CNS.

Information Security Intro.

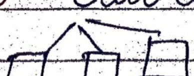
Need for Security

1. Security - State free from attacks
2. Plain text - original message S → R.
3. Cipher text - received by receiver in unreadable.
4. Encryption algo - PT → CT.
5. Decryption algo - CT → PT.
6. Keys - Used for conversion of PT to CT or vice versa
7. Cryptography - Scheme / Study of encryption
8. Crypt Analysis - Scheme of decryption

7. CRYPTOGRAPHY

The study of encryptions are divided into 3 divisions

1) Type of operations used for PT → CT.
   Ex: Substitution, Transposition
2) No of keys
   - Symmetric (only 1 key for En & De)
   - Asymmetric (2 keys, 1 - En, 2 - De)
3) The way in which PT is processed
   a) stream cipher - bit by bit
   b) block cipher



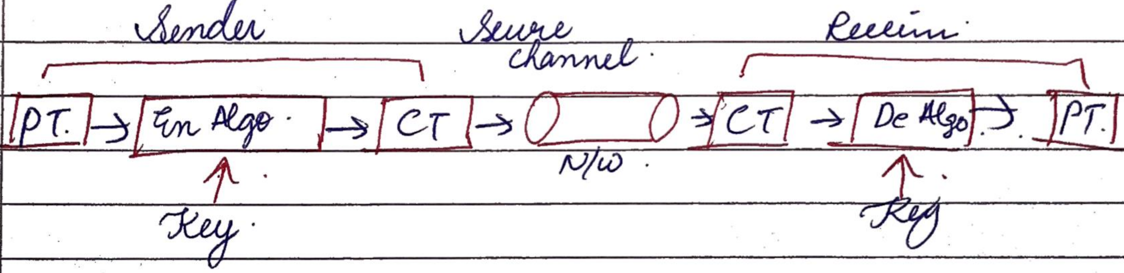
perform op on each block independently

8. CRYPTANALYSIS.

It represents attack.

a) Crypt Analytic Attack:
By observing the plain text to cipher text, and the patterns, we are identifying the key

b) Brute force:- By all possible combinations of plain text and keys to get CT.



SECURITY ATTACKS.

An action that changes security of information owned by an org

1) Passive Attack:- observes data but does not modify

2) Active Attack:- Modifies the data.

Types of Passive Attacks

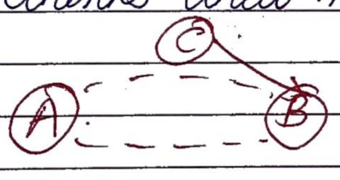
a) Release of message contents:-
A sends info to B and C which is an unauthorised person simply observes the data.

Ex:- phone tap, email tap.

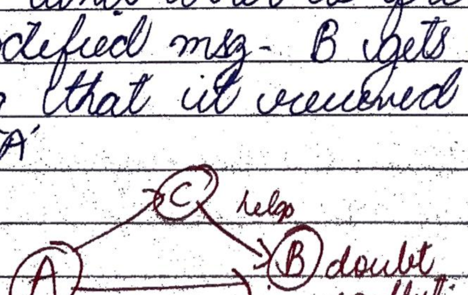
b) Traffic Analysis:- A wants to send message to B, A sends unencrypted message. So the attacker observes pattern, length frequency, location etc

Types of Active Attacks:

a) Masquerade:- A and B are 2 authorized persons where A is the sender & B is the receiver, We have an unauthorized person 'C', through some communication channel, if A wants to send message to B, before A sends the message to B, C sends the message to B. Here Masq means one entity pretends to be other entity. B thinks that message is from A.

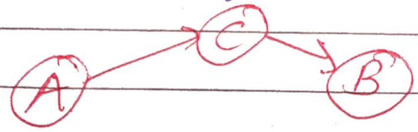


b) Replay:- Here A sends some info to B and the same info is attacked by 'C' and the info is transfered to 'B'
∴ B receives 2 messages. One msg is from A and other is from C which is the modified msg. B gets doubt conflict thinking that it received msg two times from 'A'

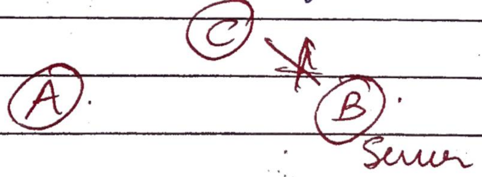


c) Modification of message

The message that wants to be send by A, is received by the attacker, the attacker reviews the message and modifies it, and that modified message is sent to B



d) Denial of Service :- If 'B' is the server and A is a client the attacker simply breaks the n/w. It disturbs all the services provided by the server by the name of 'A'



SECURITY SERVICES

Services means some of the techniques used for providing security.

1) Authentication : assurance that communication is authentic

a) peer authentication :
we have to identify the entities that are connected

b) Data Origin Authentication :- Identify the Source of data unit.

2). Access control :- ability to limit & control access of a system. prevents unauthorized access of the system.

3) Data Confidentiality Prevents passive attacks

a) connection oriented - All user data on a connection over a period of time

b) connectionless confidentiality - providing security only for a single block of data

c) Selective field - Only a single block of user data, the selective field is taking

d) Traffic flow :- Observes the source of data then provides security at the source data

4) Data Integrity :- assurance that receiver receives exactly same info as sent by the sender

a) connection oriented - over a connection all user data is secured

b) connectionless - assurance for a single block of user data

c) Selective field - providing security for a selective field

__/__/__

5) Non Repudiation:- prevents either sender or receiver from denying of service

6) Availability of Services- availability of properties/resources of the system is to be provided.

SECURITY MECHANISMS

* Under OSI Security Architecture we design security attacks, security services and security mechanisms
* The security services are implemented by using security mechanisms
* Security mechanism is a process designed to detect or prevent the security attacks
* The security mechanisms under X.800 are divided into 2 categories
* 1) Specific Security Mechanisms:- These are the mechanisms that are applied to specific protocol
a) Encipherment :- By using maths formula we convert PT → CT
b) Digital Signature - The sender performs some algorithmic transformation on the data

_/_/_

and it produces some output. That output is called Signature. And the Sender Sends data + sign combined to the receiver. The receiver performs same algo on the data and if he gets same sign, then msg is transferred correctly.

c) Access control :- It provides access rights to the resources

d) Data Integrity :- Variety of mechanisms are used to assure data integrity

f) Authentication exchange :- By using information exchange we can assure identity of an entity.

g) Traffic Padding :- The gaps in the data Stream are filled with some bits so that it is difficult to detect the frequency of bits i.e traffic analysis is reduced.

h) Routing Control :- Selection of Secure routes for certain data and if any Suspect occurs, we can change the route.

i) Notarization :- We have to use a trusted 3rd party to assure certain properties of data exchange

___/___/___

* 2) Pervasive Security Mechanisms
These are not Specific to any protocol.

a) Trusted functionality:- info must be correct w.r.t criteria

b) Security Label make a marking bound to a resource which Specifies Security attributes of the resource

c) Event detection:- If any security-related event occurs, those events are detected

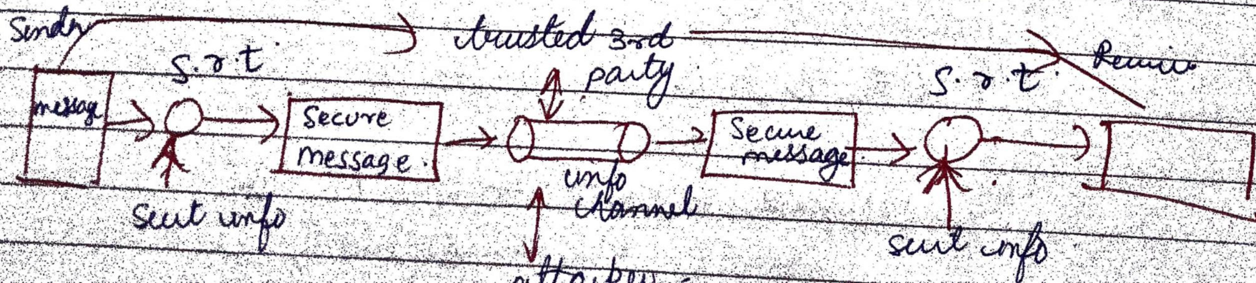
d) Security Recovery:- If any security problem occurs, we have to recover from it.

e) Security Audit Trail:- Once the data is collected we have to make an independent review and check the examination of System records.

srt :- security related transformations

A MODEL FOR N/W SECURITY

* A message is transfered from one party to another party
* That info is being transfered through internet
* The two parties must co-operate with each other for the exchange of data.
* The main aim is that we must establish a logical connection or a channel b/w the two parties by using TCP/IP protocol
* Whatever the techniques we are using, all the " are providing Security as 2 components.
  1) Security related info :- In the info to be sent we have to apply Security related info
  2) Secret info :- The info that is shared by the 2 parties to prevent from attacks
* During these 2 implementation, we require a trusted party (third) to secure transactions



The general model in designing a to sec
shows 4 basic task

1) Design an algo
2) Generate ~~keys~~ the secret info used with the algo
3) Develop methods for distribution/ sharing of secret info
4) Specify a protocol that is to be used by two parties involved in the communication

CLASSICAL ENCRYPTION TECHINQUE

PT → CT

1) Substition - replacement
2) Transposition - rearrangement

Types of Substitution Techniques

a) Ceaser Cipher:-
Each and every letter in the plain text is replaced with 3 letters further from the plain text

Eg:- a b c d e f g h i j k . . . .

P.T = a b c
C.T = d e f

Formula :- CT = E (P.T, 3)
PT = D (C.T, 3)

Formula for Ceaser Cipher:-

$$CT = (PT+3) \bmod 26$$

Eg PT = w x y z

w = 22 | y = 24
x = 23 | z = 25

(22+3) % 26
25 i.e. z

Problem :- It is very easy for the attacker as the key size is always fixed

b) Mono alphabetic cipher:-

Each and every letter in the plain text is replaced with 'K' letters from the plain text

eg:- a b c d e f g h - - - .

if K=4

P-T = w x y z
C-T = a b c d

formulae:- CT = E(PT, K)
PT = D(CT, K)

$$CT = (PT+K) \bmod 26$$

Eg:- w = (22+4) % 26
= a

c) Play Fair Cipher.

We require a P.T and a key. and the procedure for filling the matrix is as follows.

Procedure

1. Take a 5×5 matrix.
2. Ex key is: playfairexample.
3. top left → right
4. repeated letters are ignored
5. remaining letters in alphabetical order with condition both (i/j) occupies single cell

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

Rules for performing encryption.

1. Divide the P.T into Pair of letters. if there is a single letter, add 'x'
   Eg:- we | lc | om | ex.
2. If the pair contains repeated letters then add 'x' b/w both
   Eg:- hello
   he | ll | o
   he | lx | lo.

3) If two letters are in same row replace with immediate right letters
4) If two letters are in same coloumn replace with below letters
5) If the two letters are not in same row and same coloumn, then draw a rectangle / square as corners of P.T and remaining 2 corners on same row becomes CT

3. Eg:- PT - cd
CT - dg

Eg2. PT - ch
CT - db.

4. Eg:- PT - ed
CT - do.

Eg2 PT - dv
CT - oa

5) Eg:- PT - cx.
CT - gr.

Problem:- PT:- good morning.
Key:- playfairexample.

∴ PT = Go|od|mo|rn|in|gx.

go - dq.
od - vo.
mo - es.
rn - cu
in - rk
gx - qg.

∴ CT = dqvoescurkqg

d) Hill Cipher :-

It is a multi letter cipher which encrypts a group of letters : diagraphs, trigraph, polygraph

Expressed as

$$C.T = E(K, PT) = P \times K \mod 26.$$
$$P.T = D(K, CT) = CK^{-1} \mod 26.$$
$$= P \times K \times K^{-1} \mod 26$$

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

Eg:- C = KP modes.

ex: PT : He/ll/lo/wo/or/ld.

~~CP~~ Sol:

PT = He/ll/lo/wo/or/ld.

$$Key = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix}.$$

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

$$\begin{bmatrix} h \\ e \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 7 \\ 4 \end{bmatrix} \mod 26 = \begin{bmatrix} 2\times7 + 1\times4 \\ 3\times7 + 4\times4 \end{bmatrix} \mod 26$$

$$\begin{bmatrix} 18 \\ 37 \end{bmatrix} \bmod 26 \Rightarrow \begin{bmatrix} 18 \\ 11 \end{bmatrix} = \begin{bmatrix} s \\ l \end{bmatrix}$$

CT for He = Sl

b) $P = \begin{bmatrix} l \\ l \end{bmatrix} = C = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 11 \\ 11 \end{bmatrix} \bmod 26$

$$= \begin{bmatrix} 33 \\ 77 \end{bmatrix} \bmod 26 = \begin{bmatrix} 7 \\ 25 \end{bmatrix} = \begin{bmatrix} h \\ z \end{bmatrix}.$$

c) $P = \begin{bmatrix} o \\ w \end{bmatrix}$ $C = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 14 \\ 22 \end{bmatrix} \bmod 26$

$$= \begin{bmatrix} 50 \\ 130 \end{bmatrix} \bmod 26 = \begin{bmatrix} 26 \\ 0 \end{bmatrix} = \begin{bmatrix} y \\ a \end{bmatrix}$$

d). $P = \begin{bmatrix} o \\ r \end{bmatrix}$ $C = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \bmod 26. \begin{bmatrix} 14 \\ 17 \end{bmatrix}.$

$$= \begin{bmatrix} 45 \\ 110 \end{bmatrix} \bmod 26 = \begin{bmatrix} 19 \\ 6 \end{bmatrix} = \begin{bmatrix} t \\ g \end{bmatrix}$$

9 21 22 10 23 30 17 18

e). ~~ldle~~ $P = \begin{bmatrix} l \\ d \end{bmatrix}$ $C = \begin{bmatrix} 2 & 1 \\ 3 & 4 \end{bmatrix} \bmod 26. \begin{bmatrix} 11 \\ 3 \end{bmatrix}.$

$$= \begin{bmatrix} 25 \\ 50 \end{bmatrix} \bmod 26 = \begin{bmatrix} 22 \\ 24 \end{bmatrix} = \begin{bmatrix} z \\ y \end{bmatrix}$$

7 10 20 22 23 14 15 18 19

C K P

$[\ ]_{3\times1}$ $[\ ]_{3\times3}$ $[\ ]_{3\times1}.$

—/—/—

Transposition Techniques

performing permutation on plain Text

Techniques

1) Rail fence

In this, the PT is written rewritten as sequence of diagonals, of any depth, read as sequence of rows.

Ex:- meet me after the yoga party

r1 → m e m a t r h p r y

r2 → e t e f e t e a t

CT → memat rhpryetefeteat

It is easy to break so we require a complex scheme

Matrix written row by row and read in column by column is columnar transposition

key ⇒ Order of columns

PT → attack postponed until twoam

PT →

a t t a c k p
o s t p o n e
d u n t i l t
w o a m x y z

Key → 4 3 1 2 5 6 7

CT → ttnaaptmtsuoaodwcoixtncypetz

Ex 2. if key is other than numbers
Eg
LASER. then give numbering in A-Z
∴ laser
3 1 5 2 4 which is the order of the columns

To make it more complex, we are converting this CT (CT) into matrix rep,
∴ we call it double Transposition

CT → ttnaaptmtsuoaodwcoixknzypetz.

Matrix → PT

t	t	n	a	a	p	l
m	t	s	u	o	a	o.
d	w	c	o	i	x	k.
n	l	y	p	e	t	z

Key: 4 3 1 2 5 6 7.

∴ For the second iteration the above is the PT, now if we take the same key then we get CT as
CT: nscyauopttwltmdnaoiepaxtlok

Trick To Visualize double Transposition

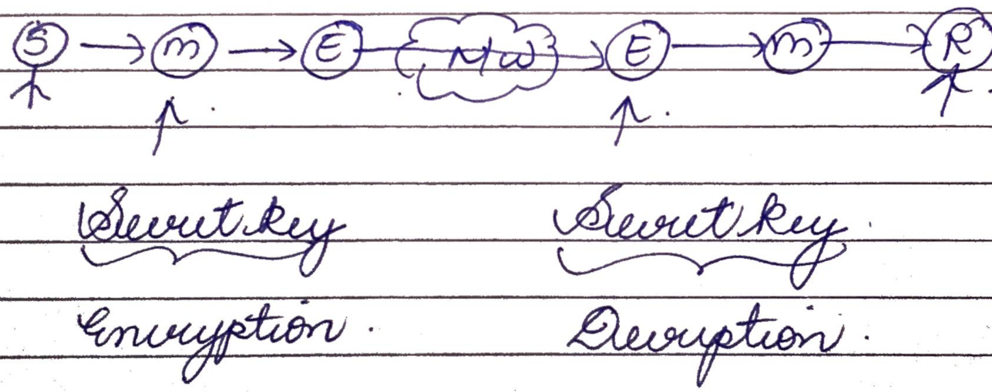
Key:	④	③	①	②	⑤	⑥	⑦
PT:	01	02	03	04	05	06	07
	08	09	10	11	12	13	14
	15	16	17	18	19	20	21
	22	23	24	25	26	27	28

CT → 0310172404111825020916230108152205121926061320270714212 8.

M → 03 10 17 24 04 11 18.
25 02 09 16 23 01 08
15 22 05 12 19 26 06.
13 20 27 07 14 21 28

SYMMETRIC KEY CRYPTOGRAPHY

one key → on S & R.

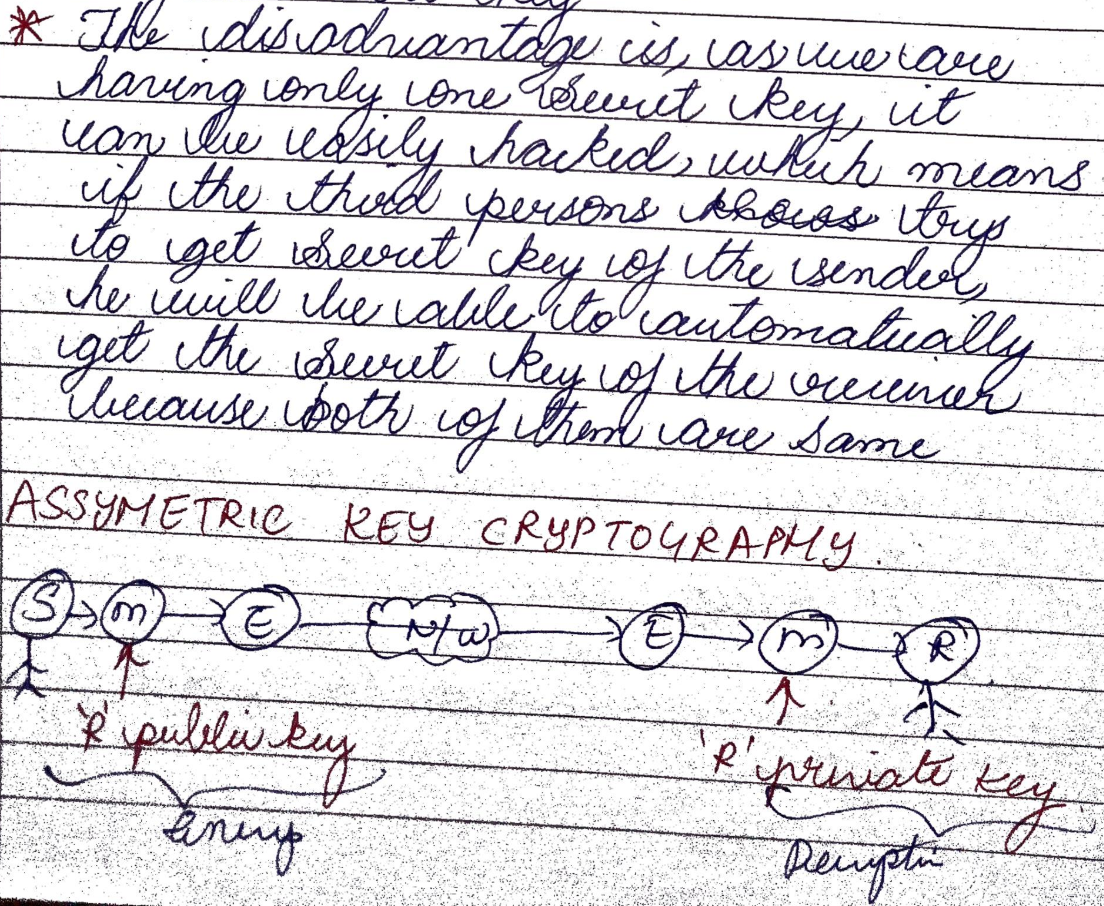


* Here we use only one key for both sender and receiver
* That one key is used for both encryption process as well as decryption process
* A sender wants to send a messay say "Hello" to the receiver,
* After generating the message, the message needs to be encrypted.
* And this encryption process is done with the help of that key

_/_/_

* The encrypted message will be entering the network (Transmission media)
* Then it reaches the receiver
* The receiver can't understand the encrypted message so he has to decrypt the message into plain text again
* In order to decrypt this encrypted message, you have to apply the same secret key which the sender used on this encrypted message and he will decrypt it and get the normal message
* The advantage is it can be easily implemented because of there is only one secret key
* The disadvantage is, as we are having only one secret key, it can be easily hacked, which means if the third persons tries to get secret key of the sender, he will be able to automatically get the secret key of the receiver because both of them are same

ASSYMETRIC KEY CRYPTOGRAPHY



diff

* Here, we use 2 keys, one for sender & other for receiver
* One is public key and other is private key
* public key is the key known to everybody
* private key, everyone won't know
* First a sender will generate a message that he wants to send to the receiver
* After that, the sender has to encrypt the message
* We will be using Receiver's public key to encrypt the message
* After the encrypted message is generated, this encrypted message will be transmitted on the receiver side through the network
* On reaching receiving side, this encrypted message should be converted into normal message
* On using private key of R on this encrypted message it will be converted into normal msg which is in readable format received & read by the receiver. This process is called decryption
* ∴ In this process we have additional security compared to symmetric key cryptography

STEGANOGRAPHY

* Basic idea → information hide / covered writing

* It is the practice of concealing messages / file / image (any type of information) within another file, message or image / video

* Note → later, we will extract it at its destination

* It is derived from Greek words Steganos meaning covered or concealed & graphia which means writing

* Steganography is diff from cryptograp but, using both together can improve security for protected data / info & prevent the detection of secret communication

* In cryptography, we make the data unreadable (by encryption). In Steganography, we are hiding the sentence of data

* Various forms of Steganography are
1) Text Steg---
2) Audio "

3) video St - - -
4) images St - -

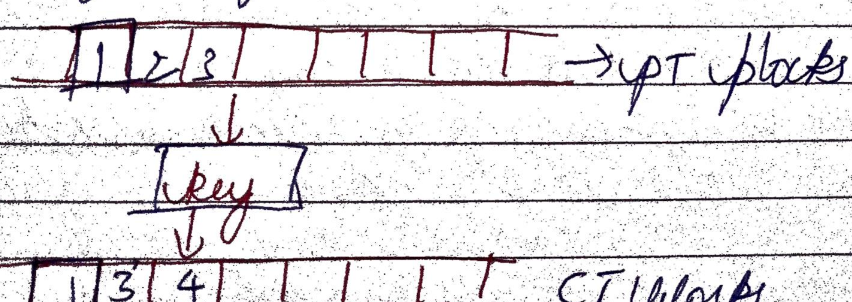
∴ In short, steganography can be used to hide/conceal any type of digital content (including text, image, video, audio)

UNIT-2

BLOCK CIPHER PRINCIPLES AND ALGO'S

Block cipher.

* Our main motto is to convert PT to CT so that the attacker dosent understand what we are sending
* Here we divide PT into no of blocks.
* After that we convert each individual block into CT each CT block
* Block size → (40, 56, 64, 128, 256 bits)
* Also make sure that PT block size = CT block size.
* The conversion of PT to CT will happen through key



* Then we combine CT blocks

Block Cipher Principles
(design principles)

1) No of Rounds (eg 10R, 16R, 20R → hard)
2) Design of function F (Non linear func)
3) Key Schedule Algo · abcd abc

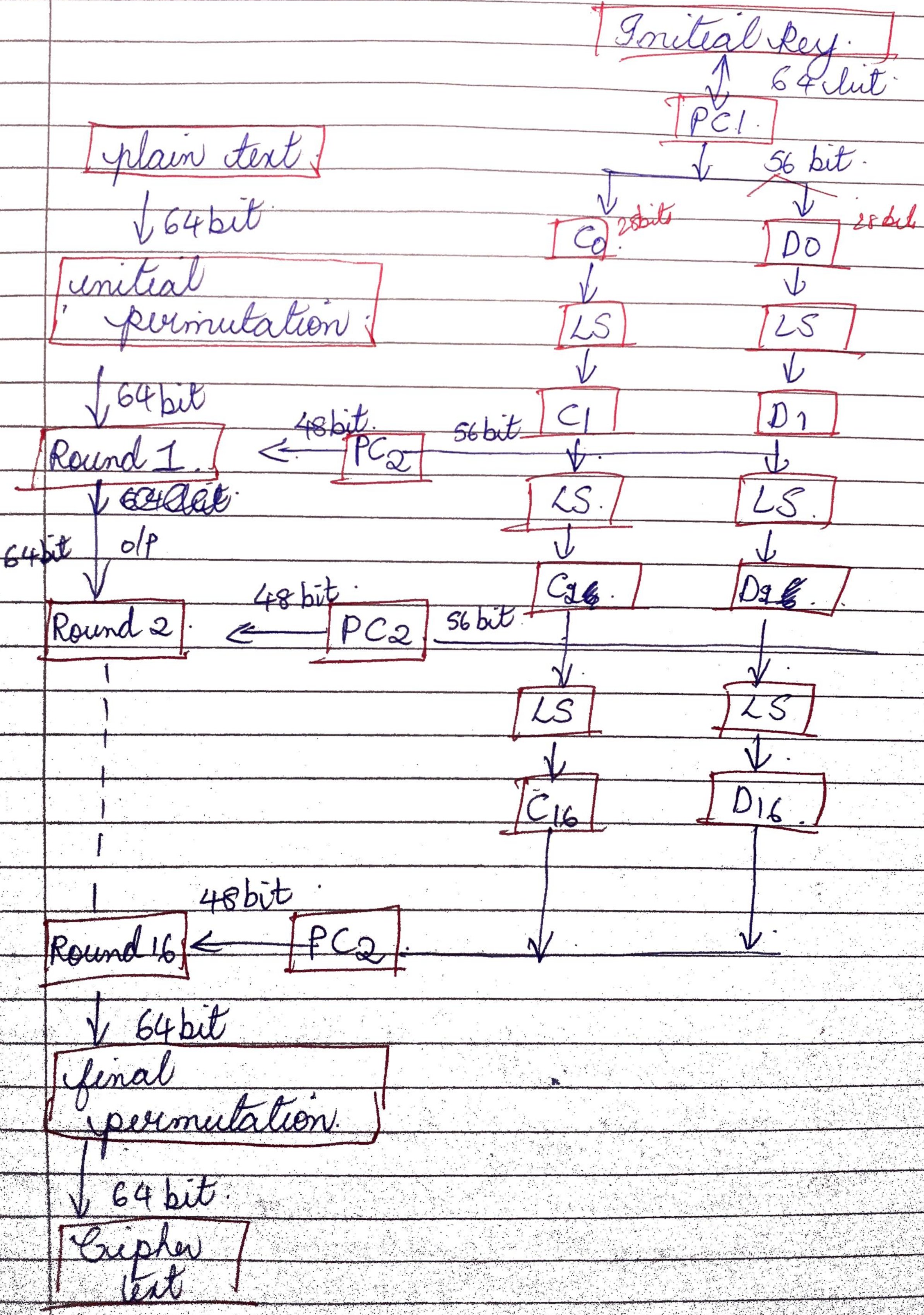
Block cipher modes of operations:
ECB, CBC, CFB, OFB, CTR modes

Block cipher Algos:
DES, AES, Blowfish

DES ALGORITHM.
data Encryption Algorithm.

* It comes under block cipher algo
* Converts plain text to cipher text
* has total of 16 rounds
* Size of plain Text = 64 bits ∴ CT = 64
  Key Size = 48 bits (8 bits for parity &
  -8 bits for rearrangt)
* In each round 4 steps are performed

① Dividing bits
② parts 32 bits each
③ Bit Shuffling
④ Non linear substitutions
⑤ Exclusive OR Operations


Initial key
64 bit
PC1
56 bit
C0
28 bit
D0
28 bit
LS
LS
C1
D1
LS
LS
C2
D2
LS
LS
C16
D16
plain text
64 bit
initial permutation
64 bit
Round 1
48 bit
PC2
56 bit
64 bit
o/p
Round 2
48 bit
PC2
56 bit
Round 16
48 bit
PC2
64 bit
final permutation
64 bit
Cipher text

In PC1,

- Initially 64 bits, 8 parity bits are to be removed from 8th position
  64 = (8×8) i.e 56.
- Then apply left circular shift after dividing 56 bits into 2 parts. C_0 & D_0 each having 28 bits
- D_1 & C_1 are obtained as a result.
- Left circular shift ?
  move the bits based on round no's.
  for rounds 1,2,9,16, → ① bit shift
  other rounds — ② — bit shift

In PC_2

- C_1 and D_1 are combined to form 56 bits again
  Permuted choice 2 is applied.
  56 bits are rearranged permuted &
  48 bits are selected
  ↓
  key for round ①

Round: i/p — 64 bits + 48.

AES Algorithm.
Advanced Encryption Standard

- has i/p array, state array and a key array.

AES Encryption & Decryption.
It has block cipher array algo

* Input Array (4×4)

8	8	8	8
8	8	8	8
8	8	8	8
8	8	8	8

each cell = 1 byte / 8 bits.
Total = 16 cells.
16×8 = 128 bits.
= 4 words (32 each)

PT is represented in the i/p array.

* State Array:-

word / byte

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

used to store intermediate states within the rounds.

Total ④ words.

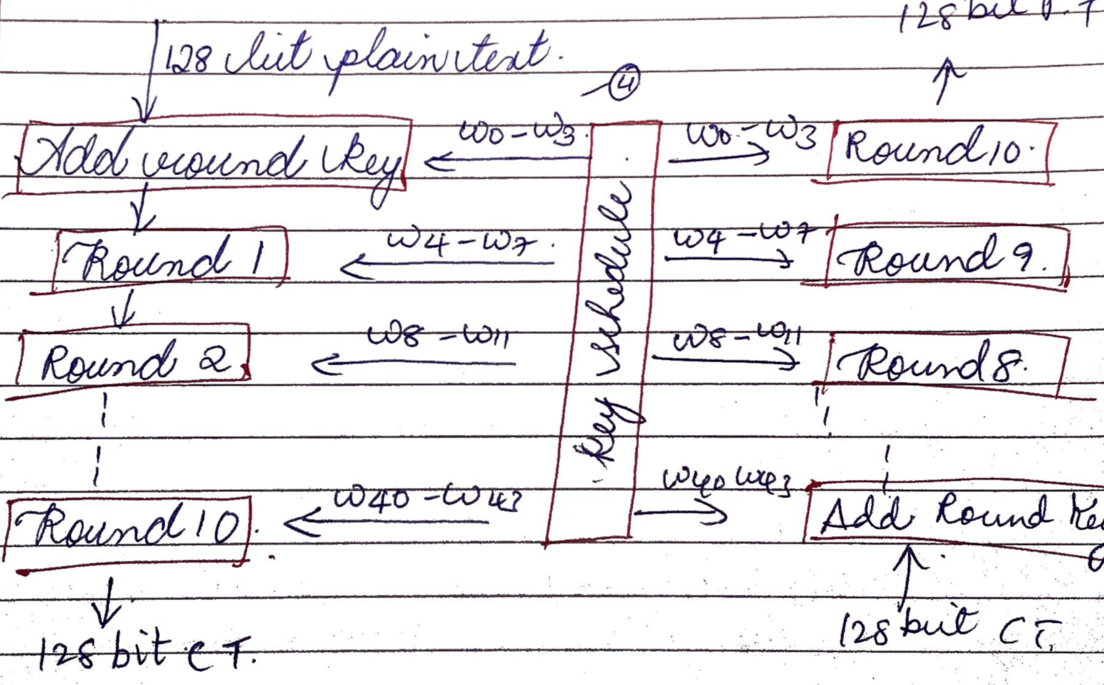
* Key Array:- Actually ④ words.
They are expanded into (44) words.
Each Round = 4 words.
∴ 10 Rounds × 4 words.
40 + 4 (for Add Round Key).
= 44 words

K0	K4	K8	K12
K1	K5	K9	K13
K2	K6	K10	K14
K3	K7	K11	K15

→ | w0 | w1 | w2 | ---- | w41 | w42 | w43 |

4 words → 44 words

Diagram :-



Encryption   Decryption

No of Rounds = 10 (for encry & decryp)

In each round (4 steps)

1) Substitute bytes.
2) Shift rows (LCS)
3) mix columns - Not in Round 10.
4) Add Round key
   ↓
   XOR operation b/w PT & key

Blowfish Algo

- Block Cipher Algo.
- Symmetric Key Cryptography.
- ip size = 64 bits.
- key size = variable length key
  (from 32 to 448)

properties:-

- fast.
- Takes less memory.
- Simple to understand & implement.
- more secured (bcz of var length key)

blowfish algo has 2 parts.

① Key Generation.
② Data Encryption.

* Key Generation.

① keys are stored in an array.
$K_1, K_2, K_3 \dots K_n$ $[1 \le n < 14]$.
↓
length of each block = 32 bits.
$(32 \times 14 = 448 \text{ bits})$

② Initialize an array (P)
$P_1, P_2, P_3 \dots P_{18}$.
↓
length of each word = 32 bits

③ Initialize S-boxes (4)

$S_1 \Rightarrow S_0, S_{11} \ldots S_{255}$ (Substitu- Boxes)
$S_2 \Rightarrow S_0, S_{11} \ldots S_{255}$
$S_3 \Rightarrow \ldots \quad \text{''}$
$S_4 \Rightarrow \ldots S_{255}$

(4) Initialize each element of P array & S-boxes with hexadecimal value

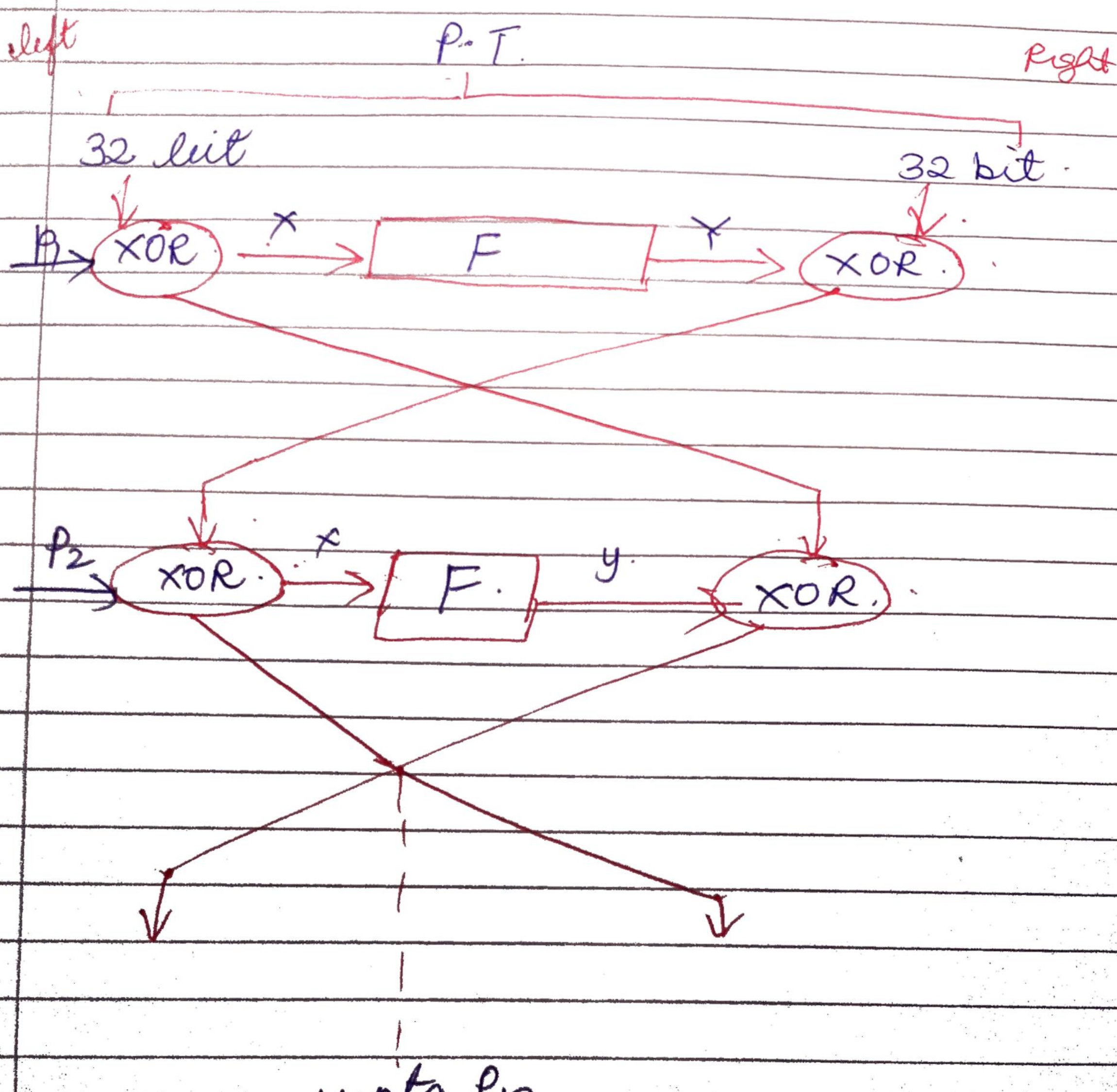
(5) XOR operations are performed.

$P_1 = P_1 \text{ XOR } K_1$
$P_2 = P_2 \text{ XOR } K_2$
$\vdots$
$P_{14} = P_{14} \text{ XOR } K_{14}$
$P_{16} = P_{15} \text{ XOR } K_1$
$\vdots$
$P_{18} = P_{18} \text{ XOR } K_4$

(6) Take 64 bit P.T (Initially all bits are 0).
(0, 0, 0 . . . . 0).
Subkey is generated.

* Data Encryption

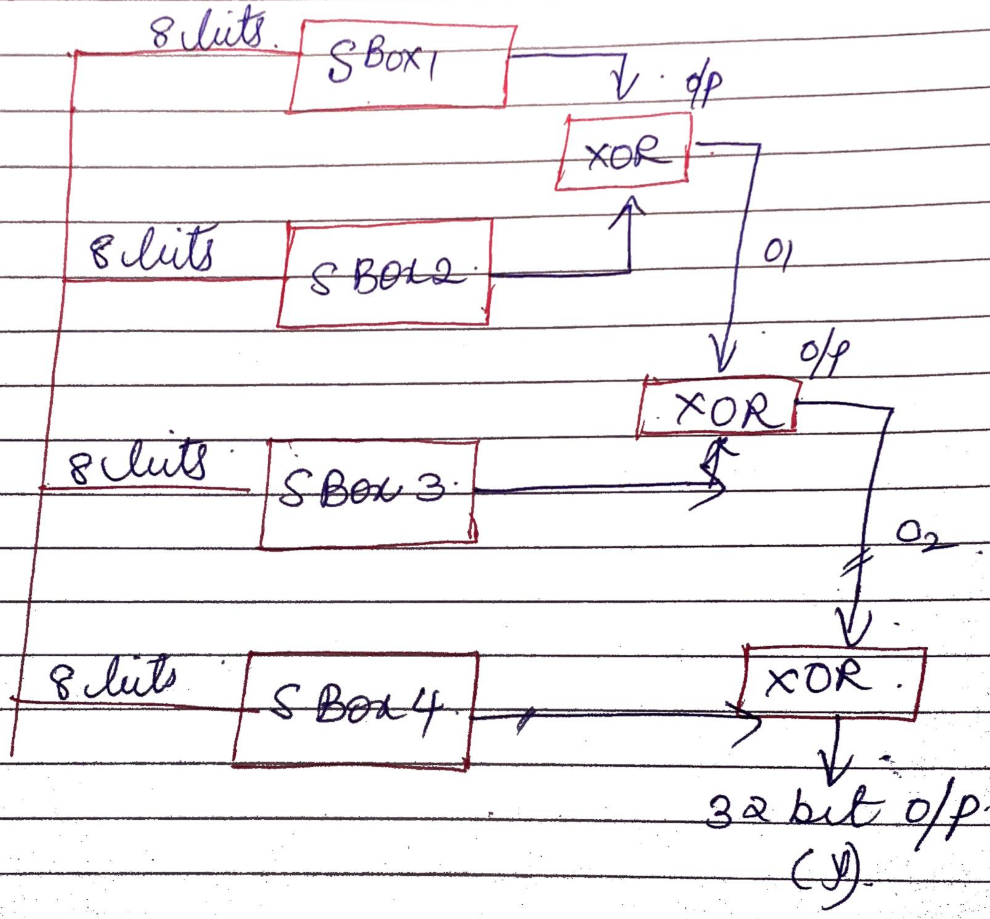
__/__/__



upto P18

Cipher Text is generated.
(64 bits)

* Data Encryption function.



IDEA ALGORITHMS
International Data Encryption Algorithm

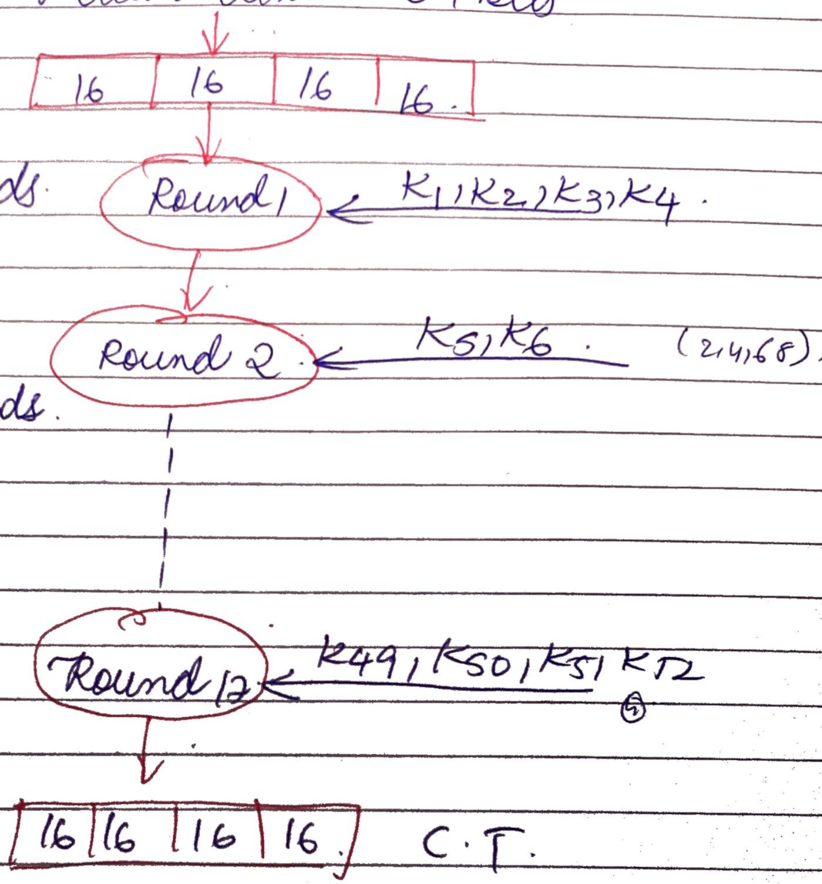
- Block Cipher Algorithm
- Symmetric Key Cryptography
- Feistal Cipher
- i/p size = 64 bits - 16, 16, 16, 16
- Key size = 128 bit into 52 sub keys

- No of Rounds = 17.

Plain text = 64 bits

Odd Rounds.
4 keys.

Even Rounds.
2 keys.

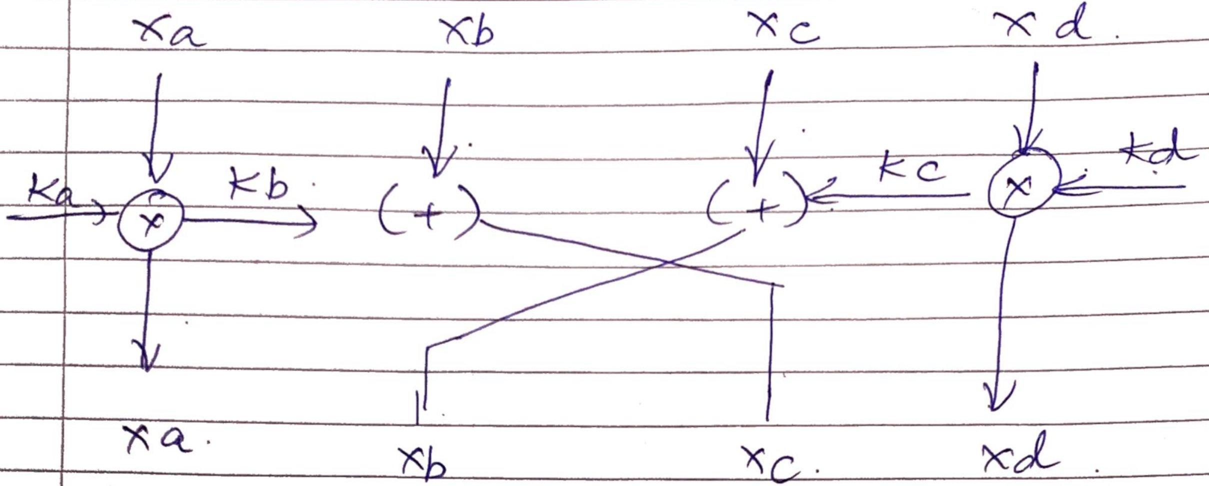


Rounds

Odd Rounds:- i/p = ④ parts.
keys = ④.

x_a, x_b, x_c, x_d.
k_a, k_b, k_c, k_d.
16 16 16 16

__/__/__



* Even Rounds.

i/p = (4) parts.
keys = (2).

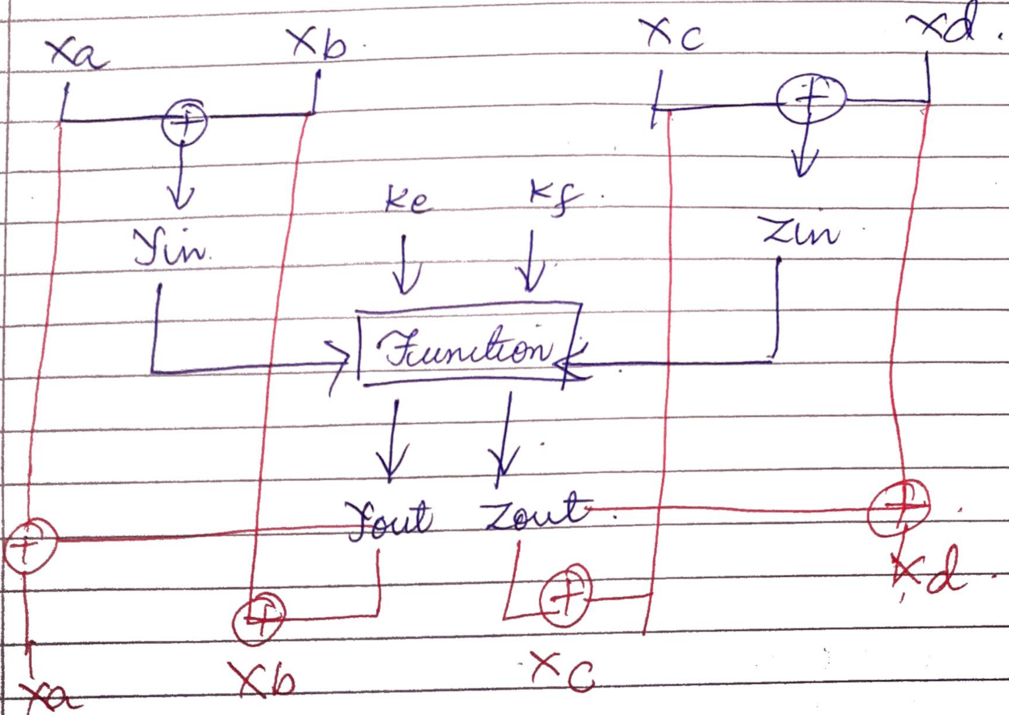
Xa ⊕ Xb   Xc ⊕ Xd.

Ke, Kf.

i/p = 4 but key = 2 ∴ Take (2) parameters.

Yin = Xa ⊕ Xb.
Zin = Xc ⊕ Xd.

__/__/__



xa = xa ⊕ Yout

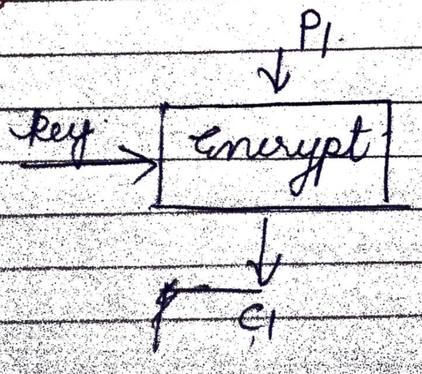
xb = xb ⊕ Yout

xc = xc ⊕ Zout

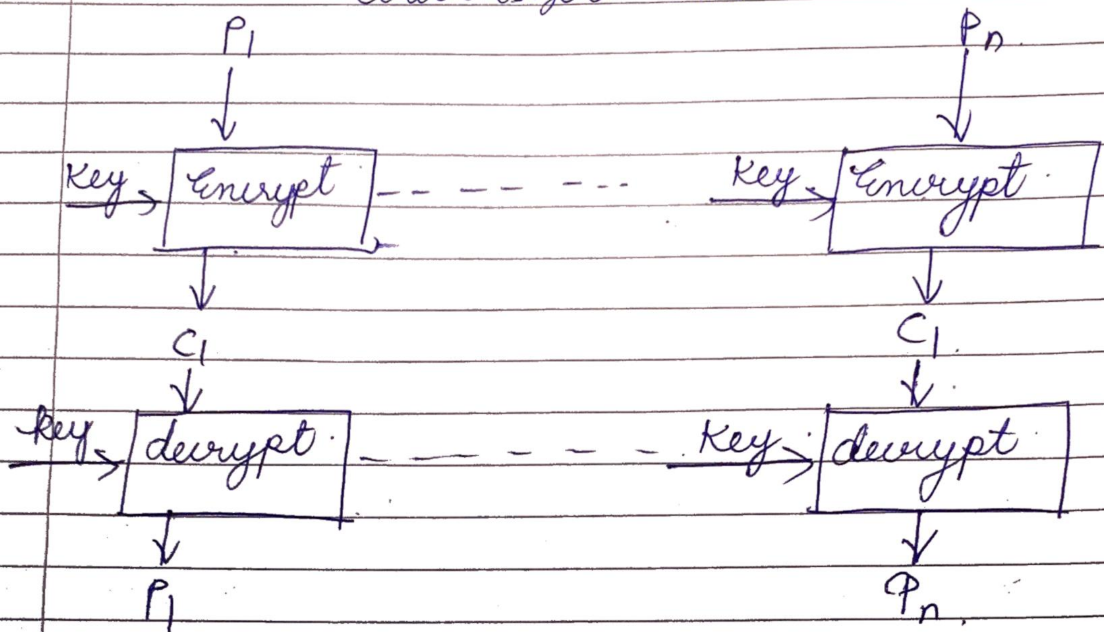
xd = xd ⊕ Zout

BLOCK CIPHER MODES OF OPERATION.
(5 modes)

① Electronic Code Block : (ECB



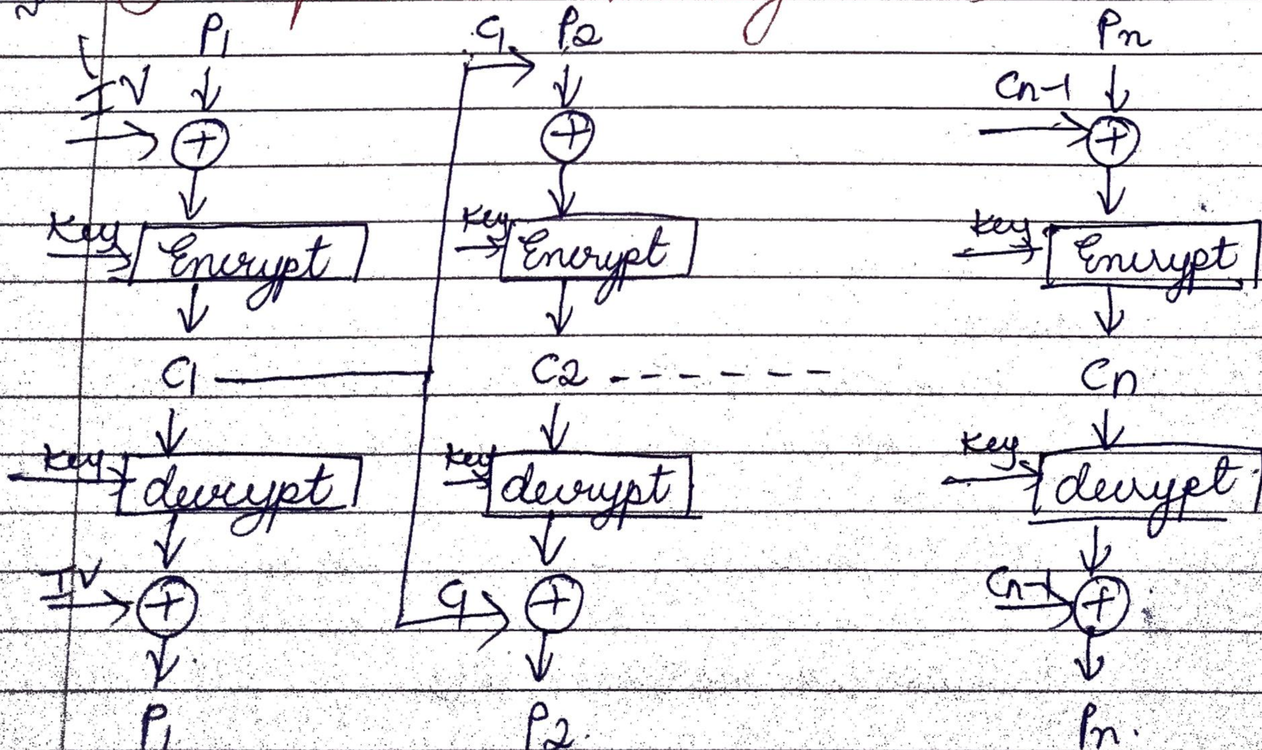
Block size = 64 bits



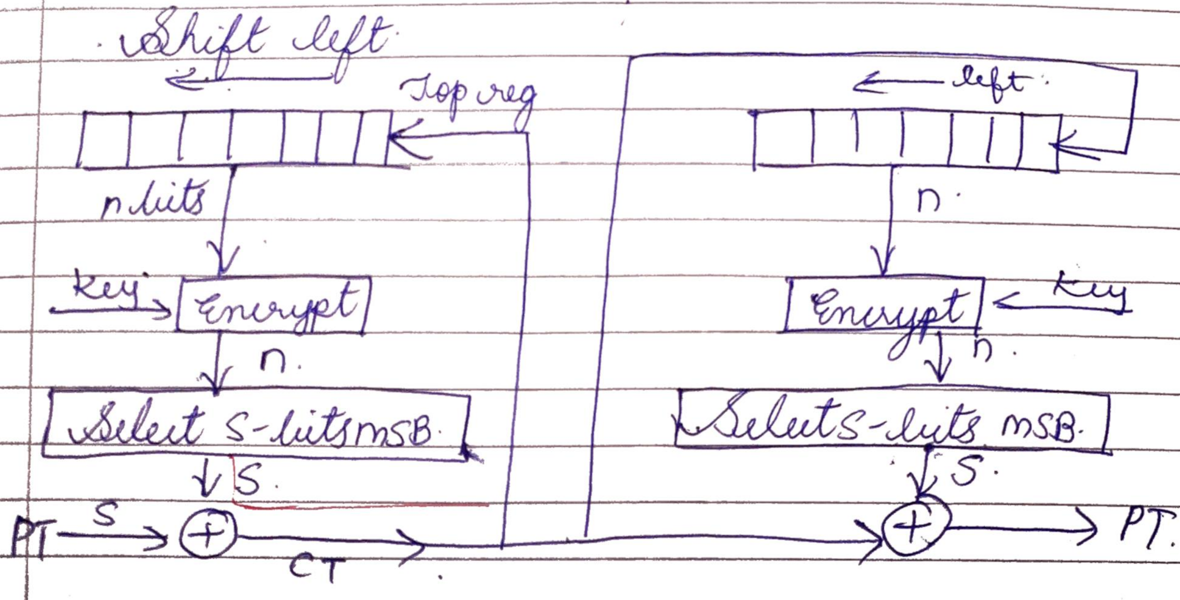
∴ This is suitable for short messages

Init vector

(2) Cipher Block Chaining (CBC)



③ Cipher Feedback Mode (CFB)



Encryption   Decryption.

- Both side we use encrypt func only.
- Top register → filled with IV

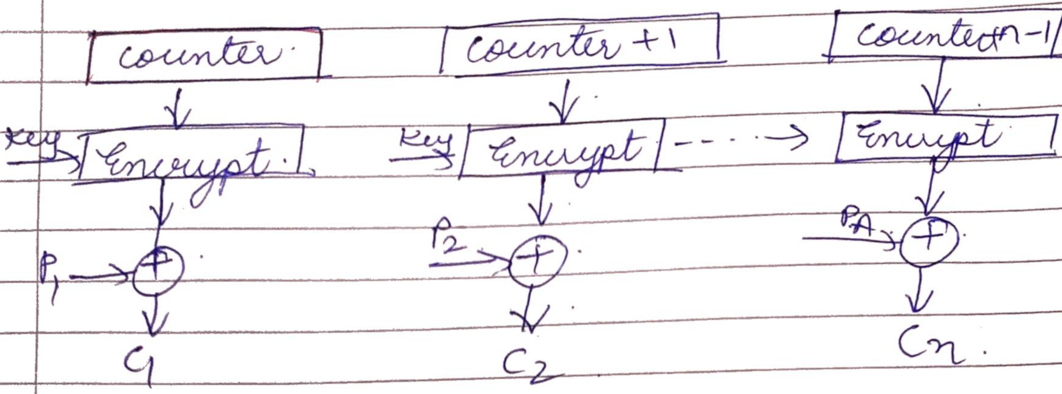
④ Output Feedback Mode (OFB).

- Same as Cipher Feedback Mode.
but
instead of Cipher text, Output is given as feedback.

- O/p refers to S-MSB bits

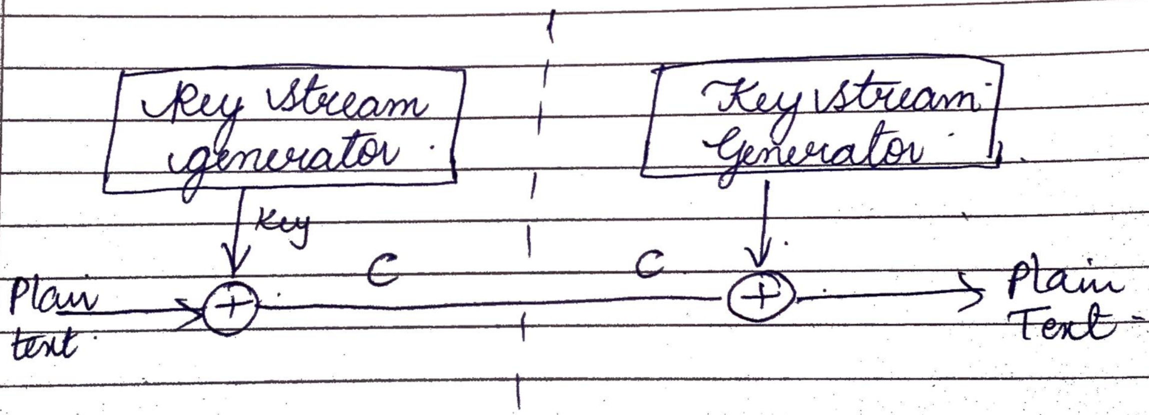
⑤ Counter mode (CTR)

Counter size = Plain Text size



STREAM CIPHER :-

Plain Text is divided into no of stream



- BITWISE XOR is performed
- considers each unit one by one

Eg:-

m_1	m_2	m_3	....	m_i	→ P.T.
k_1	k_2	k_3	....	k_i	→ Keys
c_1	c_2	c_3	....		→ C.T.

(CT ⊕ Key ⇒ PT) decryption

	c_1	c_2	c_3	c_4	....	c_i
⊕	k_1	k_2	k_3	k_4	....	k_i
	P_1	P_2	P_3	P_4	....	P_i

PT	1	1	0	0
key.	1	0	1	1
CT	0	1	1	1

perform XOR.
diff = 1
same = 0.

⇓

CT	0	1	1	1
key	1	0	1	1
PT.	1	1	0	0

RC4 ALGO.

– Stream Cipher Algo.

Procedure:-

1. Uses an array (S) – State vector of length 256 (0 – 255).
2. It has a key encoded with ASCII.
3. It has a key array of length 256 (0 – 255)

Steps

1. Key Scheduling.
2. Key stream Generation
3. Encryption & Decryption

__/__/__

Key Scheduling :- no of iterations = size of S-array

j = 0
for i = 0 to 255 do
  j = [j + S(i) + T(i)] mod 256.
  swap (S[i], S[j]);

Here,
S[i] → State vector.
T[i] → Key array.
(Temp vector)

Example :-

	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
S-array =	0	1	2	3	4	5	6	7

Key array = [1 2 3 6]
Plain Text = [1 2 2 2]

Initialize T-array with Key.

T = [1 2 3 6 1 2 3 6].

(1) j = 0.
for i = 0 to 7.
j = [0 + 0 + 1] mod 8.
= 1 mod 8 = 1 ⇒ j = 1
swap S(0) & S(1).
S = [1 0 2 3 4 5 6 7].

(2) for i = 1
j = (1 + 0 + 2) mod 8.
= 3 mod 8 = 3 = j
swap S(1) & S(3).
S = [1 3 2 0 4 5 6 7].

3) for i = 2.
j = (3 + 2 + 3) mod
= 8 mod 8 =
∴ j = 0
swap S(2).

Key Stream Generation

No of Iterations = size of key.

```
i, j, = 0;
while (true)
 i = (i+1) mod 256;
 j = (j+S[i]) mod 256;
 swap (S(i), S[j]);
 t = (S[i] + S[j] mod 256;
  k = S[t];
```

New key is obtained.
 (used for encryption and decryption)

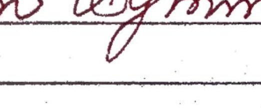
Encryption & Decryption

Enc – PT XOR New Key.
 (first convert into binary).

dec – CT XOR New Key.

Key Distribution (In Symmetric key):
 (4) ways

1. Physical delivery (most secured, more time)
2. Key Distribution Centre (KDC)
3. Using Previous Keys encrypt old → new
4. Using third Party



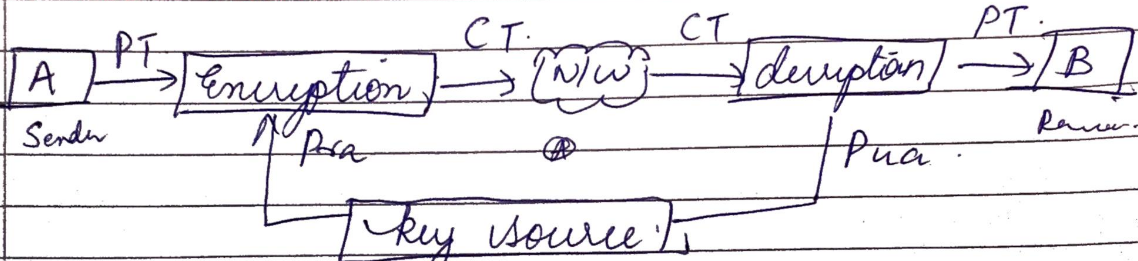
→ generate & send (S&R) less time, authentic, 3rd Party

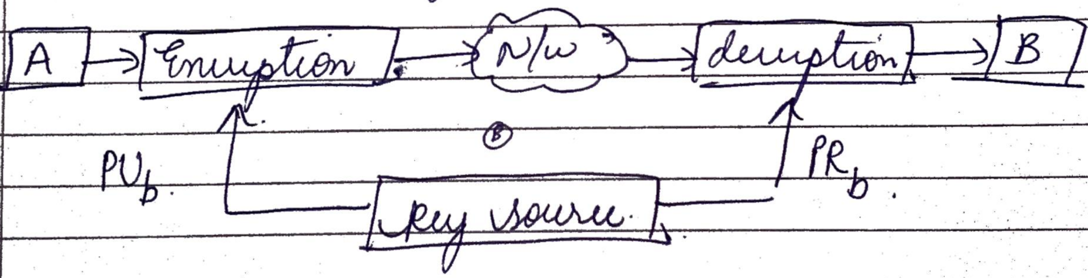
UNIT-2 - part 2.

Principles of Public Key Cryptosystems (Asymmetric Key Cryptography)

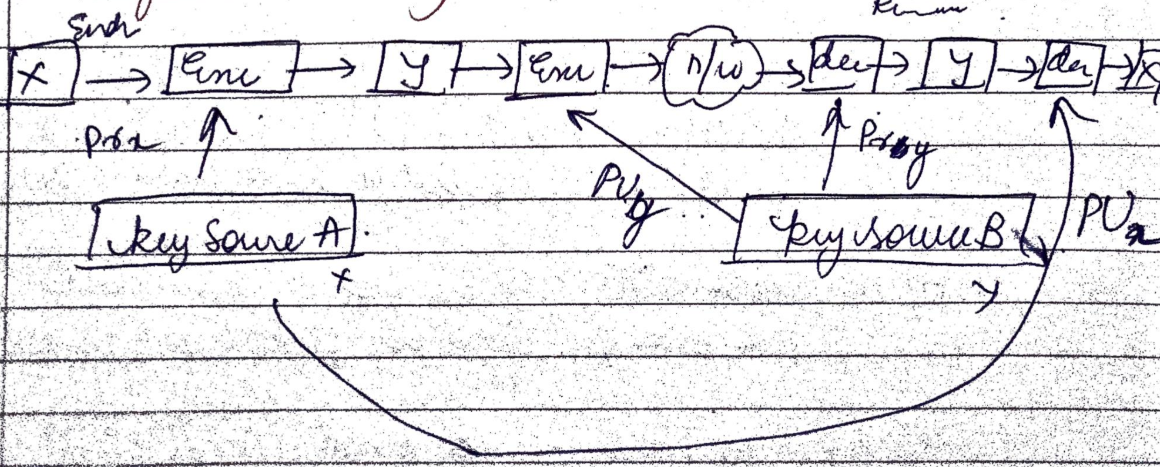
There are 2 principles:
① Authentication
② Confidentiality

① Authentication





② Confidentiality



RSA Algorithm
(Rivest Shamir Adleman).
- asymmetric key Algo & Block Cipher Algo

③ steps :-

① Key generation.
② Encryption.
③ Decryption.

Key Generation.

① Select ② large prime no's P & q.
↓
(for more security).

$P = 3$ and $q = 11$

② Calculate $\boxed{n = p * q}$ ⇒ $n = 3 \times 11 = 33$.
$\boxed{n = 33}$.

③. Calculate $\boxed{\phi(n) = (P-1)(q-1).}$
$\phi(n) = (P-1)(q-1)$
$\phi(n) = (3-1)(11-1) = 2 \times 10 = 20$

④ Choose the value of e such that.
$1 < e < \phi(n)$ and $\gcd(\phi(n), e) = 1$.

Let $e = 7$ ⇒ $1 < 7 < 20$ and $\gcd(20, 7) = 1$ ∴ $\boxed{e = 7}$

(5) Calculate $d = e^{-1} \bmod \phi(n)$.

$ed = 1 \bmod \phi(n)$.

$ed \bmod \phi(n) = 1$.

$\Rightarrow ed \bmod \phi(n) = 1$

$7 \times d \bmod 20 = 1$.

$7 \times 3 \bmod 20 \Rightarrow 21 \bmod 20$

$\therefore \boxed{d = 3}$.

(6) public key = $\{e, n\} \Rightarrow \{7, 33\}$

(7) private key = $\{d, n\} \Rightarrow \{3, 33\}$.

Encryption.

$\boxed{C = m^e \bmod n}$

M = no of digits in PT. (Assume $\boxed{M < n}$.)

C = CT.

Let $\boxed{m = 31}$.

$c = (31)^7 \bmod 33$

$= 4$.

$\therefore \boxed{C = 4}$.

Decryption.

$\boxed{m = c^d \bmod n}$.

$m = (4)^3 \bmod 33$

$= 64 \bmod 33 = 31 \therefore \boxed{m = 31}$

Diffie - Hellman Key Exchange Algo

- not an encryption / decryption Algo.
- used to exchange keys b/w sender and receiver
- Asymmetric key cryptography.

Procedure.

① consider a prime no. q.
let $q = 7$.

② Select α such that $\alpha < q$ and α is primitive root of q

primitive root - ?

$\alpha^1 \bmod q$
$\alpha^2 \bmod q$
$\alpha^3 \bmod q$
⋮
$\alpha^{q-1} \bmod q$ should have values $\{1, 2, 3 .. q-1\}$.

Ex.
$\alpha = 3$ & $q = 7$.
$3^1 \bmod 7 = 3$.
$3^2 \bmod 7 = 2$
$3^3 \bmod 7 = 6$.
$3^4 \bmod 7 = 4$.
$3^5 \bmod 7 = 5$.
$3^6 \bmod 7 = 1$.
$= \{7, 3, 2, 6, 4, 5, 1\}$
$(1, 2, 3, 4, 5, 6)$
$\therefore$ 3 is primitive root of q

③ Assume X_A (Primitive key of A)

=

③ Assume XA (private key of A) and $X_A < q$.

calculate $Y_A = \alpha^{X_A} \bmod q$

Ex = q = 7 & α = 5
and let XA = 3.   YA = public key of A.

$Y_A = (5)^3 \bmod 7 = 125 \bmod 7 = 6$.

$Y_A = 6$

④ Assume XB and $X_B < q$.
Calculate $Y_B = \alpha^{X_B} \bmod q$.

let $X_B = 4$,

$Y_B = (5)^4 \bmod 7 = 625 \bmod 7 = 2$.

$Y_B = 2$.   XB = Pvt key of B
YB = Pub key of B

⑤ Calculate secret keys K1 & K2 for exchanging.

K1 ⇒ Person A & K2 ⇒ Person B

$K_1 = (Y_B)^{X_A} \bmod q$ $K_2 = (Y_A)^{X_B} \bmod q$

After Calculating, if $K_1 = K_2$ then Success.

$K_1 = (2)^3 \bmod 7 = 8 \bmod 7 = 1 \Rightarrow \boxed{K_1 = 1}$.

$K_2 = (6)^4 \bmod 7 = 1296 \bmod 7 = 1 \Rightarrow \boxed{K_2 = 1}$.

$K_1 = K_2$ ∴ success.

Key exchanged successfully

Elgmal Algorithm.

Asymetric Key Cryptography.

Steps:

1. Key generation.
2. Encryption.
3. Decryption.

① Key generation

1. Select large prime no's (P) $\boxed{P = 11}$
2. Select a dec. key also called private key $\boxed{d = 3}$
3. Select 2nd part of encry key $(e_1) = 2$. $\boxed{e_1 = 2}$.
4. Select 3rd " " " (e_2).

$$e_2 = e_1^d \bmod p.$$
$$= (2)^3 \bmod 11 = 8 \bmod 11 = 8$$
$$\boxed{e_2 = 8}$$

5. public key = (e_1, e_2, p)
   private key = d.

$\boxed{\text{pub key} = 2, 8, 11}$

_/_/_

② Encryption:

① Select random Integer R $\boxed{R=4}$.

② calculate $C_1 = E_1^{r} \bmod P = 2^4 \bmod 11 = 16 \bmod 11 = 5$

$\boxed{C_1 = 5}$.

③ calculate $C_2 = (PT \times e^{2R}) \bmod P$ PT = Assume (2)

$= (7 \times 8^4) \bmod 11$.

$= 28672 \bmod 11 = 6$.

$\boxed{C_2 = 6}$.

④ CT = Cer $(C_1, C_2) = (5, 6)$.

③ Decryption:

① $PT = [C_2 \times (C C_1)P)^{-1}] \bmod P$.

$= (6 \times (C5)^3)^{-1}) \bmod 11$.

$= 6(5^3)^{-1} \bmod 11$

$= 6(125)^{-1} \bmod 11$.

$= 125 \times x \bmod 11 = 1$.

If $x = 3$, $125 \times 3 \bmod 11 = 375 \bmod 11 = 1$.

$\therefore \boxed{x = 3}$

$6 \times 3 \bmod 11 = 18 \bmod 11 = 7$

Key Distribution in Asymetric Key (public key Cryptography).

4 ways

1. public Announcement. [A] → broadcast key to all users.
2. public key discovery directory (telephone dir)
3. public key Authority (TTP) — user B → PU dir; 3rd: (A) → Q → (B). user R.
4. Certificate Authority TTP. → id + pub key of user

Knapsack Algorithm

- My Hellman.
- Asymetric key cryptography

Ex: weights = (1, 6, 8, 15 and 24).
In general Knapsack, we select weights to achieve a sum.
If we want a sum = 30.
we Select 1, 6, 8 and 15.

let plainText = 1 0 0 1 1.   1 1 0 1 0.

1 0 0 1 1	1 1 0 1 0
× 1 6 8 15 24	× 1 6 8 15 24
1+15+24 = 40	1+6+15 = 22

CT = PT × corresponding wts
∴ CT = 40   2 2.

Key generation

① public key (Hard Knapsack)
② private key (Easy Knapsack).
↓
done first (i.e we find pvt key first)

Example :-

{1, 2, 4, 9, 20, 40}.
weights are always in increasing order.

① first find private key (Assume)
D = {1, 2, 4, 10, 20, 40} — Pvt key.

Select ② no's "n" and "m"
⇒ m > sum of all no's in sequence.

Sum = 77 ∴ let $m = 110$

⇒ n = select so that it has no common factor with m.

let $n = 31$.

Now $(D_i \times n) \bmod m$ ∀ elements in D

$(1 \times 31) \bmod 110 = 31$.
$(2 \times 31) \bmod 110 = 14$.
$(10 \times 31) \bmod 110 = 90$.
$(20 \times 31) \bmod 110 = 70$.
$(40 \times 31) \bmod 110 = 30$

_/_/_

⇒ {31, 62, 14, 90, 70, 30}
⇓
public key.

* Encryption:

Now Assume PT. ① ② ③
let PT = 100100 | 111100 | 101110.

divide into 6-6 parts (no. of elements in sequence = 6).

1st part ⇒ 100100 = 1×31 + 0×62 + 0×14 +
1×90 + 0×70 + 0×30
= 31 + 90 = 121.

2nd part ⇒ 111100 = 31 + 62 + 14 + 90 + 0 + 0 = 197

3rd part ⇒ 101110 = 31 + 0 + 14 + 90 + 70 + 0 = 205.

∴ CT = [121 197 205]

* Decryption:

calculate $n^{-1} = 31^{-1}$.
31 · x mod 110 = 1 then we get x = 71.

(CT × x) mod m from seq D = {1, 2, 4, 10, 20, 40}

(121×71) mod 110 = 11 = 100100 (1+10 = 11)
(127×71) mod 110 = 17 = 111100 (1+2+4+10 = 17)
(205×71) mod 110 = 35 = 10110 (1+4+10+20 = 35)

UNIT–III

Message Authentication

Authentication–?

Verifying the identity of user (from correct person or not).

How it is done?
by authenticator
↓
generated by authentication function.

3 Authentication functions
① message encryption
② message Authentication code (MAC)
③ Hash Functions (H).

① message encryption
PT — CT
↓
acts as authenticator

② message authentication code.
$C(M,K) = o/p$ (fixed length code).

C = authentication function
m = message
K = key.
o/p > MAC code = acts as authenticator

③ Hash Function (H)

Similar to MAC but key ⇄ hash func

H(M) = fixed length code (Hash code h).

H - Hash function

h - hash code - acts as authenticator

MD5 (Message Digest - 5).

- developed by Rivest
- fast and produces 128 bit message digests

* Working of MD5.

① Padding

original message + Padding.
(so that total length is 64 bit less than exact multiple of 512)

Example :- Original msg = 1000 bits + (padding)

~~512 × 1 = 512 bits.~~ (512-64 < 1000)

~~512 × 2 = 1024 bits~~ (1024-64 < 1000)

512 × 3 = 1536 bits ✓.

1536 - 64 = 1472 (Tot length)

∴ Add 472. 1000 (OG).

1000 + 472 = 1472 bits

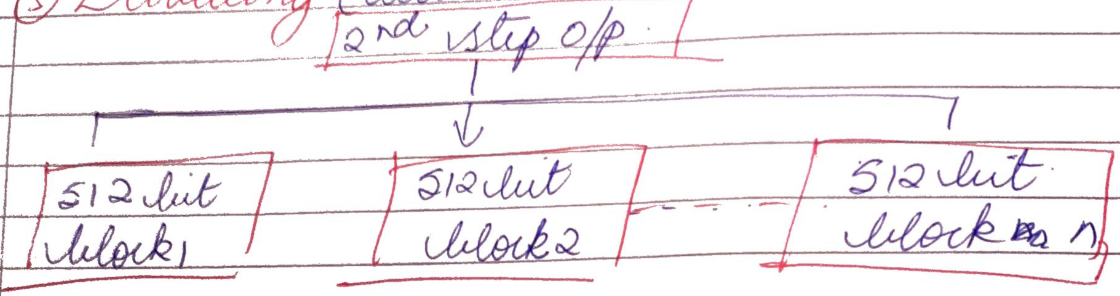
② Appending

Append the org lt before padding.
Calculate length mod 64.

most of the cases, 64 bits is obtained as answer.
(∵ append 64 bits)

So again it becomes multiple of 512

③ Dividing (each 512 bits)



④ Initialising : (chaining variable).

each 32 bit
Ⓐ, Ⓑ, Ⓒ & Ⓓ — values predefined.

⑤ processing (512 bit blocks)

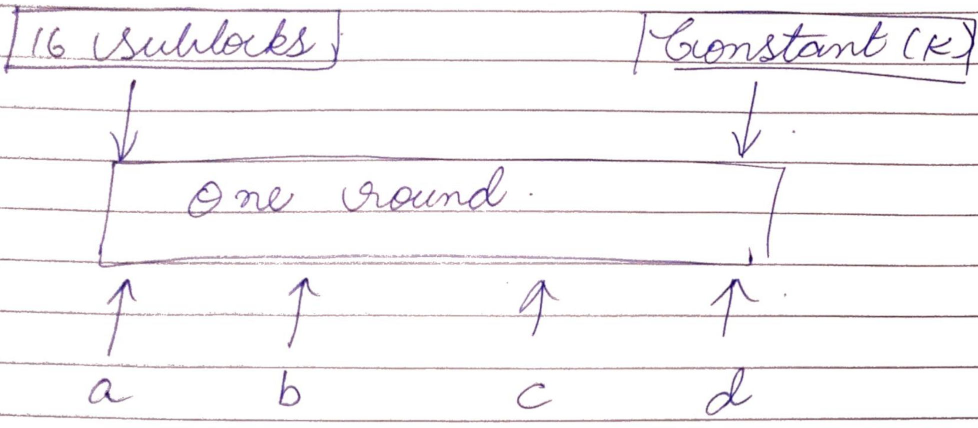
→ ① copy ④ chaining variables into some corresponding variables.

A = a, B = b, C = c, D = d.

→ ② Divide 512 bit blocks into 16 - 32 bit blocks

→ ③ Four rounds.
16 sub blocks & a constant (t)

___/___/___



a = b + ((a + process P(b, c, d) + m[i] + T[k]))

SHA Algorithm.
Secure Hash Algo.

- modified version of MD5.
In MD5 - length of o/p = 128 bits.
In SHA - length of o/p = 160 bits.

* working
1) padding - MD5 - 64 bit < (x) 512 ← Tot length.
2) appending - same - length mod 64 → (x).
3) divide the i/p into 512 bit blocks
4) Initialize ⑤ chaining variables
(A, B, C, D and E)
5) Process blocks.
- copy corresponding variables
A = a, B = b, C = c, D = d, E = e.
- divide into no of 512 bit block
(16 - 32)

- four rounds (each round = 20 steps

Message Authentication Code (MAC).

- Similar to message digest
- Symmetric key cryptography is used

*** working of MAC**

If sender wants to send a message M

M
↓ ⟵ Symmetric key (K). (M+K)
M1 (MAC code) CT

Now,

(PT) M + M1 > sent to Receiver
(CT)
↓
Calculate his own MAC (M2).
key ⇒ K (same one as M1)

Now
on receiver's side, M1 & M2 are compared.
M1 = M2 ⇒ no change in message.
M1 ≠ M2 ⇒ message is changed.

*** Significance of MAC**

1) Receiver can know if msg is changed / no
2) Receiver has assurance that msg is fr
correct sender bcz of same key for (S) & (R)

* HMAC : (Hash Based MAC).

— used in SSL.

* working of HMAC

original msg (m) ⟶ message digest is generated.

↓ (MD5 / SHA)

message digest ↓ ⟵ key (K).

↓

encryption

↓

MAC (CT)

In MAC — direct MAC is generated.
In HMAC — MAC is generated with the help of msg digest

* CMAC (Cipher Based MAC.

has message size limit

- based on block Cipher.
- given message is divided into equal no. of blocks and each block is encrypted separately.

Ex: 1 0 1 1 1.

A1	A2	A3	A4	A5
1	0	1	1	1
↓E	↓E	↓E	↓E	↓E
C1	C2	C3	C4	(C5) ⟶ acts as MAC

$C_1 = E(K, A_1)$

$C_2 = E(K, (A \oplus C_1))$

$C_3 = E(K, (A_3 \oplus C_2))$

$C_4 = E(K, (A_4 \oplus C_3))$

$\vdots$

$C_n = E(K, (A_n \oplus C_{n-1}))$

↓

Acts as MAC

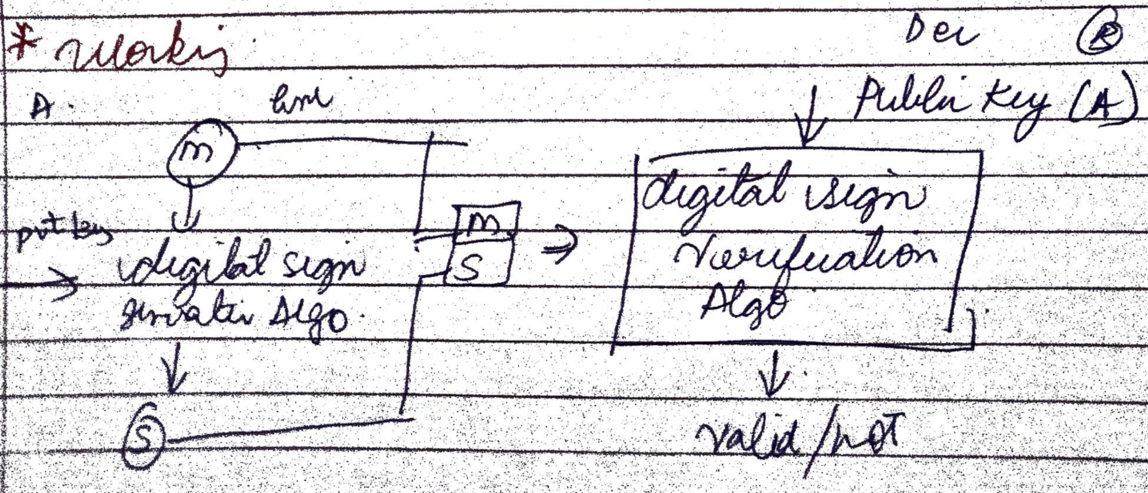
* Digital Signature -

- Asymmetric key Cryptography
- Encryption private key is used
- Decryption public key

* Used for authentication and Non Repudiation

* Signature : Proof of Identity (Is it from correct sender / not

* Working



If the message matches ⇒ valid
If the message not matching - not valid

~~Extensional~~ Elgamal Digital Signature

- digital Signature Scheme.
Encryption ⇒ Public key.
Decryption ⇒ Private key.

Working.

1) Select a prime no (q).
2) Select a primitive root (α) of q.
3) Generate a random integer (X_A).
$1 < X_A < q-1$.
4) Compute $Y_A = (\alpha)^{X_A} \bmod q$.
5) Generate keys for user (A)
private key $= X_A$.
public key $= \{q, \alpha, Y_A\}$.
6) Generate hash code (m) for the P.T (M)
$m = H(M) \quad 0 < m \leq q-1$.
7) Generate a random Integer (k).
$1 \leq k \leq q-1$ and gcd $(k, q-1)$.
8) Now calculate S_1 & S_2.
$S_1 = \alpha^k \bmod q$.
$S_2 = k^{-1}(m - X_A S_1) \bmod$.
9) Now we got the signature pair (S_1, S_2)

Now at user B's side,
Calculate V_1 & V_2

$V_1 = \alpha^m \bmod q$.

$V_2 = (Y_A)^{S_1} (S_1)^{S_2} \bmod q$.

If $V_1 = V_2$.

$\Rightarrow$ signature is valid.

If $V_1 \neq V_2$.

$\Rightarrow$ not valid.

Example:-

let $q = 19$ and $\alpha = 10$.

Now random Integer X_A $(1 < X_A < q-1)$.

$1 < X_A < 18$.

$X_A = 16$

$Y_A = \alpha^{X_A} \bmod q = (10)^{16} \bmod 19$

$= 4$

$Y_A = 4$

(A) Keys:- private key $\Rightarrow X_A \Rightarrow 16$

public key $\Rightarrow \{q, \alpha, Y_A\} \Rightarrow (19, 10, 4)$

Now generate hash code (m)

$m = H(M)$. $\quad 0 < m < q-1$

$0 \leq m \leq 18$.

$m = 14$

Generate (k), $0 \le k \le q-1$ and $\gcd(k, q-1) = 1$
$0 \le k \le 18$ and $\gcd(k, 18) = 1$
$\therefore$ $k = 5$

Calculate $S_1 = \alpha^k \bmod q = (10)^5 \bmod 19$
$= 3$

$S_1 = 3$

$S_2 = k^{-1}(m - X_A S_1) \bmod q-1$

$k^{-1} \Rightarrow k^{-1} \bmod q-1$ $\quad$ $5 \times 11 = 55$
$5^{-1} \bmod q-1$ $\quad$ $\frac{5 \times 4}{18} = 0$ Rem
$5 \times ? = 1 \pmod{18}$.

$\therefore k^{-1} = 11$

$S_2 = k^{-1}(m - X_A S_1) \bmod q-1$.
$= 11\ (14 - 16 \times 3) \bmod 18$.
$= -374 \bmod 18 = 4$.

$\therefore S_2 = 4$

$\therefore S_1, S_2 = (3, 4)$.

At B' end

$V_1 = \alpha^m \bmod q$.
$= 10^{14} \bmod 19 = 16$
$V_1 = 16$

$V2 = (YA)^{S_1} (S_1)^{S_2} \bmod q$

$= 4^3 \times 3^4 \bmod 19$

$= 5184 \bmod 19 = 16$

$V2 = 16$

Now $V_1 = V_2$

Signature is Valid

Public Key Infrastructure

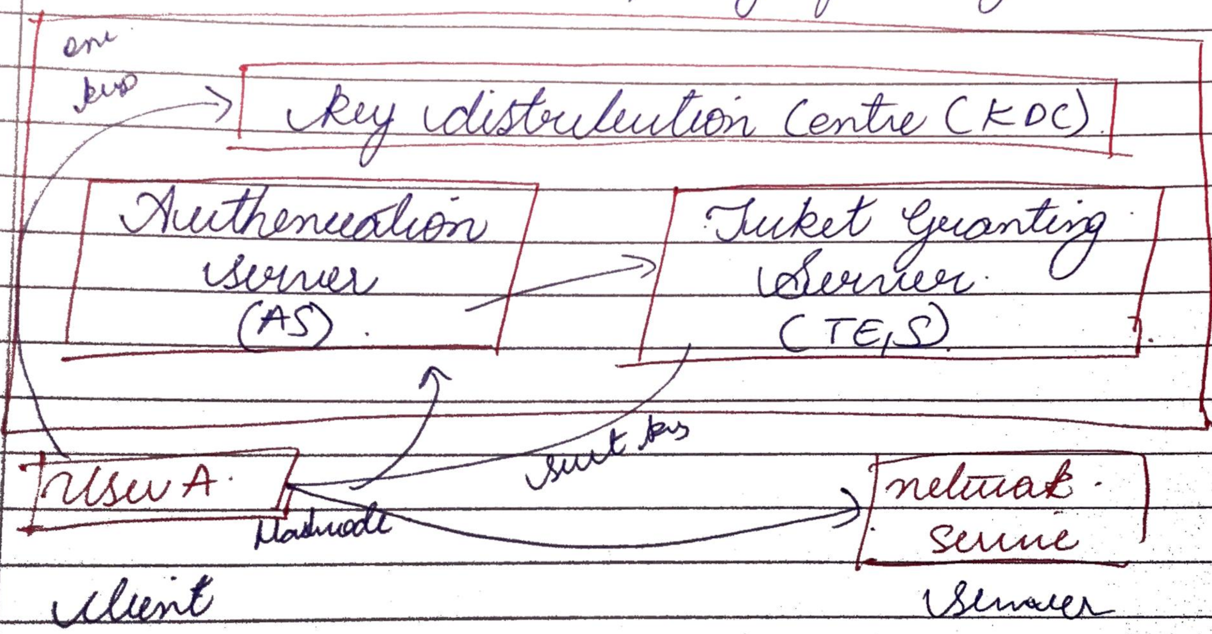
- Standard followed for managing storing and revoking the digital certificate
- follows asymmetric key cryptography.
- Includes message digests, Integrity
  digital signatures Authentication
  Non Repudiation
  Encryption services (confidentiality)

Architecture of PKI: - ④ parts

1) Certificate Repository
2) Entity
3) Registration Authority (RA)
4) Certificate Authority (CA)

Kerberos:-

- It is n/w authenticated protocol
- client server architecture
- Symmetric key
- requires a 3rd party for key (KDC)



X-509 Authentication Service.

- digital signature accepted internationally
- does not generate any keys. but provides a way to access public keys.

There are several elements in X509 certificate

It has ③ versions

Versions (1,2,3)
Serial number
Signature Algorithm Identifier
Issuer Name
Validity period
Subject Name
public key Info
Issuer Unique Id
Subject Unique Id
Extensions

version1
version2
version3

UNIT-~~3~~4

Web Security