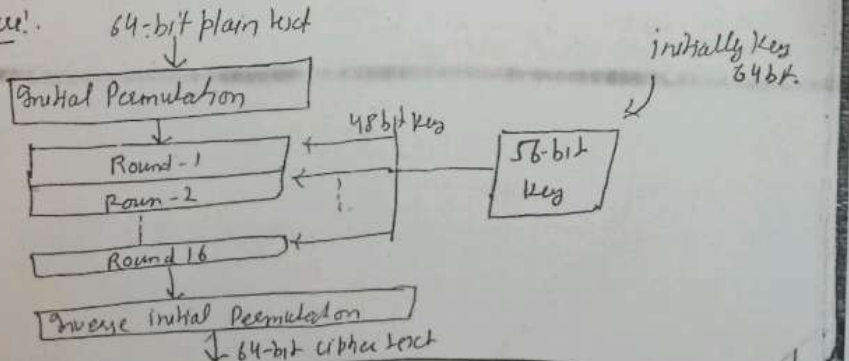
UNIT = 3

Private / Symmetric Key :- It is a type of encryption where only one key (secret key) is used for both encrypt and decrypt electronic information. The entities communicating via symmetric encryption must exchange the key so that it can be used in the decryption process.

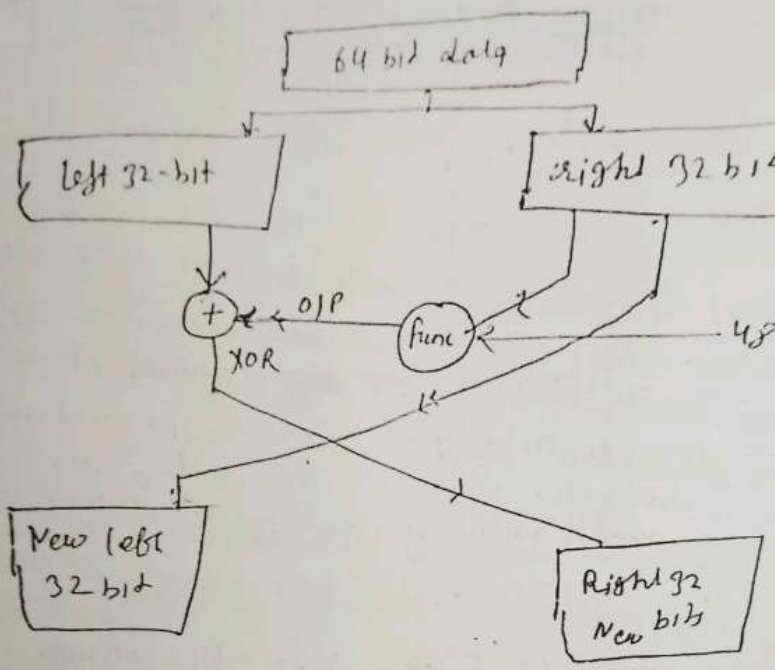
DES (Data Encryption Standard) :- It is block cipher it also symmetric cipher. It used 64 bit plain text block, It encrypts the data in blocks size 64 bits each. On this total 16 no. of rounds ↓ each is a diestel round.

Steps:-
(i) Initial permutation.
(ii) 16 diestel rounds.
(iii) swapping / left right swap.
(iv) final Permutation / Inverse initial Permutation

Basic structure:-

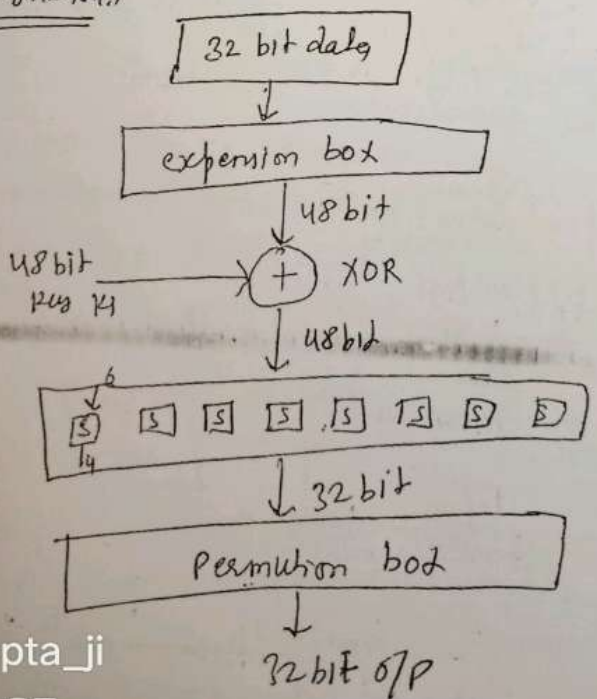


Feistel rounds



Round-1
Round-II

Function defination:



* S = S box
Substitution boxes.

what ha
32 bit
explain
DONT
So here
There
Now,
Now
do
an S-bo
S
4 bit
Ther
How conv

	0
0	3
1	4
2	10

what happen in expansion box!.

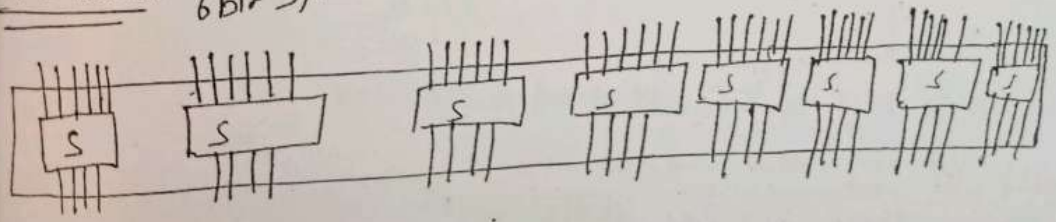
32 bit data will be ⟶ 1's and 0's form but for explanation let us conside a text.

DONT	GIVE	THEM	ONEY	TOTH	ATPE	RSON

T..	GIVE	T	E	THEM	O	F	RSON	D

So here every 4bit block is converted to a 6 bit block.
There were 8 blocks of 4bit each = 32bit.
Now, there are 8 block of 6bit each = 48 bit.
Now these 48 bit ⟶ XOR with 48bit key and given to
to S-box's

In S-boxes!. 6bit I/P



4bit O/P

O/P = 4×8 = 32bits

There will 32-bit will go into the Permutation box.

How converted eg:- 0 0 1 0 1 1 → 01 → 1
0101 → 5 S-box

(column, row)

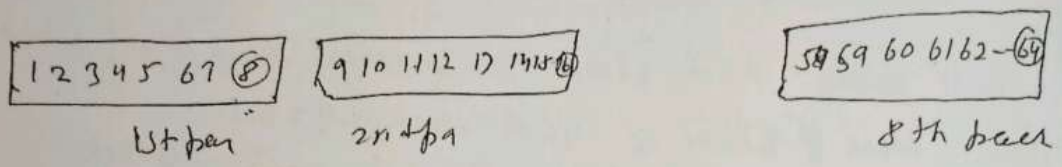
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	3	5	7	0	1	4	.	.	.	.	.	.	.	1	4	5
1	4	2	1	6		7								2	3	8
2	10													3	4	0
3														11	10	3

How 16-subkeys are generated!

Actually, we have 64-bit key which go as a input to P-1 (Permuted choice-1) and we get o/p as 56 bit key.

Inside (PC-1)

64 bit key divided into 8-parts each of 8-bit
8*8 = 64 bit



1st part 2nd part 8th part

From each part last bit → discarded.

i.e bit = 8, 16, 24, 32 —— 64 discarded.

Hence we have 8 parts of 7 bits each

8*7 = 56 bits

→ o/p of PC-1 is 56-bits which is divided into 2 parts of 28 bit each → C_0 ; D_0.

Now, these bits are shifted with left shift on each round.

- in Round no. i = 1, 2, 9, 16 → 1 shift rotated left by 1 bit
- in other rounds two halves rotated left by 2 bits.

After shifting we get (C_1 ; D_1) which goes o/p to PC-2

Inside PC-2 56 bit → 48 bit using predefined table.

Then we get our 1st key for Round-1

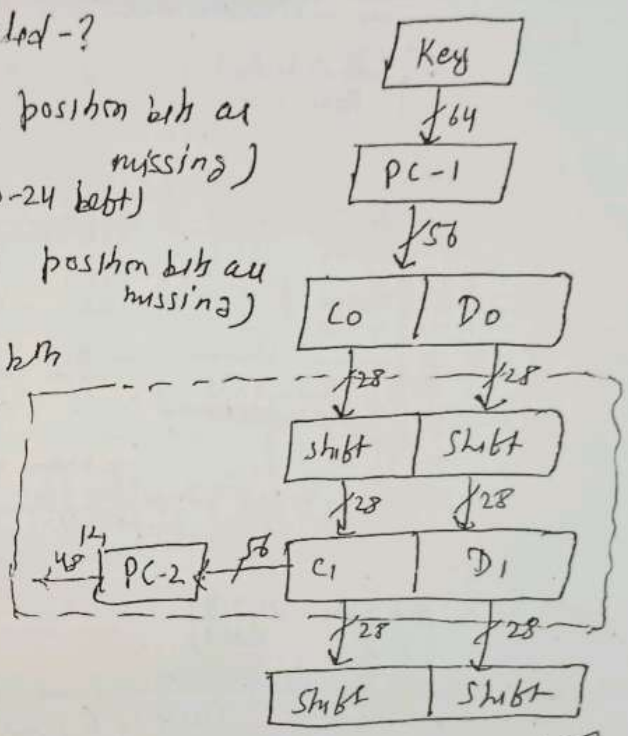
So C1 → 28 bit → (1-28)
D1 → 28bit → (29-56)

Now 56 bit to how 48 selected -?

left half C_1 (9, 18, 22, 25. position bit are missing)
10-24 left)

Right half D_1 (35, 38, 43, 54 position bit are missing)
discarded)
i.e - 24 bit

Similarly = 16 times.

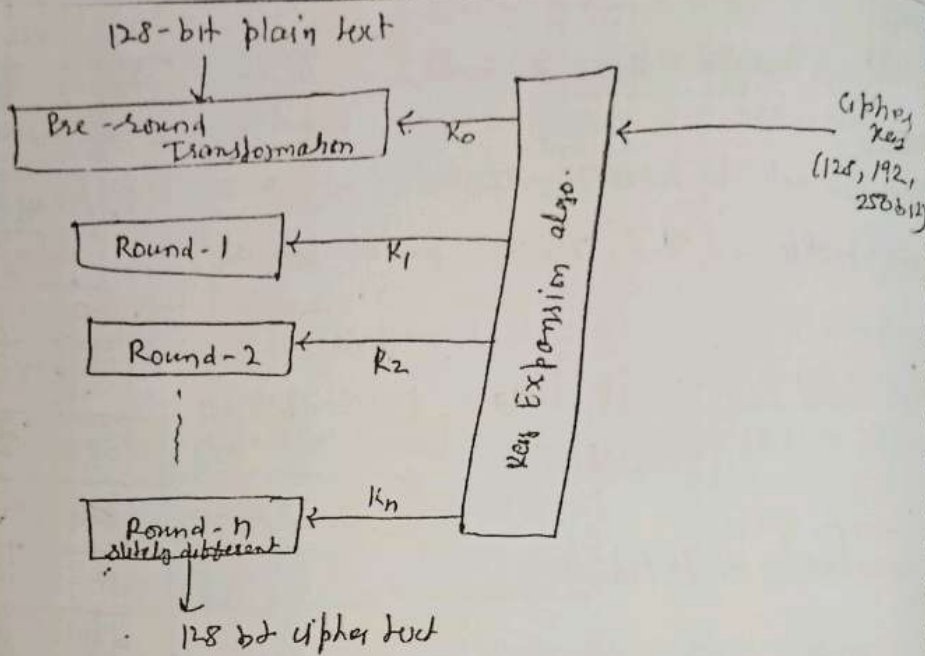


AES (Advanced Encryption Standard).

It symmetric key block ciphers. It established in 2001 by the U.S NIST. Fixed block size = 128 bit. i.e 16 bytes = 4 words.

Rounds	no. of bit in key	
10	128	AES-128 version
12	192	AES-192 version
14	256	AES-256 version.

No. of keys generated by key expansion algorithm = (no. of rounds + 1)

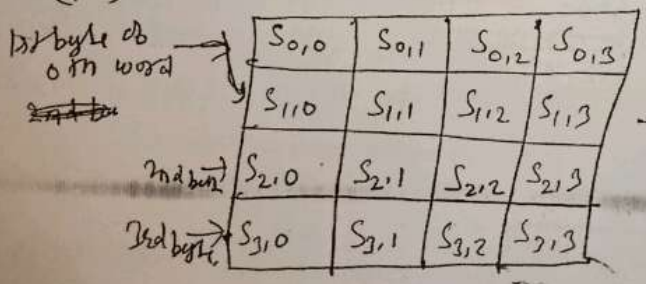


State :- 16 bytes (4 × 4) stores intermediate results.

Input array format :- (4×4 i.e 16 byte i.e. 128 bits)

State matrix array → for storing intermediate result)

(4×4)

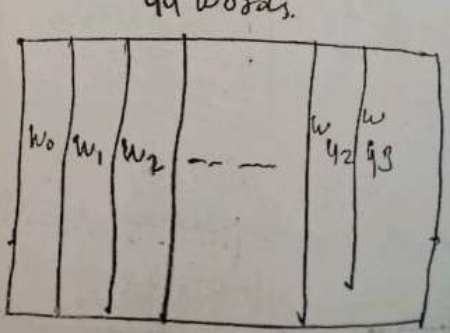


Key 128 bit i.e 4 words

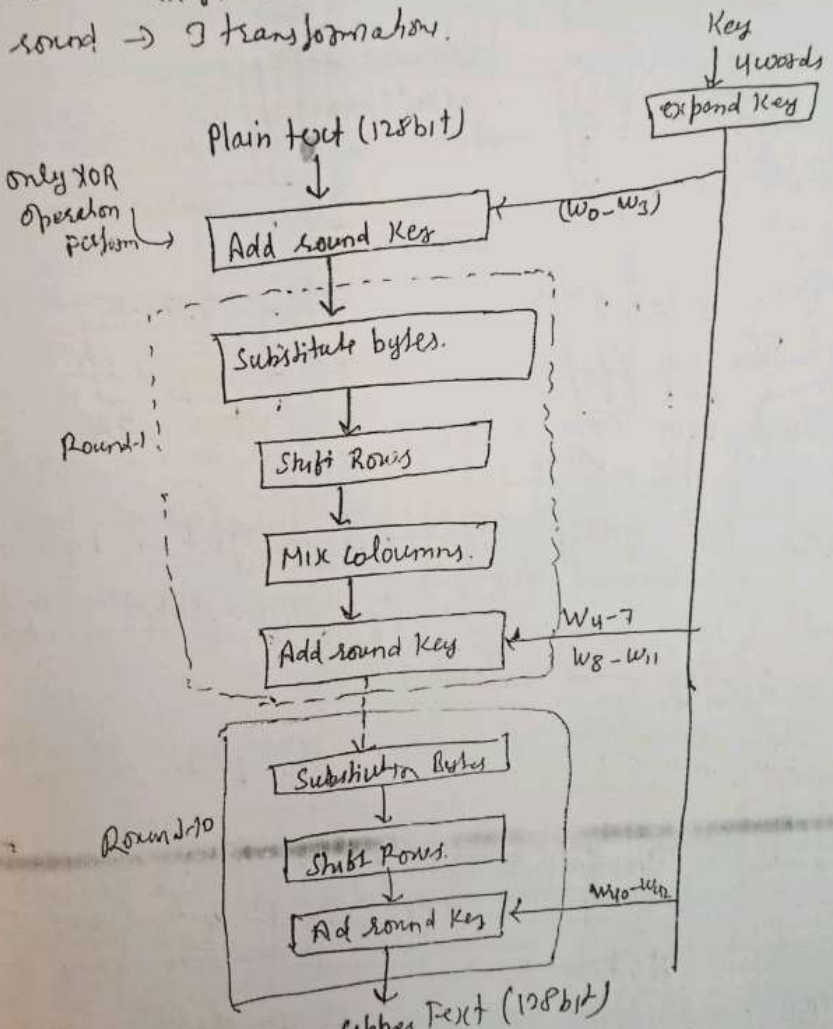
W_0	W_1	W_2	W_3
K_0	K_4	K_8	K_{12}
K_1	K_5	K_9	K_{13}
K_2	K_6	K_{10}	K_{14}
K_3	K_7	K_{11}	K_{15}

expand Key algo

44 words.

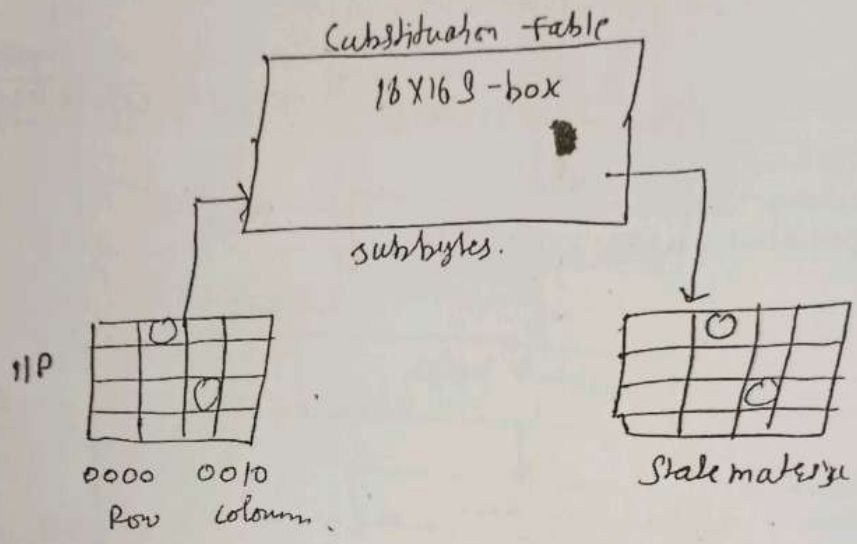


encryption algo → cipher, decryption algo = inverse cipher.
In decryption rounds keys are applied in reverse order.
It is much stronger than DES
each round except the last uses 4 transformations. but last round → 3 transformation.



Transformation:- Substitution, permutation, mixing, Regaddi

① Substitution:- only one table is used for transformation of bytes, which means that if 2 bytes are same the transformation is also same.



Subbytes:- at encryption side we interpret the byte as 2 hexa decimal digits.

1 hexa ⟶ row
2 hexa ⟶ Column
} of the substitution table.

Transformation is done are byte at a time.

② Permutation:- In this permute/shift the bytes. In DES permutation was done at bit level.
AES ——————— byte level.

Shift Rows
- Shifting is done to the left
- no. of shifts depends on the rows of the state matrix.

Row 0	63	C9	FE	30
Row 1	F2	F2	63	26
Row 2	C9	C3	7D	D4
Row 3	BA	63	82	DU

Shift Row 1 → Shift left

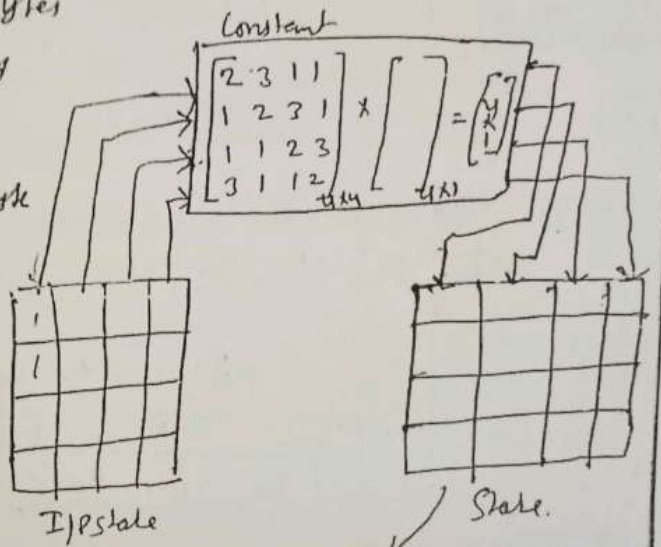
63	C9	FE	30	→ No shifting
(F2)	(63)	(26)	F2	→ 1 byte shift
(7D)	(D4)	C9	C3	→ 2 byte shift
(D4)	BA	63	82	→ 3 byte shift.

as a Input.

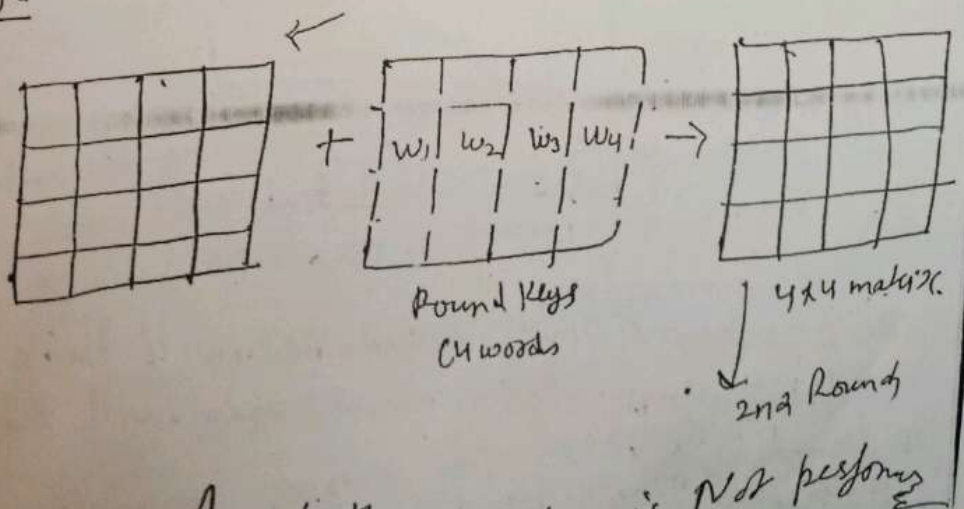
③ Mixing!- mix columns!- for encryption.

Take each word/colown i.e 4 bytes or 4×1 matrix and multiply it with the constant matrix

The o/p is (4×1) matrix of 4 byte and is stored in the o/p of state matrix.



④ Key adding!-



Last Round the Mixing is Not performed

Feistel cipher structure:

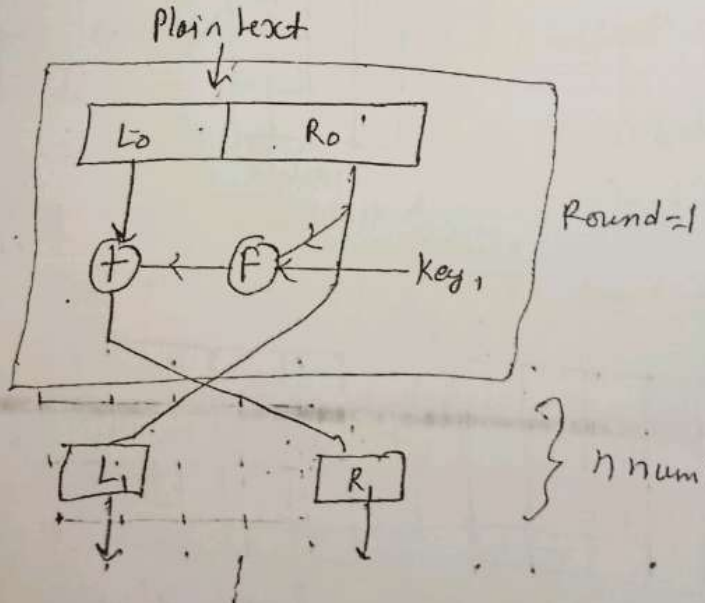
most of the block cipher techniques follows this structure.

(i) The plain text is divided into 2 halves L_0 and R_0

→ The 2 halves of the data pass through n rounds of processing and then combine to produce the cipher text block.

→ on the right halves we apply a function and in the function we will used a subkey generated from the master key.

The o/p of this is XORed with the left halves and then their o/p will be swapped.



This is only one single Round. we will have n Rounds - depends upon algo. all rounds will have same structure.

If any algo we divide the plain text in 2 halves and apply the function on RHS and XOR with LHS and the o/p is swapped then, that algo follows feistel structure

Terms!. Imp.

① Block size!- Larger block size, more security.

② Key size!- Large key size, but more security but decrease the speed of encryption / decryption.

③ no. of rounds!- more rounds, more secure.

④ Subkey generation algo!- more complex algo, harder for attacker to steal data.

⑤ Function / Round function (F)!- more complex function, harder for the encrypt / analysis to attack.

Modes of operation!.

for different types of message we need different modes of operations.

mainly 5 types.

① ECB = electronic code book mode.

② CBC = Cipher block chaining mode

③

It is already defined / Explain in perivious chapter

Asymmetric key cryptography also

* RSA :- Rivest - Shamir - Adleman Algo. (1978)

It is an asymetric cryptographic algo. It used 2 keys. one is public and other one is private key concept is used here.

Public Key :- known to all users in Network.

Private Key!- Kept secret, not sharable to all.

If public key of user A is used for encryption we have to use the private key of same user for decryption

The RSA scheme is a block cipher in which the plain text and cipher text are integer b/w 0 and n-1 for some values n.

1. Key generation:-
   (i) select 2 larger prime numbers p and q.
   (ii) Calculate $n = p*q$
   (iii) Calculate $\phi(n) = (p-1)*(q-1)$ // eulers Totient function
   (iv) choose value of e

$$1 < e < \phi(n) \text{ and } gcd(\phi n, e) = 1$$

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

$$ed \equiv 1 \bmod \phi(n)$$

   (vi) public key = $\{e, n\}$
   (vii) privat key = $\{d, n\}$

Example:- Let $p = 3, q = 11$

$n = p*q = 3*11 = 33$

$\phi(n) = 2*10 = 20$

So let $\boxed{e = 7}$ as $1 < 7 < 20$ and $gcd(7, 20) = 1$

Now, $d \equiv e^{-1} \bmod \phi(n)$

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

$7*d \equiv 1 \bmod \phi(n)$

$(7*d) \bmod 20 = 1$ (∵ d = 3)

// find multiplicative inverse of 7, multiplicative $\phi(n)$, i.e. we must find a no. satisfying a value greater than 1 i.e. (7*d) should be 1

(i) [illegible]

(ii) Calculate $n = p * q$

(iii) Calculate $\phi(n) = (p-1) * (q-1)$ # euler's

(iv) Choose value of e

$1 < e < \phi(n)$ and $\gcd(\phi n, e)$

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

$ed \equiv 1 \bmod \phi(n)$

(vi) public key = $\{e, n\}$

(vii) prival key = $\{d, n\}$

Example:- Let $p = 3, q = 11$

$n = p * q = 3 * 11 = 33$

$\phi(n) = 2 * 10 = 20$

So let $\boxed{e = 7}$ as $1 < 7 < 20$ and gcd(7,

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

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

$7 * d \equiv 1 \bmod \phi(n)$

$(7 * d) \bmod 20 = 1$ (∵ 23) // find multiplicative inverse of 7 mod 20 so that

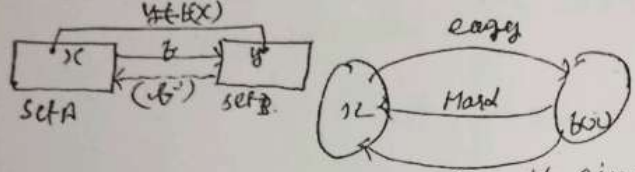
multiplicative inverse mod 20 and find a no. satisfying a value

than this i.e (7*d) should be 1

→ symmetric to x-axis
→ If we draw a line, it will touch a max of 3 points.

Trapdoor function: It is a function that easy to compute in one direction, yet difficult to compute in the opposite direction (finding its inverse) without special information called the trapdoor.

A function is a rule that associates one element in set A called the domain to other element in set B called range.



Algo:

Step-1 Let $E_p(a,b)$ be the elliptic curve.

Step-2 Consider the equation $Q = kP$.
where Q, P → points on curve and $k < n$.

Step-3 If k and P are given, it should be very easy to find Q, but if we know Q and P, it should be extremely difficult to find k.
↳ This is called the discrete algorithm problem for elliptic curve.

It is a one way function. → Trapdoor function.

ECC - Key exchange:

Global Public elements.

$E_q(a,b)$: elliptic curve with parameters a, b and q.
↳ prime no. or an integer of the form 2^m.

G: point on the elliptic curve whose order is large value of n.

User-A Key generation.

Select private key n_A $\quad n_A < n$.

Calculate public key P_A $\quad \boxed{P_A = n_A \times G}$

User-B Key generation.

Select privat key n_B $\quad n_B < n$

Calculate public key P_B $\quad \boxed{P_B = n_B \times G}$

Calculation of secret key by user A

$$\boxed{K = n_A \times P_B}$$

Calculation of secret key by user B

$$\boxed{K = n_B \times P_A}$$

User → Encryption:

- Let the message be M
- First encode this message M into a point on elliptic curve.
- Let the point be $\boxed{P_m}$ → Now this point encrypted
- → for encryption choose a random +ve number K.
- The cipher point will be

$$\boxed{C_m = \{kG, P_m + kP_B\}}$$ → for encryption Public key of B used.

This point will be sent to the receiver.

Decryption:-

For decryption, multiply the 1st point in the pair with receivers secret key

i.e. $K_G * n_B$ // for decryption private key of B used.

Then subtract it from 2nd point/coordinate in the pair.

$$\boxed{i.e = P_m + kP_B - (k G * n_B)}$$

but we know $P_B = n_B * G$.

So $= P_m + kP_B - kP_B$

$= P_m \rightarrow$ original point.

→ So receiver gets the same point

Diffie-Hellman Key exchange Algorithm:- It is not a encryption algorithm. It is used to exchange the secret keys b/w 2 users. We will use asymmetric encryption to exchange the secret key. b/w the users.

Why we use:- bcoz when we are sending a key to receiver it can be attacked in b/w

Algorithm:-

(i) Consider a prime number q

(ii) Select (α) such that it must be the primitive root of q and $\boxed{\alpha < q}$

a = a is primitive root of q if

$a \bmod q$

$a^2 \bmod q$ → $a^{q-1} \bmod q$

$a^3 \bmod q$

gives results {1, 2, 3. ___ q-1}

i.e. values should be repeated & we should have all values in the o/p set from 1 to q-1

Example:- q = 7

$3^1 \bmod 7 = 3$
$3^2 \bmod 7 = 2$
$3^3 \bmod 7 = 6$
$3^4 \bmod 7 = 4$
$3^5 \bmod 7 = 5$
$3^6 \bmod 7 = 1$

Let q = 7 (prime)

$\alpha < q$ it is primitive root.

α, q → global public elements (knowing everyone)

x → private key of user

y → public key of user

i) assume x_A (private key of A) and $x_A < q$

Calculate $\boxed{Y_A = \alpha^{x_A} \bmod q}$

Example:- Key generation of Person 1

Assume $x_A = 3$

$Y_A = \alpha^{x_A} \bmod q \Rightarrow 5^3 \bmod 7 = 125 \bmod 7$

$\boxed{Y_A = 6}$

ii) assum x_B (private key of B) $x_B < q$

Calculate $Y_B = \alpha^{x_B} \bmod q$.

Eg: $x_B = 4$

$Y_B = 5^4 \bmod 7 = 2$

$Y_B = 2$.

Now we will calculate secret key.

To calculate the secret key both the sender & receiver will use public key.

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

(public key.)

$K_A = (Y_B)^{X_A} \bmod q \Rightarrow 2^3 \bmod 7 = 2^3 \bmod 7 \Rightarrow K = 1$

$K_B = (Y_A)^{X_B} \bmod q \Rightarrow 6^4 \bmod 7 \Rightarrow (36 \times 36) \bmod 7 \Rightarrow K = 1$

Thus the keys are exchanged.

$K_1 = K_2$ then we say exchange is successful.

Digital Signature:

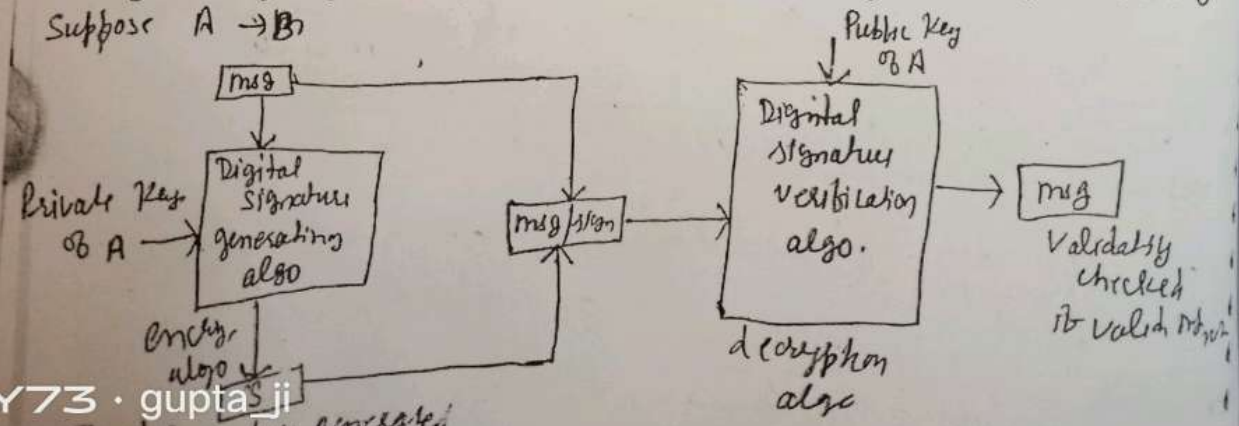
It is very important in e-commerce, online transaction. It based on asymmetric key cryptography.

encryption → private key.

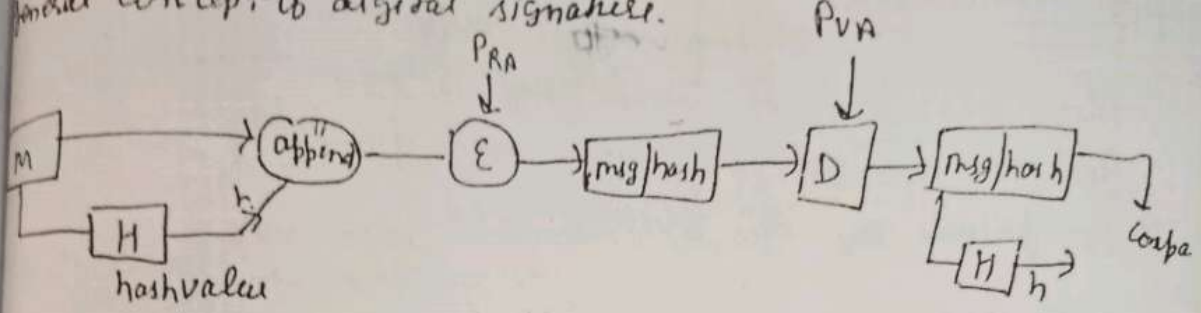
decryption → public key.

It is used for msg authentication & non repudiation & msg integrity. And it is not used for confidentiality.

Suppose A → B



General concept of digital signature.



H = hash function, h → hash code, // = append.

*Note:- The signature must use some info unique to the sender to prevent both forgery & denial.

→ also provide msg integrity.
bcoz if msg changed then at receiver side we will not get the exact msg.

*Note:- when we sign a document digitally we send the signature as a seprate documents.

∴ Sender sends 2 documents → msg → signature

⇒ DS must use some info unique to the sender, to prevent forgery & denial.

→ It must be easy to produce digital signature.
→ It must be easy to verify & recognize the DS.

We need:- (i) Key generation algo → to generate private key
(ii) Signing algo → M or private key → DS.
(iii) verifying alg → using public key & sign.

Msg authentication:- It can verify that msg is used by Alice bcoz Alice's public key is used in verification. And we can get the same msg digest / hash value only if private key of Alice is used. - Achieved.

Msg Integrity:- If msg is changed in between any how the receiver will not get the same hash value / msg digest. So if hash value / msg not same the msg changed. ∴ hash function helps in preserving the Integrity of msg.

Non-repudiation:- achieved by using a trusted 3rd party.

KNAPSACK Algorithm:- Developed by Ralph and Martin Hellman.

It is first General Public Key algo.
In this some weight are given and we choose weights and find some no. total.
like given weight = 1, 6, 8, 15 and 24.
Total = 30
= (1, 6, 8, 15)

eg:- encryption → suppose → P.T = 10011 11010

10011	11010
1 6 8 7 5 24	1 6 8 15 24
1+15+24 = 40	1+6+15 = 22

C.T = 4022.

How to generate Keys → public → Encrypt (Hard Knapsack)
private → Decrypt (Easy Knapsack)

First choose Easy Knapsack nd derived the "Hard Knapsack.

Easy Knapsack!. The weights are in super increasing sequence.
like → {1, 2, 4, 9, 20, 38}

Key generation'. Decryption

Super increasing Seq (D) = (1, 2, 4, 10, 20, 40), (Private Key)

n and m
↳ should be greater than sum of all no. in sequence
m → 110 , n → 31 }) recives should know this
↳ multiplier [no. factor is common with modulus

(1 × 31) mod 110 ⇒ 31 = E2 [31, 62, 14, 90, 70, 30)
(2 × 31) mod 110 ⇒ 62 Public Key
(4 × 31) mod 110 ⇒ 14 ↓
(10 × 31) mod 110 ⇒ 90 Encryption -
(20 × 31) mod 110 ⇒ 70 Suppose send msg
(40 × 31) mod 110 ⇒ 30 [100100 111100 10111 0] P.T.

{1, 2, 4, 10, 20, 40}
11 ⇒ (1, 10) → 100100
17 ⇒ (1, 2, 4, 10) → 111100
35 ⇒ (1, 4, 10, 20) ⇒ 101110.
Plaintext recived.

100100 ⇒ 31 + 90 = 121
111100 ⇒ 31 + 62 + 14 + 90 = 197
101110 ⇒ 31 + 14 + 90 + 70 = 205
C.T ⇒ 121 197 205
↓ Decryption.

You have just find the Inverse of 'n' ⇒ $31^{-1} = n^{-1}$

$31 \times x \bmod 110 = 1 \Rightarrow 31^{-1} = 71$

use private key ⇒ multiply C.T with 71 and then mod 110

(121 × 71) mod 110 = 11
(197 × 71) mod 110 = 17 ⟹ plain text generate add the private key and
(205 × 71) mod 110 = 35 bit for 1's. place ko kon se no

Public Key Infrastructure:-

It uses a pair of keys to achieve the underlying security service. The key pair comprises of private key and public key.

Since the public keys are in open domain, they are likely to be abused. It is thus necessary to establish and maintain some kind of trusted infrastructure to manage these keys.

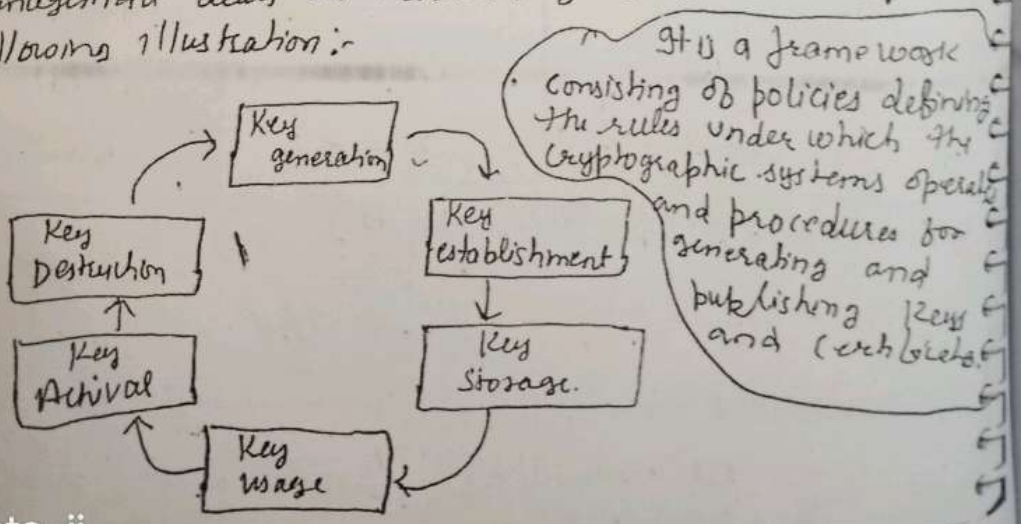
Key Management:-

It goes without saying that the security of any cryptosystem depends upon how securely its keys are managed. Without secure procedures for the handling of cryptographic keys, the benefits of the use of strong cryptographic schemes are potentially lost.

It is observed that cryptographic schemes are rarely compromised through weakness in their design, However they are often compromised through poor key management.

Aspects:-

- Cryptographic key are nothing but special pieces of data.
- Key management refers to the ~~retire~~ secure administration of cryptographic keys.
- Key management deals with entire key lifecycle as depicted in following illustration:-



There are two specific requirements of key management for public key cryptography.

- Secrecy of private keys:- Throughout the key lifecycle, secret keys must remain secret from all parties except those who are owner and are authorized to use them.
- Assurance of public keys:- In public key cryptography, the public key are in open domain and seen as public pieces of data. By default there are no assurances of whether a public key are correct, with whom it can be associated, or what it can be used for. Thus key management of public keys needs to focus much more explicitly on assurance of purpose of public keys.

PKI ⇒ It provides assurance of public key. It provides the identification of public keys and their distribution. An anatomy of PKI comprises of the following components.

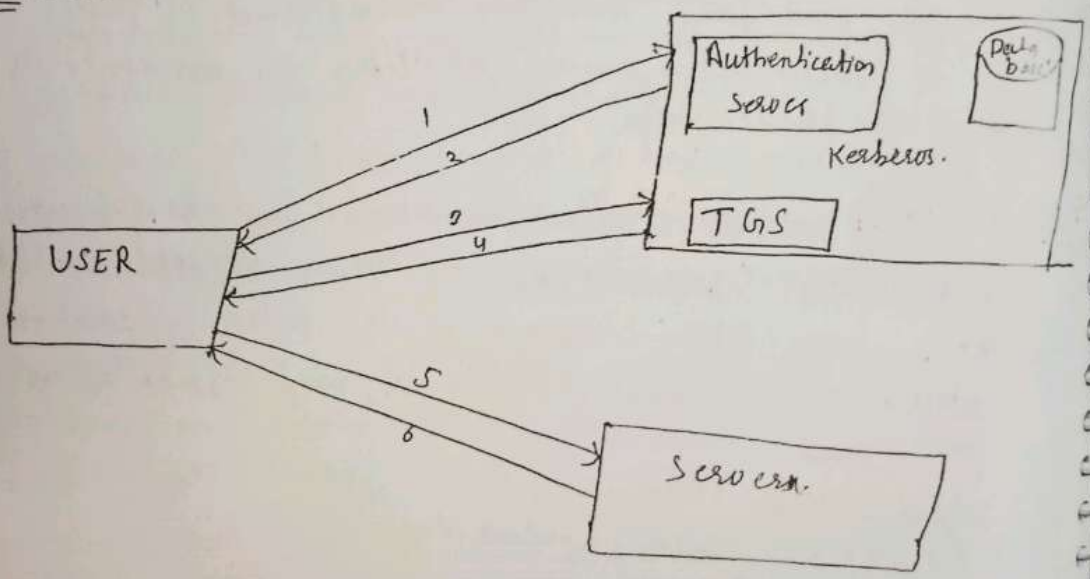
- Public key certificate, commonly ~~used~~ referred to as digital certificate.
- Private key token.
- Certification Authority.
- Registration Authority.
- Certificate management System.

Kerberos:- It provides a centralized authentication server whose function is to authenticate users to servers to users. In Kerberos Authentication server and database is used for client authentication. Kerberos runs as a third-party trusted server known as key Destribution Center. Each user and server ~~network~~ is a principal.

The main components of Kerberos:-

- Authentication server (AS):- It performs the initial authentication and ticket for Ticket Granting Service.
- Database:- The Authentication Server verifies access rights of users in database.
- Ticket Granting Server (TGS):- It issues the ticket for the servers.

Overview:-



Step-1:- User login and request services on host. Thus user request for ticket-granting-service.

Step-2:- AS verifies user's access right using database and then gives Ticket-granting-ticket and session key. Results are encrypted using password of user.

Step-3:- Decryption of message is done using the password then send the ticket to ticket Granting Server. The Ticket contains authenticators like user name and network address.

Step-4!- Ticket granting server decrypts the ticket send by user and authenticator verifies the request then creates the ticket for requesting services from the server.

Step-5!- User send the Ticket and Authenticator to the server

Step-6!- Server verifies the Ticket and authenticators then generate the access to the services. After this user can access the services.

Secret Sharing Scheme!- In cryptography, SS refers to any method for distributing a secret among a group of participants, each of which allocates a share of the secret. The secret can only be reconstructed when the shares are combined together; Individual shares are of no use on their own.

The secret is opened only when specific conditions are fulfilled. Each of n participants is given a number of shares and any group of t (threshold) or more shares together can open the secret but no group of less than t shares can.

A secure secret sharing scheme distributes shares so that anyone with fewer than 't' shares has no more information about the secret than someone with 0 shares.

Consider the naive secret sharing scheme in which the secret phrase - "password" is divided into the shares "pa---," "--ss---," "---wo--," "---rd,". A person with 2 shares knows only that the password consists of eight

letters. He would have to guess the password from 26^8 = 268 billion possible combinations. A person with one share however, would have to guess only the size letters from 26^6 = 308 million combinations. This system is also a secure secret sharing scheme. because a player with less than 't' shares gains such significant information about the content of the secret. In a secure scheme even a player missing only one share. Should still face 26^8 = 208 billions combinations.

It invented by both Adi Shamir and George Blakley independently in 1979.

*Threshold scheme: Let t, w be two integers with t ≤ w. A (t, w) threshold scheme is a method of sharing a msg M among a set of w participants such that any subset consisting of t participants can reconstruct the message M, but no subset of smaller size can reconstruct M.

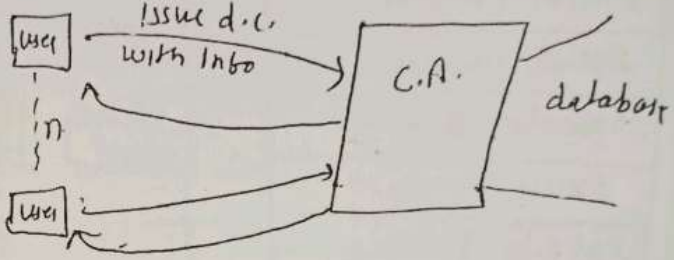
Digital Certificates:- It is small file on computer / electronic device. File extension is generally (.cer). It establishes the relation b/w a user and the public key. It is standard follow is PKI. Digital certificates must be issued by trusted party. or trusted entity.

Sample Digital Certificate :-
- user name :- XYZ
- Public Key :- <12$a#>
- Serial no :- 12345
- other Info :- Email-id.
- Valid from :- 31 Jan 2006
- Valid To :- 31 Jan 2016
- Issuer Name :- Verisign.

Fields of Digital certificate.
- → Version : X.509
- → Signature Algo Identifiers
- → Issuer user I D.
- → Certificate Authority Digital signature.

CA Digital Signature!- This field used during digital certificate verification.

Certification Authority!- (CA) It trusted agency that can issue digital certificate. like Verisign; Entrust.



X.509 Certificate!-

The use of CA solved the problem of public key fraud. The Internet community has accepted the ITU-T recommendation X.509 as a way to Unify certificate formats.

Format of certificate!-

① Version number!- It is prosping the version no. of X-509.

② Serial no. :- It is unique no. for each certificate issuues.

③ Signature algo ID!- It Identifying the signature algo used in the certificate.

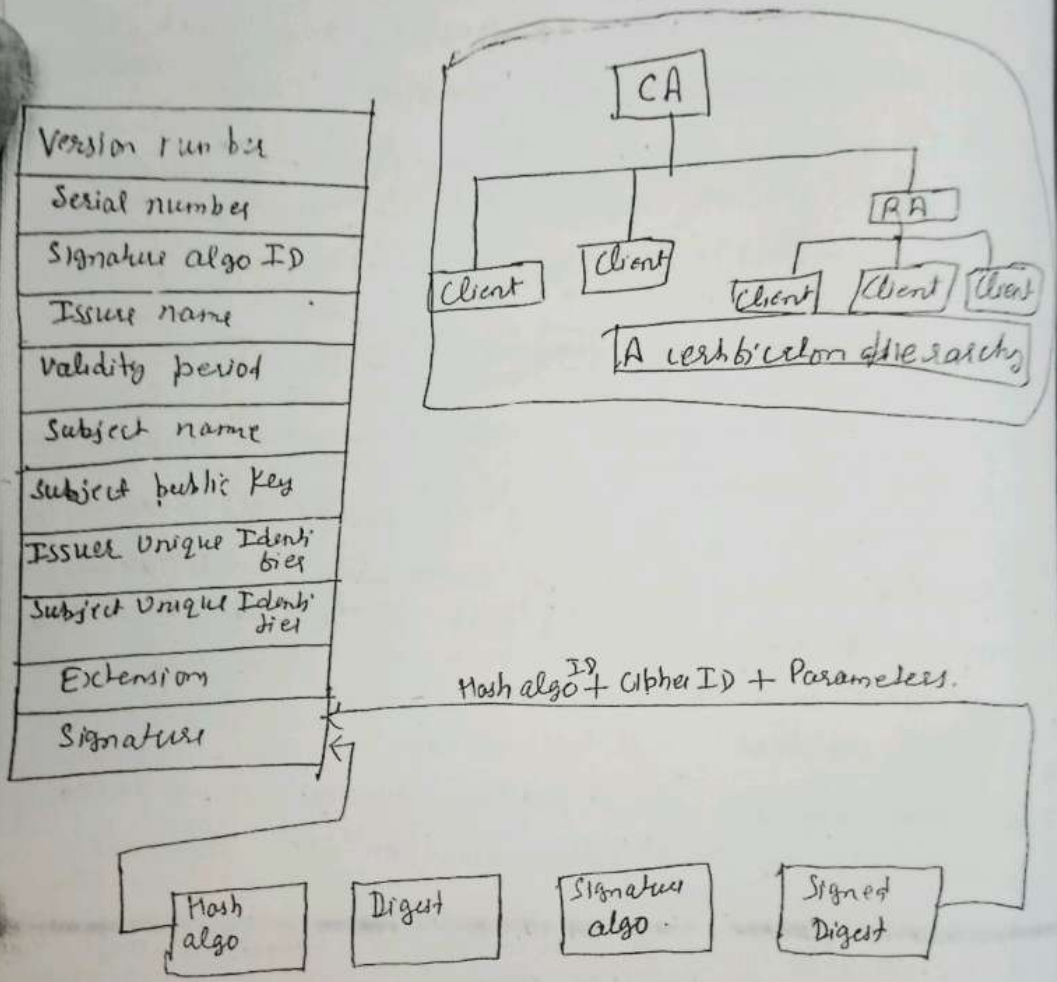
④ Issuer name :- This field Identifying this certificate authority that is issued the certificate

⑤ Validity period!-

⑥ Subject name!- which is specify defining the entity that owns the public key.

Subject public key!- It giving the value of public key owner of the DC and identifying the Public

key algorithm.
And the 3 field are optional ⇒ Issuer unique identifier
Subject unique identifier
Extensions.



→ Certificate Renewal:- a period of validity if expire the user
this.

→ C. Revocation:- Before expire the user one certificate suddenly
If there are any changes be done. it
update the Digital certificate

31/03/021

ATTACKS :-

gt is mainly
① Passive
Method:- Re
• Traffic en
② Active at
• Masque
reciever
party