Module-1

Classical Encryption Techniques

Contents

§ A model for Network Security

§ Classical encryption techniques

- Symmetric cipher model
- Substitution ciphers
- Caesar Cipher
- Monoalphabetic Cipher
- Playfair Cipher
- Hill Cipher
- Polyalphabetic Ciphers
- One time pad,
- Steganography.

A Model for Network Security

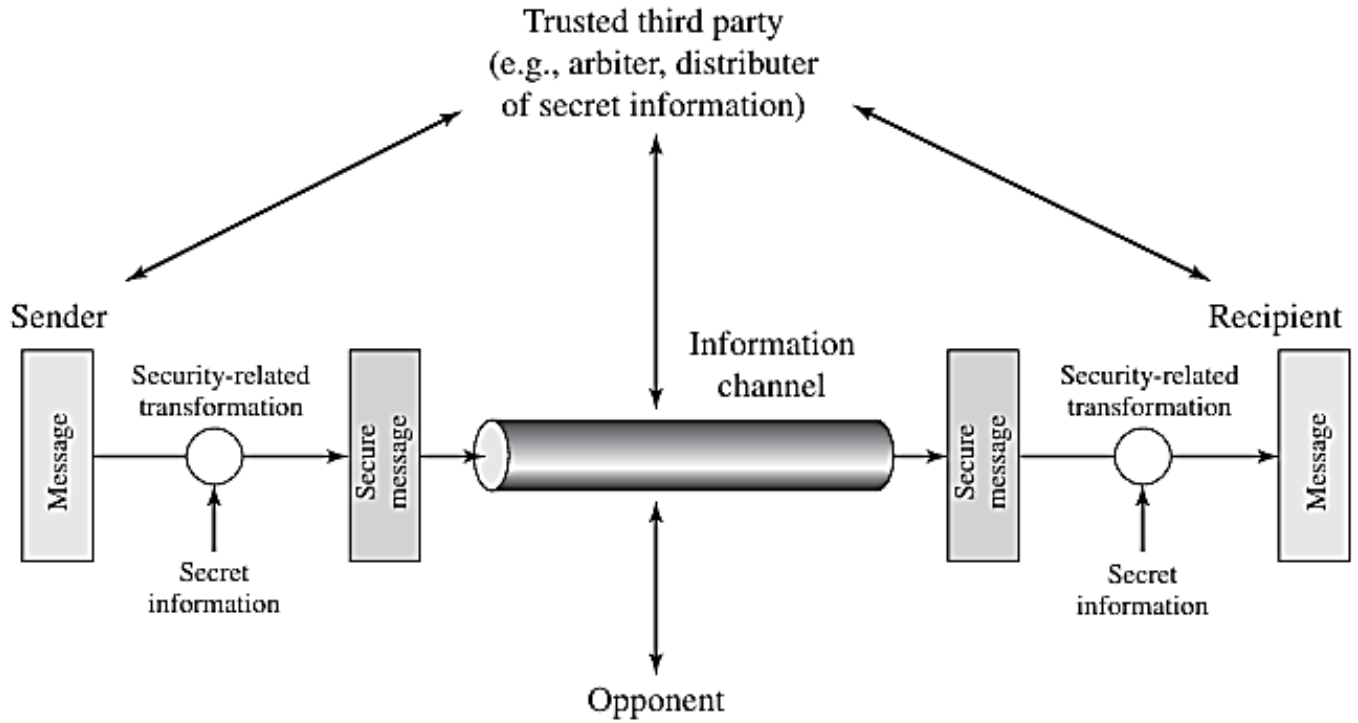


Figure 1.1 : Model for Network Security

§ All the techniques for providing security have two components:

1. A security-related transformation on the information to be sent. Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent, and the addition of a code based on the con tents of the message, which can be used to verify the identity of the sender.
2. Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception.

§ A trusted third party may be needed to achieve secure transmission.

— For example, a third party may be responsible for distributing the secret information to the two principals while keeping it from any opponent.

— Or a third party may be needed to arbitrate disputes between the two principals concerning the authenticity of a message transmission.

§ This general model shows that there are four basic tasks in designing a particular security service:

1. Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose.
2. Generate the secret information to be used with the algorithm.
3. Develop methods for the distribution and sharing of the secret information.
4. Specify a protocol to be used by the two principals that makes use of the security algorithm and the secret information to achieve a particular security service.

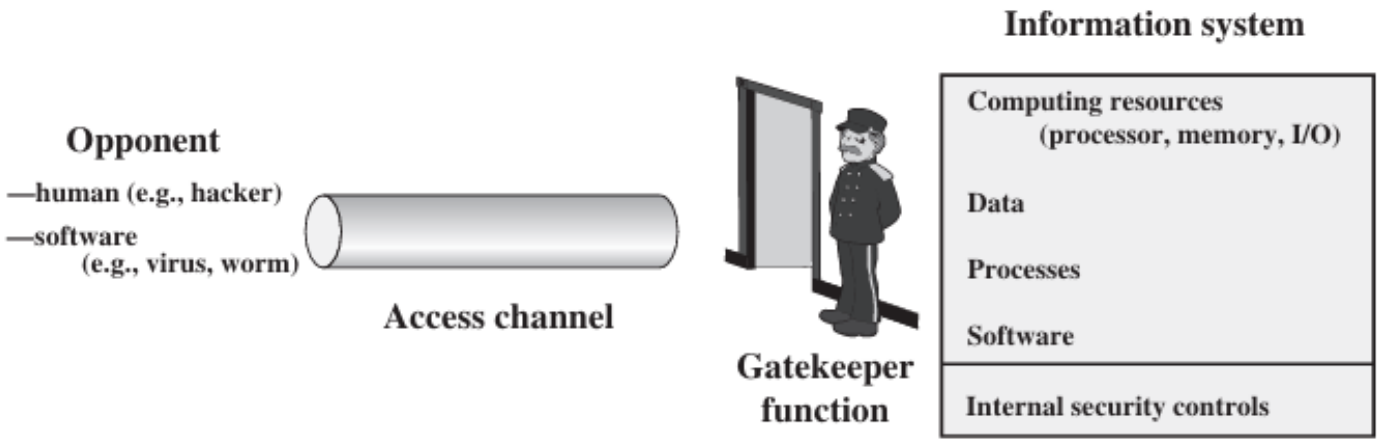


Figure 1.2 : Network Access Security Model

- § Figure 1.2 reflects a concern for protecting an information system from un wanted access.
- § The hacker can be someone who, with no malign intent, simply gets satisfaction from breaking and entering a computer system.
- § The intruder can be a disgruntled employee who wishes to do damage or a criminal who seeks to exploit computer assets for financial gain (e.g., obtaining credit card numbers or performing illegal money transfers).
- § Another type of unwanted access is the placement in a computer system of logic that exploits vulnerabilities in the system and that can affect application programs as well as utility programs, such as editors and

§ Another type of unwanted access is the placement in a computer system of logic that exploits vulnerabilities in the system and that can affect application programs as well as utility programs, such as editors and compilers.

§ Programs can present two kinds of threats:

1. Information access threats: Intercept or modify data on behalf of users who should not have access to that data.
2. Service threats: Exploit service flaws in computers to inhibit use by legitimate users

§ Viruses and worms are two examples of software attacks. Such attacks can be introduced into a system by means of a disk that contains the unwanted logic concealed in otherwise useful software. They can also be inserted into a system across a network

§ The security mechanisms needed to cope with unwanted access fall into two broad categories

1. The first category might be termed a gate keeper function. It includes password-based login procedures that are designed to deny access to all but authorized users and screening logic that is designed to detect and reject worms, viruses, and other similar attacks.
2. Once either an unwanted user or unwanted software gains access, the second line of defense consists of a variety of internal controls that monitor activity and analyze stored information in an attempt to detect the presence of unwanted intruders.

Basic Concepts

§ **Plaintext:** The original message

§ **Cipher text :** The coded message

§ **Enciphering / Encryption:** The process of converting plaintext to cipher text using a cipher and a key

§ **Deciphering / Decryption:** the process of restoring the plaintext from the cipher text

§ **Cryptanalysis :** techniques used for deciphering a message without any knowledge of the enciphering details .Also called code breaking

§ **Cryptology :** Both cryptography and cryptanalysis

Symmetric Cipher Model

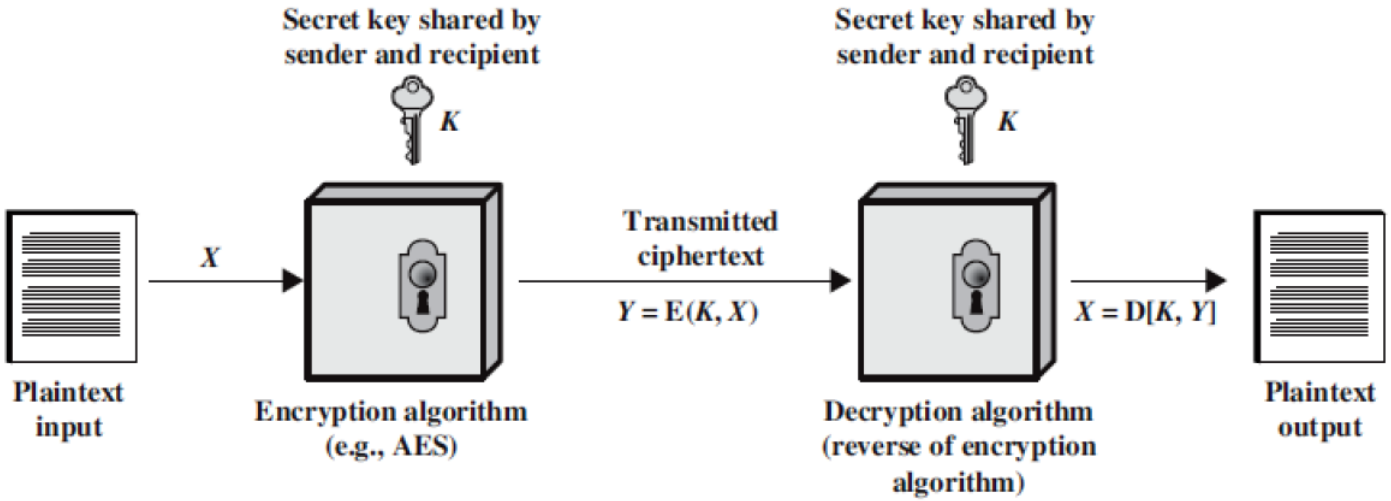


Fig: Simplified Model of Symmetric Encryption

§ A symmetric encryption scheme has five ingredients

1. Plaintext: The original intelligible message or data that is fed into algorithm as input
2. Encryption algorithm: performs various substitution and transformations on the plaintext
3. Secret key: input to the encryption algorithm.
4. Cipher text: scrambled message produced as output
5. Decryption algorithm: takes cipher text and secret key and produces the original plaintext

§ Two requirements for secure use of symmetric encryption

– a strong encryption algorithm

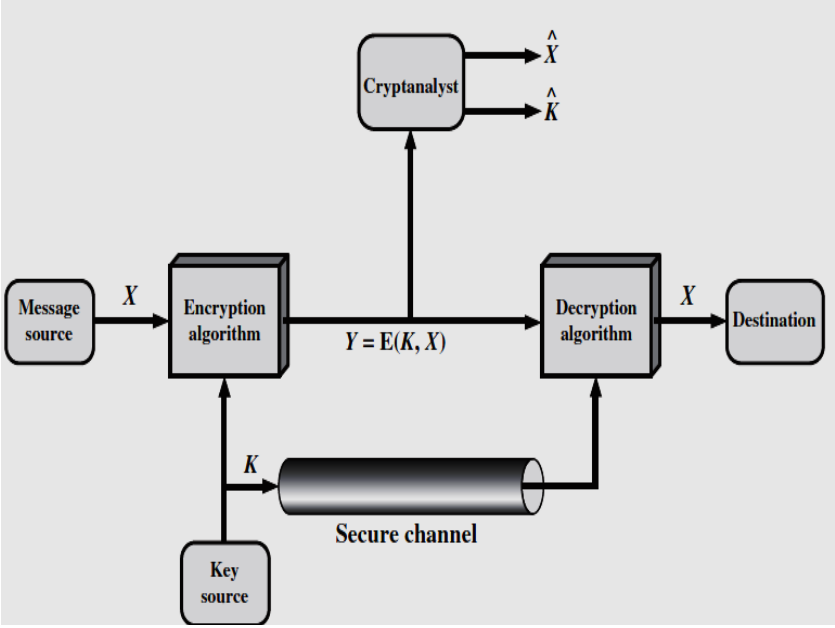


Fig: Model of Symmetric Cryptosystem

§ A source produces a message in plaintext, $X = [X_1, X_2, .., X_M]$.

§ For encryption, a key of the form $K = [K_1, K_2, \ldots, K_J]$ is generated.

§ If the key is generated at the message source, then it must also be provided to the destination by means of some secure channel

§ Alternatively, a third party could generate the key and securely deliver it to both source and destination

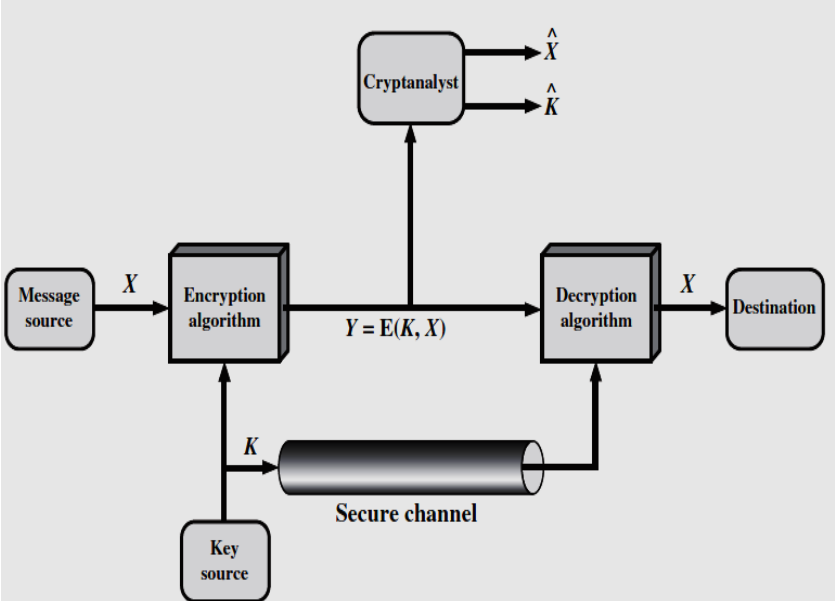


Fig: Model of Symmetric Cryptosystem

§ With the message X and the encryption key K as input, the encryption algorithm forms the ciphertext $Y = [Y_1, Y_2, \ldots, Y_N]$.

$$Y = E(K, X)$$

§ The intended receiver, in possession of the key, is able to invert the transformation:

$$X = D(K, Y)$$

Cryptography

- Cryptographic systems are characterized along three independent dimensions
  - The type of operations used for transforming plaintext to ciphertext
    - Substitution
    - Transposition
  - The number of keys used
    - symmetric, single-key, secret-key, or conventional encryption
    - asymmetric, two-key, or public-key encryption
  - The way in which the plaintext is processed
    - Block cipher
    - Stream cipher

Cryptanalysis and Brute-Force Attack

§ There are two general approaches to attacking a conventional encryption scheme

1. Cryptanalysis

- rely on the nature of the algorithm plus some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs
- exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used

2. Brute-force attack

- The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained

Type of Attack	Known to Cryptanalyst
Ciphertext Only	• Encryption algorithm • Ciphertext
Known Plaintext	• Encryption algorithm • Ciphertext • One or more plaintext–ciphertext pairs formed with the secret key
Chosen Plaintext	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	• Encryption algorithm • Ciphertext • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Text	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

Table 1: Types of attacks on Encrypted Messages

substitution technique

§ 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

Caesar Cipher

§ involves replacing each letter of the alphabet with the letter standing three places further down the alphabet. For example,

§ plain: meet me after the toga party

cipher: PHHW PH DIWHU WKH WRJD SDUWB

0	1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	G	H	I	J	K	L	M

13	14	15	16	17	18	19	20	21	22	23	24	25
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

plain text : 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 cipher text: 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

§ Then 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* takes on a value in the range 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

KEY	PHHW PH DIWHU WKH WRJD SDUWB
1	oggv og chvgt vjg vqic rctva
2	nffu nf bgufs uif uphb qbsuz
3	meet me after the toga party
4	ldds ld zesdq sgd snfz ozqsx
5	kccr kc ydrcp rfc rmey nyprw
6	jbbq jb xcqbo qeb qldx mxoqv
7	iaap ia wbpan pda pkcw lwnpu
8	hzzo hz vaozm ocz ojbv kvmot
9	gyyn gy uznyl nby niau julns
10	fxxm fx tymxk max mhzt itkmr
11	ewwl ew sxlwj lzw lgys hsjlq
12	dvvk dv rwkvi kyv kfxr grikp
13	cuuj cu qvjuh jxu jewq fqhjo
14	btti bt puitg iwt idvp epgin
15	assh as othsf hvs hcuo dofhm
16	zrrg zr nsgre gur gbtn cnegl
17	yqqf yq mrfqd ftq fasm bmdfk
18	xppe xp lqepc esp ezrl alcej
19	wood wo kpdob dro dyqk zkbdi
20	vnnc vn jocna cqn cxpj yjach
21	ummb um inbmz bpm bwoi xizbg
22	tlla tl hmaly aol avnh whyaf
23	skkz sk glzkx znk zumg vgxze
24	rjjy rj fkyjw ymj ytlf ufwyd
25	qiix qi ejxiv xli xske tevxc

Table 2: Brute force cryptanalysis

§ Three important characteristics of this problem enabled us to use a bruteforce cryptanalysis

- The encryption and decryption algorithms are known
- There are only 25 keys to try
- The l

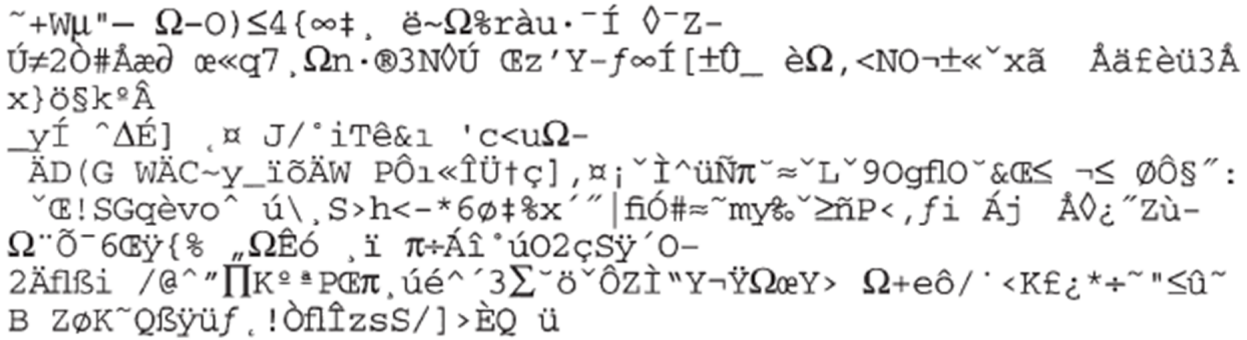


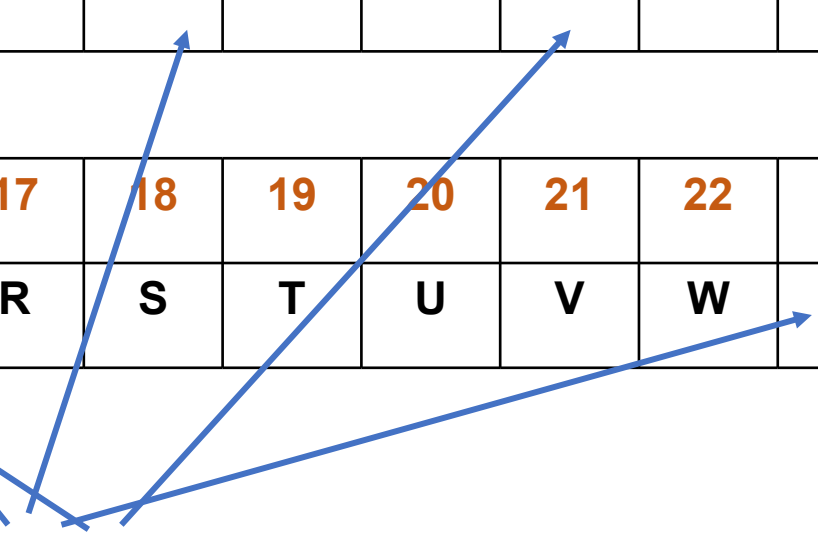
Fig: sample of compressed text

Monoalphabetic cipher

- The “cipher” line can be any permutation of the 26 alphabetic characters,then there are 26! possible keys
- This would seem to eliminate brute-force techniques for cryptanalysis
- single cipher alphabet (mapping from plain alphabet to cipher alphabet) is used per message
- English language- the nature of the plaintext is known

0	1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	G	H	I	J	K	L	M

13	14	15	16	17	18	19	20	21	22	23	24	25
N	O	P	Q	R	S	T	U	V	W	X	Y	Z



Example: Plain Text: MYSURU

cipher text: BFXPIP

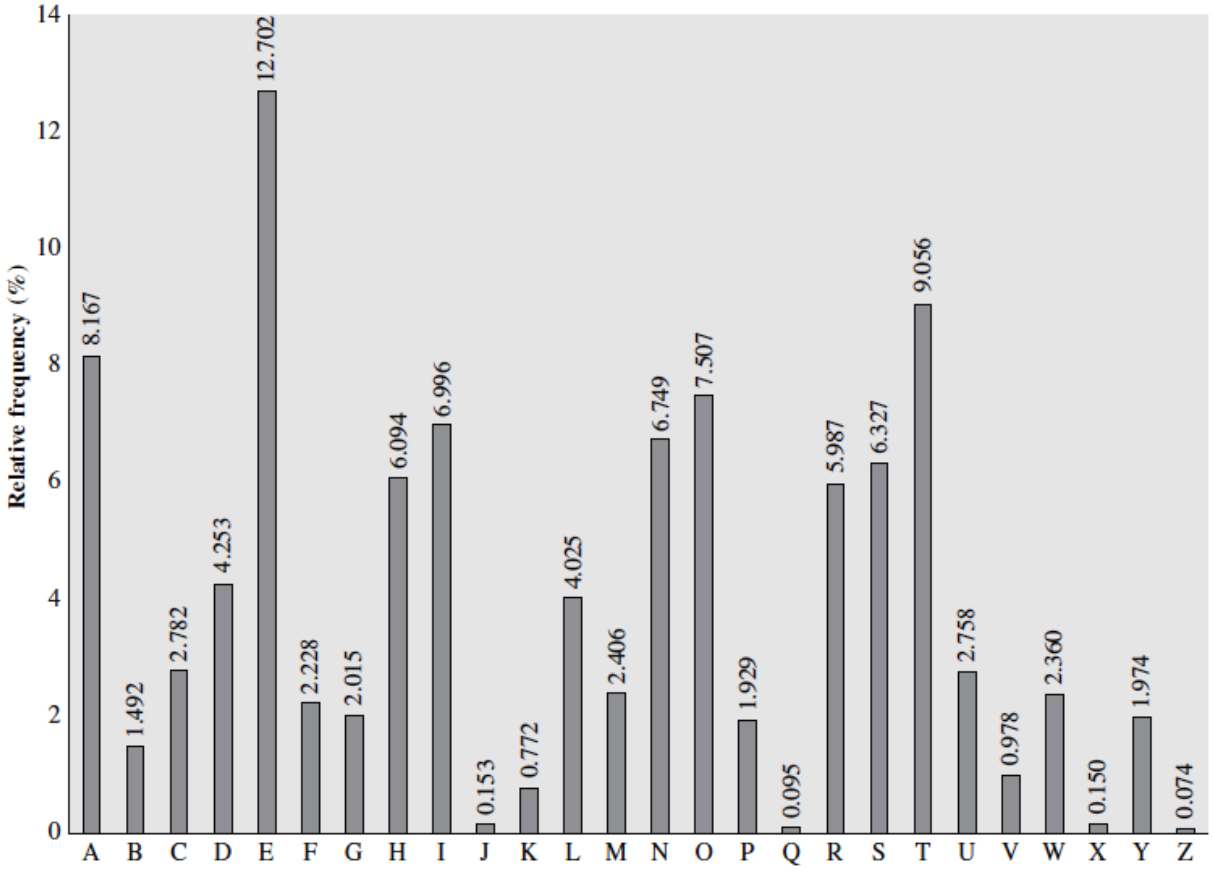


Fig:Relative Frequency of Letters in English Text

Monoalphabetic Cipher example: GZGEWVGRNCP

CT	G	Z	G	E	W	V	G	R	N	C	P
PT	E		E				E				
PT	E		E			T	E				
PT	E		E			T	E			A	
PT	E		E			T	E		L	A	
PT	E		E			T	E		L	A	N
PT	E		E			T	E	P	L	A	N
PT	E	X	E	C	U	T	E	P	L	A	N

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPOUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ

UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
t a e e te a that e e a a
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
e t ta t ha e ee a e th t a
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
e e e tat e the t

it was disclosed yesterday that several informal but direct contacts have been made with political representatives of the viet cong in moscow

Pros and cons

Pros

1. Better security than Caeser cipher

Cons

1. Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphabet
2. Prone to guessing attack using the English letter frequency of occurrence of letters

Playfair Cipher

§ Multiple-letter encryption cipher which treats digrams in the plaintext as single units and translates these units into ciphertext digrams

§ The Playfair algorithm is based on the use of a 5 * 5 matrix of letters constructed using a keyword.

§ For the encryption process let us consider the following example

key: monarchy

Plaintext: instruments

§ **The Playfair Cipher Encryption Algorithm:**

The Algorithm consists of 2 steps:

1. Generate the key Square(5×5):

- The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext.

- The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.

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

2.Algorithm to encrypt the plain text: The plaintext is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter.

For example

PlainText: "instruments"

After Split: 'in' 'st' 'ru' 'me' 'nt' ‘sz’

Rule 1: Pair cannot be made with same letter. Break the letter in single and add a bogus letter to the previous letter.

example : Plain Text: “hello”

After Split: ‘he’ ‘lx’ ‘lo’ --- Here **‘x’** is the bogus letter.

Rule 2: If the letter is standing alone in the process of pairing, then add an extra bogus letter with the alone letter

e.g.,: Plain Text: “helloe”

After Split: ‘he’ ‘lx’ ‘lo’ ‘ez’ -----Here **‘z’** is the bogus letter.

Rule 3: If both the letters are in the same column |↓| wrap around i.e., Take the letter below each one (going back to the top if at the bottom).

e.g.,: Diagraph: "me"

Encrypted Text: cl (m -> c, e -> l)

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

Rule 4: If both the letters are in the same row |→| wrap around i.e., Take the letter to the right of each one (going back to the leftmost if at the rightmost position).

Example : Diagraph: "st“

Encrypted Text: tl(s -> t, t -> l)

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

If neither of the above rules is true: Form a rectangle with the two letters and take the letters on the horizontal opposite corner of the rectangle.

Example: Diagraph: "nt“

Encrypted Text: rq (n -> r, t -> q)

For example

Plain Text: "instrumentsz"

Encrypted Text: gatlmzclrqtx

Encryption

i -> g n -> a s -> t t -> l r -> m u -> z m -> c e -> l

n -> r t -> q s -> t z -> x

Using this Playfair matrix:

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

Encrypt this message: Must see you over Cadogan West. Coming at once.

Hill Cipher

§ Multi-letter cipher

§ Developed by the mathematician Lester Hill in 1929

§ Encrypts group of letters: digraph, trigraph or polygraph

§ Review few terminologies from linear algebra

- matrix arithmetic modulo 26
- Square matrix
- Determinant
- Multiplicative inverse

The Hill Algorithm

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

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

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

Encryption

$$C_1 = (P_1 K_{11} + P_2 K_{21} + P_3 K_{31}) \bmod 26$$

$$C_2 = (P_1 K_{12} + P_2 K_{22} + P_3 K_{32}) \bmod 26$$

$$C_3 = (P_1 K_{13} + P_2 K_{23} + P_3 K_{33}) \bmod 26$$

Example: Encryption

§ Plain text: pay more money

§ Key:

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

P	A	Y	M	O	R	E	M	O	N	E	Y
15	0	24	12	14	17	4	12	14	13	4	24

PT: pay mor emo ney

- Encrypting : **pay**

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

$$(C_1,C_2,C_3)=(15\ 0\ 24)\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \bmod 26$$

$$= (15*17+0*21+24*2 \quad 15*17+0*18+24*2 \quad 15*5+0*21+24*19) \bmod 26$$

$$= (303 \quad 303 \quad 531) \bmod 26$$

$$= (17 \quad 17 \quad 11)$$

$(C_1,C_2,C_3) = (R \quad R \quad L)$

- Encrypting : **mor**

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

$$(C_1,C_2,C_3)=(12\ 14\ 17)\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \bmod 26$$

$$= (12*17+14*21+17*2 \quad 12*17+14*18+17*2 \quad 12*5+14*21+17*19) \bmod 26$$

$$= (532 \quad 490 \quad 677) \bmod 26$$

$$= (12 \quad 22 \quad 1)$$

$(C_1,C_2,C_3) = (M \quad W \quad B)$

- Encrypting : **emo**

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

$$(C_1,C_2,C_3)=(4\ 12\ 14)\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \bmod 26$$

$$= (4*17+12*21+14*2 \quad 4*17+12*18+14*2 \quad 4*5+12*21+14*19) \bmod 26$$

$$= (348 \quad 312 \quad 538) \bmod 26$$

$$= (10 \quad 0 \quad 18)$$

(C_1,C_2,C_3) = (K A S)

- Encrypting : **ney**

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

$$(C_1,C_2,C_3)=(13\ 4\ 24)\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \bmod 26$$

$= (13*17+4*21+24*2 \quad 13*17+4*18+24*2 \quad 13*5+4*21+24*19) \bmod 26$

$= (353 \quad 341 \quad 605) \bmod 26$

$= (15 \quad 3 \quad 7)$

$(C_1,C_2,C_3) = (P \quad D \quad H)$

PT	P	A	Y	M	O	R	E	M	O	N	E	y
CT	R	R	L	M	W	B	K	A	S	P	D	H

Plain text: pay more money

Cipher text: rrlmwbkaspdh

Decryption requires K^{-1}, the inverse matrix K.

$$K^{-1} = \frac{1}{\text{Det } K} \times \text{Adj } K$$

To find Det K, Adj K

To find the determinant of K: $\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$

Det $\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$ mod 26

$= 17(18x19 - 2x21) - 17(19x21 - 2x21) + 5(2x21 - 2x18) \bmod 26$

$= 17(342 - 42) - 17(399 - 42) + 5(42 - 36) \bmod 26$

$= 17(300) - 17(357) + 5(6) \bmod 26$

$= 5100 - 6069 + 30 \bmod 26$

$= -939 \bmod 26$

$= -3 \bmod 26$

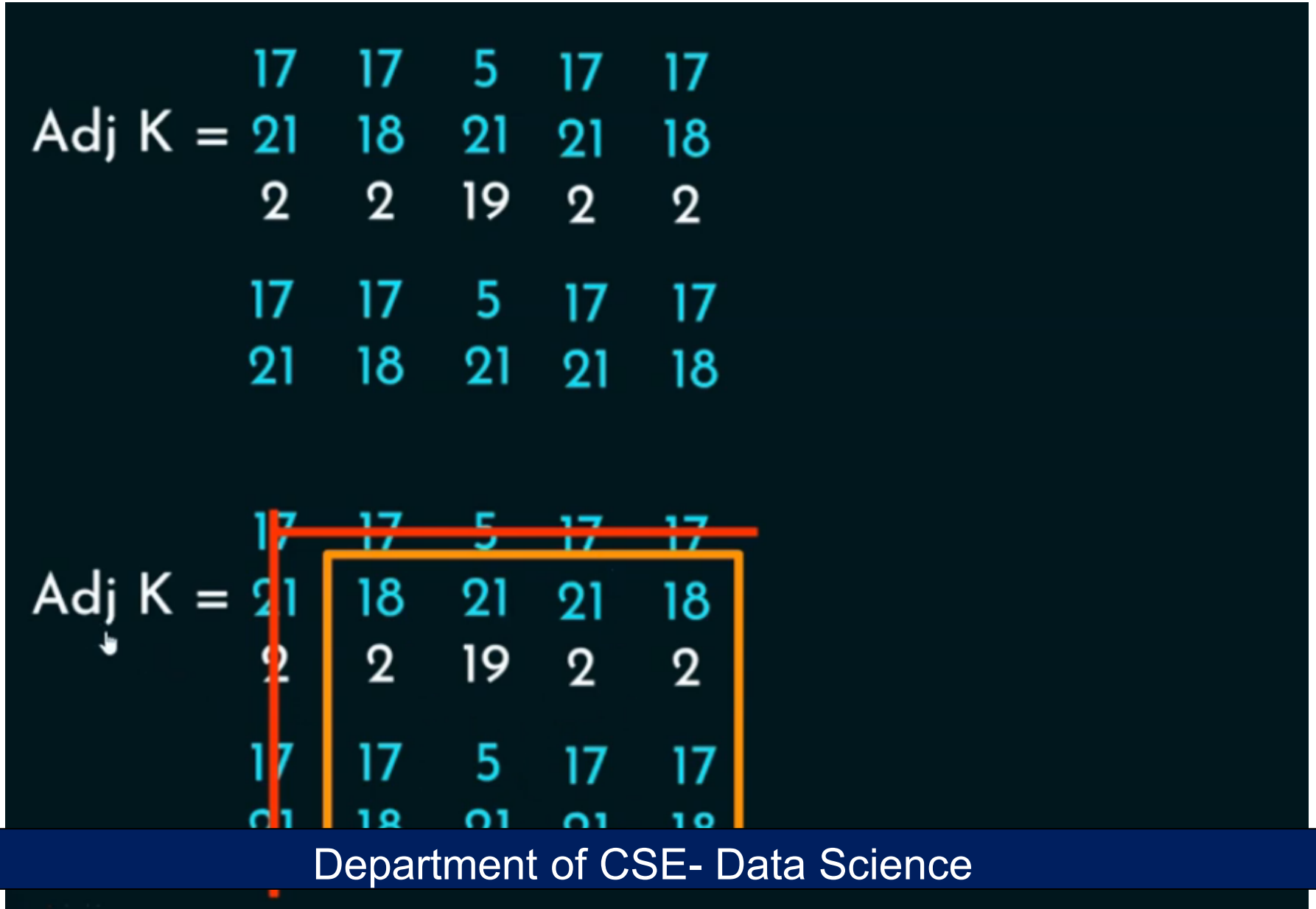
$= 23$

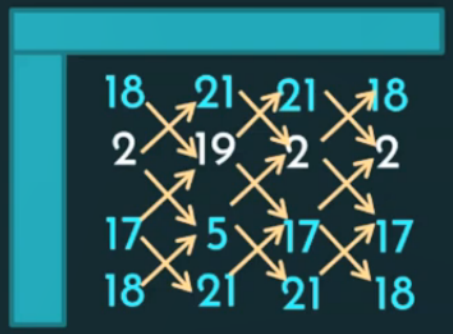
To find Adjoint K

$$\text{Adj K} = \begin{vmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{vmatrix}$$

$$\text{Adj K} = \begin{vmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{vmatrix}$$

$$\text{Adj K} = \begin{matrix} 17 & 17 & 5 & 17 & 17 \\ 21 & 18 & 21 & 21 & 18 \\ 2 & 2 & 19 & 2 & 2 \end{matrix}$$


Adj K =
17 17 5 17 17
21 18 21 21 18
2 2 19 2 2
17 17 5 17 17
21 18 21 21 18
Adj K =
18 21 21 18
2 19 2 2
17 5 17 17



Performing the operation - Column wise

Entering the matrix - Row wise

$$= \begin{matrix} 18\times19-2\times21 & 2\times5-17\times19 & 17\times21-18\times5 \\ 21\times2-19\times21 & 19\times17-5\times2 & 5\times21-21\times17 \\ 21\times2-2\times18 & 2\times17-17\times2 & 17\times18-21\times17 \end{matrix}$$

$$= \begin{matrix} 300 & -313 & 267 \\ -357 & 313 & -252 \\ 6 & 0 & -51 \end{matrix} \bmod 26$$

$$= \begin{pmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{pmatrix} \bmod 26$$

Decryption requires K^{-1}, the inverse matrix K.

$$K^{-1} = \frac{1}{\text{Det } K} \times \text{Adj } K$$

$$K^{-1} = \frac{1}{23} \times \begin{pmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{pmatrix} \bmod 26$$

$$K^{-1} = 23^{-1} \times \begin{pmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{pmatrix} \bmod 26$$

Decryption requires K^{-1}, the inverse matrix K.

$$K^{-1} = \frac{1}{\text{Det } K} \times \text{Adj } K$$

$$K^{-1} = \frac{1}{23} \times \begin{pmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{pmatrix} \bmod 26$$

$$K^{-1} = 23^{-1} \times \begin{pmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{pmatrix} \bmod 26$$

$23^{-1} \times 23 = 1 \bmod 26$	$9 \times 23 = 25 \bmod 26$
$1 \times 23 = 23 \bmod 26$	$10 \times 23 = 22 \bmod 26$
$2 \times 23 = 20 \bmod 26$	$11 \times 23 = 19 \bmod 26$
$3 \times 23 = 17 \bmod 26$	$12 \times 23 = 16 \bmod 26$
$4 \times 23 = 14 \bmod 26$	$13 \times 23 = 13 \bmod 26$
$5 \times 23 = 11 \bmod 26$	$14 \times 23 = 10 \bmod 26$
$6 \times 23 = 8 \bmod 26$	$15 \times 23 = 7 \bmod 26$
$7 \times 23 = 5 \bmod 26$	$16 \times 23 = 4 \bmod 26$
$8 \times 23 = 2 \bmod 26$	$17 \times 23 = 1 \bmod 26$

$$K^{-1} = 17 \times \begin{pmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{pmatrix} \bmod 26$$

$$K^{-1} = \begin{pmatrix} 238 & 425 & 119 \\ 119 & 17 & 136 \\ 102 & 0 & 17 \end{pmatrix} \bmod 26$$

$$K^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

$$K \times K^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$K = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \quad K^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

This is demonstrated as

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} = \begin{pmatrix} 443 & 442 & 442 \\ 858 & 495 & 780 \\ 494 & 52 & 365 \end{pmatrix} \bmod 26$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Decrypt "RRLMWBKASPDH" using Hill cipher with key

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

$P = C\ K^{-1} \bmod 26$

R	R	L	M	W	B	K	A	S	P	D	H
17	17	11	12	22	1	10	0	18	15	3	7

Decrypting: RRL

$$(P_1\ P_2\ P_3) = (R\ R\ L) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

Decryption

$$(C_1\ C_2\ C_3) = (17\ 17\ 14) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$$= (17x4+17x15+11x24 \quad 17x9+17x17+11x0 \quad 17x15+17x6+11x17) \bmod 26$$

$$= (587\ 442\ 544) \bmod 26$$

$$= (15\ 0\ 24)$$

$$= (P\ A\ Y)$$

Decrypting: MWB

$$(P_1\ P_2\ P_3) = (M\ W\ B) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$$(C_1\ C_2\ C_3) = (12\ 22\ 1) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$$= (12x4+22x15+1x24 \quad 12x9+22x17+1x0 \quad 12x15+22x6+1x17) \bmod 26$$

$$= (402\ 482\ 329) \bmod 26$$

$$= (12\ 14\ 17)$$

$$= (M\ O\ R)$$

Decrypting: KAS

$$(P_1\ P_2\ P_3) = (K\ A\ S) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$$(C_1\ C_2\ C_3) = (10\ 0\ 18) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$$= (10\times4+0\times15+18\times24 \quad 10\times9+0\times17+18\times0 \quad 10\times15+0\times6+18\times17) \bmod 26$$

$$= (472\ 90\ 456) \bmod 26$$

$$= (4\ 12\ 14)$$

$$= (E\ M\ O)$$

Decrypting: PDH

$$(P_1\ P_2\ P_3) = (P\ D\ H) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$$(C_1\ C_2\ C_3) = (15\ 3\ 7) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$$= (15\times4+3\times15+7\times24 \quad 15\times9+3\times17+7\times0 \quad 15\times15+3\times6+7\times17) \bmod 26$$

$$= (273\ 186\ 362) \bmod 26$$

$$= (13\ 4\ 24)$$

$$= (N\ E\ Y)$$

CT	R	R	L	M	W	B	K	A	S	P	D	H
PT	p	a	y	m	o	r	e	m	o	n	e	y

Polyalphabetic Cipher

★ To improve on the simple monoalphabetic technique.

★ General name: Polyalphabetic substitution cipher.

Common features

1. A set of related monoalphabetic substitution rules is used.
2. A key determines which particular rule is chosen for a given transformation.

Vigenere Cipher

★ It consists of the 26 Caesar ciphers with shifts of 0 through 25.

Encryption process:

$$C_i = (P_i + K_{i \bmod m}) \bmod 26$$

Decryption process:

$$P_i = (C_i - K_{i \bmod m}) \bmod 26$$

Example:

Plaintext: we are discovered save yourself

Key: deceptive

Vigenere Cipher

Key : deceptivedeceptivedeceptive

Plaintext : wearediscoveredsaveyourself

Ciphertext : ZICVTWQNGRZGVTWAVZHCQYGLMGJ

Key	3	4	2	4	15	19	8	21	4	3	4	2	4
PT	22	4	0	17	4	3	8	18	2	14	21	4	17
CT	25	8	2	21	19	22	16	13	6	17	25	6	21

Key	15	19	8	21	4	3	4	2	4	15	19	8	21	4
PT	4	3	18	0	21	4	24	14	20	17	18	4	11	5
CT	19	22	0	21	25	7	2	16	24	6	11	12	6	9

§ The strength of this cipher is that there are multiple ciphertext letters for each plaintext letter, one for each unique letter of the keyword. Thus, the letter frequency information is obscured.

Cryptanalysis

§ Determining the length of the keyword

§ Key and the plaintext share the same frequency distribution of letters, a statistical techniques can be applied

Autokey system

★ The periodic nature of the keyword can be eliminated by using a non-repeating keyword that is as long as the message itself.

★ Vigenère proposed autokey system, in which a keyword is concatenated with the plaintext itself to provide a running key.

Example

Key : deceptivewearediscoveredsav

Plaintext : wearediscoveredsaveyourself

Ciphertext : ZICVTWQNGKZEIIGASXSTSLVVWLA

One – Time Pad

- Random key that is as long as the message
- The key need not be repeated
- In addition, the key is to be used to encrypt and decrypt a single message and then is discarded
- Each new message requires a new key of the same length as the new message
- Such a scheme, known a one-time pad, is unbreakable.
- No statistical relationship to the plain text
- Because the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code

Example

§ Consider the ciphertext

ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

§ We now show two different decryptions using two different keys:

ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

key: pxlmvmsydofuyrvzwc tnlebnecvgdupahfzzlmnyih

plaintext mr mustard with the candlestick in the hall

ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS

key : pftgpmiydgaxgoufhkllmhsqdqogtewbqfgyovuhwt

plaintext: miss scarlet with the knife in the library

- Suppose that a cryptanalyst had managed to find these two keys.
- Two possible plaintexts are produced. How is the cryptanalyst to decide which is the correct

  decryption (i.e., which is the correct key)?
- If the actual key were produced in a truly random fashion, then the cryptanalyst cannot say that one of these two keys is more likely than the other.
- Thus, there is no way to decide which key is correct and therefore which plaintext is correct.
- In fact, given any plaintext of equal length to the ciphertext, there is a key that produces that plaintext. Therefore, if you did an exhaustive search of all possible keys, you would end up with many legible plaintexts, with no way of knowing which was the intended plaintext.
-

- The security of the one-time pad is entirely due to the randomness of the key.

Two fundamental difficulties

§ The practical problem of making large quantities of random keys

§ Even more daunting is the problem of key distribution and protection

§ Because of these difficulties, the one-time pad is of limited utility and is useful primarily for low-bandwidth channels requiring very high security

Perfect secrecy

- The one-time pad is the only cryptosystem that exhibits what referred to as perfect secrecy
- perfect secrecy is the notion that , given an encrypted message (or ciphertext) from a perfectly secure encryption system(or cipher), absolutely nothing will be revealed about the unencrypted message(or plaintext) by the cipherext.

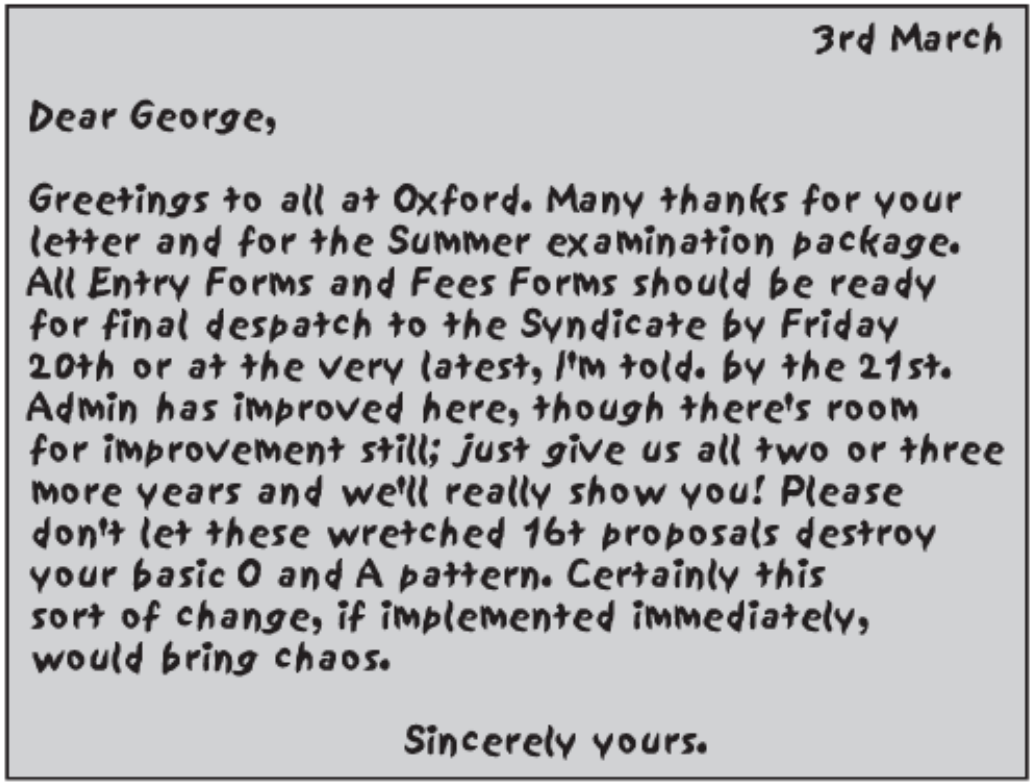
Steganography

§ Steganography is the practice of concealing a message within another message or physical object in a way that the hidden message is not obvious to an observer.

§ It differs from cryptography, which aims to make a message unreadable, as steganography focuses on concealing the very existence of the message itself

§ A simple form of steganography, but one that is time-consuming to con struct, is one in which an arrangement of words or letters within an apparently innocuous text spells out the real message.

§ For example, the sequence of first letters of each word of the overall



3rd March

Dear George,

Greetings to all at Oxford. Many thanks for your letter and for the Summer examination package. All Entry Forms and Fees Forms should be ready for final despatch to the Syndicate by Friday 20th or at the very latest, I'm told. by the 21st. Admin has improved here, though there's room for improvement still; just give us all two or three more years and we'll really show you! Please don't let these wretched 16+ proposals destroy your basic O and A pattern. Certainly this sort of change, if implemented immediately, would bring chaos.

Sincerely yours.

Figure : A Puzzle for Inspector Morse

a subset of the words of the overall message is used to convey the hidden message. See if you can decipher this; it's not too hard.

§ Various other techniques have been used historically; some examples are the following

- Character marking: Selected letters of printed or typewritten text are over written in pencil. The marks are ordinarily not visible unless the paper is held at an angle to bright light.
- Invisible ink: A number of substances can be used for writing but leave no visible trace until heat or some chemical is applied to the paper.
- Pin punctures: Small pin punctures on selected letters are ordinarily not visible unless the paper is held up in front of a light.
- Typewriter correction ribbon: Used between lines typed with a black ribbon, the results of typing with the correction tape are visible only under a strong light

- Steganography has a number of drawbacks when compared to encryption. It requires a lot of overhead to hide a relatively few bits of information
- Alternatively, a message can be first encrypted and then hidden using steganography.
- The advantage of steganography is that it can be employed by parties who have something to lose should the fact of their secret communication (not necessar ily the content) be discovered.
- Encryption flags traffic as important or secret or may identify the sender or receiver as someone with something to hide.

Block Ciphers and The Data Encryption Standard

Topics covered

- Traditional block Cipher structure
- The data encryption standard
- A DES example
- The strength of DES
- Block cipher design principles

Traditional block Cipher structure

Stream Ciphers and Block Ciphers

§ stream ciphers process messages a bit or byte at a time when en/decrypting

§ Ex: autokey

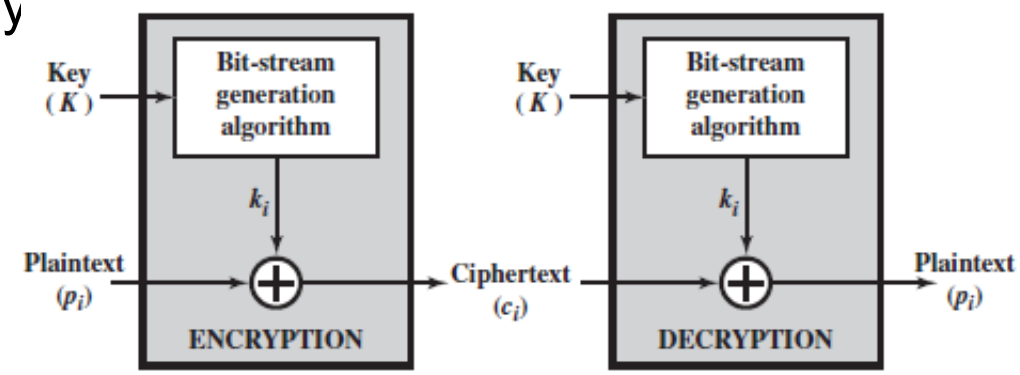


Fig: Steam cipher using algorithmic bit stream generator

§ block ciphers process messages in blocks, each of which is then en/decrypted

§ Typically

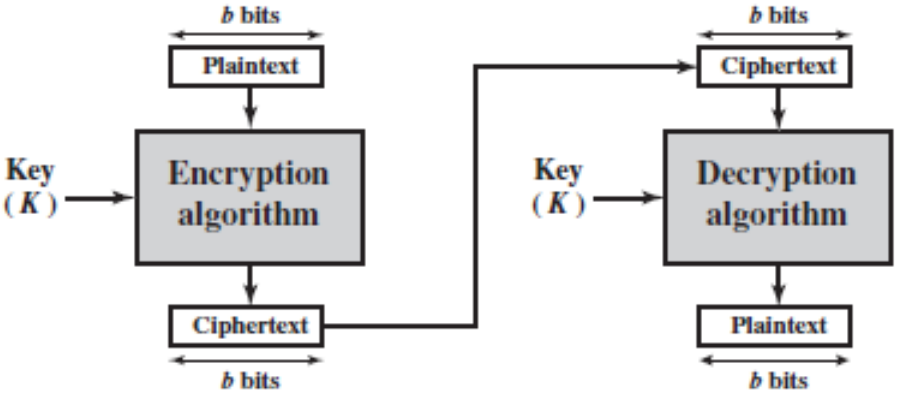


Fig: Block cipher

§ many current ciphers are block ciphers

Motivation for the Feistel Cipher Structure

§ Most symmetric block ciphers are based on a Feistel Cipher Structure

§ A block cipher operates on a plaintext block of *n* bits to produce a ciphertext block of *n* bits.

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

§ The follo ular transformations for *n* = 2.

Reversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	00
11	01

Irreversible Mapping

Plaintext	Ciphertext
00	11
01	10
10	01
11	01

⟸ a ciphertext of 01 could have been produced by one of two plaintext blocks

- So if we limit ourselves to reversible mappings, the number of different transformations is 2^n

Plaintext	Ciphertext
0000	1110
0001	0100
0010	1101
0011	0001
0100	0010
0101	1111
0110	1011
0111	1000
1000	0011
1001	1010
1010	0110
1011	1100
1100	0101
1101	1001
1110	0000
1111	0111

Ciphertext	Plaintext
0000	1110
0001	0011
0010	0100
0011	1000
0100	0001
0101	1100
0110	1010
0111	1111
1000	0111
1001	1101
1010	1001
1011	0110
1100	1011
1101	0010
1110	0000
1111	0101

Table: Encryption and Decryption Tables for Substitution Cipher for n=4

§ Feistel refers to this as the *ideal block cipher*, because it allows for the maximum number of possible encryption mappings from the plaintext block

The Feistel Cipher

§ Feistel proposed that we can approximate the ideal block cipher by utilizing the concept of a product cipher, which is the execution of two or more simple ciphers in sequence in such a way that the final result or product is cryptographically stronger than any of the component ciphers.

§ Develop a block cipher with a key length of *k* bits and a block length of *n* bits, allowing a total of 2^k possible transformations, rather than the 2^n transformations available with the ideal block cipher.

§ Feistel proposed the use of a cipher that alternates substitutions and permutations

- **Substitution:** Each plaintext element or group of elements is uniquely replaced by a corresponding ciphertext element or group of elements.
- **Permutation:** A sequence of plaintext elements is replaced by a permutation of that sequence. That is, no elements are added or deleted or replaced in the sequence,

§ Claude Shannon introduced idea of substitution-permutation (S-P) networks in 1949 paper

§ form basis of modern block ciphers

§ S-P nets are based on the two primitive cryptographic operations seen before:

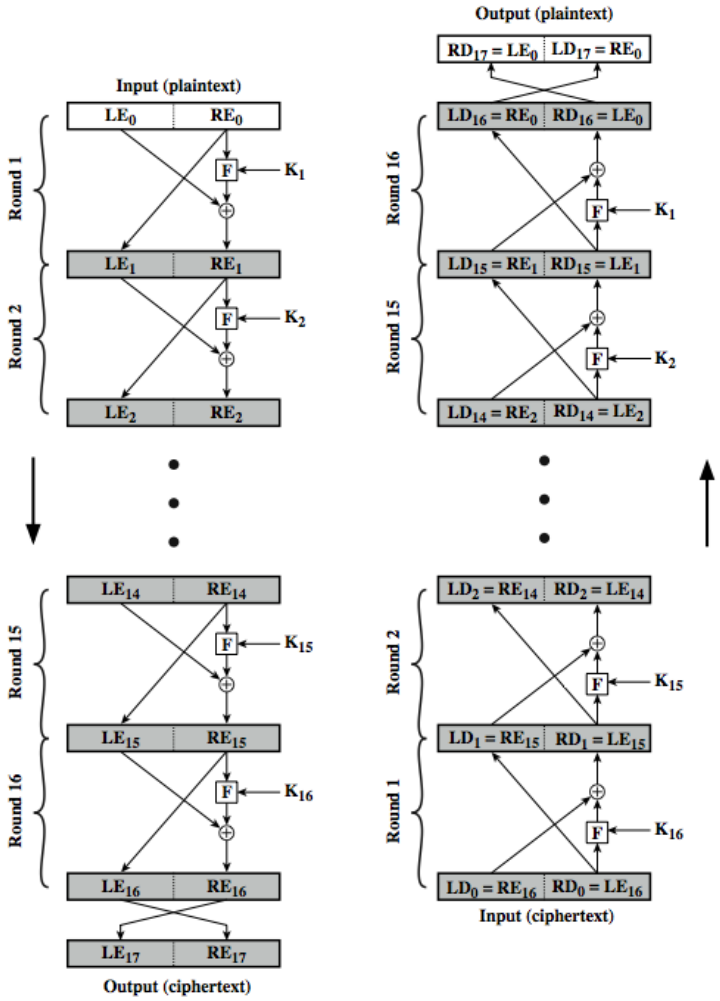
1. substitution (S-box)
2. permutation (P-box)

§ provide confusion & diffusion of message & key

§ more practically Shannon suggested combining S & P elements to obtain:

- **diffusion** – dissipates statistical structure of plaintext over bulk of ciphertext
- **confusion** – makes relationship between ciphertext and key as complex as

Feistel Cipher Structure



§ 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, $L0$ 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 L_{i-1} and R_{i-1} derived from the previous round, as well as a subkey K_i derived from the overall K.

§ In general, the subkeys K_i are different from K and from each other.

- All rounds have the same structure.
- A **substitution** is performed on the left half of the data by applying a *round function* F to the right half of the data and then taking the exclusive-OR of the

§ Following this substitution, a **permutation** is performed that consists of the interchange of the two halves of the data.

Feistel Decryption Algorithm

§ same as the encryption process. The rule is as follows:

- Use the ciphertext as input to the algorithm, but use the subkeys K_i in reverse order.
- That is, use K_n in the first round, K_{n-1} in the second round, and so on, until K_1 is used in the last round.

§ Example: Suppose that the blocks at each stage are 32 bits (two 16-bit halves) and that the key size is 24 bits. Suppose that at the end of encryption round fourteen, the value of the intermediate block (in hexadecimal) is DE7F03A6. Then

- *LE*14 = DE7F and *RE*14 = 03A6. Also assume that the value of K_{15} is 12DE52.
- After round 15, we have LE_{15} = 03A6 and RE_{15} = F(03A6, 12DE52) DE7F.
- Now let's look at the decryption. We assume that *LD*1 = *RE*15 and *RD*1 = *LE*15, as shown in Figure and we want to demonstrate that *LD*2 = *RE*14 and *RD*2 = *LE*14.

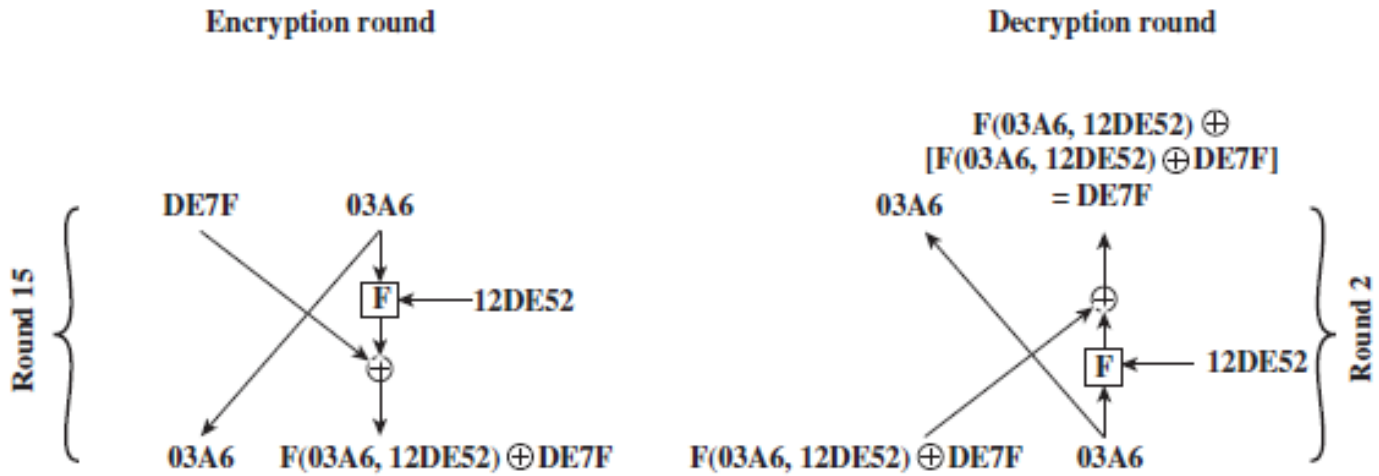


Fig: Feistel example

Feistel Cipher Design Elements

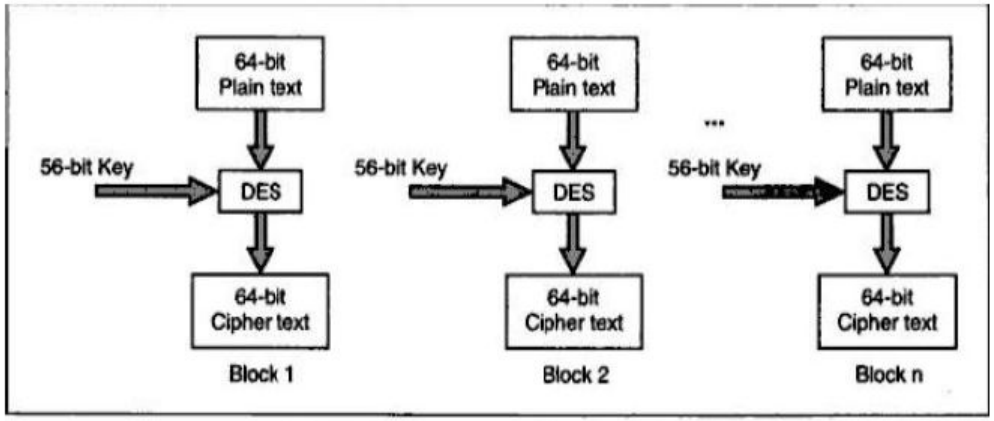
- **Block size:** Larger block sizes mean greater security but reduced encryption/decryption speed for a given algorithm. Traditionally, a block size of 64 bits has been considered a reasonable tradeoff
- **Key size:** Larger key size means greater security but may decrease encryption/ decryption speed. Key sizes of 64 bits or less are now widely considered to be inadequate, and 128 bits has become a common size.
- **Number of rounds:** The essence of the Feistel cipher is that a single round offers inadequate security but that multiple rounds offer increasing security. A typical size is 16 rounds.
- **Subkey generation algorithm:** Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.
- [illegible] cryptanalysis

§ There are two other considerations in the design of a Feistel cipher:

- **Fast software encryption/decryption:** In many cases, encryption is embedded in applications or utility functions in such a way as to preclude a hardware implementation.
- **Ease of analysis:** if the algorithm can be concisely and clearly explained, it is easier to analyze that algorithm for cryptanalytic vulnerabilities and therefore develop a higher level of assurance as to its strength.

Data Encryption Standard(DES)

- Is landmark in cryptographic algorithms
- adopted in 1977 by NBS (now NIST)
- Based on Feistel structure
- Symmetric cipher algorithm and use block cipher method for encryption and decryption



DES Encryption

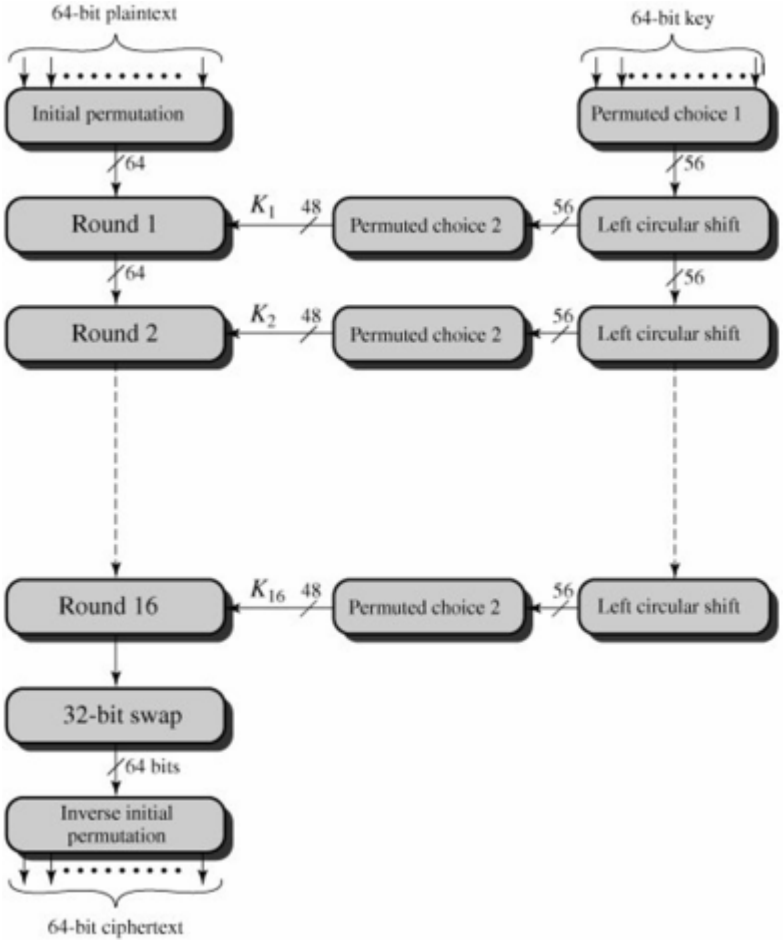


Fig: General depiction of DES Encryption Algorithm

- First, the 64-bit plaintext passes through an initial permutation (IP) that rearranges the bits to produce the *permuted input*.
- This is followed by a phase consisting of sixteen rounds of the same function, which involves both permutation and substitution functions.
- The output of the last (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 **preoutput**.
- Finally, the preoutput is passed through a permutation [IP^{-1}] that is the inverse of the initial permutation function, to produce the 64-bit ciphertext

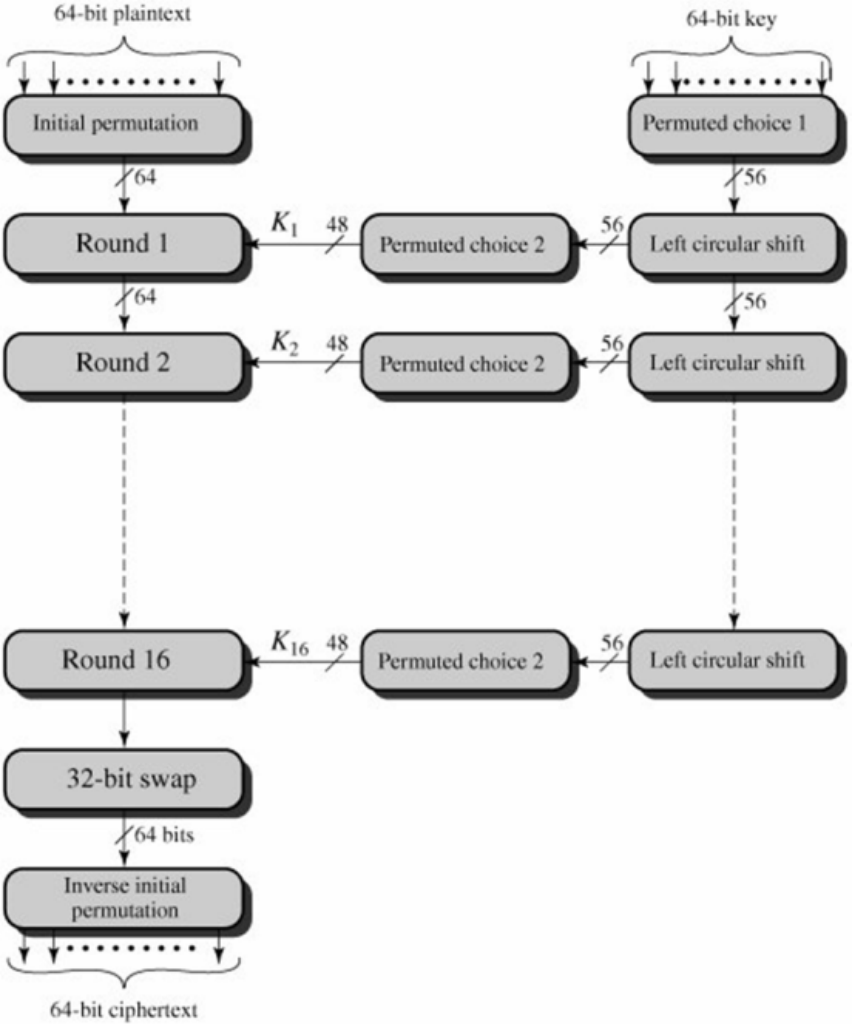
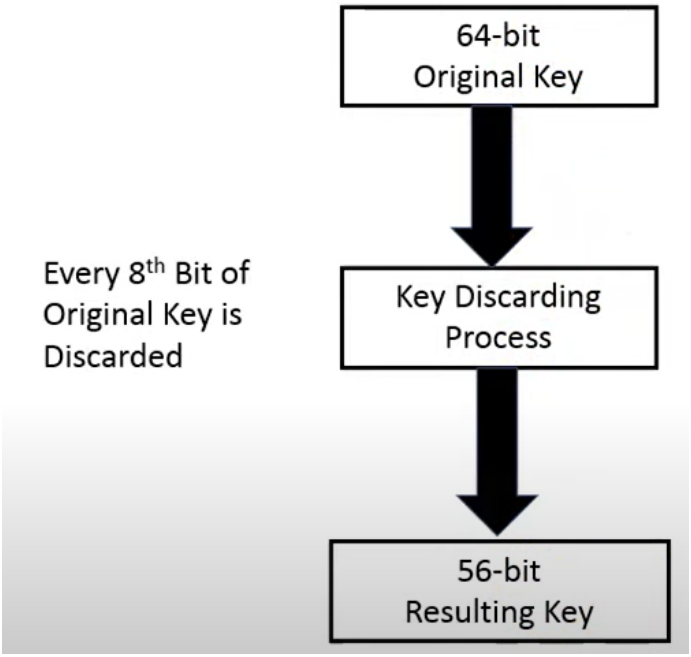


Fig: General depiction of DES Encryption Algorithm

- The right-hand portion of Figure shows the way in which the 56-bit key is used.
- Initially, the key is passed through a permutation function.
- Then, for each of the sixteen rounds, a *subkey* (K_i) is produced by the combination of a left circular shift and a permutation.
- The permutation function is the same for each round, but a different subkey is produced because of the repeated shifts of the key bits.

Key discarding process

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



Example:

1	2	21	38	58	15	37	26
22	55	44	3	53	27	11	60
49	28	14	42	61	48	63	41
18	39	56	10	64	16	62	8
45	40	20	54	4	33	34	52
7	30	47	59	32	5	35	25
29	12	13	6	24	46	57	36
17	23	50	31	43	51	9	19

1	2	21	38	58	15	37
22	55	44	3	53	27	11
49	28	14	42	61	48	63
18	39	56	10	64	16	62
45	40	20	54	4	33	34
7	30	47	59	32	5	35
29	12	13	6	24	46	57
17	23	50	31	43	51	9

Key transformation

§ initially 64-bit key is transformed into a 56-bit key by discarding every 8th bit of the initial key.

§ From this 56-bit key, a different 48-bit Sub Key is generated during each round using a process called **key transformation**.

§ For this, the 56-bit key is divided into two halves, each of 28 bits.

§ These halves are circularly shifted left by one or two positions, depending on the round.

§ For

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
#key bits shifted	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

position

for o

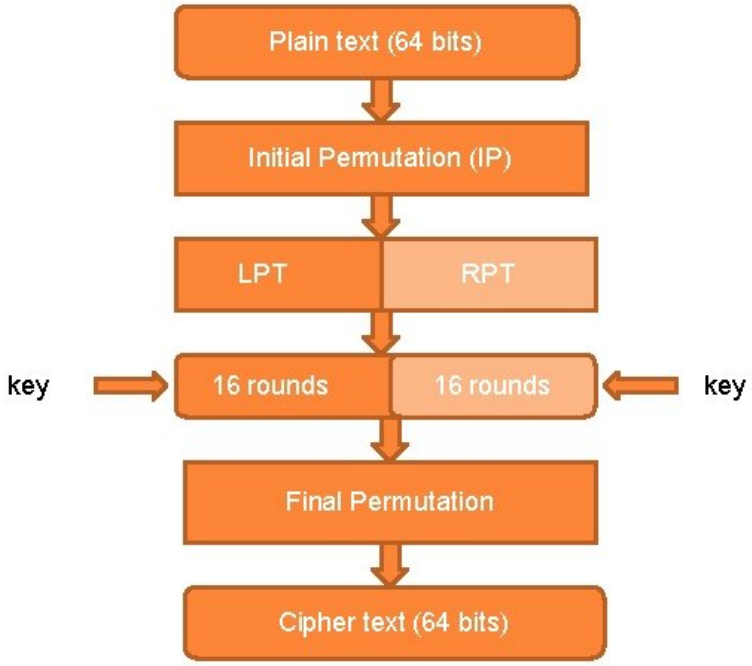
Fig: number of key bits shifted per round

§ After an appropriate shift, 48 of the 56 bit are selected

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Figure - compression permutation

§ Since the key transformation process involves permutation as well as a selection of a 48-bit subset of the original 56-bit key it is called **Compression Permutation.**



1. In the first step, the 64-bit plaintext block is handed over to an initial Permutation (IP) function.
2. The initial permutation is performed on plain text.
3. Next, the initial permutation (IP) produces two halves of the permuted block; Left Plain Text (LPT) and Right Plain Text (RPT).
4. Now each LPT and RPT the go through 16 rounds of encryption process.
5. In the end, LPT and RPT are rejoined and a Final Permutation (FP) is performed on the combined block
6. The result of this process produces 64 bit cipher

Initial Permutation (IP)

- The initial permutation (IP) happens only once
- Bit sequence changed as per IP table

ex: 1st bit take 40th position

58th bit take the 1st position

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: Initial permutation table

DES Round Structure

§ uses two 32-bit L & R halves

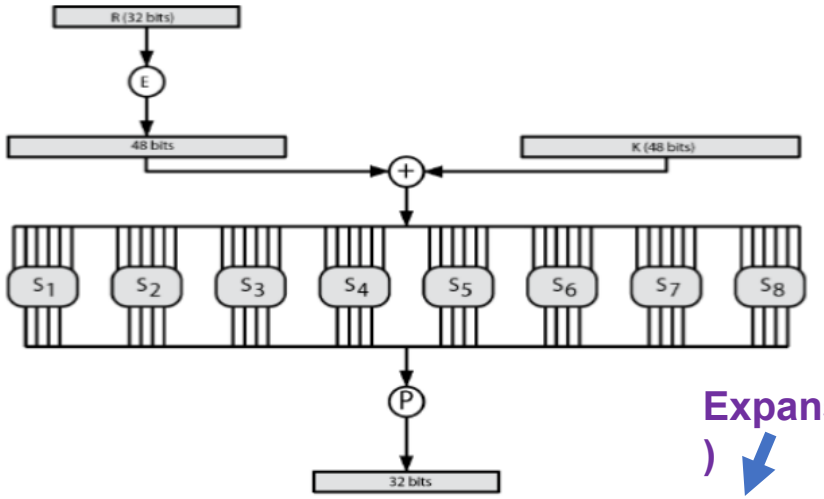
§ as for any Feistel cipher can describe as:

$L_i = R_{i-1}$

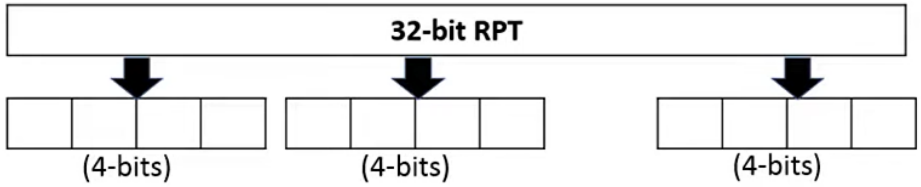
$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

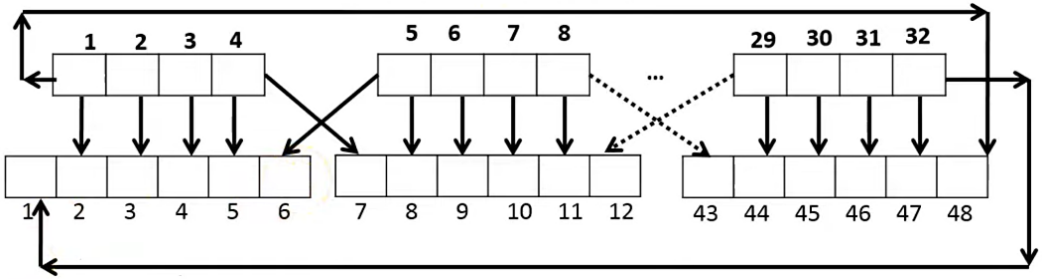
§ F takes 32-bit R half and 48-bit subkey:

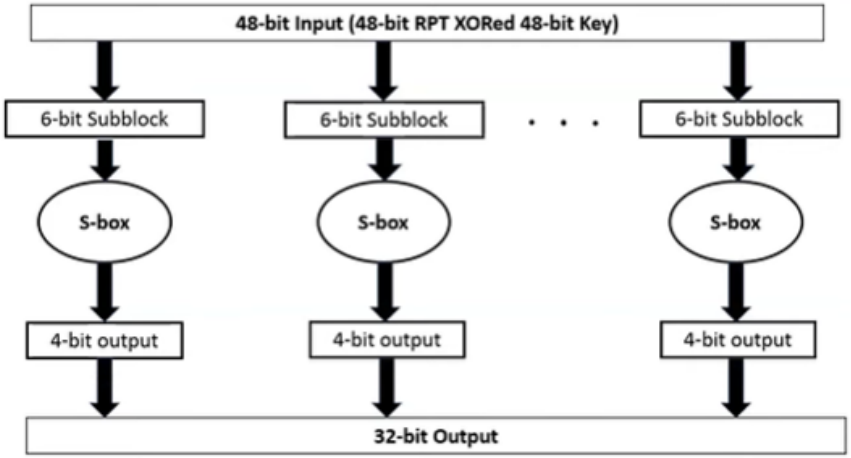
- expands R to 48-bits using perm E
- adds to subkey using XOR
- passes through 8 S-boxes to get 32-bit result

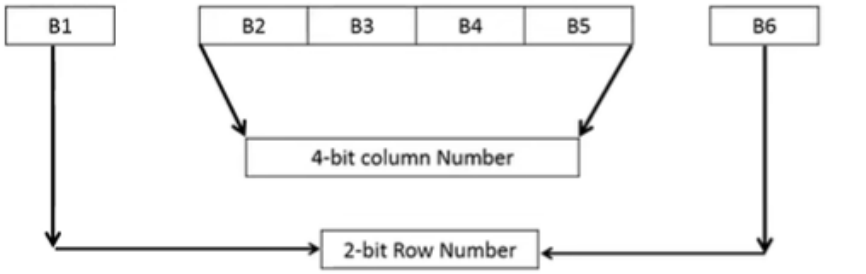


Expansion(E)



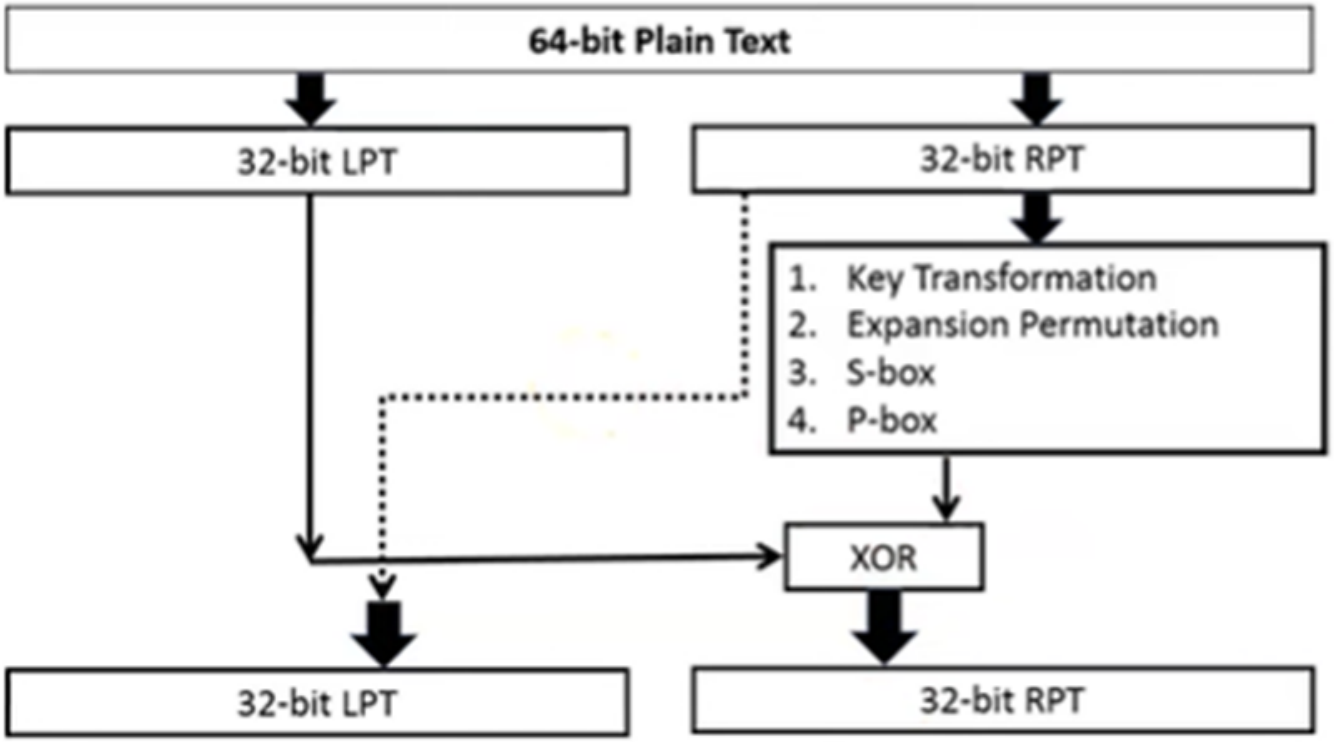






S_5		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

Example: 011011 → 1001



- 1st round of encryption is completed. Now remaining 15 rounds will be performed same as 1st round.

Final Permutation

- At the end of the 16 rounds, the final permutation is performed (only once).

- **For Example:**
  - ✓ 40th bit of input take 1st Position as per below permutation table.

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

- The output of the final permutation is *the 64-bit encrypted block (64-bit cipher text block).*

DES Decryption

- With Feistel design, do encryption steps again using subkeys in reverse order (K_{16} … K_1)
- IP undoes final FP step of encryption
- 1^{st} round with K_{16} undoes 16^{th} encrypt round
- 16^{th} round with K_1 undoes 1^{st} encrypt round
- final FP undoes initial encryption IP
- thus recovering original data value

DES Example

- The plaintext,key, and resulting ciphertext are as follows:

  Plaintext: **02468aceeca86420**

  Key: **0f1571c947d9e859**

  Ciphertext: **da02ce3a89ec**

Round	K_i	L_i	R_i
IP		5a005a00	3cf03c0f
1	1e030f03080d2930	3cf03c0f	bad22845
2	0a31293432242318	bad22845	99e9b723
3	23072318201d0c1d	99e9b723	0bae3b9e
4	05261d3824311a20	0bae3b9e	42415649
5	3325340136002c25	42415649	18b3fa41
6	123a2d0d04262a1c	18b3fa41	9616fe23
7	021f120b1c130611	9616fe23	67117cf2
8	1c10372a2832002b	67117cf2	c11bfc09
9	04292a380c341f03	c11bfc09	887fbc6c
10	2703212607280403	887fbc6c	600f7e8b
11	2826390c31261504	600f7e8b	f596506e
12	12071c241a0a0f08	f596506e	738538b8
13	300935393c0d100b	738538b8	c6a62c4e
14	311e09231321182a	c6a62c4e	56b0bd75
15	283d3e0227072528	56b0bd75	75e8fd8f
16	2921080b13143025	75e8fd8f	25896490
IP^{-1}		da02ce3a	89ecac3b

Avalanche effect

- A small change in plain-text or key should create a significant change in the cipher-text.
- DES has been proved to be strong with regard to this property
- Ex:

plaintext: 0000000000000000

ciphertext:4789FD476E82A5F1

Key: 22234512987ABB23

plaintext: 0000000000000001

ciphertext: 048FD5C15A63F5F2

Plaintext: 02468aceeca86420

Table 3.3 Avalanche Effect in DES: Change in Plaintext

Round		δ
	02468aceeca86420 12468aceeca86420	1
1	3cf03c0fbad22845 3cf03c0fbad32845	1
2	bad2284599e9b723 bad3284539a9b7a3	5
3	99e9b7230bae3b9e 39a9b7a3171cb8b3	18
4	0bae3b9e42415649 171cb8b3ccaca55e	34
5	4241564918b3fa41 ccaca55ed16c3653	37
6	18b3fa419616fe23 d16c3653cf402c68	33
7	9616fe2367117cf2 cf402c682b2cefbc	32
8	67117cf2c11bfc09 2b2cefbc99f91153	33
9	c11bfc09887fbc6c 99f911532eed7d94	32
10	887fbc6c600f7e8b 2eed7d94d0f23094	34
11	600f7e8bf596506e d0f23094455da9c4	37
12	f596506e738538b8 455da9c47f6e3cf3	31
13	738538b8c6a62c4e 7f6e3cf34bc1a8d9	29
14	c6a62c4e56b0bd75 4bc1a8d91e07d409	33
15	56b0bd7575e8fd8f 1e07d4091ce2e6dc	31
16	75e8fd8f25896490 1ce2e6dc365e5f59	32
IP^{-1}	da02ce3a89ecac3b 057cde97d7683f2a	32

64-bit intermediate values at the end of each round

Number of bits that differ between the two intermediate values

Original key: 0f1571c947d9e859
Altered key: 1f1571c947d9e859

Table 3.4 Avalanche Effect in DES: Change in Key

Round		δ
	02468aceeca86420 02468aceeca86420	0
1	3cf03c0fbad22845 3cf03c0f9ad628c5	3
2	bad2284599e9b723 9ad628c59939136b	11
3	99e9b7230bae3b9e 9939136b768067b7	25
4	0bae3b9e42415649 768067b75a8807c5	29
5	4241564918b3fa41 5a8807c5488dbe94	26
6	18b3fa419616fe23 488dbe94aba7fe53	26
7	9616fe2367117cf2 aba7fe53177d21e4	27
8	67117cf2c11bfc09 177d21e4548f1de4	32

Round		δ
9	c11bfc09887fbc6c 548f1de471f64dfd	34
10	887fbc6c600f7e8b 71f64dfd4279876c	36
11	600f7e8bf596506e 4279876c399fdc0d	32
12	f596506e738538b8 399fdc0d6d208dbb	28
13	738538b8c6a62c4e 6d208dbbb9bdeeaa	33
14	c6a62c4e56b0bd75 b9bdeeaad2c3a56f	30
15	56b0bd7575e8fd8f d2c3a56f2765c1fb	33
16	75e8fd8f25896490 2765c1fb01263dc4	30
IP^{-1}	da02ce3a89ecac3b ee92b50606b62b0b	30

Strength of DES

1. Key size
2. Nature of algorithm

Key size

§ 56-bit keys have $2^{56} = 7.2 \times 10^{16}$ values

§ brute force search looks hard

§ DES finally and definitively proved insecure in July 1998, when the Electronic Frontier Foundation (EFF) announced that it had broken a DES encryption using a special-purpose "DES cracker" machine that was built for less than $250,000.

§ The attack took less than three days.

Nature of algorithm

§ possibility that cryptanalysis is possible by exploiting the characteristics of the DES algorithm.

§ The focus of concern has been on the eight substitution tables or S-boxes, that are used in each iteration

§ Because the design criteria for these boxes, and indeed for the entire algorithm, were not made public, there is a suspicion that the boxes were constructed in such a way that cryptanalysis is possible for an opponent who knows the weaknesses in the S-boxes.

§ no one has so far succeeded in discovering the supposed fatal weaknesses in the S-boxes.

Timing Attacks

§ a timing attack is one in which information about the key or the plaintext is obtained by observing how long it takes a given implementation to perform decryptions on various ciphertexts.

§ A timing attack exploits the fact that an encryption or decryption algorithm often takes slightly different amounts of time on different inputs.

- DES appears to be fairly resistant to a successful timing attack

Block Cipher Design Principles

§ Three critical aspects of block cipher design:

1. The number of rounds

- The greater the number of rounds, the more difficult it is to perform cryptanalysis, even for a relatively weak F
- The number of rounds is chosen so that known cryptanalytic efforts require greater effort than a simple brute-force key search attack

2. Design of the function F

- The heart of a Feistel block cipher is the function F, which provides the element of confusion in a Feistel cipher. Thus, it must be difficult to “unscramble” the substitution performed by F.
- One obvious criterion is that F be nonlinear. the more difficult it is to approximate F by a set of linear equations, the more nonlinear F is.

3. Key scheduling

- select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key.