pmatrix} 0 & 13 & 14 \\ 19 & 6 & 4 \\ 17 & 1 & 25 \end{pmatrix} \begin{pmatrix} 2 \\ 14 \\ 4 \end{pmatrix} \text{mod} 26$$

$$= \begin{pmatrix} 238 \\ 138 \\ 148 \end{pmatrix} \text{mod} 26 = \begin{pmatrix} 4 \\ 8 \\ 18 \end{pmatrix}$$

Ciphertext = 4 = E , 8 = I and 18 = S

Ciphertext = EIS

Ex. 2.4.4 : *Use polyalphabetic ciphers to encrypt plain text "SHE IS VERY HAPPY AND BEAUTIFUL GIRL" use key 'ANOTHER'.*

Ans. :

Keyword	anoth	erano	there	nothe	raner	here
Plaintext	shein	svery	hppys	andbe	autif	ulgir
Ciphertext	SUNBZ	ZVRLV	TWTPA	ABULE	LTVTN	SKZR

Review Questions

1. *Use play fair cipher to encrypt the following message "This is a columnar transposition" use key - APPLE.*
2. *Using hill cipher encrypt plain text "COE" use key "ANOTHERBZ".*
3. *Explain feistel cipher in detail.*
4. *Using Playfair cipher encrypt message "We live in a world full of beauty" use key "ANOTHER".*
5. *Use poly alphabetic ciphers to encrypt plain text "SHE IS VERY HAPPY AND BEAUTIFUL GIRL" use key 'ANOTHER'.*
6. *Explain the operation of polyalphabetic cipher.*
7. *Encrypt the plain text 'COE' using hill cipher, use keyword 'ANOTHERBZ'.*

2.5 Transposition Ciphers

- A transposition cipher rearranges the characters in the plaintext to form the ciphertext. The letters are not changed.
- The rail fence cipher is composed by writing the plaintext in two rows, proceeding down, then across and reading the ciphertext across, then down.
- For example, to encipher the message "meet me after this party" with a rail fence of depth 2, we write the following :

m	e	m	a	t	r	h	s	a	t	
e	t	e	f	e	t	i	p	r	y	

- The ciphertext is

MEMATRHSATETEFETIPRY

- Attacking a transposition cipher requires rearrangement of the letters of the ciphertext.
- A pure transposition cipher is easily recognized because it has the same letter frequencies as the original plaintext.

Plaintext : The book is suitable for self study.

Key : 5 6 4 1 3 2

Key :	5	6	4	1	3	2
Plaintext :	t	h	e	b	o	o
	k	i	s	s	u	i
	t	a	b	l	e	f
	o	r	s	e	l	f
	s	t	u	d	y	

Ciphertext : BSLEDOIFFOUELYESBSUTKTOSHIART.

2.5.1 Comparison of Substitution and Transposition Ciphers

	Substitution ciphers	Transposition ciphers
Definition	Each letter or group of letters of the plaintext are replaced by some other letter of group of letters, to obtain the ciphertext.	Letters of the plaintext are permuted in some form.
Example	Hill cipher, one time pad	Rail fence cipher
Strength	1.Exhaustive search is infeasible. 2.Through to be unbreakable by many back then.	1.Reduce redundancies in plaintext. 2.Transposition cipher can be made more secure by performing more than one stage of transposition.
Drawback	1.Brute force attack is easy	1.The ciphertext has the same letter frequency as the original plaintext. 2.Guessing the number of columns and some probable words in the plaintext holds the key.

Ex. 2.5.1 : *Use the transposition cipher to encrypt the plain text "WE ARE THE BEST" use the key "HEAVEN".*

Sol. :

Key	→	H	E	A	V	E	N
Key number	→	4	2	1	6	3	5
Plaintext	→	W	E	A	R	E	T
		H	E	B	E	S	T

Arrange the key number as per ascending order.

Key	→	A	E	E	H	N	V
Key number	→	1	2	3	4	5	6
Plaintext	→	A	E	E	W	T	R
		B	E	S	H	T	E

Ciphertext = ABEEESWHTTRE

Review Question

1. *What is transposition cipher ? Use transposition cipher to encrypt the plain text "WE ARE THE BEST" use key "HEAVEN".*

2.6 Steganography Applications and Limitations

- Steganography is derived from the Greek for covered writing and essentially means "to hide in plain sight".
- As defined as it is the art and science of communicating in such a way that the presence of a message cannot be detected. Simple steganographic techniques have been in use for hundreds of years, but with the increasing use of files in an electronic format new techniques for information hiding have become possible.
- Fig. 2.6.1 shows how information hiding can be broken down into different areas. Steganography can be used to hide a message intended for later retrieval by a specific individual or group.
- The other major area of steganography is copyright marking, where the message to be inserted is used to assert copyright over a document. This can be further divided into watermarking and fingerprinting.
- Steganography and encryption are both used to ensure data confidentiality. However, the main difference between them is that with encryption anybody can see that both parties are communicating in secret.
- Steganography hides the existence of a secret message and in the best case nobody can see that both parties

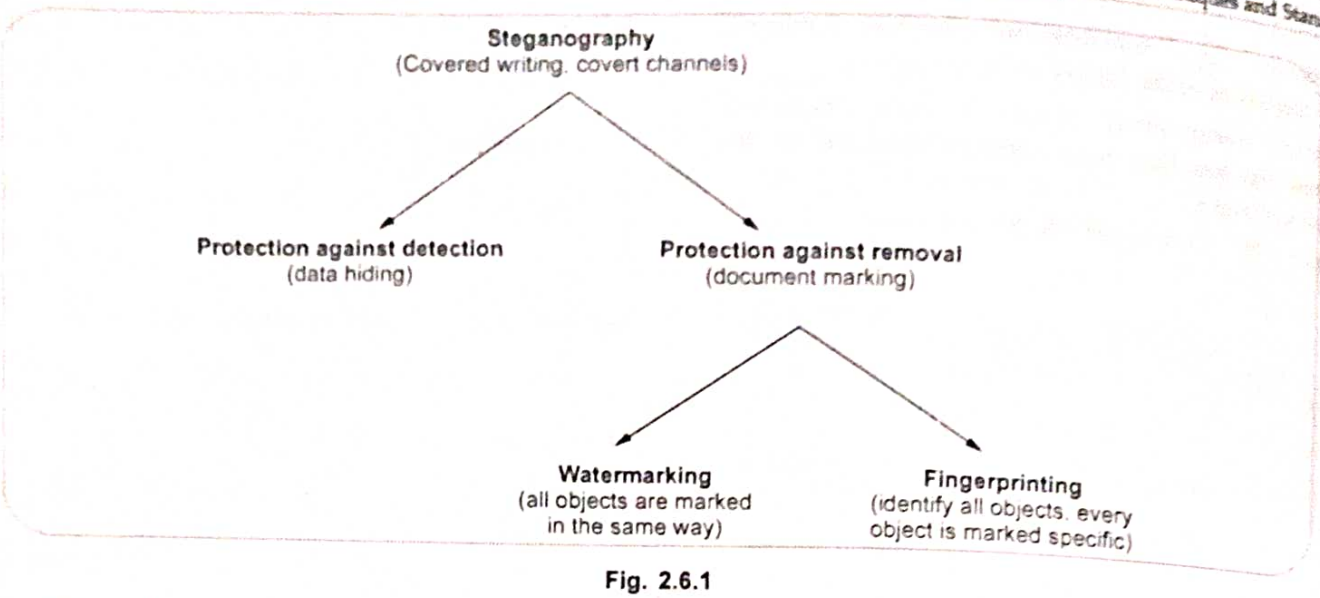

Fig. 2.6.1

are communicating in secret. This makes staganography suitable for some tasks for which encryption aren't, such as copyright marking.

- Adding encrypted copyright information to a file could be easy to remove but embedding it within the contents of the file itself can prevent it being easily identified and removed.
- Steganography provides a means of secret communication which cannot be removed without significantly altering the data in which it is embedded. The embedded data will be confidential unless an attacker can find a way to detect it.
- The following is a list of main requirements that staganography techniques must satisfy :
  1. The integrity of the hidden information after it has been embedded inside the stego object must be correct.
  2. The stego object must remain unchanged or almost unchanged to the naked eye. If the stego object changes significantly and can be noticed, a third party may see that information is being hidden and therefore could attempt to extract or to destroy it.
  3. In watermarking, changes in the stego object must have no effect on the watermark.
  4. Finally, we always assume that the attacker knows that there is hidden information inside the stego object.
- Fig. 2.6.2 shows a simple process in steganography. In this example, a secret image is being embedded inside a cover image to produce the stego image.
- The first step in embedding and hiding information is to pass both the secret messge and the cover message into the encoder. Inside the encoder, one or several protocols will be implemented to embed the secret information into the cover message. The type of protocol will depend on what information you are trying to embed and what you are embedding it in.
- A key is often needed in the embedding process. This can be in the form of a public or private key so you can encode the secret message with your private key and the recipient can decode it using your public key.
- In embedding the information this way, you can reduce the chance of a third party attacker getting hold of the stego object and decoding it to find out the secret information.
- In general the embedding process inserts a mark, M, in an object, I. A key, K, usually produced by a random number generator is used in the embedding process and the resulting marked object, $\tilde{I}$, is generated by the mapping : $I \times K \times M \rightarrow \tilde{I}$.
- Having passed through the encoder, a stego object will be produced. A stego object is the original cover object with the secret information embedded inside. This object should look almost identical to the cover object as otherwise a third party attacker can see embedded information.
- Having produced the stego object, it will then be sent off via some communications channel, such as email to the intended recipient for decoding.

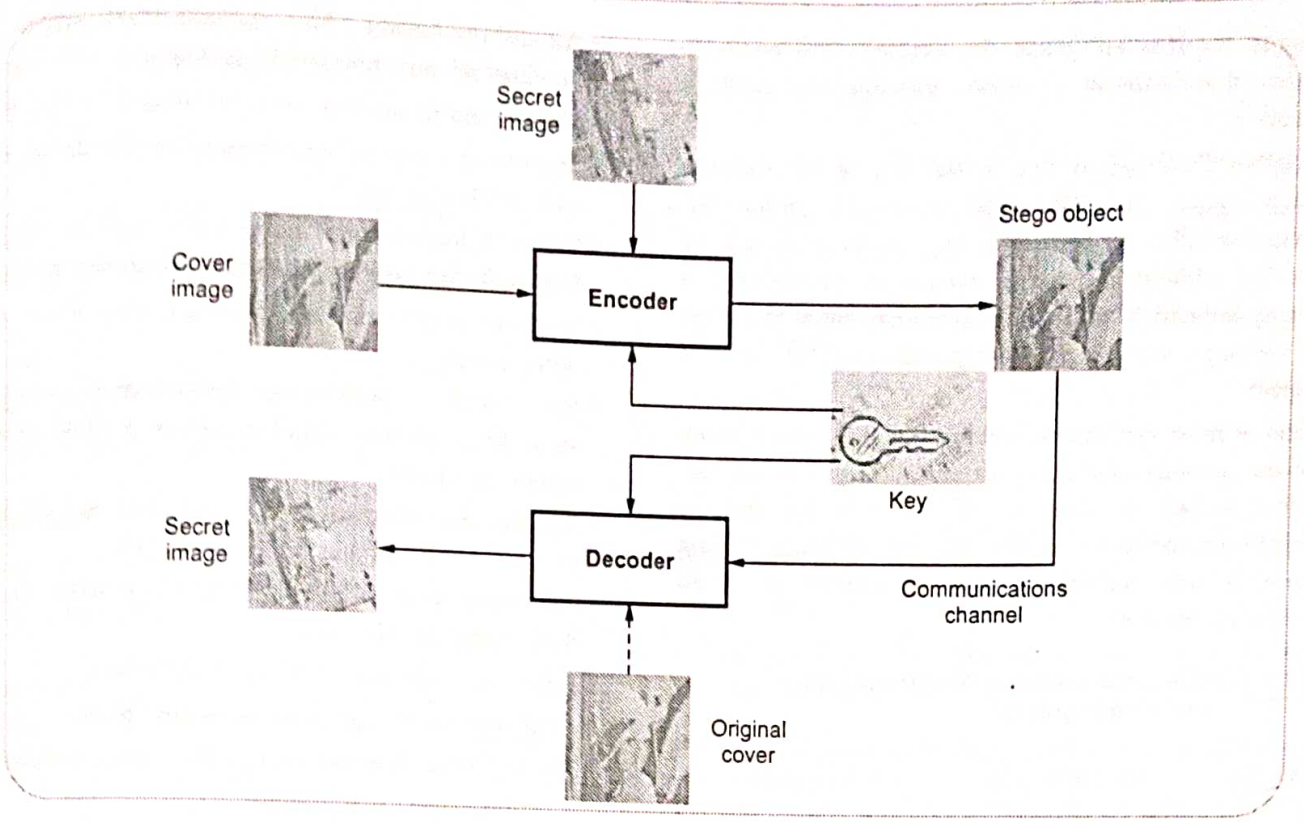

Fig. 2.6.2 Process in steganography

- The recipient must decode the stego object in order for them to view the secret information. The decoding process is simply the reverse of the encoding process. It is the extraction of secret data from a stego object.
- In the decoding process, the stego object is fed into the system. The public or private key that can decode the original key that is used inside the encoding process is also needed so that the secret information can be decoded.
- Depending on the encoding technique, sometimes the original cover object is also needed in the decoding process. Otherwise, there may be no way of extracting the secret information from the stego object.
- After the decoding process is completed, the secret information embedded in the stego object can then be extracted and viewed.
- The generic decoding process again requires a key, K, this time along with a potentially marked object, $\tilde{I}'$. Also required is either the mark, M, which is being checked for or the original object, I and the result will be either the retrieved mark from the object or indication of the likelihood of M being present in $\tilde{I}'$. Different types of robust marking systems use different inputs and outputs.

Steganographic Techniques

1. **Genome steganography :** Encoding a hidden message in a strand of human DNA.
2. **Hiding in text :** Information hidden in documents by manipulating the positions of lines and words, hiding the data in html files.
3. **Hiding in the disk space :** Hiding the data in unused or reserved space.
4. **Hiding data in software and circuitry :** Data can be hidden in the layout of the code distributed in a program or the layout of electronic circuits on a board.
5. **Information hiding in images :** Ranges from least significant bit insertion to masking and filtering to applying more sophisticated image processing algorithms.
6. **Hiding in network packets :** Hidden in packets transmitted through the Internet.

Limitations

- With encryption, Bob can be reasonably sure that he has received a secret message when a seemingly meaningless file arrives. It has either been corrupted or is encrypted. It is not so clear with hidden data; Bob

simply receives an image, for example and needs to know that there is a hidden message and how to locate it.

- Another limitation is due to the size of the medium being used to hide the data. In order for steganography to be useful the message should be hidden without any major changes to the object it is being embedded in. This leaves limited room to embed a message without noticeably changing the original object.
- This is most obvious in compressed files where many of the obvious candidates for embedding data are lost. What is left is likely to be the most perceptually significant portions of the file and although hiding data is still possible it may be difficult to avoid changing the file.

2.6.1 Difference between Steganography and Cryptography

Sr. No.	Steganography	Cryptography
1.	Steganography hides a message within another message and tries to offer the impression that nothing but normal text, audio or video message is being exchanged.	In cryptography the message is encrypted; it looks like a meaningless jumble of characters.
2.	Unknowing message passing.	Knowing message passing.
3.	Steganography does not alter the structure of the secret message.	Cryptography alter the structure of the secret message.
4.	Steganography fails when the "enemy" detects that there is a secret message present in the Steganography medium.	Cryptography fails when the "enemy" is able to access the content of the cipher message.

Review Questions

1. *What is steganography ? Explain its application.*
2. *What is steganography ? What are applications and limitations of steganography ?*

2.7 Block Ciphers

- A block cipher operates on blocks of data.
- Algorithm breaks the plaintext into blocks and operates on each block independently.
- A block cipher operates on blocks of data.
- Algorithm breaks the plaintext into blocks and operates on each block independently.
- Usually blocks are 8 or 16 bytes long.
- Security of block ciphers depends on the design of the encryption function.
- Software implementations of block ciphers run faster than software implementation of the stream ciphers.
- Errors in transmitting one block generally do not affect other blocks.
- Each block is enciphered independently, using the same key, identical plaintext blocks produce identical ciphertext blocks.
- Suppose that plaintext is 227 bytes long and the cipher you are using operates on 16-byte blocks.
- Algorithm grabs the first 16-bytes of data, encrypts them using the key table.
- Algorithm produces 16-bytes of ciphertext.
- After first block, algorithm takes next block.
- The key table does not change from block to block.

$$\text{Plaintext} = 227 \text{ bytes}$$

$$\text{Block size} = 16 \text{ bytes} = \frac{227}{16}$$

$$= 14 \text{ blocks plus } 3 \text{ bytes}$$

- Algorithm encrypts 14 blocks and 3 bytes remain.
- For encrypting last 3 bytes data padding is used.
- Extra bytes are added to make the last block size to 16 bytes.
- Whoever decrypts the ciphertext must be able to recognize the padding.
- One problem with block ciphers is that if the same block of plaintext appears in two places, it encrypts to the same ciphertext.
- To avoid having these kinds of copies in the ciphertext, feedback modes are used.
- Cipher block chaining does not require the extra information to occupy bit spaces, so every bit in the block is part of the message.
- Before a plaintext block is enciphered, that block is XOR'ed with preceding ciphertext block.
- In addition to the key, this technique requires an initialization vector to XOR the initial plaintext block.
- For decrypting the data, copy a block of ciphertext, decrypt it and XOR the result with the preceding block of ciphertext.

- Taking E_K to be the encipherment algorithm with key K and I to be the initialization vector, the cipher block chaining technique is

$$C_0 = E_K(m_0 \oplus I)$$

$$C_i = E_K(m_i \oplus C_{i-1}) \quad \text{for } i > 0$$

2.7.1 Advantages and Disadvantage of Block Cipher

Advantages :

1. High diffusion
2. Immunity to insertation of symbols.

Disadvantages :

1. Slowness of encryption
2. Error propagation.

Review Question

1. *What is block cipher ? Explain counter mode of block cipher.*

2.8 Stream Cipher

- Stream cipher algorithms are designed to accept a crypto key and a stream of plaintext to produce a stream of ciphertext.
- Fig. 2.8.1 shows the stream cipher.

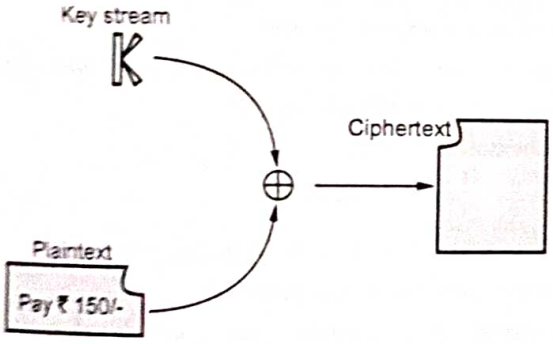

Fig. 2.8.1 Stream cipher

- Stream cipher is similar to a one time pad.
- A stream cipher encrypts smaller block of data, typically bits or bytes.
- A key stream generator outputs a stream of bits $K_1, K_2, K_3 \ldots\ldots K_i$.
- This key stream is XORed with a stream of plaintext bits $P_1, P_2, P_3 \ldots\ldots P_i$ to produce the stream of ciphertext bits.

$$C_i = P_i \oplus K_i$$

- At the description end, the ciphertext bits are XORed with an identical key stream to recover the plaintext bits.

$$P_i = C_i \oplus K_i$$

- The system security depends entirely on the insides of the keystream generator.

2.8.1 Advantages and Disadvantages of Stream Cipher

Advantages :

1. Speed of transformation
2. Low error propagation.

Disadvantages :

1. Low diffusion
2. Susceptibility to malicious insertation and modifications.

2.8.2 Comparison between Stream and Block Cipher

Sr. No.	Stream cipher	Block cipher
1.	Stream ciphers operate on smaller units of plaintext.	Block ciphers operate on larger block of data.
2.	Faster than block cipher.	Slower than stream cipher.
3.	Stream cipher processes the input element continuously producing output one element at a time.	Block cipher processes the input one block of element at a time, producing an output block for each input block.
4.	Requires less code.	Requires more code.
5.	Only one time of key use.	Reuse of key is possible.
6.	Ex. - One time pad	Ex. - DES
7.	Application - SSL (secure connections on the web.)	Application - Database, file encryption.
8.	Stream cipher is more suitable for hardware implementation.	Easier to implement in software.

2.9 Block Cipher Modes of Operation

Different types of cipher block modes are discussed here.

1. Electronic Code Book (ECB)

- A block of plaintext encrypts into a block of Ciphertext. Block size is 64-bits.
- Each block is encrypted independently.
- Plaintext patterns are not concealed since identical blocks of plaintext give identical blocks of ciphertext.
- It is not necessary to encrypt the file linearly.

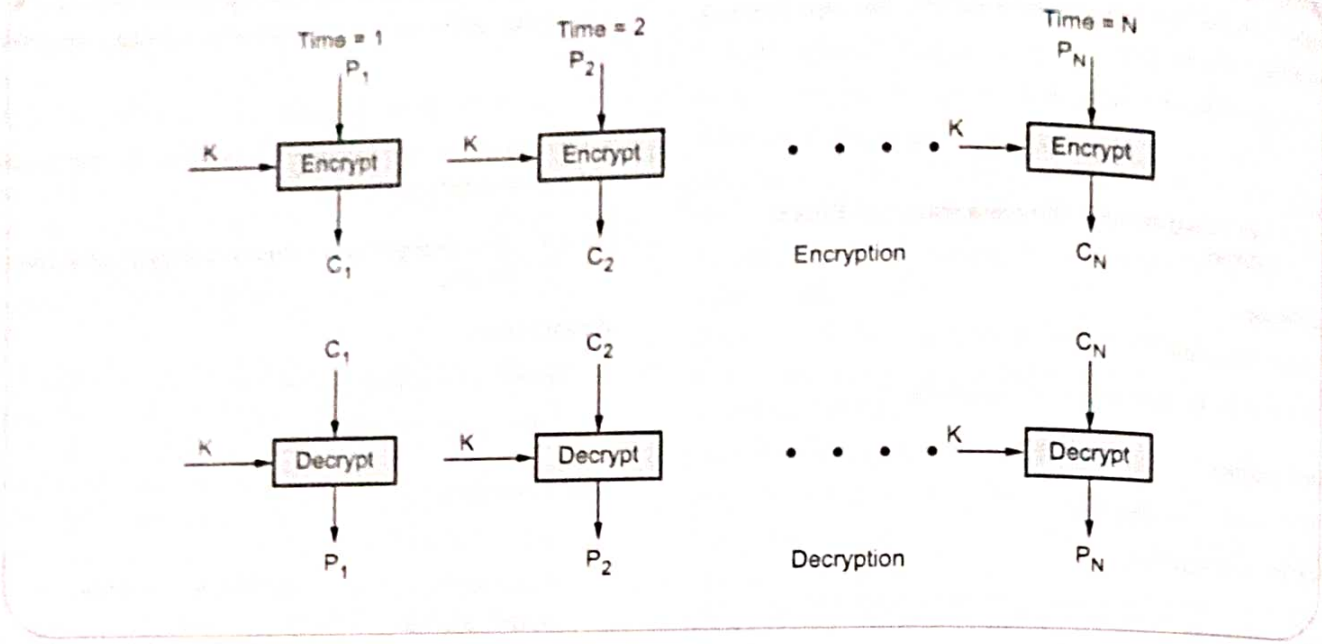

Fig. 2.9.1 ECB mode

- User can encrypt the 10 blocks in the middle first, then the blocks at the end, and finally the blocks in the beginning.
- Because of this, encrypted files are accessed randomly like a data base.
- It is very easy to parallelize the process.
- Pad the last block with some regular pattern i.e. zeros, ones to make it a complete block.
- End of file character is used to denote the final plaintext byte before padding.
- ECB method is ideal for a short amount of data, such as an encryption key.
- For lengthy messages, the ECB mode may not be secure.
- Used in secure transmission of single values i.e. an encryption key.
- ECB has security problems that limit its usability.
- Patterns in the plaintext can yield patterns in the ciphertext.
- It is also easy to modify a ciphertext message by adding, removing or switching encrypted blocks.
- Synchronization error is unrecoverable.

2. Cipher Block Chaining Mode (CBC)

- The plaintext is XORed with the previous ciphertext block before it is encrypted.
- The CBC mode is iterative mode.
- After a plaintext block is encrypted, the resulting ciphertext is also stored in a feedback register.
- Before the next plaintext block is encrypted, it is XORed with the feedback register to become the next input to the encrypting routine.
- The encryption of each block depends on all the previous blocks.
- A ciphertext block is decrypted normally and also saved in a feedback register.
- After the next block is decrypted, it is XORed with the results of the feedback register.
- Mathematically it is

$$C_i = E_k(P_i \oplus C_{i-1})$$

$$P_i = C_{i-1} \oplus D_k(C_i)$$

- It hides patterns in the plaintext.
- In order to guarantee that there is always some random looking ciphertext to apply to the actual plaintext, the process is started with a block of random bits called the Initialization Vector (IV).
- When used in networking messages, most CBC implementations add the IV to the beginning of the message in plaintext.
- A single bit error in a plaintext block will affect that ciphertext block and all subsequent ciphertext blocks.
- CBC mode is self recovering.

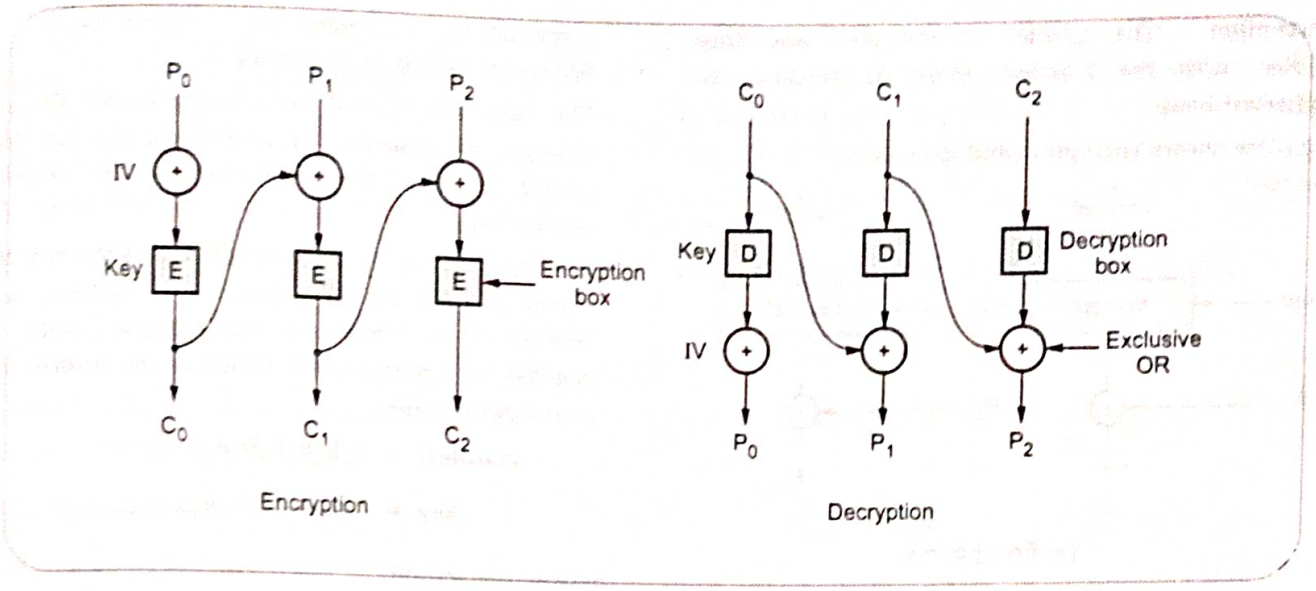

Fig. 2.9.2 CBC

- Two blocks are affected by an error, but the system recovers and continues to work correctly for all subsequent blocks. Synchronization error is unrecoverable.
- Encryption is not parallelizable.
- Decryption is parallelizable and has a random access property.

3. Cipher Feedback Mode (CFB)

- Data is encrypted in units that are smaller than a defined block size.
- It is possible to convert the DES into stream cipher using cipher feedback mode.
- Fig. 2.9.3 shows encryption and decryption process.

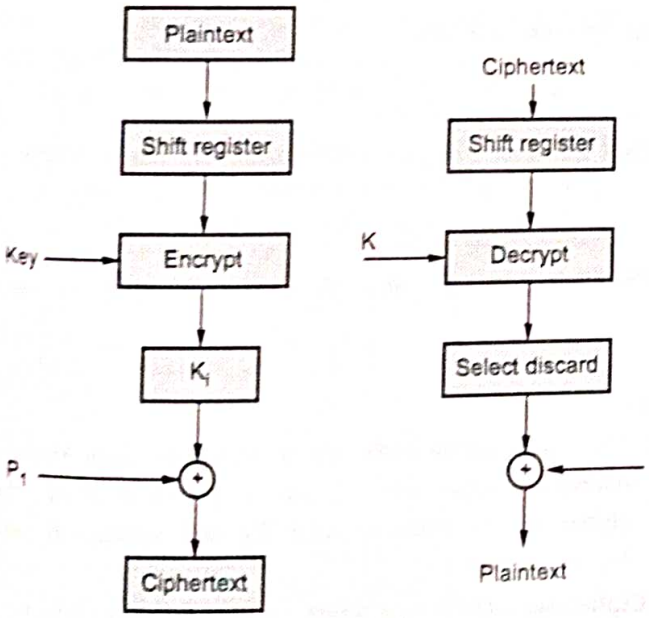

Fig. 2.9.3 CFB Modes

- More than one message can be encrypted with the same key, provided that a different initialization vector is used.
- CFB speed is the same as the block cipher.
- Encryption is not parallelizable, decryption is parallelizable and has a random access property.
- CFB is self recovering with respect to synchronization errors as well.

Advantages

1. Simplicity
2. Need not be used on a byte boundary.
3. Input to the block cipher is randomized.
4. Ciphertext size is the same size as the plaintext size.

Disadvantages

1. Encryption is not parallelizable.
2. Plaintext is somewhat difficult to manipulate.

4. Counter Mode

- Block ciphers in counter mode use sequence numbers as the input to the algorithm.
- More than one message can be encrypted with the same key, provided that a different initialise vector is used.
- Plaintext is very easy to manipulate, any change in ciphertext directly affects the plaintext.
- Synchronization error is unrecoverable.
- A ciphertext error affects only the corresponding bit of plaintext.

- **Encryption** : The counter is encrypted and then XORed with the plaintext block to produce the ciphertext block.
- Fig. 2.9.4 shows encryption and decryption.

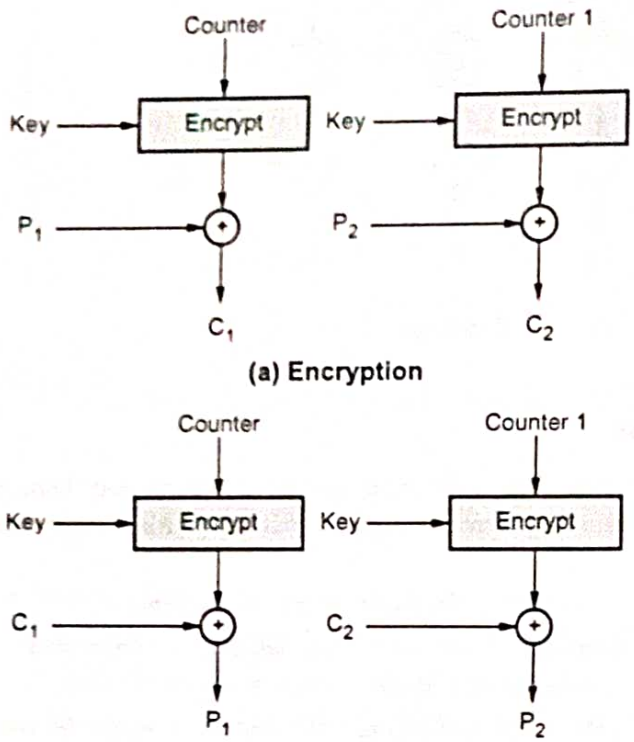

Fig. 2.9.4 Counter mode

Advantages

1. Simple to implement.
2. It provides confidentiality.
3. Random access of block is possible.
4. Efficiency is same as block cipher.

Review Questions

1. *Explain the operation of Cipher Block Chaining (CBC) Mode.*
2. *Explain Cipher Feedback Mode (CFB) Block cipher.*

2.10 Simple DES

- Takes an 8-bit block plaintext, a 10-bit key and produces an 8-bit block of cipher-text.
- Decryption takes the 8-bit block of cipher-text, the same 10-bit key and produces the original 8-bit block of plaintext.
- It was designed as a test block cipher for learning about modern cryptanalytic techniques such as linear cryptanalysis, differential cryptanalysis and linear-differential cryptanalysis.
- The same key is used for encryption and decryption. Though, the schedule of addressing the key bits is altered so that the decryption is the reverse of encryption.
- An input block to be encrypted is subjected to an initial permutation IP. Then, it is applied to two rounds of key-dependent computation. Finally, it is applied to a permutation which is the inverse of the initial permutation.

$$\text{plaintext} = b_1b_2b_3b_4b_5b_6b_7b_8$$

$$\text{key} = k_1k_2k_3k_4k_5k_6k_7k_8k_9k_{10}$$

Subkey generation

- First, produce two subkeys K_1 and K_2:

$$K_1 = P8(LS_1(P10(\text{key})))$$

$$K_2 = P8(LS_2(LS_1(P10(\text{key}))))$$

where P8, P10, LS_1 and LS_2 are bit substitution operators.

- For example, P10 takes 10 bits and returns the same 10 bits in a different order :

$$P10(k_1k_2k_3k_4k_5k_6k_7k_8k_9k_{10}) = k_3k_5k_2k_7k_4k_{10}k_1k_9k_8k_6$$

It's convenient to write such bit substitution operators in this notation :

P10 : (10 bits to 10 bits)

3	5	2	7	4	10	1	9	8	6

P8 : (10 bits to 8 bits)

6	3	7	4	8	5	10	9

LS_1 ("left shift 1 bit" on 5 bit words) : 10 bits to 10 bits

2	3	4	5	1	7	8	9	10	6

LS_2 ("left shift 2 bit" on 5 bit words) : 10 bits to 10 bits

3	4	5	1	2	8	9	10	6	7

Encryption

- The plain text is split into 8-bit blocks; each block is encrypted separately. Given a plaintext block, the cipher text is defined using the two subkeys K_1 and K_2, as follows :

$$\text{Ciphertext} = IP^{-1}(f_{K_2}(SW(f_{K_1}(IP(\text{plaintext}))))))$$

where :

Initial Permutation (IP) : 8 bits to 8 bits

2	6	3	1	4	8	5	7

IP^{-1} (8 bits to 8 bits)

4	1	3	5	7	2	8	6

Switch (SW) : 8 bits to 8 bits

5	6	7	8	1	2	3	4

and $f_K( )$ is computed as follows.

We write exclusive-or (XOR) as +.

$f_K(L, R) = (L + F_K(R) , R)$

$F_K(R) = P4 (S0(lhs(EP(R)+K)) , S1(rhs(EP(R)+K)))$

4 bits to 8 bits

4	1	2	3	2	3	4	1

P4 (4 bits to 4 bits)

2	4	3	1

lhs (8 bits to 4 bits)

1	2	3	4

rhs (8 bits to 4 bits)

5	6	7	8

$S0(b_1 b_2 b_3 b_4)$ = The $[b_1 b_4, b_2 b_3]$ cell from the "S-box" S0 below, and similarly for S1.

S0

	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3
3	3	1	0	3

S1

	0	1	2	3
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	1	1	0	3

- **Algorithm :**

The block of 12 bits is written in the form L_0R_0, where L_0 consists of the first 6 bits and R_0 consists of the last 6 bits. The i^{th} round of the algorithm transforms an input $L_{i-1} R_{i-1}$ to the output $L_i R_i$ using an 8-bit K_i derived from K.

- Fig. 2.10.1 shows one round of a Feistel system.

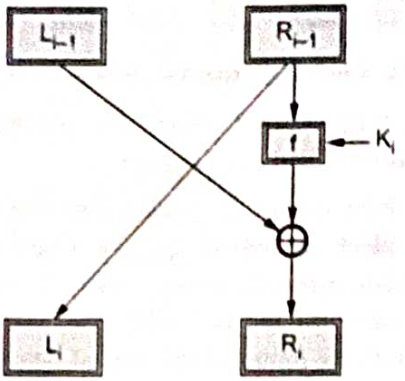

Fig. 2.10.1 One round of a Feistel system

- The output for the i^{th} round is found as follows :

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

- This operation is performed for a certain number of rounds, say n, and produces $L_n R_n$.
- The ciphertext will be $R_n L_n$.
- Encryption and decryption are done the same way except the keys are selected in the reverse order.
- The keys for encryption will be $K_1, K_2, \ldots\ldots K_n$ and for decryption will be $K_n, \ldots K_{n-1} \ldots\ldots K_1$.
- **Function $f(R_{i-1}, K_i)$:** The function $f(R_{i-1}, K_i)$, depicted in the Fig. 2.10.2 below, is described in following steps.

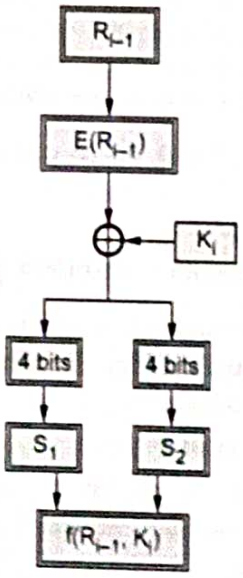

Fig. 2.10.2 The Function $f(R_{i-1}, K_i)$

1. The 6-bits are expanded using the following expansion function. The expansion function takes 6-bit input and produces an 8-bit output. This output is the input for the two S-boxes.

Fig. 2.10.3 The expansion function, $E(R_{i-1})$

2. The 8-bit output from the previous step is Exclusive-ORed with the key K_i
3. The 8-bit output is divided into two blocks. The first block consists of the first 4 bits and the last four bits make the second block. The first block is the input for the first S-box (S1) and the second block is the input for the second S-box (S2).
4. The S-boxes take 4-bits as input and produce 3-bits of output. The first bit of the input is used to select the row from the S-box, 0 for the first row and 1 for the second row. The last 3 bits are used to select the column.
5. The output from the S-boxes is combined to form a single block of 6-bits. These 6 bits will be the output of the function $f(R_{i-1}, K_i)$.

Example : Let the output from the expander function be 11010010.

Solution : 1101 will be the input for the S1 box and 0010 will be the input for the S2 box. The output from the S1 box will be 111, the first of the input is 1 so select the second row and 101 will select the 6^{th} column. Similarly the output from the S2 box will be 110. In above example we have the S1 output 111 and S2 output 110. So the output for the function

$f(R_{i-1}, K_i)$ will be 111110, the S1 output followed by the S2 output.

2.11 Data Encryption Standard (DES)

- DES Encryption standard (DES) is **a symmetric key block cipher** published by the National Institute of Standards and Technology (NIST).
- It encrypts data in 64-bit block.
- DES is symmetric key algorithm : The same algorithm and key is used for both encryption and decryption.
- Key size is 56-bit.
- The encryption process is made of two permutations i.e. P-boxes, which is called initial and final permutation.
- DES uses both transposition and substitution and for that reason is sometimes referred to as a **product cipher**. Its input, output and key are each 64-bits long. The sets of 64-bits are referred to as **blocks**.
- The cipher consists of 16 rounds or iterations. Each rounds uses a separate key of 48-bits.
- Fig. 2.11.1 shows DES encryption algorithm. First, the 64-bit plaintext passes through an Initial Permutation (IP) that rearranges the bits to produce the permuted input. (See Fig. 2.11.1 on next page.)
- Then there is a phase consisting of 16 rounds of the same function, which involves both permutation and substitution functions.
- The output of the sixteenth round consists of 64-bits that are a function of the input plaintext and the key.
- The left and right halves of the output are swapped to produce the pre-output. At last, the pre-output is passed through a permutation (IP^{-1}) that is the inverse of the initial permutation function, to produce the 64-bit ciphertext.

Initial permutation

- Table shows the initial permutation and its inverse. The input to a table consist of 64-bits numbered from 1 to 64.
- The 64 entries in the permutation table contain a permutation of the numbers from 1 to 64. Each entry in the permutation table indicates the positon of a numbered input bit in the output, which also consists of 64-bits.

Initial Permutation (IP) table

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

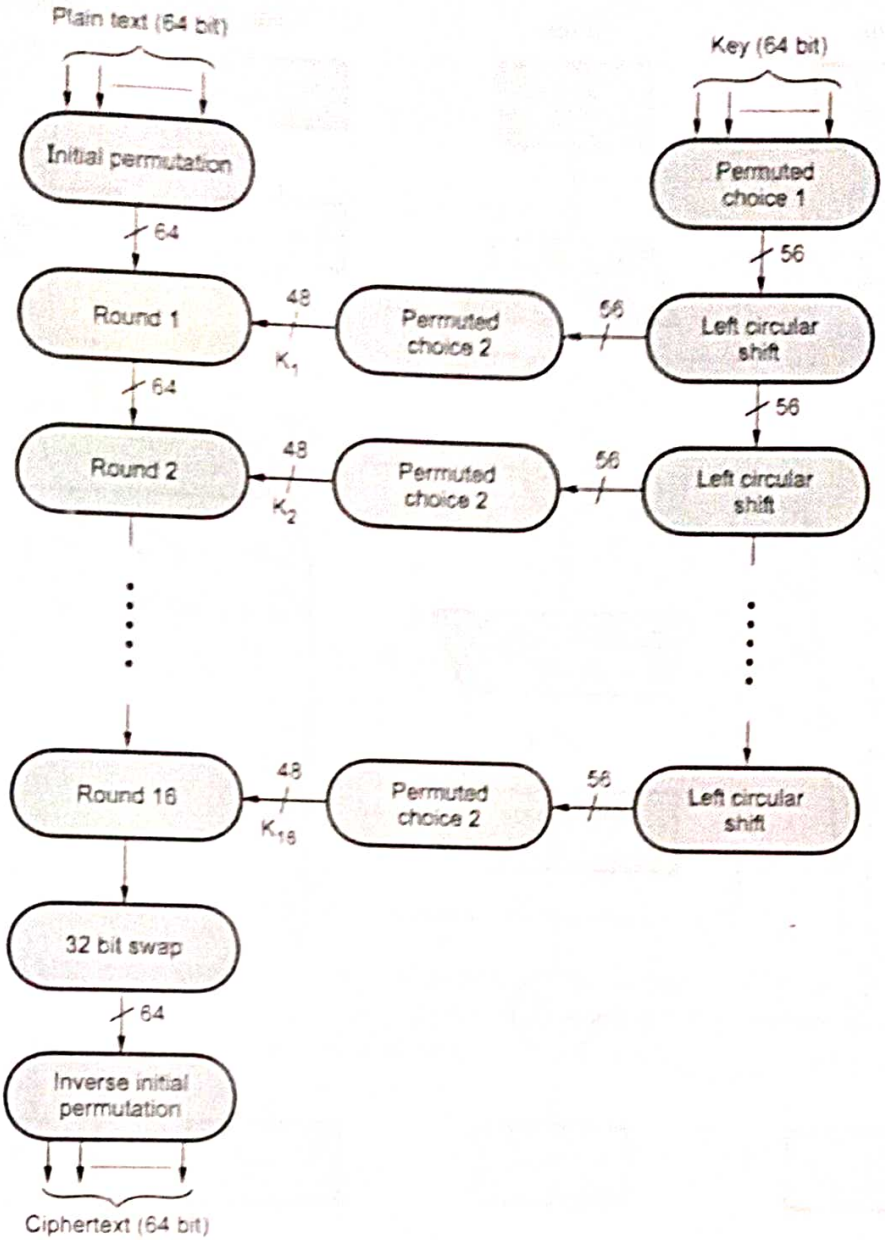

Fig. 2.11.1 DES encryption algorithm

Inverse Initial Permutation (IP^{-1})

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

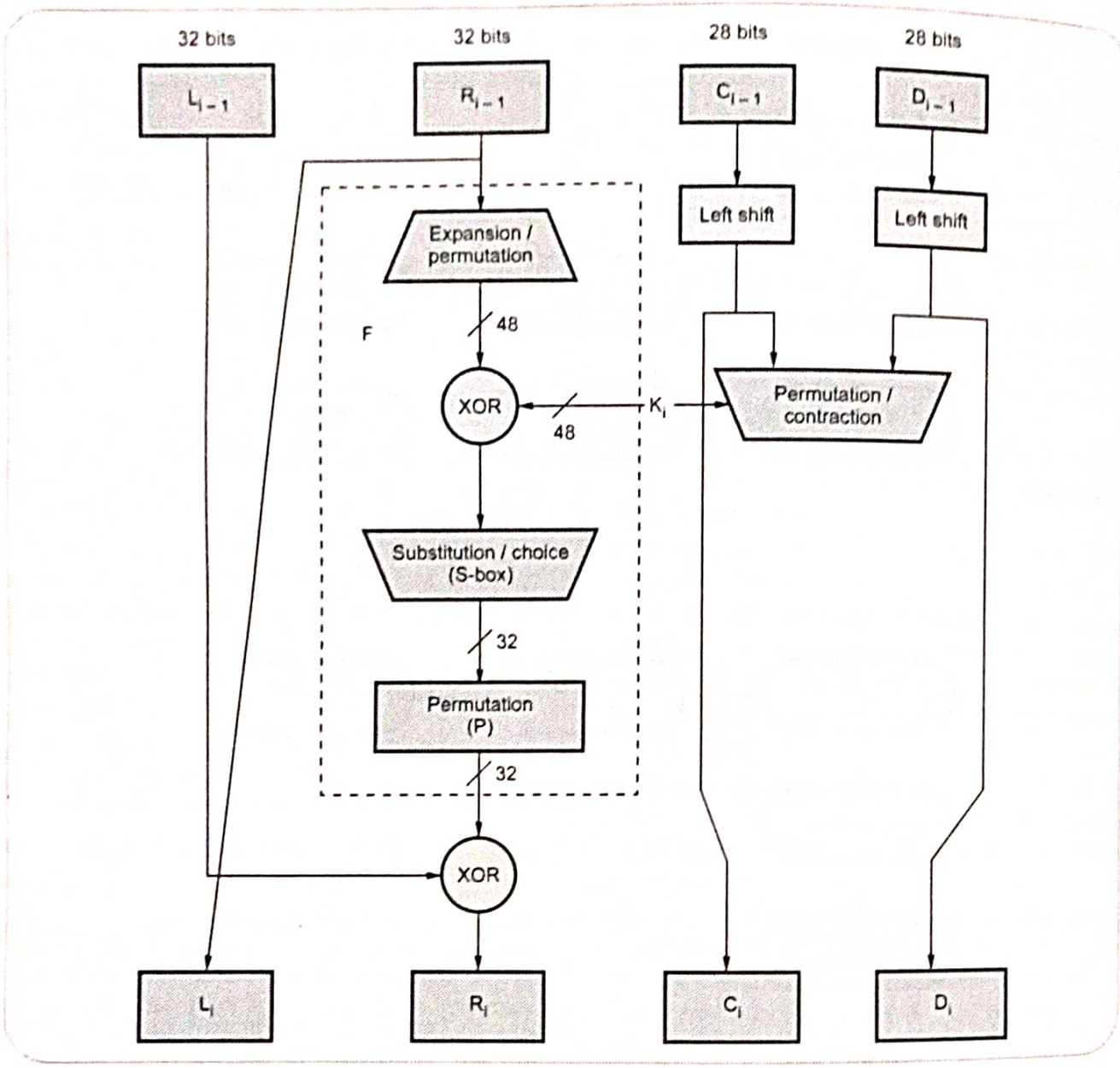

Fig. 2.11.2 Single round of DES algorithm

2.11.1 Details of Single Round

- Fig. 2.11.2 shows single round of DES algorithm. The left and right halves of each 64-bit intermediate value are treated as separate 32-bit quantities, labeled L and R.
- The overall processing at each round can be summarised in the following formulae :

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \text{ X } F(R_{i-1}; K_i)$$

- The left output (L_i) is simply copy of the right input (R_{i-1}). The right output (R_i) is the XOR of left input (L_{i-1}) and right input (R_{i-1}) and key for this stage is K_i. In this stage, the substitution and permutation both functions are used.
- Fig. 2.11.3 shows role of S-boxes in the function F. It consists of set of eight S-boxes, each of which accepts 6 bits as input and produces 4 bits as output.
- The 48 bit input block is divided into 8 subblocks and each subblock is given to a S-box. The S-box transforms the 6 bit input into a 4 bit output.

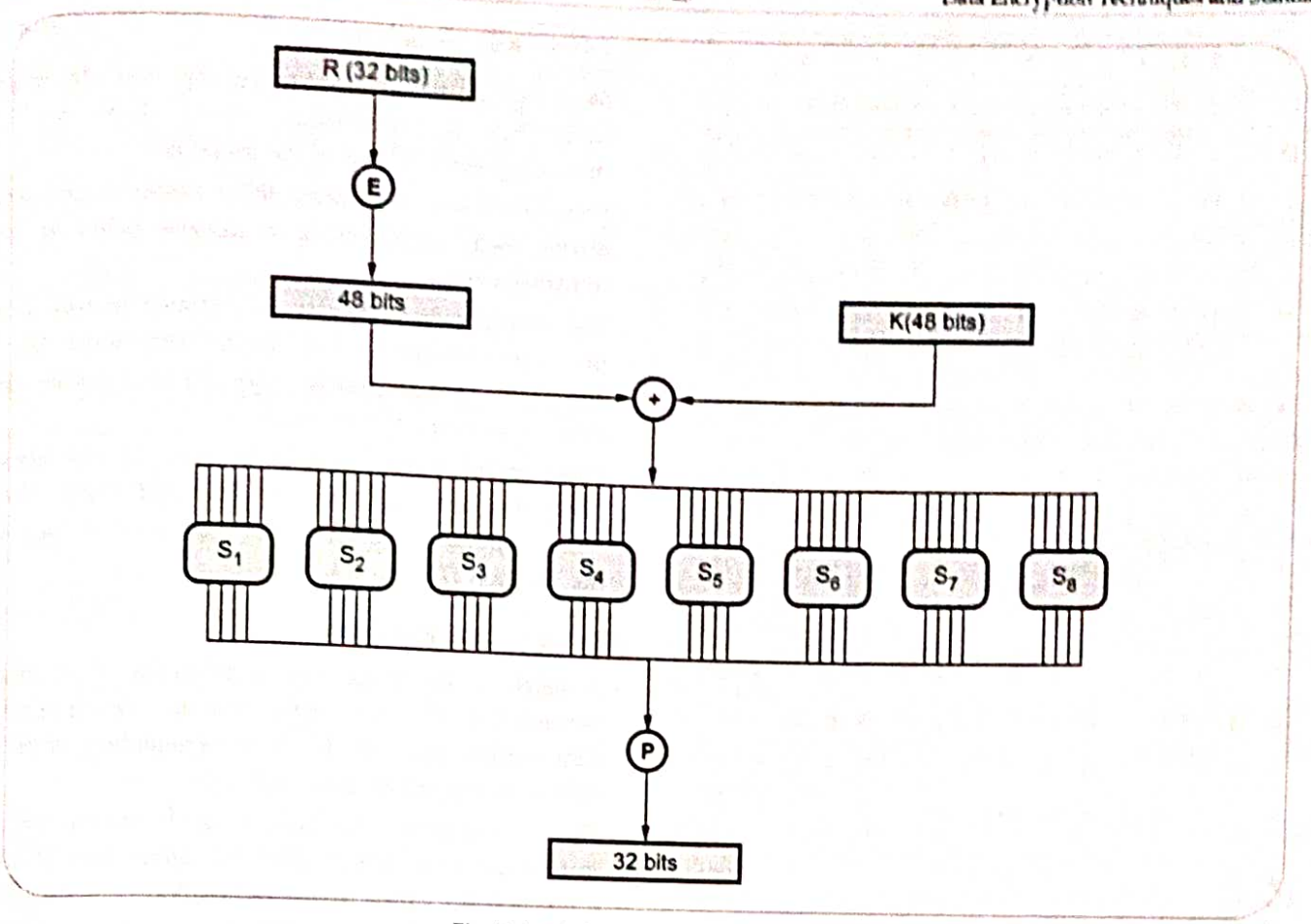

Fig. 2.11.3 S-boxes in the function (F)

- First and last bits of the input to box S_i form a 2-bit binary number to select one of four substitutions defined by the four rows in the table for S_i. Two bits can store any decimal number between 0 and 3. This specifies the row number. The middle four bits select one of the sixteen columns.
- Following table gives the S-box value for DES

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

- Fig. 2.11.4 shows the selection of an entry in a S-box based n the 6-bit input. For example, in S_2, for input 101101, the row is 11 and the column is 0110. The value in row 3, column 6 which select row 3 and column 6 of S_2 box. The output is 4.

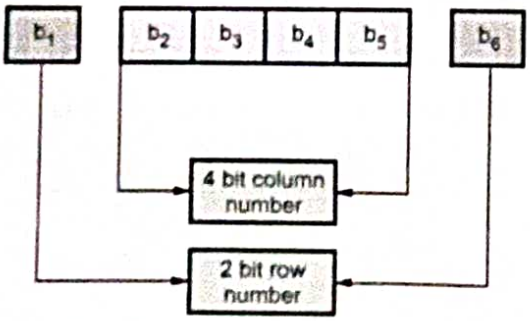

Fig. 2.11.4 Selecting entry in S-box

2.11.2 Key Generation

- 64-bit key is used as input to the algorithm. The initial 64-bit key is transformed into a 56-bit key by discarding every 8^{th} bit of the initial key.
- From 56-bit key, a different 48-bit subkey is generated during each round using a process called as key transformation.
- 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 rotation, of 1 or 2-bits.
- These shifted values serve as input to the next round. They also serve as input to Permuted choice Two, which produces a 48-bit output that serves as input to the function $F(R_{i-1}, K_i)$.

2.11.3 DES Encryption

- A block to be enciphered is subjected to an initial permutation IP, then to a complex key-dependent computation and finally to a permutation which is inverse of the initial permutation IP.
- The key-dependent computation can be simply defined in terms of a function *f*, called the cipher function, and a function KS, called the key schedule.
- Given two blocks L and R of bits, LR denotes the block consisting of the bits of L followed by the bits of R.

1. **Initial permutation** : The 64-bits of the input block to be enciphered are first subjected to the permutation, called the initial permutation.
2. **Key dependent computation** : The computation which uses the permuted input block as its input to produce the pre-output block consists. Cipher function *f* which operates on two blocks, one of 32-bits and one of 48-bits, and produces a block of 32-bits. Let the 64 bits of the input block in an iteration consist of a 32-bit block L followed by a 32-bit block R. Using the notation defined in the introduction the input block is then LR. Let K be a block of 48 bits chosen from the 64-bit key. Then the output L' R' of an iteration with input LR is defined by :

$$\left.\begin{aligned} L' &= R \\ R' &= L (+) f(R, K) \end{aligned}\right\} \quad \ldots (2.11.1)$$

where (+) denotes bit-by-bit addition modulo 2.

As before, let the permuted input block be LR. Finally, let L_0 and R_0 be respectively L and R and let L_n and R_n be respectively L and R of equation (2.11.1); hence L and R are respectively L_{n-1} and R_{n-1} and K is K_n, i.e. when n is in the range from 1 to 16,

$$\text{Then } L_n = R_{n-1}$$

$$R_n = L_{n-1} \oplus f(R_{n-1}, K_n)$$

The pre-output block is then $R_{16}L_{16}$.

3. **Key schedule** : Key generation technique is shown in the Fig. 2.11.5.

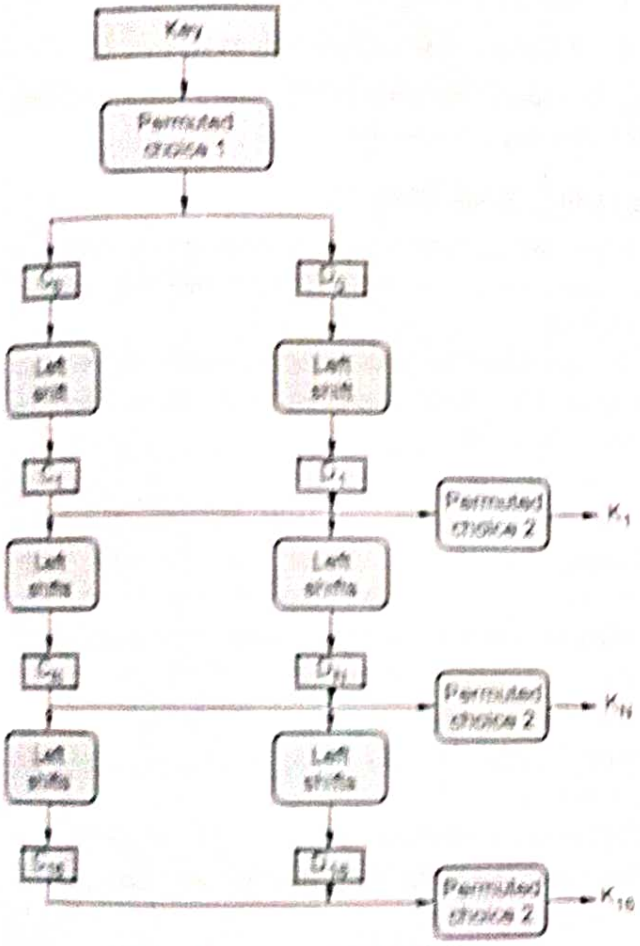

Fig. 2.11.5 Key generation techniques

The input of the first iteration of the calculation is the permuted input block. If L, R is the output of the 16th iteration then RL is the pre-output block. At each iteration a different block K of key bits is chosen from the 64-bit key designated by KEY. Let KS be a function which takes a integer n in the range from 1 to 16 and a 64-bit block KEY as input and yields as output a 48-bit block K_n which is a permuted selection of bits from KEY i.e.

$$K_n = KS(n, KEY)$$

with K_n determined by the bits in 48 distinct bit positions of KEY. KS is called the key schedule.

2.11.4 DES Decryption

- The permutation IP^{-1} applied to the pre-output block is the inverse of the initial permutation IP applied to the input. Consequently, to decipher it is only necessary to apply the very same algorithm to an enciphered message block, taking care that at each iteration of the computation the same block of key bits K is used during decipherment as was used during the encipherment of the block only in a reverse order.
- For the decipherment calculation with $R_{16}L_{16}$ as the permuted input, K_{16} is used in the first iteration, K_{15} in the second, and so on, with K_1 used in the 16th iteration.

2.11.5 DES Weak Keys

- With many block ciphers there are some keys that should be avoided, because of reduced cipher complexity.
- These keys are such that the same sub-key is generated in more than one round, and they include :
  1. **Weak keys** : The same sub-key is generated for every round and DES has 4 weak keys.
  2. **Semi-weak keys** : Only two sub-keys are generated on alternate rounds and DES has 12 of these (in 6 pairs).
  3. **Demi-semi weak keys** : Have four sub-keys generated.
- None of these cause a problem since they are a tiny fraction of all available keys however they MUST be avoided by any key generation program.

2.11.6 Advantages of DES

1. As 56-bit keys are used there are 70 quadrillion possible key values and hence a specific key cannot be identified easily.
2. As the length of the key is increased the security provided by the algorithm also increases.
3. The security of the DES algorithm resides in the key.

2.11.7 Disadvantages of DES

1. As it is a symmetric algorithm both sender and receiver must have same key, there is a possibility that the key is intercepted.

2. The design of S boxes makes it susceptible to linear cryptanalysis attack.
3. It is susceptible to differential cryptanalysis attack and brute force attack taking advantage of which DES crackers have been designed.
4. It has certain weak keys which generate the same key for all cycles of the algorithm like when all key bits are either 0s or 1s or if one half of the key bits are 0s or 1s. They are 0000000 0000000, 0000000 ffffffff, ffffffff 0000000, ffffffff ffffffff.
5. Some initial keys produce only two subkeys while some produce only four. They are called possible weak keys.

Possible techniques for improving DES

- Multiple enciphering with DES
- Extending DES to 128-bit data paths and 112-bit keys
- Extending the key expansion calculation.

2.11.8 Block Cipher Design Principles

The criteria for the **S-boxes** are as follows :

1. No output bit of any S-box should be too close a linear function of the input bits.
2. Each row of an S-box should include all 16 possible output bit combinations.
3. If two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits.
4. If two inputs to an S-box differ in the two middle bits exactly, the outputs must differ in at least two bits.
5. If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must not be the same.
6. For any non zero 6-bit difference betwen inputs, no more than 8 of the 32 pairs of inputs exhibiting that difference may result in the same output difference.

Criteria for **permutation P** are as follows.

1. The four output bits from each S-box at round i are distributed so that two of them affect middle bits of round (i + 1) and the other two affect end bits.
2. The four output bits from each S-box affect six different S-boxes on the next round, and no two affect the same S-box.
3. For two S-boxes j, k, if an output bit from S_j affects a middlle bits of S_{tock} on the next round, then an output bit from S_k cannot affect a middle bit of S_j.

2.11.9 Double DES

- Using two encryption stages and two keys.
  A) The plain text to ciphertext is as follows,
  $C = E_{K_2}(E_{K_1}(P))$ where k_1 and k_2 are the key.
  B) Ciphertext to plain text is as follows,
  $P = D_{K_1}(D_{K_2}(C))$
- Double DES suffers from Meet-in-the-Middle Attack.
- Meet-in-the-Middle Attack is as follows,
  1. Assume $C = E_{K_2}(E_{K_1}(P))$
  2. Given the plaintext P and ciphertext C
  3. Encrypt P using all possible keys K_1
  4. Decrypt C using all possible keys K_2

Fig. 2.7.6 shows the meet-in-the-middle attack for double DES. (See Fig. 2.7.6 on next page.)

2.11.10 Triple DES

- Triple DES is simply another mode of DES operation. It takes three 64-bit keys, for an overall key length of 192 bits.
- The procedure for encryption is exactly the same as regular DES, but it is repeated three times. Hence the name triple DES.
- Triple DES uses 2 or 3 keys.
- The data is encrypted with the first key (K_1), decrypted with the second key (K_2), and finally encrypted again with the third key (K_3).
- Triple DES with three keys is used quite extensively in many products including PGP and S/MIME.
- Brute force search impossible on Triple DES.
- Meet-in-middle attacks need 256 Plaintext-Ciphertext pairs per key.
- Cipher text is produced as $C = E_{K_3}[D_{K_2}[E_{K_1}[P]]]$
- Fig. 2.11.7 shows the 3DES method with three key.

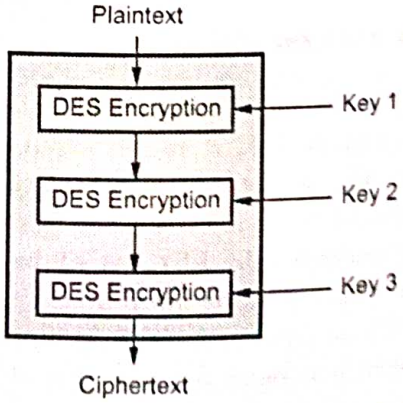

Fig. 2.11.7 3DES with three key method

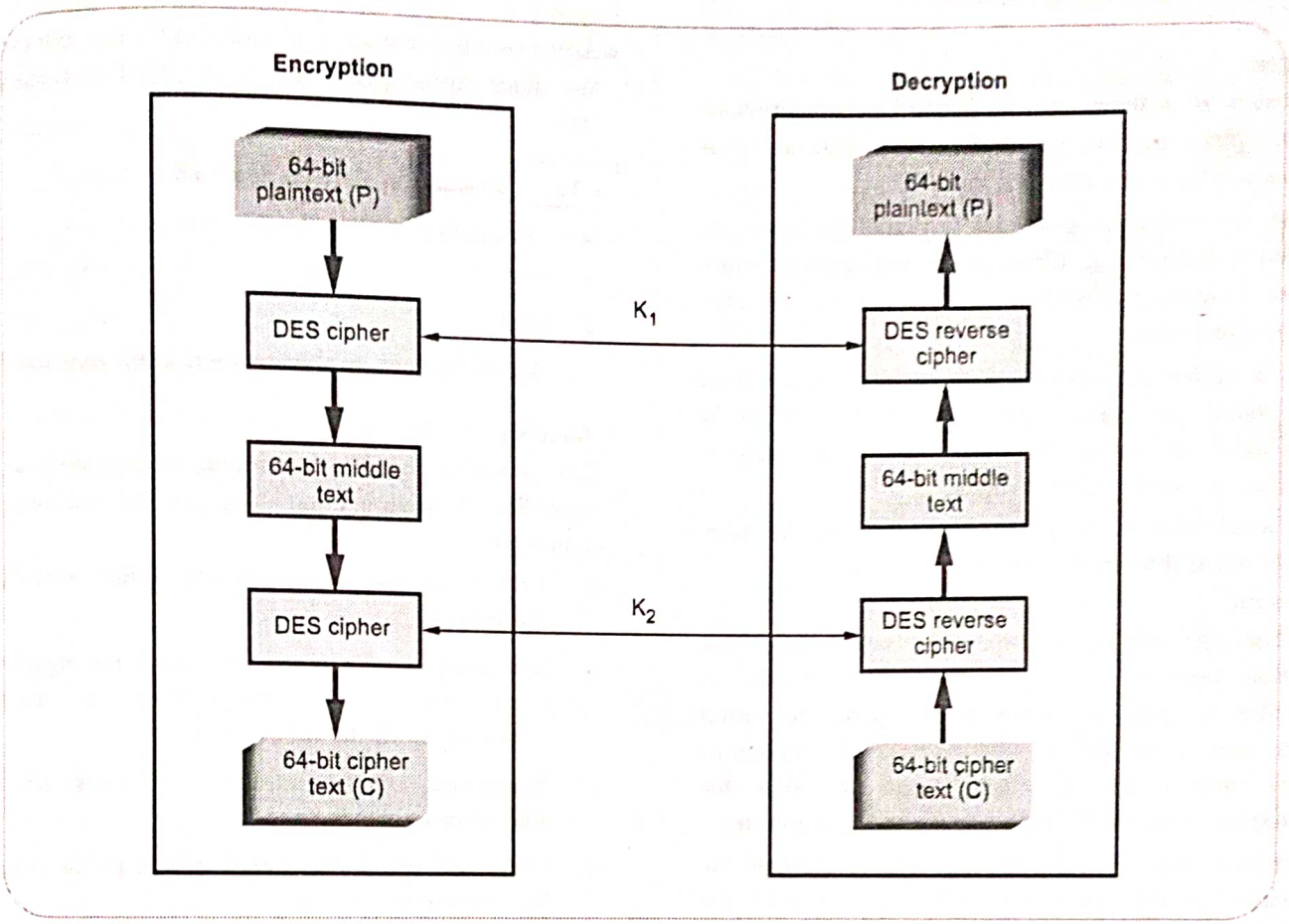

Fig. 2.11.6 Meet-in-the-middle attack for double DES

- Triple DES runs three times slower than standard DES, but is much more secure if used properly.
- The procedure for decrypting something is the same as the procedure for encryption, except it is executed in reverse.
- Like DES, data is encrypted and decrypted in 64-bit chunks.
- There are some weak keys that one should be aware of : If all three keys, the first and second keys, or the second and third keys are the same, then the encryption procedure is essentially the same as standard DES. This situation is to be avoided because it is the same as using a really slow version of regular DES.
- The input key for DES is 64-bits long; the actual key used by DES is only 56-bits in length.
- The least significant (right-most) bit in each byte is a parity bit, and should be set so that there are always an odd number of 1s in every byte.
- These parity bits are ignored, so only the seven most significant bits of each byte are used, resulting in a key length of 56-bits.
- This means that the effective key strength for Triple DES is actually 168-bits because each of the three keys contains 8 parity bits that are not used during the encryption process.

Review Questions

1. *Explain the operation of DES algorithm in detail.*
2. *Explain 3 DES algorithm in detail.*
3. *What is weak key in DES algorithm ? Explain with example.*
4. *Explain operation of 3DES algorithm.*
5. *Explain the operation of triple DES algorithm.*

2.12 Confusion and Diffusion

Diffusion

- Diffusion is making output dependent on previous input (plain/cipher-text). Ideally, each output bit is influenced by every previous input bit.
- These are measures to thwart cryptanalysis based on statistical analysis. In diffusion, the statistical structure of the plaintext is dissipated into long range statistics of the cipher-text.
- This is achieved by having each plaintext letter affect the value of many cipher-text digits, which is equivalent to saying that each cipher-text digit is affected by many plaintext digits.
- The letter frequencies in the cipher-text will be more nearly equal than in the plaintext.

Confusion

- In Shannon's original definitions, confusion makes the relation between the key and the cipher-text as complex as possible. Confusion is making the output dependent on the key. Ideally, every key bit influences every output bit. Confusion tries to hide the connection between the cipher-text and the secret key.
- Confusion seeks to make the relationship between the statistics of the cipher-text and the value of the encryption key as complex as possible. This is achieved by the use of a complex substitution algorithm. These operations became the cornerstone of modern block cipher design.

2.12.1 Distinguish between Diffusion and Confusion

No.	Diffusion	Confusion
1.	Diffusion hides the relation between the ciphertext and the plaintext.	Confusion hides the relation between the ciphertext and key.
2.	If a single symbol in the plaintext is changed, several or all symbols in the ciphertext will also be changed.	If a single bit in the key is changed, most or all bits in the ciphertext will also be changed.
3.	In diffusion, the statistical structure of the plain text is dissipated into long-range statistics of the cipher text. This is achieved by permutation.	In confusion, the relationship between the statistics of the cipher text and the value of the encryption key is made complex. It is achieved by substitution.

2.13 Advance Encryption Standard (AES)

- Advanced Encryption Standard (AES) is a symmetric key block cipher published by the NIST in December 2001.

2.13.1 Evaluation Criteria for AES

- NIST evaluation criteria for AES are
  1. Security
  2. Cost
  3. Algorithm and implementation characteristics.

1. Security

- This refers to the effort required to cryptanalyse an algorithm. Following parameters are also consider for evaluation.
  a. **Actual security** compared to other submitted algorithms.
  b. **Randomness :** The extent to which the algorithm output is indistinguishable from a random permutation on the input block.
  c. **Soundness** of the mathematical basis for the algorithm's security.
  d. Other security factors raised by the public during the evaluation process.

2. Cost

a. **Licensing requirements :** When the AES is issued, the algorithm specified in the AES shall be available on a worldwide, non-exclusive, royalty free basis.

b. **Computational efficiency :** The evaluation of computational efficiency will be applicable to both hardware and software implementations.

c. **Memory requirements :** The memory requirement for implementing the algorithm in hardware and software will be considered.

3. Algorithm and Implementation Characteristics

This category includes a variety of considerations, including flexibility, suitability for a variety of hardware and software implementations; and simplicity, which will make an analysis of security more straight forward.

The following criteria were used in the final evaluation :

1. **General security :** NIST relied on the public security analysis conducted by the cryptographic community.

2. **Software implementations :** It includes execution speed, performs across a variety of platforms and variation of speed with key size.
3. Restricted space environments.
4. Hardware implementations.
5. Attacks on implementations.
6. Encryption versus decryptions.
7. Key agility.
8. Other versatility and flexibility.
9. Potential for instruction level parallelism.

2.13.2 AES Cipher

- AES is a non-Feistel cipher that encrypts and decry a data block of 128-bits.
- The key size can be 128, 192 or 256-bits. It depends number of rounds.
- The number of rounds : 10 rounds for 128-b 12 rounds for 192-bits and 14 rounds for 256-bits.

Characteristics

1. Resistance against all known attacks.
2. Speed and code compactness on a wide range platforms.

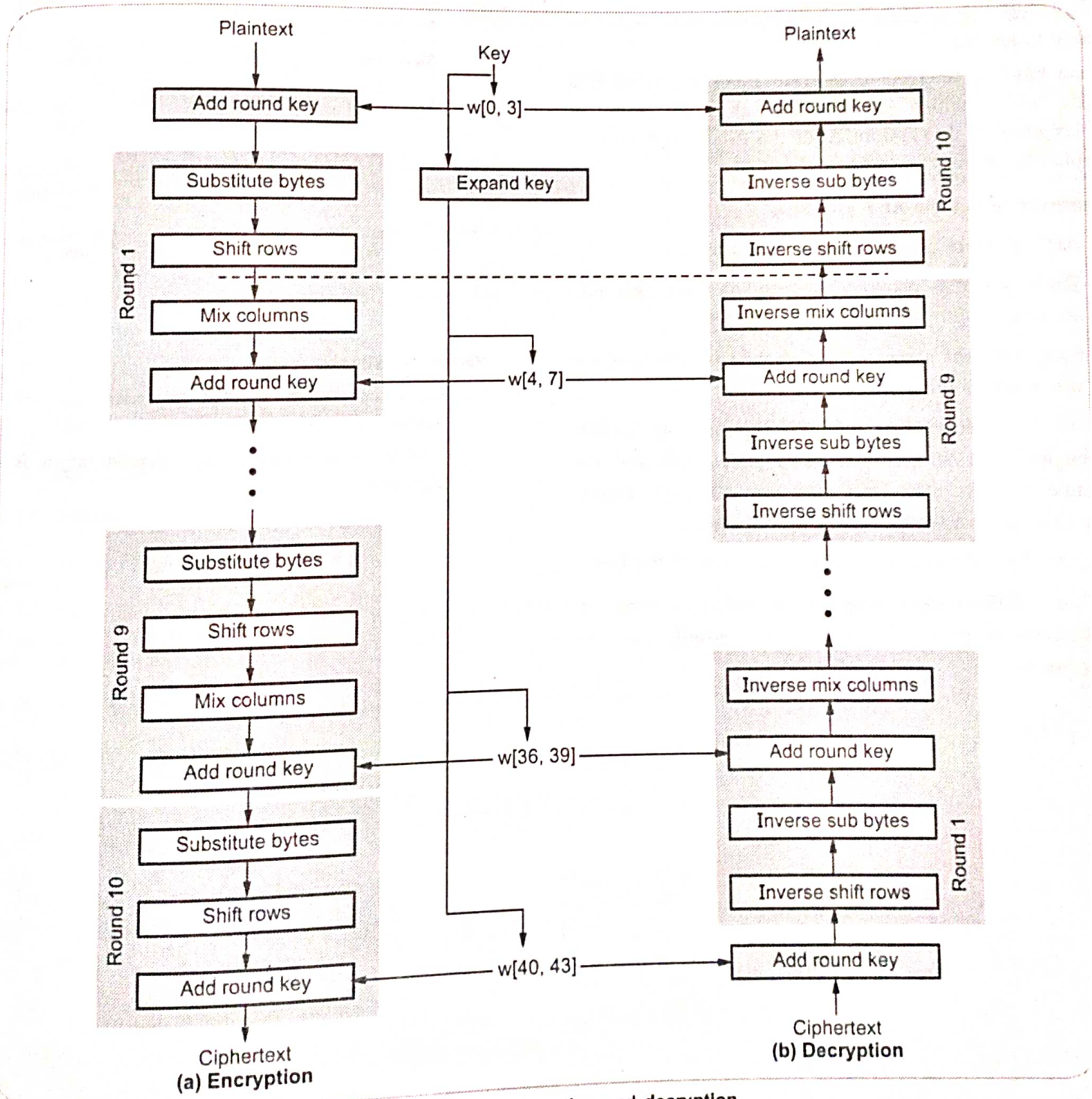

Fig. 2.13.1 AES encryption and decryption

3. Design simplicity.

- For 128-bits AES, each round contains four steps :
  i. Byte substitution
  ii. Row shift
  iii. Column mixing
  iv. Round key addition
- The input to the encryption and decryption algorithms is a single 128-bit block. The block is represented as a row of matrix of 16 bytes.
- Fig. 2.13.1 shows the overall structure of AES. (See Fig. 2.13.1 on previous page.)
- AES use several rounds in which each round is made of several stages. Data block is transformed from one stage to another.
- Data block is referred to as **state**. Block is copied into state array which is modified at each stage of encryption or decryption. After the final stage, state is copied to an output matrix.

Comments about the AES structure

1. AES structure is not a Feistel structure.
2. The key that is provided as input is expanded into an array of forty-four 32-bit words, w(i).
3. Four different stages are used, one of permutation and three of substitution.
4. For both encryption and decryption, the cipher begins with an AddRoundkey stage, followed by nine rounds that each includes all four stages, followed by a tenth round of three stages.
5. Only the AddRoundkey stage make use of the key.
6. The AddRoundkey stage is, in effect, a form of Vernam Cipher and by itself would not be formidable.
7. Each stage is easily reversible.
8. The decryption algorithm makes use of the expanded key in reverse order.
9. Once it is established that all four stages are reversible, it is easy to verify that decryption does recover the plaintext.
10. The final round of both encryption and decryption consists of only three stages.

2.13.3 Comparison between AES and DES

Sr. No.	Parameters	AES	DES
1	Block size	128-bits	64-bits
2	Key length	128, 192, 256-bits	56-bits (effective length)
3	Encryption primitives	Substitution, shift, bit mixing	Substitution, Permutation
4	Cryptographic primitives	Confusion, Diffusion	Confusion, Diffusion
5	Design rationale	Closed	Open

Review Questions

1. *Explain operation of AES algorithm and state its application.*
2. *Explain the operation in key expansion process in AES algorithm.*