Advanced Encryption Standared. (AES) Algorithm

* AES does not follow Feistel structure.

* The Plain Text size is 128 bits
The ~~cy~~ cipher Text is 128 bits.

* The size of key is not fixed.
There are three diff. keys sizes.

Key Sizes are 128 bits, 192 bits and 256 bits.

* No. of Rounds

No. of Rounds depends on size of the key

- If the size of key is 128 bit, no. of rounds = 10.
- If the size of key is 192 bits, no. of Rounds = 12
- If the size of key is 256 bits, no. of Rounds = 14.

Based on PT, Key, no. of Rounds, we discuss the Block diagram of AES Algorithm,

Before going to the Block diagram, First the Plain Text 128 bits is represented in a 4x4 column major matrix.

PT = 128 bits → 4x4 Column major matrix.

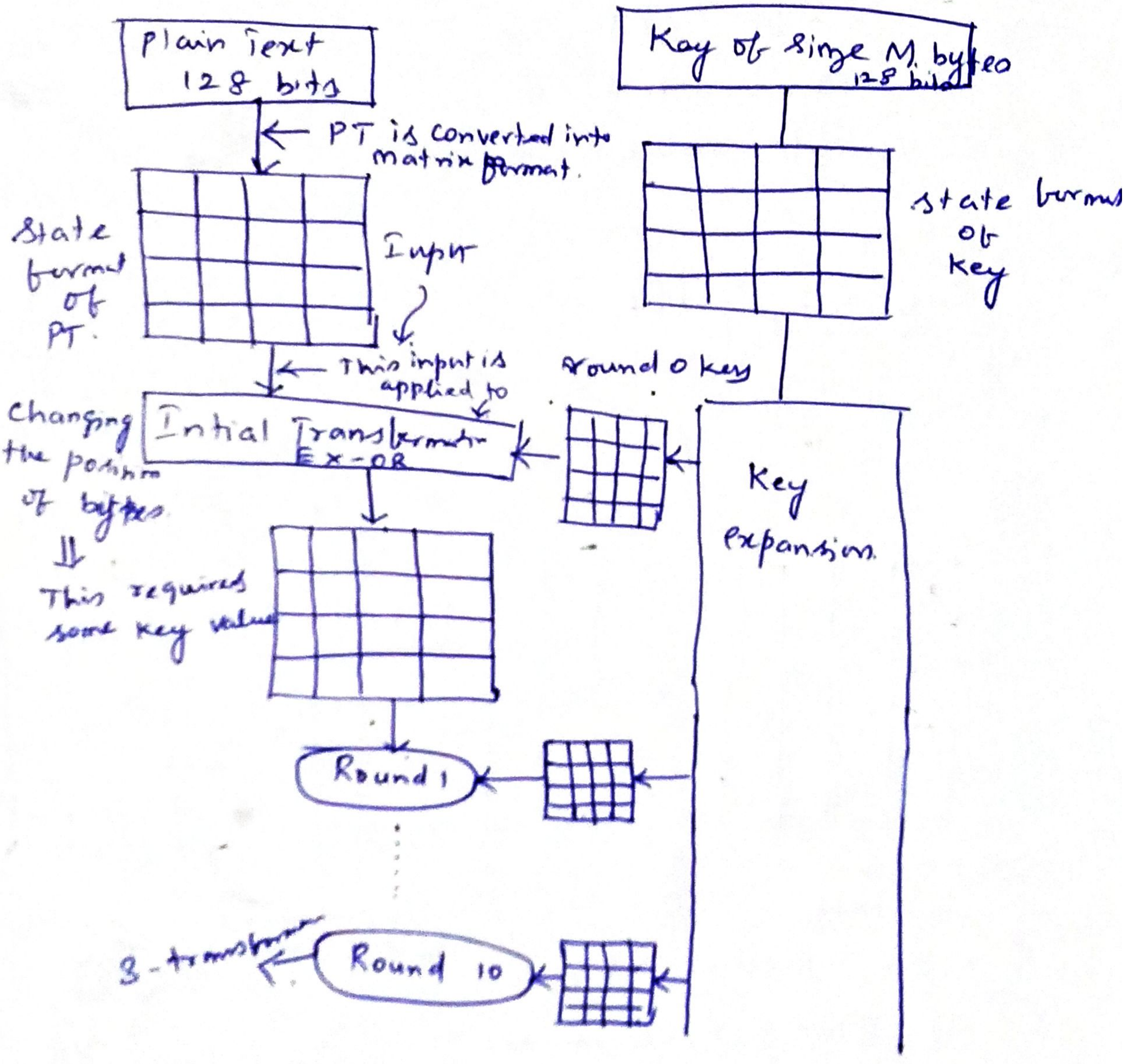
To perform Enc and DEC operation, the 128 bits are repre, in 4x4 matrix.

This Column major Matrix is called STATE

Each block size is 16 bytes

STATE represents the 128 bit plain Text in Matrix format.

Block Diagram of AES



Each, Round we apply 4 transformation, but in the Last Round we apply only 3 transformation (mix col) is ignored

The 4 – transformations are

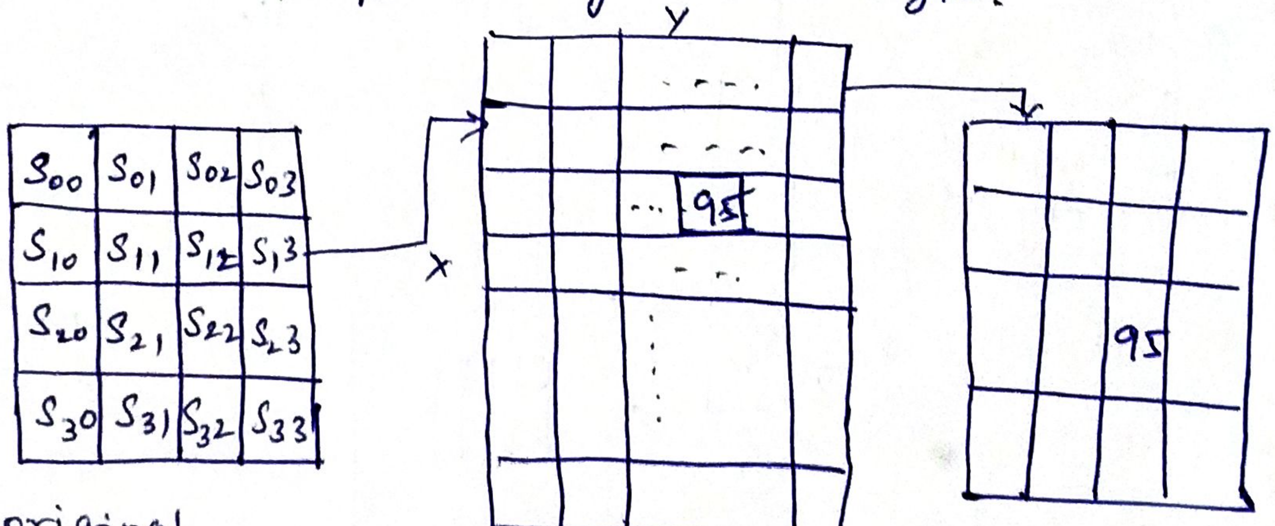
1. Substitute byte
2. Shift Rows
3. Mix – column matrix.
4. Add column matrix

} Each round, we perform these 4 transformations.

1. Substitute byte

It is a substitution Tech – one byte is replaced by another byte.

eg



S_{00}	S_{01}	S_{02}	S_{03}
S_{10}	S_{11}	S_{12}	S_{13}
S_{20}	S_{21}	S_{22}	S_{23}
S_{30}	S_{31}	S_{32}	S_{33}

original State.

16 x 16 matrix. S – box.

New state.

Each box represents 8 bits.

L – 4 bits → row number

R – 4 bits → column number.

eg

suppose S_{12} = 37 → ~~8~~

↓ 7th column.

3rd row

then Control goes to S-box. In 3rd Row, 7th Column we have 95

In the New State $\boxed{out_{12}}$ ~~with~~ is replaced with the value 95.

After Completion of substitute byte we move to Shift Rows. So New State is the input to Shift Rows ie for ~~2~~ Second transformation

2. Shift Rows

S_{00}	S_{01}	S_{02}	S_{03}
S_{10}	S_{11}	S_{12}	S_{13}
S_{20}	S_{21}	S_{22}	S_{23}
S_{30}	S_{31}	S_{32}	S_{33}

We will Shift the Rows of this NEW State of Previous transformation.

row 0 → No Change
row 1 → one left circular Shift
row 2 → two left circular Shift
row 3 → three left cirular Shift.

S_{00}	S_{01}	S_{02}	S_{03}
S_{11}	S_{12}	S_{13}	S_{10}
S_{22}	S_{23}	S_{20}	S_{21}
S_{33}	S_{30}	S_{31}	S_{32}

This is the matrix, ~~we~~ after performing Shift-Row transformation.

3. Mix- Column.

The matrix we got from 2nd transformation is multiplied by some fixed matrix.

The fixed matrix Contains only three values

1 - No change

2 - One left shift operation.

3 - Perform one left shift and perform XOR with left shifted and before left shift value.

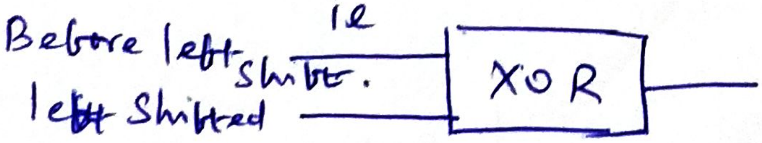
The byte is replaced with the Corresponding value.

If a byte is multiplied by one, then No Change.

If a col. byte is multiplied by two, one left shift.

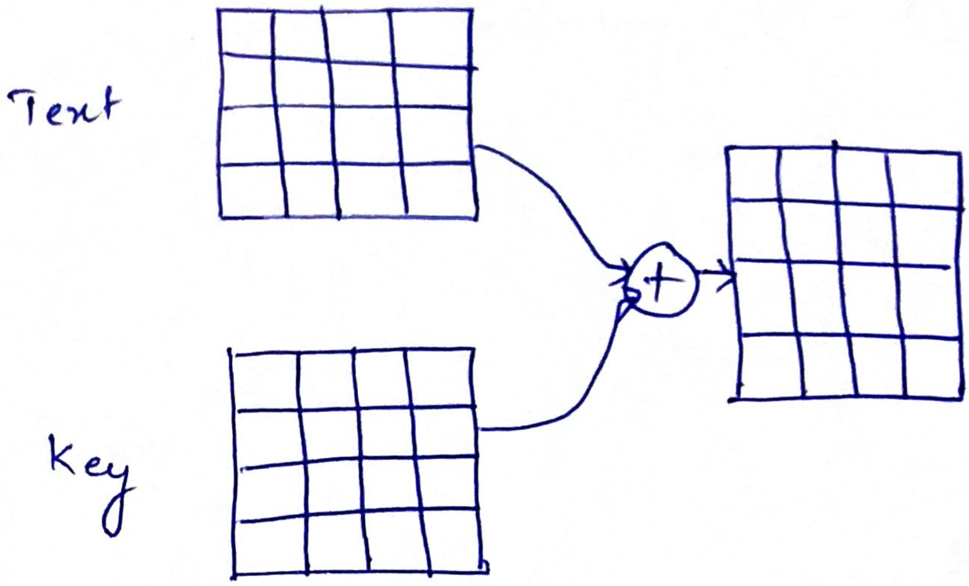
If multiplied by 3, then One left shift and Perform XOR. with left shifted and before left shift.

ie



$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 3 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

4. Add - Round key



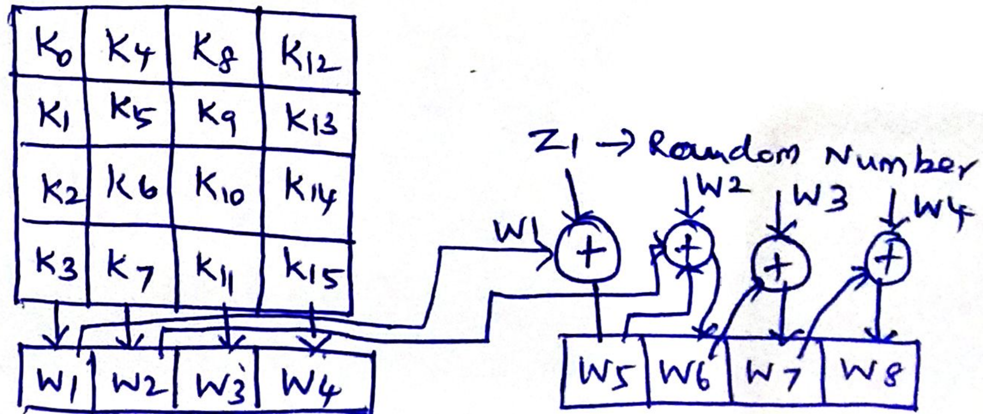
In every round we must add a key to the text. The correspondings positions of Text and key are Ex-ORed.

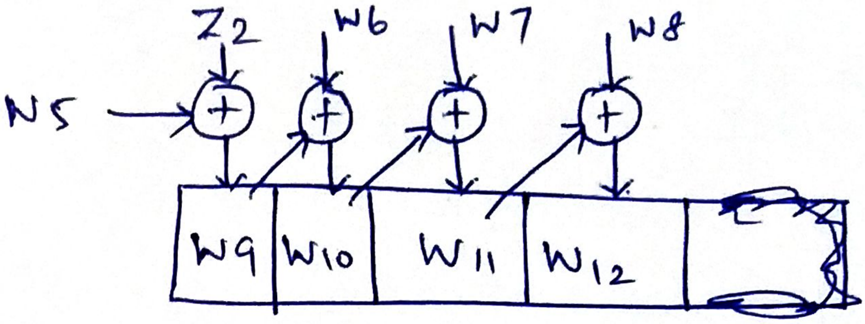
The Final Round contains only three transformations.

Mix- column Round is ignored.

Key- Expansion

Key also represented in STATE format.





Z2
W6
W7
W8
N5
+
+
+
+
W9
W10
W11
W12

AES

→ Block size – 128 bit Plain Text (4 words / 16 bytes).

→ No. of Rounds – 10

→ one word – 32 bits.

→ Key size – 128 bit (4 word / 16 bytes). → key is processed in term of words.

→ No. of subkeys – 44 subkeys.

→ Each Subkey size – 32 bit / 1-word / 4-byte

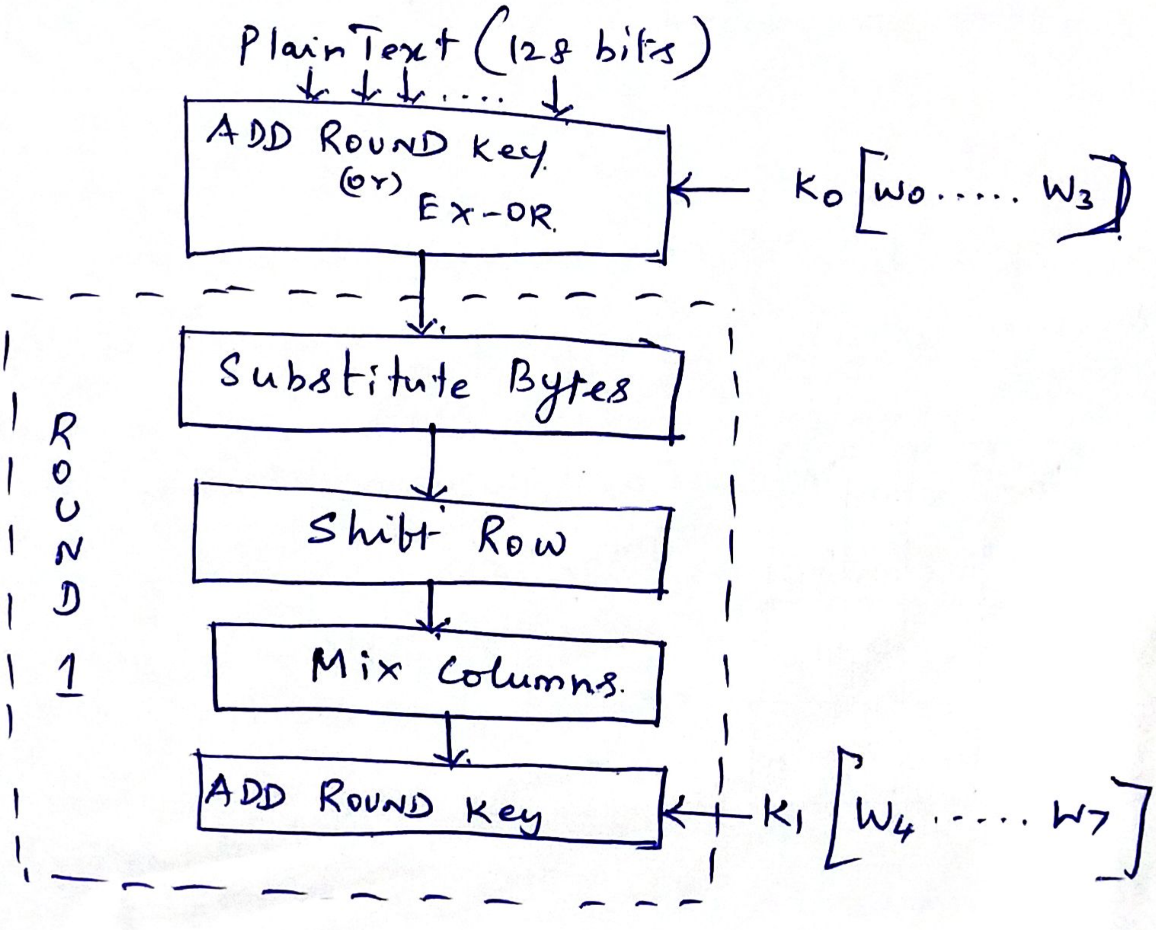
→ Each Round – 4 subkeys (128 bits / 4 words / 16 bytes.)

10 Rounds, so 40 subkeys is used. and

→ Pre Round Calulation – 4 subkeys (128 bits / 4 words / 16 bytes)

→ Cipher Text – 128 bits (4 words / 16 bytes).

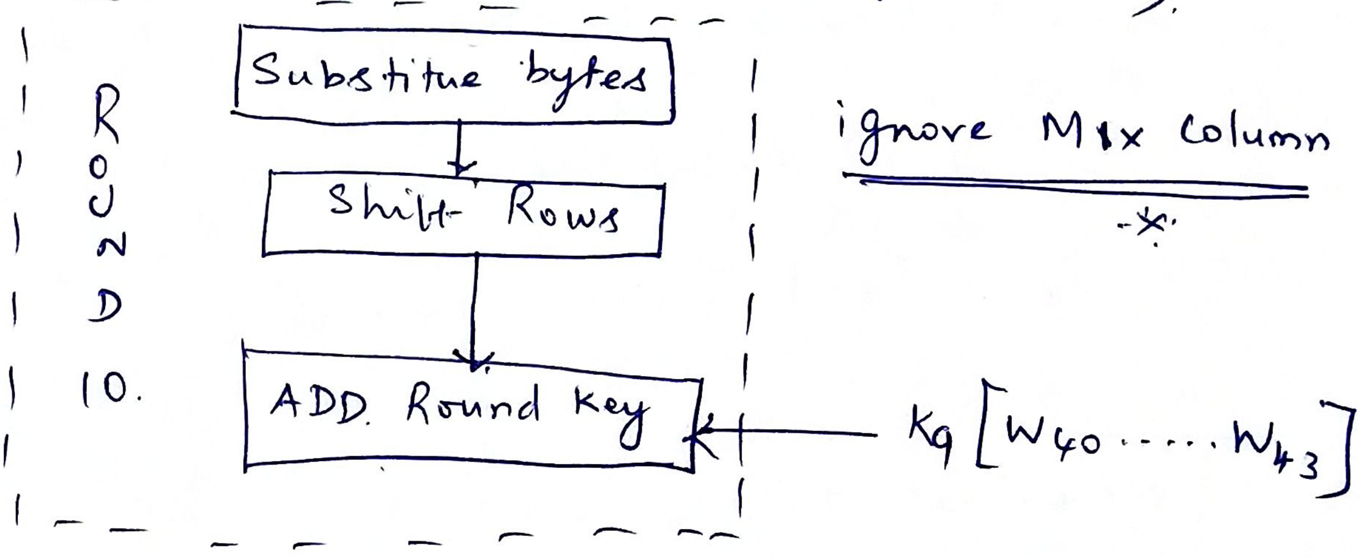
Block Diagram



In ~~the~~ Round-2, We use $W_8 \ldots W_{11}$ – K2
Round-3, $W_{12} \ldots W_{15}$ – K3
⋮
Round 10, $W_{40} \ldots W_{43}$ – K9.

These 44 words are generated from 128 bit key.

In Round 10 (Need not apply Mix col.)



ignore Mix Column

-*-

Representation.

The 128 bits PlainText is Stored in input Arrays. which is a 4×4 matrix. or Table.

Plaintext.

in_0	in_4	in_8	in_{12}
in_1	in_5	in_9	in_{13}
in_2	in_6	in_{10}	in_{14}
in_3	in_7	in_{11}	in_{15}

Each column represents one byte.

TOT 16 bytes, SO
16 × 8 = 128 bits.

Intermediate Results stored in State Array →

byte
word

S00	S01	S0,2.	S0,3
S1 0	S1,1	S1,2	S1,3
S2 0	S2,1	S2,2	S2,3
S3 0	S3,1	S3,2	S3,3

one Word.

output → output Array.

out0	out4	out8	out12
out1	out5	out9	out 13
Out 2	out6	out10	out 14
out 3	out7	out11	out15

key

1word	2word	3word	4word
K0	K4	K8	K12
K1	K5	K9	K13
K2	K6	K10	K14
K3	K7	K11	K15

→

w0	w1	w2	------	w43

128 bits.

4-words ⟶ 44 words.

* Swapping Tech.

Left half will be stored in Right most bits and Right half bit will be stored in Left half bit.

* Function is determined depends on the algorithm we use.

* The Security mainly depends on the.

- function
- no. of Rounds
- no. of keys.

→ * The Right most bits are encrypted by using the sub key which is ~~created~~ generated from master key and a logical function is used to encrypt the Right half data bits

→ * The Encrypted Right half bits are Ex-ORed with Left half bits.

→ * Then the swapping is done.

* The above three points are called Round one.

* To Provide more Security, no. of Rounds will be increased.

BLOCK CIPHER DESIGN PRINCIPLES.

→ Plain Text in two equal halves.

→ Block Size

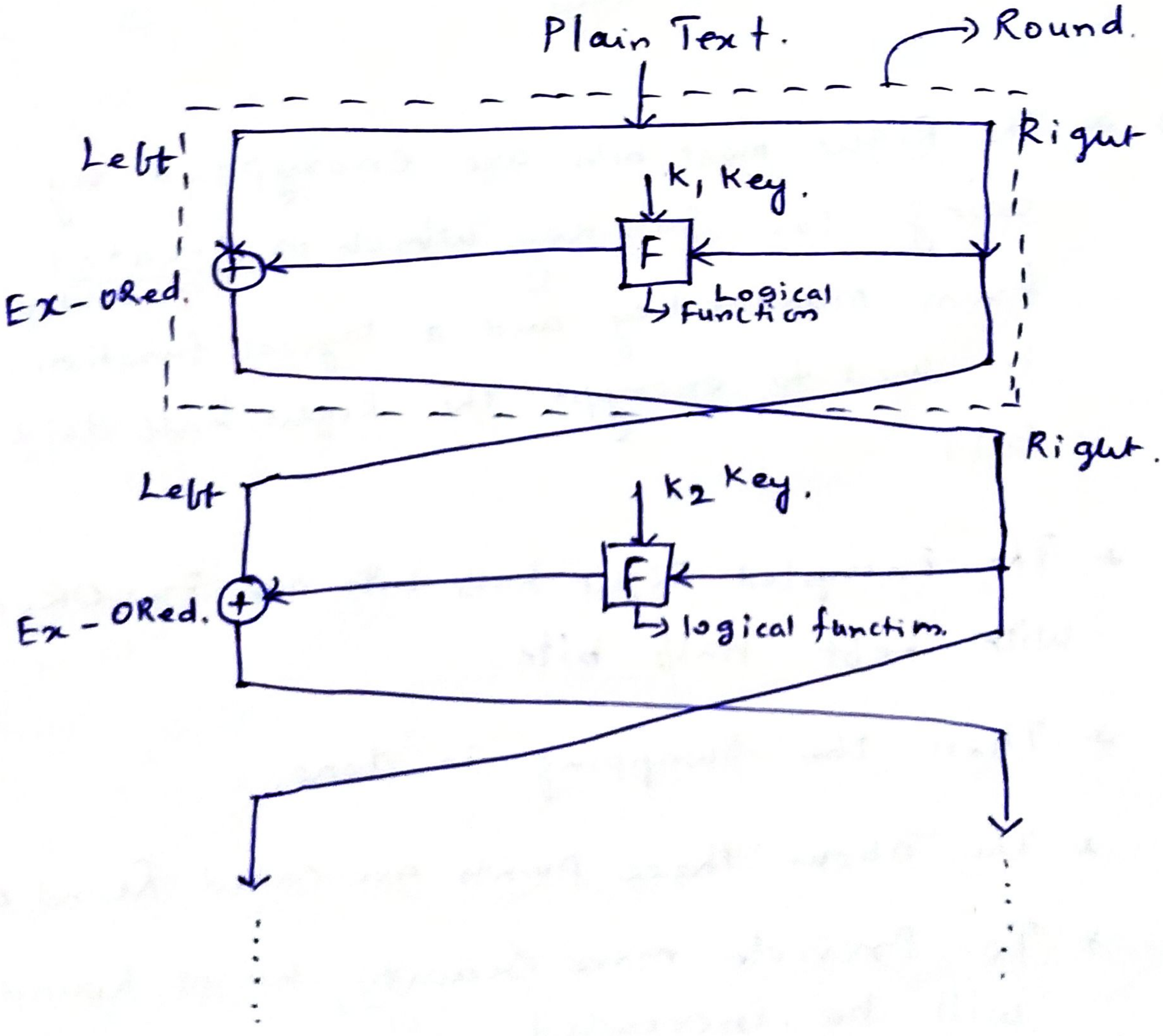
→ Key size

→ No. of Rounds

→ No. of Sub Keys.

→ Round function.

FEISTEL STRUCTURE.



3	2	4	5	1
W	E	L	C	O
M	E	T	O	M
Y	S	E	S	S
I	O	N	x	y

$\boxed{x \quad y}$ → dummy Char.

write down the text from lower key Value.

Cipher Text = OMSYEESOWMYILTENCOSX

In order to improve the security, dual cipher is Considered.

(ie) Combination of both Rail Fence and Row Trans.

→ again this CT is as PT.

Transposition Tech → Rearrange the order of bits.
No replacement/Substition.

1. Rail Fence cipher

2. Row Transposition cipher.

1. Rail Fence cipher

Rearrange the order of plain Tex bit.

eg

WELCOME TO my SESSION – Plain Text.

```
W   L   O E O Y E   S   O
  E   C M ~~F~~T M S   S   I   N
```

cipher Text = WLDEOYESOE CMTMSSION.

~~Vog very easy to break~~.

Very easy to break. this cipher Text.

2. Row Transposition. Cipher.

e.g

Plain Text = "WELCOME TO MY SESSION"

Key → Unique number should be Considered.

0 to 9.

Key → (3 2 4 5 1) → No repetition in the Key.

apply step-5 → differant Row and column.

HE — WF

apply step-5 → differan. Row and column.

LX → UP

apply step-5 - differentiate Row and column.

LO → NS

$\boxed{\text{HELXLO} \rightarrow \text{WFUPNS.}}$

Example - 2

PT = BALLOON

Key = Network.

CT = ?

Step - 1

BA/LL/OO/N

BA/LX/LO/ON = CB/UP/NS/NE/

* BA → apply step-3

BA → CB → step 3

LX → UP → step 5

LO → NS → step 5

ON → NE → step-3

2. Play - Fair cipher.

PT = HELLO

Key = NETWORK.

CT = ?

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

5 x 5 Table.

Rule

1. Divide PT to pair of letters.
2. Differentiate Repeated letters in the pair with dummy letter.
3. If pair of PT letters are in Same Row, Replace them with Right most letters.
4. If the PT letters are in Same Column, Replace with beneath letters.
5. PT letters are in different Row and Colun (diagonal)

Step-1

HE / LL / O

↳ differentiate with x.

STEP-2

HE / Lx / LO

→ CT(L) = (12 + 4) mod 26

= 16 mod 26

CT(L) = 16 = P

→ CT(L) = (12 + 4) mod 26

= 16 mod 26

CT(L) = 16 = P

→ CT(O) = (15 + 4) mod 26

= 19 mod 26

= 19

CT(O) = 19 = S

Plain Text = HELLO

Cipher Text = LIPPS

MOD function

MOD = Remaindor of Division

5 mod 2 = 1

2)5(2
4
1 → Remaindor.

5 MOD 2 = 1

2 MOD 5 = 2

5)2(0
0
2 — Remaindor.

a mod b is a
if a < b.

Example - 2

PT = ZOO

K = 4

CT(Z) = (26 + 4) mod 26

= 30 mod 26

= 4.

CT(Z) = 4 = D

CT(O) = (15 + 4) mod 26

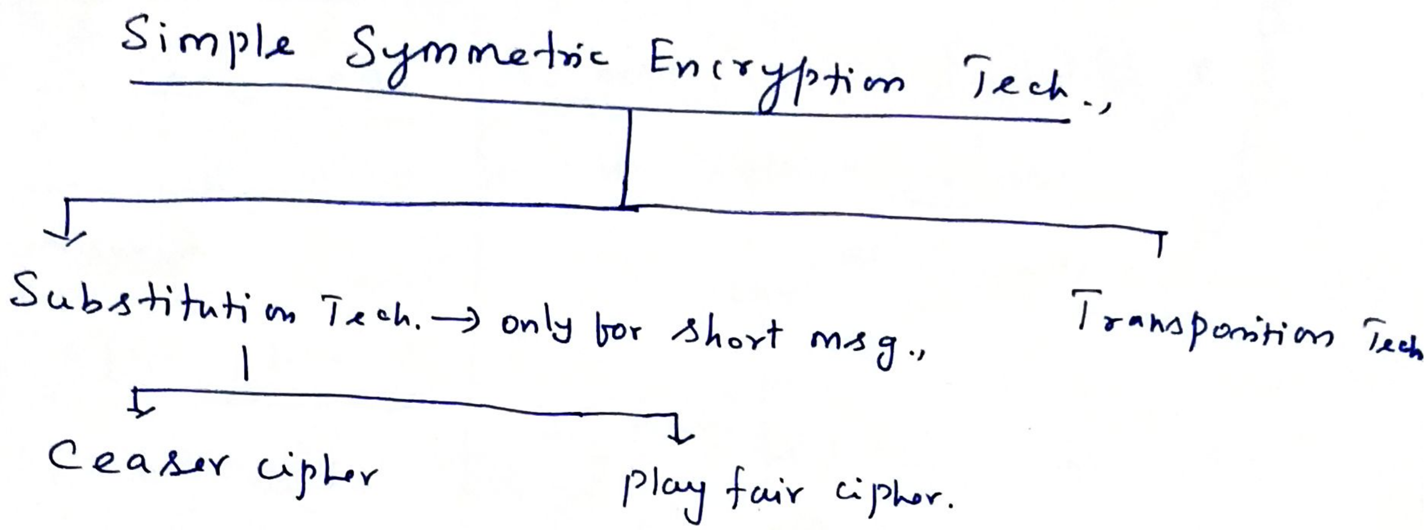
= 19 mod 26

= 19

CT(O) = 19 = S

Plain Text = ZOO

Cipher Text = DSS



Simple Symmetric Encryption Tech.,

Substitution Tech. → only for short msg.,

Transposition Tech

Ceaser cipher

play fair cipher.

1. Ceaser cipher

* we must use a single key for both ENC and DEC.

* key - Numerical, K.

$1 \leq K \leq 26.$

cipher - $CT = (PT + K) \bmod 26.$

Example

PT = HELLO

K = 4

→ $CT(H) = (8+4) \bmod 26$

$= 12 \bmod 26.$

$CT(H) = 12 = \underline{\underline{L}}.$

→ $CT(E) = (5+4) \bmod 26$

$= 9 \bmod 26$

$CT(E) = 9 = \underline{\underline{I}}$

A	1
B	2
C	3
D	4
E	5
F	6
G	7
H	8
I	9
J	10
K	11
L	12
M	13
N	14
O	15
P	16
Q	17
R	18
S	19
T	20
U	21
V	22
W	23
X	24
Y	25
Z	26.

Diffie Helman Key exchange.

- * Not an encryption Algorithm
- * used to exchange Secret key / Symmetric key.
- * We use Asymmetric Encryption for obtaining the Key exchange process.

  * Public Key * Private Key.

* **Procedure**

1. Select two numbers and make them as Public.

   $q \rightarrow$ Prime number

   $\alpha \rightarrow$ Primitive Root. of q.

2. User A and user B want to exchange the keys,

   user A generates $\boxed{X_A < q}$ → random number

   Illy User B generates $\boxed{X_B < q}$ → random Number.

   Based on X_A Value, User A Calculates

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

   User B calculates

   $$\boxed{Y_B = \alpha^{X_B} \bmod q}$$

In the above Keys all X represents Private Key ie. X → Private Key

Y ⇒ Public key.

ie X_A is the private Key of user A

Y_A is the Public Key of user A.

Illy X_B is the Private Key of user B.

Y_B is the Public key of user B.

Once ~~the~~ X_A, Y_A, X_B and Y_B are generated, we have to generate the Keys.

Suppose Y_A is transfered to Y_B. or viceversa to perform Enc or DEC, we have to transfer the Public key.

eg.

$$\boxed{K = (Y_B)^{X_A} \bmod q}$$

this Key is generated in user A side.

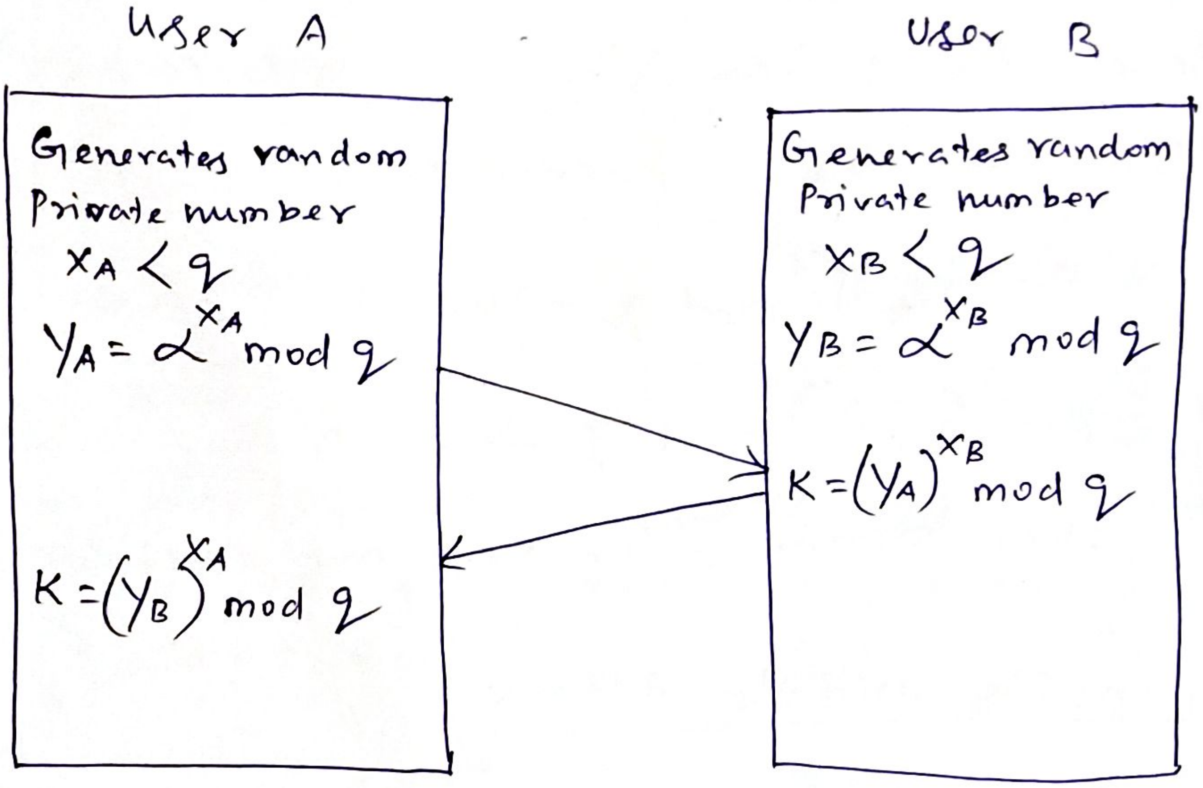
Suppose Y_A is transfered to user B.

$$\boxed{K = (Y_A)^{X_B} \bmod q}$$

this Key is generated in user B side.

These two Keys are same. (-x-)

Key exchange Protocols


User A
User B
Generates random Private number
$X_A < q$
$Y_A = \alpha^{X_A} \bmod q$
$K = (Y_B)^{X_A} \bmod q$
Generates random Private number
$X_B < q$
$Y_B = \alpha^{X_B} \bmod q$
$K = (Y_A)^{X_B} \bmod q$

* Suppose user A wants to Communicate with user B, user A generates a random Private number

$$\boxed{X_A < q}$$

* From this X_A, user A Calculates

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

* Now to Communicate with user B, user A transfer Y_A to user B, then user B accepts the request and then user B wants to Communicat with user A,

* user B generates a random Private number

$$X_B < q$$

and also calculate $Y_B = \alpha^{X_B} \bmod q$ and this Y_B is transfered to user A.

* Now user A has Y_B and user B has Y_A. then the key is formed as

$$K = (Y_B)^{X_A} \bmod q \rightarrow \text{user A.}$$

$$K = (Y_A)^{X_B} \bmod q \rightarrow \text{user B.}$$

In this manner, the keys are exchanged.

Man in the middle Attack.

* User C is an attacker who sits between user A and user B.

* Similar to user A and user B, user C also generates two random numbers.

$$\underbrace{\begin{array}{ccc} XD1 & \text{and} & XD2 \\ ||| & & ||| \\ X_A & & X_B. \end{array}}$$

based on these two values user c calculate Y_1 and Y_2

* user c intercepts the message, and change the message as

$$Y_A = \alpha^{XD1} \bmod q$$

Illy if user B wants to transfer the message to user A, the user C intercepts the message and change the message as

$$Y_B = \alpha^{XD2} \bmod q.$$

So user C can hack the ~~the~~ system.

Thus the user C intercepts the message sent by user A and modifies the message, and transfer to user B.

Similarly the process is vice-versa.

Ex.

$q = 11, \ \alpha = 2, \ X_A = 6, \ X_B = 8$

user A.

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

$= 2^6 \bmod 11$

$Y_A = 9$

$11) 64 (5$

55

9

user B.

$Y_B = \alpha^{X_B} \bmod q$

$= 2^8 \bmod 11.$

$= 256 \bmod 11$

$Y_B. = 3$

$11) 256 (23$

22

36

33

3

User A

$$K = (Y_B)^{X_A} \bmod q$$

Suppose user A wants to generate a key, then user A uses his. Private key and user B Public key.

(ie) user A → Private
user B → Public.

$$K = (Y_B)^{\overset{\text{user A's Private key.}}{X_A}} \bmod q$$

$\underset{\text{user B's Public key.}}{Y_B}$

user B

$$K = (Y_A)^{\overset{\text{user B's Private key.}}{X_B}} \bmod q$$

$\underset{\text{user A's Public key.}}{Y_A}$

Already we said that these two keys are equal.

$$K = (Y_B)^{X_A} \bmod q$$

$= 3^6 \bmod 11.$

$= 3^2 \times 3^2 \times 3^2 \bmod 11$

$= 9 \times 9 \times 9$

$= 81 \times 9$

$= 729. \bmod 11.$

$= \boxed{3}$

```
11)729(66
   66
   ---
    69
    66
    ---
     3
```

$$K = (Y_A)^{X_B} \bmod q.$$

$$= 9^8 \bmod 11.$$

$$= (9^2)^4 \bmod 11.$$

$$= (4)^4 \bmod 11.$$

$$= \boxed{3}$$

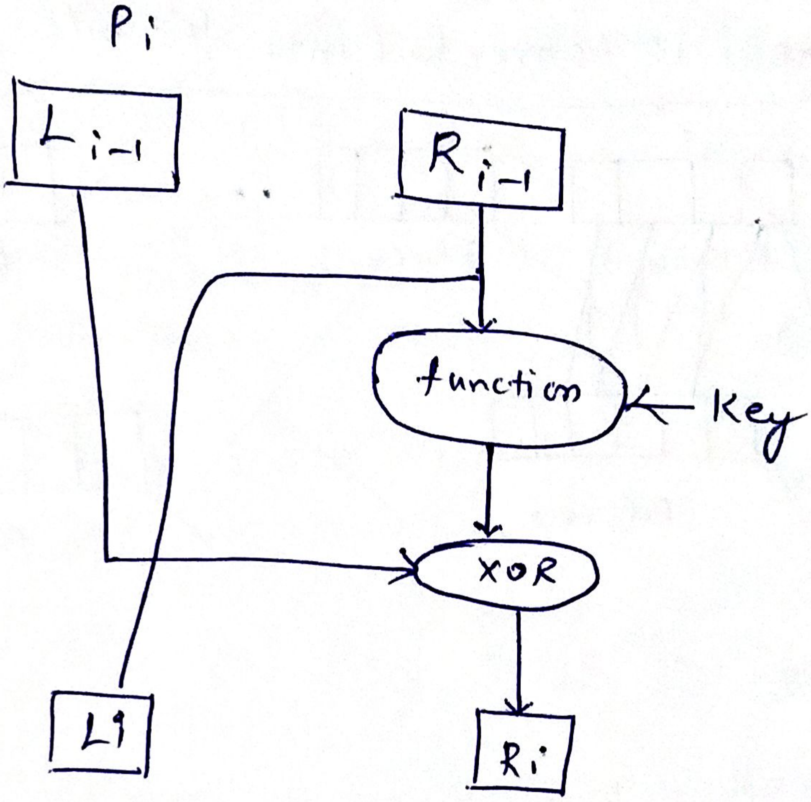
$$11\overline{)81}(7$$
$$\underline{77}$$
$$4$$

$$11\overline{)256}(23$$
$$\underline{22}$$
$$36$$
$$\underline{33}$$
$$3$$

User A's Key value is 3

user B's Key value is 3.

Feistel cipher/ structure



Parameters

$L_i = R_{i-1}$

$R_1 = L_{i-1}$ XOR (function (R_{i-1}, k)).

1 No. of Rounds 16

2 Block size 16 bytes

3 Key 128 bits

4. Sub keys.

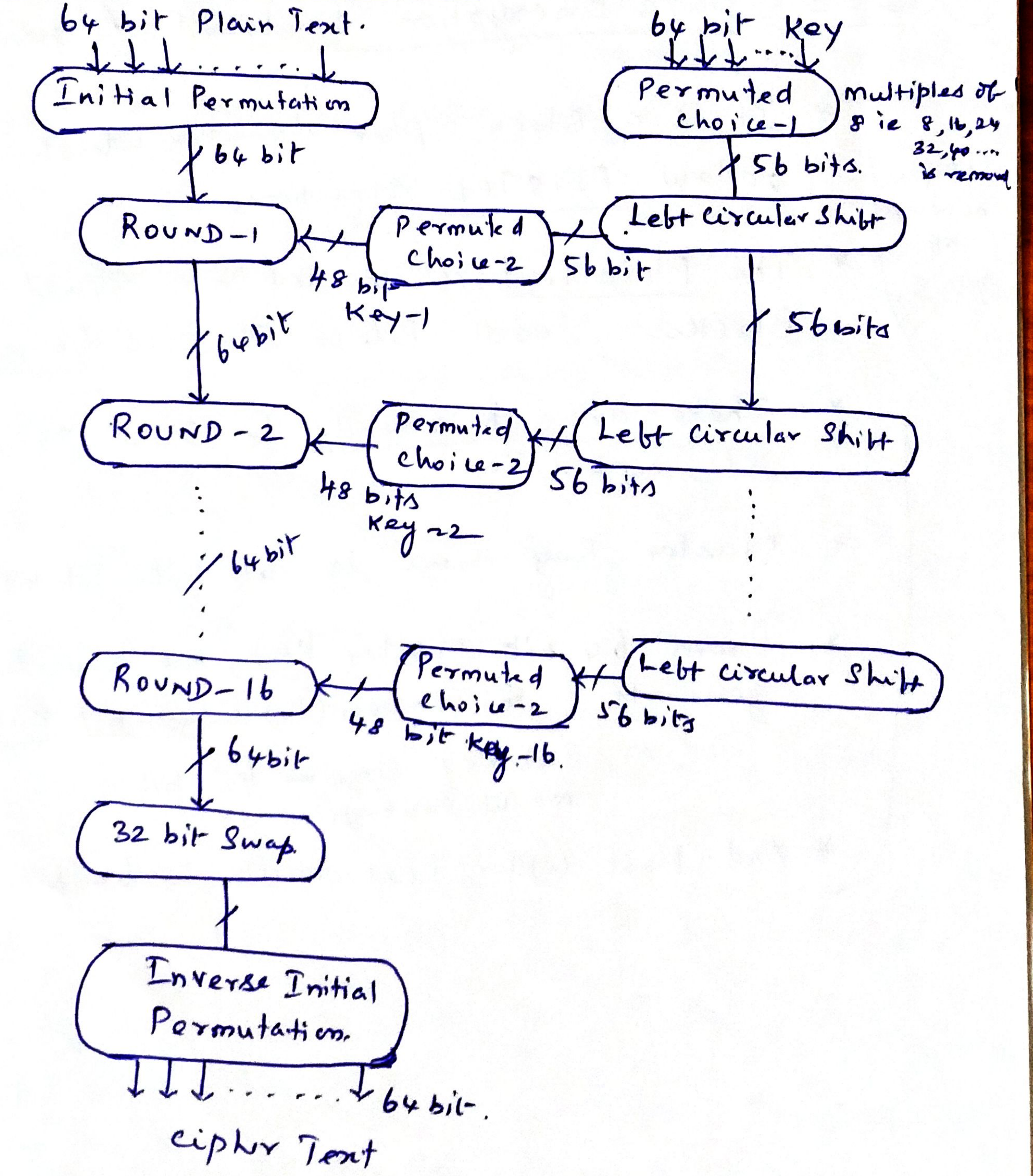
A) DES - Data Encryption Standered Algorithm.

simple overview of DES.

- * It is a Block cipher algorithm which follow FEISTEL Structure.
- * The Plain Text is divided into equal Blocks. each Block size is 64 bit
- * There are 16 rounds in DES to get cipher Text.
- * Master Key size is 64 bit (56 bits).
- * from 64 bit master key, we have to generate ~~16~~ sixteen (16) 48-bit subkeys

  (ie) Subkey size — 48 bit.
  No. of subkeys — 16.
- * And final cipher Text width is 64-bit.

Block Diagram of DES.



In the initial Permutation, the numbers are arranged in the following order

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	. . .	. . .	. . .	. . .	. . .	31	32
33	34	35	36	37-	38-	39	40
41	42	43	44	45	46	47	48
49	. . .	. . .	. . .	. . .	. . .	. . .	56
57	. . .	. . .	. . .	. . .	. . .	. . .	64

original order of bits.

Bit positions are Changed as

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

This is initial Permutation

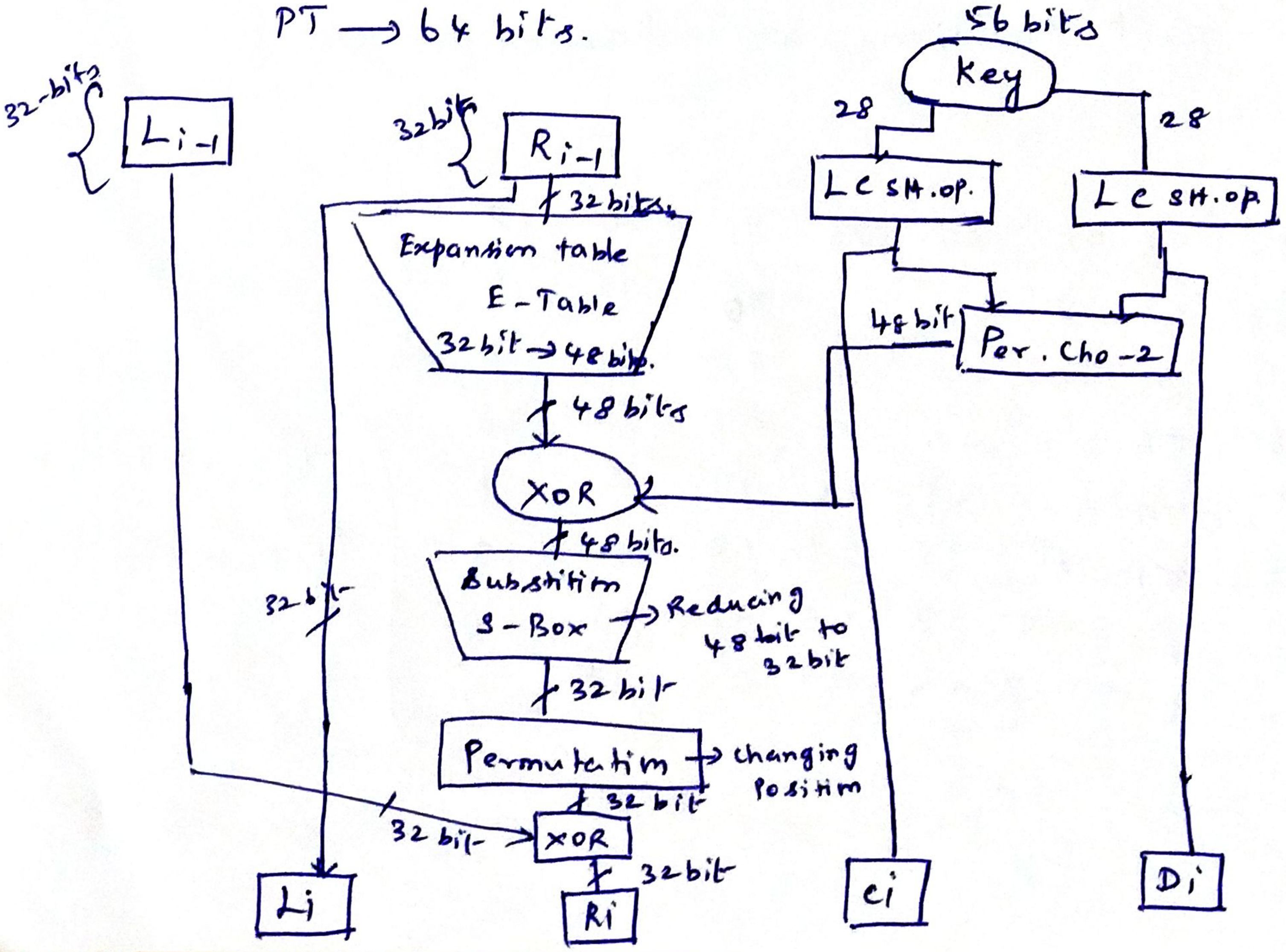
Inverse initial Permutation.

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

Left half bits

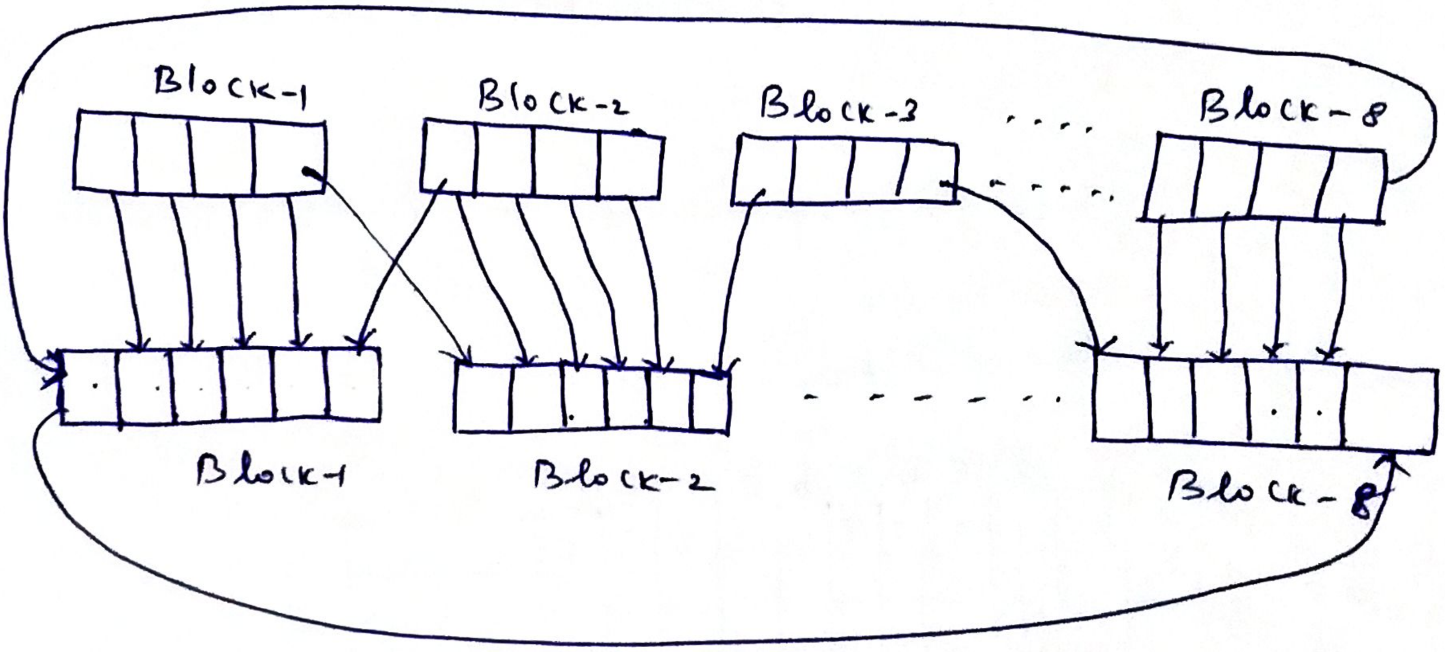
~~Left~~ Right side.
32 bit

Round operation (encryption).



Expansion Table

The 32-bit input is Converted into 48bits



* In the First one, Block size is 4 bits so 8 blocks are available, 8×4 = 32 bits

* In the Second case, Block size is 6 bits so 8 ~~blocks~~ are available, 8×6 = 48

Hence, 32 bit input is Converted into 48 bits data.

Expansion Table.

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Substitution Box - S-Box

The 48-bit input is converted into 32 bits

In DES algorithm ~~whe~~ we have
8 - S boxes.

Each S box is represented in Matrix format,

4 - rows
16 - columns.

4 rows		0	1	2	3	4	5	6	7	8	9	10	.....	15
	0	1	2	4	0	7	8	0	1	2	4	5		
	1												- . .	9
	2			8									- - . .	
	3												- - - - . . .	

we will fill up the boxes ~~by~~ with our own numbers

Each block contains 6 bits.
and there are 8 blocks, so we have 48 bits

~~1001~~ ~~10010~~

eg

1 $\boxed{0\ 0\ 1\ 0}$ 0 → 6 bits.

8 blocks × 6 bits = 48 bits

represents row number 10 → 2

reprsents column number. 0010 → 2

converts

In Row 2 and column 2 we have the number 8

ie $\boxed{1000}$ → 4 bits.

8 blocks × 4 bits = 32 bits.

Elliptic Curve Cryptography

* The ENC and DEC operations are very fast Compared to the previous algorithms.

* uses a small Key compared to Public Key Enc Sys.

* Even though it uses small Key, it provides Same level of Security.

* ~~Elliptic Curves are defined~~ by ~~over real number~~

* ~~Elliptic curves are defined~~ by ~~over~~ Zp Prime Number

* To Perform ENC or DEC, we need to use Elliptic Curves.

* The Elliptic curves are defined over different variables.

* Some of the possible cases are.

1. Elliptic curves are defined over real nos.,
2. " " " " " Zp. (Prime)
3. " " " " " GF (2^m).

GF → a Finite Field.

2^m → no. of entries (or) elements

1. Elliptic Curve over Real Number

* Elliptic Curves are not ellipse

It uses Cubic equations.
* ↳ eq. used to calculate the circumfrence of ellipse

* The Cubic eq., is defined as

$$y^2 = x^3 + ax + b.$$

* To plot a curve ~~on~~ using the Cubic eq., we need to compute $y = \sqrt{x^3 + ax + b}$ for every Combination of a and b.

* Y Value will Produce +ve and −ve Values

* Suppose we plot a graph for this +ve and −ve values, then the line is Symmetric. ie $y = 0$.

* Rules of addition

Suppose in the given Curve, any three points are joined by a straight line, then the Sum = 0.

O acts as identity element

* If P is a point, then P+O = P.

* If Point P is $P(x_P, y_P)$ then

Negative point (x and y are co-oridinates)

$$-P(x_P, -y_P)$$

for each x value in y, we are getting +ve and -ve values.

* Consider two points A and B with different x-Coordinates, then we have to Calculate the intersection points between A and B, we get the third point.

2. Elliptic Curve over Zp (Prime Numbers)

* Elliptic Curve is also called as Prime Curve.

* The cubic equation Contains a set of Variables and Co-efficients for eg a, b, x and y.

* All values are in range betwen 0 to P-1.

* In this case the Cubic eq is

$$\boxed{y^2 \bmod P = (x^3 + ax + b) \bmod P.}$$

Rules of addition

1. $P + \overset{\text{identity.}}{O} = P$

P → Point

* If $P = x_p, y_p)$ for x-co-ordinate then

$$-P\,(x_p, -y_p).$$

ie $P + (-P) = 0.$

3. <u>Elliptic Curve over $GF(2^m)$</u>

* Cubic eq., Contains a set of variables and Coefficients from 2^m ~~[illegible]~~ elements.

* The Cubic equation is

$$\boxed{y^2 + xy = x^3 + ax^2 + b.}$$

* <u>Rules</u>.

* $P + \overset{\text{acts as identity}}{O} = P.$

* If $P = (x_p, y_p)$, then

$$-P = (x_p,\ x_p + y_p).$$

Now we ~~will~~ discuss the ENC and DEC operation using elliptic curve crypography.

* To perform elliptic curve crypography we apply Diffie Hellman Key exchange concepts.

* The exchange of Keys b/w two users is implemented using elliptic Curve Concepts.

* basic Concepts of Dibbie Hellman

* Choose a large integer q

q → is a prime Number

(or) q is represented as 2^m

* Pick an integer G.

G → is a Point on the Curve.

G → is larger than n.

* Consider two users A and B.

User A	User B.
$n_A < n$	$n_B < n$.
n_A → Private key.	n_B → Private key.
$P_A = n_A \times G$.	$P_B = n_B \times G$.
P_A → Public key.	P_B → Public key.
$K = n_A \times P_B$ ↳ B's Public key	$K = n_B \times P_A$ ↳ A's Public key

* q and G are Public elements for both A and B

The above ~~Def~~ Diffie Hellman Concepts are implemented using elliptic curve Concepts.

* Now the Plain Text is represented as x and y co-ordinates

$$PT = (x, y) \Rightarrow \underbrace{P_m}_{\text{plain Text in } x, \& y \text{ format}}$$

* To Perform ENC or DEC, we require a elliptic curve over G. ie an integer G. and an elliptic group ↳ 3rd method.

$E_q(a, b)$ as parameters.

So using G and $E_q(a, b)$ we will perform ENC or DEC.

* Let us assume a random integer K.

* To Perform ENC operation, in user A side.

~~Cm = {K, G}~~

$$C_m = \{ \underbrace{K}_{\text{any Random integer}} \overset{\text{is our Parameter}}{G}, \underbrace{P_m}_{\text{Plain Text}} + K \underbrace{P_B}_{\text{Public key of B}} \}$$

To Perform DEC in the User B side.

$$\boxed{P_m = P_m + kP_B - n_B \times \{kG\}}$$

~~$P_m + k(n_B \times G) - n_B \times \{k G\}$~~

$$= P_m + k(n_B \times G) - n_B \times k \times G.$$

$$= P_m.$$

Cryptography and Network Security

Introduction

Computer data often travels from one Computer to another, leaving the Safety of its protected Physical Surroundings.

Once the data is out of hand, People with bad intention Could modify or forge your data, either for amusement or for their own benefit.

Cryptography can reformat and transform our data, making it safer on its trip between Computers. The Technology is based on the essentials of secret codes, augmented by modern mathematics that protects our data in powerful ways.

Computer Security

- generic name for the collection of tools designed to protect data and to prevent hackers.

Network Security

- Measures to protect data during their transmission.

Internet Security

- measures to Protect data during their transmission over a Collection of inter Connected Networks.

Basic Concepts

1. Plain Text:

The original intelligible message, readable format.

2. cipher Text:

The transformed message, Non-readable format.

3. cipher:

An algorithm for transforming an ~~it~~ intelligible message into one that is unintelligible by transposition and/or Substitution methods.

4. Key:

Some critical information used by the cipher, Known only to the Sender and receiver.

5. Encipher (encode)

The process of Converting PlainText to Cipher text using a cipher and a Key.

6. Decipher (Decode)

The Process of Converting cipher Text back into PlainText using a cipher and a key.

7. Cryptography

The art or Science encompassing the Principles and Methods of transforming an intelligible message into one that is unintelligible, and then retransforming that message back to its original form

(ie) Study of ENCryption.

8. Cryptanalysis

The Study of Principles and methods of transforming an unintelligible message back into an Intelligible Message without Knowledge of the Key. Also called Code Breaking

(ie) Study of DEcryption

9. Cryptology

Both Cryptography and Cryptanalysis.

10. code

An algorithm for transforming an intelligible message into an unintelligible one using a Code-book.

ENC can be done in two ways.

1. Stream cipher

Bit by bit Conversion,
only for Short message.

2. Block cipher

group of bits
block by block Conversion.

IP Security Architecture (IP → Internet protocol)

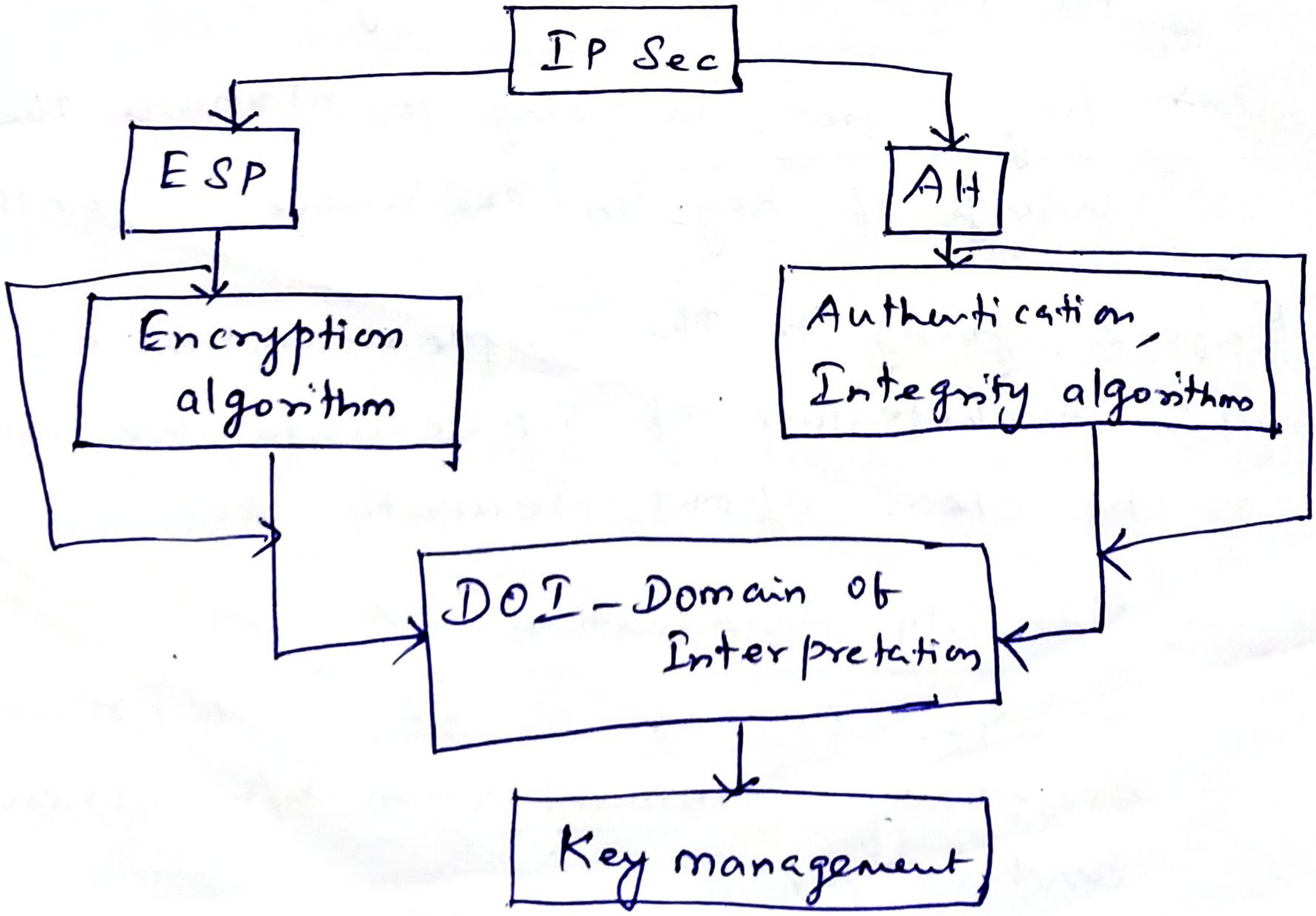
* IP Security is implemented using two protocols.

1. ESP → Encapsulating Security Payload.
2. AH. → Authentication Header.

* Services given by IP Security are:

1. Authentication → provided by using AH.
2. Confidentiality → provided by using ESP.
3. Integrity. → provided by using AH.

Architecture of IP Security


IP Sec
ESP
AH
Encryption algorithm
Authentication, Integrity algorithms
DOI - Domain of Interpretation
Key management

* IP Security is Implemented using two Protocols
  a) ESP b) A.H

* ESP Contains all ENCs algorithms
* A.H. Contains all Authentication and Integrity algorithms.

* Now these two algorithms are Combined together. [DOI]

* To Provide more security, we Combine both ESP algorithms and A.H. algorithms ie Domain of Interpretation. DOI.

* Finally, to provide Secured transformation, we have to generate Keys.

* Key mgmt., is going to Manage the usage of Keys in the above algorithms.

Before going to the Implementation of this Architecture of IP Security, we have to be clear about Security Association.

Security Association. (SA)

The purpose of SA, is to Provide one-way Communication b/w client and Server.

eg

Client → Server

The Communication is established by sending Security Association of client to the Server.

(ie) SA$_c$ is sent to the server.

IIIly.

from Ser-ver to client

client ← Server.

Server sends SA$_s$ to client

ie To Provide Comm., b/w two users, they have to provide their Security Association (SA)

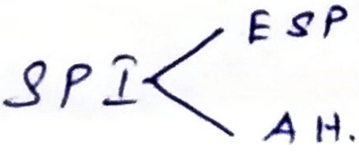
* what are the parameters ~~used in SA~~ that are used to identify SA.?

1. Security Parameter Index: SPI.

(ie)

A unique number ~~to~~ assigned to each SA.

This SPI parameter is used in both ESP and ~~AH~~ AH.



2. IP destination Address

This will determine who is the receiver.

3. Protocol Identifier.

which protocol we use to transfer the data. either ESP or A.H.

These are the parameters that are used to identify the Security Associtation.

* All SA parameters are stored in SAD

Security Association Database.

In addition to the above parameters, some more parameters that are associated with SA.

1. SPI – Security Parameter Index.

A Unique number assigned to each and every SA.

2. Sequence number Counter.

- The entire message is divided into no. of packets.
- Each packet is given a unique number.

- If the Counter value is 100, then 100 packets are to be transfered.
- For each transformation of a packet, the Counter is decremented by one.

3. Sequence number overflow

The maximum limit of the sequence number allocated, to the Buffer.

eg the ~~Size~~ Size of the buffer is only 100 packets.
If we try to Store more than 100 packets into the Buffer then, overflow occurs.

4 Anti - Replay window

The purpose ~~is to~~ of Anti-Replay window is to avoid duplicate of Packets.

Suppose packet-1 is sent by user A. If any other user will send the same packet, the receiver has to ignore the duplicate of the same copy.

5 ESP Info Encapsulating Security payload

It provides information regarding all encryption algorithms.

6. AH info:

It provides information regarding all. Authentication algorithms.

7. Life time of Security Association (SA):

The Period of Validity of SA

(x) 8 IP Security Protocol mode.

we have two protocols

- ESP
- AH

These two protocols working in two modes

1. Transport mode
2. Tunnel mode

ie In which mode our IP Security is executed.?

Implementation of ESP and A·H Protocols

There are two versions of IPs

- IP - version 4
- IP - Version 6.

Format of IPV4.

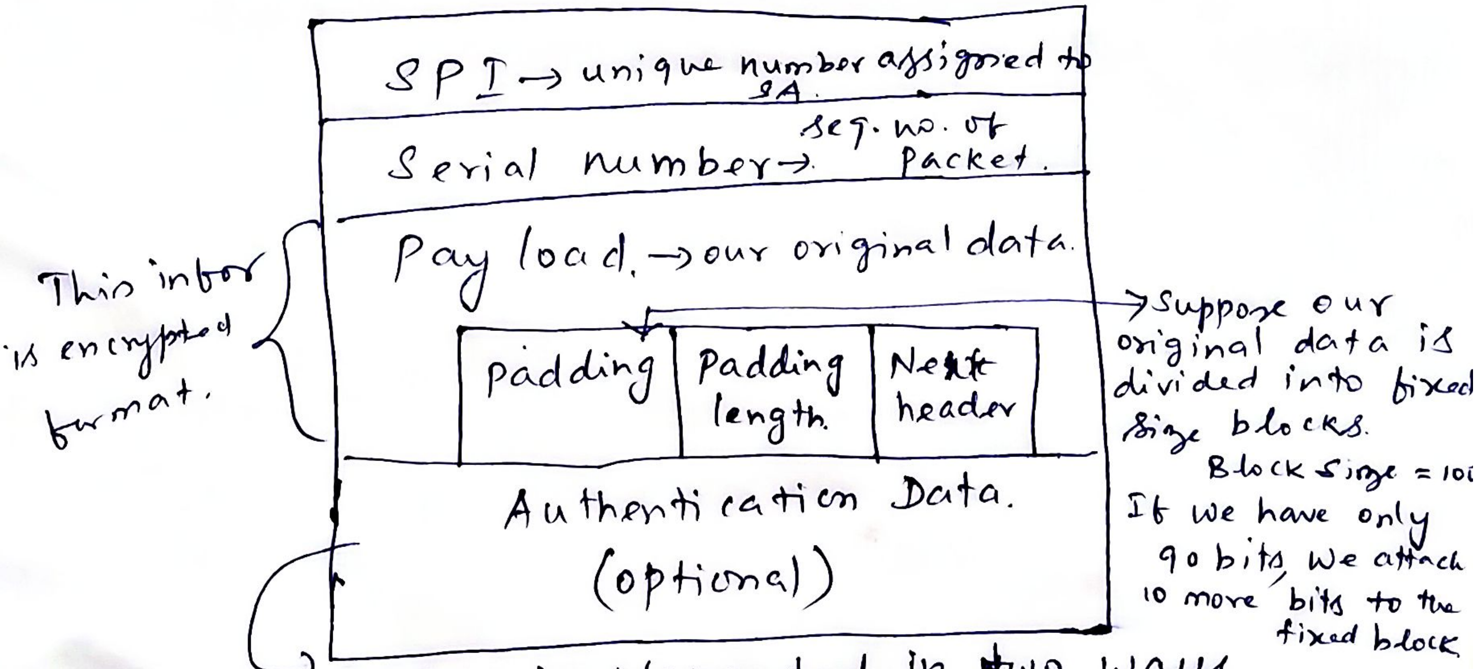
original header(IP)	TCP	Data.

Format of IPV6.

original header (I.P)	Extension header	TCP	Data.

ESP.

Packet format.



SPI → unique number assigned to SA.

Serial number → seq. no. of packet.

Pay load. → our original data.

padding	Padding length.	Next header

Authentication Data. (optional)

This infor is encrypted format.

→ Suppose our original data is divided into fixed size blocks. Block size = 100. If we have only 90 bits, we attach 10 more bits to the fixed block.

ESP is implemented in two ways.

1. ESP with Authentication
2. ESP without Authentication (optional)

Now attach this ESP with IP, we have two modes 1. Tran~~port~~ mode 2. tunnel mode.

* In trans~~fer~~port mode the format of IPV4 is as ~~follows~~. Combined with ESP.

Transport mode

IPV4	Original IP header	ESP header	TCP	Data	ESP Trailer	ESP Authentication (optional)

This is in Encrypted format

IIIy.

IPV6	original IP header	Extension header	ESP header	Tcp	Data	ESP trailer	ESP Authentication

ESP in Tunnel mode

The entire IP Packet is Considered as one Inner Packet. For that inner packet we are applying ESP.

Tunnel mode

IPV4.	new IP header	ESP header	original IP header	Tcp	Data	ESP trailer	ESP Auth

IPV6	New IP header	Extension header	ESP header	original IP header	Extension header	TCP.	Data	ESP trailor	ESP authentication

2) Authentication Header

Packet format

<table>
<tr><td>Next header</td><td>Padding Length</td><td>Reserved</td></tr>
<tr><td colspan="3">SPI</td></tr>
<tr><td colspan="3">Sl. No</td></tr>
<tr><td colspan="3">Authentication Data
(Integrity checksum)</td></tr>
</table>

Transport mode

IPV4	original Ip header	AH	TCP	Data.	
IPV6	original IP header	Extension header	AH	TCP	Data.

Tunnel mode

IPV4.	New IP header.	HA	original IP header	TCP	Data.		
IPV6	New IP header	Extension header	A.H.	original IP	TCP	Data.	

Key mgmt,

we have two types of keys
1. Secret key
2. Public key

Secret keys are managed by Diffie hell man key exchanged algorithms.

The key mgmt, is generally implemented in two ways.

~~If the~~ 1. Manual approach.
2 Automated approach.

If the size of the N/w is small we use Manual approach.
If the size of the N/w is large we use Automated approach.

In Automated approach we have two algorit

1. Oakley → A refinement of Diffie Hell-man key exchange algorithm.

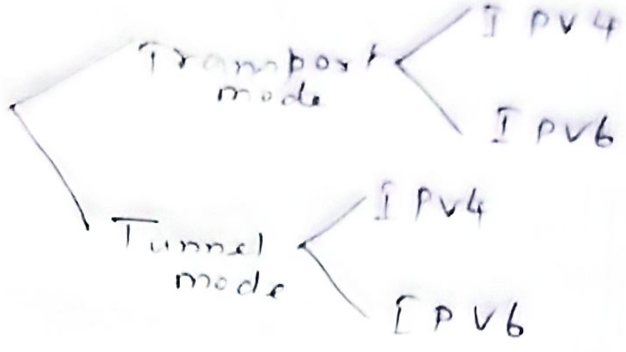
2. ISAKMP.
Internet Security Association Key Mgmt, Protocol.

overview

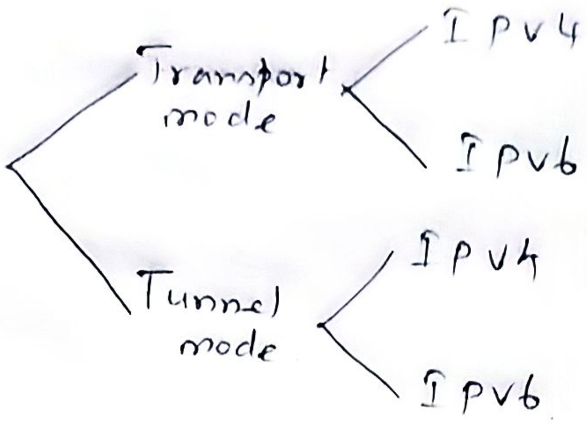
IP

IPV4, IPv6 → version of IP

1. ESP
(Packet format) :



2. AH
(Packet format) :



3. Key mgmt.,

- First we should know the structure of IPV4, IPv6.
- Then Packet format of ESP and A.H.
- Transport mode → simply we add ESP header and A.H. heade.
- Tunnel mode → Entire IP is Considered as Inner Packet. For that inner packet we add the header.

IP Security Architecture (IP → Internet protocol)

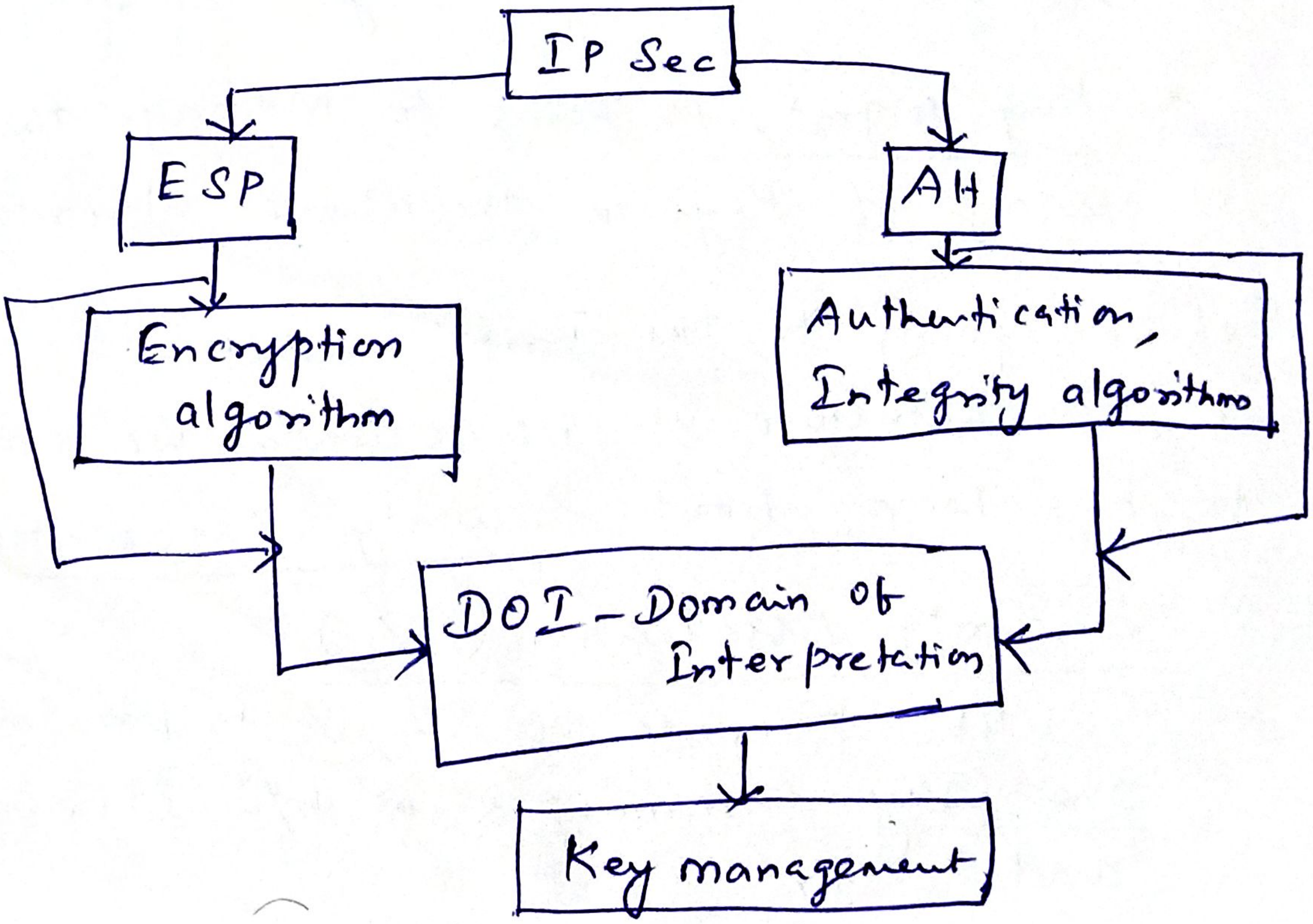
* IP Security is implemented using two protocols.

1. ESP → Encapsulating Security Payload.
2. AH. → Authentication Header.

* Services given by IP security are:

1. Authentication → Provided by using AH.
2. Confidentiality → provided by using ESP.
3. Integrity. → provided by using AH.

Architecture of IP Security


IP Sec
ESP
AH
Encryption algorithm
Authentication, Integrity algorithms
DOI - Domain of Interpretation
Key management

* IP Security is Implemented using two protocols
  a) ESP b) A.H.

* ESP Contains all ENCs algorithms

* A.H. Contains all Authentication and Integrity algorithms.

* Now these two algorithms are Combined together. [DOI]

* To Provide more security, we Combine both ESP algorithms and A.H. algorithms ie Domain of Interpretation DOI.

* Finally, to provide secured transformation, we have to generate keys.

* Key mgmt., is going to Manage the usage of keys in the above algorithms,

Before going to the Implementation of this Architecture of IP Security, we have to be clear about Security Association.

Security Association. (SA)

The purpose of SA, is to Provide one-way Communication b/w client and Server.

eg

Client → Server

The Communication is established by sending Security Association of client to the server.

IIIly. (ie) SA_c is sent to the server.

from Server to client

client ← Server.

Server sends SA_s to client

ie To provide Comm., b/w two users, they have to provide their Security Association (SA)

* what are the parameters ~~used in SA~~ that are used to identify SA.?

1. Security Parameter Index: SPI.

(ie)

A unique number ~~to~~ assigned to each SA.

This SPI parameter is used in both ESP and ~~AH~~: A H.

$$SPI \begin{cases} ESP \\ AH. \end{cases}$$

2. <u>IP destination Address</u>

This will determine who is the receiver.

3. <u>Protocol Identifier.</u>

which protocol we use to transfer the data. either ESP or A.H.

These are the parameters that are used to identify the Security Associaition.

* All SA parameters are Stored in <u>SAD</u> Security Association Database.

In addition to the above parameters, some more parameters that are associated with SA.

1. <u>SPI</u> – Security Parameter Index.

A Unique number assigned to each and every SA.

2. <u>Sequence number Counter.</u>

– The entire message is divided into no. of Packets.

– Each packet is given a Unique number.

- If the Counter value is 100, then 100 packets are to be transfered.
- For each transformation of a packet, the Counter is decremented by one.

3. Sequence number overflow

The maximum limit of the sequence number allocated, to the Buffer.

eg the ~~size~~ size of the buffer is only 100 packets.
If we try to store more than 100 packets into the Buffer then, overflow occurs.

4. Anti-Replay window

The purpose ~~is to~~ of Anti-Replay window is to avoid duplicate of Packets.

Suppose packet-1 is sent by user A. If any other user will send the same packet, the receiver has to ignore the duplicate of the same copy.

5. **ESP Info~~r~~:** Encapsulating Security Payload.

It provides information regarding all encryption algorithms.

6. **A.H. info:**

It provides information regarding all. Authentication algorithms.

7. **Life time of Security Association (SA)**

The Period of Validity of SA.

(ix) 8. **IP Security Protocol mode.**

we have two protocols

- ESP
- AH.

These two protocols working in two modes.

1. Transport mode.
2. Tunnel mode.

ie. In which mode our IP Security is executed?

Implementation of ESP and A·H Protocols

There are two versions of IPs.

- IP - version 4.
- IP - Version 6.

Format of IPV4.

original header(IP)	TCP	Data.

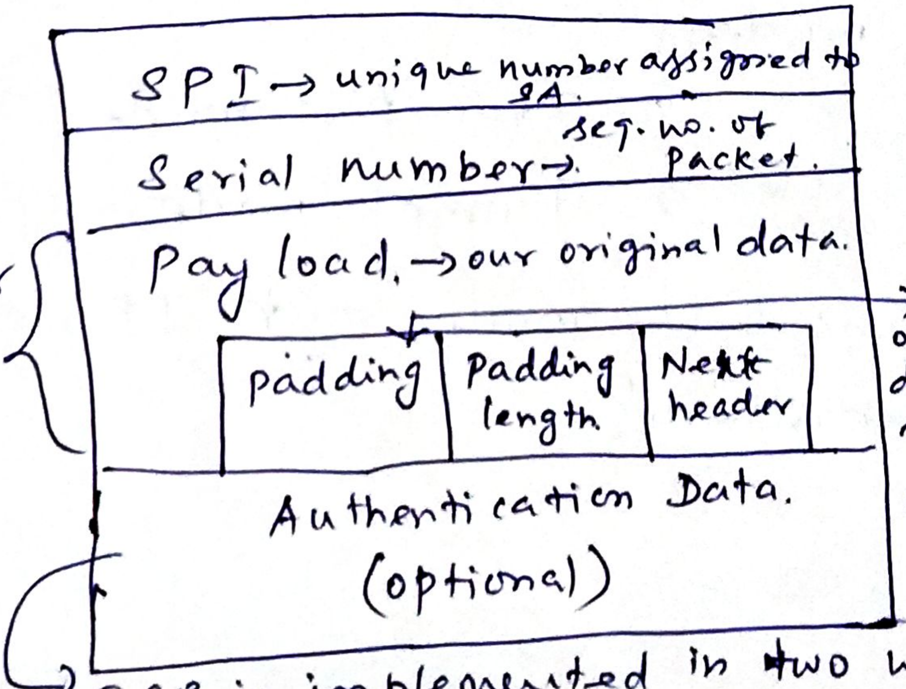
Format of IPV6.

original header (I.P)	Extension header	TCP	Data.

ESP.

Packet format.

This info is encrypted format.



SPI → unique number assigned to SA.
Serial number → seq. no. of Packet.
Pay load. → our original data.
padding
Padding length
Next header
Authentication Data.
(optional)

→ Suppose our original data is divided into fixe
size blocks.
Block size = 1
If we have only 90 bits, we attac
10 more bits to th
fixed bloc

ESP is implemented in two ways.

1. ESP with Authentication
ESP without Authentication (optimal)

Block cipher Modes of operations

ENC Procedure

Stream cipher
(Bit by Bit conversion)
- short message

Block cipher.
(Block by Block conversion)
- long size message.

Here we discuss the modes of operations that are performed on security for ENC and DEC.

There are Five Modes of operations

1. Electronic Code Book (ECB)
2. Cipher Block Chaining (CBC)

} Block cipher

3. Cipher Feed Back mode (CFB)
4. output Feed Back mode (OFB)

} Stream cipher

5. Counter mode.

1. Electronic Code ~~Block~~ Book (ECB)

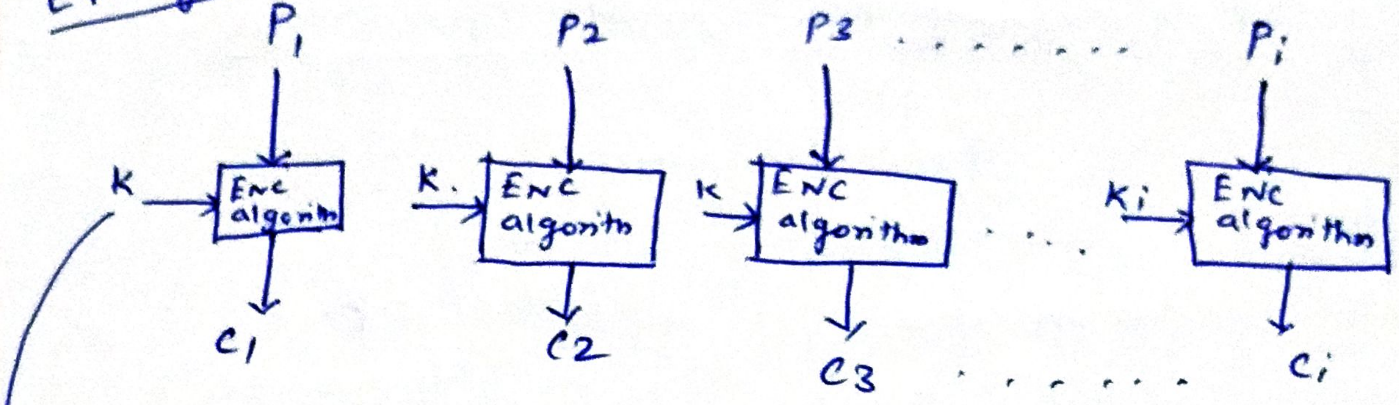
* It is a Block cipher approach

* The Plain Text is divided into no. of Blocks

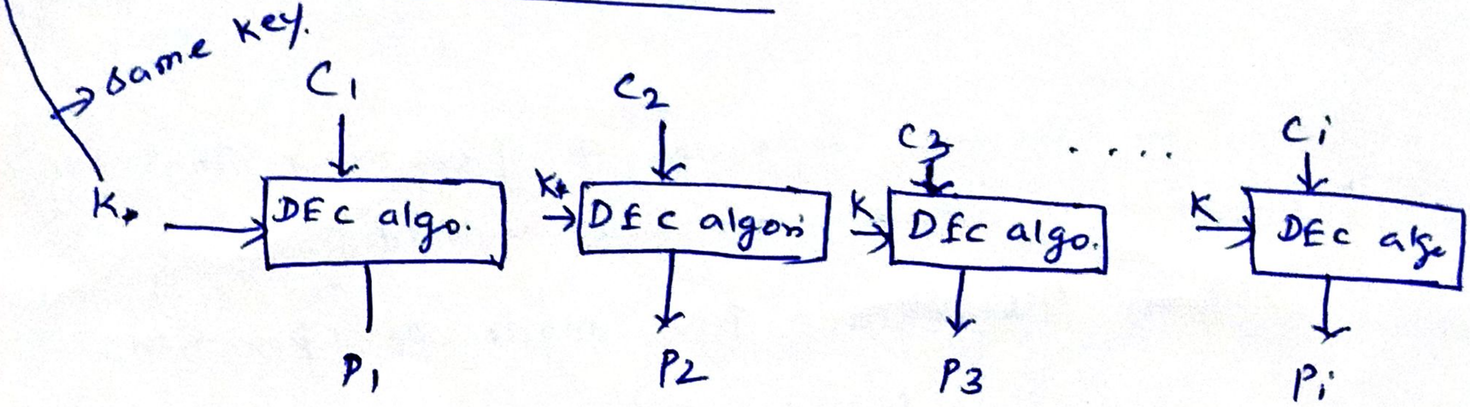
* Perform Encryption or Decryption operation on each block Independently.

(ie) The Result of one block does not affect the other block.

Encryption operation



For DECRYPTION operation



Advantage

Each block is independent of other block.

Disadvantage

* If plain Text Contains more no. of similar blocks. eg P1 and P5 are similar message then the cipher Text is also similar

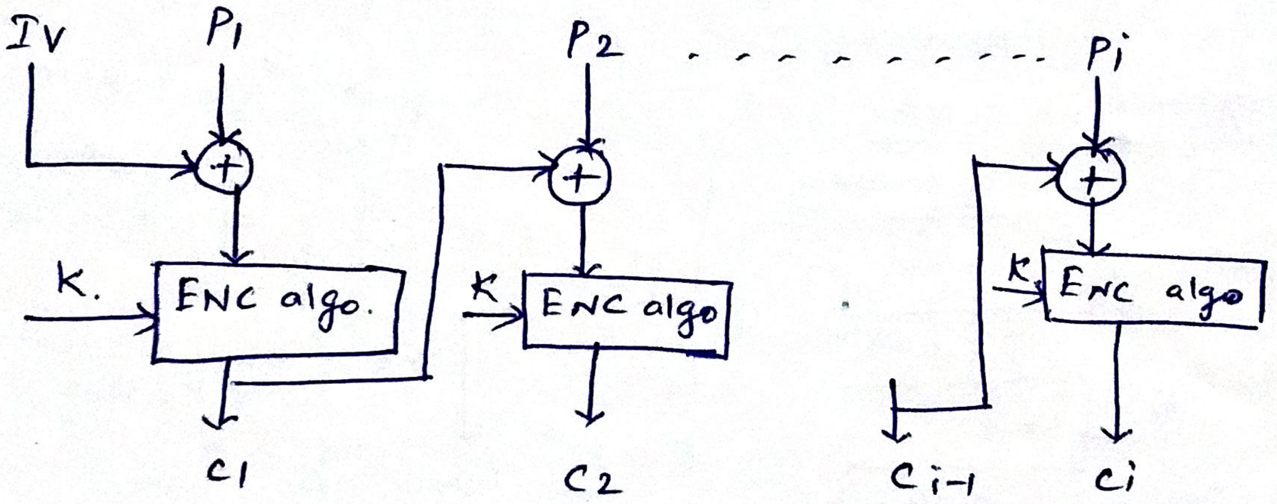
* It is easy for the attacker to attack the message based on freq. of words or bytes in the cipher Text.

* To overcome this Problem we move to next mode of operation.

2. cipher Block Chaining (CBC)

- * The message is divided into diff.: no. of Blocks.
- * output of ~~one~~ First Block is the input of next Block.
- * For the First Block we use Initial Vector ie a Random number.

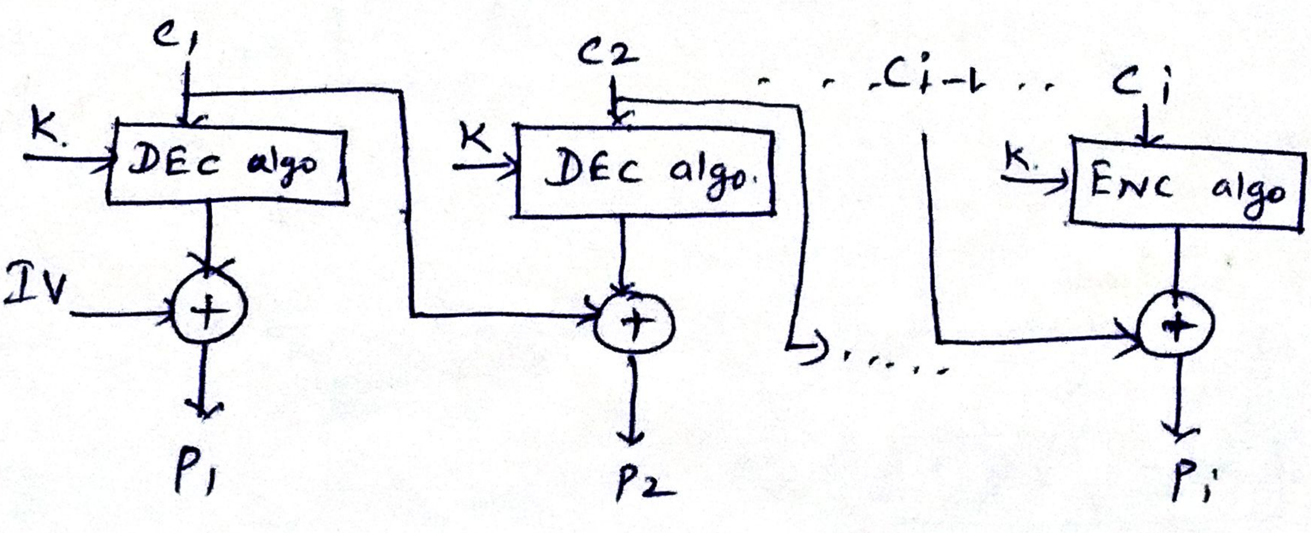
Encryption operation



C_{i-1} = IV random number → Initially we donot ha any cipher Text.

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

DEcryption operation



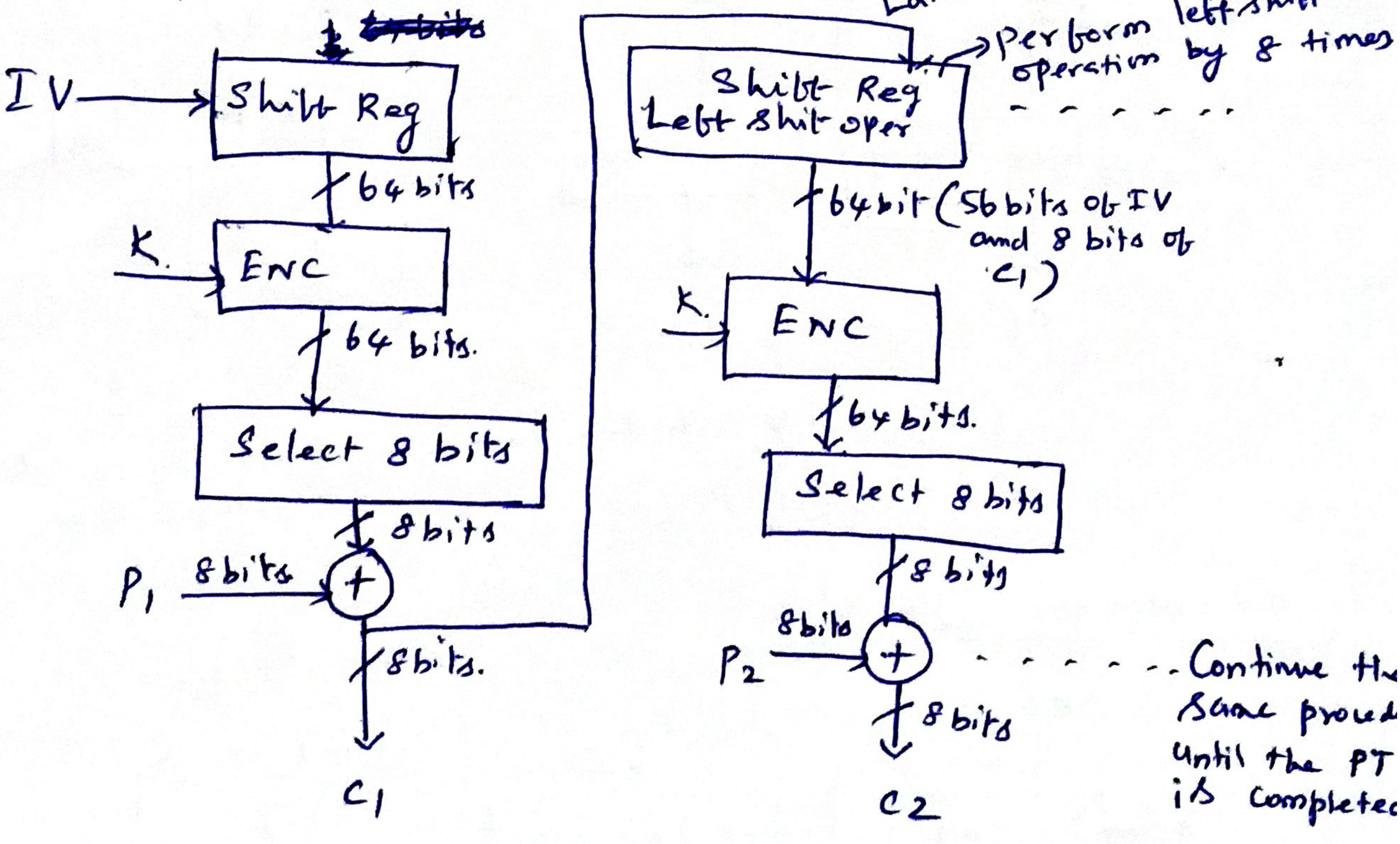
Disadvantages

- * Each and Every Block depends on the Previous Block.
- * If any one Block fails, from that onwards the complete system fails.

3. cipher Feed Back mode (CFB)

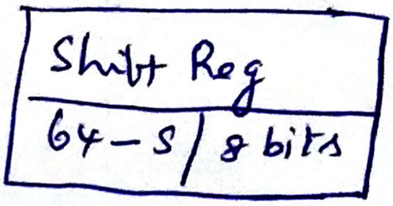
- * In this CFB we apply stream cipher approach.

Encryption operation.

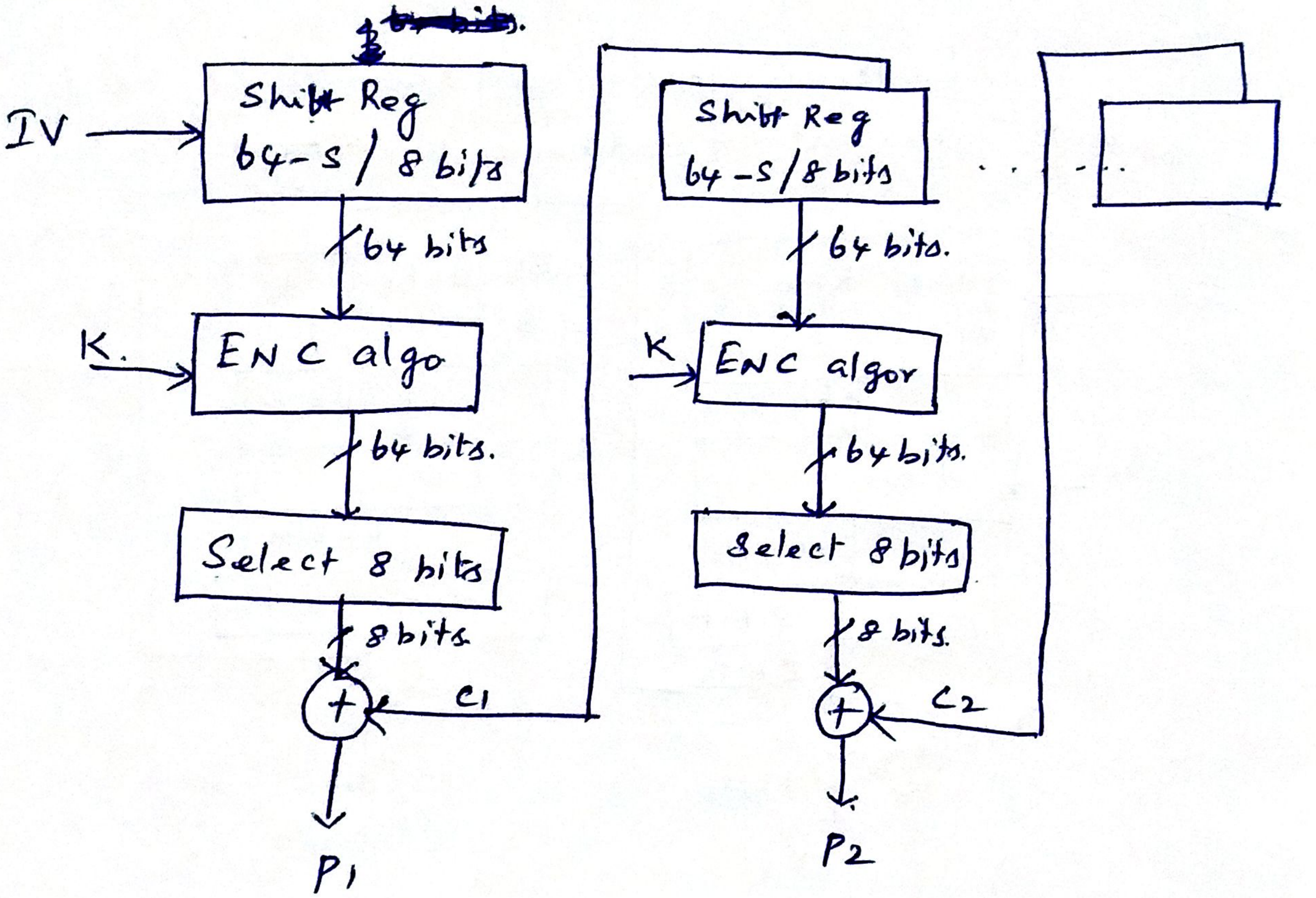


Continue the Same procedure until the PT is complete

S = 8 bits.



Decryption operation



In ENC, the result of Ex-OR is copied into Shift Reg but in DEC, the previous result is Ex-ORed with previous operation.

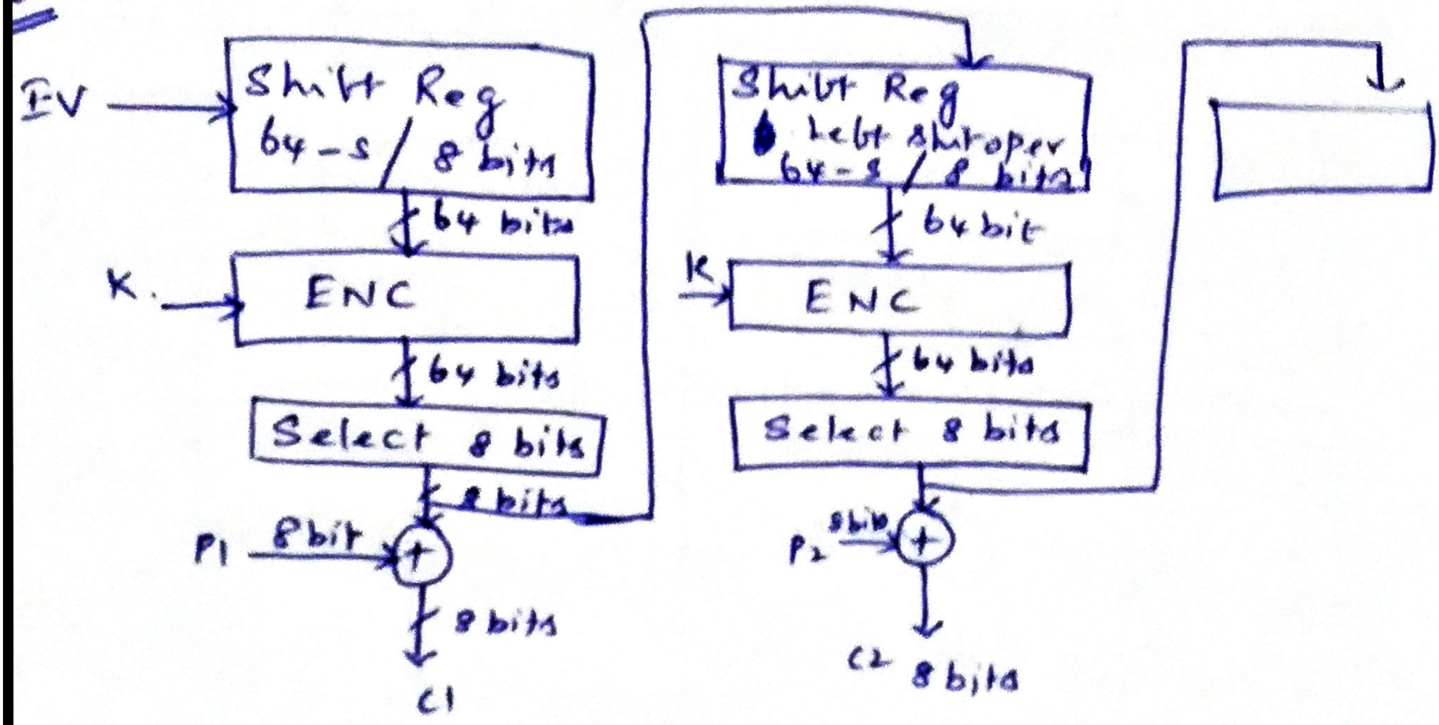
In this mode, one of the problem is Padding suppose in the Stream operation, the Last block Contains 5 bits., we have to add 3 more bits to the Last block.

So Padding is a major problem in Cipher feed back mode.

4. Output feedback mode

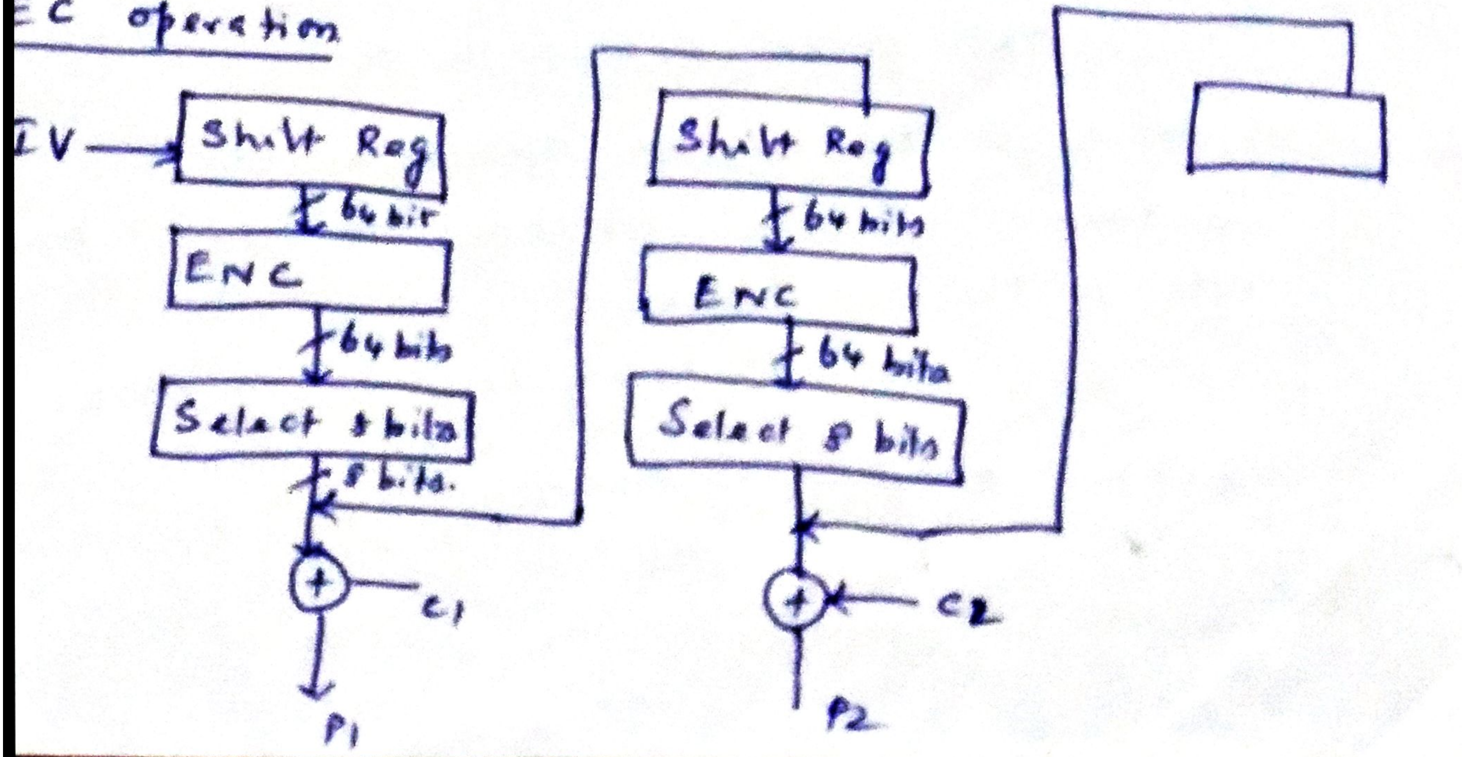
Both cipher feedback mode and output feedback is almost similar.

c operation



In this mode, we do not pass the cipher Text to the next Block, instead, We Pass Selected 8 bits from Encrypted data.

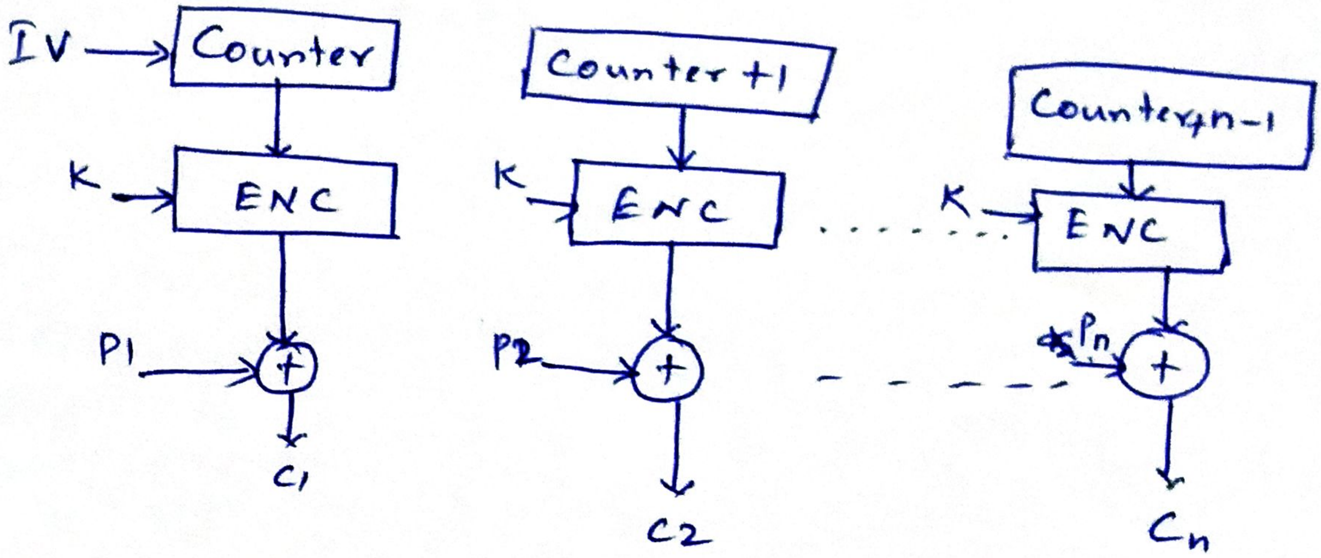
EC operation



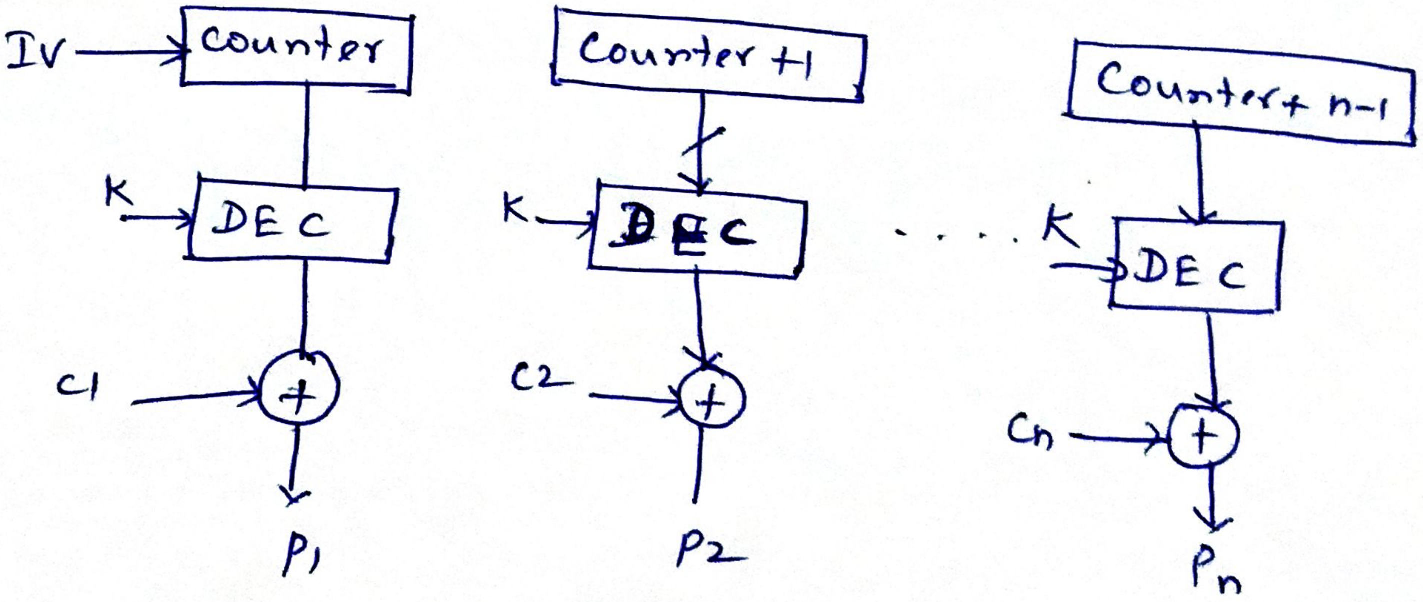
5. Counter mode:

In counter mode, a Counter Value is Placed for each and every Block, after completion of a block, a counter Value is incremented by one.

ENC operation



DEC operation



Advantage

1. Hardware efficiency → multi tasking.
   → ENC and DEC are Performed in Parallel
2. Software efficiency
3. Security.

Number Theory

To Perborm large mathematical calculations in RSA algorithm and Dibfie Hellman Key exchange algorithm, we are badly in need ob Number Theory concepts.

Here we discuss the Basic Concepts ob Number Theory.

1. Prime Number
2. Relative Prime Number
3. Modular arithmatic
4. Congruent modulo.

1. Prime Number

If a Number P is divisible by one(1) and itselb, then the number P is said to be a prime Number.

eg

P = 5 is a prime number.

5 is divisible by 1 and the number itself.

So 5 is a prime Number.

$$5 \Rightarrow \overset{\checkmark}{1}\ \underbrace{\cancel{2}\ \cancel{3}\ \cancel{4}}_{\text{Not divisible}}\ \overset{\checkmark}{5}$$

Relative Prime Numbers.

Two numbers a and be are said to be relative Prime Numbers if a and b do not have common factors

ie $\gcd(a, b) = 1$.

eg

$a = 15$, $b = 28$

* factors of $a = \{1, 3, 5\}$ → except that numbers ie 15

factors of $b = \{1, 2, 4, 7, 14\}$ → ~~expedet~~ except that number. ie 28.

* Both factors of a and b do not have a common numbers.

then $a = \{\cancel{1}, 3, 5\}$

$b = \{\cancel{1}, 2, 4, 7, 14\}$

Factors of a~~m~~ and b $= \{2, 3, 4, 5, 7, 14\}$

* otherwise Calculate $\gcd(a, b) = 1$

$\gcd(15, 28) = 1$

15)28(1

15

13)15(1

13

2)13(6

12

1)2(2

2

0.

∵ By dividing which number we get remainder 0, that number becomes gcd. In this eg, by dividing 1, we get remainder 0 for the numbers 15 and 28.

∴ $\gcd(15, 28) = 1$.

Then a and b are relatie Prime nos.

3. Modular Arithmatic

The mod function accepts only integer numbers as input and returns output as the remainder of any two integer numbers.

$$\boxed{a \ \text{mod} \ n = \text{remainder.}}$$

eg -1 $a = 15$, $n = 3$ then $15 \text{ mod } 3 = 0$

eg-2 $a = 15$, $n = 4$ then $15 \text{ mod } 4 = 3$

Quotient

$4)15(3$
12
$\boxed{3}$

Remainder

4. Congruent modulo. (Symbol $\equiv$).

Two integers a and b are said to be Congruent to n.
↳ integer number

ie represented as

$$a \ (\text{mod } n) = b \ (\text{mod } n)$$

⇓ implies

$$\boxed{a \equiv b \ (\text{mod } n)} \text{-}✱.$$

For Fermat's Theorem, Chinese theorem and Euler's theorem we have to use this Congruent theorem. So this formula is very much important for many algorithms.

$a \equiv b \pmod{n}$

a Congruent to b w.r.t n.

Here we divide a by n, we get the remainder b.

Illy we divide ~~bg~~ by n we get remainder as a.

Ex. $a = 73$ $b = 4.$ $n = 23.$

a mod n

73 mod 23

$23 \overline{)73} (3$

69

Remainde $= \underline{4}$

b mod n

4 mod 23

Remainder = 4.

$a \equiv b \pmod{n}$

$73 \equiv 4 \pmod{23}$

$73 \bmod 23 \equiv 4.$

$\overline{4.}$

Properties of Congruent modulo.

1. $a = b \pmod{n}$

This is satisfied if $\frac{n}{(a-b)}$

ie n is multiples of (a-b)

ex $a = \overset{30}{\cancel{15}}$, $b = \overset{10}{\cancel{8}}$, $n = 5.$

$(a-b) = (30-10) = 20.$

$= 5 \times 4 = 20 \rightarrow$ a-b is the multiple of n.

2. $a \bmod n = b \bmod n \Rightarrow a \equiv b \bmod n.$

3. If $a \bmod n = b \bmod n$, $b \bmod n = c \bmod n$

then $a \bmod n = c \bmod n.$

$$\Downarrow$$

$$a \equiv c \bmod n.$$

4. $((a \bmod n) + (b \bmod n)) \bmod n. = (a+b) \bmod n$

$((a \bmod n) - (b \bmod n)) \bmod n = (a-b) \bmod n.$

$((a \bmod n) * (b \bmod n)) \bmod n = (a * b) \bmod n.$

$((a \bmod n) / (b \bmod n)) \bmod n = (a/b) \bmod n.$

5. Commutative law

$(a+b) \bmod n = (b+a) \bmod n.$

$(a * b) \bmod n = (b * a) \bmod n.$

6. Associative Law

$((a+b)+c) \bmod n = (a+(b+c)) \bmod n.$

$((a * b) * c) \bmod n = (a * (b * c)) \bmod n.$

7. Identity.

$(0+a) \bmod n = a \bmod n.$

$(1 * a) \bmod n = a \bmod n.$

PGP - (Pretty Good Privacy)

* The Purpose of PGP is to provide more security to the data.

* PGP uses both Symmetric and Asymmetric encryption algorithms to provide more security

ie to provide Privacy for data.

* PGP Provides the following Services

1. **Authentication**

   Authentication is provided by using Digital Signature.

2. **Confidentiality**

   confidentiality is provided by using Symmetric Key ENC.

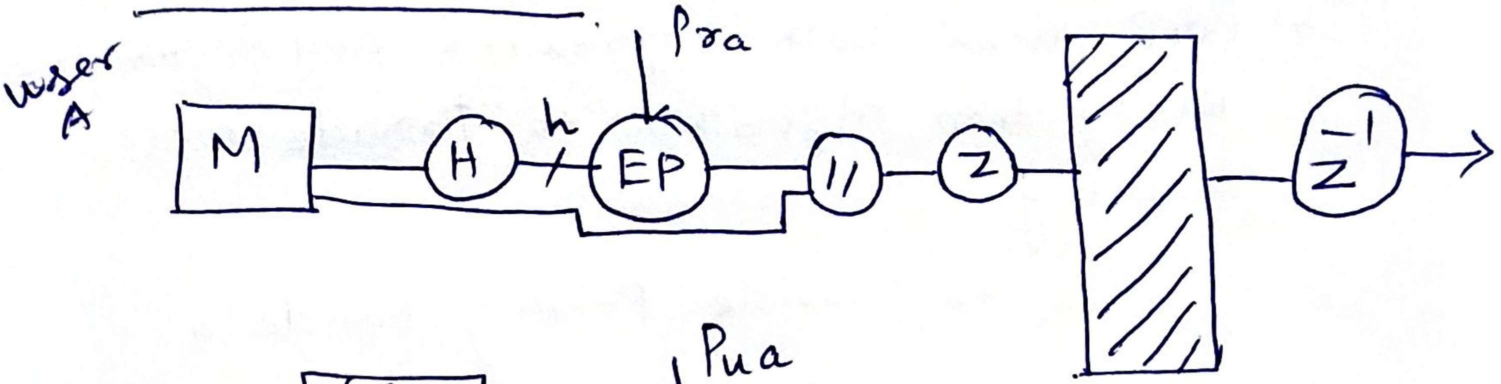
3. **E-mail Compatibility**

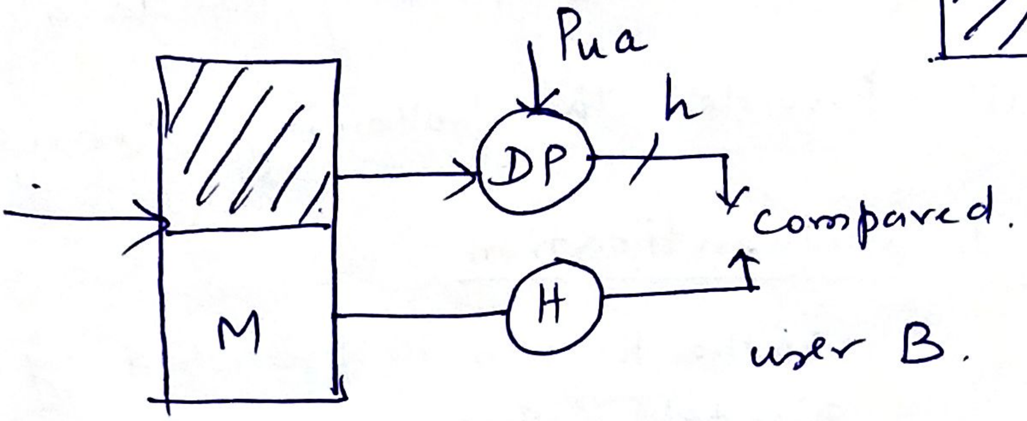
   is provided by using Confidentiality Before sending E-mail to the receiver, we must convert E-mail into Radix 64 format

4. ZIP function

ZIP algorithm used to transfer data b/w sender and receiver.

1. Authentication





- User A has some message M.
- Perform Hash function on M, it generates
- hash code - h (another name for hash code is ~~degit~~ digest.)

- We apply encryption algorithm on the hash code.

- Here our Concept is to have Authentication

* Authentication is provided by using digita signature.

* Digital Signature means encryption using the private key

* The encrypted hash code is attached with the original Message.

* Now we apply Compression function And the message is zipped, or archieve file format.

* This zipped message is received by the receiver.

* The receiver performs inverse zip (z^{-1}) operation.

* If 2^{-1} is performed, the zipped Message gets divided into two parts.

* First part is encrypted hash code and Send part is the original Message.

* For the encrypted hash code, we apply (DP) Decryption algorithm using Pua, then the result of ~~de~~ DP, gives the original hash code.

* For the second part of the original message, we apply the (H) Hash algorith.

* This gives another hash code.

* Now, Compared both hash codes.

* If both hash codes are same, then the Message is Correctly sent to the user B.

Disadvantage

Here there is no. Confidentiality For eg we directly transfer the message M to the append function.

This should be avoided, for getting Confidentiality.

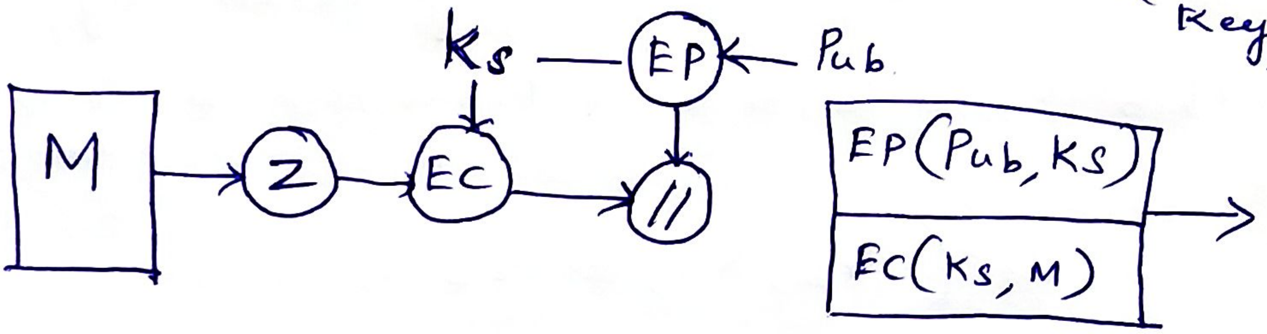
So Confidentiality is missing, but Authentication is provided,
↳ using sign.

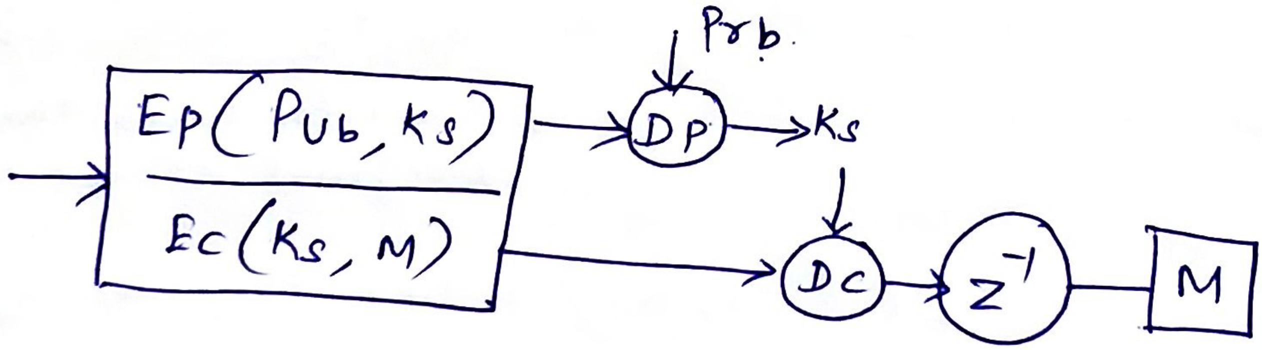
2. Confidentiality

Confi., is provided by using Conventiona Enc., ie Symmetric Key ENC.

EP → Pair of keys. (Pra)
DP → Pair of key (Pu a) → Previous algorithm.

EC
DC → Same Key is used for both. ENC and DEC. (Convential Key).





* The Compressed Message is ENC by using a secret key. Ks.

* The Secret Key should be transfered in a Secured manner to the receiver.

* So Ks is encrypted by using Pub public key of user B.

* Combine both Enc Message M and ~~encr~~ ENC Ks

EP(Pub, Ks) → ENC of secret key Ks using Public key of user B.

EC(M, Ks) → ENC of Message M using secret key Ks.

* This Information is transfered to Receiver (user B)

DP(Prb, KS) → DEC of secret key KS using Private key of user B.

→ This produces the original secret key Ks.

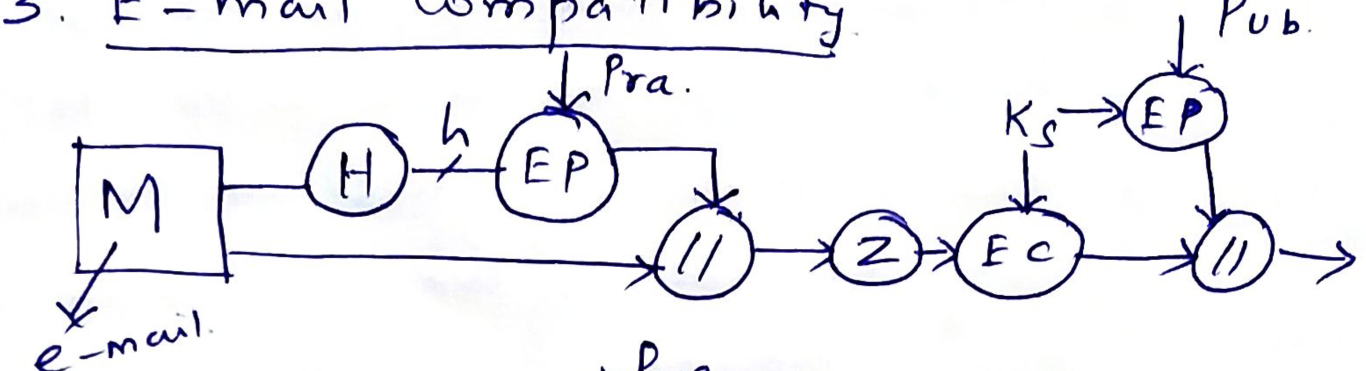
DC(M, KS) → DEC the Message by using ~~seret~~ secret key Ks.

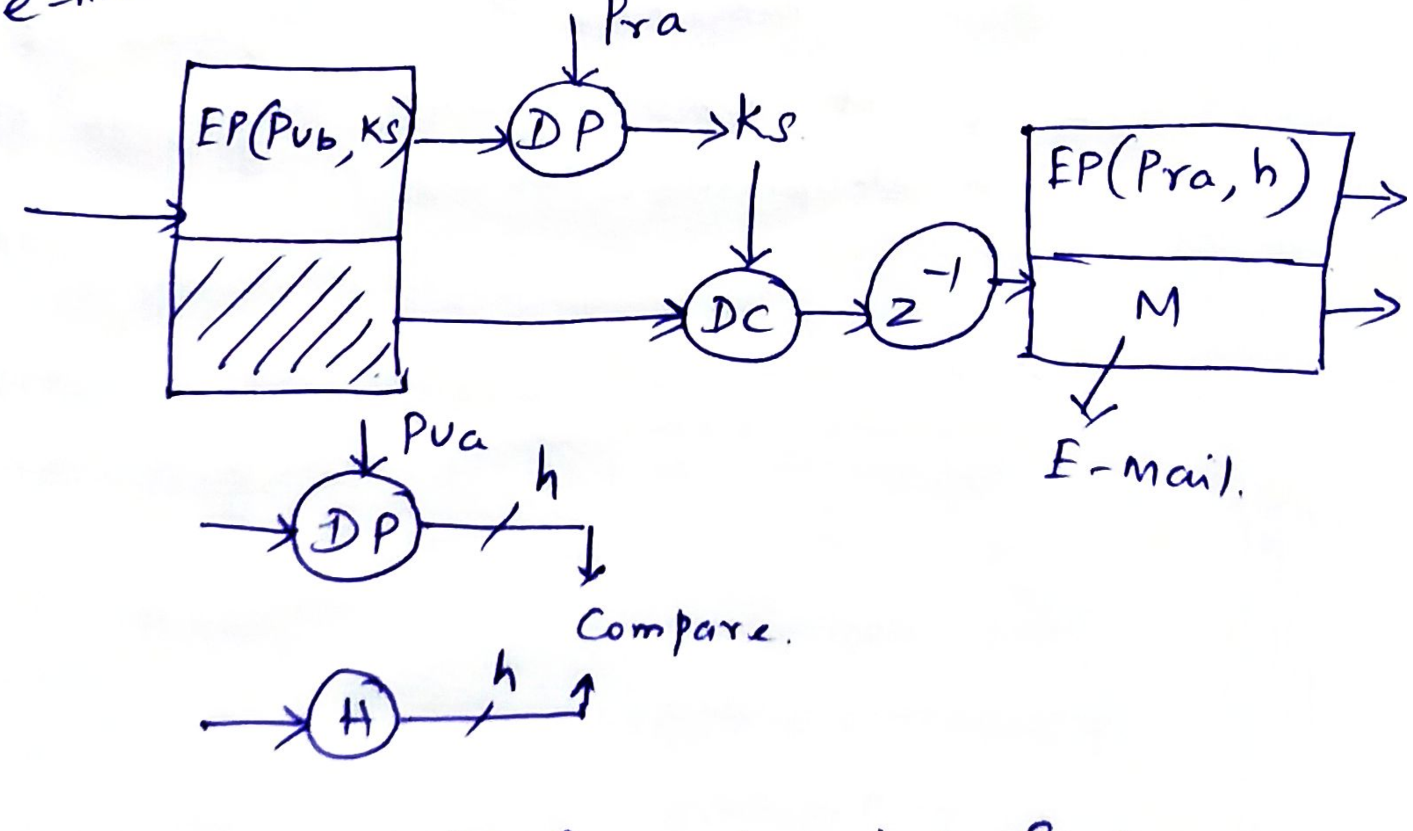
* Perform 2^{-1} inverse operation, we will get the original Message M.

* Here No Authentication is achieved. Because we have no digital signature
~~[illegible]~~ Private key

* The Purpose of PGP is to Provide more Security. So we must Provide both Authentication and Confidentiality.

3. E-mail Compatibility





* E-mail is Converted into Radix-64 format.

[redacted]

* The above fig results both Authentication and Confidentiality.

* We Combine the first two figs into single one.

* <u>Disadvantage.</u>

  - The Sender and Receiver should have the same versions of PGP.
  - The PGP is very difficult Process why because, it uses a Combination of Symmetric and Asymmetric Ke
  - ie it uses hybrid Keys, so it is a difficult Process.

Public key cryptography. (PKC)

Before discussing public key crypto., Let us discuss the problem of Symmetric key cryptography.

Symmetric key Cryp → use only one key for both ENC and DEC.

so it is very easy to break the Data because for both ENC and DEC, the same key is used.

To overcome this problem we have to discuss the Asymmetric cryp., or Public key cryptography

Here Sender performs ENC operation using one key. The Receiver performs DEC operation on another key.

In PKC, the elements are,

1. Plaintext
2. ENC algorithm. → Converts PT to unreadable forms
3. Keys — Public key / Private key.
4. DEC algorithm. → Converts CT to PT.
5. Cipher Text.

The procedure for Sending messages to the Receiver are.

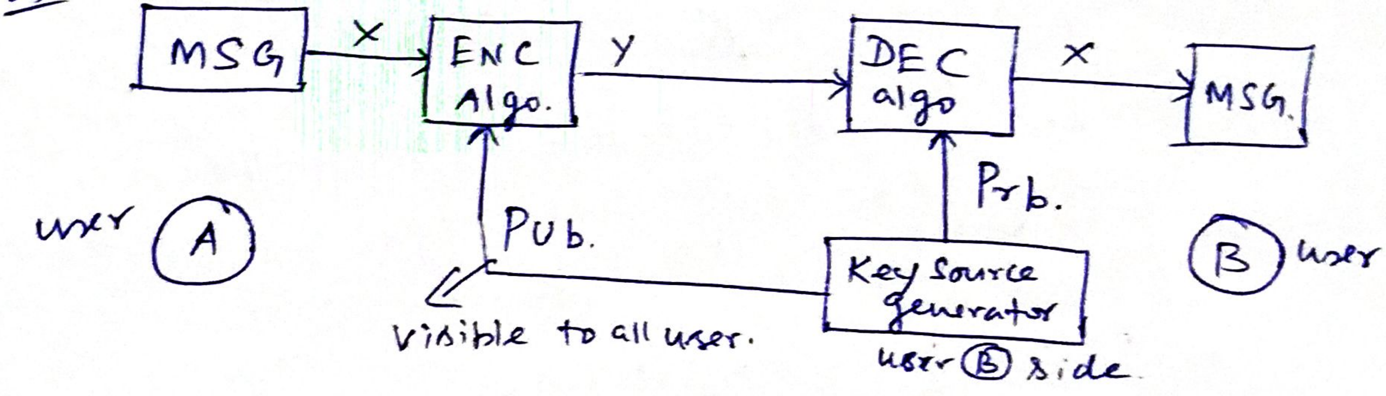
1. Both Sender and Receiver generate a pair of keys.

2. ~~In~~ From the Pair of keys, one key is placed in Public Register → this key is visible to ~~all~~ all other users in a System.

3. Another key is kept as Private key ie Secrete key → this key is visible to only one user who genrates this key.

Once theses Keys are generated, we perform ENC and DEC operation

eg



* user Ⓐ wants to sends some MSG, first we have to perform ENC operation.

* To Perform ENC algorithm we use a key. eg the public key of user Ⓑ

* User Ⓐ performs ENC using the Public key of user Ⓑ

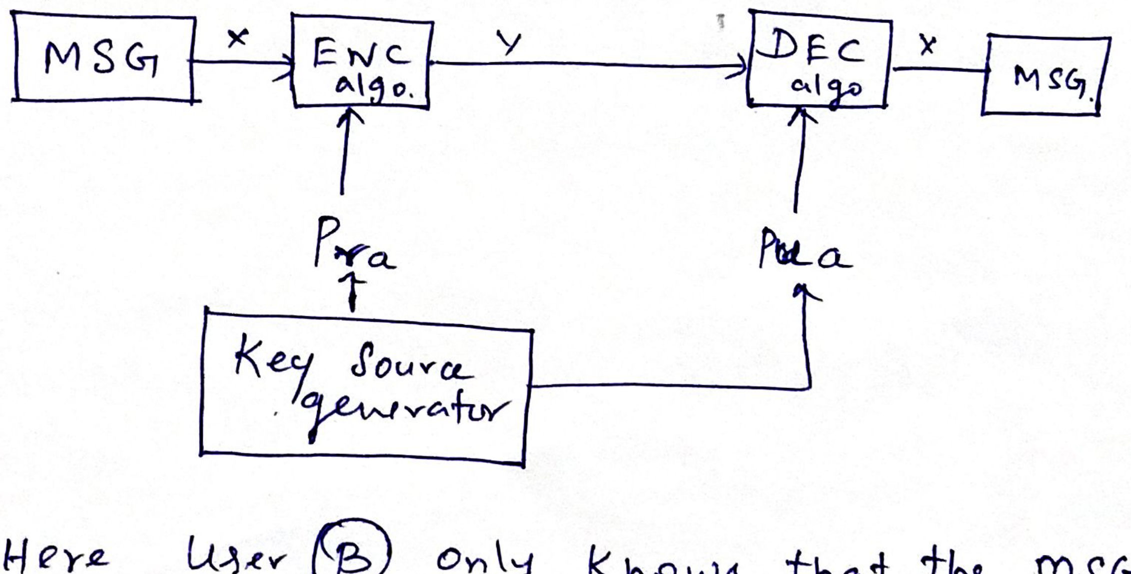
* After performing ENC ie y is tramfered to user B:

* In DEC algo. we use the Private key of user B

* Though Public Key is visible to every one in the System, only user B can decrypt the MSG. because the Secrete key ie private key is used by user B

In the Similar fasion, we more to one more type of ENC.

Authentication

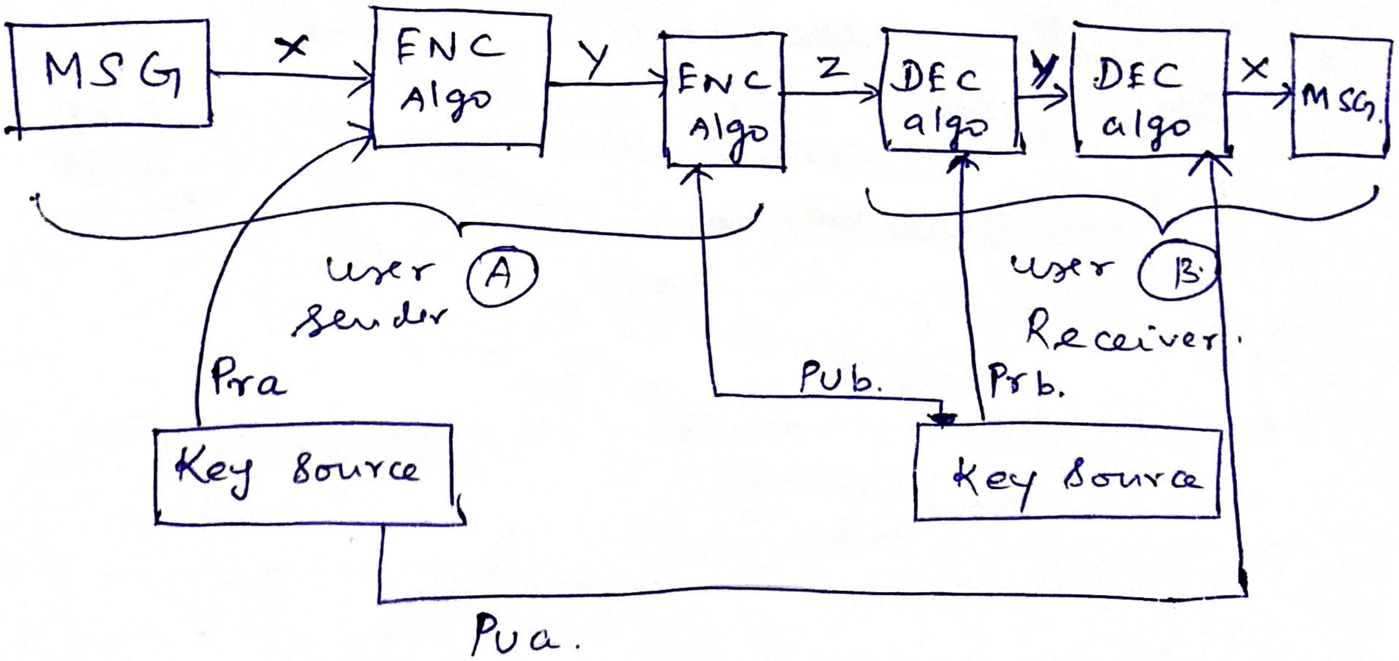


Here User B only Knows that the MSG is encrypted. others do not Know the whether the MSG is encrypted or not.

In Both Cases we use a pair of Keys.

Suppose, you want to Perform Confidentiality

Perform ~~D~~ ENC operation twice in Sender side, and ~~D~~ Perform DEC operation twice in receiver side.



Strength of DES. → Key Size
→ nature of the algorithm.

The Strength of DES lies on two facts:

a) The use of 56 bit Keys.

- 56-bit key is used in encryption,
- there are 256 possible Keys.
- A brute force attack on such number of Keys is impractical.

b) The nature of algorithm

- Cryptanalyst can perform cryptanalysis by exploiting the characteristic of DES. algorithm but no one has succeeded in finding out the weakness.

Weakness

Weakness has been found in the design of the cipher:

a) Two chosen input to an S-box can create the same output.

b) The purpose of initial and final permutation is not clear.

Timing Attack.

A timing attack is one in which inbormation about the key or the plain-Text is obtained by observing how long it takes a given implementation to perbosm decryption on various cipher-text.

A timing Attack. exploits the fact that an encryption or decryption algorithm obten takes sliguhtly dibberent amounts ob time on dibberent inputs.

RSA Algorithm

named for its inventors Rivest, Shamir, and Adleman

- * Public key algorithm or Asymmetric key algorithm.
- * In this algorithm, we use two different keys for encryption and decryption Areas

The Procedure for implementing RSA algorithm

1. Select two Large prime numbers.

   p, q (Private.) ↳ unknown to all other users.

2. Compute $n = p \times q$

   ↳ Public

3. Public key $= \{e, n\}$ → encryption exponential key

   Private $= \{d, n\}$ → decryption exponential key

$\underline{\underline{e}}$?

$\gcd(e, \underline{\phi(n)}) = 1$

↳ Euler's totient function.

$$\phi(n) = (p-1)(q-1).$$

$$1 < e < \phi(n)$$

5

~~(P-1) 6~~

~~3 x4~~

S. Saravanan.

4. $\underline{d}$?

$$de \bmod \phi(n) = 1$$

(or)

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

PT = M
CT = C

For encryption

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

For decryption

$$\boxed{M = C^{d} \bmod n}$$

Security

Mainly RSA algorithm is unsecure due to 4 reasons.

1. Brute force attack. → try all possible cases of Private key.
2. Mathematical attack.
3. Timing attack.
4. choosen ciphertext attack.

1. Brute force attack.

* trying all possible cases of Private key.
* to overcome this problem, choose key size as large one.

2. Mathematical Attack,

many different. approaches for mathematical attacks but for all approaches, the Common effort is

factoring of N. ie $N = P \times q$.

↳ this number is Public

whenever the factors of N is identified, based on factor we calculate $(P-1)(q-1)$

based on this we will get ~~e-value~~ &
e-value,
d-value.

3. Timing Attack

The attackers identifies the running time of the algorithm.

↳ decryption

To reduce the timing Attack, we have some possible cases

1. Constant exponen~~tiatime~~tiation time

* simply we add some fixed ~~day~~ delay to each and every algorithm

eg

An algorithm completes in 3 ms. ↓ running time we will add some delay ie 2 ms. Now the algorithm completes in 5 ms so that the attacker does not identify the actual running time.

but ~~P~~ Performance is degraded.

2. <u>Random delay.</u>

delay time is randomly determined, not fixed

3. <u>blinding</u>

<u>multiply</u> the Cipher Text by some Random number

4. <u>Choosen cipherText attack</u>

Simply it exploits the properties of the RSA algorithm.

<u>Ex</u>

$P = 3 \qquad q = 11$

$n = P \times q = 3 \times 11 = 33$

$\phi(n) = (P-1)(q-1)$

$= (3-1)(11-1) = 20.$

<u>e</u>

$\gcd(\phi(n), e) = 1$

$\gcd(20, e) = 1$

$1 < e < \phi(n)$

Now try all possible cases of e value.

eg assum $e = 2$

$$1 < e < \phi(n)$$

gcd procedure

$2 \overline{)\, 20\,} (10$

20

0

Our aim is gcd become 1

~~so e ≠ 2.~~

so $e = 2$ ✗ wrong.

$\boxed{e = 3}$ ✓ right.

$3 \overline{)\, 20\,} (6$

18

$2 \overline{)\, 3\,} (1$

2

$1 \overline{)\, 2\,} (2$ ✓

2

0

d

$$de \bmod \phi(n) = 1$$

$$d * 3 \bmod 20 = 1.$$

$\boxed{d = 7}$

$$7 * 3 \bmod 20 = 1$$

$$21 \bmod 20 = 1$$

Suppose Plain. Text

$M = 5$

Cipher Text $C = M^e \bmod n$

$= 5^3 \bmod 33$

$= 125 \bmod 33$

$= 26.$

$33 \overline{)\,125\,}\,(3$

99

26

Cipher Text $= 26.$

Plain. Text

$M = C^d \bmod n$

$= 26^7 \bmod 33$

$= 26^5 \times 26^2 \bmod 33$

$= 26^1 \times 26^2 \times 26^2 \times 26^2 \bmod 33$

$= \underline{\underline{5}}.$

Ex eg - 2

$P = 3\ ,\ q = 5$

$n = P * q = 15$

$\phi(n) = (P-1) * (q-1) = 2 \times 4 = 8$

$\phi(n) = 8$

e

$\gcd(e, \phi(n)) = 1$

$\downarrow$

3, 5, 7

$\boxed{e = 3}$

$\gcd(3, 8) = 1$

d

$d * e \bmod \phi(n) = 1$

$d \times 3 \bmod 8 = 1$

$\downarrow$

3

$9 \bmod 8 = 1$

$\boxed{d = 3}$

Public key $= \{e, n\} = \{3, 1$

Private key $= \{d, n\} = \{3, 1$

ENC

$\to$ plain Text.

$M = 4 < n$

$c = M^e \bmod n$

$= 4^3 \bmod 15$

$= 64 \bmod 15$

$c = 4$

$\hookrightarrow$ cipher Tex

DEC

$M = c^d \bmod n$

$= 4^3 \bmod 15$

$= 64 \bmod 15$

$M = 4$

$\hookrightarrow$ plain Text

Types of Authentication.

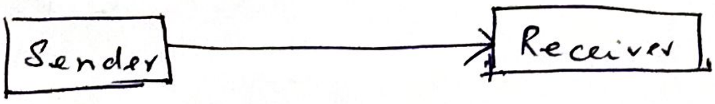
* Authentication means User identity is veribied.

* we have three types of Authentications.

1. Message Encryption.
2. Message Authentication Code (MAC)
3. Hash Function.

Eg

Consider a N/W Consists of no. of users.



Both sender and Receiver must be authenticated. ie. The sender should check that he must send the message to the right person (authorized person)
~~Mg~~ Similarly the Receiver must check that he must ~~send~~ receive the message ~~to the~~ from right person (authorized Person)

The identity of sender and Receiver must be veribied by the above three ways.

1. Message Encryption

Encryption → Converting PT into CT.

Here CT acts as Authentication.

2. MAC

In MAC, we generate MAC Code. that acts as Authentication.

$$\underset{\text{function}}{e}\left(\underset{PT}{M}, \underset{Key}{K}\right) = \text{code}$$

* Code length is fixed.

* This fixed length code is called MAC.

* This MAC is used for Authentication.

* The receiver receives the ~~A~~ MAC from sender and he has to Compare the received MAC with newely generated MAC. If both MAC code is same then both A and B are Authenticate.

3. Hash function

* This is Similar to MAC.

* $h(M, \ ) = \text{Code}$ (fixed length).

This code is called hash code

* This hash code is used for Authentication

* Simply we apply hash function on plain Text

X. * Hash function does not depend on Key. This is the difference b/w MAC and HASH

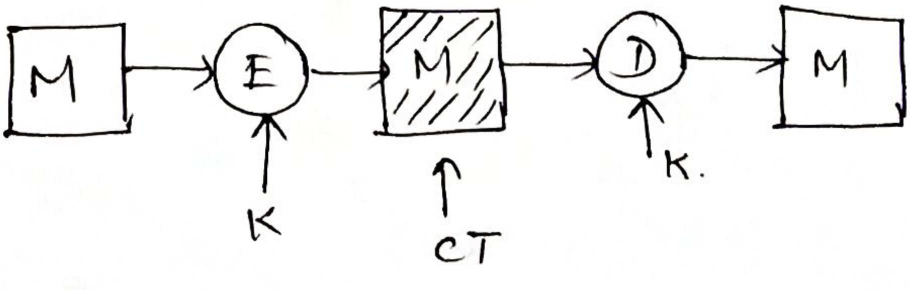
Now we discuss these three types of Authentication in detail.

1. Message Encryption

* cipher Text acts as Authentication.

* Encryption → Symmetric → Single key
  Encryption → Asymmetric → Public
  Asymmetric → Private

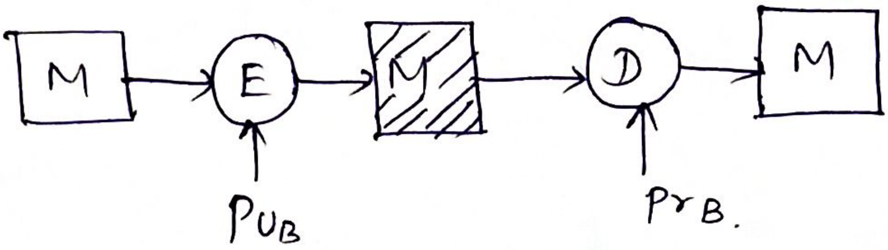
* Symmetric



Here We use single key for both Enc and DEC.

CT acts as Authentication.

Asymmetric

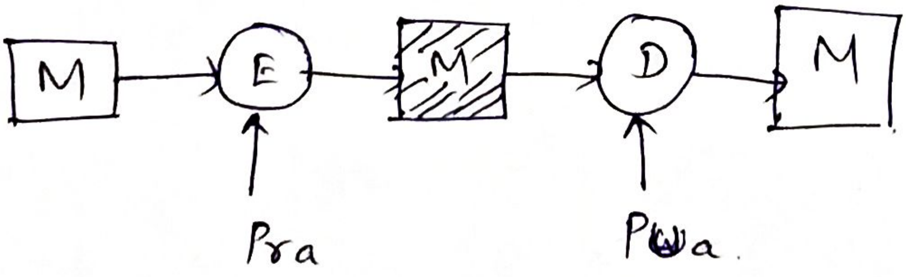


* only a single user's Pair of keys should be used.

* In this case User B's keys are used.

* For Enc, Public key of user B is used
For DEC, Private key of user B is used.

* Conbidentiality is acheived or Provided but no authentication.

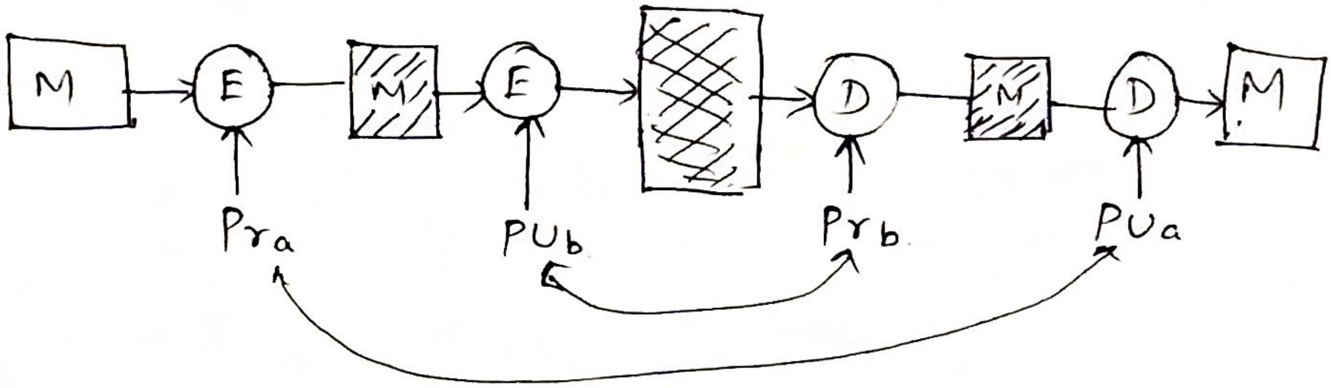


* For Enc, Private key of user A is used.

* For DEC, Public key of user A is used.

* Authentication is achieved or Provided but no Confidentiality.

* Our aim is to provide both Confidentiality and authentication.



* By doing ENC operation twice using Private key of user A and Public key of user B. and

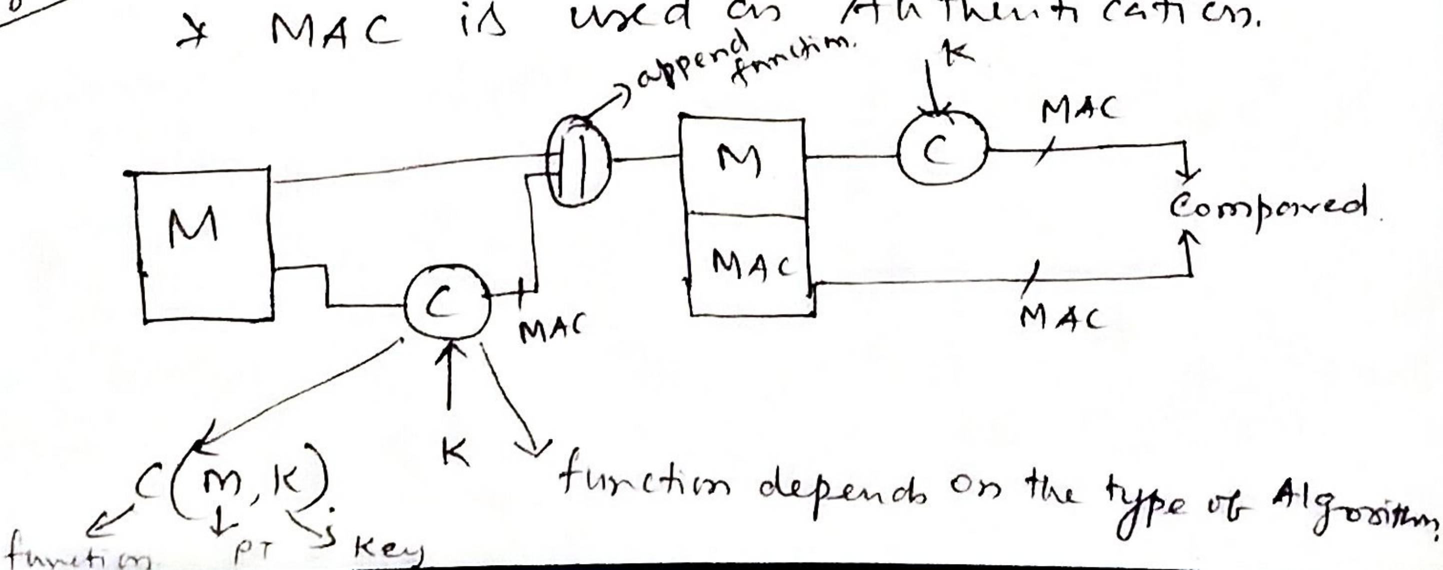
* by doing DEC operation twice using Private Key of user B and Public Key of user A,

* we will achieve both Confidentiality and authentication.

2 Message Authentication Code (MAC)

stage-1

* MAC is used as Authentication.



* M is the message, we apply ~~thi~~ a function C on this message, and it produces a MAC.

* This MAC is appended with the original message M.

* The original Message M and MAC are to be transfered to Receiver.

* At the receiver side, the same function C is applied with the same key.

* Here also it generates MAC.

* The generated MAC and the transfered MAC both are compared.

* If both MAC are same, Authentication is achieved.

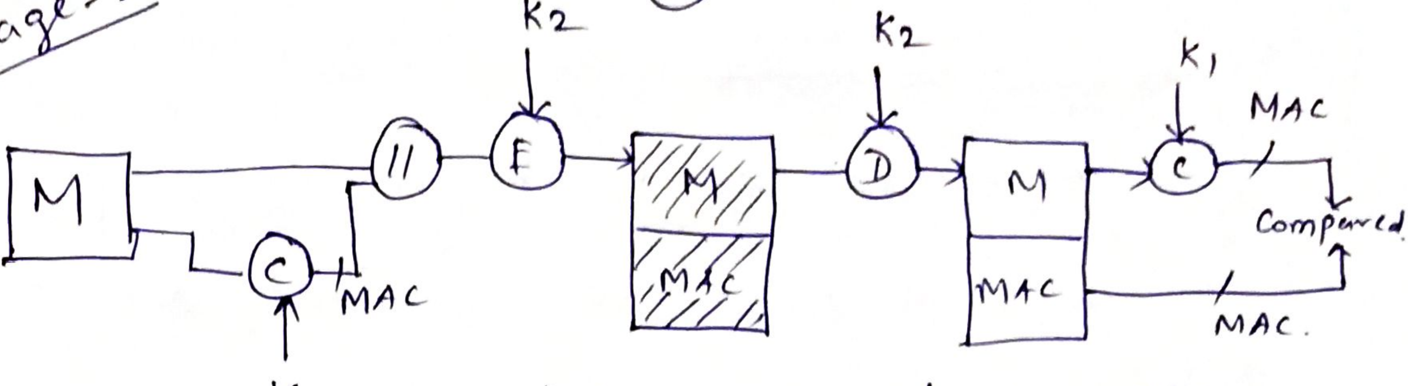
<u>Problem</u>

* directly transfer the PT. ie Message M.

* There is a chance for modifying the PT.

(ie) No integrity.

stage-2

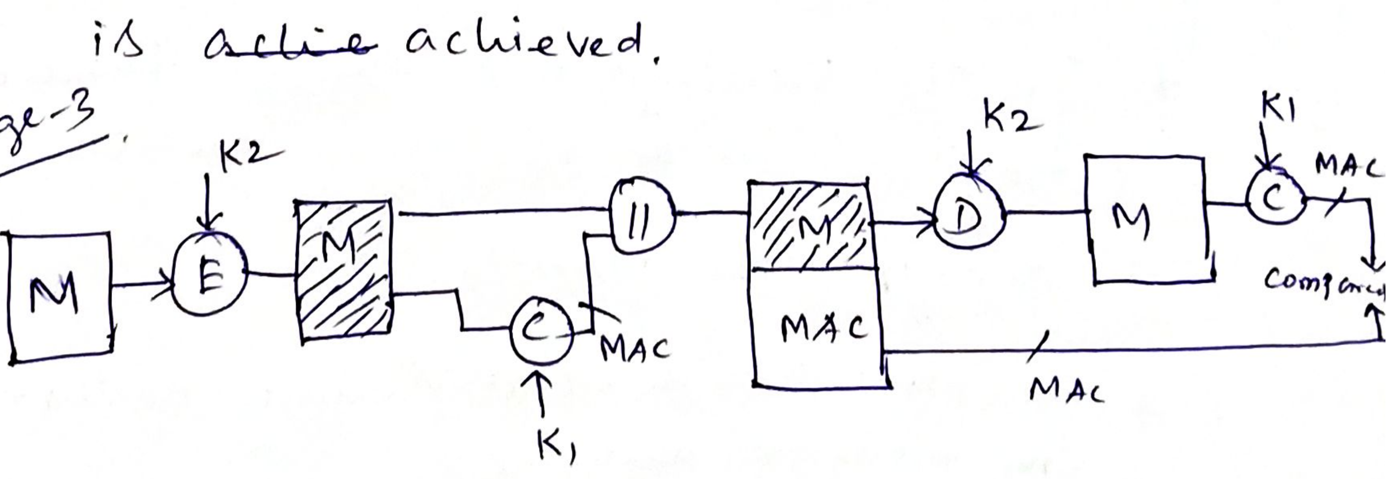


Authentication tied to PT.

- * Here, after Combining origonal Message M and MAC, we perform ENC operation.

- * At the Receiver side, we perform DEC operation.
- * again apply the function C on the message M received from sender, it generates one more MAC
- * Finally generated MAC and transfered MAC are Compared.
- * If both MAC are same, Authentication is ~~achie~~ achieved.

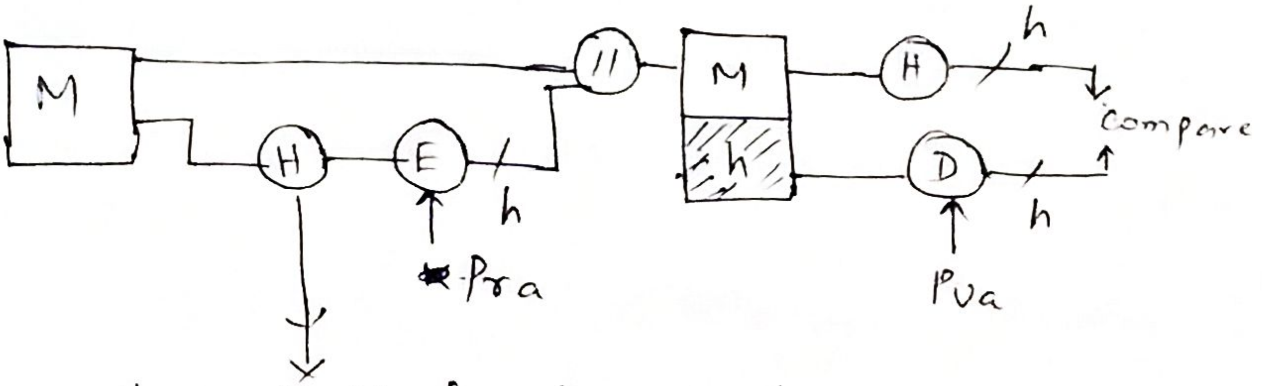
stage-3



Authentication tied to CT.

3. Hash function

- fixed length Hash code is generated
- Hash code acts as Authentication
- No Key is used in Hash function

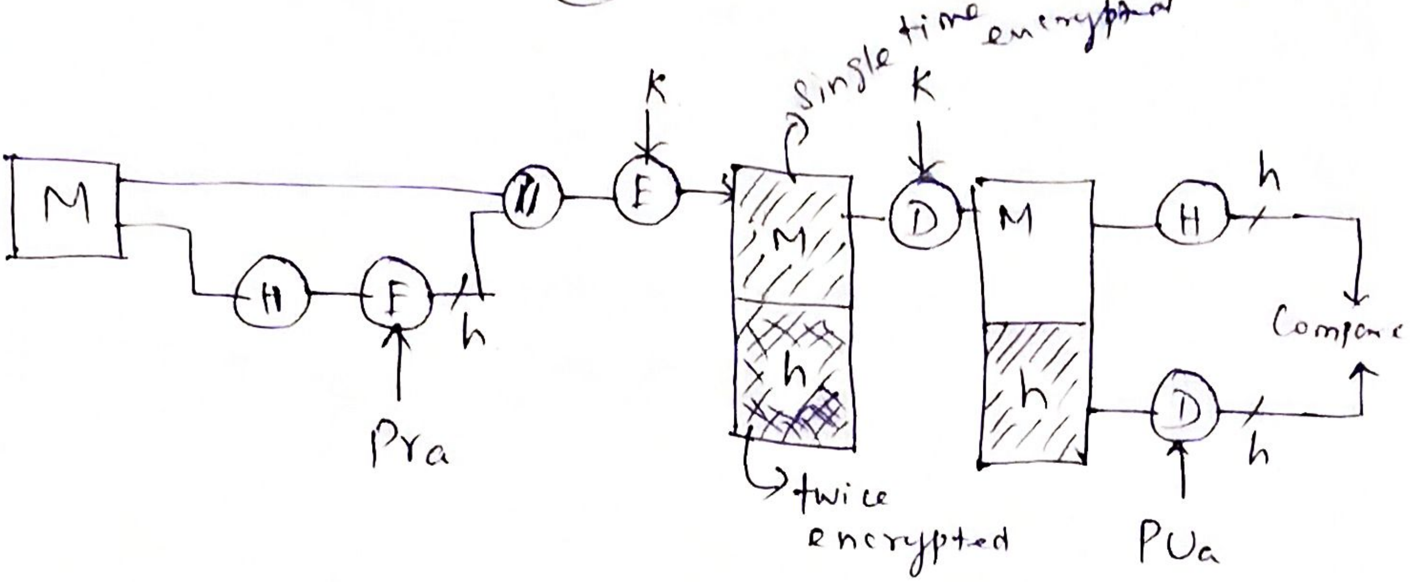


- Here Hash function H does not use a key so we have to Encrypt the (H) and we will get h ie hash code ie encrypted hash code.

- After the Comparision of both h ie hash code, if both are same, then Authentication is achieved.

- In the Receiver side we use Public key of user ~~B~~ A. So any one can decrypt the Message

- We must Provide more Security so that we move to next one



* After appending of Message M and hash code h., again we perform ENC operation

* We are getting Encrypted M and Double time encrypted h.

* DEC the message with same key

* After DEC, we will get message M and one time encrypted h.

* So again we have to perform DEC on public key of A

* At last we compare both hs. If both hs are same the Authentication is achieved