CHAPTER 2

UNIT II

Data Encryption Techniques and Standards

Syllabus :

At the end of this unit, you should be able to understand and comprehend the following syllabus topics:

- Introduction
- Cryptography
  - Substitution Ciphers
  - Transposition Ciphers
  - Stenography applications and limitations
- Encryption Methods
  - Symmetric
  - Asymmetric
- Data Encryption Standard (DES)
  - DES Design Criteria and Feistel Cipher
  - Block Ciphers and methods of operations
  - Weak Keys in DES Algorithms
  - Triple DES
- Advanced Encryption Standard (AES)

2.1 Concept Building – Information Secrecy

- Consider a scenario. You and your friend are in different cities and are chatting over phone using an application. You are discussing a new idea to found a startup and have some cool plans that can change the world and make the business very profitable.
- But wait, I have two questions for you –
  1. How do you know that your conversation (you and your friend sending and receiving text from each other) is not available for anyone else to read? [Goal - Information not available to everyone].
  2. Don't you think that your business plan is confidential and the conversation between you and your friend should somehow remain readable only to you both? [Goal - Information available to intended entities only]
- If you realize the importance of protecting the digital confidential information, you so much wish that there was a way in which you could ensure that the information is available only to you both – blink your eyes and wish granted – Welcome to the world of Cryptography!!
- The dictionary meaning of cryptography is "secret writing".

- Digitally speaking,

> ***Definition :*** *Cryptography is a science and a method of storing and transmitting information in a form that only those it is intended for can read and process its.*

- In a nutshell, the information is available in a readable form only to those who are authorised.
- Let's take a different example to understand a related concept. You might have used coupon codes in online shopping. Based on the offers running on the website, the coupon code (a set of characters) discounts the price on your chosen items. You apply the coupon code to your cart and based on various terms and conditions, the discount amount is calculated. Sometimes, you get a flat off and sometimes a percentage of the cart value.
- In this scenario, we have two interesting concepts –
  - The coupon code (a unique set of alphabets and digits).
  - The coupon code processing terms and conditions (the algorithm (rules) that determines how to apply the coupon code and how much discount to actually give you).
- In a different example, you and your friend might have some words (or codes) that are only understood by you both. When you use that word, your friend knows what exactly that really means even if you say it loud and clear and others hear it. For example, suppose you both have decided that when you say, "I eat banana", that would actually mean "Let's bunk college today". Now, when you say it, your friend gets the real meaning whereas everyone else thinks that you were referring to a fruit.
- So, an important concept to understand here is that when the real information is hidden within what is being communicated, the communicated information can be stored or transmitted without disclosing the actual meaning and we don't care if someone gets the communicated information because the real information is hidden. So, real information can freely move around hiding within the communicated information.

2.2 Introduction to Cryptography

- Now that you understand some of the scenarios around information secrecy, let's define some the basic terms that we would be using throughout the chapter.

1. Goal of cryptography

As you understand, the core goal of cryptography is to hide confidential information. So, cryptography majorly provides Confidentiality out of the CIA triad (Confidentiality, Integrity and Availability).

2. Information / Data

This is the asset that is being protected (provide confidentiality using cryptography in this case). Recall from chapter 1 that an asset is something that has value and is worth protecting. The sender ensures that the information is only readable by the intended receiver even if it is captured / seen by anyone else.

3. Unique key

This is a set of numbers (like your coupon code) that helps to make the information secret. It is like your usual key that helps to lock and unlock the door. Here like various bikes could have the same locking mechanism, a key is unique to a bike even if the bikes are of same model.

4. Algorithm

Algorithm is a process or set of rules to be followed to make the information secret. There are various algorithms (like various types of locks) available that make the information secret in different ways using the keys. In cryptography, such algorithms are also called ciphers.

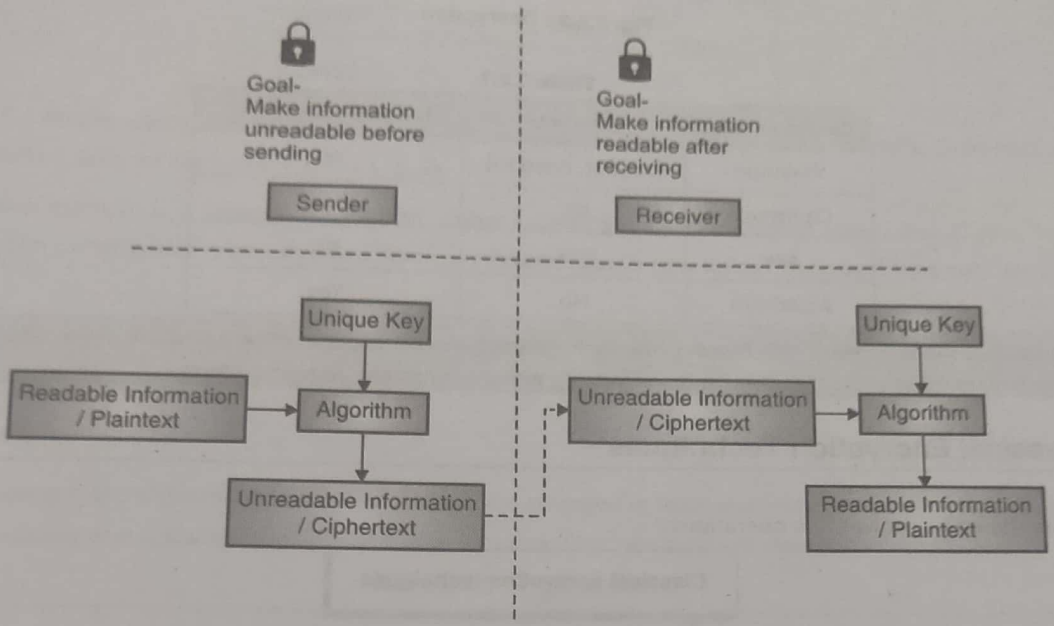


Fig. 2.2.1

5. Plaintext

The information that is readable and understandable is called plaintext.

6. Ciphertext

The information that is not-understandable even if you can read it is called ciphertext.

7. Encryption (or encipher)

Encryption is a method of converting plaintext into ciphertext by using an algorithm and a key.

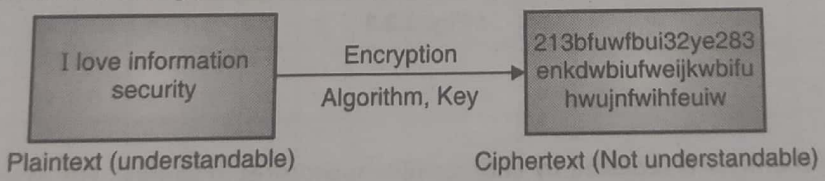


Fig. 2.2.2 : Encryption

8. Decryption (or decipher)

Decryption is a method of converting cipher text into plaintext by using an algorithm and a key.

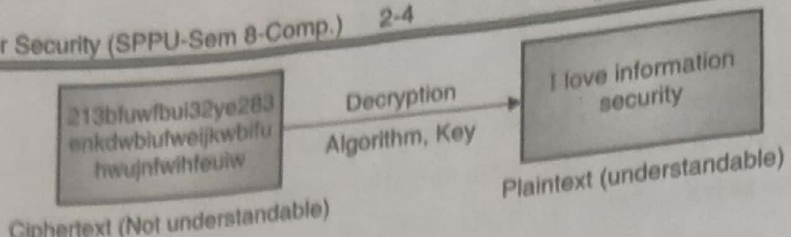


Fig. 2.2.3 : Decryption

Table 2.2.1

Crypto entities	Is secret?	Is understandable?
Plaintext	No (But, need to)	Yes
Ciphertext	No	No
Key	Yes	No
Algorithm	No	Yes

Note : The rest of the sections in this chapter heavily build upon the concepts and the introduction you got. Please take some time to make yourself familiar and comfortable with the terms before reading further.

2.3 Classical Encryption Techniques

Encryption typically involves two operations :

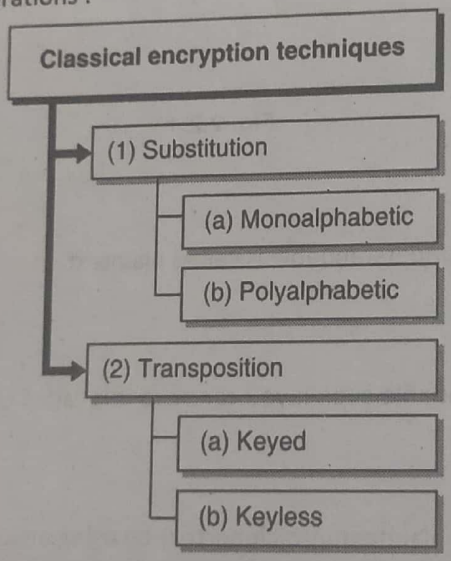


Fig. 2.3.1

2.3.1 Substitution

- In this operation, one character is replaced by another (like substitutes in games). For example, A can be substituted by D and B can be substituted by E and so on based on a chosen substitution key.
- Let's take an example. Assume that our substitution key is "shift next by 3". Recall from earlier discussion that the key is preserved secretly. Our algorithm (rule) is simple substitution. Let's apply the key and algorithm and encrypt some plaintext.

Table 2.3.1 : Simple substitution table

Plaintext	Ciphertext
I love cybersecurity	L oryh fbehuvhfxulwb
Apple	Dssoh
23456	56789

- Above is a simple substitution table where each character in plaintext is moved by 3 characters. Characters such as "y" when shifted, take the form of y →z, z→ a, a→b.
- The above example is a classical substitution cipher called Caesar cipher named after Julius Caesar. This type of substitution cipher is also referred to as a "monoalphabetic substitution cipher" because it uses only one character at a time.
- Another type of substitution cipher is called "polyalphabetic substitution cipher". In this, more than one alphabet is used at a time for encrypting plaintext. Let's learn a few polyalphabetic substitution ciphers.

1. Vignere Cipher

- Following is the Vignere table where alphabets are arranged in rows and columns. It is simple to draw. Just write a-z skipping one alphabet from left, at a time, in a row. First two rows are identical.

Table 2.3.2 : Vignere cipher

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
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
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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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
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	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	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	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	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	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	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	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

- Let's take an example:

  Plaintext : apple

  Security Key : snow

  Algorithm : Vignere Cipher

- Let's try to find out ciphertext given the information above. You will need to super-impose the security key above the plaintext you wish to encrypt. You would find the ciphertext value for each plaintext character at the intersection of column (from security key value) and row (from plaintext value).

Security Key →	s	n	o	w	s	n	o	w	← column
Plaintext →	a	p	p	l	e				← row

- Ciphertext would be
  - First letter (ciphertext of plaintext a) → intersection of column s and row a →value is s
  - Second letter (ciphertext of plaintext p) → intersection of column n and row p → value is c
  - Third letter (ciphertext of plaintext p) → intersection of column o and row p → value is d
  - Fourth letter (ciphertext of plaintext l) → intersection of column w and row l → value is h
  - Fifth letter (ciphertext of plaintext e) → intersection of column s and row e → value is w

Table 2.3.3 : Vignere cipher example

Plaintext \ Security Key	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
a	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
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	m	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

– So, the ciphertext for plaintext "apple" with security key "snow" and using algorithm "Vignere cipher" is "*scdhw*".

2. Playfair Cipher

✍ ***Definition :*** *The Playfair cipher is a symmetric-key based encryption technique that uses digraph substitution cipher.*

- This technique encrypts the pairs of alphabets (digraphs), instead of single alphabets as in the case of simple substitution ciphers like Caesar cipher. The Playfair cipher is thus significantly harder to break. It involves 625 combinations of alphabet pairs instead of just 26 in the case of single alphabets. Hence, the regular cryptanalysis techniques such as the frequency analysis are harder to perform.

Algorithm

1. Start by creating a 5x5 key square by choosing a key and filling rest of the places by the remaining alphabets such that any alphabet occurs only once in the 5x5 square. 5x5 will cover up only 25 alphabets. Hence, i and j are combined and treated as 1 position. This would then cover all the 26 alphabets in the 5x5 square.
2. Take the plaintext and remove any punctuations, special characters, numbers, etc. such that only alphabets remain in the plaintext. Then make pairs of the alphabets in the plaintext. If you have just one alphabet left out, use X to make a pair. Any pairs that have the same alphabets are also replaced by a X.
3. Use the pairs in plaintext to substitute with the key square positions. Locate the alphabets in the key square and follow the substitution rules :
   a. If the alphabets appear on the same row of the key square, replace them with the alphabets to their immediate right respectively. (If the alphabet is in the rightmost corner, wrap around to take the leftmost alphabet of the row).
   b. If the alphabets appear on the same column of the key square, replace them with the alphabets immediately below respectively. (If the alphabet is at the bottom most position, wrap around to the take the topmost alphabet of the column).
   c. If the alphabets are in different rows and columns, replace the pair with the alphabets on the same row respectively but at the corners of the rectangle defined by the original pair.

Ex. 2.3.1 : Use Playfair cipher to encrypt the word "greet" using the key "moon mission".

Soln. :

Step 1 : Construct the key square.

- Unique alphabets from the given key "moon mission" are : *m, o, n, i* and *s*. *i* and *j* are combined into one cell in the key square. Rest of the alphabets are filled serially such that the alphabets already in the key square (the key in the first row) are not repeated.
- This gives the following 5 x 5 key square.

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

Step 2 : Arrange plaintext.

- The plaintext to encrypt is "greet". It is ordered as the following pairs.

"gr", "ex", "et"

- The 2nd occurrence of *e* in *ee* is replaced with *x* to give "*ex*".

Step 3 : Apply substitution based on the key square and the plaintext pairs.

"*gr*" is encrypted as following.

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

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
  - The alphabet in the corner of the rectangle of the same row as *g* is *h*.
  - The alphabet in the corner of the rectangle of the same row as *r* is *q*.
- Hence, "*gr*" is encrypted as "*hq*".
- Now, pick the next plaintext pair "*ex*".

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

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
  - The alphabet in the corner of the rectangle of the same row as *e* is *c*.
  - The alphabet in the corner of the rectangle of the same row as *x* is *z*.
- Hence, "*ex*" is encrypted as "*cz*".
- Now, pick the next plaintext pair "*et*".

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

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
  - The alphabet in the corner of the rectangle of the same row as *e* is *d*.
  - The alphabet in the corner of the rectangle of the same row as *t* is *u*.
- Hence, "*et*" is encrypted as "*du*".
- Hence, the plaintext "*greet*" is encrypted as "*hqczdu*" using Playfair cipher using the key "*moon mission*".

Ex. 2.3.2 : Use Playfair cipher to encrypt the plaintext "Why, don't you?" using the key "keyword".

Soln. :

Step 1 : Construct the key square.

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

Step 2 : Arrange the plaintext into pairs.

(remove all punctuations).

"wh", "yd", "on", "ty", "ou"

Step 3 : Apply substitution for each plaintext pair.

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

"*wh*" is "*yl*".

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

"yd" is *"ea"*.

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

"on" is *"es"*.

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

"ty" is *"vk"*.

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

"ou" is *"ez"*.

Hence, "Why, don't you?" is encrypted as "yieaesvkez" using the key "keyword".

Ex. 2.3.3 : Use PlayFair Cipher to encrypt the message "This is a columnar transposition". Use key-APPLE.

SPPU - March 19(In Sem.), 5 Marks

Soln. :

Step 1 : Construct the key square.

Unique alphabets from the given key "apple" are → *a, p, l, e. i* and *j* are combined into one cell in the key square. Rest of the alphabets are filled serially such that the alphabets already in the key square (the key in the first row) are not repeated. This gives the following 5x5 key square.

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

Step 2 : Arrange plaintext.

- The plaintext to encrypt is "This is a columnar transposition". It is ordered as the following pairs.
- "Th", "is", "is", "ac", "ol", "um", "na", "rt", "ra", "ns", "po", "si", "ti", "on".

Step 3 : Apply substitution based on the key square and the plaintext pairs.

"th" is encrypted as following.

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

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
  - The alphabet in the corner of the rectangle of the same row as *t* is *u*.
  - The alphabet in the corner of the rectangle of the same row as *h* is *g*.
- Hence, *"th"* is encrypted as *"ug"*.
- *"is"* is encrypted as following.

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

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
  - The alphabet in the corner of the rectangle of the same row as *i* is *m*.
  - The alphabet in the corner of the rectangle of the same row as *s* is *q*.
- Hence, *"is"* is encrypted as *"mq"*.

- "ac" is encrypted as following.

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

- The alphabets appear on the same column of the key square. Hence, replace them with the alphabets immediately below respectively. (If the alphabet is at the bottom most position, wrap around to the take the topmost alphabet of the column).
  - The alphabet below *a* is *c*.
  - The alphabet below *c* is *i*.
- Hence, *"ac"* is encrypted as *"ci"*.
- "ol" is encrypted as following.

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

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
  - The alphabet in the corner of the rectangle of the same row as o is *m*.
  - The alphabet in the corner of the rectangle of the same row as *l* is *b*.
- Hence, *"ol"* is encrypted as *"mb"*.
- "um" is encrypted as following.

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

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
  - The alphabet in the corner of the rectangle of the same row as u is *s*.
  - The alphabet in the corner of the rectangle of the same row as *m* is *o*.
- Hence, *"um"* is encrypted as *"so"*.

- "na" is encrypted as following.

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

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
  - The alphabet in the corner of the rectangle of the same row as n is *i*.
  - The alphabet in the corner of the rectangle of the same row as *a* is *e*.
- Hence, *"na"* is encrypted as *"ie"*.
- "rt" is encrypted as following.

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

- The alphabets to encrypt are in the same row of the key square. Hence, you need to replace them with the alphabets to their immediate right respectively. (If the alphabet is in the rightmost corner, wrap around to take the leftmost alphabet of the row).
  - The alphabet to the right of *r* is *s*.
  - The alphabet to the right of *t* is *u*.
- Hence, *"rt"* is encrypted as *"su"*.
- *"ra"* is encrypted as following.

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

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
  - The alphabet in the corner of the rectangle of the same row as r is *q*.
  - The alphabet in the corner of the rectangle of the same row as *a* is *p*.
- Hence, *"ra"* is encrypted as *"qp"*.

- "ns" is encrypted as following.

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

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
  - The alphabet in the corner of the rectangle of the same row as n is *m*.
  - The alphabet in the corner of the rectangle of the same row as *s* is *t*.
- Hence, "*ns*" is encrypted as "*mt*".
- "po" is encrypted as following.

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

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
  - The alphabet in the corner of the rectangle of the same row as p is b.
  - The alphabet in the corner of the rectangle of the same row as *o* is *k*.
- Hence, "*po*" is encrypted as "*bk*".
- "si" is encrypted as following.

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

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
  - The alphabet in the corner of the rectangle of the same row as s is *q*.
  - The alphabet in the corner of the rectangle of the same row as *i* is *m*.
- Hence, "*si*" is encrypted as "*qm*".

- "ti" is encrypted as following.

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

- The alphabets to encrypt are in different rows and hence they form a rectangle. You need to pick the corners.
  - The alphabet in the corner of the rectangle of the same row as t is *q*.
  - The alphabet in the corner of the rectangle of the same row as *i* is *n*.
- Hence, "*ti*" is encrypted as "*qn*".
- "on" is encrypted as following.

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

- The alphabets to encrypt are in the same row of the key square. Hence, you need to replace them with the alphabets to their immediate right respectively. (If the alphabet is in the rightmost corner, wrap around to take the leftmost alphabet of the row).
  - The alphabet to the right of *o* is *i*.
  - The alphabet to the right of *n* is *o*.
- Hence, "*on*" is encrypted as "*io*".
- Hence, the plaintext message "This is a columnar transposition" is encrypted as "ugmqmq c imbsoiesuqpmtbkqmqnio" using the key APPLE and following PlayFair Cipher.

3. Hill Cipher

Definition : *The Hill cipher is a polygraphic substitution cipher based on linear algebra.*

- By polygraphic, we mean that it can work on substitution for up to 3-alphabets at a time. It arranges the key and the plaintext into a matrix format and their multiplication undergoes the mod 26 operation to find the resultant ciphertext.

Algorithm

1. Arrange the key and the plaintext in a matrix format. Use the following table for assigning numbers to alphabets for matrix operations. Note that you need to create the plaintext matrix according to the given key matrix such that multiplication is possible.

The number of columns in the key matrix must be equal to the number of rows in the plaintext matrix. If the plaintext is larger, break it up into multiple matrices and apply further steps on each of the plaintext matrix using the key. To fill the matrix, in case you are short of matrix elements, you may use zeroes.

Alphabet	Number	Alphabet	Number
A	0	N	13
B	1	O	14
C	2	P	15
D	3	Q	16
E	4	R	17
F	5	S	18
G	6	T	19
H	7	U	20
I	8	V	21
J	9	W	22
K	10	X	23
L	11	Y	24
M	12	Z	25

2. Carry out multiplication of the key and the plaintext matrix.
3. Perform mod 26 operation on the resultant multiplication.
4. Use the table again to convert numbers back to alphabets. These alphabets represent the ciphertext.

Ex. 2.3.4 : Encrypt the message "Exam" using the Hill cipher with the key $\begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix}$.

Soln. :

- Plaintext *"Exam"* when converted into a number matrix would be $\begin{bmatrix} 4 & 0 \\ 23 & 12 \end{bmatrix}$.
- Multiply the Key and Plaintext matrices.

$$\begin{bmatrix} 9 & 4 \\ 5 & 7 \end{bmatrix} \times \begin{bmatrix} 4 & 0 \\ 23 & 12 \end{bmatrix} = \begin{bmatrix} 128 & 48 \\ 181 & 84 \end{bmatrix}$$

Perform mod 26 operation on the result.

$$\begin{bmatrix} 128 & 48 \\ 181 & 84 \end{bmatrix} \bmod 26 = \begin{bmatrix} 24 & 22 \\ 25 & 6 \end{bmatrix}$$

- Converting the numbers from mod operation back to alphabets you get *"YZWG"*. Hence, encrypting the plaintext *"Exam"* using the given key gives ciphertext *"YZWG"*.

Ex. 2.3.5 : Encrypt the message "DEF" using the Hill cipher with the key $\begin{bmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 8 & 7 \end{bmatrix}$.

Soln. :

- Plaintext *"DEF"* when converted into a number matrix would be $\begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}$.
- Multiply the Key and Plaintext matrices.

$$\begin{bmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 8 & 7 \end{bmatrix} \times \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 47 \\ 40 \\ 76 \end{bmatrix}$$

- Perform mod 26 operation on the result.

$$\begin{bmatrix} 47 \\ 40 \\ 76 \end{bmatrix} \bmod 26 = \begin{bmatrix} 21 \\ 14 \\ 24 \end{bmatrix}$$

- Converting the numbers from mod operation back to alphabets you get *"VOY"*. Hence, encrypting the plaintext "DEF" using the given key gives ciphertext *"VOY"*.

Ex.2.3.6 : Using Hill Cipher encrypt the message 'ESSENTIAL'. The key for encryption is 'ANOTHERBZ'.

SPPU – May 19, 5 Marks

Soln. :

- Let's use the following table for forming a matrix for plaintext ESSENTIAL.

Alphabet	Number	Alphabet	Number
A	0	N	13
B	1	O	14
C	2	P	15
D	3	Q	16
E	4	R	17
F	5	S	18
G	6	T	19
H	7	U	20
I	8	V	21
J	9	W	22
K	10	X	23
L	11	Y	24
M	12	Z	25

- The Key ANOTHERBZ can be written in matrix form as following $\begin{bmatrix} 0 & 19 & 17 \\ 13 & 7 & 1 \\ 14 & 4 & 25 \end{bmatrix}$
- The plaintext ESSENTIAL when converted into matrix form gives $\begin{bmatrix} 4 & 4 & 8 \\ 18 & 13 & 10 \\ 18 & 19 & 11 \end{bmatrix}$
- Now, multiply the key matrix with the plaintext matrix and perform mod 26 on the resultant matrix.

$$\begin{bmatrix} 0 & 19 & 17 \\ 13 & 7 & 1 \\ 14 & 4 & 25 \end{bmatrix} \times \begin{bmatrix} 4 & 4 & 8 \\ 18 & 13 & 10 \\ 18 & 19 & 11 \end{bmatrix} = \begin{bmatrix} 648 & 570 & 187 \\ 196 & 162 & 115 \\ 578 & 583 & 387 \end{bmatrix}$$

$$\begin{bmatrix} 648 & 570 & 187 \\ 196 & 162 & 115 \\ 578 & 583 & 387 \end{bmatrix} \bmod 26 = \begin{bmatrix} 24 & 24 & 5 \\ 14 & 6 & 11 \\ 6 & 11 & 23 \end{bmatrix}$$

- Arranging the resulting matrix back to alphabets we get encrypted text as YOGYGLFLX

4. Affine Cipher

> ***Definition :*** *Affine cipher is a type of monoalphabetic substitution cipher, wherein each letter in an alphabet is mapped to its numeric equivalent, encrypted using a simple mathematical function, and converted back to a letter.*

- Each letter is encrypted using the Affine function *(Ax + B) mod 26*. Values (A, B) are called coefficients of Affine function. You use the position value of each alphabet and use the Affine function to calculate the position value of the corresponding encrypted letter. Then, you substitute each plaintext letter with the corresponding encrypted letter from the table.
- Let's see an example.

Ex. 2.3.7 : Using Affine cipher encrypt the Plaintext 'SECURITY' with key pair (5, 2).

Soln. :

- Here the coefficients of Affine function are (5, 2). So, A = 5 and B = 2.
- Create a table of all alphabets using the formula *(Ax + B) mod 26*. Use A = 5 and B = 2 for this computation.

Position Value of Alphabet	Plaintext Alphabet	Affine Calculation (A*Position + B)mod 26	Encrypted Alphabet (Position from Affine calculation)
0	A	(5 * 0 + 2) mod 26 = 2	C
1	B	(5 * 1 + 2) mod 26 = 7	H
2	C	(5 * 2 + 2) mod 26 = 12	M
3	D	(5 * 3 + 2) mod 26 = 17	R

Position Value of Alphabet	Plaintext Alphabet	Affine Calculation (A*Position + B)mod 26	Encrypted Alphabet (Position from Affine calculation)
4	E	(5 * 4 + 2) mod 26 = 22	W
5	F	(5 * 5 + 2) mod 26 = 1	B
6	G	(5 * 6 + 2) mod 26 = 6	G
7	H	(5 * 7 + 2) mod 26 = 11	L
8	I	(5 * 8 + 2) mod 26 = 16	Q
9	J	(5 * 9 + 2) mod 26 = 21	V
10	K	(5 * 10 + 2) mod 26 = 0	A
11	L	(5 * 11 + 2) mod 26 = 5	F
12	M	(5 * 12 + 2) mod 26 = 10	K
13	N	(5 * 13 + 2) mod 26 = 15	P
14	O	(5 * 14 + 2) mod 26 = 20	U
15	P	(5 * 15 + 2) mod 26 = 25	Z
16	Q	(5 * 16 + 2) mod 26 = 4	E
17	R	(5 * 17 + 2) mod 26 = 9	J
18	S	(5 * 18 + 2) mod 26 = 14	O
19	T	(5 * 19 + 2) mod 26 = 19	T
20	U	(5 * 20 + 2) mod 26 = 24	Y
21	V	(5 * 21 + 2) mod 26 = 3	D
22	W	(5 * 22 + 2) mod 26 = 8	I
23	X	(5 * 23 + 2) mod 26 = 13	N
24	Y	(5 * 24 + 2) mod 26 = 18	S
25	Z	(5 * 25 + 2) mod 26 = 23	X

- Once the table is ready, you can substitute each letter in the plaintext with its corresponding encrypted letter.

Position Value of Alphabet	Plaintext Alphabet	Affine Calculation (A*Position + B)mod 26	Encrypted Alphabet (Position from Affine calculation)
0	A	(5 * 0 + 2)mod 26 = 2	C
1	B	(5 * 1 + 2)mod 26 = 7	H

Position Value of Alphabet	Plaintext Alphabet	Affine Calculation (A*Position + B)mod 26	Encrypted Alphabet (Position from Affine calculation)
2	C	(5 * 2 + 2)mod 26 = 12	M
3	D	(5 * 3 + 2)mod 26 = 17	R
4	E	(5 * 4 + 2)mod 26 = 22	W
5	F	(5 * 5 + 2)mod 26 = 1	B
6	G	(5 * 6 + 2)mod 26 = 6	G
7	H	(5 * 7 + 2)mod 26 = 11	L
8	I	(5 * 8 + 2)mod 26 = 16	Q
9	J	(5 * 9 + 2)mod 26 = 21	V
10	K	(5 * 10 + 2)mod 26 = 0	A
11	L	(5 * 11 + 2)mod 26 = 5	F
12	M	(5 * 12 + 2)mod 26 = 10	K
13	N	(5 * 13 + 2)mod 26 = 15	P
14	O	(5 * 14 + 2)mod 26 = 20	U
15	P	(5 * 15 + 2)mod 26 = 25	Z
16	Q	(5 * 16 + 2)mod 26 = 4	E
17	R	(5 * 17 + 2)mod 26 = 9	J
18	S	(5 * 18 + 2)mod 26 = 14	O
19	T	(5 * 19 + 2)mod 26 = 19	T
20	U	(5 * 20 + 2)mod 26 = 24	Y
21	V	(5 * 21 + 2)mod 26 = 3	D
22	W	(5 * 22 + 2)mod 26 = 8	I
23	X	(5 * 23 + 2)mod 26 = 13	N
24	Y	(5 * 24 + 2)mod 26 = 18	S
25	Z	(5 * 25 + 2)mod 26 = 23	X

- So,

$S \rightarrow O$, $E \rightarrow W$, $C \rightarrow M$

$U \rightarrow Y$, $R \rightarrow J$, $I \rightarrow Q$

$T \rightarrow T$, $Y \rightarrow S$

- So, the plaintext SECURITY, when encrypted using Affine cipher using the Affine coefficients of (5,2), give OWMYJQ as the encrypted text.

- Here if you have to decrypt the ciphertext using Affine cipher, you follow the same approach. You first create the ta using the Affine coefficients and then substitute the encrypted letters with their corresponding plaintext letters.

2.3.1(A) Difference between Monoalphabetic and Polyalphabetic Ciphers

SPPU – March 19(In Se

Q. Explain Monoalphabetic and Polyalphabetic ciphers with appropriate examples. **(March 19, 5 Marks)**

- Note a key difference between monoalphabetic cipher and polyalphabetic cipher –
  - For repeated characters, in monoalphabetic cipher, ciphertext is same (for example, plaintext y is ciphertext b per example we chose earlier) whereas
  - For polyalphabetic cipher, repeated plaintext characters need not lead to the same ciphertext (for exam there are two instances of p in plaintext word apple in the polyalphabetic cipher example. One is encrypted a and another is encrypted as d).
- So, polyalphabetic ciphers are stronger than monoalphabetic since they usually give different ciphertext values f repeated characters in plaintext and hence are less prone to frequency analysis attack. Frequency analysis attack tr to find a correlation between plaintext and ciphertext and determine the security key. For example, the attack might guess that y is encrypted as b in the monoalphabetic cipher, so it could mean the security key is "shift next b 3". Once the attacker determines the key, converting any ciphertext back to plaintext is a trivial (very simple) task.
- For Cryptography to be successful, keeping the key secret is very important.

2.3.2 Transposition

- In this operation, the position of characters is jumbled up (mixed up) like a letter arranging game.

  For example, the plaintext "apple" could be transposed into ciphertext as "elpap". Note that all the characters i plaintext are also present in the ciphertext but at different positions.

  For example, position of "a" in plaintext is 1 whereas in ciphertext it is 4. It is a very simplistic example. Variou complex mathematical transposition algorithms are usually used in cryptography.
- Transposition can be carried out using two techniques – Keyed and Keyless. Let us learn about them.

1. Keyed Transposition Cipher

Definition : In keyed transposition, a random key is used to describe the transposition sequence and carr out the transposition.

This is also called Columnar Transposition Cipher.

Algorithm

1. Arrange the plaintext in a column under the given key.
2. Rearrange the plaintext column-wise in key's alphabetic order.

Ex. 2.3.8 : Use the key "ENCRYPT" to encrypt the plaintext "Save the king from attack" using transposition cipher.

Soln.:

- Draw a table and arrange the key and the plaintext under the key column-wise. Mark the alphabets in the key in their order alphabetically. For example, for the given key, the alphabet "*C*" comes first in the order of 26 alphabets.
- Then comes "*E*" and hence marked 2. Likewise mark all the alphabets in the key according to their occurrence in the alphabets.

E	N	C	R	Y	P	T
2	3	1	5	7	4	6
s	a	v	e	t	h	e
k	i	n	g	f	r	o
m	a	t	t	a	c	k

- Read the columns in order.
  - Take column C marked as 1st in order → vnt
  - Take column E marked as 2nd in order → skm
  - Take column N marked as 3rd in order → aia
- Follow likewise to get the ciphertext as "*vntskmaiahrcegteoktfa*".

Ex. 2.3.9 : Use the key "SORROW" to encrypt the plaintext "Demonetization tonight" using transposition cipher.

Soln.:

- Draw a table and arrange the key and the plaintext as following. Mark the order of the columns from left to right in case of repeated key characters. Pad the columns with "x" to fill the columns if the plaintext does not fill the table completely.

S	O	R	R	O	W
5	1	3	4	2	6
d	e	m	o	n	e
t	i	z	a	t	i
o	n	t	o	n	i
g	h	t	x	x	x

- The resulting ciphertext is "*eimhntnxmzttoaoxdtogeiix*".

Ex. 2.3.10 : Use Transposition Cipher to encrypt plain text 'I Love my India' and use the key 'HEAVEN'. [Use single columnar transposition] **SPPU - May 19, 5 Mar**

Soln. :

- Arrange the unique characters in the key in a column

H	E	A	V	N
3	2	1	5	4
i	l	o	v	e
m	y	i	n	d
i	a	-	-	-

- Now, arrange the letters in the order of columns
- This gives "oilyaimiedvn" as the ciphertext.

2. Keyless Transposition Cipher

Definition : In keyless transposition, a transposition sequence is described without a random key.

One such example of Keyless Transposition Cipher is Railfence Cipher. Railfence cipher uses the rail size as a key a does not use a random key as such. It can be easily broken.

Algorithm

1. Based on the rail size, arrange the plaintext.
2. Rearrange the plaintext row-wise to get the ciphertext.

Ex. 2.3.11 : Encrypt the plaintext "Save the king from attack" using Railfence cipher. Assume a suitable rail size.

Soln. :

- Assuming a rail size of 3. All it means is that it would have 3 rows. Arrange the plaintext in rails one alphabet at a time

rail 1 →	s				t				i				r				t				k
rail 2 →		a		e		h		k		n		f		o		a		t		c	
rail 3 →			v				e				g				m				a		

- Rearrange the plaintext rail-wise (row-wise) to get the ciphertext. Start from rail 1, then rail 2 and finally rail 3. Here the ciphertext would be "*stirtkaehknfoatcvegma*".

Ex. 2.3.12 : Encrypt the plaintext "Demonetization tonight" using Railfence cipher. Assume the rail size of 4.

Soln. :

- Arrange the plaintext in 4 rails (rows).

rail 1 →	d						t					o						g		
rail 2 →		e				e		i				i		n				i		h	
rail 3 →			m		n				z		t				t		n				t
rail 4 →				o						a						o					

- The resulting ciphertext is *"dtogeeiinihmnzttntoao"*.

2.4 Rotor Machines

Definition : A rotor machine is an electro-mechanical stream cipher device used for encrypting and decrypting secret messages.

- Rotor machines were the cryptographic state-of-the-art for a prominent period of history. They were in widespread use in the 1920s–1970s.
- The most famous example is the German Enigma machine, whose messages were deciphered by the Allies during World War II, producing intelligence code-named Ultra.

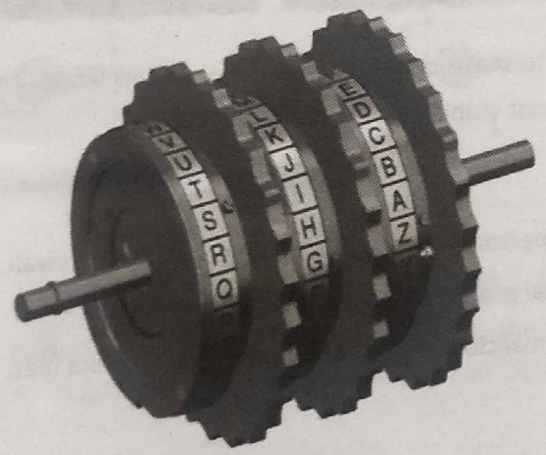


Fig. 2.4.1

- The primary component is a set of rotors, also termed wheels or drums, which are rotating disks with an array of electrical contacts on either side. The wiring between the contacts implements a fixed substitution of letters, replacing them in some complex fashion.
- On its own, this would offer little security; however, after encrypting each letter, the rotors advance positions, changing the substitution. By this means, a rotor machine produces a complex polyalphabetic substitution cipher, which changes with every keypress.

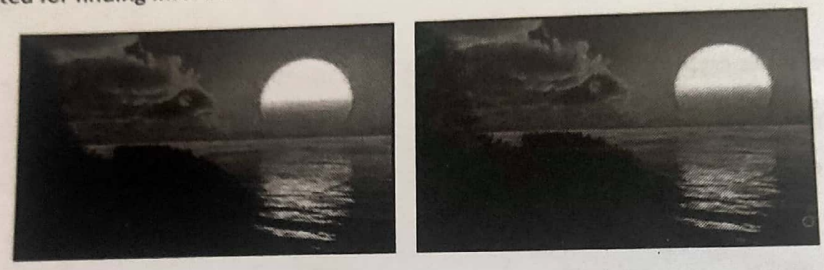
2.5 Steganography

SPPU – May 1

Q. What is Steganography? What are the applications and limitations of Steganography? (May 19, 5 Marks)

Definition : *Steganography is the practice of concealing a message within another message, image, or file.*

- The information is only hidden and not encrypted. The hiding is so non-obvious that it is difficult to discover it by anyone who is unaware of the presence of the hidden information. Only who knows what to look for and where can lookout for the hidden information.
- There are many different methods of performing steganography. The most famous of all is the one that modifies only the LSBs (Least Significant Bits). In media files such as images, audio or video, it is difficult to make out any difference between the files with modified LSBs and the files where LSBs are not modified.
- Hence, the information can be transferred hidden where generally these files are not considered harmful or are thoroughly inspected for finding information transfer. Do you see any difference between the following two images?



- That is precisely how hard it is to make out the hidden information where the variations between the two files is extremely hard to make out and not visible to the human eye.

A. Uses of Steganography

1. Leak corporate, business or personal data without being caught by firewall, IDS or other detection mechanisms.
2. Sending information in special groups without knowledge of others.
3. Attacking users with hidden malicious code in the downloaded media files.

B. Limitations of Steganography

1. Without declared algorithms, it is difficult to hide and unhide the secret message.
2. Somehow, the recipients must be told where to look for the hidden information.
3. The original image must be destroyed so that it is difficult for someone to find the difference between the images.
4. Steganography is not a real secure way of communication. It just provides security by obscurity which means that is just tries to complicate things rather than actually securing the communication.
5. Only a small amount of information can be hidden without distorting the image such that it becomes noticeable.

C. Comparison between Cryptography and Steganography

Sr. No.	Cryptography	Steganography
1.	The information is transformed.	The information is hidden.
2.	Transformed information is visible.	Hidden information is not visible.
3.	Provides Confidentiality, Integrity, Non-repudiation.	Provides Confidentiality only.
4.	Various recognized and approved algorithms.	No such specific algorithms.

2.6 Methods of Encryption

As you know, encryption is primarily driven by two components – Keys and Algorithms. Based on the number of keys used in the encryption and decryption process, the encryption methods can be classified as shown in Fig. 2.6.1.

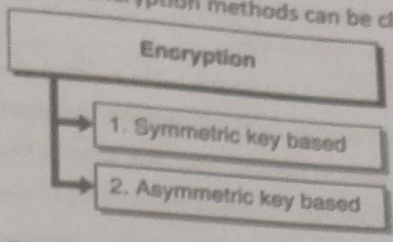


Fig. 2.6.1 : Methods of Encryption

2.6.1 Symmetric Key Encryption

- Symmetric means same.

Definition : In Symmetric Key Encryption, the key used for encryption is same as the key used for decryption.

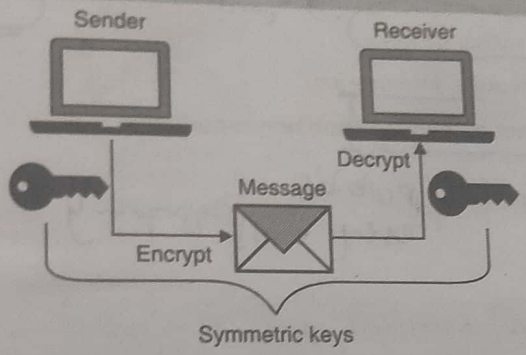


Fig. 2.6.2 : Symmetric Key Encryption

The keys are identical. The sender as well as the receiver must have exactly the same key to encrypt or decrypt. As an example, symmetric key is like your regular lock key. The same key can be used for locking the door as well as unlocking the door.

As you understand, a symmetric key is unique between a sender and a receiver. If there are more entities involved and each requires to secretly communicating with the another, you end up having multiple keys. Let's take an example, suppose there are 4 friends – A, B, C and D and each one of them require communicating with one another secretly. It is obvious that you cannot share the keys between a pair of friends with another pair of friends.

So, if A and B share a key, B and C cannot share the same key because C would also know the secret key between A and B and can then decrypt communication between A and B. So, you would require several keys to ensure that each pair of sender and receiver have a unique key. So, how many keys would you need?

You would need below keys (one for each pair of sender and receiver):

1. A < > B
2. A < > C
3. A < > D
4. B < > C
5. B < > D
6. C < > D

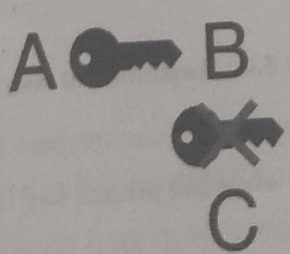


Fig. 2.6.3

- This can be effectively calculated using the formula $K = N \frac{(N-1)}{2}$ where N is the number of entities that need to secretly communicate. So, in the above example, $K = 4 \times (4-1)/2 = 6$. If we have 100 entities, it would require $100 \times (100-1)/2 = 4,950$ keys! What if the entire world wants to communicate with each other? Can you imagine? I will come back to it later on and answer that for you.
- Another problem here is how does A send the key she is using to B? If the sender and receiver have to use the same key, there should be a way to securely transfer the key. Isn't it? For example, if you have to give your house keys to your friend, you probably exchange hands in person. You don't leave the key somewhere that could potentially be picked / looked by someone else other than your friend. I will answer this question as well later on.
- Some of the examples of symmetric key algorithm are DES, AES and Blowfish.

Advantages of Symmetric Keys

1. Computationally faster than the asymmetric keys
2. Hard to break if the key used is long

Disadvantages of Symmetric Keys

1. Requires a secure mechanism to exchange keys
2. Each pair of sender and receiver require a unique key
3. Provides only confidentiality but not authenticity and non-repudiation

2.6.2 Asymmetric Key Encryption

- Unlike symmetric keys,

> ***Definition :*** *In Asymmetric Key Encryption, there are two keys that are mathematically related. If one is used for encryption, then only the corresponding other key can be used for decryption.*

- Asymmetric means not equal. In the asymmetric system, two mathematically related keys work as key pair. If you use one key for encryption, then you need the other key in the pair for decryption. The encrypting key cannot be used for decrypting.

Note : Asymmetric Keys based Cryptography is also called as Public Key Cryptography.

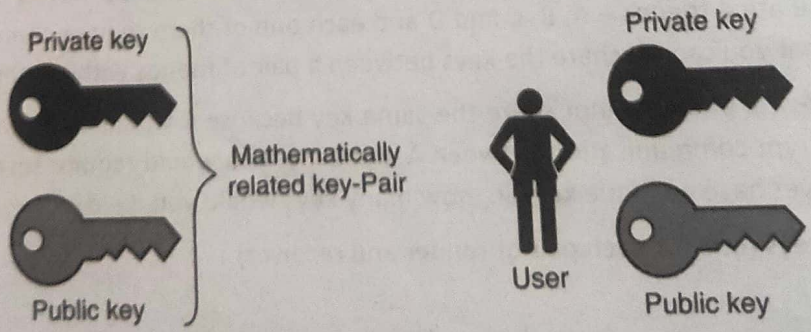


Fig. 2.6.4 : Asymmetric Key Encryption

- Let's call one of the keys as the Public Key and its counterpart in the pair as the Private Key. A user owns b keys. The public key is known to the world while the private key is known only to the user.

Table 2.6.1

Key Used	Corresponding Key Required	Security Service Provided
Encryption – Public Key	Decryption – Private Key	Confidentiality
Private Key	Public Key	Authentication and Non-repudiation

- Let us understand these two use cases of the asymmetric keys.

Use Case 1 : User A wants to send a secret message to User B

- User A knows : User A's Public Key, User A's Private Key, User B's Public Key
- User B knows : User A's Public Key, User B's Public Key, User B's Private Key

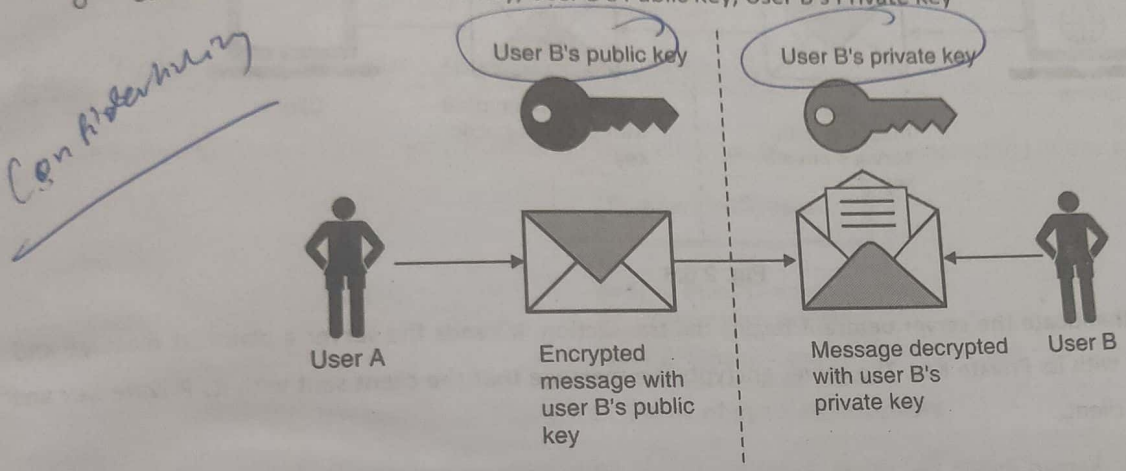


Fig. 2.6.5 : Encryption and decryption using public key cryptograpghy

- User A wants to send an encrypted message such that only User B can read it. User A encrypts the message with User B's Public Key. Now, because User A used User B's Public Key, she is sure that only User B can decrypt the message as decryption would require the corresponding Private Key and only User B knows about her Private Key.
- Hence, in this scenario you find that asymmetric keys as well can be used to send encrypted messages as you saw in the case of symmetric keys.
- One core difference to note here is that User A need not know User B's Private Key to send her an encrypted message. A separate key distribution problem does not exist as Public Keys are known to the world and only the user needs to know and protect her Private Keys. You also see that you require only two keys per entity for encrypted communication.
- So, for 100 people to send each other encrypted messages, we would require only 200 keys unlike 4,095 symmetric keys (recall from our discussion on symmetric keys in the previous section)! So, asymmetric keys help you to overcome two of the limitations of symmetric keys:
  - Key distribution and
  - Number of keys to manage
- Hence, I answer the question for you that I asked in the previous section – how do we manage keys if the whole wo wants to communicate with each other? The answer is using asymmetric keys!

Use Case 2 : Proving authenticity and non-repudiation

- The second use case of asymmetric keys is for proving authenticity of an entity. This is highly used today for server validation. Have you seen "https" in front of website address? That is one of the examples of this use case.
- Suppose, you want to do an online transaction. How do you ensure that the bank's website address you are interacting with is the right one? That's precisely what asymmetric keys help you solve as well.

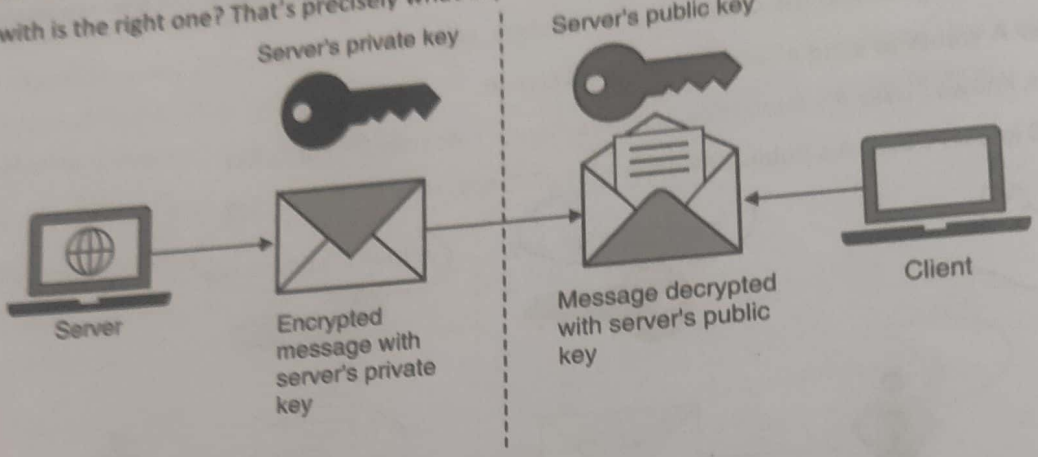


Fig. 2.6.6

- Client wants to authenticate the server before it begins the transaction. It sends the server a plaintext message and asks it to encrypt it with its Private Key. The server encrypts the message that the client sent with its Private Key and sends it back to the client.
- Client uses the world-known Public Key of the server and decrypts the message it received from the server. If the message gets successfully decrypted and it matches with what the client sent earlier to the server to encrypt, the client has now validated that it is indeed interacting with the authentic server because except the authentic server, no one else would have known server's Private Key. The client is satisfied, and it begins the secure transaction after having established the server's authenticity.
- Hence, you find that asymmetric keys could effectively be used for authentication and non-repudiation. Some examples of algorithms that use asymmetric keys are RSA, ECC, Diffie-Hellman, and El Gamal.

Advantages of Asymmetric Keys

1. Easy key distribution
2. Less number of keys to manage
3. Can also be used for providing authentication and non-repudiation

Disadvantages of Asymmetric Keys

1. Slower than symmetric keys
2. Requires significant CPU power due to complex mathematical relation between the keys

2.6.3 Comparison between Symmetric and Asymmetric Keys

Sr. No.	Comparison Attribute	Symmetric Keys	Asymmetric Keys
1.	Speed	High	Low
2.	Complexity	Low	High
3.	Number of keys	High	Low
4.	Key Distribution	Problematic	Easier
5.	Security Services	Confidentiality	Confidentiality, Authenticity, Non-repudiation

2.7 Types of Symmetric Algorithms (Ciphers)

Symmetric key based algorithms (ciphers) can work either on blocks of bits (characters) or one bit at a time.

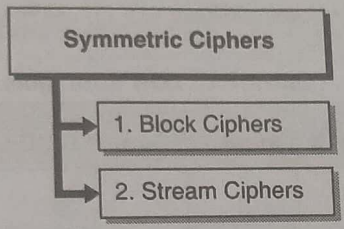


Fig. 2.7.1 : Types of symmetric ciphers

✍ ***Definition :*** *The algorithms that work on blocks are called block ciphers.*

✍ ***Definition :*** *The algorithms that work on one-bit at a time are called stream ciphers.*

2.7.1 Block Ciphers

SPPU - May 19

Q. What is block Cipher? (May 19, 2 Marks)

- In block ciphers, the information that needs to be encrypted is broken into smaller and equal block sizes. Then, the encryption operation (substitution and transposition) is applied to each block. The resultant ciphertext from each block is then combined to produce the encrypted information.
- Fig. 2.7.2 illustrates simplified block diagram of how a block cipher works. The information is broken into equal size blocks and then the encryption operation is carried out on each block. If the block size has lesser number of characters than required to form a block, then padding is done to fill the block. Padding is just filling some temporary information to form a block. Finally, the resulting encrypted information from each block is combined to get the overall encrypted message.
- DES and AES are two of the examples of Symmetric Block Ciphers.

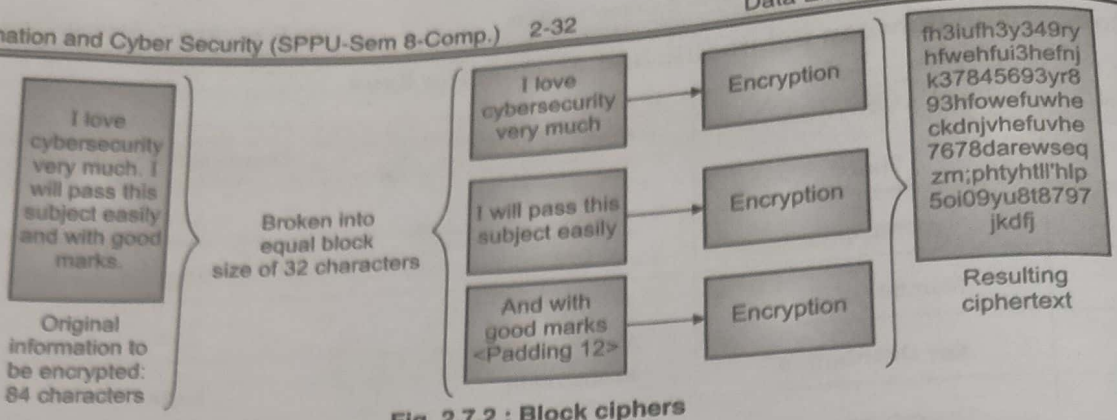


Fig. 2.7.2 : Block ciphers

2.7.2 Stream Ciphers

- Unlike block ciphers, stream ciphers work on one bit of plaintext at a time. Each bit of plaintext is combined with the bit of security key and then XORed to get ciphertext.

Note : If you recall from your logical design classes, following is the truth table of XOR. For result to be 1, both the inputs should be different.

Table 2.7.1 : XOR truth table

Input X	Input Y	Output Z (XOR X, Y)
0	0	0
0	1	1
1	0	1
1	1	0

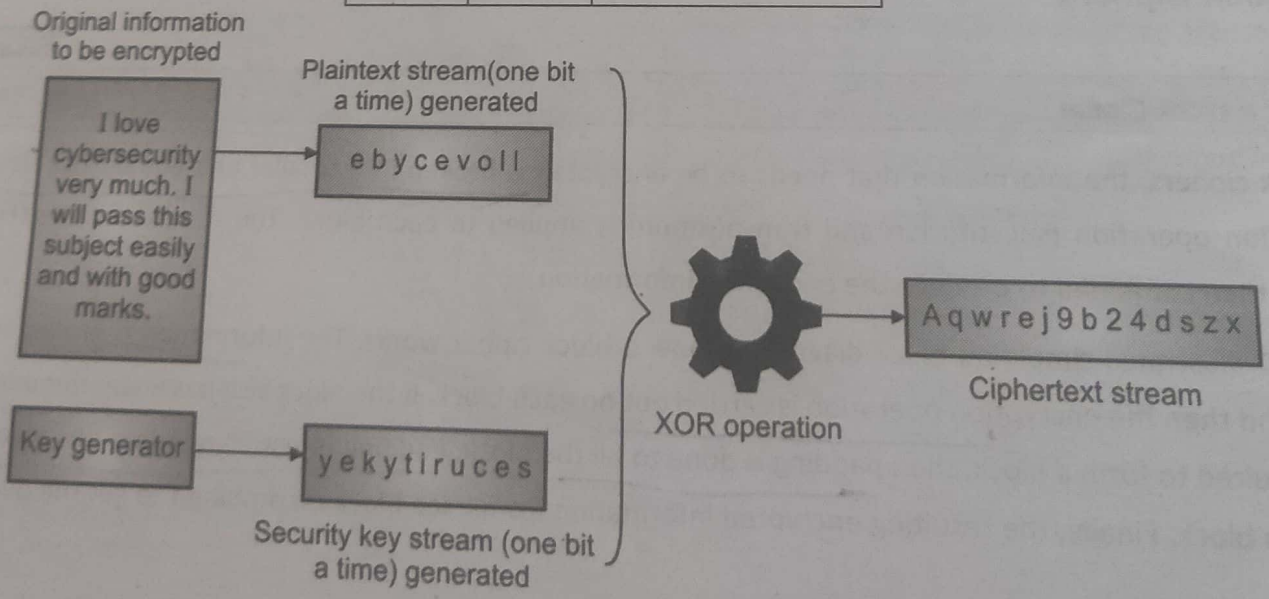


Fig. 2.7.3 : Stream ciphers

- RC4 is an example of stream cipher.

2.7.3 Comparison between Block and Stream Cipher

Sr. No.	Comparison Attribute	Block Cipher	Stream Cipher
1.	Security	High	Low
2.	Speed	Low	High
3.	Application	Non-real time such as documents	Real time data such as Voice
4.	Commonly used	Yes	No

2.8 Data Encryption Standard (DES)

Definition : Data Encryption Standard (DES) is a symmetric key based block cipher standard used for encryption and decryption.

It came into existence and usage around Nov 1976 and was predominantly used in the industry until 2002.

Major attributes of DES

each round → 1 subkey of 48 bits

- It is a symmetric key based algorithm.
- It works as a block cipher.
- It uses 64-bit blocks.
- It uses a key size of 64-bits in which 56-bits are the actual keys and 8-keys are used for error detection.
- It uses 16 rounds of operation (substitution and transposition) to convert a block of plaintext into cipher text.
- DES is now considered insecure and obsolete due to its short key-size (56-bits only).

o/p → 64 bit cipher text

2.8.1 Block Cipher Design Principles (DES Design Criteria)

- There are three critical components in designing a block cipher.

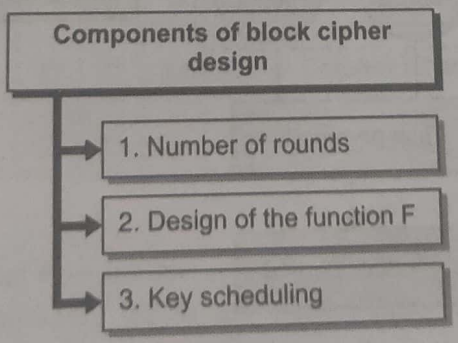


Fig. 2.8.1

- Following are the principles around each of these.

1. Design Principles for Number of Rounds in the Block Cipher Algorithm

a. The greater the number of rounds, the more difficult it is to perform cryptanalysis.

b. The number of rounds is chosen such that a known cryptanalysis takes a greater effort compared to brute-force attack.

2. **Design Principles for function F (Feistel network) in the Block Cipher Algorithm**

a. It must be difficult to re-assemble the substitution performed by the function F.

b. F is non-linear which means it is difficult to establish any relation between input to F and output from F.

c. F should have high avalanche effect.

3. **Design Principles for Key scheduling**

a. Subkey selection should be such that it is difficult to work backwards to derive the main key.

b. Subkeys should be hard to guess as well.

c. The key schedule should produce avalanche effect.

2.8.2 Block Diagram and Internals of DES

SPPU – March 19(In Sem.)

Q. Explain the operation of DES algorithm in detail. (March 19, 5 Marks)

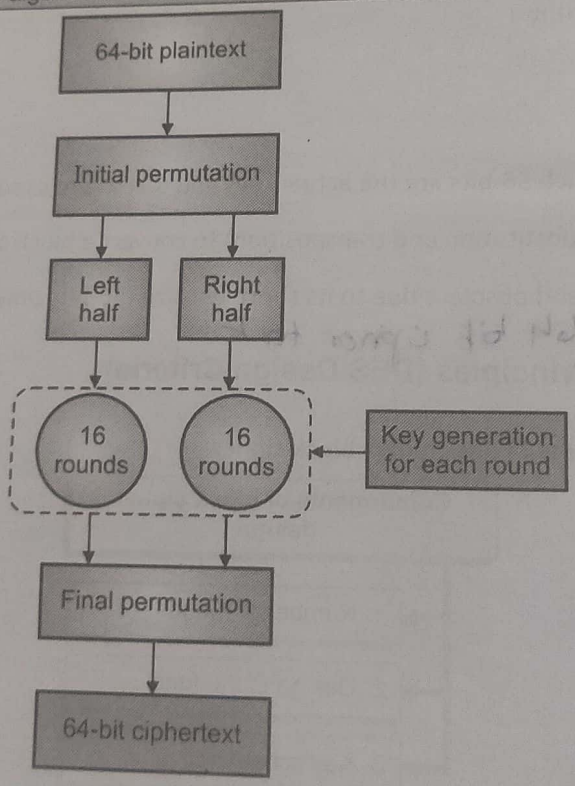


Fig. 2.8.2

– Fig. 2.8.2 shows simplistic view of DES. Let us understand what happens at each stage.

Step 1 : Creation of 64-bit blocks

In this step, the plaintext information to be encrypted is broken into 64-bit blocks. DES is a block cipher block creation is similar to as explained in the earlier section.

Table 2.8.1 : 64 bits of plaintext

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

Step 2 : Initial Permutation

In this step, the 64 bits in the plaintext blocks are re-arranged (transposed). This is done as per the diffusion property of the cipher to ensure that any small variance in plaintext produces a large variance in the cipher text.

Table 2.8.2 : Initial permutation (re-arrange bits of plaintext)

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

p 3 : Left Half and Right Half Split

In this step, the bits from the Initial Permutation stage are split into two parts – left half and right half each containing 32-bits. These individual 32-bit blocks are then continuously worked through the 16 rounds of operation.

4 : Subkey Key Generation

For the 16 rounds of operation, a unique subkey is derived for each round from the 56-bit key. The key is derived using complex mathematical functions. Each generated subkey is 48-bit long.

Step 5 : Rounds

Left half and the right half both individually go through 16 rounds of encryption operation. In each of the roun

the derived subkey is used to produce temporary ciphertext.

This temporary ciphertext produced after each round is used in the next round until the final round is comple

Each round consists of substitutions and successive permutations.

Step 6 : Final Permutation

In the last stage, we need to bring the bits back to their respective positions. The bit positions were changed at th

initial permutation stage.

Table 2.8.3 : Final permutation (re-arrange bits of ciphertext)

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

Step 7 : Final Ciphertext

Once all the steps are done, you get the final ciphertext for the plaintext given the security key of your choice via DES.

Exam Tip : If you hear that an algorithm is broken or is insecure, it means that it is computationally feasible to find out the key used for encryption. Note that cryptography heavily depends upon our understanding of mathematics and the computation power available today. What is secure and infeasible today could be insecure and feasible to crack in future.

2.8.3 Block Cipher – Modes of Operation (for DES and other Block Ciphers in General)

SPPU - March 19(In Sem.), May 19

Q. Explain following algorithms modes

i. ECB

ii. OFB **(March 19, 5 Marks)**

Q. Explain counter mode of block Cipher. **(May 19, 3 Marks)**

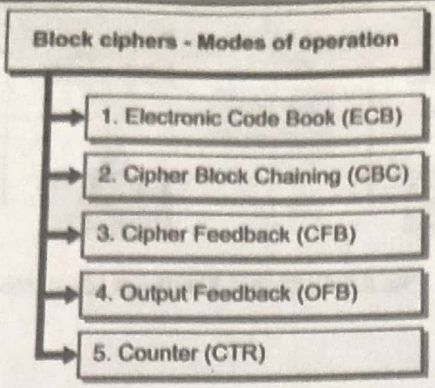


Fig. 2.8.3 : Modes of operation for block cipher

DES and other block ciphers can potentially work in several modes. Let's review them carefully.

1. **Electronic Code Book (ECB) Mode**

In this mode, the same key is used to encrypt all the blocks. Key derivatives or subkeys are not used. Additionally, each block is treated separately and the ciphertext of previous block does not influence successive blocks.

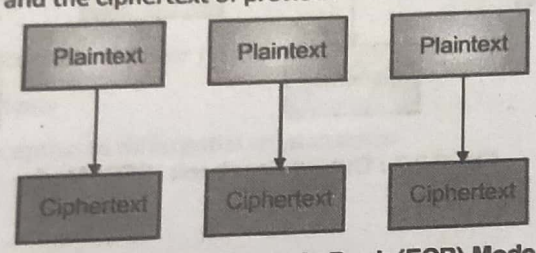


Fig. 2.8.4 : Electronic Code Book (ECB) Mode

2. **Cipher Block Chaining (CBC) Mode**

In this mode, the ciphertext of previous block is used with the next plaintext block. The two blocks (ciphertext of previous block and plaintext of next block) are XORed and then passed through the encryption operation. This generates a lot more randomness in the final ciphertext.

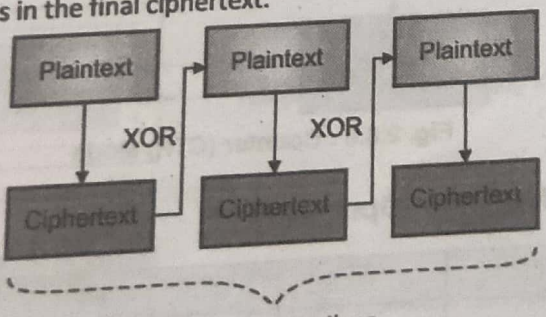


Fig. 2.8.5 : Cipher Block Chaining (CBC) Mode

3. **Cipher Feedback (CFB) Mode**

In this mode, the block cipher works like a stream cipher. The ciphertext from the previous block is XORed with the key (keystream) for the next block. This way the key increasingly becomes random and brings more randomness in the overall encryption process.

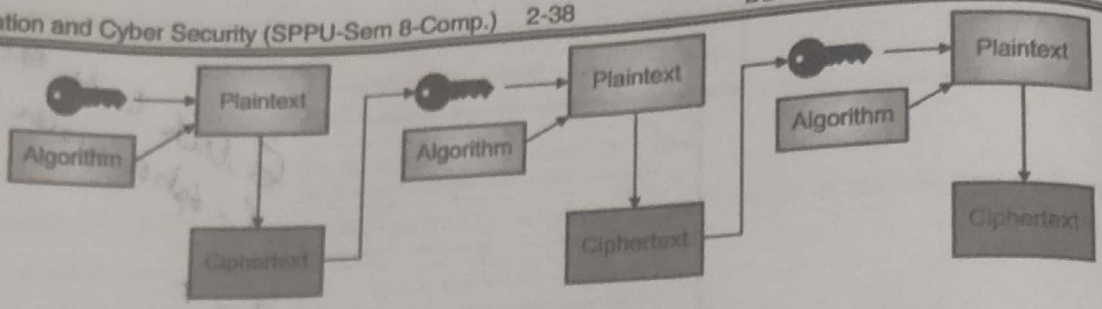


Fig. 2.8.6 : Cipher Feedback (CFB) Mode

4. **Output Feedback (OFB) Mode**

In this mode, the block cipher works like a stream cipher. The keystream used in the previous block is XORed with the keystream of the next block.

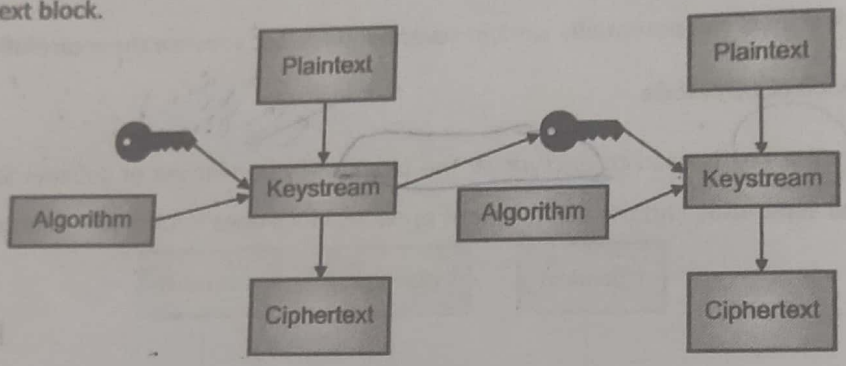


Fig. 2.8.7 : Output Feedback (OFB) Mode

5. **Counter (CTR) Mode**

In this mode as well, the block cipher works like a stream cipher. The key is converted into keystream (as used in stream cipher) and the keystream is XORed with a counter that increases for every block.

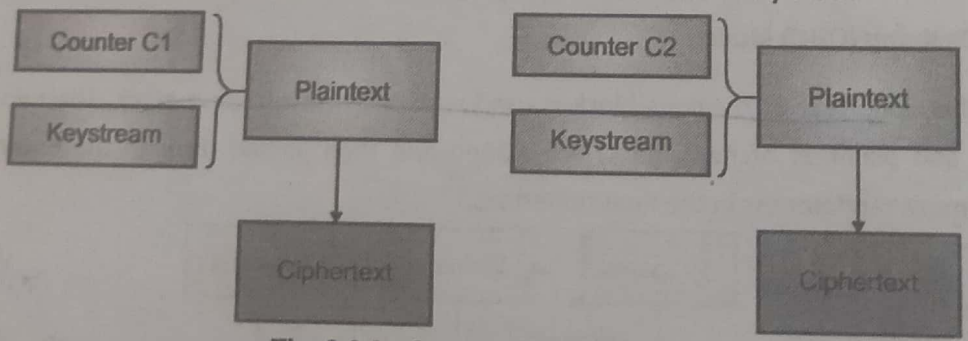


Fig. 2.8.8 : Counter (CTR) Mode

2.8.4 Comparison between Modes of Operation

Sr. No.	Mode	ECB	CBC	CFB	OFB	CTR
1.	In-Parallel block encryption	Yes	No	No	No	Yes
2.	Suited for	Small Information	Any size of information	Small Information	Small Information	Any size of information
3.	Security and randomness	Low	High	High	High	High

Sr. No.	Mode	ECB	CBC	CFB	OFB	CTR
4.	Speed	High	Medium	Medium	Medium	High
5.	Complexity	Low	High	High	High	Low
6.	Works like stream cipher?	No	No	Yes	Yes	Yes

2.8.5 Weakness in DES

1. **Small key size**

   56-bits of keys have a keyspace (possible values) of 2^{56}. While that might seem a lot, it is actually not given the compute power we have today. In 1990s, the compute power we had was significantly lower and hence was considered secure at that time.

   Definition : The type of attack where each combination is tried in an attempt to find the right combination is also called as brute force attack.

2. **Prone to linear cryptanalysis**

   DES has been proven to be susceptible to linear cryptanalysis.

3. **Prone to differential cryptanalysis**

   DES has been proven to be susceptive to differential cryptanalysis.

2.8.6 Double DES

- In order to strengthen DES, it was considered to increase the key size to 112-bits effectively. The way it was chosen to do so was to use 2 keys of 56-bits each. Let's call them K1 and K2.
- Mathematically, it can be denoted as below :

  Ciphertext C = Encryption (K2, Encryption (K1, Plaintext P))

  Plaintext P = Decryption (K1, Decryption (K2, Ciphertext C))

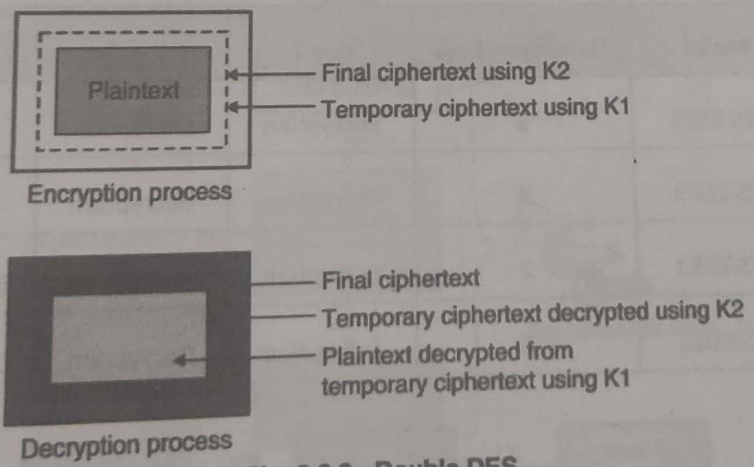


Fig. 2.8.9 : Double DES

- During the encryption process, first the plaintext is encrypted with Key K1 and then the result is again encrypted with Key K2 to get the final ciphertext for plaintext.

- During the decryption process, first the Key K2 is used to decrypt to get the ciphertext that Key K1 can decrypt to get the plaintext.
- However, Double DES was proven to be ineffective. Meet in the middle attack was shown to reduce the complexity to just 2^{57} (2^{56} attempts made twice, hence $2 \times 2^{56} = 2^{57}$) instead of 2^{112} as originally thought.
- So, using K1 if you could derive temporary ciphertext using encryption process and using K2 if you could also derive the same temporary ciphertext using decryption process, you have found a match and the keys you chose (K1 and K2) are now known to you. Hence, you could effectively find both the keys and break Double DES without original thought of complexity of 112 bits.

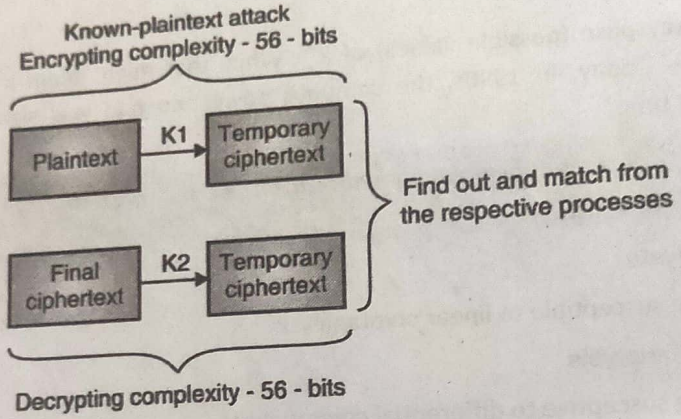


Fig. 2.8.10 : Complexity in Double DES

- Hence, Double DES was not adopted in the industry and is not used.

2.8.7 3DES or Triple DES

- Finding that Double DES was ineffective, Triple DES or 3DES was conceived. 3DES uses 48 rounds of operation and can work in the following modes using two or three keys.

Table 2.8.3

Sr. No.	Mode	Number of keys	Key 1	Key 2	Key 3
1.	DES-EEE3	3	Encryption	Encryption	Encryption
2.	DES-EDE3	3	Encryption	Decryption	Encryption
3.	DES-EEE2	2	Encryption	Encryption	Encryption Using Key 1
4.	DES-EDE2	2	Encryption	Decryption	Encryption Using Key 1

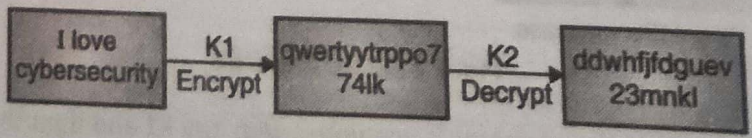


Fig. 2.8.11 : Decryption process

- You might wonder how decryption helps. Note that if you encrypt a plaintext using a key (say K1) and run the decryption process using a different key (say K2), the text (from encryption process using K1) becomes more random. The use of a different key in the decryption process from the encryption process brings added randomness and hence helps to make attacks such as linear or differential cryptanalysis extremely hard.

2.9 Advanced Encryption Standard (AES)

Definition : Advanced Encryption Standard (AES) is a symmetric key based block cipher standard used for encryption and decryption.

The standard became effective on May 26, 2002 and is predominantly used in the industry today due to its strong cipher properties. AES replaced DES as the new standard when DES was found to be insecure and vulnerable to various attacks.

Major attributes of AES

- It is a symmetric key based algorithm.
- It works as a block cipher.
- It uses 128-bit blocks.
- It can work with key sizes of 128, 192 and 256 bits.
- Number of rounds of operation depends upon the key size.
  - 128-bit key undergoes 10 rounds.
  - 192-bit key undergoes 12 rounds.
  - 256-bit key undergoes 14 rounds.
- AES is considered highly secure due to its long key sizes and is used in the industry today.

2.9.1 Block Diagram and Internals of AES

SPPU – May 19

Q. Explain working of AES in detail. (May 19, 5 Marks)

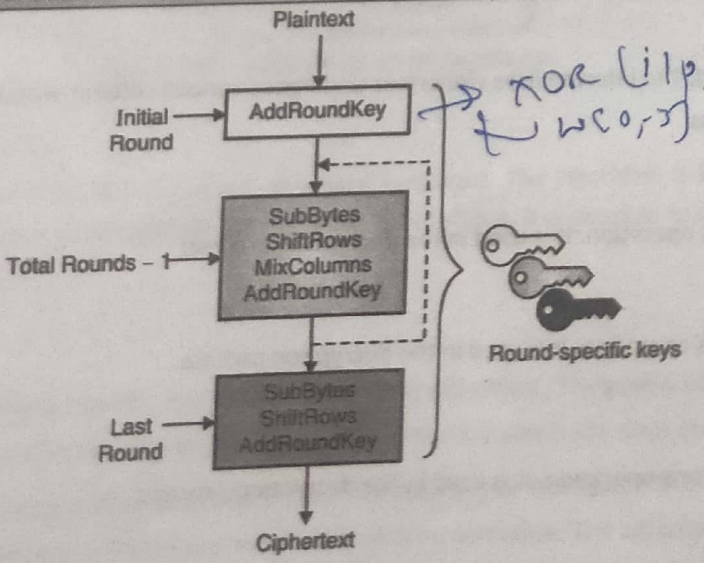


Fig. 2.9.1(a) : AES Encryption Process

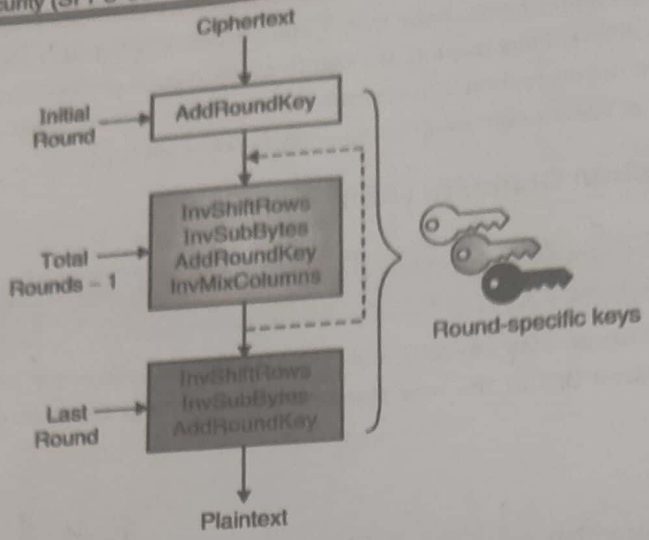


Fig. 2.9.1(b) : AES Decryption Process

Let's understand the blocks.

1. **AddRoundKey**

   In this transformation step, a round key is generated and XORed with the intermediate (temporary) ciphertext. This block is used in both encryption as well as decryption process.

2. **SubBytes**

   In this transformation step, the intermediate ciphertext undergoes various substitution operations. It is used for encryption process.

3. **ShiftRows**

   In this transformation step, the intermediate ciphertext undergoes various row-wise transposition operations. It is used for encryption process.

4. **MixColumns**

   In this transformation step, the intermediate ciphertext undergoes various column-wise transposition operations. It is used for encryption process.

5. **InvSubBytes**

   This is inverse of SubBytes operation. It is used in the decryption process.

6. **InvShiftRows**

   This is inverse of ShiftRows operation. It is used in the decryption process.

7. **InvMixColumns**

   This is inverse of MixColumns operation. It is used in the decryption process.

2.9.2 Comparison between DES and AES

Sr. No.	Comparison Attribute	DES	AES
1.	Cryptographic Strength	Low	High
2.	Key Size	56-bit	128, 192 and 256 bits
3.	Block Size	64-bit	128-bit
4.	Rounds	16	10, 12, 14 - based on key size
5.	Usage	Obsolete - Not used	Currently used industry standard

2.10 Attacks on Cryptosystems

Now that you have a general understanding of the cryptosystems, let's learn some of the possible attacks on them.

Note : The attacks described here are common for any cryptographic algorithm be it DES, AES, RSA or any other. While for some algorithms it is comparatively easier and for others it is theoretically possible. Any specific attack against an algorithm is described in its respective section. Otherwise, you could mention and elaborate on the following attacks.

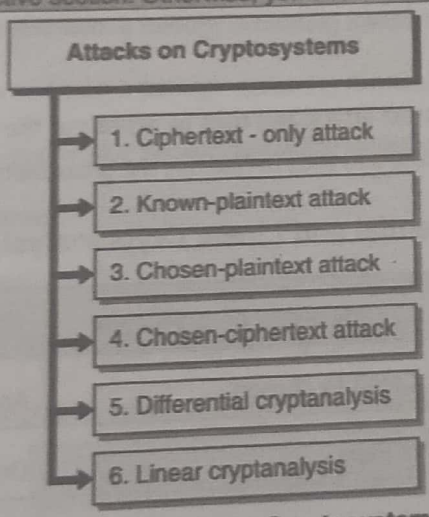


Fig. 2.10.1 : Attacks on Cryptosystems

1. **Ciphertext-only attack**

In this type of attack, the attacker has ciphertext of several messages. The algorithm is known, and the goal of the attack is to find out the key used in encryption. Once the key is found out, it is possible to decrypt messages that were encrypted using the key.

2. **Known-plaintext attack**

The attacker knows the plaintext partially and the corresponding ciphertext. The goal is to find out the key. Once the key is known, the attacker can then use the key to decrypt ciphertext for which she does not know the plaintext.

For example, you might be using a fixed greeting in your messages (For example, "Dear Friend") or you might be sending same message (for example, "Good morning") everyday to someone. The attacker could know this and also the corresponding ciphertext and try to find out the key.

3. **Chosen-plaintext attack**

The attacker knows the exact plaintext and the corresponding ciphertext. The goal is to find out the key. Once the is known, the attacker can then use the key to decrypt ciphertext for which she does not know the plaintext. example, the attacker can send you a message that she knows that you will definitely forward to your friends. W forwarding, you might encrypt the message using your key. Now, the attacker can grab the ciphertext that you s and she already knows the plaintext message she sent you earlier.

4. **Chosen-ciphertext attack**

In this attack, the attacker chooses the ciphertext that she wants to be decrypted and know the correspond plaintext. The goal again is to find out the key.

5. **Differential cryptanalysis**

In this attack, the attacker chooses a pair of plaintexts and follows through each stage in their respective encryptio process and compare the difference between the results at each stage. The key used in encrypting the pair is sam The goal again is to figure out the key by carefully studying the differences in results at various stages between th pair. Since, the attacker chooses the plaintexts, differential analysis is considered to be a type of chosen-plainte attack.

6. **Linear cryptanalysis**

The attacker carries out a known-plaintext attack and tries to figure out the key. She evaluates the input and outpu at various stages of the encryption process and tries to find out the probability of specific key values.

2.10.1 Comparison between Differential and Linear Cryptanalysis

Sr. No.	Comparison Attribute	Differential Cryptanalysis	Linear Cryptanalysis
1.	Plaintext selection	Carefully chosen	Any random plaintext
2.	Plaintext used	In Pairs	One by one
3.	Complexity of attack	High	Low
4.	Mathematical relation between plaintexts used	Specific differences (such as XOR)	Linear approximation (such as a series of XOR operations)
5.	Goal of the attack	Identify some bits of the unknown key	Identify the linear relation between some bits of the plaintext, some bits of the cipher text and some bits of the unknown key

Review Questions

Here are a few review questions to help you gauge your understanding of this chapter. Try to attempt these ques and ensure that you can recall the points mentioned in the chapter.

Introduction to Cryptography

Q. 1 With a block diagram, explain the concept of cryptography. [6 Marks]

Q. 2 Describe the various terms used in cryptography. [6 Marks]

Classical Encryption Techniques

Q. 3 Explain substitution technique used in cryptography with a suitable example. [6 Marks]

Q. 4 Explain transposition technique used in cryptography with a suitable example. [6 Marks]

Q. 5 Take an example of your choice and work it through explaining Vignere Cipher. [8 Marks]

Q. 6 Take an example of your choice and work it through explaining Playfair Cipher. [8 Marks]

Q. 7 Take an example of your choice and work it through explaining Hill Cipher. [8 Marks]

Q. 8 Take an example of your choice and work it through explaining Affine Cipher. [8 Marks]

Q. 9 Write a short note explaining the difference between monoalphabetic and polyalphabetic cipher. [6 Marks]

Q. 10 Take an example of your choice and work it through explaining Keyed Transposition Cipher. [8 Marks]

Q. 11 Take an example of your choice and work it through explaining Keyless Transposition Cipher. [8 Marks]

Q. 12 Take an example of your choice and work it through explaining Railfence Cipher. [6 Marks]

Q. 13 Write a short note on rotor machines. [4 Marks]

Q. 14 Write a short note on steganography. [5 Marks]

Q. 15 Provide a comparison between Cryptography and Steganography. [6 Marks]

Q. 16 Out of Cryptography and Steganography, which one would you suggest using for secure communication? [4 Marks]

Methods of Encryption

Q. 17 Describe Symmetric Key Encryption. [6 Marks]

Q. 18 If you were to use Symmetric Key Encryption, how many keys would you require for securely communicating amongst 100 users? [4 Marks]

Q. 19 List the advantages and disadvantages of Symmetric Key Encryption. [5 Marks]

Q. 20 Describe Asymmetric Key Encryption. [6 Marks]

Q. 21 If you were to use Asymmetric Key Encryption, how many keys would you require for securely communicating amongst 100 users? [4 Marks]

Q. 22 List the advantages and disadvantages of Asymmetric Key Encryption. [5 Marks]

Q. 23 Compare between Symmetric and Asymmetric Keys encryption. [4 Marks]

Types of Symmetric Algorithm

Q. 24 With a suitable diagram and example, explain what is a block cipher? [5 Marks]

Q. 25 With a suitable diagram and example, explain what is a stream cipher? [5 Marks]

Q. 26 Compare block and stream cipher. [4 Marks]

Data Encryption Standard (DES)

Q. 27 List the Major attributes of DES. [6 Marks]

Q. 28 List the Block Cipher Design Principles. [6 Marks]

Q. 29 Draw a block diagram of DES and explain. [8 Marks]

Q. 30 Describe various Block cipher modes of operation. [4 Marks]

Q. 31 With a suitable diagram, explain Electronic Code Book (ECB) Mode of block cipher operation. [4 Marks]

Q. 32 With a suitable diagram, explain Cipher Block Chaining (CBC) Mode of block cipher operation. [4 Marks]

Q. 33 With a suitable diagram, explain Cipher Feedback (CFB) Mode of block cipher operation. [4 Marks]

Q. 34 With a suitable diagram, explain Output Feedback (OFB) Mode of block cipher operation. [4 Marks]

Q. 35 With a suitable diagram, explain Counter (CTR) Mode of block cipher operation. [8 Marks]

Q. 36 Compare various modes of block cipher operations. [4 Marks]

Q. 37 List the weakness in DES. [4 Marks]

Q. 38 Write a short note on Double DES. [4 Marks]

Q. 39 Write a short note on Triple DES. [4 Marks]

Q. 40 Write a short note on 3DES.

Advanced Encryption Standard (AES)

Q. 41 What is AES? List its major attributes. [4 Marks]

Q. 42 Draw the block diagram of AES and explain its working. [8 Marks]

Q. 43 Compare AES and DES. Which one would you use and why? [8 Marks]

Attacks on Cryptosystems

Q. 44 Describe the various attacks possible on Cryptosystems. [6 Marks]

Q. 45 Compare differential and linear cryptanalysis. [6 Marks]