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.

Types of Cryptography
- Symmetric Cryptography (Private Key Cryptography)
- Assymmetric Cryptography (Public Key Cryptography).

- Symmetric Cryptography :- When Encryption and Decryption use same key is called Symmetric Cryptography.

- Asymmetric Cryptography :- In Asymmetric Cryptography we use different keys for encryption as well as decryption. In which one key will be Public and another one will be private.

> Note :- Encryption schemes should be Unconditionally secured (mean attacker will not be able to generate plain text from Cipher text) and Computationally secured.

Key terms in Cryptography
- Plain Text :- It's simple message, text, video etc.
- Cipher Text :- Encrypted text is known as Cipher text.
- Cipher :- It's encryption algorithm.
- Key :- Key is used in Encryption and decryption.
- Cryptanalysis :- Cryptanalysis is used to breach Cryptographic Security Systems and gain access to the content of encrypted message
- Cryptology :- It's Combination of Cryptography and Cryptanalysis.

Classical Encryption techniques :- These are old technique.
(1) Substitution technique
(2) Transposition technique

(1) Substitution technique :- Letters are replaced by other letters or Symbols.

Ex:- R → U, A → D, J → M → Caesar Cipher

(2) Transposition technique :- Applying some sort of permutation on the plaintext letters.

RAJ → AJR, JAR, JRA etc.

Techniques ⇒ Substitution Examples	Transposition Examples
1. Caesar cipher	1. Rail fence
2. Monoalphabetic Cipher	2. Row Column Transposition.
3. Playfair Cipher	
4. Hill Cipher	
5. Polyalphabetic Cipher	
6. One-Time pad	

Caesar Cipher :- Letters are replaced by other letters or Symbols.
- Replacing each letter of the alphabet with the letter standing three places further down the alphabet.
- It was used by Julius Caesar.

Algorithm :- For each plaintext letter 'P', substitute the Ciphertext letter 'C':

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

↓ encryption

For Caesar Cipher $K = 3$

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

↓ decryption

Ex:- RAJ → 17, 0, 9
UDM → 20, 3, 12
→ Applied Caesar Cipher

Caesar cipher also a shift Cipher with Key or K equal to 3.

- Pros :-

(i) Simple and Easy to implement

- Cons :-

(i) The encryption and decryption algorithm is known
(ii) There are only 25 keys to try (vuln. to Brute force attack)

Monoalphabetic Cipher :- The "cipher" line can be any permutation of the 26 Alphabetic characters. In this mapping is done randomly but one will be mapped to one and no other alphabet can't map to already mapped alphabet.

Ex:- NESO → DULA
Plain text → Cipher text

- Pros :-

(i) Better security than Caesar Cipher

- Cons :-

(i) Monoalphabetic ciphers are easy to break because they reflect the frequency data of the original alphanet.

(ii) Prone to guessing attack using the English letter frequency of occurence of letters.
(iii) A countermeasure is to provide multiple substitutes, known as homophones, for a single letter.

Playfair Ciphter :- It is also known as playfair square or wreatstone-playfair Ciphter.
- Manual symmetric encryption technique.
- The first literal digram substitution Ciphter.
- Multiple letter encryption Ciphter.
- Digrams.
- 5×5 matrix constructed using a keyboard (Ex: Monarchy).

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

MY ⇒ NC
RK ⇒ DT
WZ ⇒ XU

Rules :-
1. Digrams.
2. Repeating letters - Filler letter
3. Same Column |↓| Wrap around
4. Same row |→| Wrap around
5. Rectangle |⇆| Swap

Examples of Digrams

	Plain text		Digram
(i)	attack	→	at ta ck
(ii)	mesa academy	→	me sa ac ad em yx
(iii)	balloon	→	ba lx lo on

Examples

Plain text	encrypted text	
1. attack	→ RSSRDE	"monarchy" Key
2. mosque	→ ONTSML	

Hill ciphter :-

- Multi-letter Ciphter
- Encrypts a group of letters : digraphs, trigraphs or polygraphs.

Algorithm :-

This Can be expressed as.

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

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

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

$C_1 = (P_1K_{11} + P_2K_{21} + P_3K_{31}) \bmod 26$

$C_2 = (P_1K_{12} + P_2K_{22} + P_3K_{32}) \bmod 26$

$C_3 = (P_1K_{13} + P_2K_{23} + P_3K_{33}) \bmod 26$

- Question :- Encrypt "Pay more money" using Hill Ciphter with

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

in number

Pay = (15 0 24)

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

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

$(C_1 C_2 C_3) = (303\ 303\ 531) \bmod 26$

$(C_1 C_2 C_3) = (17\ 17\ 11)$

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

This how you Can Encrypt every word and if 3 pair is not making them Put X to make it.

- Decryptic "RRL" using Hill Ciphter.

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

- first finding $K^{-1} \Rightarrow \frac{1}{\text{Det } K} \times \text{Adj } K$

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

$= 17(18\times19 - 2\times21) - 17(19\times21 - 2\times21) + 5(2\times21 - 2\times18) \bmod 26$

$= 5100 - 6069 - 30 \bmod 26$

$= -3 \bmod 26$

$= 23$

* $\text{Adj}\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \bmod 26$

17 17 5 17 17
21 18 21 21 18
2 2 19 2 2

17 17 5 17 17
21 18 21 21 18
2 2 19 2 2
17 17 5 17 17
21 18 21 21 18

~~17~~ 17 5 17 17
21 18 21 21 18
2 2 19 2 2
17 17 5 17 17
21 18 21 21 18

$\Rightarrow$

$18\times19 - 2\times21$	$2\times5 - 17\times19$	$17\times21 - 18\times5$
$21\times2 - 19\times21$	$19\times17 - 5\times2$	$5\times21 - 21\times17$
$21\times2 - 2\times18$	$2\times17 - 17\times2$	$17\times18 - 21\times17$

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

$$\Rightarrow \begin{pmatrix} 14 & -1 & 7 \\ -19 & 1 & -18 \\ 6 & 0 & -25 \end{pmatrix} \bmod 26$$

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

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

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

$$= 17 \times \begin{pmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{pmatrix} \bmod 26 \Rightarrow \begin{pmatrix} 238 & 425 & 119 \\ 119 & 17 & 136 \\ 102 & 0 & 17 \end{pmatrix} \bmod 26$$

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

multiplicative inverse

$23^{-1} \times 23 = 1 \bmod 26$

$\hookrightarrow 17 \times 23 = 1 \bmod 26$

To know it's right you can do this $K \times K^{-1} = 1$

"RRL" = (17 17 11)

$$P = (17\ 17\ 11) \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$$P = (17\times4 + 17\times15 + 11\times24 \quad 17\times9 + 17\times17 + 0 \quad 17\times15 + 17\times6 + 17\times11) \bmod 26$$

$$P = (15\ \ 0\ \ 24) = \text{"Pay"}.$$

Polyalphabetic Substitution Cipher (Vigenere Cipher, Vernam Cipher)

- To improve on the simple monoalphabetic technique.
- General name :- Polyalphabetic substitution Cipher.

Common features.

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

Vigenere Cipher :-

- It consists of the 26 Caesar Ciphers with shifts of 0 through 25.

Encryption Process :-

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

Decryption Process :-

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

Example :-

Key repeats to cover whole plain text.

Key : Raju Raju

Plain text : RajKumar

Key	17	0	9	20	17	0	9	20	
Plain text	17	0	9	10	20	12	0	17	
Cipher text	8	0	18	4	11	12	9	11	mod 26
	I	A	S	E	L	M	J	L	

- Vigenere Cipher - Cryptanalysis

1. Determining the length of the keyword.
2. Key and the plaintext share the same frequency distribution of letters, a statistical technique can be applied.

↓

Like in monoalphabetic Cipher

- Autokey System
(1) The periodic nature of the keyword can be eliminated by using a non-repeating keyword that is as long as the message itself

Example: ← key →
key: R a j u R a j K ←
Plain text: R a j K U M A X

It offer better security

Vernam Cipher:
- Need of ultimate defense against Cryptoanalytic attack
- Length of the keyword = Length of the plaintext
- No statistical relationship to it
- System works on binary bits rather than letter
- System can be expressed as

$$C_i = P_i \oplus K_i$$

(⊕ : XOR)

Cryptographic Stream Generator → K_i

$P_i \rightarrow \oplus \rightarrow C_i$

Encryption

- Vernam Cipher – Cryptanalysis

1. Construction of key
2. The use of a running loop of tape that eventually repeated the key
3. The System worked with a very long but repeating keyword
4. The technique can be broken with sufficient Ciphertext the use of known or probable plaintext Sequences or both.
5. Decryption can be done like this, $P_i = C_i \oplus K_i$ (⊕ : XOR)

One Time Pad :

- Improvement to the Vernam Cipher
- It yields the ultimate in Security.
- Random key that is as long as the message.
- The key need not be repeated.
- In addition, the key is to be used to encrypt and decrypt a single message and then is discarded.
- Each new message requires a new key of the same length as the new message
- Such a Scheme, known as a one-time pad, is unbreakable.
- It produces random output.
- No statistical relationship to the plaintext.
- Because Ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the Code.
- The Code is Unbreakable.
- The Security is due to the randomness of the key.
- It's Only System referred to as Perfect Secrecy.

★ Two fundamental difficulties of One Time pad :-

(1) The practical problem of making large quantities of random keys.

(2) Even more daunting is the problem of key distribution and protection.

(3) Because of these difficulties, the One-time pad is of limited utility and is useful primarily for low-bandwidth Channels requiring very High Security.

* Perfect Secrecy :- When encrypted message from a perfectly secure encryption system, absolutely nothing will be revealed about the unencrypted message by the Ciphertext.

Transposition Cipher :-
- Some sort of permutation applied on the plaintext letters.
- This technique is referred to as a transposition Cipher.
- The simplest such Cipher is the rail fence.

Rail Fence Technique :- The plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.

Example :- Encipher "Rajkumar" with a rail fence of depth 2.

Plain text : Rajkumar

Depth : 2

R		J		U		A	
	A		K		M		R

Cipher text :- RJUAAKMR

Row Column Transposition :-
1. A more complex scheme
2. Rectangle
3. Write : Row by row
4. Read : Column by Column.
5. Key : Order of the Column.

Example :- "Raj Kumar"

Plain text :- "Raj Kumar"

Key ⇒ 3 1 4 2 ——→ This Our key

R	a	j	K
u	m	a	r

Ciphertext :- "AKRJMRUA"

NOTE ⇒ To make strong ciphertext this process can be repeated many time which give Ciphter text which is hard to break.

Prime Numbers : A prime number is a number greater than 1 with only two factors - itself and one. It can not be divided further by any other numbers without leaving a number.
Example :- 2, 3, 5 and 7.

- All numbers have prime factors

Numbers	Prime factorization
10	$2^1 \times 5^1$
11	$1^1 \times 11^1$
100	$2^2 \times 5^2$
37	$1^1 \times 37^1$

- Facts about primes
1. Only even prime : 2
2. Smallest prime number : 2
3. Except for 2 and 5, all prime numbers end in the digit 1, 3, 7 or 9.

- Why prime numbers in Cryptography?
1. Many encryption algorithm are based on prime numbers.
2. Very fast to multiply two large prime numbers.
3. Extremely Computer-intensive to do the reverse.
4. Factoring very large prime numbers is very hard i.e take a Computers a long time.

Composite Numbers :- Numbers which are not prime number are Called Composite numbers or numbers which have more than 2 factors.
Ex :- 33 is Composite numbers.

Modular Arithmetic :-

- System of arithmetic for integers
- wrap around after reaching a certain value called modulus.
- Central mathematical concept in Cryptography.

* Congruence :-

In Cryptography, Congruence (≡) instead of equality (=).

Examples:

$15 \equiv 3 \pmod{12}$

$23 \equiv 11 \pmod{12}$

$33 \equiv 3 \pmod{10}$

$10 \equiv -2 \pmod{12}$

(**) $35 \equiv 25 \pmod{10} \equiv 5 \pmod{10}$

$\therefore a \equiv b \pmod{m}$

$a = Km + b$

m	a
	b

K

* Properties of Modular Arithmetic

1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a+b) \bmod n$
2. $[(a \bmod n) - (b \bmod n)] \bmod n = (a-b) \bmod n$
3. $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

* Properties of Modular Arithmetic

	Property	Expression
1.	Commutative Laws	$(a+b) \bmod n = (b+a) \bmod n$ $(a \times b) \bmod n = (b \times a) \bmod n$
2.	Associative Laws	$[(a+b)+c] \bmod n = [a+(b+c)] \bmod n$ $[(a \times b) \times c] \bmod n = [a \times (b \times c)] \bmod n$
3.	Distributive Laws	$[a \times (b+c)] \bmod n = [(a \times b) + (a \times c)] \bmod n$
4.	Identities	$(0+a) \bmod n = a \bmod n$ $(1 \times a) \bmod n = a \bmod n$
5.	Additive Inverse	For each $a \in Z_n$, there exists a '-a' such that $a + (-a) \equiv 0 \bmod n$.

Modular Exponentiation:-

- It is a type of exponentiation performed over a modulus.
- $a^b \bmod n$ or $a^b \pmod m$.

Examples:

(i) $11^3 \bmod 13$

$= 11 \bmod 13 \times 11 \bmod 13 \times 11 \bmod 13$

$= -2 \bmod 13 \times -2 \bmod 13 \times -2 \bmod 13$

$= (-2)^3 \bmod 13$

$= -8 \bmod 13$

$= 5 \bmod 13$

(ii) $31^{5000} \bmod 30$.

$= 1^{5000} \bmod 30$

$= 1 \bmod 30$.

(iii) $88^7 \bmod 187$

$88^1 \bmod 187 = 88$

$88^2 \bmod 187 = 88^1 \times 88^1 \bmod 187 = 88 \times 88 \bmod 187 = 77$

$88^4 \bmod 187 = 88^2 \times 88^2 \bmod 187 = 77 \times 77 \bmod 187 = 132$

$88^7 \bmod 187 = 88^4 \times 88^2 \times 88^1 \bmod 187 = (132 \times 77 \times 88) \bmod 187$

$= 894,432 \bmod 187 = 11$

(iv) What is "the last two digits" of 29^5?

$29^1 \bmod 100 = 29$ or -71

$29^2 \bmod 100 = 29^1 \times 29^1 \bmod 100 = 29 \times 29 = 841 \bmod 100 = 41$ or -59

$29^4 \bmod 100 = 29^2 \times 29^2 \bmod 100 = 41 \times 41 = 1681 \bmod 100$

$= 81$ or -19

$29^5 \bmod 100 = 29^4 \times 29^1 \bmod 100$

$= -19 \times 29 \bmod 100$

$= -551 \bmod 100$

$= -51 \bmod 100$

$= 49$

$29^5 \bmod 100 = 49$

* Always take smaller

Last two digits means mod 100 because remainder will be under 100 and it will last two digit numbers.

Euclidean algorithm :

- Euclidean or Euclid's Algorithm.
- For Computing GCD (Greatest Common Divisor).
- aka HCF (Highest Common factor).

Example :

	25	150
Divisors	1, 5, 25	1, 2, 5, 3, 6, 10, 15, 25, 30, 50, 75, 100
Common Divisors	1, 5, 25	
GCD	25	

GCD(25, 150) = 25 → 27th On Composite numbers.

	13	31
Divisors	1, 13	1, 31
Common Divisors	1	
GCD	1	

GCD(13, 31) = 1 → 27th On prime numbers.

If numbers are prime then ans will be 1 only.

* Using Euclid's Algorithm :

GCD(12, 33) = 3

Condition : A > B Always

B	A	B	R
2	33	12	9
1	12	9	3
3	9	3	0
X	3	0	X

Answer ← 3

0 → Stop when B have zero

Code :-

Prerequisite : $a > b$

```
Euclid_GCD(a,b):
    if(b=0) then
        return a;
    else
        return Euclid_GCD(b, a mod b);
```

Relatively Prime (Co-prime) Numbers :- Two numbers are said to be relatively prime, if they have no prime factors in common, and their only common factor is 1 or GCD is 1.

- GCD(a,b) = 1 then a and b are relatively prime numbers.

Example :- GCD(13, 31) = 1, Yes it's Relatively primes

Eulex's Totient function :-

- Denoted as $\phi(n)$.
- $\phi(n)$ = Number of positive integers less than 'n' that are relatively prime to n.

Example :- i) find $\phi(5)$.

Here $n = 5$

Numbers less than 5 are 1, 2, 3 and 4.

GCD	Relatively Prime
GCD(1,5) = 1	Yes
GCD(2,5) = 1	Yes
GCD(3,5) = 1	Yes
GCD(4,5) = 1	Yes

$\therefore \phi(5) = 4.$

* Formulas to find Euler's Tutient ($\phi(n)$).

	Criteria of 'n'	Formula
$\phi(n)$	'n' is prime	$\phi(n) = (n-1)$
	$n = P \times q$. 'P' and 'q' are prime	$\phi(n) = (P-1) \times (q-1)$.
	$n = a \times b$ Either 'a' or 'b' is composite	$\phi(n) = n \times \left(1 - \frac{1}{P_1}\right)\left(1 - \frac{1}{P_2}\right) \cdots$
	Both 'a' and 'b' are Composite	Where $P_1, P_2, \ldots$ are distinct Primes.

Examples :-

(i) Find $\phi(5)$

5 is prime-numbers

$\phi(5) = (n-1)$

$= (5-1)$

$= 4$

So, there are 4 numbers that are lesser than 5 and relatively prime to 5.

(ii) Find $\phi(35)$

Here $n = 35$

'n' is a product of two prime numbers 5 and 7.

Let $P = 5$ and $q = 7$

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

$= (5-1) \times (7-1)$

$= 4 \times 6$

$\phi(35) = 24$

So, there are 24 numbers that are lesser than 35 and relatively prime to 35.

(iii) Find $\phi(1000)$.

Here $n = 1000 = 2^3 \times 5^3$.

Distinct prime factors are 2 and 5.

$\phi(n) = n \times \left(1 - \frac{1}{p_1}\right) \times \left(1 - \frac{1}{p_2}\right) \cdots$

$\phi(1000) = 1000 \times \left(1 - \frac{1}{2}\right) \times \left(1 - \frac{1}{5}\right)$

$\phi(1000) = 1000 \times \frac{1}{2} \times \frac{4}{5}$

$\phi(1000) = 400$.

(iv) Find $\phi(7000)$

Here $n = 7000 = 2^3 \times 5^3 \times 7^1$.

Distinct prime factors are 2, 5 and 7.

$\phi(n) = n \times \left(1 - \frac{1}{p_1}\right) \times \left(1 - \frac{1}{p_2}\right) \times \left(1 - \frac{1}{p_3}\right) \cdots$

$\phi(7000) = 7000 \times \left(1 - \frac{1}{2}\right) \times \left(1 - \frac{1}{5}\right) \times \left(1 - \frac{1}{7}\right)$

$\phi(7000) = 7000 \times \frac{1}{2} \times \frac{4}{5} \times \frac{6}{7}$

$\phi(7000) = 2400$

Fermat's Little Theorem:

- If 'p' is a prime number and 'a' is a positive integer not divisible by 'p' then $a^{p-1} \equiv 1 \pmod{p}$.

Example:

(i) Does fermat's theorem hold true for $p = 5$ and $a = 2$?

Given: $p = 5$ and $a = 2$

$a^{p-1} \equiv 1 \pmod{p} \Rightarrow 2^{5-1} \equiv 1 \pmod{5}$.

$2^4 \equiv 1 \pmod{5}$

$16 \equiv 1 \pmod{5}$

Yes, it holds true for $p = 5$ and $a = 2$.

Euler's theorem :

- For every positive integer 'a' and 'n', which are said to be relatively prime then $a^{\phi(n)} \equiv 1 \bmod n$.

Example :

(i) Prove Euler's theorem hold true for $a = 3$ and $n = 10$.

Given : $a = 3$ and $n = 10$.

$$a^{\phi(n)} \equiv 1 \pmod{n}$$
$$3^{\phi(10)} \equiv 1 \pmod{10}$$
$$\phi(10) = 4$$
$$3^4 \equiv 1 \pmod{10}$$
$$81 \equiv 1 \pmod{10}$$

(ii) Does Euler's theorem hold true for $a = 2$ and $n = 10$?

Given : $a = 2$ and $n = 10$.

$$a^{\phi(n)} \equiv 1 \pmod{n}$$
$$2^{\phi(10)} \equiv 1 \pmod{10}$$
$$\phi(10) = 4$$
$$2^4 \equiv 1 \pmod{10}$$
$$16 \not\equiv 1 \pmod{10}$$

* Because a and n are not relative prime that's why they not hold Euler's theorem.

NOTE : a and n should be relatively prime

Primitive Root :-

- 'x' is said to be a primitive root of prime number 'p', if $x^1 \bmod p$, $x^2 \bmod p$, $x^3 \bmod p$, ..., $x^{p-1} \bmod p$ are distinct.

Example :-

ii) Is 2 a primitive root of prime number 5?

Sol:

$2^1 \bmod 5$	$2 \bmod 5$	2	Yes
$2^2 \bmod 5$	$4 \bmod 5$	4	Yes
$2^3 \bmod 5$	$8 \bmod 5$	3	Yes
$2^4 \bmod 5$	$16 \bmod 5$	1	Yes

Yes, 2 is a primitive root of prime number 5.

iii) Is 3 a primitive root of prime number 7?

Sol:

$3^1 \bmod 7$	$3 \bmod 7$	3	Yes
$3^2 \bmod 7$	$9 \bmod 7$	2	Yes
$3^3 \bmod 7$	$6 \bmod 7$	6	Yes
$3^4 \bmod 7$	$18 \bmod 7$	4	Yes
$3^5 \bmod 7$	$12 \bmod 7$	5	Yes
$3^6 \bmod 7$	$15 \bmod 7$	1	Yes

- A number 'x' is a primitive root modulo n if every Coprime to n is Congruent to a power of 'x' modulo n.

Multiplicative Inverse :-

* For simple integers.

$5 \times 5^{-1} = 1$

$5 \times \frac{1}{5} = 1$

$A \times A^{-1} = 1$

A^{-1} is multiplicative inverse of A.

* Under mod n

$A \times A^{-1} = 1 \bmod n$

Example :-

(i) $3 \times ? \equiv 1 \bmod 5$

Ans: is 2 because product will be 6 and 6 is divided by 5 will give 1 as reminder.

(ii) $2 \times ? \equiv 1 \bmod 11$

Ans: is 6

(iii) $4 \times ? \equiv 1 \bmod 5$

4 is Ans

(iv) $5 \times ? \equiv 1 \bmod 10$

We don't have multiplicative inverse because 5 and 10 are not co-prime

> NOTE: Numbers should be co-prime like 2 and 11

(v) The M.I for 2(mod 5) is 3.

(vi) The M.I for 2(mod 7) is 4.

Extended Euclidean algorithm (for finding multiplicative inverse):-

Algo:- For finding multiplicative inverse of Two number a and b, and a and b both should be co-prime numbers.

Q	A	B	R	T_1	T_2	T

In starting $T_1 = 0$ and $T_2 = 1$

$$T = T_1 - T_2 \times Q$$

Example:-

(i) What is the multiplicative inverse of 3 mod 5.

Ans:- 3 and 5 are co-prime numbers.

Q	A	B	R	T_1	T_2	T
1	5	3	2	0	1	-1
1	3	2	1	1	-1	2
2	2	1	0	-1	2	-5
X	1	0	X	2	-5	X

Answer is 2

Stop when B = 0 and Answer will be in T_1

(ii) What is the multiplicative inverse of 11 mod 13.

Ans:- 11 and 13 are co-prime numbers.

Q	A	B	R	T_1	T_2	T
1	13	11	2	0	1	-1
5	11	2	1	1	-2	6
2	2	1	0	-2	6	-13
X	1	0	X	6	-13	X

∴ 6 is M·I of 11 mod 13.

(ii) Find the M·I of 11 mod 26.

Q	A	B	R	T_1	T_2	T
2	26	11	4	0	1	-2
2	11	4	3	1	-2	5
1	4	3	1	-2	5	-7
3	3	1	0	5	-7	26
X	1	0	X	-7 or 19	26	X

↳ keep in mind

Ans:- ∴ 19 is the M·I of 11 mod 26.

The Chinese Remainder Theorem :-

- The Chinese Remainder Theorem (CRT) is used to solve a set of different congruent equations with one variable but different moduli which are relatively prime as shown below:

$$X \equiv a_1 \pmod{m_1}$$
$$X \equiv a_2 \pmod{m_2}$$
$$\vdots \quad \vdots \quad \vdots$$
$$X \equiv a_n \pmod{m_n}$$

CRT states that the above equations have a unique solution of the moduli are relatively prime.

$X = (a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} + \dots + a_n M_n M_n^{-1}) \bmod M$

Example:-

i) Solve the following equations using CRT.

$X \equiv 2 \pmod 3$

$X \equiv 3 \pmod 5$

$X \equiv 2 \pmod 7$

NOTE: 3, 5, 7 are Co-prime Numbers → Should be Co-primes.

Solution:-

$X = (a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} + a_3 M_3 M_3^{-1}) \bmod M.$

$X \equiv a_1 \pmod{m_1}$	$X \equiv 2 \pmod 3$
$X \equiv a_2 \pmod{m_2}$	$X \equiv 3 \pmod 5$
$X \equiv a_3 \pmod{m_3}$	$X \equiv 2 \pmod 7$

Given		To find		
$a_1 = 2$	$m_1 = 3$	$M_1 = 35$	$M_1^{-1} = 2$	$M = 105$
$a_2 = 3$	$m_2 = 5$	$M_2 = 21$	$M_2^{-1} = 1$	
$a_3 = 2$	$m_3 = 7$	$M_3 = 15$	$M_3^{-1} = 1$	

$M = m_1 \times m_2 \times m_3$

$M = 3 \times 5 \times 7 = 105$

$M_1 = 105/3 = 35$

$M_2 = 105/5 = 21$

$M_3 = 105/7 = 15$

$M_n = \frac{M}{m_n}$

$M_n \times M_n^{-1} = 1 \bmod m_n$

$M_1 \times M_1^{-1} = 1 \bmod m_1$	$M_2 \times M_2^{-1} = 1 \bmod m_2$	$M_3 \times M_3^{-1} = 1 \bmod m_3$
$35 \times M_1^{-1} = 1 \bmod 3$	$21 \times M_2^{-1} = 1 \bmod 5$	$15 \times M_3^{-1} = 1 \bmod 7$
$M_1^{-1} = 2$	$M_2^{-1} = 1$	$M_3^{-1} = 1$

$X = (a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} + a_3 M_3 M_3^{-1}) \bmod M$

$= (2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1) \bmod 105$

$= 233 \bmod 105$

$X = 23$

Ans: $X = 23$

(ii) Solve the following equation with CRT:

$4x \equiv 5 \pmod 9$

$2x \equiv 6 \pmod{20}$

- $4x \equiv 5 \pmod 9$

  $x \equiv 4^{-1} \bmod 9 \times 5 \bmod 9$

  $4^{-1} \Rightarrow 7$ because $4 \times 7 \bmod 9 = 1 \bmod 9$

  $x \equiv 7 \times 5 \bmod 9$

  $x \equiv 35 \bmod 9$

  $x \equiv 8 \pmod 9$
- $2x \equiv 6 \pmod{20}$

  $x \equiv 3 \pmod{20}$

$x = (a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1}) \bmod M$

$M = m_1 \times m_2$

$= 9 \times 20 = 180$

$M_1 = M/m_1 = 180/9 = 20$

$M_2 = M/m_2 = 180/20 = 9$

$M_1 \times M_1^{-1} = 1 \bmod m_1$

$20 \times M_1^{-1} = 1 \bmod 9$

$M_1^{-1} = 5$

$M_2 \times M_2^{-1} = 1 \bmod m_2$

$9 \times M_2^{-1} = 1 \bmod 20$

$M_2^{-1} = 9$

$x = (a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1}) \bmod M$

$= (8 \times 20 \times 5 + 3 \times 9 \times 9) \bmod 180$

$= (800 + 243) \bmod 180$

$= 1043 \bmod 180$

$x = 143$

The Discrete Logarithm Problem :-

Understanding :-

$5^x \bmod 17 =$ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Equally Distributed

→ 5 is primitive root of 17

Finding in the way is easy

If Ans is given then finding X is hard, think prime no. is Large

- For smaller value of 'p' is may be easy to find x.
- If 'p' is very Large, then finding X is hard.

Example :-

(i) Solve $\log_2 9 \bmod 11$?

Solution :-

Here $P = 11$, $g = 2$, $X = 9$

$$\log_g X \equiv n \pmod P$$

$$X \equiv g^n \pmod P$$

$$9 \equiv 2^n \pmod{11}$$

Try 'n' = 1, 2, 3 ....

Ans : 6 is n.

$$9 \equiv 2^6 \pmod{11}$$

$\therefore n = 6$

Factoring – Fermat's Algorithm:

- To factor 'n'.
- $n = x \cdot y$ ~~where x and y are not prime numbers.~~
- Works well when x and y are close.

Formula:

$n = x^2 - y^2$

$x^2 = n + y^2$

$x = \sqrt{n + y^2}$

Try different values for 'y' from 1 to n.

Example:

i) Factor $n = 187$.

$x = \sqrt{n + y^2}$

$x = \sqrt{187 + y^2}$

$x = \sqrt{187 + 1^2} = \sqrt{188} \neq$ integer

$x = \sqrt{187 + 2^2} = \sqrt{191} \neq$ integer

$x = \sqrt{187 + 3^2} = \sqrt{196} = 14$

~~Ans: $x = 14, y = 3$~~

$n = x^2 - y^2$

$n = (x + y)(x - y)$

$n = (14 + 3)(14 - 3)$

$n = 17 \times 11$

$187 = 17 \times 11$

The prime factors of 187 are 17 and 11.

Fermat's Primality Test :-

Is 'p' Prime :-

$a^p - a \rightarrow$ 'p' is prime if this is a multiple of 'p' for all $1 \leq a < p$.

Example :-

(i) Is 5 prime?

Soln :-

$1^5 - 1 = 1 - 1 = 0$

$2^5 - 2 = 32 - 2 = 30$

$3^5 - 3 = 243 - 3 = 240$

$4^5 - 4 = 1024 - 4 = 1020$

∴ 5 is prime

Drawback :-

(i) It's not easy to find when number is Large.

(ii) It gives wrong answer like for 561 it say it's prime number but its not.

(iii) Less accuracy.

Miller-Rabin primality Test :-

- Probabilistic Primality test.

Algorithm :- (i) Find $n-1 = 2^k \times m$

(ii) Choose 'a' such that $1 < a < n-1$

(iii) $b_0 = a^m \pmod n, \ldots, b_i = b_{i-1}^2 \pmod n$

+1 = Composite

-1 = Prime Number

* Start from $b_0 \ldots$ to b_i untill to find +1 or -1.

Example:-

i) Is 561 prime.

Given $n = 561$

$\frac{560}{2^1} = 280$	$\frac{560}{2^2} = 140$	$\frac{560}{2^3} = 70$	$\frac{560}{2^4} = 35$	$\frac{560}{2^5} = 17.5$

Till find point

Step 1:

$n - 1 = 2^k \times m$

$560 = 2^4 \times 35$

So $k = 4$, and $m = 35$

Step 2:

Choosing $a = 2$; $1 < 2 < 560$

Step 3:

Compute $b_0 = a^m \pmod n$

$b_0 = a^m \pmod n$

$b_0 = 2^{35} \pmod{561} = 263$

is $b_0 = \pm 1 \pmod{561}$? NO

So Calculate b_1

$b_1 = b_0^2 \pmod n$

$b_1 = 263^2 \pmod{561}$

$b_1 = 166$

is $b_1 = \pm 1 \pmod{561}$? NO

So Calculate b_2

$b_2 = 166^2 \pmod{561}$

$b_2 = 67$

Is $b_2 = \pm 1 \pmod{561}$? NO

~~$b_3 = b_2^2$~~

So Calculate b_3

$b_3 = 67^2 \pmod{561}$

$b_3 = 1 \rightarrow$ Composite Number

∴ 561, because b_3 is $+1$.

↓

is Composite

Group :-

- A group Gr denoted by $\{Gr, \circ\}$ is a set under some operation ($\circ$) if it satisfies the GAIN properties.

G - Closure
A - Associative
I - Identity
N - Inverse

Abelian Group :-

A group is said to be Abelian if it already a group and Commutative property is also satisfied i.e $(a \circ b) = (b \circ a)$ for all a, b in Gr

		Property	Explanation
Abelian group	Group	Closure	a, b $\in$ Gr, then $(a \circ b) \in Gr$.
		Associative	$a \circ (b \circ c) = (a \circ b) \circ c$ for all a, b, c $\in$ Gr
		Identity element	$(a \circ e) = (e \circ a) = a$ for all a, e $\in$ Gr.
		Inverse element	$(a \circ a') = (a' \circ a) = e$ for all a, a' $\in$ Gr
		Commutative	$(a \circ b) = (b \circ a)$ for all a, b $\in$ Gr

Example :- Question: Is $(Z, +)$ a abelian group.

Answer :- Closure : if $a = 5$, $b = -2 \in Z$ then $a + b = -3 \in Z$ It's Closure

Associative : $5 + (3 + 7) = (5 + 3) + 7 \in Z$, Yes

Identity : $(5 + 0) = (0 + 5) = 5$ for all $a \in Z$ Yes

Inverse : $(5 + (-5)) = (-5 + 5) = 0$ for all $a', a \in Z$, yes

Commutative : $(5 + 9) = (9 + 5)$ for all $9, 5 \in Z$, Yes

It's Closure, Associative, Identity, Inverse and Commutative that it's Abelian group.

Notations :

N → Set of all natural numbers.
W → Set of all whole numbers
Z → Set of all integers.
C → Set of all Complex numbers.
Q → Set of all rational numbers.
R → Set of all real numbers.
Z^+ → Set of all positive integers.
Z^- → Set of all negative integers.

Cyclic Group :-

- A Group G denoted by $\{G, \circ\}$ is said to be a Cyclic Group, if it contains at-least one Generator element.

Example :-

(i) Prove that $(G, *)$ is a Cyclic Group, where $G = \{1, \omega, \omega^2\}$

Because operation is multiplication

Composition table

*	1	ω	ω^2
1	1	ω	ω^2
ω	ω	ω^2	1
ω^2	ω^2	1	ω

$1^1 = 1$	$\omega^1 = \omega$	$(\omega^2)^1 = \omega^2$
$1^2 = 1$	$\omega^2 = \omega^2$	$(\omega^2)^2 = \omega$
$1^3 = 1$	$\omega^3 = 1$	$(\omega^2)^3 = 1$
$1^4 = 1$	$\omega^4 = \omega$	$(\omega^2)^4 = \omega^2$
Not a Generator	Generator	Generator

The Generators of $(G, *)$ are ω and ω^2.
∴ $(G, *)$ is a Cyclic Group.

(ii) When does group Gr with operation '*' is said to be a Cyclic group?

Solution:

Let us take an element x

$Gr = \{\ldots, x^{-4}, x^{-3}, x^{-2}, x^{-1}, 1, x, x^2, x^3, x^4, \ldots\}$

= group Generated by x

If $Gr = \langle x \rangle$ for some x, then we Call Gr a Cyclic group.

(iii) When does group Gr with operation '+', is said to be a Cyclic group?

Solution:

Let us take an element y

$Gr = \{\ldots, -4y, -3y, -2y, -y, 0, y, 2y, 3y, 4y, \ldots\}$

= Group Generated by y

If $Gr = \langle y \rangle$ for some y, then we Call Gr a Cyclic group.

Rings:

- A ring R denoted by $\{R, +, *\}$ is a set of elements with two binary operations, called addition and multiplication such that for all $a, b, c \in R$ the following axioms are obeyed:
  - * Group (A1 - A4), Abelian Group (A5).
  - * Closure under multiplication (M1): If $a, b \in R$ then $ab \in R$.
  - * Associativity under multiplication (M2): $a(bc) = (ab)c$ for all $a, b, c \in R$
  - * Distributive laws (M3):

    $a(b+c) = ab + ac$ for all $a, b, c \in R$

    $(a+b)c = ac + bc$ for all $a, b, c \in R$

A for mean for Addition operation

M for mean for multiplication operation.

Note: Substraction $[a - b = a + (-b)]$

Commutative Rings :-

- A ring is said to be commutative, if it satisfies the following additional condition:
  Commutativity of multiplication (M4): $ab = ba$ for all $a, b \in R$

Integral Domain :-

- An integral domain is a commutative ring that obeys the following axioms:
  Multiplicative identity (M5): There is an element $1 \in R$ such that $a1 = 1a = a$ for all $a \in R$.
  No Zero divisor (M6): if $a, b \in R$ and $ab = 0$, then either $a = 0$ or $b = 0$.

Fields :-

- A field F, sometimes denoted by $\{F, +, *\}$ is a set of elements with two binary operations, called addition and multiplication, such that for all $a, b, c \in F$ the following axioms are obeyed:
  (A1 - M6): F satisfies axioms A1 - A5 and M1 - M6.
  (M7) Multiplicative inverse: For each a in F, except 0, there is an element a^{-1} in F such that $aa^{-1} = a^{-1}a = 1$

$$aa^{-1} = a^{-1}a = 1$$

NOTE: $a/b = a(b^{-1})$

Examples :-

(i) Rational numbers
(ii) Real numbers
(iii) Complex numbers

Finite Fields :-

- A finite field or Galois field is a field that contains a finite number of elements.
- As with any field, a finite field is a set on which the operations of multiplication, addition, subtraction and division are defined and satisfy certain basic rules (All axioms should be satisfied).
- The most common examples of finite fields are given by the integers (mod p) when p is a Prime numbers.

* Applications :-

1. Cryptography and Coding theory.
2. Number theory, Algebraic geometry, Galois theory and finite geometry.

Confusion :-

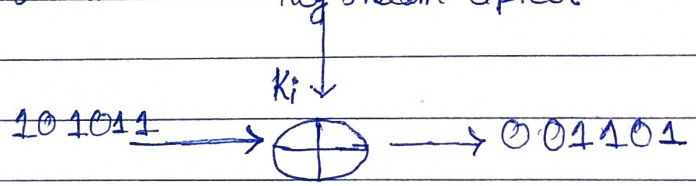
- Making the relationship between the encryption key and the ciphertext as complex as possible.
- Relationship between CT and PT is obscured.
- Given CT, no info about PT, key, Encryption algorithm etc.
- Example : Substitution.

Diffusion :-

- Making each plaintext bit affect as many cipher text bits as possible.
- 1 bit change in PT, significant effect on CT.
- Example : Transposition or permutation.

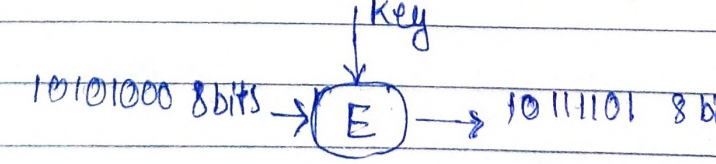
Stream cipher :-

- Each plaintext digit is encrypted one at a time with the corresponding digit of the keystream, to give a digit of the ciphertext stream.



Block Ciphers :-

- A block cipher is a deterministic algorithm operating on fixed-length groups of bits, called blocks

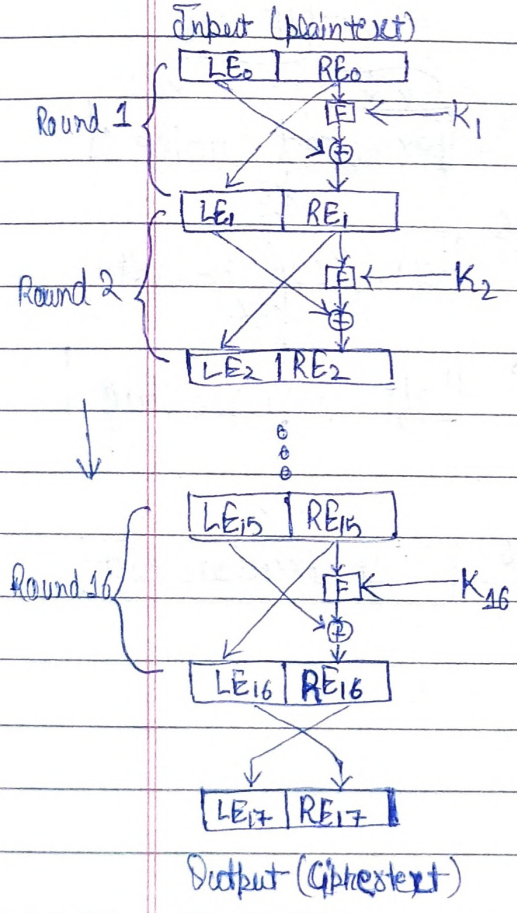


- Size of block decides by encryption algorithm.

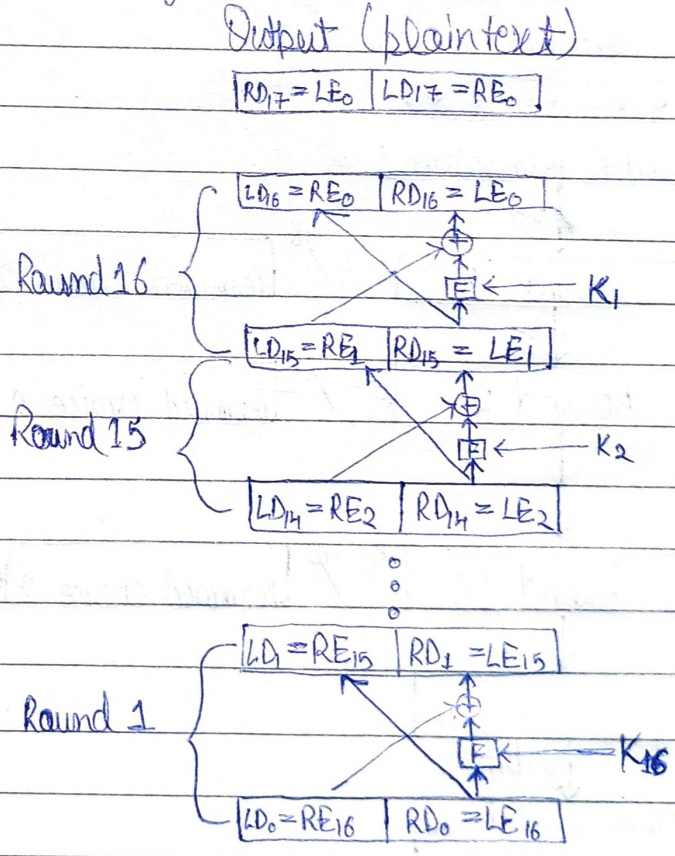
Property	Stream Ciphers	Block Cipher
Length	Bit or Bytes	Block Size - 64 bits, 128 Bits
Design	Complex	Simple
Principle	Confusion	Confusion and diffusion
Speed	Faster	Slower
Encryption Mode	CFB (Cipher Feedback) and OFB (Output feedback).	Electronic Code book (ECB) and Cipher Block Chaining (CBC)
Decryption	XOR	Reverse of encryption
Example	Vernam Cipher	DES, AES.

Feistel structure :-

Encryption:



Decryption:



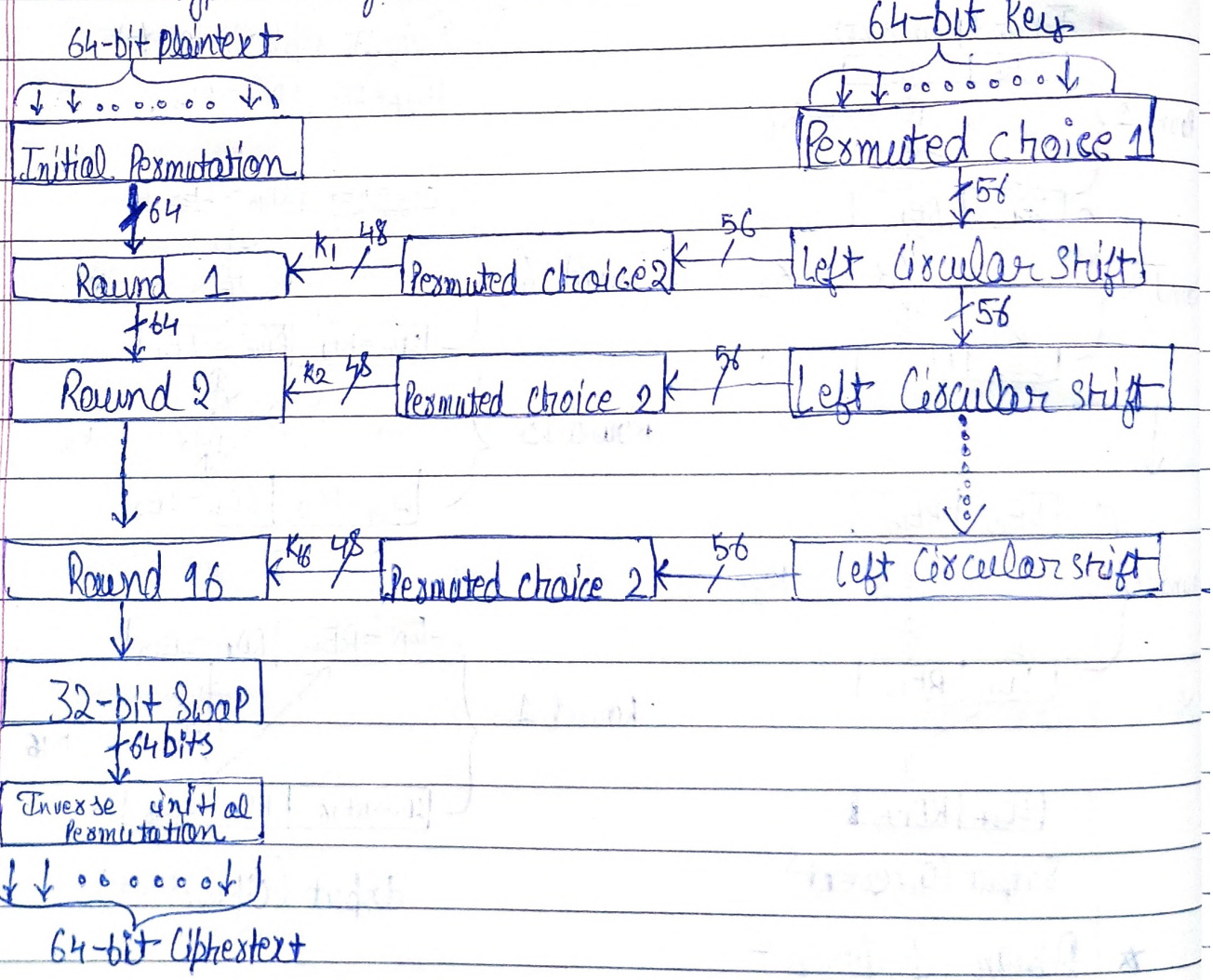
Design features :-

1. Block Size
2. Subkey generation algorithm
3. Fast Encryption/decryption
4. Number of rounds
5. Key Size
6. Round function
7. Ease of Analysis

Data Encryption Standard :

- Symmetric Block Ciphers
- A.K.a Data Encryption Algorithm
- Adopted by NIST in 1977.
- Advanced Encryption Standard (AES) in 2001.
- Input : 64 bits
- Output : 64 bits
- Main Key : 64 bits.
- Sub Key : 56 bits.
- Round Key : 48 bits
- No. of rounds : 16 rounds.

* DES Encryption Algorithm.



★ Initial permutation :-

Used to change A to A'

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

★ Inverse Initial permutation :- A → A'

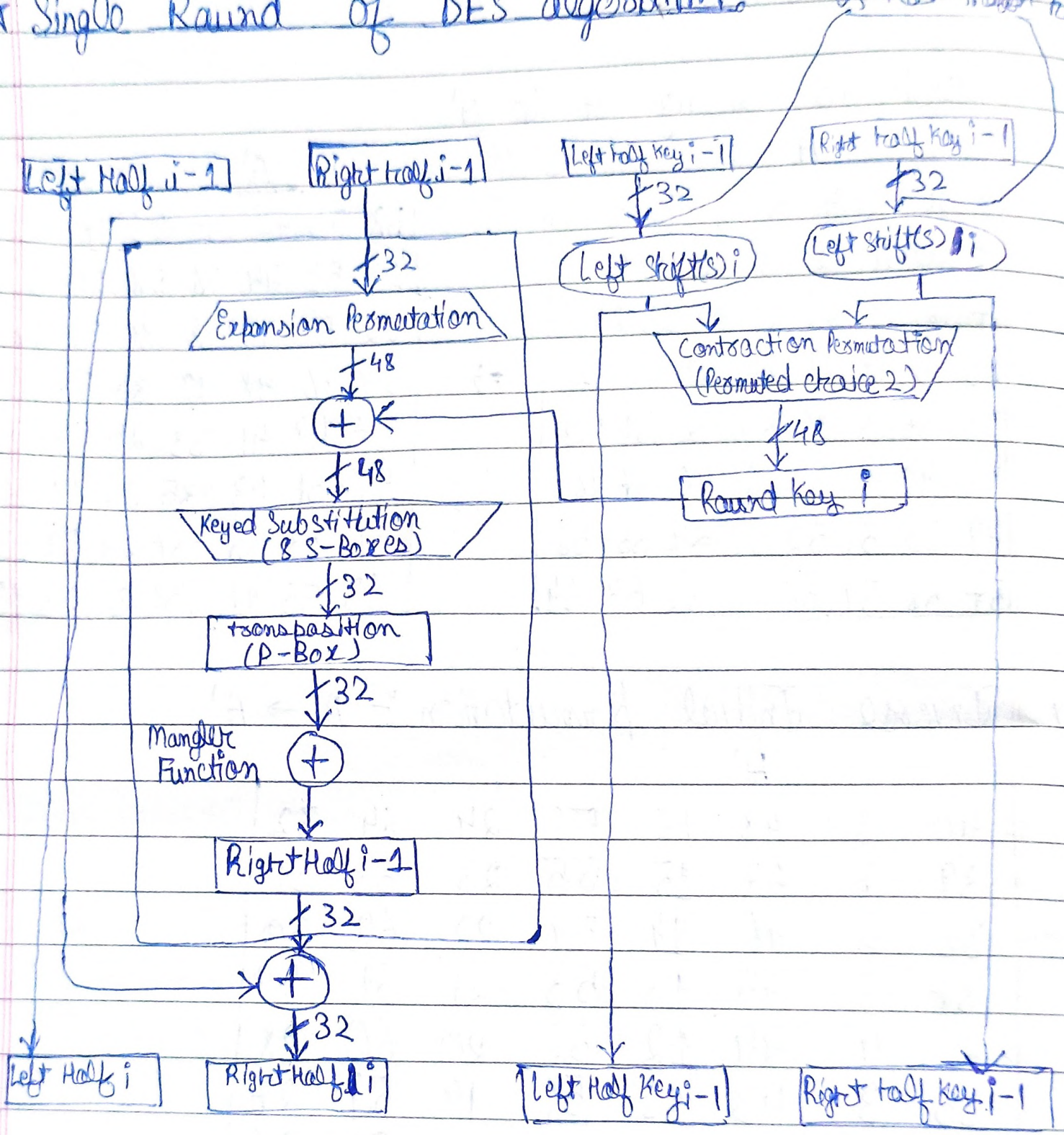
A'							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

★ Permuted choice 1 :- [+] Bits 8, 16, 24 ...
[+] effective Key length is 56 Bits

★ Left Circular shift :- [+] LS_i = 1 shift for i = 1, 2, 9, 16
[+] LS_i = 2 shift for i = after rounds

★ Permuted choice 2 :- [+] 8 bits are dropped like choice 1.
[+] 48 bit are ~~permuted~~ permuted.

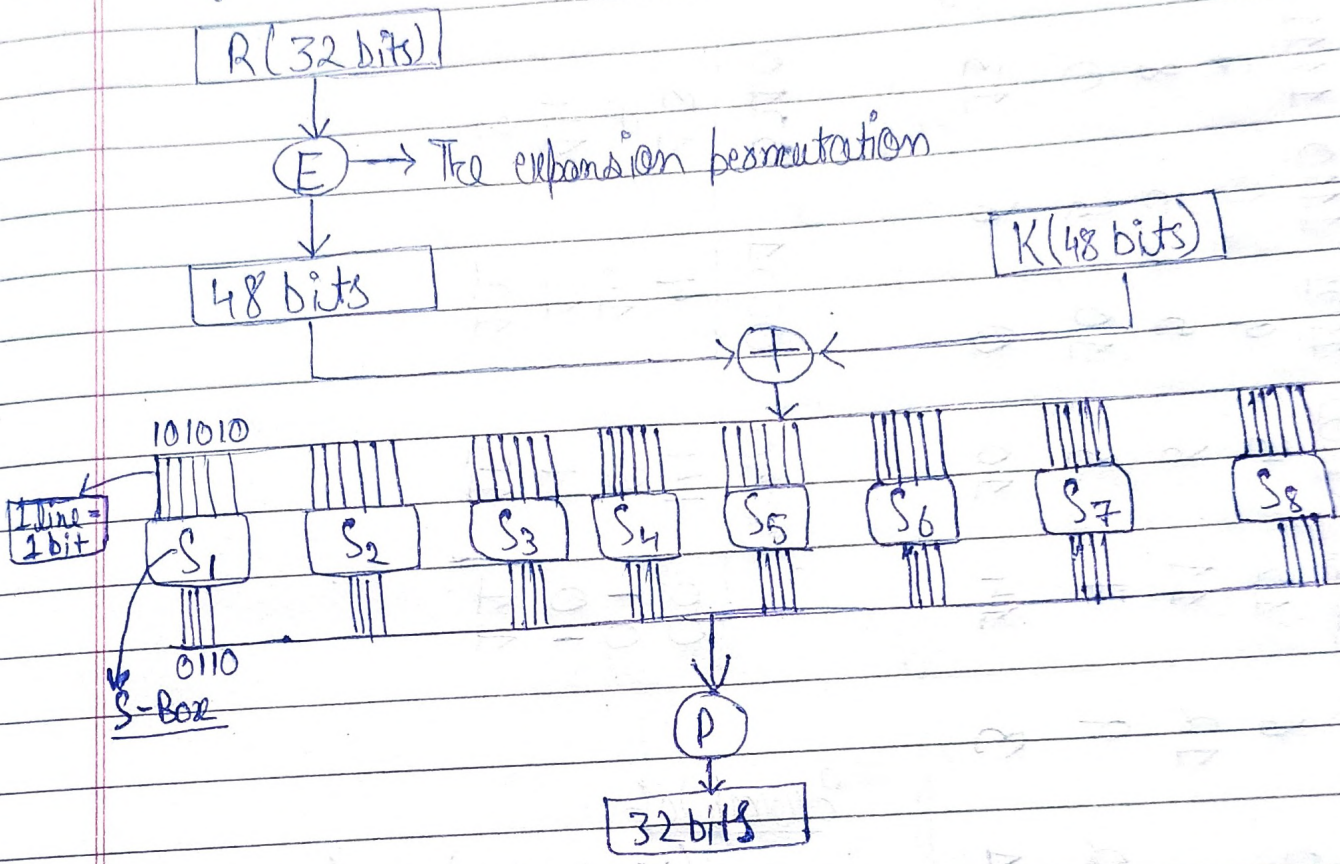
★ Single Round of DES algorithm :-



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

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

★ Mangler function:



★ The expansion permutation: A → A'

	A				
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

A'

* S-Box :- It's used to do substitution and to achieve confusion.

	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	1	2	15	11	8	3	10	6	12	5	9	0	7
01	4	14	2	13	1	10	6	12	11	6	5	3	8
10	8	13	6	2	11	15	12	9	7	3	10	5	0
11	2	4	9	1	7	5	11	3	14	10	0	6	13

	0000	0001	0010
00	14	4	13
01	0	15	7
10	4	1	14
11	15	12	8

Example:-

S(1001010) = 6 = 0110

10 → row

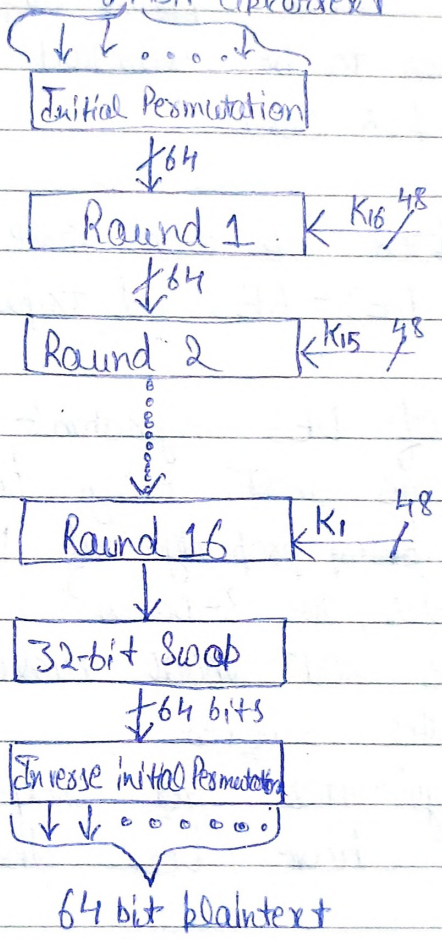
0101

↑ column

★ P-Box : Its used to do Permutation and to achieve diffusion.

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

★ DES decryption :-



★ Avalanche effect in DES :-

- A desirable property of any encryption algorithm.
- Avalanche effect :- Change in ciphertext on changing bit in plain text or private key Key.
- DES - Strong Avalanche effect.
  1. 1 bit change in PT - 34 bits change in CT on average.
  2. 1 bit change in Key - 35 bits changes in CT on average.

★ The strength of DES :-
1. The use of 56-Bit Keys.
2. The nature of DES algorithm.

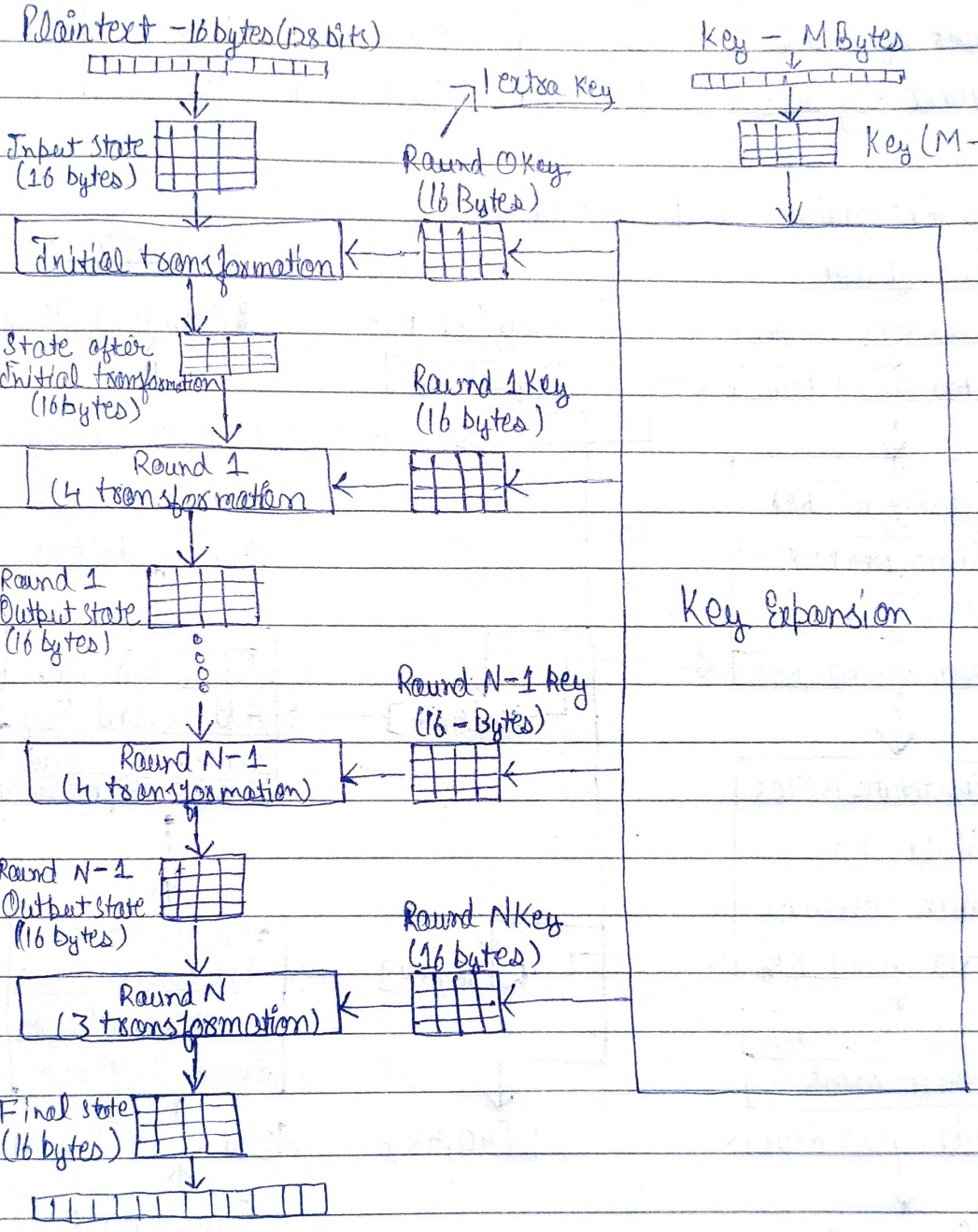
1. The use of 56-Bit Keys :-
- Subkey size : 56 bit keys.
- 2^{56} possible keys.
- 7.2×10^{16} keys.
- A brute-force attack appears impractical.
- key space has to be searched
- DES Cracker - $250,000.
- Deep Cracker
- The attack took less than three days.
- Alternatives to DES - AES and triple DES.

2. The nature of DES algorithm :-
- Cryptoanalysis is not easy because of Confusion and diffusion ~~first~~ property of DES using eight substitution tables, or S-boxes.
- S-boxes were not made Public.
- Weakness in the S-boxes.
- Number of regularities and unexpected behaviors of the S-boxes have been discovered

★ The timing attack :-
- Information about the key or the plaintext.
- How long it takes a given implementation to perform decryption on various Ciphertexts?
- A timing attack.
- Fairly resistant to a successful timing attack.
- Timing attack on DES, triple DES and AES?

AES (Advanced Encryption Standard) :-

- NIST in 2001.
- Symmetric block Ciphar.
- Widely used.



★ Relation B/w Key - M Bytes and Rounds.

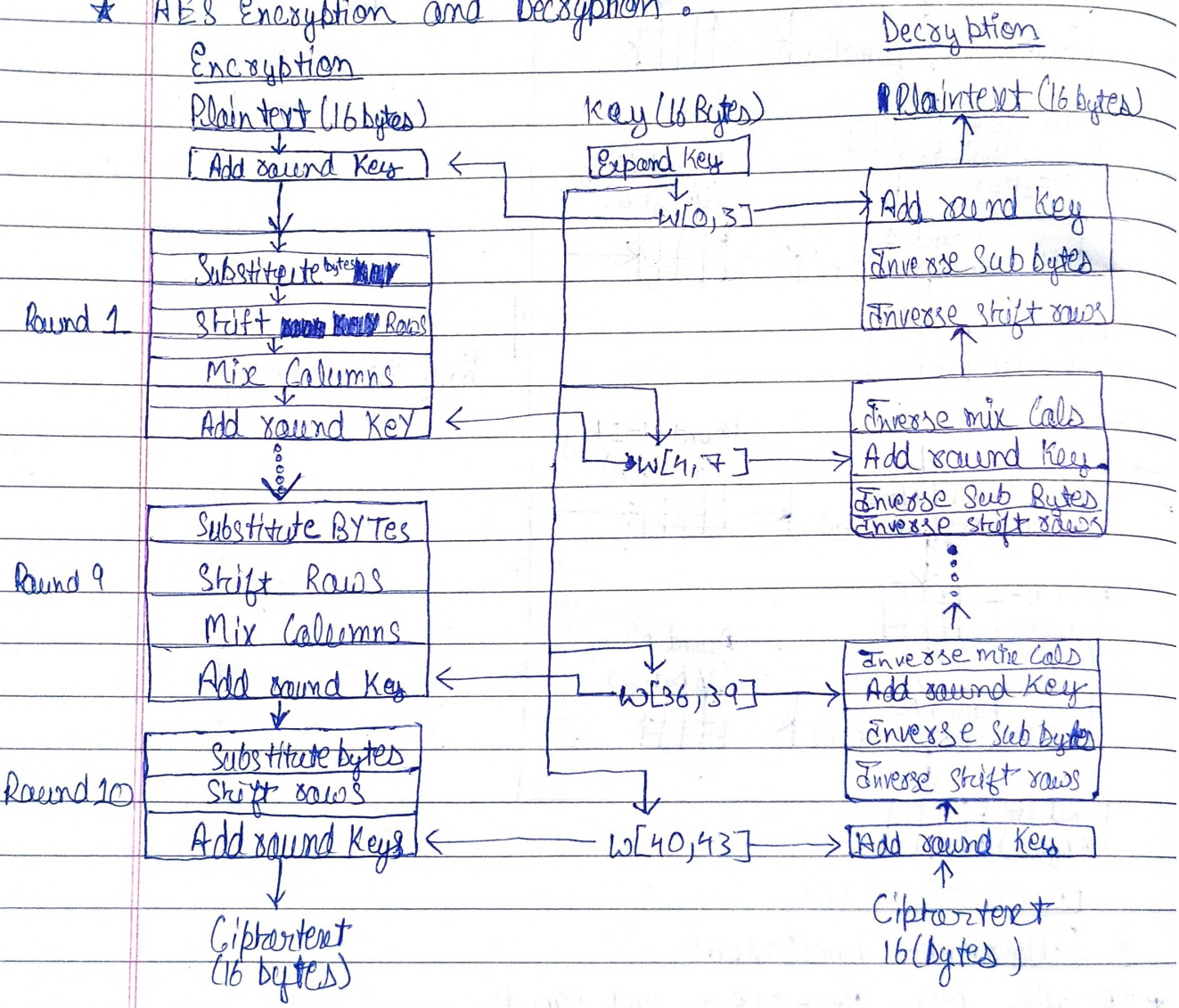
No. of Rounds	Key Size (in bits)
10	128
12	192
14	256

One extra key for Round 0 : Round 0 key

AES parameters :-

	AES-128	AES-192	AES-256
Key Size	128	192	256
Plain text Size	128	128	128
Number of rounds	10	12	14
Round Key Size	128	128	128

★ AES Encryption and Decryption :-

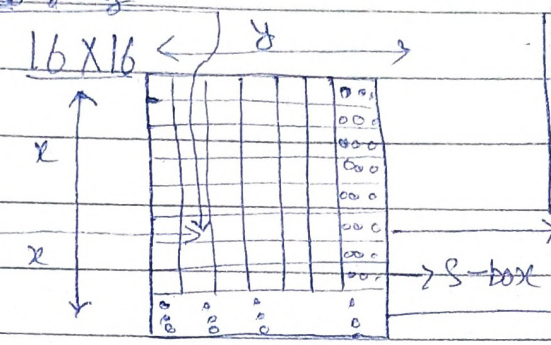


- w = 32 bit and four word = 128 bit.
- We have only 10 Round because Key Size is 16 bytes

★ AES Transformation Functions :-

1. Substitute Bytes — Substitution function
2. Shift Rows — Permutation function
3. Mix Columns — Substitution function
4. Add round key — Substitution function

1. Substitute bytes :-



To map on x need 4 bit because 16 can be represent in 4 bit and for y is same.

Search on internet - S-box-AES

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

→ 1 box = 1 byte

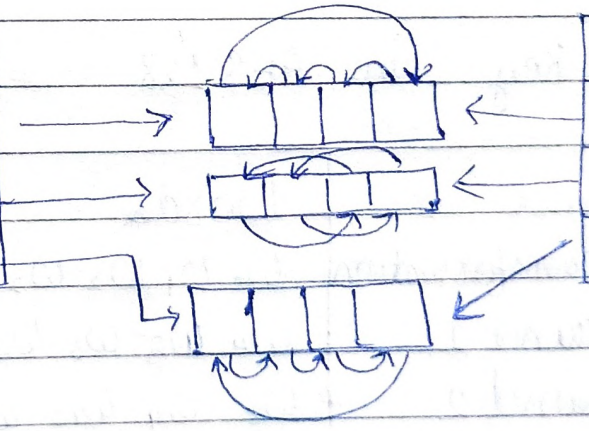
Starting 4 = x Bits

ending 4 = y

$S_{0,0}$	$S_{0,1}$	$S'_{0,2}$	$S'_{0,3}$
$S'_{1,0}$	$S'_{1,1}$	$S'_{1,2}$	$S'_{1,3}$
$S'_{2,0}$	$S'_{2,1}$	$S'_{2,2}$	$S'_{2,3}$
$S'_{3,0}$	$S'_{3,1}$	$S'_{3,2}$	$S'_{3,3}$

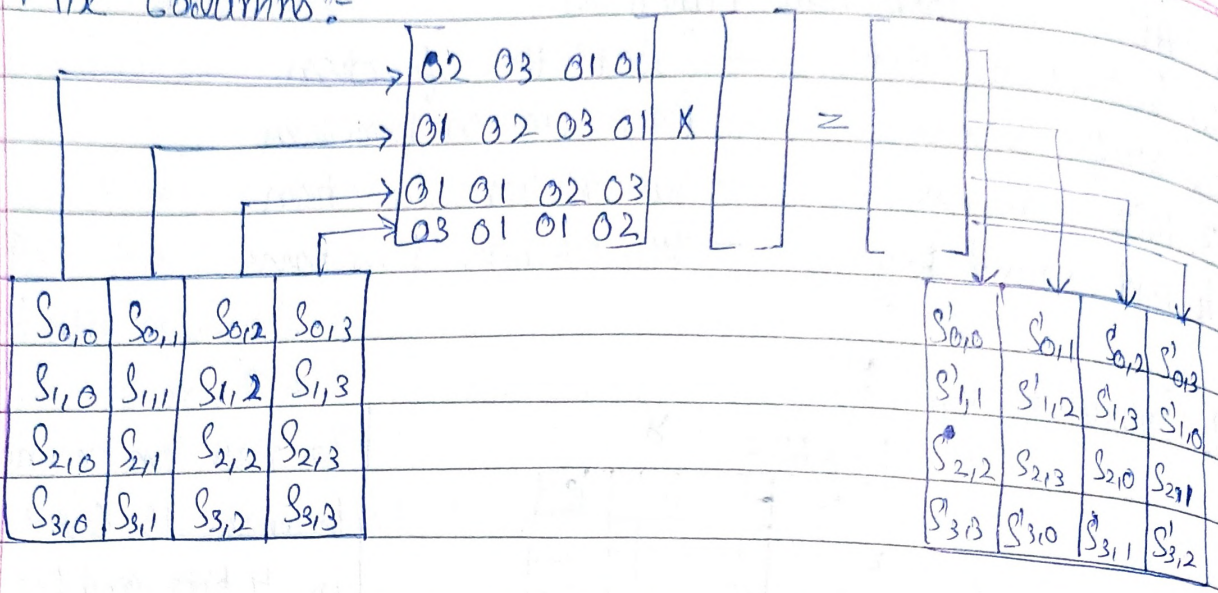
2. Shift Rows :-

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

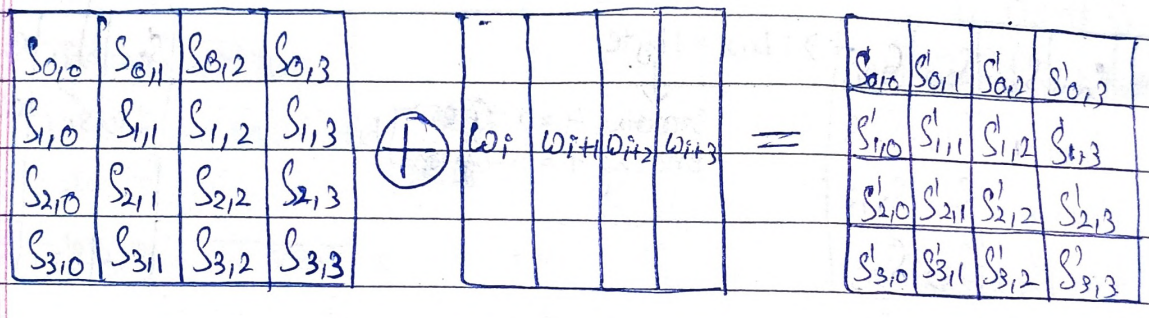


$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,0}$
$S_{2,2}$	$S_{2,3}$	$S_{2,0}$	$S_{2,1}$
$S_{3,3}$	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$

3. Mix Columns :-



4. Add round key :-

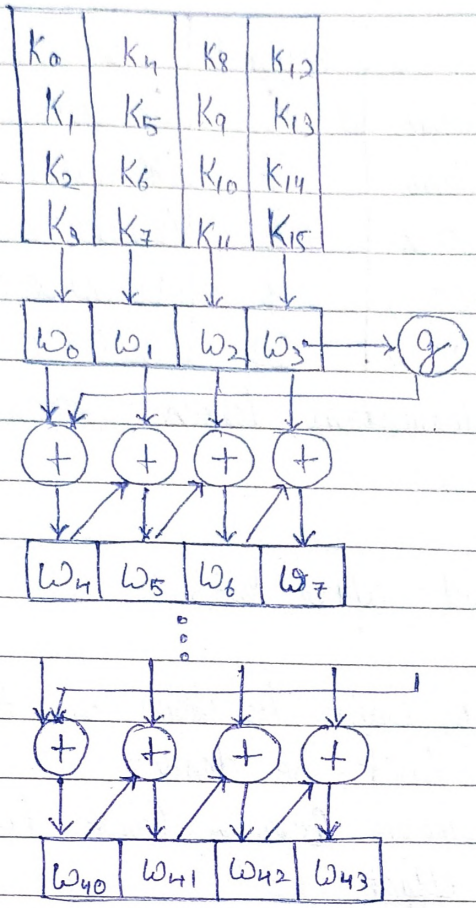


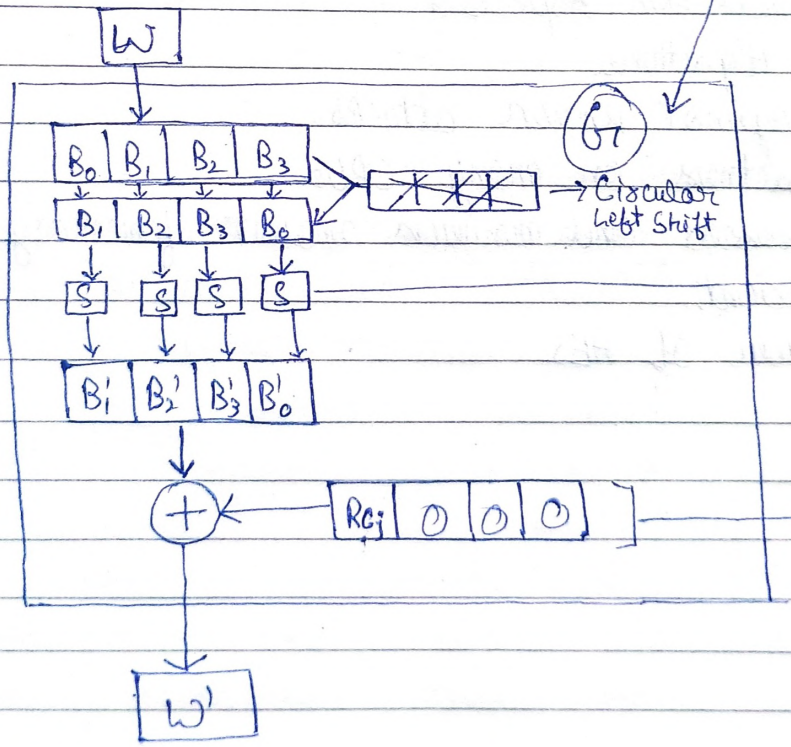
★ Key expansion :-

- Round key in AES 128

Round	Words
Initial transformation	w_0 w_1 w_2 w_3
Round 1	w_4 w_5 w_6 w_7
Round 2	w_8 w_9 w_{10} w_{11}
...	
Round 10	w_{40} w_{41} w_{42} w_{43}

- AES Key Expansion :-





S-Box in AES
Same as in Substitute Byte and Search on internet coz it's Big

Last 3 Column will Contain zero and Rcj will have value according to round

Rcj = Round Constant

★ Round Constant in AES-128 :-

Round	RC	Round	RC
1	$(01\ 00\ 00\ 00)_{16}$	6	$(20\ 00\ 00\ 00)_{16}$
2	$(02\ 00\ 00\ 00)_{16}$	7	$(40\ 00\ 00\ 00)_{16}$
3	$(04\ 00\ 00\ 00)_{16}$	8	$(80\ 00\ 00\ 00)_{16}$
4	$(08\ 00\ 00\ 00)_{16}$	9	$(1B\ 00\ 00\ 00)_{16}$
5	$(10\ 00\ 00\ 00)_{16}$	10	$(36\ 00\ 00\ 00)_{16}$

NOTE: Initial transformation takes $(00\ 00\ 00\ 00)_{16}$ as the RC.

★ AES Security :-

- AES was designed after DES.
- Simplicity.
- Brute-force attack can't be done on AES.
- Statistical attacks can't be done.
- Differential and linear Attacks can't be performed.
- Strong Avalanche effect.

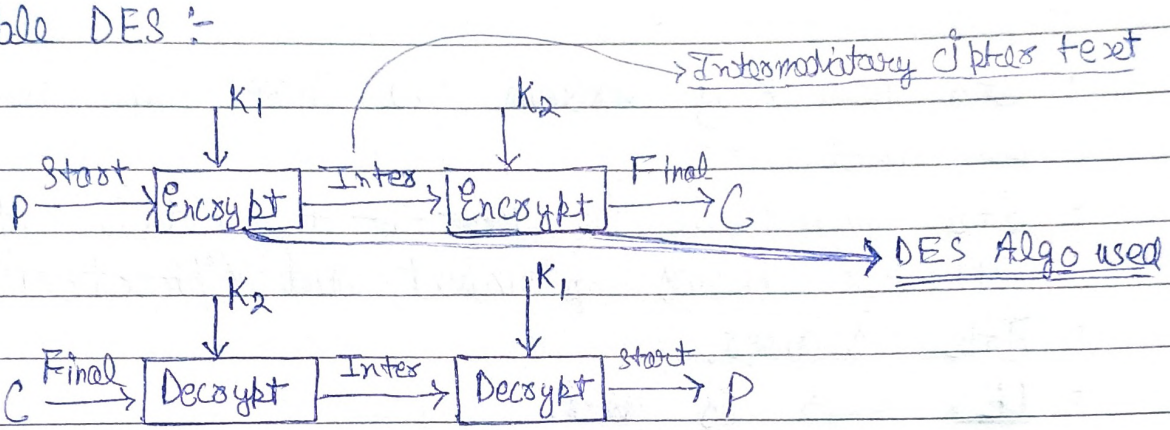
★ AES implementation Aspects :-

- Simple Algorithms.
- Resistant against known attacks.
- Code Compactness on many CPUs.
- Cheap processors and minimum amount of memory.
- Very efficient.
- Implementation of AES.

Multiple Encryption and Triple DES :-

- Drawbacks of DES
- Potential vulnerability of DES to a brute-force attack
- AES - Completely a new algorithm.
- Another alternative.
- Use of multiple encryption with DES and multiple keys.

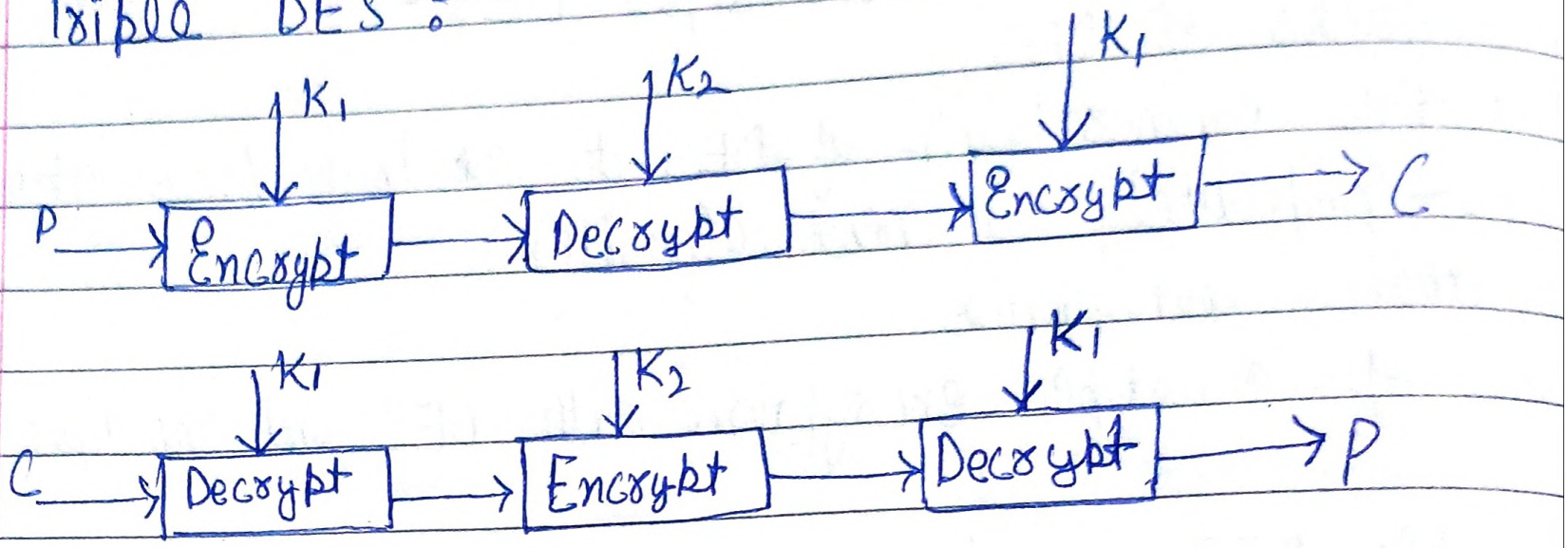
★ Double DES :-



- Meet-in-the-middle (MITM) attack :-

1. It is known-plaintext type of attack.
2. First attacker will use plaintext and apply all 2^{56} keys and will save all intermediate Ciphar.
3. Then attacker will use Ciphertext (he knows about Ciphertext and plaintext that's it's known plaintext attack) and decrypt using all 2^{56} keys He will save all intermediate Ciphar
4. He will match intermediate ciphar from encryption and decryption if matches from all then He will have private key K_1 Corresponding to that intermediate Ciphar from encryption and same as K_1 he will have K_2 Corresponding to that intermediat Ciphrex from decryption.

★ Triple DES :-



- It's Two key version and Three Can also be used.
- It's vulnerable to Meet-in-the-middle if attacker knows plaintext and Ciphertext.
- It's slower.
- DES algo is used.

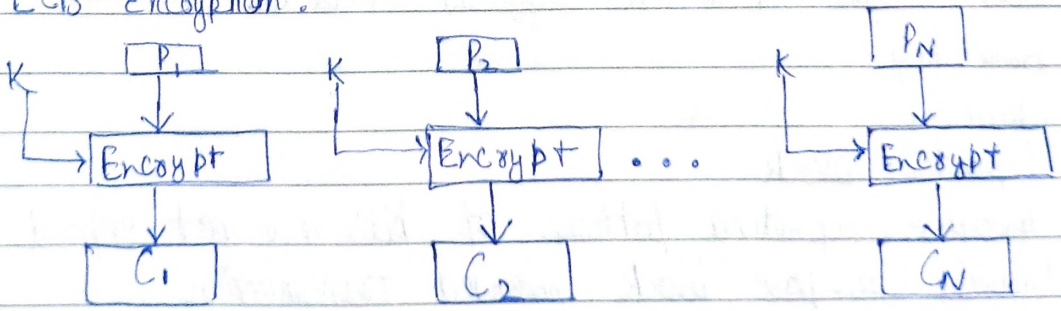
Block operation :-

- We don't just 'run a cipher' - we need a mode of operation.
- Fixed - length block.
- 'b' bits input and 'b' bits output.
- If the amount of PT (Plain text) to be encrypted is > 'b' bits?
- Breaking the plaintext into 'b' bits in each block.
- 5 modes of operation defined by NIST.
  (i) Electronic Codebook (ECB)
  (ii) Cipher Block chaining (CBC).
  (iii) Cipher Feedback (CFB).
  (iv) Output Feedback (OFB).
  (v) Counter mode (CTR).
- Does security issue arise when multiple blocks are encrypted.
- Different application - Different modes of operation.

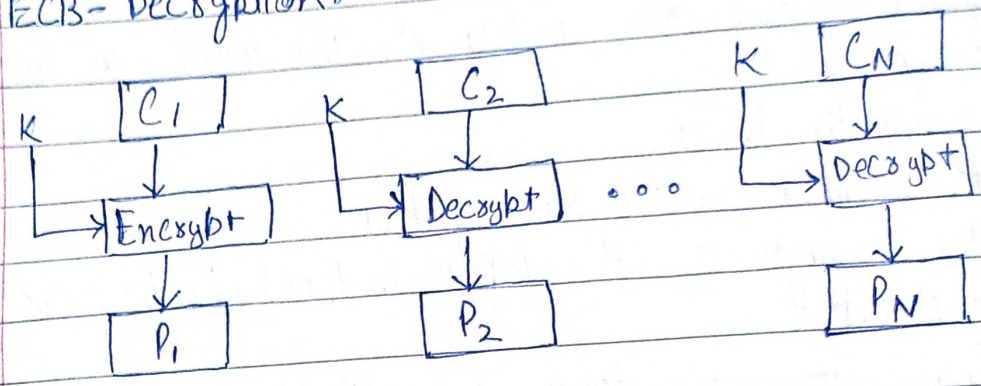
★ Electronic Codebook (ECB):-

- Data is length than 128 bit then data is broken in 128 bit and if last block size is less than 128 bit then its padded with 0 (zero) in end.
- Each block is encoded independently using the same key.

ECB - Encryption:



ECB - Decryption:



Pros:

1. Simplest mode.
2. Ideal for short amount of data. Ex: Secure transfer of AES or DES key.
3. Independent - Can encrypt any block.
4. Fast - parallelism

Cons:

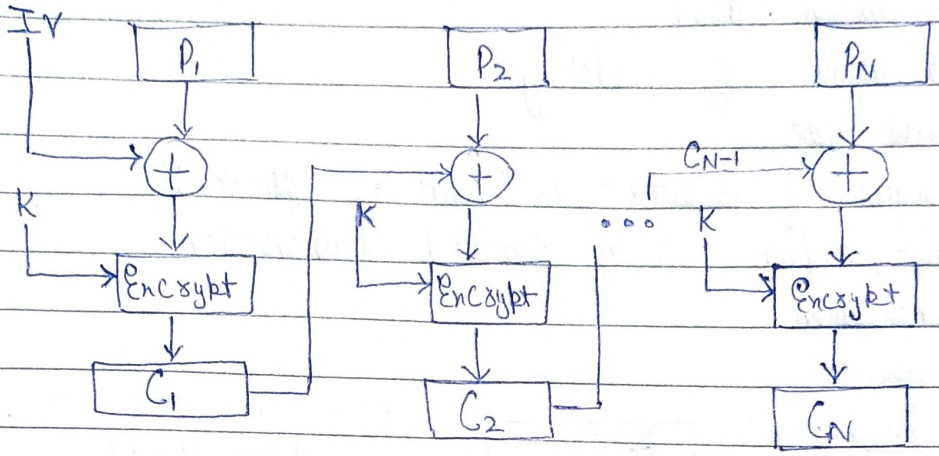
1. Not secure for lengthy messages Ex: attacker will know pattern of plaintext if ~~repeat~~ repeating.
2. Cryptanalyst can exploit the regularities of the message.
3. Ex: Like attacker can still know image of Penguin even if it's encrypted.

★ Cipher Block Chaining (CBC):

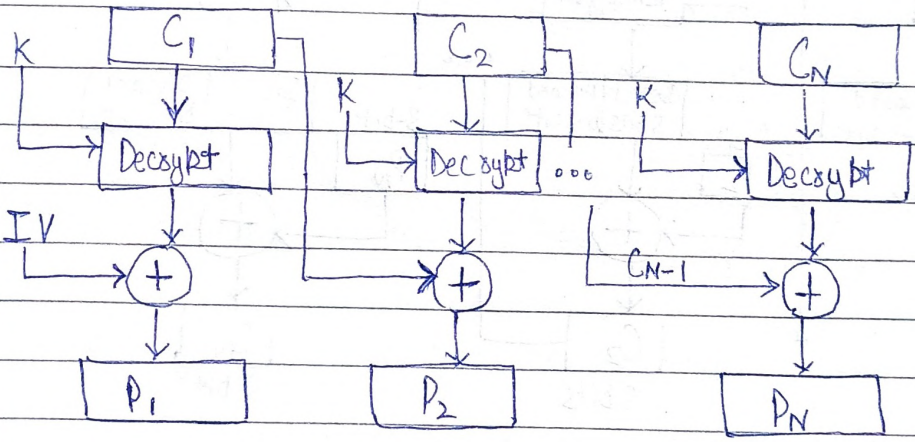
- ECB - Same PT block - Same CT block.
- CBC - Same PT block - Different CT block.
- Same key.
- Chaining.
- Dependent block.
- Therefore, repeating patterns of bits are not exposed.
- General - purpose block oriented transmission.

CBC - Encryption:

→ Initialization vector: known to both sender and receiver.



CBC - Decryption



Pros:

1. Appropriate mode for encrypting messages of length greater than 'b' bits.
2. Confidentiality + Authentication (Because receiver knows sender knows Initialization vector and that's why able to generate Ciphertext and Sender knows only person with IV will be able to decrypt it)
   ↓ Initialization vector

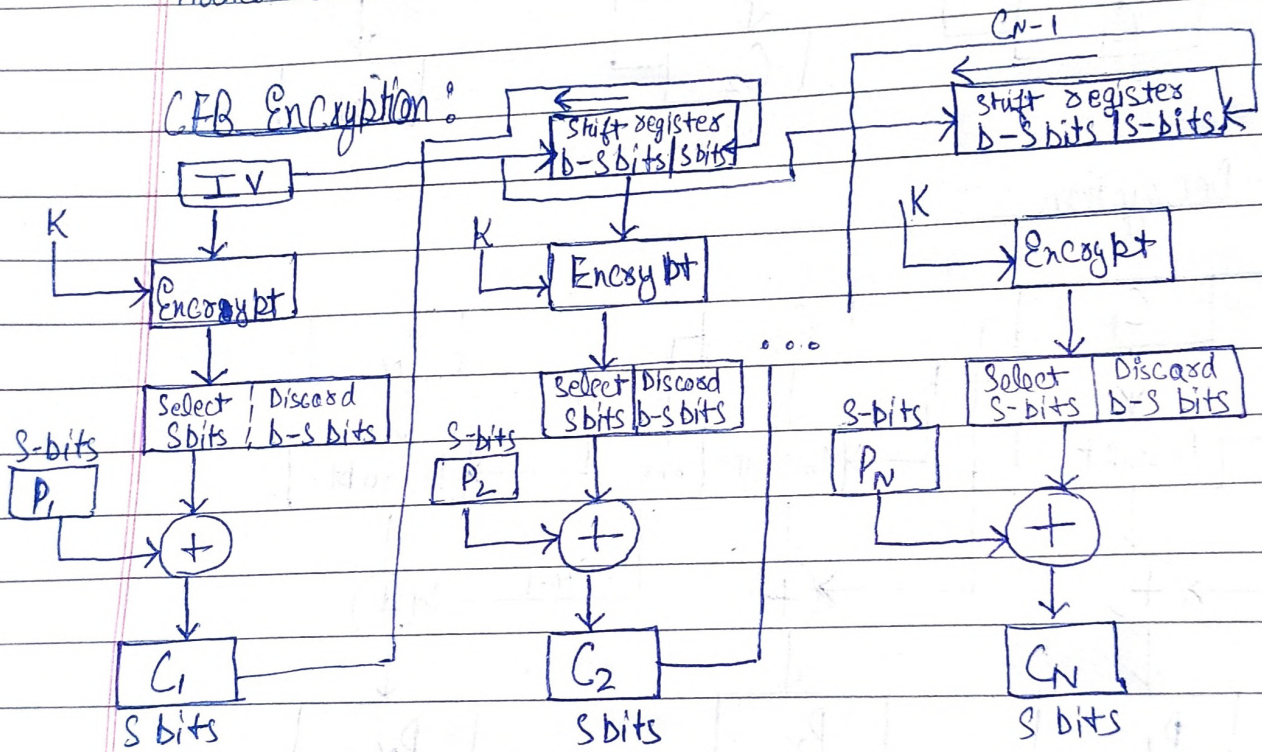
Cons:

1. Slow - No parallelism
2. Cannot encrypt any block since we need the ciphertext of previous block
3. IV which must be known to Sender and receiver.
   ↓ Initialization Vector

★ Cipher Feedback (CFB) :-

- Convert a block cipher to stream cipher
- Why stream cipher?
  (i) No need of padding
  (ii) Real time
  (iii) Length of plaintext = Length of Ciphertext
- General-purpose stream oriented transmission
- Authentication

CFB Encryption :

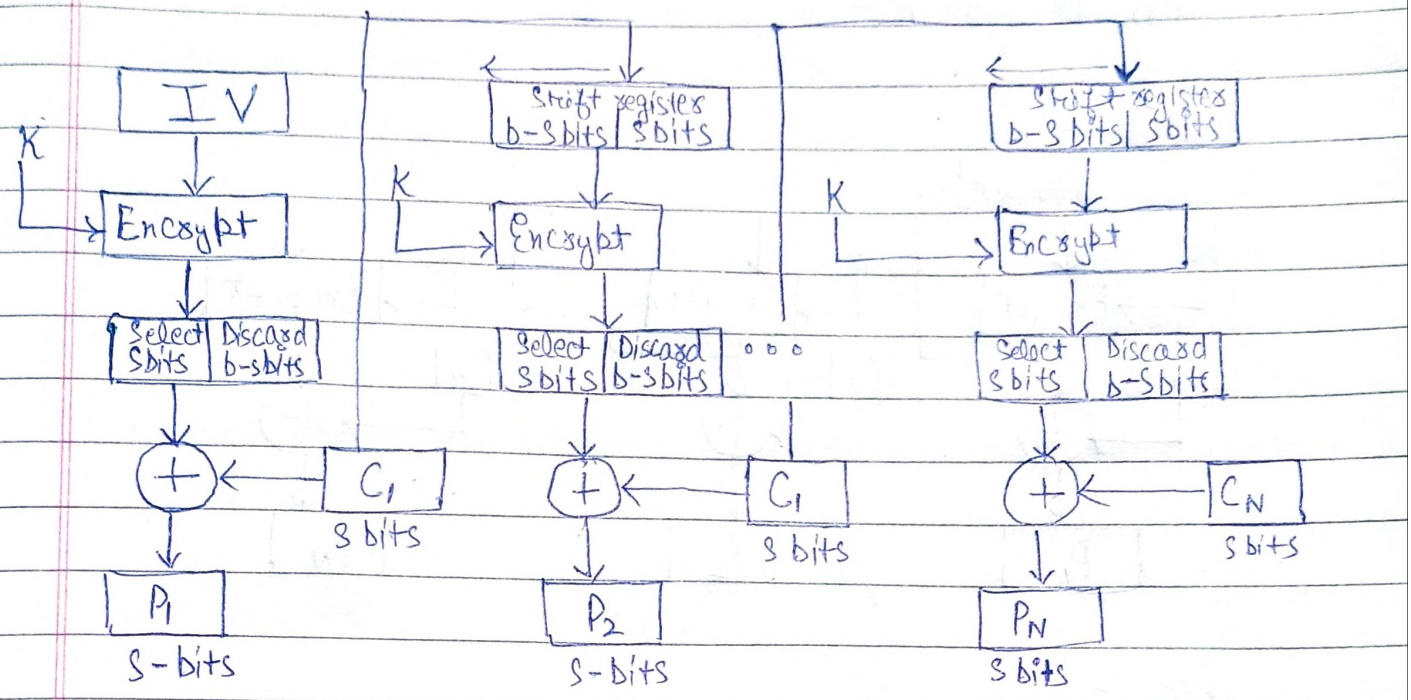


Shift register :- It performs left shift on IV total S-times and Concat Ciphertext text of previous Plaintext which of S size.

Pros :-

1. Can operate in real time.
2. Need of padding is eliminated.
3. Encryption function does decryption as well.
4. Length of PT = Length of CT.

CFB Decryption:-

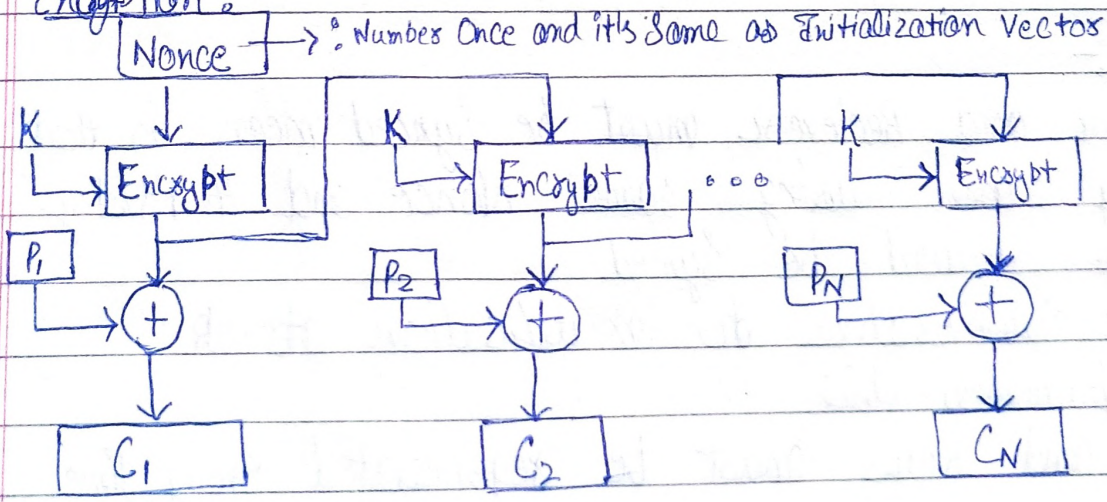


Cons:

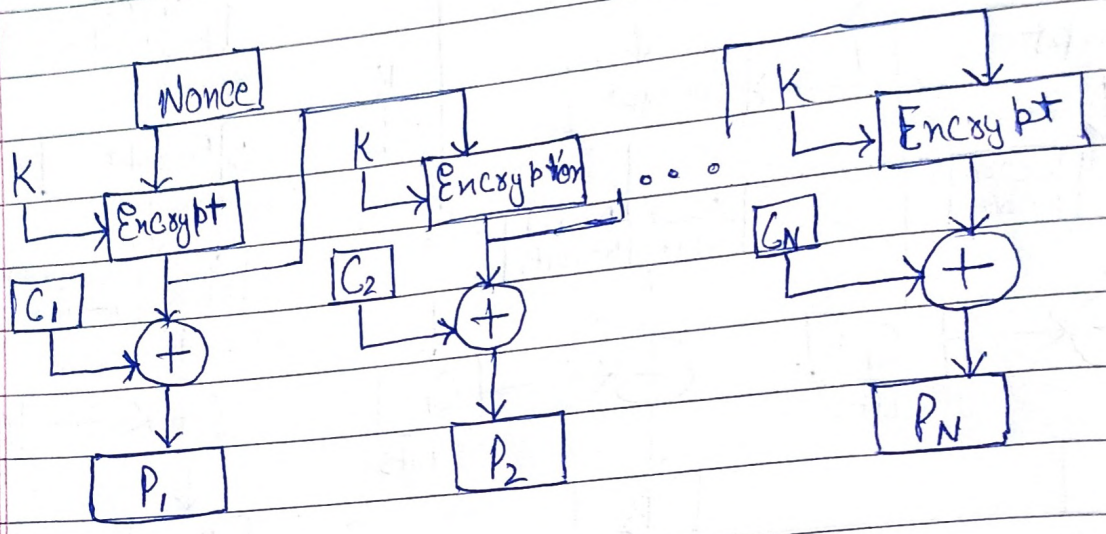
1. Chances of wastage of transmission Capacity.
2. Not a typical stream ciptor.

☆ Output Feedback:-

- Encryption:-



- OFB - Decryption :-



Pros :-

1. Bits error in transmission does not propagate meaning is that C_1 is not used in Encryption of C_2 and that's why any error in C_1 will not effect C_2.
2. Same PT - Same Key - Different CT.
3. PT length can be of random choice

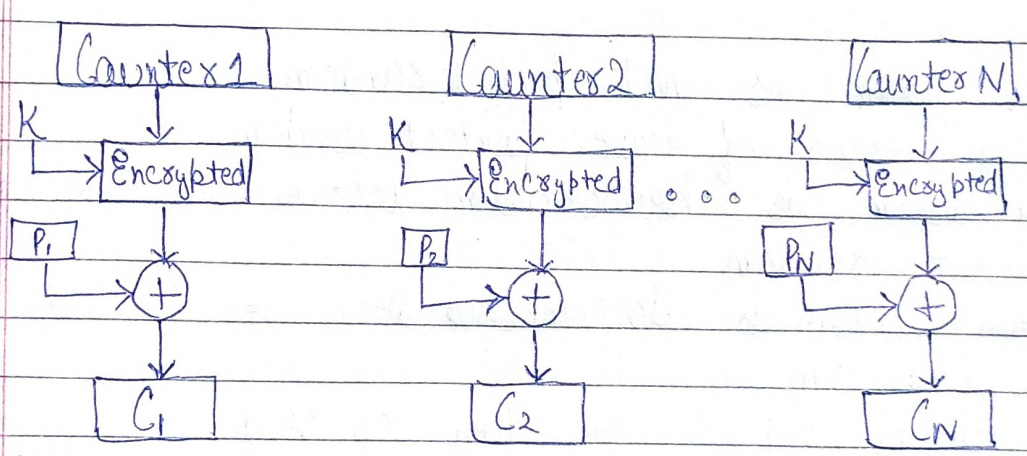
Cons :-

1. Sender and Reciever must be Synced mean is that they are using same Nonce and that's why they should be Synced.
2. More Vulnerable to modification attack
3. No parallelizable
4. IV and keys must be regenerated every time.

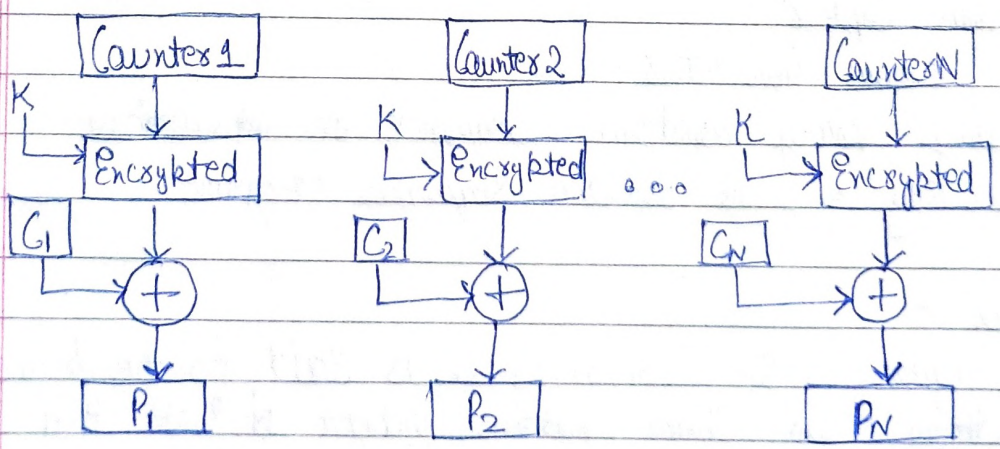
★ Counter Mode :-

- Application in ATM (Asynchronous transfer mode), IP Sec etc.
- Counter as replacement of Nonce or IV (Initialization Vector)
- Size of Counter = plaintext block size
- Different Counter value for each plaintext
- Counter value is initialized
- Counter ++.
- Decryption - Same sequence of Counter values is used.
- Decryption - Initial value of Counter is made available
- What about the last block?

CTR-Encryption :-

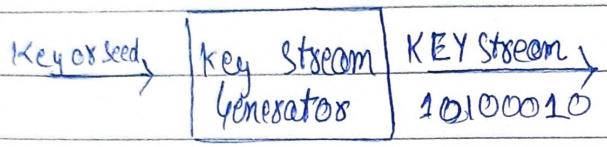


CTR-Decryption :-



- Pros:
1. Hardware efficiency.
2. Software efficiency.
3. Preprossing.
4. Random Access
5. Provable Security.
6. Simplicity

Pseudorandom Number Generator:-



- Key Stream is not perfect random.
- Only nature of have perfect random.
- Key Stream is Pseudorandom because it's close to perfect random.
- When randomness is repeated then it's called pseudorandom.
- Randomness depends on Key or seed.

- Stream Ciphor.
- Key Stream Generator.
- Generating truly random Sequence in impractical.
- Same Key = Same random Sequence everytime.

★ Period:-

- The Sequence $S = S_0, S_1, S_2, S_3 \ldots$ is said to be periodic if there is some positive integer N such that $S_{i+N} = S_i$ and smallest value of N is called the period of the Sequence.

* Run :- A Run is a binary sequence is defined as a set of consecutive 0s or 1s. Example :- 0000 or 1111 or 00.

* Autocorrelation :- Let $A = (a_0, a_1, a_2, \ldots a_{n-1})$ be a binary sequence of length n, then autocorrelation $C(i)$ of A is defined as $A \oplus (A <<< i)$.

* Autocorrection Function :- The auto-correlation function $C(\tau)$ of a binary sequence $a_0, a_1, a_2, \ldots, a_p$ is defined as
$C(\tau) = \frac{1}{p} \sum_{n=0}^{p-1} a_n \oplus a_{n+\tau}$ where

τ = Phase shift of the sequence $\{a_n\}$.
$C(\tau)$ = measure of amount of similarity between the sequence and the phase shift.

Golomb's Randomness Postulates :-
(1) In every period, the number of 1's differs from the number of 0's by at most 1.
(2) In every period, half the runs have length 1, 1/4th have length 2, 1/8th have length 3, 1/16th have length 4 and so on and so forth, as long as the number of runs so indicted exceeds 1. Moreover, for each of these lengths, there are almost equally runs of 0's and 1's.
(3) The auto-correlation function should be two valued.
$C(\tau) = \frac{1}{p} \sum_{k=0}^{n} a_n \oplus a_{n+\tau}$

Example :-
i) Consider the periodic sequence $S = 0001011$.
(a) No. of 0s = 4 , No. of 1s = 3
$4 - 3 = 1 \leq 1$
It follow rule one of Golomb's rules.

(b) Runs : 000 1 0 11

Total runs = 4

2 runs of length 1

1 run of length 2

1 run of length 3

It's satisfied 2nd rule of Golomb's rules.

(c)

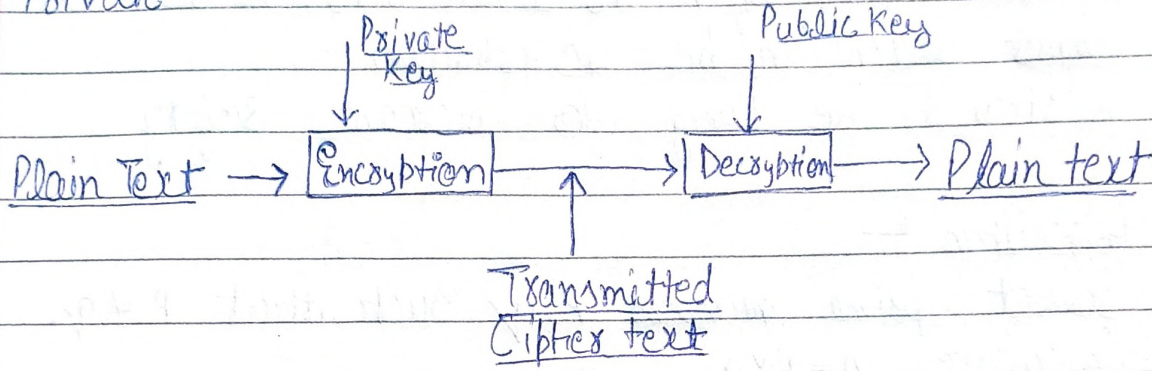
T	S<<<T	S ⊕ S<<<T	C(T)
0	0001011	0000000	0/7
1	0010110	0011101	4/7
2	0101100	0100111	4/7
3	1011000	1010011	4/7
4	0110001	0111010	4/7
5	1100010	1101001	4/7
6	1000101	1001110	4/7

It's satisfied third rule of Golomb's

* Three rules are satisfied by 0001011 and that's why we can say it's pseudo random.

Asymmetric Cryptography:-

- It's also known as Public-key Cryptography.
- Different keys are used to Encrypt and Decrypt message.
- One Key will be Public and One key will Private



It's example of blockchain where Encryption key is Private key and decryption key is Public.

★ Symmetric vs Asymmetric Cryptography:-

Private-Key Cryptography	Public-Key Cryptography
Same algorithm Same Key Key is kept secret faster	Different algorithm Different Key One of two key is kept secret slower
Ex:- Classical Encryption	RSA, Diffie-Hellman, ECC and Rabin CryptoSystem

★ Applications of Public-Key Cryptosystems:-

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	✓	✓	✓
ECC	✓	✓	✓
Diffie-Hellman	X	X	✓
DSS	X	✓	X

The RSA Algorithm :-

- General-purpose approach to public-key encryption.
- Block Cipher.
- Plain text and Ciphertext are integers between 0 and $n-1$ for some 'n'.
- A typical size of 'n' is 1024 bits, ~~or 309 decimal digits~~ and 'n' can be 2048 bits also.
- n should be Large to increase safety.

★ Algorithm :-

1. Select prime numbers P, q such that $P \neq q$.
2. Calculate $n = P \times q$
3. Calculate $\phi(n) = (P-1) \times (q-1)$.
4. Select integer 'e' such that $GCD(\phi(n), e) = 1$ and $1 < e < \phi(n)$.
5. Calculate 'd' such that $e \times d \equiv 1 \bmod \phi(n)$.
6. Public key $PU = \{e, n\}$. Here Public key is Encryption key.
7. Private key $PR = \{d, n\}$. Here Private key is decryption key.

- Encryption by Bob using Alice's Public key :-

  Plaintext : $M < n$
  Ciphertext : $C = M^e \bmod n$

- Decryption by Alice using Alice's Private key.

  Ciphertext : C
  Plain text : $M = C^d \bmod n$

Examples:-

Q1. Perform Encryption for the plaintext 20 using RSA algorithm with the values $P=5$, $q=11$, 13 as the Public key.

Ans. $P=5$, $q=11$ and $P \neq q$

$n = 5 \times 11 = 55$.

$\phi(n) = (P-1) \times (q-1) = (5-1)(11-1) = 40$.

$e = 13$ $\because$ $GCD(40,13) = 1$ and $1 < 13 < 40$.

$d = 37$ $\because$ $13 \times 37 = 1 \bmod 40$.

$PU = \{13, 55\}$.

$PR = \{37, 55\}$.

★ Encryption:

- Plaintext: $m = 20$

  $\because 20 < 55$

- Ciphertext:

  $C = 20^{13} \bmod 55 = 25$

★ Decryption:

- Ciphertext: $C = 25$
- Plaintext: $M = 25^{37} \bmod 55 = 20$

NOTE:-

1. When Our Encryption key is private and decryption key is public then this method provides authentication because receiver knows Only person with private key can Encrypt.
2. When Our decryption key is private and encryption key is Public then this method provides Confidentiality because receives will only be able to decrypt with his private key.

★ Encryption key is private to sender and provides authentication

★ Encryption key is Public to sender and provides Confidentiality

★ The Security of RSA :-

1. Brute Force attacks.
2. Mathematical attacks.
3. Timing Attacks.
4. Chosen Ciphertext attacks.

1. Brute Force attack :-
   - This involves trying all possible keys.
   - Larger key space.
   - Larger number of bits in d, the better.

2. Mathematical attacks :-
   - The factoring problem.
   - 3 approaches to attacking RSA mathematically :-
   (i) Factor 'n' in to two primes.
   This enables the Calculation of $\phi(n) = (p-1)(q-1)$ which in turn helps to determine $ed \equiv 1 \bmod \phi(n)$.
   (ii) Determine $\phi(n)$ directly, without p and q.
   This in turn helps to determine $ed \equiv 1 \bmod \phi(n)$.
   (iii) Determine d directly, without first determining $\phi(n)$.

3. Timing attacks :-
   - Running time of the decryption algorithm.
   - The timing attack is a serious threat.
   - Countermeasures :- (i) Constant exponentiation time.
   - (ii) Random delay (iii) Blinding.

4. Chosen Ciphertext Attacks :-
   - Vulnerable to a chosen Ciphertext attack.
   - Sophisticated CCAs (Chosen ciphertext attacks).
   - Countermeasures :- (i) Padding.
   (ii) OAEP - Optimal Asymmetric Encryption Padding.

Diffie – Hellman Key Exchange :-

★ Need :-

1. Two parties with no prior knowledge.
2. Insecure channel.
3. Jointly establish a shared secret key.

★ Algorithm :-

- Global Public Elements :- q – Prime number and α – Primitive root of q and $\alpha < q$.
- Global Public elements will be known to everyone on internet.

Alice :-	Bob :-
Key Generation by Alice	Key Generation by Bob
• Select Private key X_A, $X_A < q$	• Select Private key X_B, $X_B < q$
• Calculate Public Key Y_A	• Calculate Public Key Y_B
$Y_A = \alpha^{X_A} \bmod q$	$Y_B = \alpha^{X_B} \bmod q$
• Calculation of shared secret key by Alice	• Calculation of shared secret key by Bob.
$K_A = Y_B^{X_A} \bmod q$	$K_B = Y_A^{X_B} \bmod q$

- At the end both are able to generate shared secret key without revealing private keys.
- Shared Secret Key = $K_A = K_B = \alpha^{X_A X_B} \bmod q = Y_B^{X_A} \bmod q = Y_A^{X_B} \bmod q$.
- Shared key is same for both that's why they can prefer only Symmetric algo.

Example :-

Q1. Find the secret key shared between user A and user B using Diffie-Hellman key exchange algorithm for the following values.

$q = 353$

α (primitive root) $= 3$

$X_A = 45$ and

$X_B = 50$.

Sol.

$Y_A = 3^{45} \bmod 353 = 143 \rightarrow$ Public key Y_A

$Y_B = 3^{50} \bmod 353 = 155 \rightarrow$ Public key Y_B

$K_A = Y_B^{X_A} \bmod q$

$= 155^{45} \bmod 353$

$= 197$

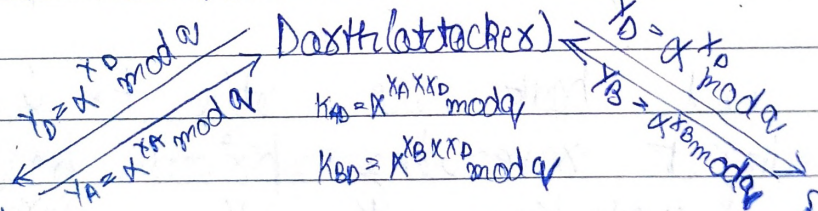
$K_B = Y_A^{X_B} \bmod q$

$= 143^{50} \bmod 353$

$= 197$

Shared Secret key $= K_A = K_B = 197$.

★ Man-in-the-middle attack :-



Alice:

$K_{AD} = \alpha^{X_A * X_D} \bmod q$

Bob:

$K_{BD} = \alpha^{Y_B * X_D} \bmod q$

Alice and Bob can be attacked by attacker because Diffie-Hellman is vulnerable to man-in-the-middle attack. They are thinking that they are talking to each other but Dosth is reading all messages by decrypting and encrypting them and he can even change them.

* Applications :-
1. Secure Shell (SSH).
2. Transport Layer Security (TLS) / Secure Socket Layer (SSL).
3. Public Key Infrastructure (PKI).
4. Internet protocal Security (IKE).
5. Internet protocal Security (IPSec).

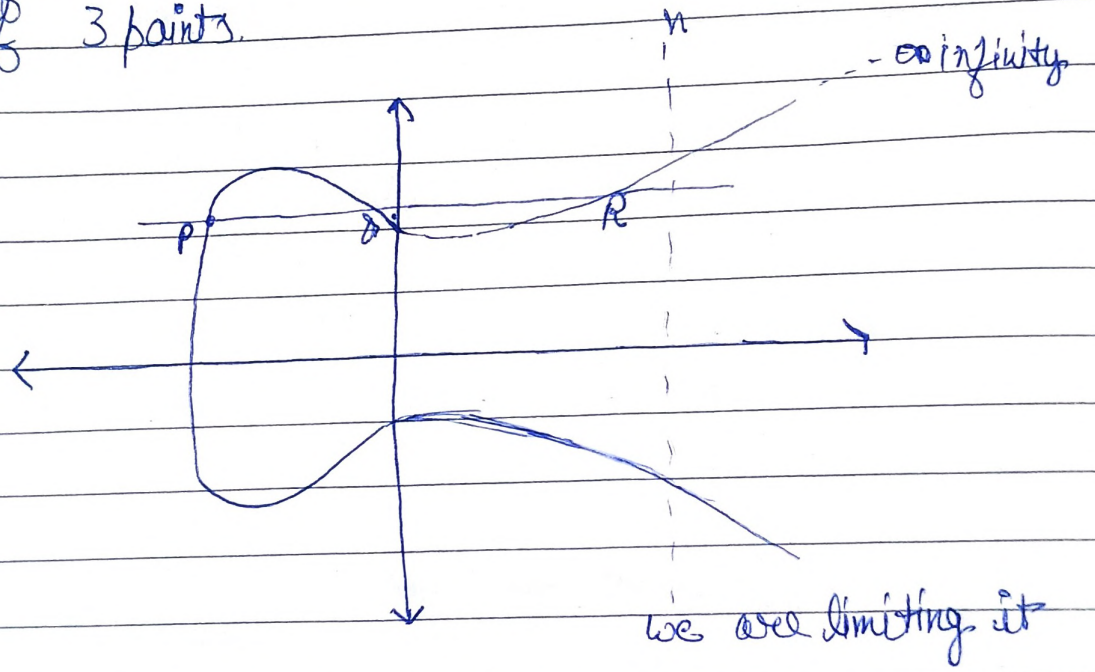
* Limitations :-
1. Non-authenticated key-agreement protocal.
2. Poor authentication.
3. Man-in-the-middle attack.
4. Cannot be used for encrypting messages.

Elliptic Curve:-

- An elliptic curve is a set of points that satisfy the mathematical equation:
  $y^2 = x^3 + ax + b$ where a and b are some constants with condition $4a^3 + 27b^2 \neq 0$.

Properties:

1. The elliptic curve is required to be non-Singular.
2. No Cusps or self-interactions.
3. This is equivalent to the condition: $4a^3 + 27b^2 \neq 0$
4. An elliptic curve is an abelian variety.
5. Symmetric to x-axis.
6. If we draw a line, it will touch a max of 3 points.



- A Trapdoor function is a function that is easy to Compute in One direction, yet ~~different~~ difficult to Compute in the opposite direction (finding its inverse) without special information, Called the trapdoor.

Special information is Called trapdoor value.

Elliptic Curve – types

1. Elliptic Curve over real numbers
2. Elliptic Curve over Complex numbers
3. Elliptic Curve over Finite Fields. ✓

Elliptic Curve over finite fields.

1. Elliptic Curve over finite fields is used in Cryptography.
2. Finite fields : GF(2), GF(5), GF(8) etc.
3. But in Cryptography we prefer Prime numbers that's why we will take finite fields : GF(P).
4. $E_p(a,b) = (y^2 \equiv x^3 + ax + b \pmod P)$.
5. NOTE : a and b must be lesser than P.
6. $4a^3 + 27b^2 \pmod P \neq 0$

Example :-

Q. Find all the points on the $E_5(1,1)$.

Ans. GF(5) = (2,5) = {0,1,2,3,4}.

The equations :

$y^2 \equiv x^3 + ax + b \pmod P$ — ①

$4a^3 + 27b^2 \pmod p \neq 0$ — ②

Given : $P = 5$, $a = 1$ and $b = 1$.

Substitute the value of a, b and P in ②,

$4 + 27 \pmod 5 \neq 0$

$31 \bmod 5 \neq 0$

$1 \neq 0$

∴ Substitute the values of a, b and P in ①.

$y^2 \equiv x^3 + ax + b \pmod P$.

$y^2 \pmod P = x^3 + ax + b \pmod P$

$y^2 \pmod 5 = x^3 + x + 1 \pmod 5$

	$y^2 \pmod 5$	$x^3+x+1 \pmod 5$
0	0	1
1	1	3
2	4	1
3	4	1
4	1	4

The points on the $E_5(1,1)$ are $(0,1), (0,4), (2,1), (2,4), (3,1), (3,4), (4,2), (4,3)$.

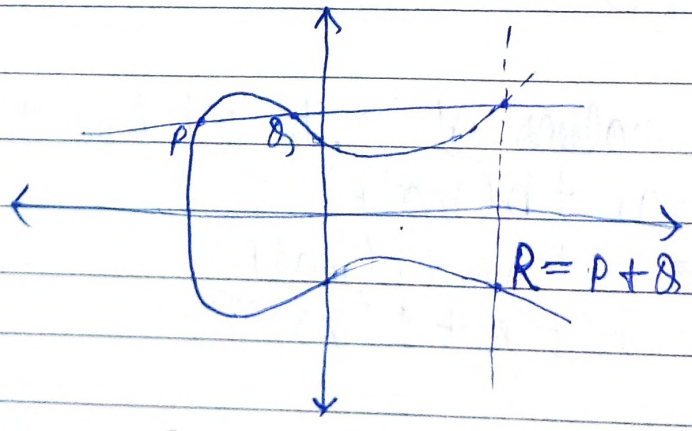
Q2. Does the point $(2,4)$ lie on Elliptic Curve $E_5(1,1)$?

Ans:- $a=1, b=1, P=5$

$$y^2 \equiv x^3 + ax + b \pmod P$$
$$y^2 \equiv x^3 + x + 1 \pmod 5$$
$$y^2 \equiv x^3 + x + 1 \pmod 5$$
$$4^2 \equiv 2^3 + 2 + 1 \pmod 5$$
$$16 \equiv 11 \pmod 5$$
$$16 \bmod 5 = 11 \bmod 5$$
$$\mathbf{1 = 1}$$

L.H.S is equal to R.H.S that's point $(2,4)$ lie on elliptic curve $E_5(1,1)$.

Adding two points on the elliptic curve.



★ Let $P=(x_1, y_1)$ and $Q=(x_2, y_2)$ be in the elliptic curve $E_p(a,b)$ and $Q \neq -P$, then the addition does the elliptic curve $E_p(a,b)$ is $R(x_3, y_3)$ such that

$$x_3 = \lambda^2 - x_1 - x_2 \pmod p$$
$$y_3 = \lambda(x_1 - x_3) - y_1 \pmod p$$
$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod p$$

Q1. Find $P+Q$, if $P=(3,8)$ and $Q=(6,5)$ on $E_{11}(1,1)$.

Ans:- $x_1=3$, $y_1=8$, $x_2=6$, $y_2=5$, and $p=11$.

Since $P \neq Q$ and $P \neq -Q$.

$$x_3 = \lambda^2 - x_1 - x_2 \pmod p$$
$$y_3 = \lambda(x_1 - x_3) - y_1 \pmod p$$
$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod p$$
$$\lambda = \frac{5-8}{6-3} \pmod{11}$$
$$\lambda = \frac{-3}{3} \bmod 11 = -1 \bmod 11 = 10$$
$$x_3 = \lambda^2 - x_1 - x_2 \pmod p = 10^2 - 3 - 6 \pmod{11}$$
$$x_3 = 3$$
$$y_3 = \lambda(x_1 - x_3) - y_1 \pmod p$$
$$y_3 = 10(3-3) - 8 \pmod{11}$$
$$y_3 = -8 \pmod{11}$$
$$y_3 = 3$$
$$R = (P+Q) = (x_3, y_3) = (3,3)$$

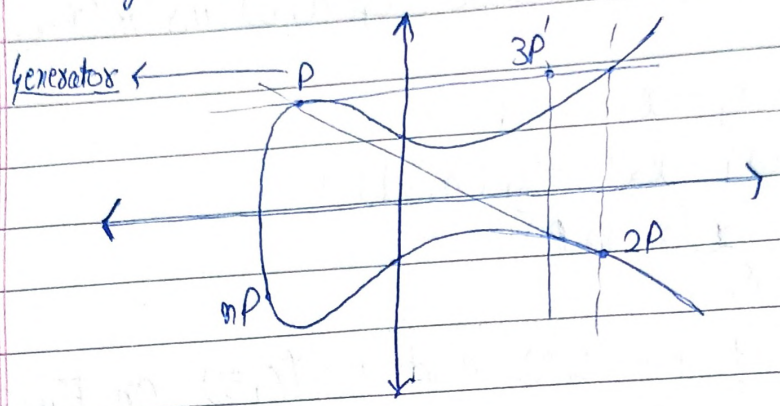
R lie on $E_{11}(1,1)$.

Let's check it that (3,3) lie on $E_{11}(1,1)$.

$$y^2 \equiv x^3 + ax + b \pmod p.$$
$$y^2 \pmod{11} = x^3 + x + 1 \pmod{11}$$
$$3^2 \pmod{11} = 3^3 + 3 + 1 \pmod{11}$$
$$9 \pmod{11} = 27 + 3 + 1 \pmod{11}$$
$$9 = 9$$

Not checking for $4a^3 + 27b^2 \pmod p \neq 0$ because we checked knows it's true or You can do also.

★ Adding two same points



- Let $P = (x_1, y_1)$ and $Q = (x_1, y_1)$ be in the elliptic curve $E_p(a, b)$ i.e., $P = Q$, then the addition over the elliptic curve $E_p(a, b)$ is $2P(x_3, y_3)$ such that

$$x = x_1, \quad y = y_1$$
$$x_3 = \lambda^2 - x - x \pmod p$$
$$y_3 = \lambda(x - x_3) - y \pmod p$$
$$\lambda = \frac{3x^2 + a}{2y} \pmod p$$

Q1:- Find $2P$, if $P = (4, 6)$ on $E_{11}(1, 1)$.

Given: $x = 4$, $y = 6$, $p = 11$, $a = 1$ and $b = 1$.

Since $P = Q$

$$x_3 = \lambda^2 - x - x \pmod p$$
$$y_3 = \lambda(x - x_3) - y \pmod p$$
$$\lambda = \frac{3x^2 + a}{2y} \pmod p$$
$$\lambda = \frac{3(4)^2 + 1}{2(6)} \pmod{11}$$
$$\lambda = \frac{49}{12} \pmod{11} = 49 \times 12^{-1} \pmod{11}$$

↓ using multiplicative inverse

$$\lambda = 49 \times 1 \pmod{11}$$
$$\lambda = 5$$

$x_3 = \lambda^2 - x - x \pmod p$

$x_3 = 5^2 - 4 - 4 \pmod{11}$

$x_3 = 6$

$y_3 = \lambda(x - x_3) - y \pmod p$

$y_3 = 5(4 - 6) - 6 \pmod{11}$

$y_3 = -16 \pmod{11}$

$y_3 = 6$

$R = (P + P) = 2P = (x_3, y_3) = (6, 6)$.

Let's check if $(6, 6)$ lie on $E_{11}(1, 1)$.

Not checking for $4a^3 + 27b^2 \pmod p \neq 0$ but you can do also and it's true.

$y^2 \equiv x^3 + ax + b \pmod p$

$y^2 \pmod p = x^3 + ax + b \pmod p$

$y^2 \pmod{11} = x^3 + x + 1 \pmod{11}$

$6^2 \pmod{11} = 6^3 + 6 + 1 \pmod{11}$

$36 \bmod 11 = 223 \bmod 11$

$3 = 3$

L.H.S equal to R.H.S that's why $(6, 6)$ lie on $E_{11}(1, 1)$

★ Adding two points on the elliptic curve have mainly two type of formula's One for $P \neq Q$ and $Q \neq -P$ and other for $P = Q$

◎ if $P \neq Q$ and $Q \neq -P$ then

$x_3 = \lambda^2 - x_1 - x_2 \pmod p$

$y_3 = \lambda(x_1 - x_3) - y_1 \pmod p$

$\lambda = \dfrac{y_2 - y_1}{x_2 - x_1} \pmod p$

◎ if $P = Q$ then

$x_3 = \lambda^2 - x - x \pmod p$

$y_3 = \lambda(x - x_3) - y \pmod p$

$\lambda = \dfrac{3x^2 + a}{2y} \pmod p$

- $2P = P + P \rightarrow P = Q$ formula's will be used.
- $3P = 2P + P \rightarrow P \neq Q$ formula's will be used.
- $4P = 2P + 2P \rightarrow P = Q$ formula's will be used.
- $4P = 3P + P \rightarrow P \neq Q$ formula's will be used.

Q1. Find 3P, if P = (4,6) on $E_{11}(1,1)$.

Ans :- We know $2P + P = 3P$

Let us first find out 2P.

$2P = (6,6)$

$P = (4,6)$

Since $P \neq Q$

$x_3 = \lambda^2 - x_1 - x_2 \pmod p$

$y_3 = \lambda(x_1 - x_3) - y_1 \pmod p$

$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod p$

$\lambda = \frac{6-6}{4-6} \pmod{11}$

$\lambda = 0 \pmod{11} = 0$

$\lambda = 0$

$x_3 = \lambda^2 - x_1 - x_2 \pmod p$

$x_3 = 0^2 - 6 - 4 \pmod{11}$

$x_3 = 1$

$y_3 = \lambda(x_1 - x_3) - y_1 \pmod p$

$y_3 = 0(6-1) - 6 \pmod{11}$

$y_3 = -6 \pmod{11}$

$y_3 = 5$

$R = (2P+P) = 3P = (x_3, y_3) = (1,5)$

Lets check if it lie on $E_{11}(1,1)$

Not checking for $4a^3 + 27b^2 \pmod p \neq 0$

Because it's true but you can check also.

$y^2 \equiv x^3 + ax + b \pmod p$

$y^2 \pmod p = x^3 + ax + b \pmod p$

$y^2 \pmod{11} = x^3 + x + 1 \pmod{11}$

$5^2 \pmod{11} = 1^3 + 1 + 1 \pmod{11}$

$25 \pmod{11} = 3 \pmod{11}$

$3 = 3$

(1,5) lie on $E_{11}(1,1)$ because L.H.S equal to R.H.S.

★ Properties of elliptic curve over finite fields

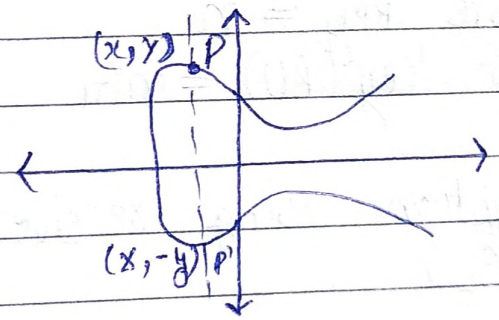
- Consider the elliptic curve E such that the points on the elliptic curve over a finite field F_p, $E(F_p)$, is a finite abelian group.

1. Closure :- if $P \in E(F_p)$ and $Q \in E(F_p)$, then $P+Q \in E(F_p)$.

2. Associativity :- if $P \in E(F_p)$, $Q \in E(F_p)$ and $R \in E(F_p)$, then $(P+Q)+R = P+(Q+R)$.

3. Identity :- If $P \in E(F_p)$, there exist an identity element which is the point at infinity O, such that $P+O = O+P = P$

4. Additive inverse :- If $P \in E(F_p)$, then every point P has an inverse $P' \in E(F_p)$ such that $P+P' = O$.



5. Commutative :- If $P \in E(F_p)$ and $Q \in E(F_p)$, then $P+Q = Q+P$.

★ Elliptic Curve Cryptography :-

- Steps :-

1. Encode the plaintext message as a point on the Elliptic Curve
2. Generate the Public and private keys
3. Perform encryption using receiver's public key
4. Decrypt the message using private key.

1. Encoding plaintext as point on Elliptic Curve :-
   It can be done by various methods you can learn more about it.

2. Generating Public and private key :-
   Let $G = (x, y) \in E_p(a, b)$
   Select Private key $= n$
   Compute Public key $(PU) = nG$

3. Perform encryption using receiver's public key :-
   $C = (C_1, C_2)$
   $C_1 = KG$
   K = any random number between 1 and $P-1$
   $\therefore 1 < K < P-1$
   $C_2 = P_M + KPU$

   $C = \{(KG, (P_M + KPU)\}$

4. Decrypt the message using private key
   $P_M = C_2 - (nC_1)$

Q1. Perform encryption and decryption using ECC on the Elliptic Curve $E_{11}(1,1)$ with the plaintext message $(4,6)$.

Sol:- Given data : $P = 11$, $a = 1$, $b = 1$ and $P_m = (4,6)$

- Plaintext on elliptic Curve $= (4,6)$
- Generating the Public and Private keys

  Let $G_r = (1,5) \in E_{11}(1,1)$

  Select Private Key $n = 2$

  Compute Public Key $(PU) = n\,G_r$

  Let's take $n = 2$

  ∴ The Public Key $(PU) = 2G_r$

  Compute $2G_r = 2(1,5)$

  Since $P = 8$,

$$x_3 = \lambda^2 - x - x \pmod p$$
$$y_3 = \lambda(x - x_3) - y \pmod p$$
$$\lambda = \frac{3x^2 + a}{2y} \pmod p$$
$$\lambda = \frac{3(1)^2 + 1}{2(5)} \pmod{11}$$
$$\lambda = 4 \times 10^{-1} \bmod 11$$

  Note:- Multiplicative Inverse to find $10^{-1} \bmod 11$

$$\lambda = 4 \times 10 \bmod 11 = 7$$
$$x_3 = \lambda^2 - x - x \pmod p$$
$$x_3 = 7^2 - 1 - 1 \pmod{11}$$
$$x_3 = 3$$
$$y_3 = \lambda(x - x_3) - y \pmod p$$
$$y_3 = 7(1-3) - 5 \pmod{11}$$
$$y_3 = -19 \pmod{11}$$
$$y_3 = 3$$

  Public Key $(PU) = (x_3, y_3)$

- Perform encryption using receiver's Public Key

$$C = \{kG_r, (M + kPU)\}$$

$C = (C_1, C_2)$

Let $K = 2$

$C = \{KG, (M + KPU)\}$

$C = \{2(1,5), ((4,6) + 2(3,3))\}$

$P = (3,3)$

$2P = P + P$

will use $P = Q$ formula's to find sum

$x_3 = \lambda^2 - x - x \pmod P$

$y_3 = \lambda(x - x_3) - y \pmod P$

$\lambda = \frac{3x^2 + a}{2y} \pmod P$

$\lambda = \frac{3(3)^2 + 1}{2(3)} \pmod{11}$

$\lambda = 14 \times 3^{-1} \pmod{11}$

$\lambda = 14 \times 4 \pmod{11}$

$\lambda = 1$

$X_3 = \lambda^2 - x - x \pmod P$

$X_3 = 1^2 - 3 - 3 \pmod{11}$

$X_3 = -5 \bmod 11$

$X_3 = 6$

$y_3 = \lambda(x - x_3) - y \pmod{11}$

$Y_3 = 1(3 - 6) - 3 \pmod{11}$

$Y_3 = 5$

$2(3,3) = (6,5)$

$C = \{2(1,5), ((4,6) + (6,5))\}$

$(4,6) + (6,5)$:

$X_3 = \lambda^2 - x - x_2 \pmod P$

$Y_3 = \lambda(x_1 - x_3) - y_1 \pmod P$

$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod P$

$$\lambda = \frac{5-6}{6-4} \pmod{11}$$

$$= \frac{-1}{2} \pmod{11}$$

$$= -1 \times 2^{-1} \pmod{11}$$

$$= -1 \times 6 \pmod{11}$$

$$= -6 \pmod{11}$$

$$= 5$$

$$x_3 = 5^2 - 4 - 6 \pmod{11}$$

$$= 15 \bmod 11$$

$$= 4$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \pmod{p}$$

$$y_3 = 5(4-4) - 6 \pmod{11}$$

$$= -6 \pmod{11}$$

$$= 5$$

$$(4,6) + (6,5) = (4,5)$$

$$C = \{2(1,5), (4,5)\}$$

$$P = (1,5)$$

$$2P = P + P$$

$2P = (3,3)$, using $P = Q$ formula's to find sum.

$$C = \{(3,3), (4,5)\}$$

- Decrypt the message using private key

$$M = C_2 - (nC_1)$$

$$C = (C_1, C_2)$$

$$C = \{(3,3), (4,5)\}$$

$$C_1 = (3,5)$$

$$C_2 = (4,5)$$

$$M = (4,5) - 2(3,3)$$

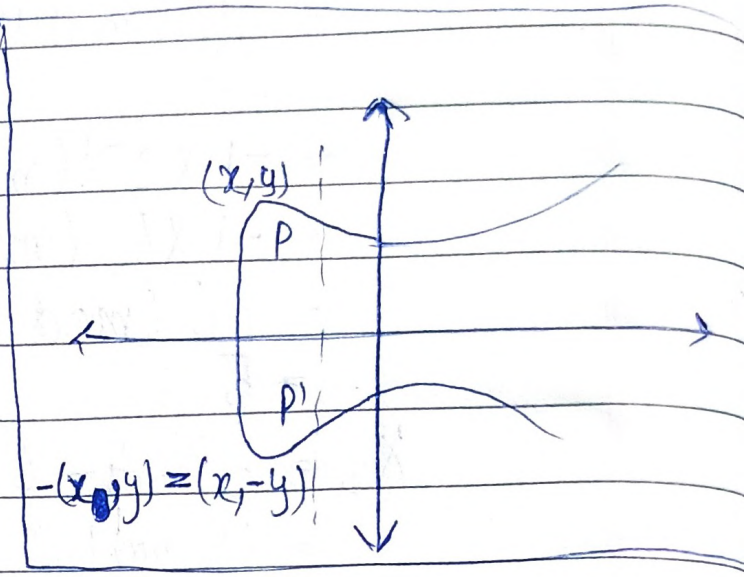
$$P = (3,3)$$

$2P = P + P$

$2P = (6, 5)$, using $P = Q$ formula's to find Sum.

$M = (4, 5) - (6, 5)$

$M = (4, 5) + (6, -5)$



$X_3 = \lambda^2 - x_1 - x_2 \pmod P$

$Y_3 = \lambda(x_1 - x_3) - y_1 \pmod P$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod P$$

$$= \frac{6-5}{6-4} \pmod{11} = \frac{1}{2} \pmod{11} = 2^{-1} \bmod 11 = 6$$

$X_3 = \lambda^2 - x_1 - x_2 \pmod P$

$X_3 = 6^2 - 4 - 6 \pmod{11}$

$X_3 = 26 \pmod{11}$

$X_3 = 4$

$Y_3 = \lambda(x_1 - x_3) - y_1 \pmod P$

$Y_3 = 6(4 - 4) - 5 \pmod{11}$

$Y_3 = -5 \pmod{11}$

$Y_3 = 6$

$(4, 5) + (6, 6) = (4, 6)$

$M = (4, 6)$

M is equal to Encrypted Message. Done :)

Elliptic Curve Diffie-Hellman (ECDH):-

* Algorithm :-

- Global Public ~~ele~~ element :-

  $E_q(a,b)$ — Elliptic Curve

  q — Prime Number or 2^m $(2^3=8)$

  G_1 — Point on the elliptic curve whose order is Large value n.

- Global element will be known to everyone

Alice :	Bob :
• Key Generating by Alice • Select Private key n_A, $n_A < n$ • Calculate Public key P_A $P_A = n_A \times G_1$	• Key Generation by Bob. • Select Private key n_B, $n_B < n$ • Calculate Public Key P_B $P_B = n_B \times G_1$
• Calculation of shared secret key by Alice. $K = n_A \times P_B$	• Calculate of shared secret key by Bob. $K = n_B \times P_A$

- Static Key : either Alice or Bob Key should be static meaning Key (Public Key) should be reused. Authenticated also.
- Ephemeral Key :- keys should not be used in future and keys will be unauthenticated
- P_A and P_B Can be static or Ephemeral key meaning P_A and P_B Can follow one of way i.e static or Ephemeral.

Q1. Find the shared secret Key for the elliptic Curve $E_{11}(1,1)$ with a point on the Curve $(4,6)$ and the private Keys of users A and B are 2 and 4 respectively.

Ans:- Given data :-

$P = 11$

$a = 1$

$b = 1$

$G_1 = (4, 6)$

$n_A = 2$

$n_B = 4$

Global Public elements :-

$E_{11}(1, 1)$ – Elliptic Curve

$G_1 = (4, 6) \in E_{11}(1, 1)$

Alice :-

$n_A = 2$

$P_A = 2(4, 6) = (6, 6)$ → $P = 0$ formula's used to do Sum

Bob :-

$n_B = 4$

$P_B = 4(4, 6) = (0, 10)$ → find Sum 4 time using $P = 0$

Alice :-

$K = n_A \times P_B$

$K = 2 \times (0, 10)$ → find Sum 2 time using $P = 0$

$K = (3, 8)$

Bob :-

$K = n_B \times P_A$

$K = 4 \times (6, 6)$ → find Sum 4 time using $P = 0$

$K = (3, 8)$

Secret shared $K = (3, 8)$ Generated.

Advantages of ECC over RSA :-

1. ECC takes one sixth of the computational effort.
2. Less storage, less power, less memory and less bandwidth.
3. Same level of security that we get with RSA 1024-bits.
4. Very fast Key Generation.
5. ECC is 15 times faster than RSA
6. Perfect forward secrecy.
7. Smaller Key size enables ECCs to be implemented on Smartcards, Contactless and WSN.
8. Both stand-Alone mode and Hybrid mode (RSA + ECC) of encryption.
9. Web applications.
10. Popular Blockchain such as bitcoin and ethereum used ECDSA for Signing transactions.

Weaknesses of ECC :-

1. Browsers support using ECC certificates.
2. ECC Signature verification is a Compute intensive task and it can be slower than RSA on devices with slower processors.
3. Many patents (especially for binary search), Creating some risk and Uncertainty.

Strengths and weaknesses of RSA :-

Strengths
- Easy to implement and understand than ECC.
- Signing and decryption are similar; encryption and verification are similar.
- Widely deployed, better industry support.

Weaknesses
- Very slow Key Generation.
- Slow Signing and decryption, which are slightly tricky to implement securely.
- The key is vulnerable to mathematical and timing attacks if poorly implemented.

ElGamal Cryptosystem :-

- Public key or Asymmetric Cryptosystem.
- Based on Deffie-Hellman key exchange.
- GnuPG and other Cryptosystem.
- DSA is a variant of ElGamal Signature Scheme.

Algorithm :-

- Global Public elements :-
  - q — Prime number.
  - α — Primitive root of q and $\alpha < q$.

- Alice :-
  - Key generation by Alice
  - Select Private key $= X_A < q-1$
  - Calculate $Y_A = \alpha^{X_A} \bmod q$
  - Public key $= \{q, \alpha, Y_A\}$

- Bob :-
  - Encryption by Bob using Alice's Public key.
  - Plaintext $= M < q$
  - Select random integer $= K < q$
  - Calculate $K = Y_A^K \bmod q$
  - Calculate $C_1 = \alpha^K \bmod q$
  - Calculate $C_2 = K \times M \bmod q$
  - Ciphertext $= (C_1, C_2)$.

- Alice :-
  - Decryption by Alice using Alice's Private key.
  - Ciphertext $= (C_1, C_2)$
  - Calculate $K = C_1^{X_A} \pmod q$
  - Plaintext: $M = (C_2 K^{-1}) \bmod q$

Example :-

(i) Global Public elements

$q = 19$

$\alpha = 10$ and $10 < 19$.

Alice :-

Key Generation by Alice

Select Private Key $= X_A = 5 < 18$

Calculate $Y_A = 10^5 \bmod 19$

$= 3$

Public Key $= \{19, 10, 3\}$

Bob :-

Encryption by Bob using Alice's Public Key

Plaintext $= M = 17 < 19$

Select random integer $= K = 6 < 19$

Calculate $K = 3^6 \bmod 19 = 729 \bmod 19 = 7$

Calculate $C_1 = 10^6 \bmod 19 = 11$

Calculate $C_2 = 7 \times 17 \bmod 19 = 5$

Ciphertext $= (11, 5)$

Alice :-

Decryption by Alice using Alice's private key

Ciphertext $= (11, 5)$

Calculate $K = 11^5 \pmod{19} = 7$ and $K^{-1} = 11$

Plaintext : $M = (5 \times 11) \bmod 19 = 17 = 17$

Quadratic Residue modulo n :-

Let a and $n \in \mathbb{Z}$ with $n > 0$ and $GCD(a, n) = 1$. a is said to be a Quadratic residue modulo n, if $x^2 \equiv a \pmod{n}$ is solvable. Otherwise, it is a Quadratic non-residue.

Example :-

(i) Find the Quadratic residues mod 5.

Soln :-

	$x^2 \bmod 5$
1	1
2	4
3	4
4	1

The Quadratic residues are 1 and 4.
The Quadratic non-residues are 2 and 3.

(ii) Find the Quadratic residues for Z_{11}.

	$x^2 \bmod 11$
1	1
2	4
3	9
4	5
5	3
6	3
7	5
8	9
9	4
10	1

The quadratic residues mod 11 are 1, 3, 4, 5, 9.
The quadratic non-residues mod 11 are 2, 6, 7, 8 and 10.

Rabin Cryptosystem :-

- Public key or Asymmetric Cryptosystem.
- Related to the difficulty of integer factorization
- Mathematical proven to be Computationally secure against a chosen-plaintext attack.

Algorithm.

- Select prime numbers p, q such that $p, q \equiv 3 \pmod 4$.
- Calculate $n = p \times q$
- Public key $PU = \{n\}$
- Private key $PR = \{p, q\}$
- Encryption using Public key
  $C \equiv m^2 \bmod n$ (Not injective)
- Decryption using private key
  $x \equiv c^{\frac{1}{4}(p+1)} \bmod p$
  $s \equiv c^{\frac{1}{4}(q+1)} \bmod q$
  $m_1 \equiv x \cdot p \cdot s + y \cdot q \cdot x \pmod n$
  $m_2 \equiv n - m_1$
  $m_3 \equiv x \cdot p \cdot s - y \cdot q \cdot x \pmod n$
  $m_4 \equiv n - m_3$

NoTe:- Receiver Cannot determine which is the original message without further information.

Not injective :- one domain have many range.

Q1. Alice selects two prime numbers p and q. Let $p = 19$ and $q = 31$ such that $p \equiv 3 \pmod 4$ and $q \equiv 3 \pmod 4$. Suppose Bob wants to encrypt Capital letter Y and send it to Alice. Perform encryption and decryption on these.

Ans:- Given data:
$p = 19$, $q = 31$, and $m = 89$.

Calculate $n = p \times q$

$= 19 \times 31$

$= 589$

Public Key $PU = \{589\}$

Private Key $PR = \{19, 31\}$

Bob generates the Ciphertext

$C \equiv m^2 \bmod n$

$C \equiv 89^2 \bmod 589$

$C \equiv 264 \bmod 589$

Bob sends this to Alice.

Alice receives $C \equiv 264 \bmod 589$

Alice needs to solve x and s.

Alice solves x and s

$x \equiv C^{\frac{1}{4}(p+1)} \bmod p$

$x \equiv 264^{\frac{1}{4}(19+1)} \bmod 19$

$x \equiv 6 \bmod 19$

$s \equiv C^{\frac{1}{4}(q+1)} \bmod q$

$s \equiv 264^{\frac{1}{4}(31+1)} \bmod 31$

$s \equiv 14 \bmod 31$

Alice then needs to find the value of x and y.

$p \cdot x + q \cdot y = 1$

Linear Diophantin equation → Solved by extended euclidean

$19x + 31y = 1$

↳ if $p \cdot x + q \cdot y = GCD(p, q)$

$19x + 31y = GCD(31, 19)$

$31 = 19(1) + 12 \quad -(1)$

$19 = 12(1) + 7 \quad -(2)$

$12 = 7(1) + 5 \quad -(3)$

$7 = 5(1) + 2 \quad -(4)$

$5 = 2(2) + 1$ — (5)

Start with last equation,
Sub previous and evaluate.

$5 = 2(2) + 1$

$1 = 5 - 2(2)$

Substitute (4)

$1 = 5 - 2[7 - 5(1)]$

$1 = 5 - 7(2) + 5(2)$

$1 = 5(3) - 7(2)$

Substitute (3)

$1 = 5(3) - 7(2)$

$1 = [12 - 7(1)](3) - 7(2)$

$1 = 12(3) - 7(3) - 7(2)$

$1 = 12(3) - 7(5)$

Substitute (2)

$1 = 12(3) - 7(5)$

$1 = 12(3) - [19 - 12(1)](5)$

$1 = 12(3) - 19(5) + 12(5)$

$1 = 12(8) - 19(5)$

Substitute (1)

$1 = 12(8) - 19(5)$

$1 = [31 - 19(1)](8) - 19(5)$

$1 = 31(8) - 19(8) - 19(5)$

$1 = 31(8) - 19(17)$

$1 = 31(8) + 19(-13)$ — (6)

$1 = 31y + 19x$ — (7)

from 6 and 7

$x = -13$

$y = 8$

$m_1 \equiv x \cdot P \cdot S + y \cdot Q \cdot R \pmod{n}$

$m_1 \equiv -13 \times 19 \times 4 + 8 \times 31 \times 6 \bmod 589 \equiv 500 \bmod 189$

$m_2 \equiv n - m_1$

$m_2 \equiv 589 - 500 \equiv 89$

$m_3 \equiv -13 \times 19 \times 4 - 8 \times 31 \times 6 \bmod 589$

$m_3 \equiv 469 \bmod 589$

$m_4 \equiv 589 - 469 \equiv 120$

Decryption answers is 500, 89, 469 and 120 but actual ans is 89 because it looks plaintext.

Knapsack Cryptosystem :-

- Public Key Cryptosystem
- Based on Subset Sum problem
- A special Case of the Knapsack problem
- Attack by Ada Shamir

★ Algorithm :-

1. Choose a super-increasing sequence $(a_1, a_2, a_3 \ldots, a_n)$.
   Super-increasing : $a_n > a_{n-1} + a_{n-2}$
2. Choose a modulus $M > a_1 + a_2 + a_3 + \ldots + a_n$.
3. Choose a multiplier W such that $1 < W < M-1$ and $GCD(W, M) = 1$
4. Compute $b_i = W \times a_i \pmod M$.

Public Key : $[b_1, b_2, b_3, \ldots, b_n]$
Private Key : $[M, W, (a_1, a_2, a_3 \ldots, a_n)]$

Encryption :-

Message $m \in \{0,1\}^n = m_1 m_2 m_3 \ldots m_n$.
Ciphertext $C = \sum_{j=1}^{n} m_i b_i$ → Note! a, b, m length should be equal

Decryption :-

Ciphertext $C = \sum_{j=1}^{n} m_i b_i$

$S = C \times W^{-1} \pmod M$.

Solve using $(a_1, a_2 \ldots a_n)$ and s.

Example :-

Q1. Encrypt [01101] using Knapsack Cryptosystem

Ans1. • Choose a super-increasing sequence (2, 3, 7, 15, 31)
• Choose a modulus $M = 61 > 2+3+7+15+31$.

• Choose a multiplier $\omega = 17$ such that $1 < 17 < 60$ and $GCD(17, 61) = 1$

• Compute $b_i = \omega \times a_i \pmod M$.

a_i	$\omega \times a_i$	$b_i = \omega \times a_i \pmod M$
2	34	34
3	51	51
7	119	58
15	255	11
31	527	39

Public Key : [34, 51, 58, 11, 39]
Private Key : [61, 17, (2, 3, 7, 15, 31)]

• Encryption :-

Message $m = [0\ 1\ 1\ 0\ 1]$

$b_i = [34, 51, 58, 11, 39]$

0	1	1	0	1
34	51	58	11	39

$C = 51 + 58 + 39$

$C = 148$

Decryption

$S = C \times \omega^{-1} \pmod M$

$S = 148 \times 18 \pmod{61} \equiv 41$

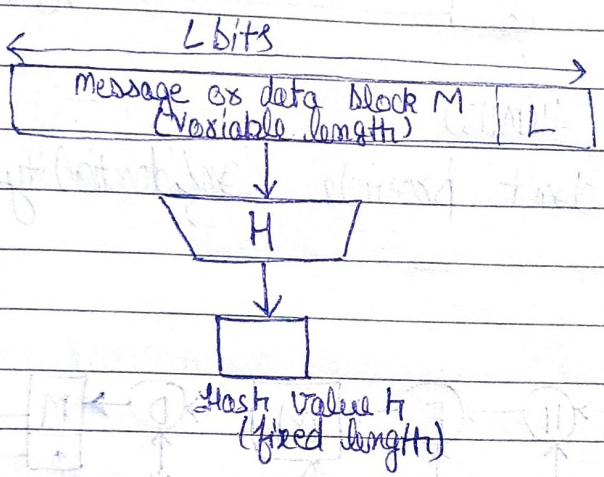
i	a_i	S	$S \geq a_i$	m
5	31	41	True	1
4	15	41 − 31 = 10	False	0
3	7	10	True	1
2	3	10 − 7 = 3	True	1
1	2	0 = 3 − 3	False	0

$m = [0\ 1\ 1\ 0\ 1]$

DONE :)

Cryptographic Hash function :-

- Hash Function H.
- $h = H(M)$.
- Evenly distributed and apparently random.
- Avalanche effect should be maximum.
- Computationally efficiently :- Function should not take more time to generate Hash.
- Computationally infeasible :- It should not be possible to generate message from Hash
- One-way property.
- Collision free property.

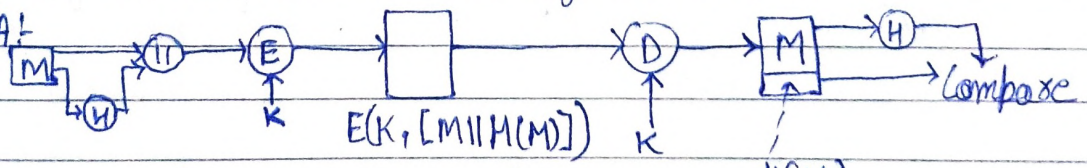


☆ Applications of Cryptographic Hash function :-

1. Message Authentication.
2. Digital Signatures.
3. Other Applications

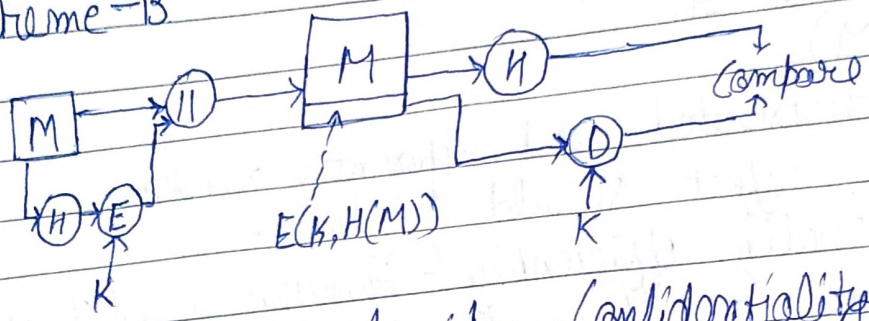
1 Message Authentication :- Private key (Cryptographic key) is used.

(a) Scheme-A :-



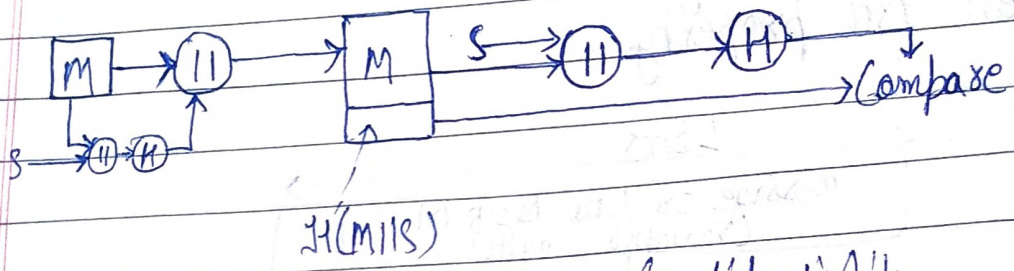
It provides Confidentiality and Authentication.
It also provides integrity.

(b) Scheme-B



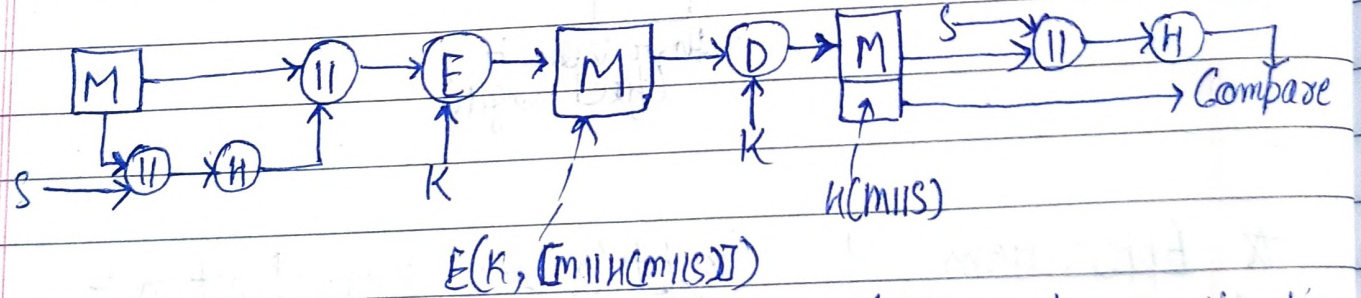
This scheme don't provide Confidentiality.

(c) Scheme-C



It also don't provide Confidentiality.

(D) Scheme-D



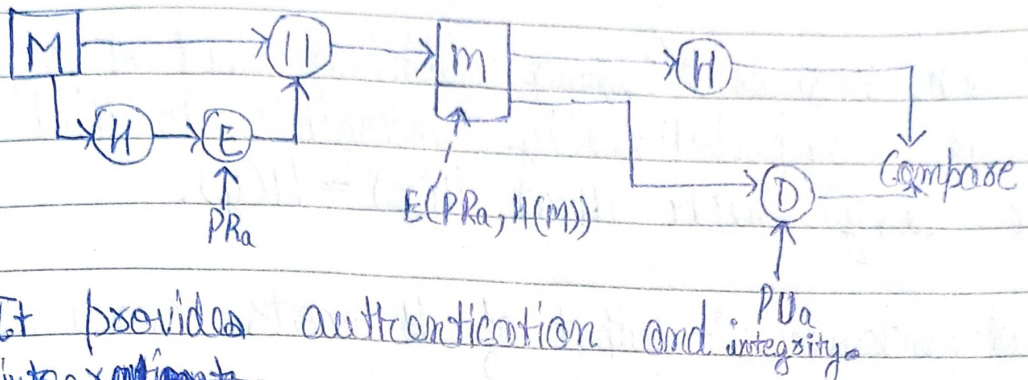
We achieve both using this Confidentiality and authentication and this also provide integrity.

- Main intention is to know that data was modified or not but different scheme provides different advantage like Confidentiality, integrity, authenti etc.

2. Digital Signatures :-

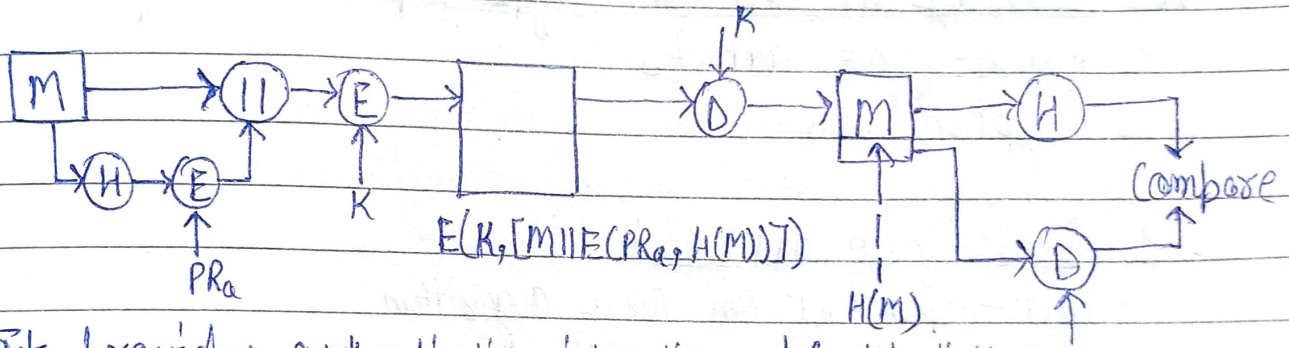
- Public key cryptography is used.

(a) Case - A



- It provides authentication and integrity.

(b) Case - B



- It provides authentication, integration and Confidentiality.

3. Other Applications :-

(i) One-way password file.
(ii) Intrusion detection.
(iii) Virus detection.
(iv) Pseudorandom Function (PRF).
(v) Pseudorandom Random Number Generation (PRNG).

★ Security Requirements :-

(a) variable input size.
(b) Fixed output size
(c) Efficiency
(d) Preimage resistant (One way property) :- For any given hash value it is computationally infeasible to find y such that $H(y) = h$.

(e) Second preimage resistant (weak collision resistant property):
For any given block, it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$

(f) Collision resistant (Strong collision resistant property):-
It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.

(G) Pseudorandomness:- Output of H meets standard tests for pseudorandomness.

★ Security Attacks on Cryptographic Hash function:-
1. Brute-force attacks
2. Cryptanalysis

1. Brute-force attacks:-
- Not-dependent on any algorithm.
- Depends only on bit length.
- Preimage and second preimage resistant attacks.

(i) Preimage resistant attacks:- An adversary wishes to find a value such that $H(y)$ is equal to a given hash value.

(ii) Second preimage resistant:- An adversary wishes to find two messages or data blocks x and y, that yield the same hash function: $H(x) = H(y)$.
- Birthday paradox:- The birthday paradox, also known as the birthday problem, states that in a random group of 23 people, there is about a 50 percent chance that two people have the same birthday.

2. Cryptanalysis :-

- Cryptanalytic attacks on hash functions.
- No exhaustive search.
- An ideal hash algorithm – Cryptanalytic effort >= Brute force.
- A typical secure hash function
- Compression function can be used to find patterns.

SHA (Secure Hash Algorithm):

- Developed by NIST.

★ SHA Parameters :-

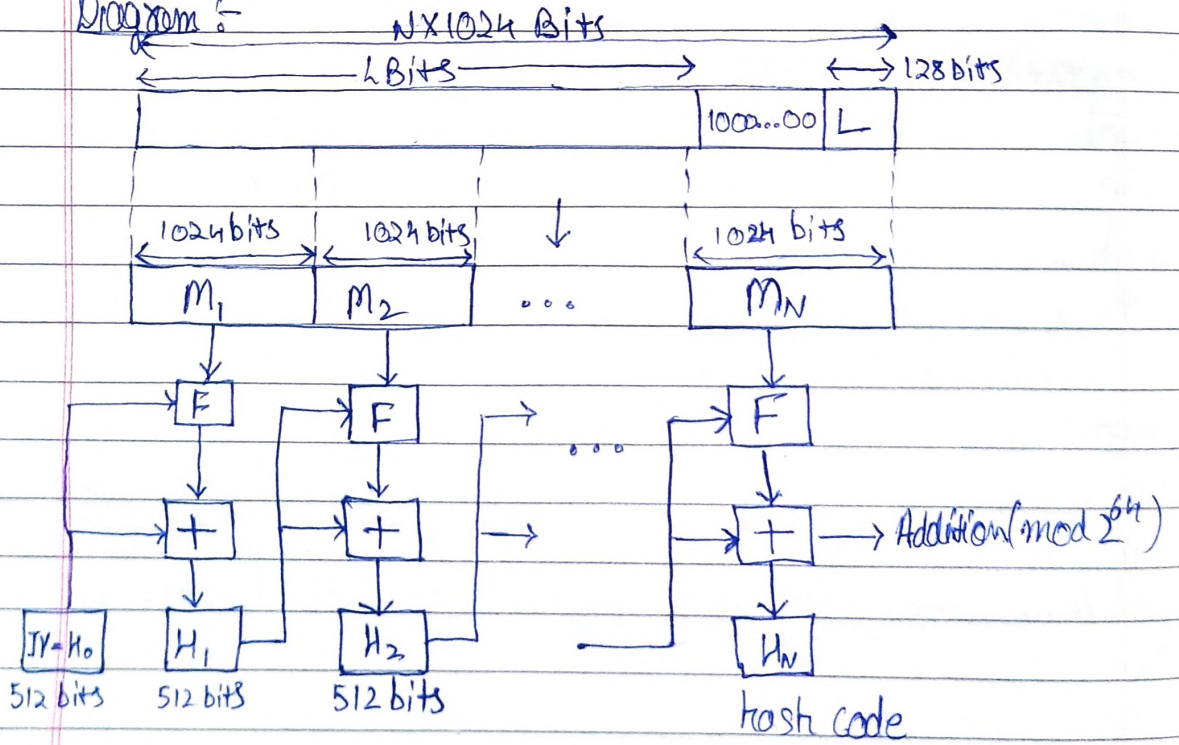
	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message Digest	160	224	256	384	512
Message Size	$<2^{64}$	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$
Block Size	512	512	512	1024	1024
Word Size	32	32	32	64	64
Number of Steps	80	64	64	80	80

★ We take SHA-512 to learn how SHA works.

★ SHA-512 Message Digest Generation :-

1. Appending padding bits (to make total length N X 1024 bits)
2. Append length
3. Initialize Hash buffer.
4. Processing the message in 1024-bits (128 word) blocks.
5. Output.

Diagram :-



1. Step-1 :- Appending padding bits :-
- Message padding.
- Length $\equiv$ 896 (mod 1024)
- Padding is always added.
- Number of padding bits is in the range of 1 to 1024.
- The padding - Single 1 bits followed by the necessary number of 0 bits.

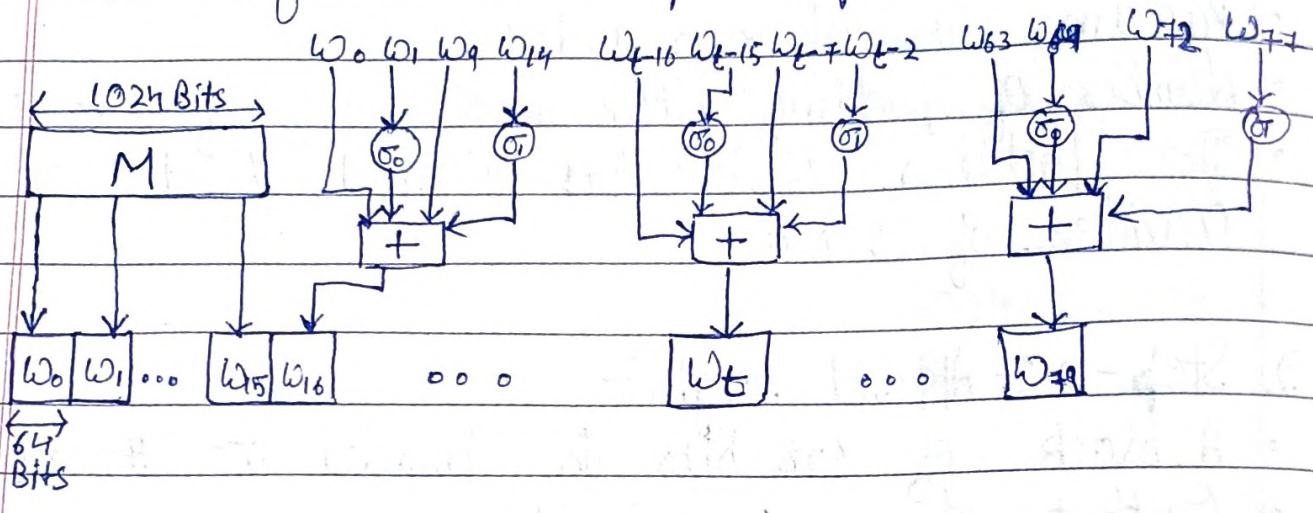
2. Step-2 :- Append length :-
- A block of 128 bits is appended to the message
- Output of step 1 and step2
- Message that is an integer multiple of 1024 bits in length.
- Expanded message = Sequence of 1024-bit blocks $M_1, M_2 \dots M_N$

3. Step3 :- Initialize hash buffer :-
- A 512-bit buffer is used to hold intermediate and final results of the hash function
- Buffers = 64 bit registers named a, b, c, d, e, f, g, and h.
- Values of registers :-

a = 6A09E667F3BCC908 e = 510E527FADE682D1
b = BB67E8584CAA73B f = 9B05688C2B3E6C1F
c = 3C6EF372FE94FB2B g = 1F83D9ABFB41BD6B
d = A54FF53A5F1D36F1 h = 5BE0CD19137E2179

4. Step-4 :- Processing the message in 1024-Bit (128 word) blocks

- Creation of 80-word input sequence



$$W_t = \sigma_1^{512}(W_{t-2}) + W_{t-7} + \sigma_0^{512}(W_{t-15}) + W_{t-16}$$

$\boxed{+}$ = addition (modulo 2^{64})

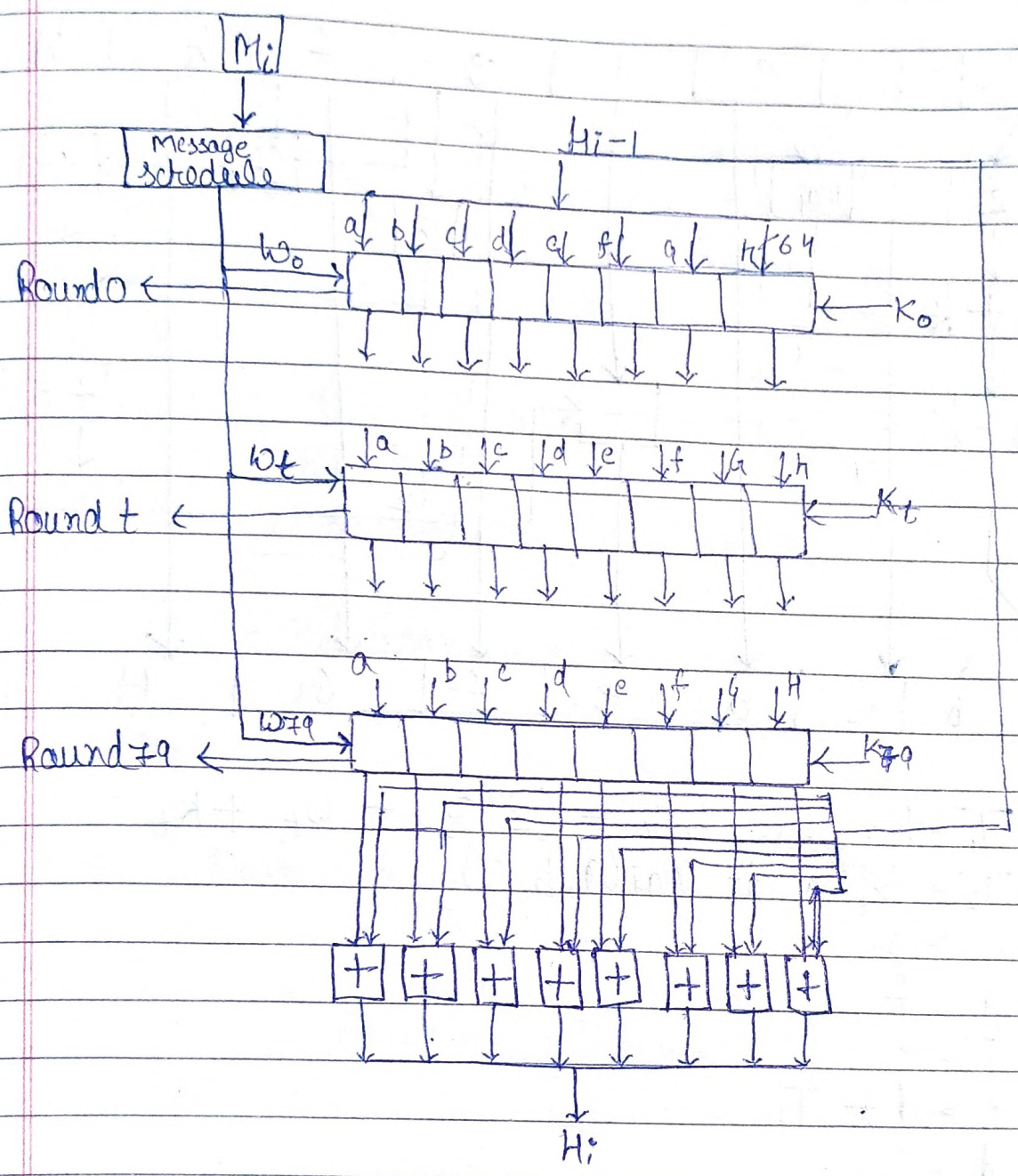
$$\sigma_0^{512}(x) = ROTR^1(x) \oplus ROTR^8(x) \oplus SHR^7(x)$$

$$\sigma_1^{512}(x) = ROTR^{19}(x) \oplus ROTR^{61}(x) \oplus SHR^6(x)$$

$ROTR^n(x)$ = Circular right shift - argument x by n bits

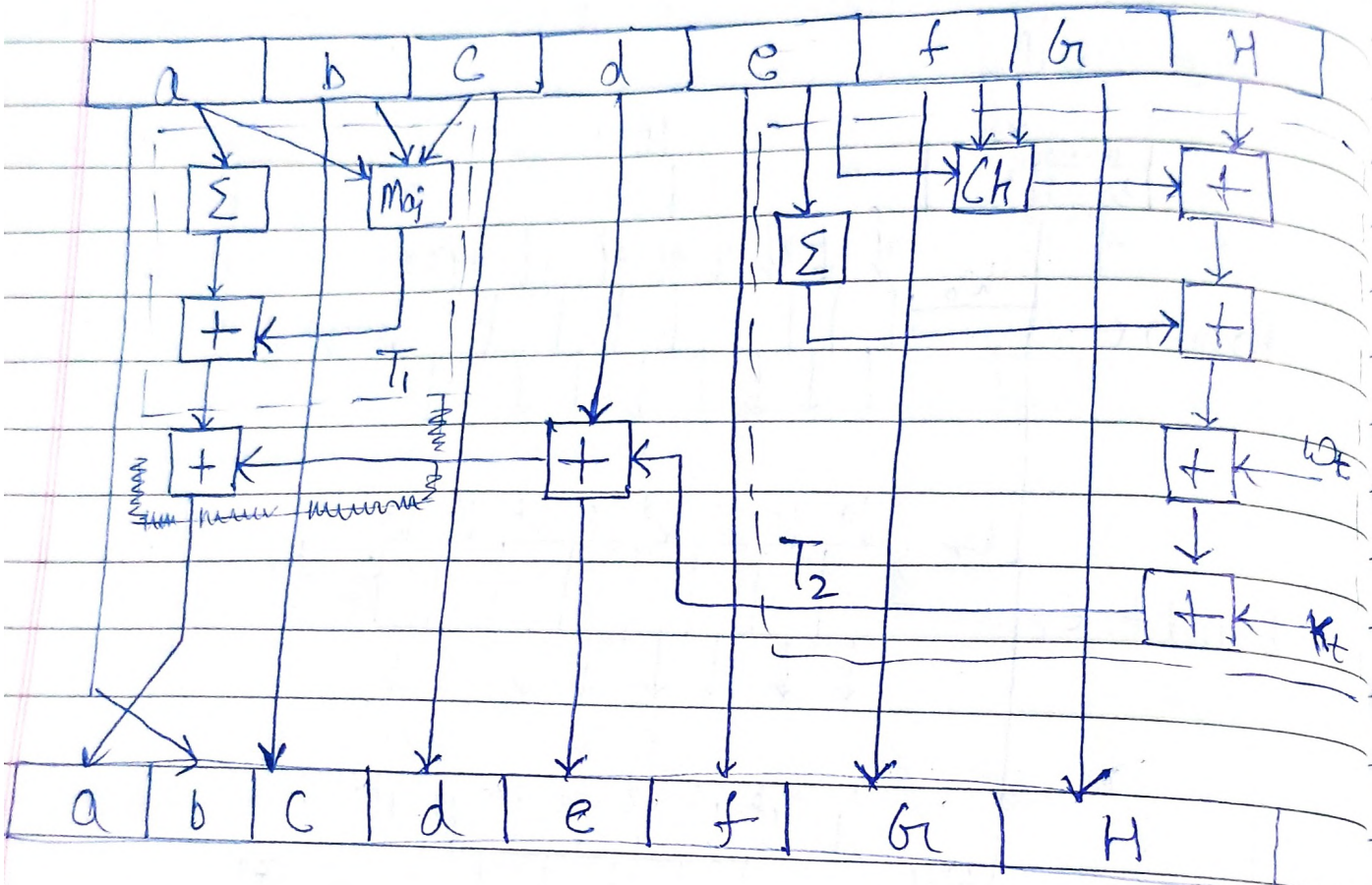
$SHR^n(x)$ = Left shift - argument x by n bits

- Processing the message in 1024-bit (128 word) blocks:-



NoTe :- For Constants $K_0 - K_{79}$ you Can Search them On internet with SHA-512 Constants

- SHA-512 Round function



$T_1 = h + Ch(e, f, Gr) + (\Sigma_1^{512} e) + W_t + K_t$

$T_2 = (\Sigma_1^{512} a) + Maj(a, b, c)$

$h = g$

$g = f$

$f = e$

$e = d + T_1$

$d = c$

$c = b$

$b = a$

$a = T_1 + T_2$

DELTA

- Notations

$Maj(a,b,c) = (a \text{ AND } b) \oplus (a \text{ AND } c) \oplus (b \text{ AND } c)$

$\Sigma_0^{512} a = ROTR^{28}(a) \oplus ROTR^{34}(a) \oplus ROTR^{39}(a)$

$\Sigma_1^{512} e = ROTR^{14}(e) \oplus ROTR^{18}(e) \oplus ROTR^{41}(e)$

$ROTR^n(x) =$ Circular right shift - argument x by n bits

$W_t =$ a 64-bit word derived from the current 512-bit input block

$K_t =$ a 64-bit additive constant

$\boxplus =$ addition modulo 2^{64}.

$Ch(e,f,g) = (e \text{ AND } f) \oplus (NOT\ e \text{ AND } g)$

5. Step 5 :- Output :-

$H_0 = IV$

$H_i = SUM_{64}(H_{i-1}, abcdefgh_i)$

$MD = H_N$

Message Authentication Codes :-

- Cryptographic Checksum.
- T = MAC(K, M).
- Message authentication mechanism.
- To verify the integrity of a message.
- Symmetric encryption.
- MAC - Uses a secret key.

★ Requirement for MAC :-

1. Disclosure.
2. Traffic Analysis.
3. Masquerade
4. Content Modification.
5. Sequence Modification
6. Timing modification.
7. Source Repudiation
8. Destination Repudiation.

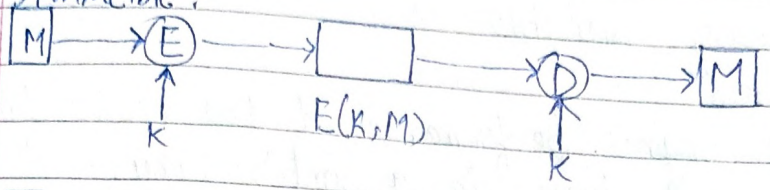
★ Message authentication functions :-

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

1 Hash function :- Hash function used to check message integrity because if something get change in message then it hash also get changed.

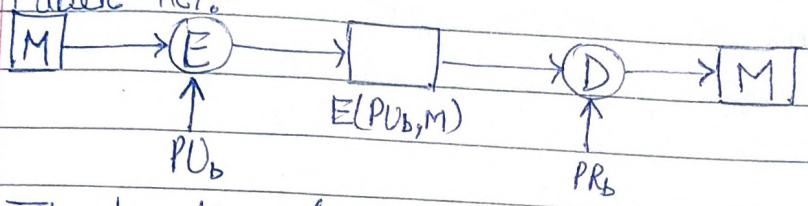
2. Message Encryption :-

(i) Symmetric :-



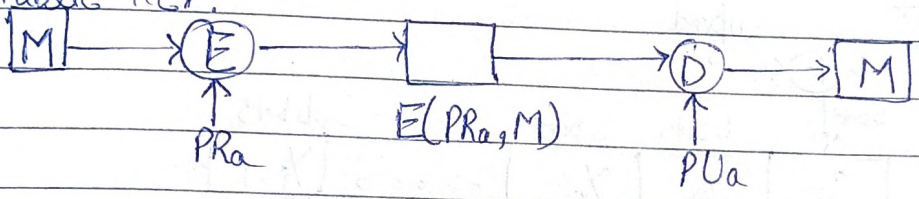
It provides Confidentiality and Authentication

(ii) Public-Key :-



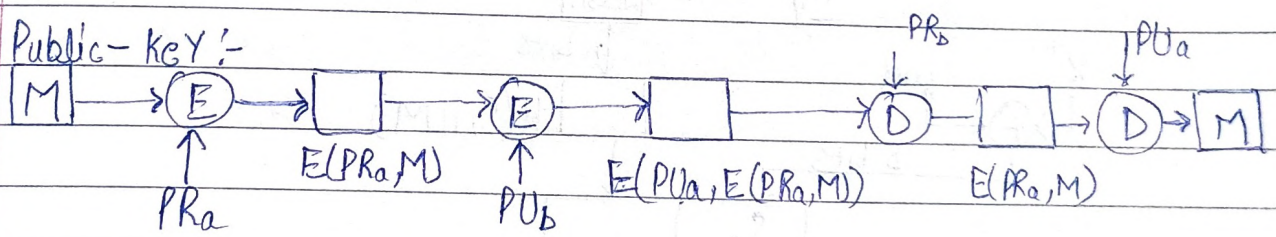
It provides Confidentiality.

(iii) Public-Key :-



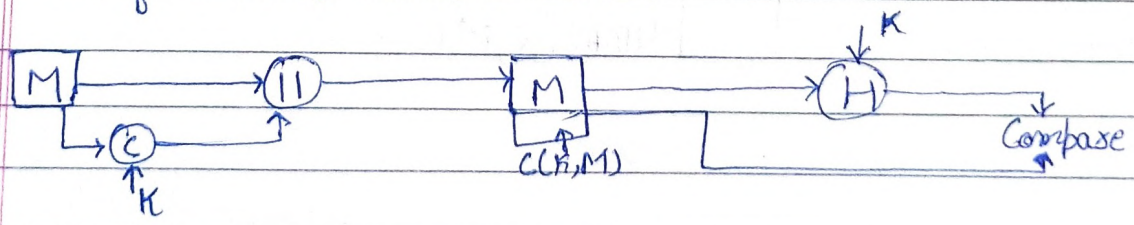
It provides Authentication and Signature.

(iv) Public-Key :-



It provides Confidentiality, Authentication and Signature

3. Message Authentication Codes :-



It provides authentication and Integrity.

* We can also achieve Confidentiality using Symmetric key encryption.

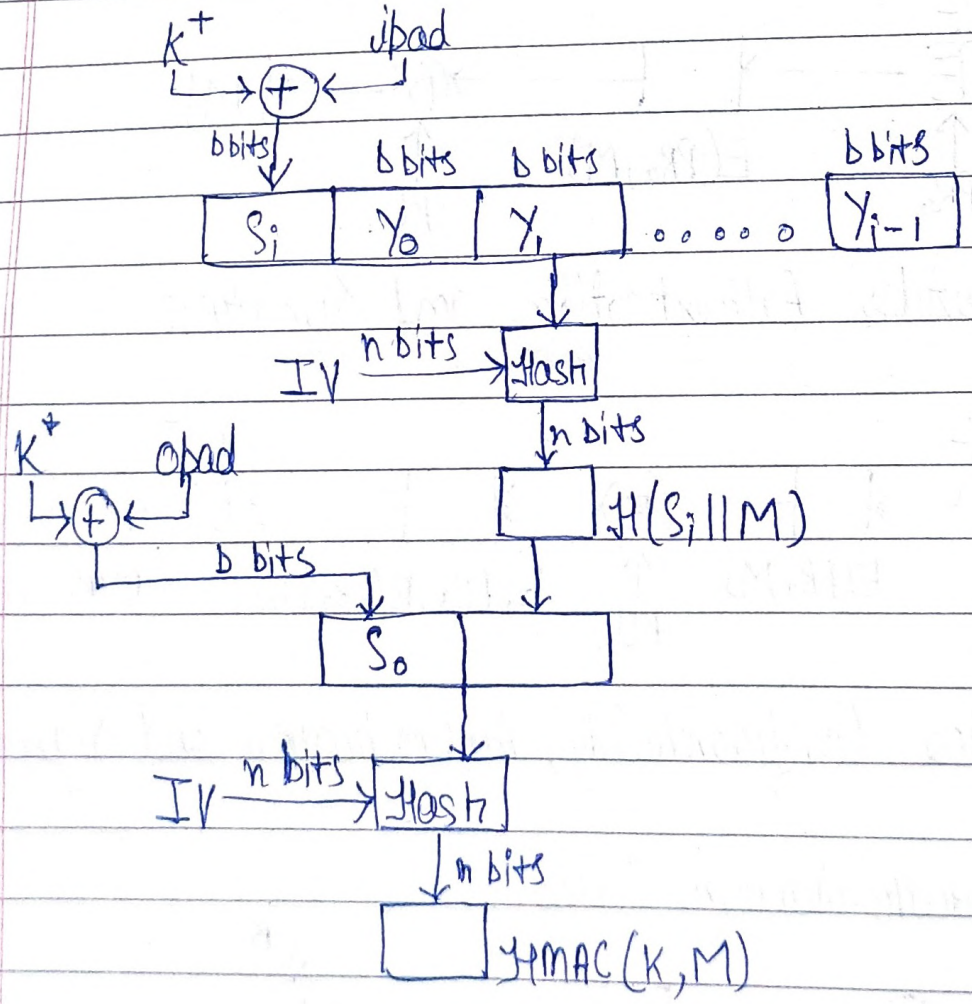
★ HMAC (Hash based MACs) :-

- To use hash functions that perform well in software.
- Freely and widely available.
- Secure and faster.
- To preserve the original performance of the hash function.
- To use and handle keys in a simple way.
- The strength of the authentication mechanism.

★ Two objectives:

1. Black box.
2. Secure.

★ HMAC structure and algorithm :-



- process :-

1. Append zeros to the left of K to create a b-bit string K^+.
2. Perform bitwise XOR K^+ with ipad ($36_H = 00110110_2$) to produce the b-bit block S_i.
3. Append M to S_i.
4. Apply H to the stream generated in step 3.
5. XOR K^+ with opad ($5C_H = 01011100_2$) to produce the b-bit block S_o.
6. Append the hash result from step 4 to S_o.
7. Apply H to the stream generated in step 6 and output the result.

DAA :-

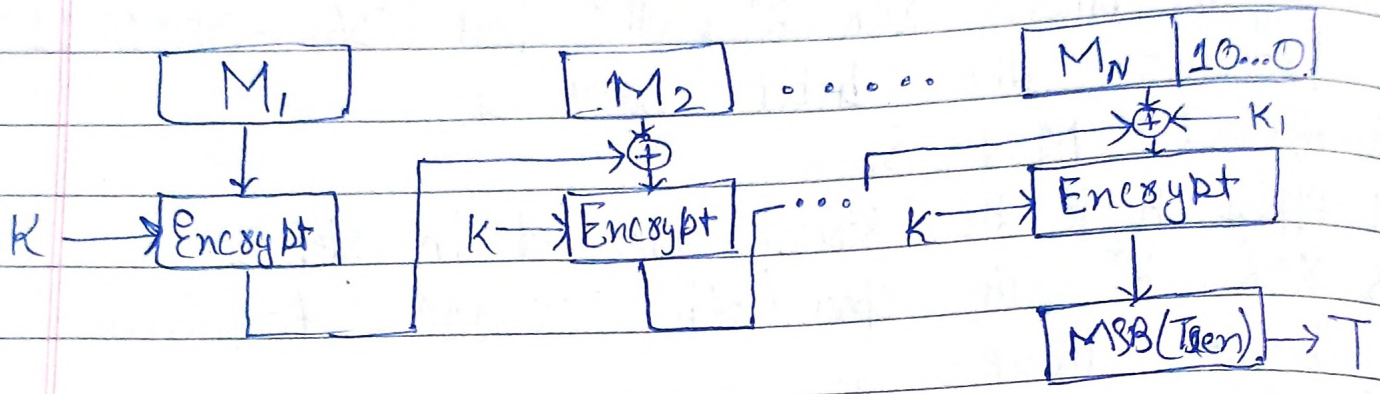
- Data authentication algorithm
- Based on DES
- Widely used
- Security weaknesses.
- CBC mode of operation of DES with I.V = 0.
- 64-Bit block.

DAA = DES algorithm for encryption and decryption with CBC block operation and Initialization vector is zero.

CMAC :-

- Cipher-based MAC
- CMAC mode of operation for use with AES and 3-DES.
- Message is an integer multiple 'n' of the cipher block length 'l'.
- $M = M_1, M_2, M_3 \ldots M_n$ (n blocks).
- K-bit encryption with Key 'K'.
- AES key size = 128 or 192 or 256 bits.
- 3-DES Key size = 112 or 168 bits.

★ CMAC Diagram :-



- 10...00 from ~~left~~ Right side if length of M_n is less than what encryption algo takes
- K_1 is n-bit constant

★ CMAC Process :-

- CMAC is calculated as follows.

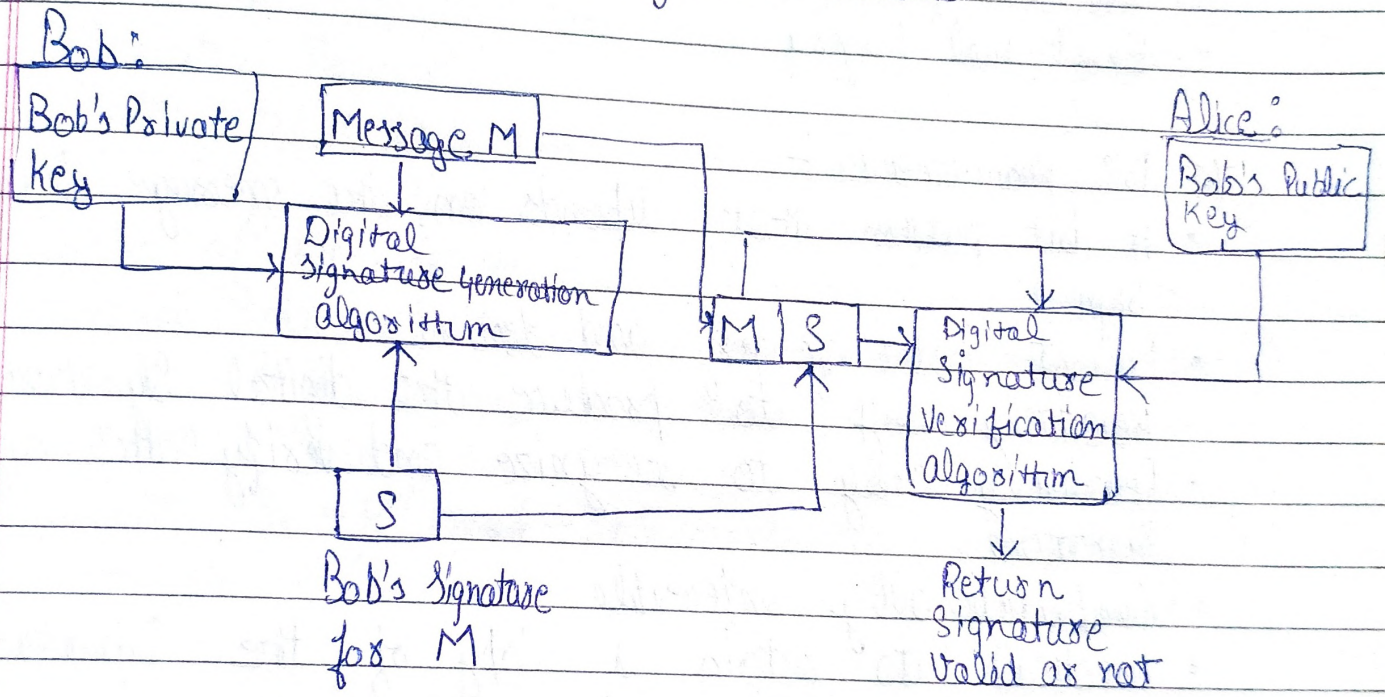
1. $C_1 = E(K, M_1)$
2. $C_2 = E(K, [M_2 \oplus C_1])$
3. $C_3 = E(K, [M_3 \oplus C_2])$

   ⋮
4. $C_n = E(K, M_n \oplus C_{n-1} \oplus K_1])$
5. $T = MSB_{Tlen}(C_n)$ → MSB bits token of Tag length.

Digital Signature :-

- Authentication mechanism.
- Asymmetric encryption used.
- Attach a Code that acts as a Signature.
- Signature = $E_{PR}[H(M)]$.
- Guarantees the source and integrity of the message.
- DSS.

★ Generic model of Digital Signature Process:-



★ Properties of Digital Signatures:-

1. It must verify the author and the date and time of the signature.
2. It must authenticate the contents of the time of the signature.
3. It must be verifiable by third parties, to resolve disputes.

★ Attacks and Forgeries :-

- Key-only attack.
- Known message attack.
- Generic chosen message attack
- Directed chosen message attack
- Adoptive chosen message attack
- Total break.
- Universal forgery.
- Selective forgery.
- Existential forgery.

★ DS Requirements :-

- A bit pattern that depends on the message being signed.
- Prevent both forgery and denial.
- Relatively easy to produce the digital signature.
- Relatively easy to recognize and verify the digital signature.
- Computationally infeasible.
- Practical to retain a copy of the signature in storage.

The ElGamal digital Signature :-

- Based on like Elgamal encryption

* Algorithm :-

- Global Public elements

1. q — Prime number
2. α — Primitive root of q

- Alice

1. Key Generation by Alice :-
2. Select X_A ; $1 < X_A < q-1$
3. Calculate $Y_A = \alpha^{X_A} \bmod q$
4. Private Key = X_A
5. Public Key = $\{q, \alpha, Y_A\}$
6. Signing of Message :-
7. Random integer K ; $1 \leq K \leq q-1$ and $GCD(K, q-1) = 1$
8. Compute $S_1 = \alpha^K \bmod q$; $(C_1 = S_1)$
9. Compute $K^{-1} \bmod (q-1)$
10. Compute $S_2 = K^{-1}(m - X_A S_1) \bmod (q-1)$ → $\boxed{m = H(M)}$

Signature = (S_1, S_2)

- Bob

1. Verifying a Signature :-
2. Compute $V_1 = \alpha^m \bmod q$
3. Compute $V_2 = (Y_A)^{S_1} (S_1)^{S_2} \bmod q$
4. The Signature is valid if $V_1 = V_2$

- How $V_1 = V_2$?

1. $\alpha^m \bmod q = (Y_A)^{S_1} (S_1)^{S_2} \bmod q$
2. $\alpha^m \bmod q = \alpha^{X_A S_1} \alpha^{K S_2} \bmod q$
3. $\alpha^{m - X_A S_1} \bmod q = \alpha^{K S_2} \bmod q$
4. $m - X_A S_1 \equiv K S_2 \bmod (q-1)$
5. $m - X_A S_1 \equiv K K^{-1}(m - X_A S_1) \bmod (q-1)$

Example :-

(i) Global elements :-

$q = 19$

$\alpha = 10$

- Alice :-

1. Key generation by Alice :-
2. Select $X_A = 16$, $1 < X_A < 18$
3. Calculate $Y_A = 10^{16} \bmod 19 = 4$
4. Private key $= X_A = 16$
5. Public key $= \{q, \alpha, Y_A\} = \{19, 10, 4\}$
6. Signing a message $m = 14 \rightarrow m = H(M) = 14$
7. Random integer $K = 5$, $1 \leq K \leq 18$ and $GCD(5, 18) = 1$
8. Compute $S_1 = 10^5 \bmod 19 = 3$
9. Compute $K^{-1} \bmod (q-1) = 5^{-1} \bmod 18 = 11$
10. Compute $S_2 = K^{-1}(m - X_A S_1) \bmod (q-1)$

$= -374 \bmod 17 = 4$

11. Signature $= (S_1, S_2) = (3, 4)$

- Bob :-

1. Verifying a signature :-
2. Compute $V_1 = \alpha^m \bmod q$

$= 10^{14} \bmod 19 = 16$

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

$= (4)^3 (3)^4 \bmod 19$

$= 16$

The signature is valid since $V_1 = V_2$

The Schnorr Digital Signature Scheme :-

- Based on discrete Logarithms
- Minimizes the message-dependent amount of computation
- Does not depend on message.
- Can be done during the idle time of the processor
- $P-1 \equiv (\text{mod } q)$
- $P \approx 2^{1024}$ and $q \approx 2^{160}$.

Process :-

1. Generation of private and public Keys :
2. Signature Generation
3. Signature verification.

1. Generation of private and public keys :-

- Choose prime P and q, such that q is a prime factor of P-1.
- Choose 'x' such that $x^q = 1 \bmod q$.
- Choose private key 's' such that $0 < s < q$.
- Choose Public key $v = x^{-s} \bmod P$.

2. Signature Generation :-

- Using 'v' and 's' users Generate Signature as follows:
- Choose a random integer 'r' such that $0 < r < q$.
- Compute 'x' such that $x = x^r \bmod P$.
- Compute 'e' such that $e = H(M \| x)$.
- Compute 'y' such that $y = (r + se) \bmod q$.
- Signature = (e, y).

3. Signature Verification :-

- Compute x' such that $x' = x^y v^e \bmod P$.
- Verify that $e = H(M \| x')$.

★ How verification works?

- $x' = \alpha^{y} v^{e}$
- $x' = \alpha^{y} \alpha^{-se}$
- $x' = \alpha^{y-se}$
- $x' = \alpha^{r}$
- $x = x$

Hence $H(M||x') = H(M||x)$.

Digital Signature Standard :-

- Published by NIST
- DSS uses SHA and presented as DSA
- Latest version incorporates RSA and ECC.

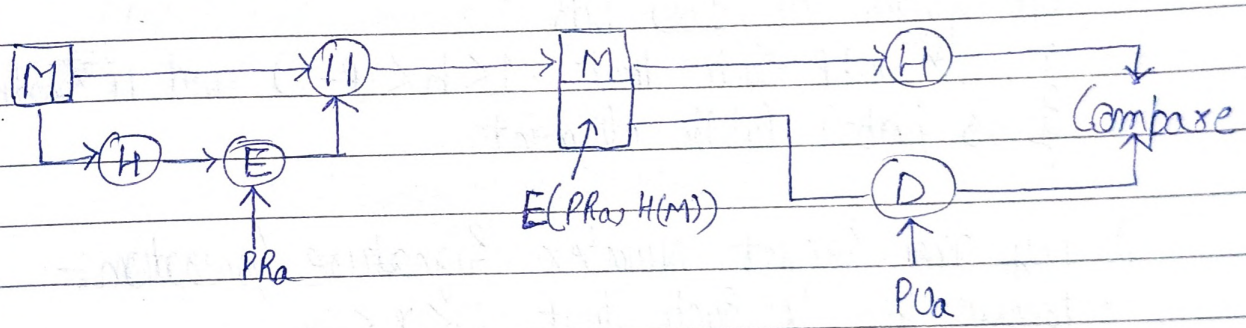
The DSS Approach :-

- Provides only digital signature function
- No encryption or key exchange.
- Not a public key technique.

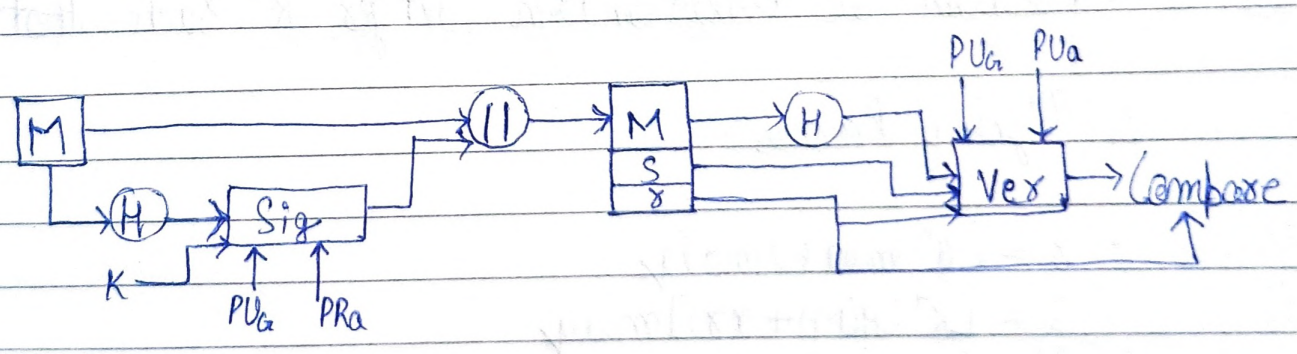
Two approaches to Digital Signatures.

1. RSA approach.
2. DSS approach

1 RSA - Approach :-



2. DSS-approach :-



Digital Signature Algorithm :-

- DSA
- Based on the difficulty of computing discrete Logarithms
- Based on originally presented ElGamal and Schnorr schemes

★ Phases of Digital Signature algorithm

1. Global public-key Components.
2. Key and secret number Generation.
3. Signing process
4. Verifying process.

1. Global Public-Key Components :-

- Prime number 'p' such that $2^{L-1} < p < 2^L$ for $512 \leq L \leq 1024$
- 'L' is a multiple of 64.
- 'q' is a prime divisor of p-1 where $2^{159} < q < 2^{160}$ i.e. bit length of 160 bits.
- $g = h^{\frac{p-1}{q}} \bmod p$ such that $1 < h < (p-1)$ and $h^{\frac{p-1}{q}} \bmod p > 1$. g is Global Public element.

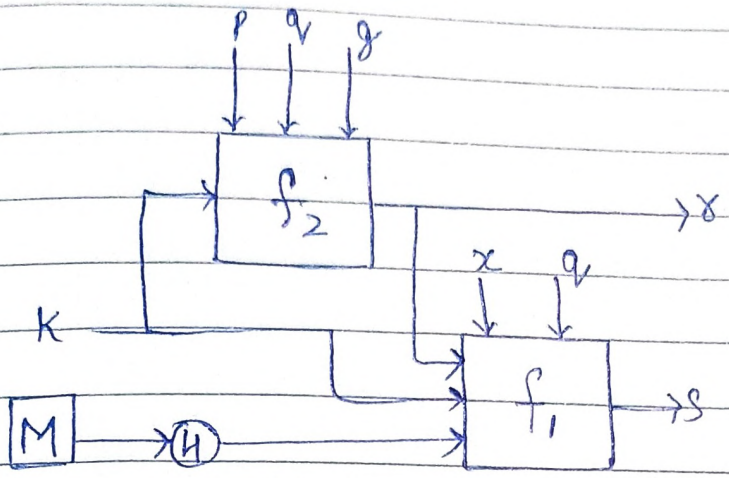
2. Key and secret Number ~~Signature~~ Generation :-

- Private Key 'x' such that $0 < x < q$.
- Public Key 'y' such that $y = g^x \bmod p$
- Random or Pseudorandom integer 'K' such that $0 < K < q$

3. Signing Process :-

- $r = (g^k \bmod p) \bmod q$
- $s = [K^{-1}(H(M) + xr)] \bmod q$
- Signature = (r, s)

- Diagram of Signing process:-



4. Verifying process:-

- $w = (s')^{-1} \bmod q$
- $u_1 = [H(M')w] \bmod q$
- $u_2 = (r')w \bmod q$
- $v = [(g^{u_1} y^{u_2}) \bmod p] \bmod q$
- Test $\Rightarrow v = r'$

- Diagram of Verifying process