③ Public key Encryption

* RSA Algorithm: (Rivest Shamir Aldmen)

(1) choose two large prime number P and Q

(2) calculate $N = P \times Q$

(3) select the public key (i.e. encryption key) E such that it is not a factor of $(P-1)$ and $(Q-1)$ (Φ) → denotes

(4) select the private key (i.e. description key) D such that the following equation is true

(5) $(D \times E) \bmod (P-1) \times (Q-1) = 1 = (D \times E) \bmod \Phi = 1$

(6) for encryption calculate ciphertext CT follow

(7) $CT = PT^E \bmod N$

(8) Sends CT as the cipher Text to reciver

(9) For decryption calculates the PT from CT

$\therefore PT = CT^D \bmod N$

(1) Perform encryption and Decryption using RSA Algorithm:

$p = 7,\ q = 11,\ m = 8,\ e = 17$

Solⁿ:-

(1) P and Q are (7, 11) respectively;

Now,

calculate, $N = P \times Q$

$= 7 \times 11 = 77$

$\therefore N = 77$

(2) Calculate Φ using (P-1) and (q-1)

$\Phi = (p-1)(q-1)$

$= (7-1)(11-1)$

$= (6)(10)$

$\Phi = 60$

(3) Now, calculate

~~(D) x~~ $(D \times E) \bmod (p-1) \times (q-1) = 1$

$\therefore (D \times E) \bmod \Phi = 1$

$D \cdot 17 \bmod 60 = 1$

we need to find D.

So, $ax + by = gcd(60, 17)$

$ax + by = gcd(a, b)$

~~ax+by=~~

$\Phi x + ey = gcd(\Phi, e)$

$60x + 17y = gcd(60, 17)$

NO	a	b	d	k
1	1	0	60	-
2	0	1	17	3
3	1	-3	9	1
4	-1	4	8	1
5	2	-7	1	

$17 \overline{)60}$ quotient 3

-51

7

(0-1)×3

= -3

60-17×3

60-51 = 9

9)17 quotient 1

-9

0-1×1

∴ $x = 2$ ∴ $y = -7$

Now,

$60(2) + 17(-7) = gcd(60, 17)$

$120 + (-119) = gcd(60, 17)$

$1 = gcd(60, 17)$

$\boxed{d = -7}$

if $d > \phi$

$\therefore d = d \bmod \phi$

if d = negative

$d = d + \phi$

$= -7 + 60$

$\boxed{d = 53}$

D.E. $\bmod \phi = 1$

$53 \cdot 17 \bmod 60 = 1 = 901 \bmod 60 = 1$

Now, (4) calculate $C = P^E \bmod N$

(5) $PT = C^D \bmod N$

$\therefore C = P^E \bmod N$

$= 8^{17} \bmod 77$

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1
1	0	0	0	1

$8^1 \bmod 77 = 8$

$8^2 \bmod 77 = 64$

$8^4 \bmod 77 = 15$

$8^8 \bmod 77 = 71$

$8^{16} \bmod 77 = 36$

$8^{8(2)} \bmod 77$

$(71)^2 \bmod 77$

$(8 \times 36) \bmod 77$

$\boxed{C = 57}$

$P = C^d \bmod n$

$= 57^{53} \bmod 77$

2^5	2^4	2^3	2^2	2^1	2^0
32	16	8	4	2	1
1	1	0	1	0	1

$57^1 \bmod 77 = 57$

$57^2 \bmod 77 = 15$

$57^4 \bmod 77 = 71$

$57^8 \bmod 77 = 36$

$57^{16} \bmod 77 = 64$

$57^{32} \bmod 77 = 15$

$= (57 \times 71 \times 64 \times 15) \bmod 77$

$= 8$

* Diffie- Helman key exchange

(a) It is not encryption algorithm
(b) used to exchange secret key between 2 users
(c) we will use asymmetric encryption to exchange the secret key ↓ (public and private key concepts)
(d) we use this algorithm, because when we are sending a key to recivers, it can be attacked in between it.

* Algorithm

(1) consider a prime number 'q'

(2) Select α such that it must be the primitive root of q and $\alpha < q$

'a' is a primitive root of q if

$a^1 \bmod q$
$a^2 \bmod q$
$a^3 \bmod q \ldots a^{q-1} \bmod q$
given result $\{1, 2, 3, \ldots q-1\}$

i.e. values should not be repeated & use should have all values in the o/p set from 1 to q-1

Example:

calculating Primitive root

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

→ Distinct

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

(3) Assume x_A (Private key of 'A') and $x_A < q$

calculate,

$y_A = \alpha^{x_A} \bmod q$ → public key of 'A'

(4) Assume x_B (Private key of 'B') and $x_B < q$

Calculate,

$y_B = \alpha^{x_B} \bmod q$ → public key of 'B'

(5) Now, we will calculate secret key, both the sender and Reciver will use public keys.

$k_1 = (y_B)^{x_A} \bmod q$

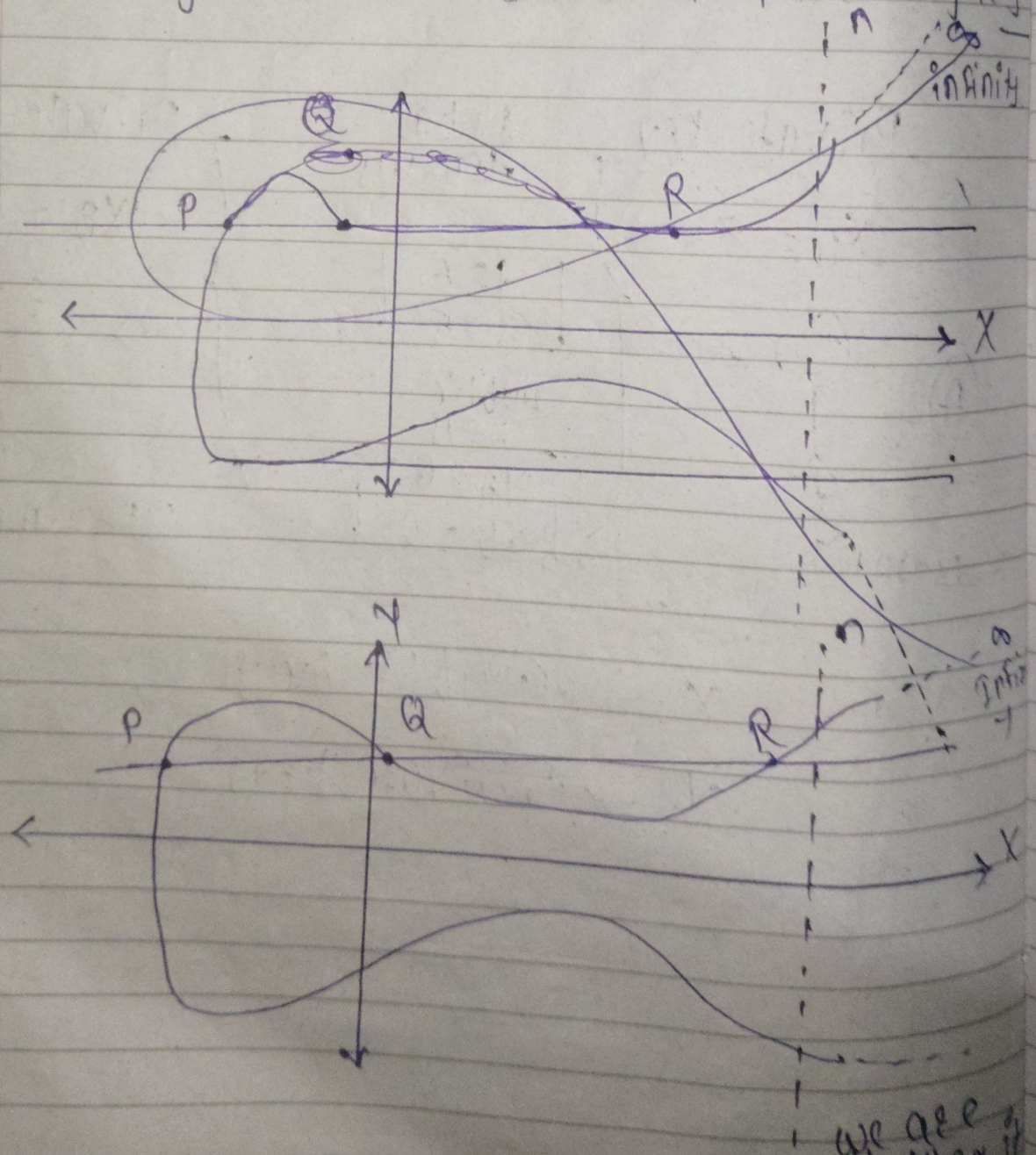
$k_2 = (y_A)^{x_B} \bmod q$

public keys known to all

∴ $k_1 = k_2$, then we say key exchange succesfully.

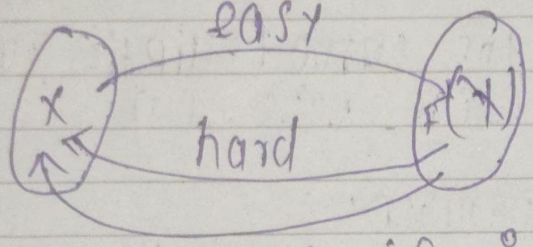
* Elliptic curve Cryptography (ECC)

(1) It is asymetric / public key Crypto System
(2) It provides equal security with smaller key size (e.g. as compared to RSA) as compared to non ECC algo.
    i.e. small key size and high security
(3) It make use of Elliptic curves
(4) Elliptic curves are defined by some mathematical function
    e.g. $y^2 = x^3 + ax + b$ → eqn of degree 3



→ Symmetric to x-axis
→ 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 difficult to compute in the opposite direction (finding its inverse) without special the information, called the trap door



easy if given "t" → trapdoor value

Let $E_p(a,b)$ be the elliptic curve
consider the equation $Q = kP$

where Q, P → points on curve and $k < n$

If k and P → given, it should be 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 curves

It is one way fn → Trap door function
i.e. A → B is easy but coming B → is very difficult

Algorithm is somewhat similar to Diffie Hellman but some changes.

* ECC Algorithm

ECC-key exchange

Global public elements

Eq(a,b) : elliptic curve with parameters a,b and q → prime No. or an integer of form 2^m.

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

→ User A key generation:

select private key (n_A)

calculate public key P_A

$n_A < n$ ∴ $P_A = n_A \times G$

→ User B key generation

select private key (n_B)

calculate public key P_B

$n_B < n$ ∴ $P_B = n_B \times G$

Calculation of secret key by user A

$K_A = [K = n_A \times P_B]$

Calculation of secret key by user B

$K_B = [K = n_B \times P_A]$

Ecc Encryption

→ let the message be m
→ first encode this message m into a point on elliptic curve
→ let this point be P_m

↓

Now, this point is encrypted

for encryption, choose a random positive integer k.

The cipher point will be → for encryption public key of B used.

$\therefore C_m = \{kG, P_m + kP_B\}$

This point will be sent to the reciver

Ecc Decryption

For Decryption, multiply 1st point in the pair with recivers secret key

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

then subtract it from 2nd point /coordinate in the pair

i.e. $P_m + kP_B - (kG * n_B)$

but we know $P_B = n_B \times G$

So $= P_m + kP_B - kP_B = [P_m]$ (orignal point)

So, Recivees gets the same point.

④ Hash function and Digital Signature

* Authentication functions

Authentication ⟶ verifying the user identity

An authenticator must be there to authenticate the message

Types of Authentication ↑ Types of function produce Authenction

(i) message encryption
(ii) MAC (message authenctication code)
→ we will have some authencation function and we apply them on the plaintext along with the key which produces a fixed length code called MAC.

$C(m,k)$ ⟶ Fixed length code (MAC)

C ← authentication function, m ← message, k ← key

This will acts as an authenticator here.

(iii) Hash function (H)

↓

Independnt of key

$H(\vec{m})$ = Fixed length code (Hash code 'h') acts as an authenticator

($\vec{m}$ → msg)

(Refer the PPT Diagram)
for, (i) (ii) and (iii) → Types of Authencator

* Digital Signature

The Digital Signature is a cryptographic techniques that verifies the authenticity, integrity and non-repudation of digital message or content or document.

It is created by applying a hash function to the message and then encrypting the hash using the sender's private key.

The Recivers can verify using sender's public key.

It work ons: (1) who sent message (authentication)
(2) the message not change (integrity)
(3) the sender cannot deny sending it (non-repudiation)

* Steps in Digital Signature

(1) Sender Creates message Digest
(2) Sender Encrypts the message Digest with their private key
(3) Sender sends the message + Digital Signature
(4) Recipee Decrypts the Digital Signature using Sender's public key.
(5) Recipees computes their own message Digest
(6) Recipees compares the two Digest message

* Hash Algorithm : MD5 (message Digest 5)

(1) fast and Produce 128-bit message digests
(2) the input text is processed in 512 bit blocks which is further divided into 16-32 bit sub blocks.
(3) the output algorithm is set of four 32 bit blocks which make up the 128 bit message digest

Working of (MD5)

(1) padding is done such that total length is 64 bit less than exact multiple of 512.

ex Orignal message = 1000 bit we add the Padding of 472 to make the length of 1472.

- these because if we add 64 bit to 1472 we get 1536 which is multiple of 512 (1536 = 512 x 3)
- Thus orignal message will have length of 448 bit 960 bit, 1472 bit if we add 64 then we will get

512 x 1 = 512
512 x 2 = 1024
512 x 3 = 1472 multiple of 512.

[Orignal message] ⊕ [padding] (1-512) bit

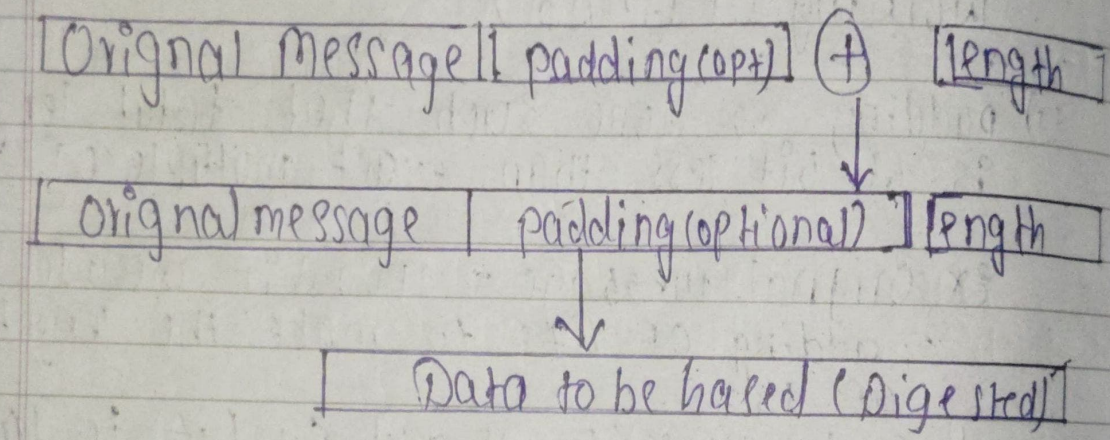
↓

[Orignal message | padding] (1-512) bit

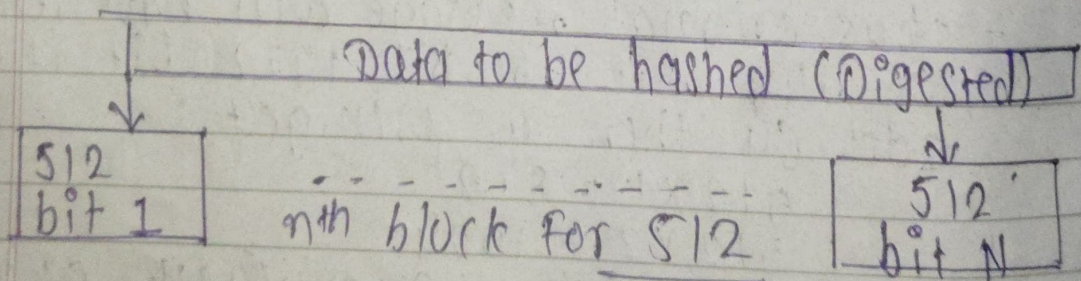
Total length of this should be 64 bits of less than 512

Step 2: Appending Length

(a) After padding bit are added
(b) Next step is calculate the orignal length of the message and add it to the end of the message after padding
(c) The length of message is calculated excluding the padding bit
(d) Orignal message is 1000 so consider only 1000.



Step 3: Divide the input into 512 bit blocks

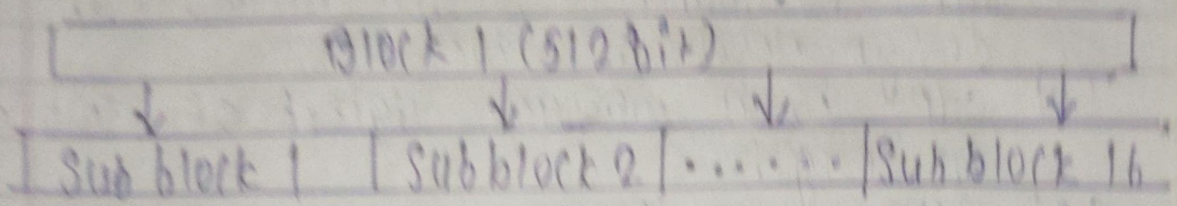


Step 4: Inatilized 4-chaining variable (32-bit) A, B, C, D

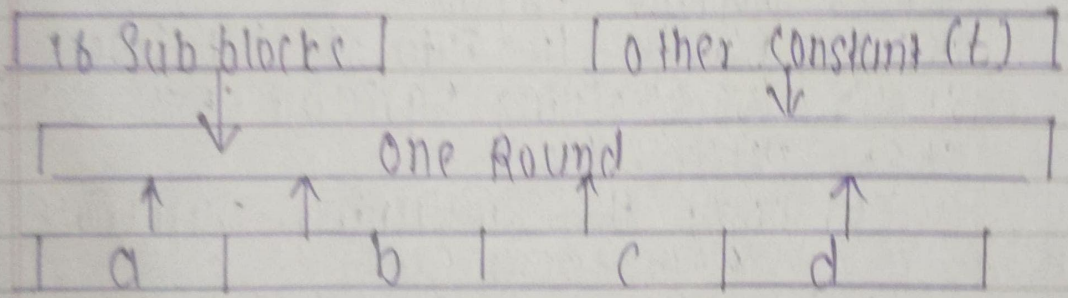
Step 5: Process Blocks

(a) Copy four chaining variable into four as, {A=a, B=b, C=c, D=d.}

(b) Divide the current 512 bit blocks into 16 sub block each sub block is 32 bit



(c) NO we have 4 rounds.
In each round we process all the 16 sub blocks belongs to a block
- Input to each blocks:
  (1) all the 16 sub-blocks
  (2) The variable a, b, c, d and
  (3) Some constants designed as t.



(Refer PPT Diagram)

mathematically Represented as:

$$a = b + ((a + \text{process } P(b,c,d) + M[i] + T[k]) \lll s$$

* Secure Hash Algorithm (SHA)

(a) SHA is modified version of MD5 and its design closely resembles MD5.
(b) O/P is a message digiest of 160 bits in length.
(c) SHA properties:
    (1) Generating orignal message from digest
    (2) Finding two message generating same Digest

Working of (SHA) → (same as MD5)

Step 1: Padding
Step 2: Append length
Step 3: Divide I/p into 512 bit blocks
Step 4: Five chaining variable (A, B, C, D, E) (5×32) = 160 bits
Step 5: Process block
    (a) Copy the chainning variable to A-E to a-e
    (b) Divide into 512 bit block into 16 sub blocks containing 32 bit
    (c) SHA has four rounds, each round consisting of 20 steps
    (d) This make it total of 80 iteration.

* Comparision Between MD5 and SHA

MD5	SHA
(i) Bit length 128	(i) Bit length 160
(ii) Attack to find	(ii) Attack to find

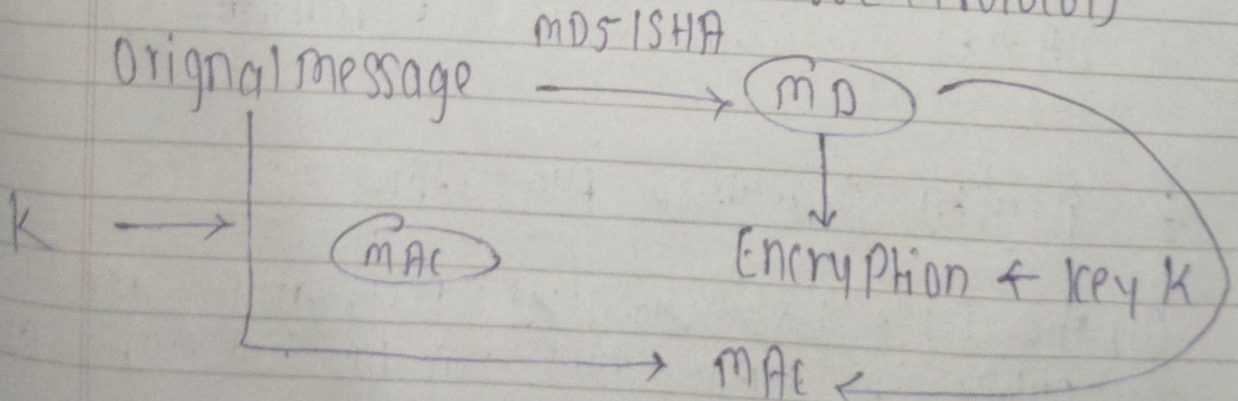
Orignal message is 2^{128} operations.	Orignal message is 2^{160} operations
(iii) Two message with Same MD 2^{64} operation	(iii) 2^{80} operations
(iv) Succesfull Attacks are Some reported incidents of MD5	(iv) Succesfull Attacks no such claims
(v) Speed is faster	(v) Speed is slower

* Hash-Based message Authentication code (HMAC)

(a) HMAC work with MD or SHA1

(b) It uses the shared Symmetric key to encrypt the message Digest which produces the output of MAC

(c) Used for Security implementation in internet Protocol (IP) and also in SSL (Protocol)



Working: (HMAC) (PPT Diagram)

Step 1: make the length of k equal to b
→ length of k<b (Add as many as 0 bit to left k)
→ length of k=b
→ length of k>b

Step 2: XOR k with ipad to produce S1.

Step 3: Append m to S1

Step 4: Message Digest Algorithm: Select MD algorithm is applied to the output of Step 3 and get output is H

Step 5: XOR k with opad to produce S2

Step 6: Append H TO S2.

Step 7: message Digest Algorithm

* Digital Signature Algorithm

(a) the first three variables p, q, and g are public in nature and can be sent across an insecure network.
(b) The private key is x and corresponding public key is y.

Working:

Step 1: The Sender Generates a random number k which is less than q.

Step 2: The Sender now calculate

$r = (g^k \bmod p) \bmod q$

$s = (k^{-1}(H(m) + xr)) \bmod q$

where r and s are Signatures of Sender. The Sender Sends these values to the Recivers

Step 3: To Verify the Signature, the recive calculates:

$w = s^{-1} \bmod q$

$u1 = (H(m) * w) \bmod q$

$u2 = (rw) \bmod q$

$v = ((g^{u1} * y^{u2} \bmod p) \bmod q$

* Digital Certificates and Public key Infrastructure (PKI)

(a) Related to the idea of Asymmetric-key cryptography.

(b) Includes message Digests, Digital Signatures and Encryption Services → Integrity, Authentication, non-repudation, confidentiality

(c) To enable this services Digital Certificates are required.

(d) PKIX and PKCS are two popular Standards for digital certificates and PKI.

* Digital certificate (Standard follow is PKI

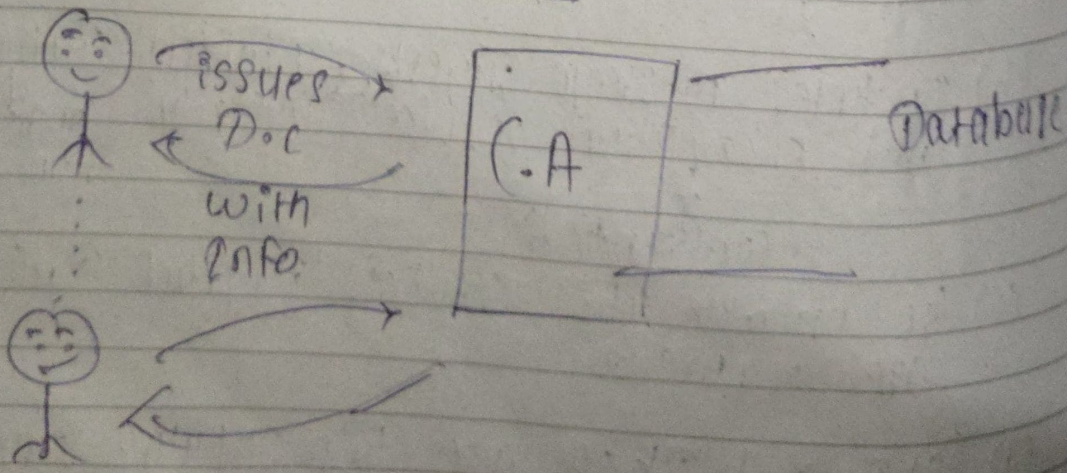
(a) Small file on computer / electronic device.
(b) File extension is generally (.cer), must be issued by trusted Authority
(c) DC establishes the relation b/w a user and the public key.

Sample Digital certificate

Subject Name	Pawan Rathod
Public key	<Pawan's key>
Serial Number	1021021
Other Data	& mail - pavem@123
valid from	30 Oct 2025
valid To	31 Oct 2026
Issuer Name	VeriSign

* Certification Authority (CA)

(a) CA is Trusted Agency that can issues a digital certificates
(b) Field of CA (From PPT)
(c)

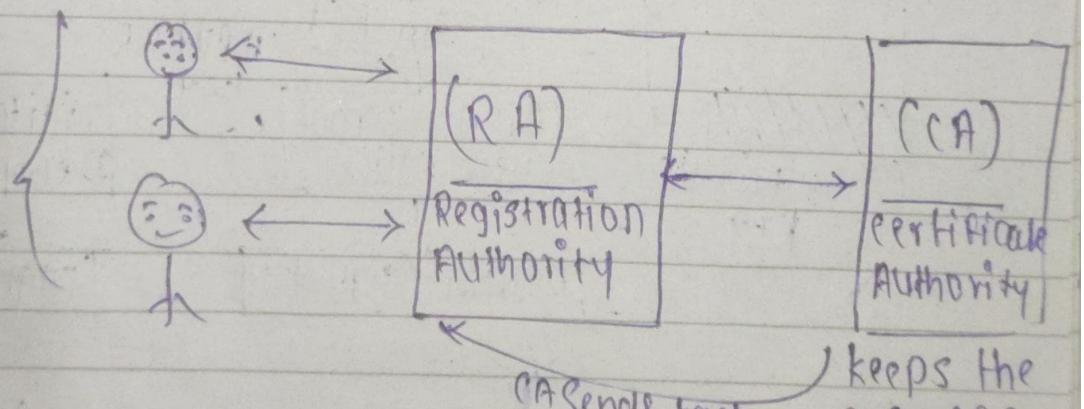


* Digital Certificate Creation (RA)

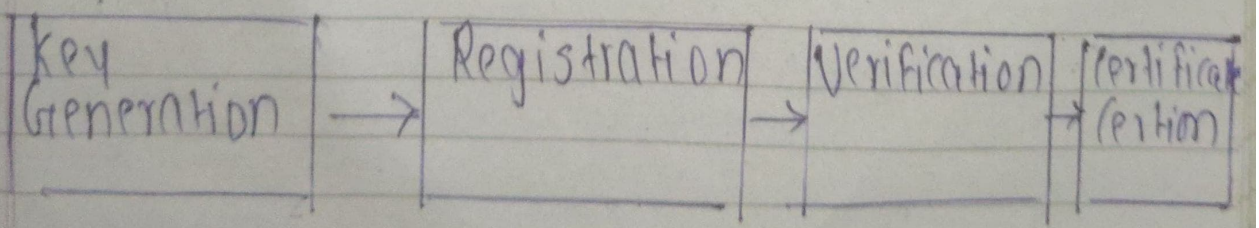
➡ RA is an intermediate between end user and CA which assist to CA in its day to day activity

* Registration Authority

(a) Assists the CA in its daily-work
(b) Services of RA:
  (1) Accepting & verifying Reg info about new users.
  (2) Generating keys on Behalf of users.
  (3) Key Backup and Recovery.
  (4) certification Revocation



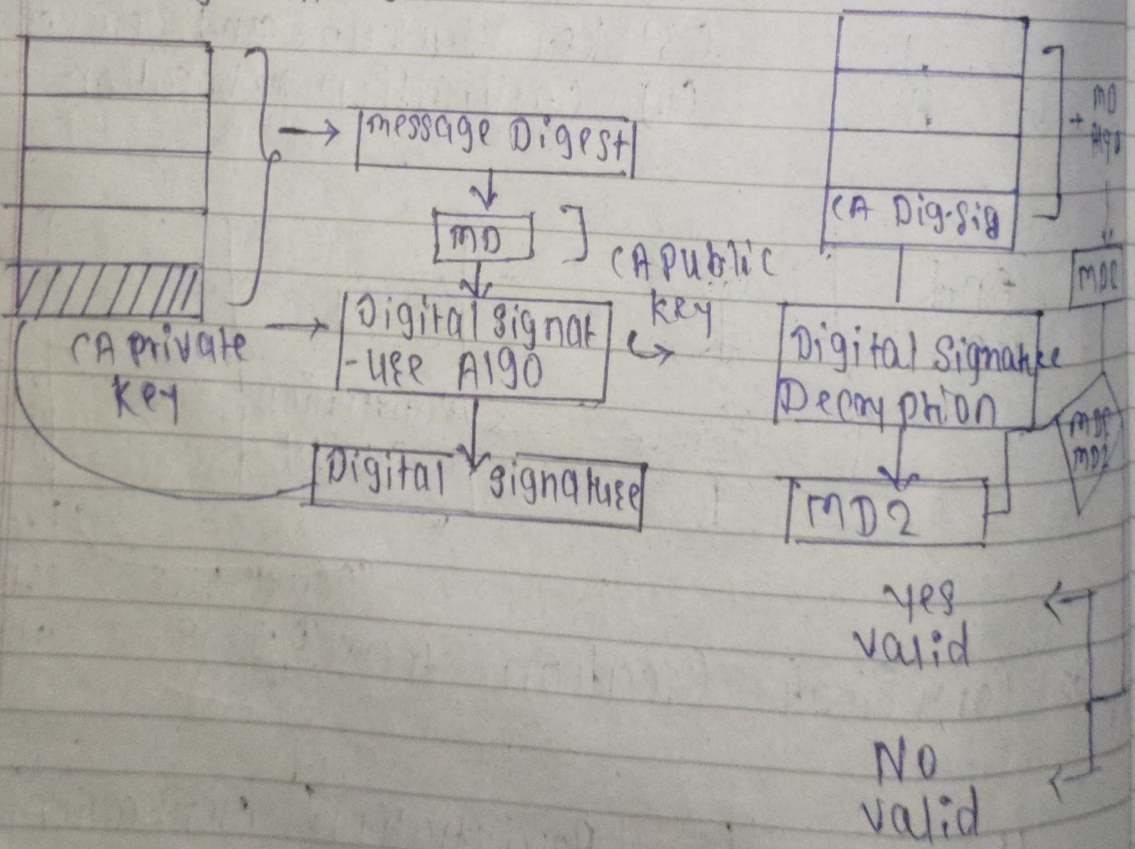
* Certificate Creation Steps



(a) Create private/public key pairs.
(b) Private key is kept secret, public key is send to (RA), with other details
(c) RA registers the user
(d) RA verifies credintals, also verify that user who sends public key contains corresponding Private key.
(e) RA pass all Details
(f) Hence, certificate is created

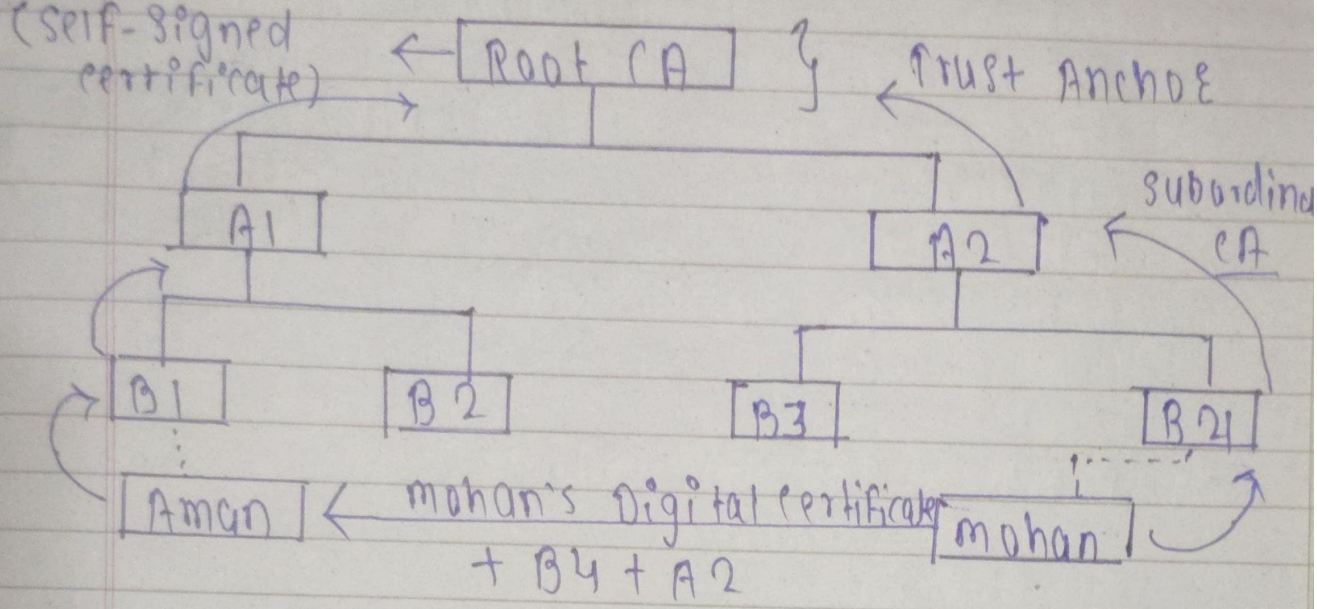
* Verifying Digital certificates

CA Sign the D.C.

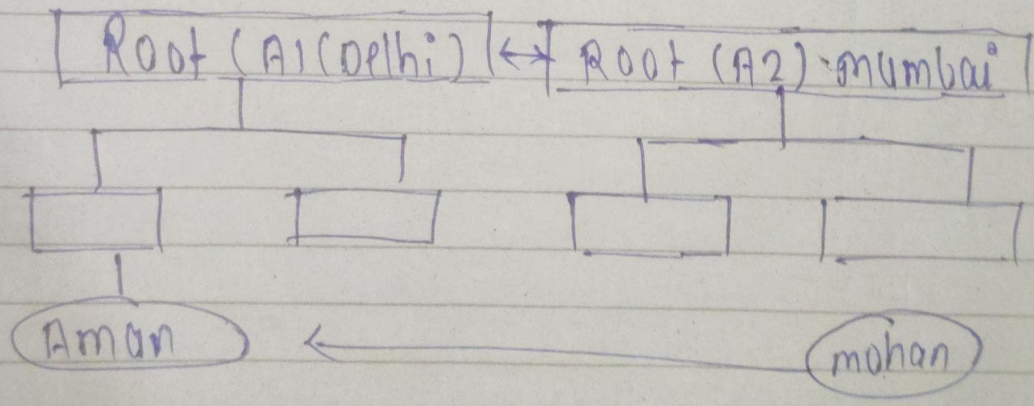


* Certificates Hierarchies and Self-Signed Digital certificates

(a) Also called as chain of Trust
(b) All the CA's are grouped into multiple levels of CA hierarchy.



* Cross Certification



Root CA1 has obtained a certificate for itself from Root CA2 and vice versa

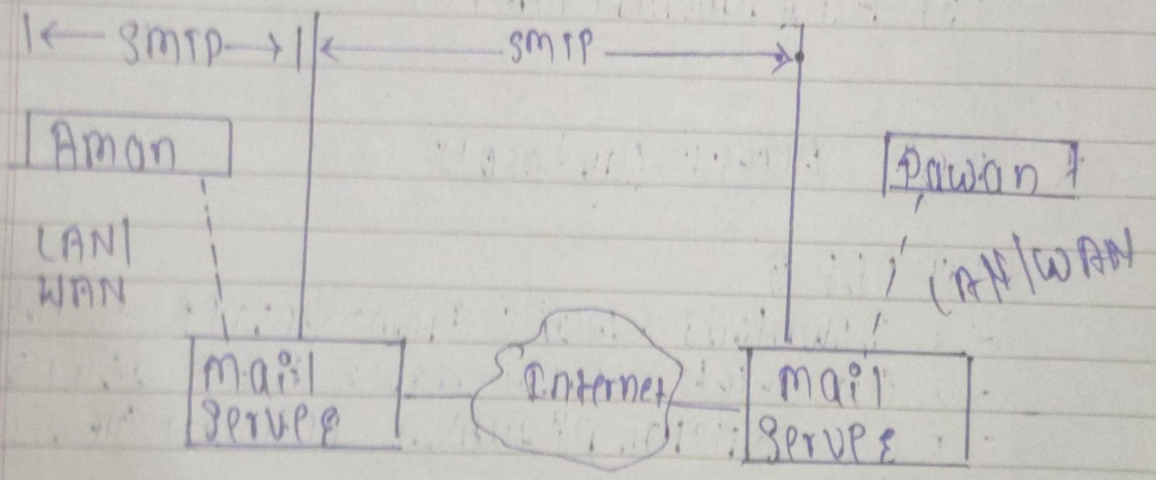
(5) Email and Web Security.

* Email Security ensures that all email communication is confidental, authenciated, and protected from attacks.

* SMTP (Simple Mail Transpoat protocols)

(a) Smtp is a standard protocols used for Sending emails over the intert.

(b) Smtp uses commands and Responses to transfer message blw an MTA client and MTA Server



(c) Example of commands:

(1) HELO
(2) MAIL FROM
(3) RECPT TO
(4) DATA
(5) QUIT
(6) VRFY
(7) HELP
(8) RSET

Example of Response

(1) positive condition Reply
(2) Transient Negative completion Reply
(3) permenant Negative completion Reply

(d) mail Transfer Phase:

(1) connection establishment
(2) message Transfer
(3) connection termination

* MAIL message format

(1) Header
(2) Body
(3) Attachments (MIME)

* mail Access protocols

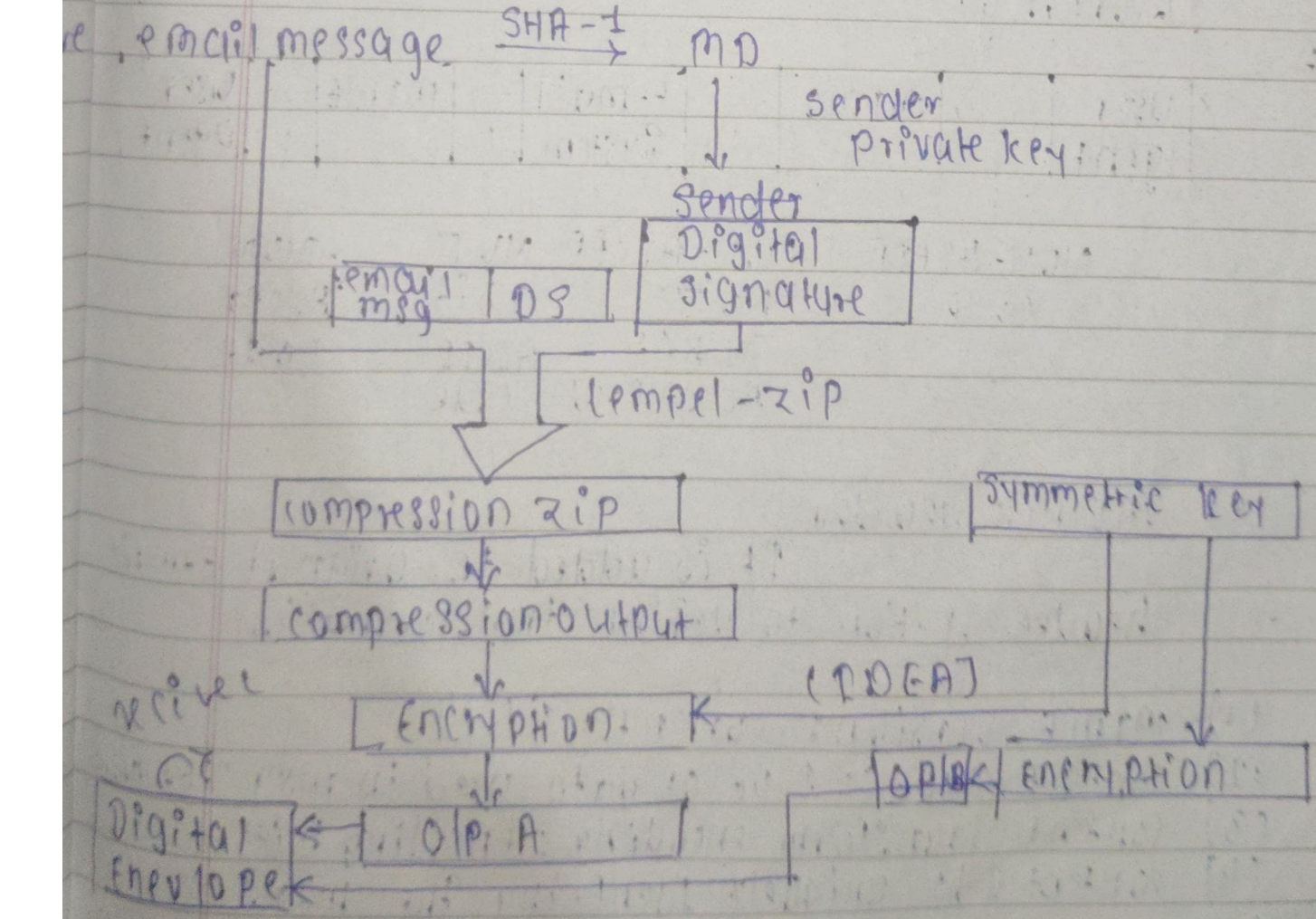
(1) POP 3
→ Downloads emails from server to local Drive
→ can delete from server after downloads
→ NO Synchronization between Devices

(2) IMAP

→ keeps mails on server
→ Support multiple device Synchronization
→ folder Support (Inbox, Sent, Trash)
→ User can downolds files partially
→ User can search content of files
→ User can check emails header prior to downloding

* Pretty Good Privacy (PGP)

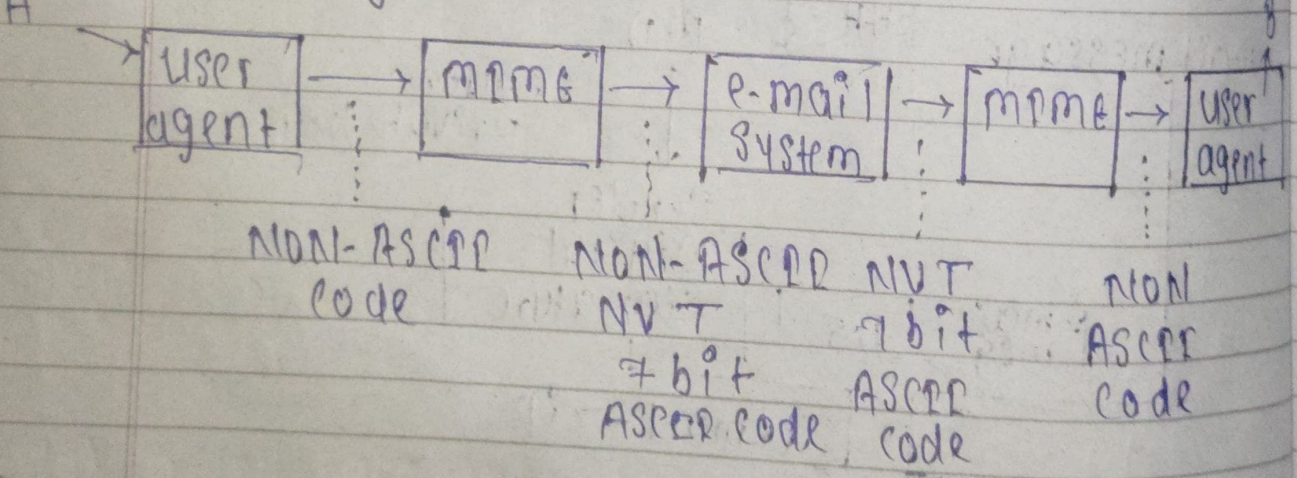
(a) PGP is one of most powerful tools used for email security.
(b) Developed by Philip Zimmerman
(c) PGP is an email encryption and authencation system that uses a combination of:
  (1) Symmetric key encryption.
  (2) Asymmetric key encryption
  (3) Hashing
  (4) Digital Signature
  (5) Compression
  (6) Segmentation
(d) PGP Working:
  Using public key and private key



* MIME (Multipurpose Internet mail Extension)

(a) mime is a standard which porposed by Bell communication in order to expand the limited capaibilites of email.
(b) mime is a supplementry protocol or a add on which allows non-ascii data to be sent through email (using SMTP)
(c) It allows users to exchange diffrent kinds of data files on the internet like; audio, video, images, etc.
(d) MIME was designed mainly for SMTP, it content type are also important in other communication protocols.

Working:



(e) MIME Header:

It is added to orignal e-mail header section to define transformation

(1) MIME version: (urrently (1.1)
(2) Content Type : Type of data used in msg like audio
(3) Content Transform encoding : encoding bits.
(4) Content Id: Uniquely Identfying the msg.
(5) Content Description: Defines actual img, videoetc

* S/MIME (Secure/multipurpose Internet mail extension)

(a) provides security for commercial emails
(b) Extension of MIME protocol
(c) It is widely accepted method (or more precisely, a protocol) for sending digitally signed and encrypted msg.
(d) S/MIME is based on asymmetric key encryption
(e) Functions: (1) Authecation
(2) message integrity
(3) NON-repudation of signature
(4) Privacy
(5) Data security
(f) In short, S/MIME is a protocol used to encrypts emails and digitally sign them.
(g) It provides 2 security serves

(1) Digital Signature (Provide auth + non repudation)
(2) Msg encryption (Provide confidentally + data integrity)

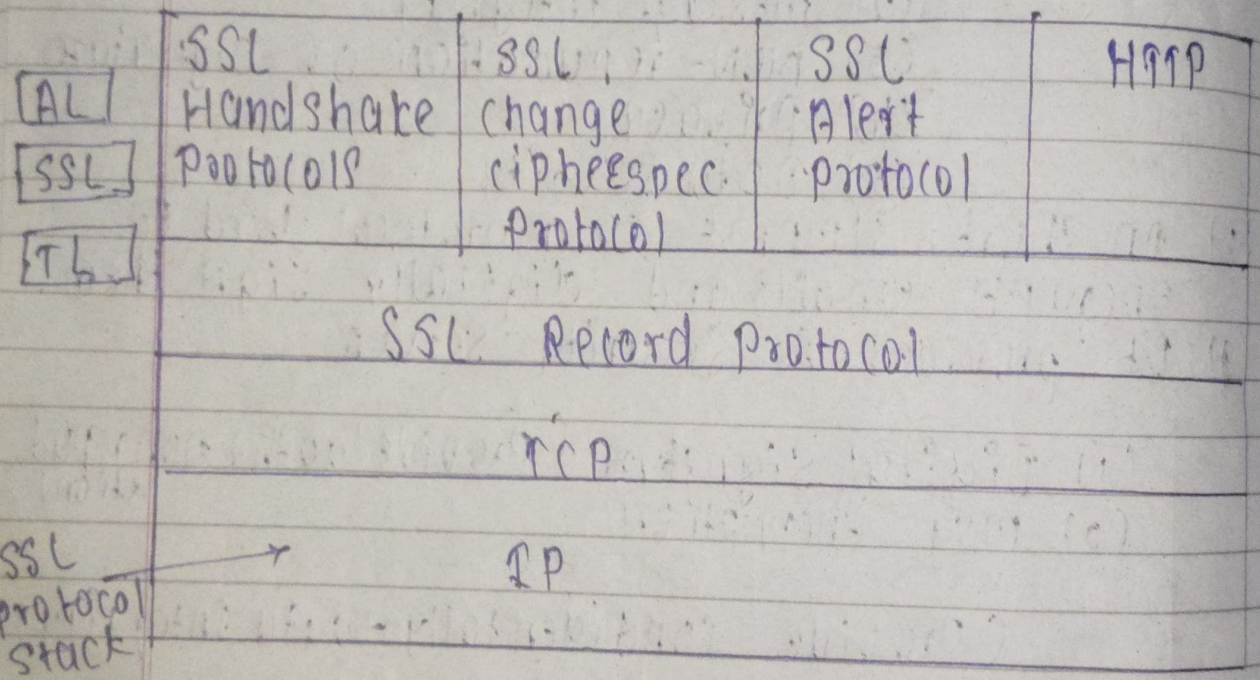
* SSL (Secure Socket layer)

(a) SSL (Secure Socket layer) is a cryptographic protocols that provides a secure communication between a client (browser) and server (website)

(b) It provides two basic security services "Authentication" and "confidentality"

(c) SSL is between Application layer and Transport layer

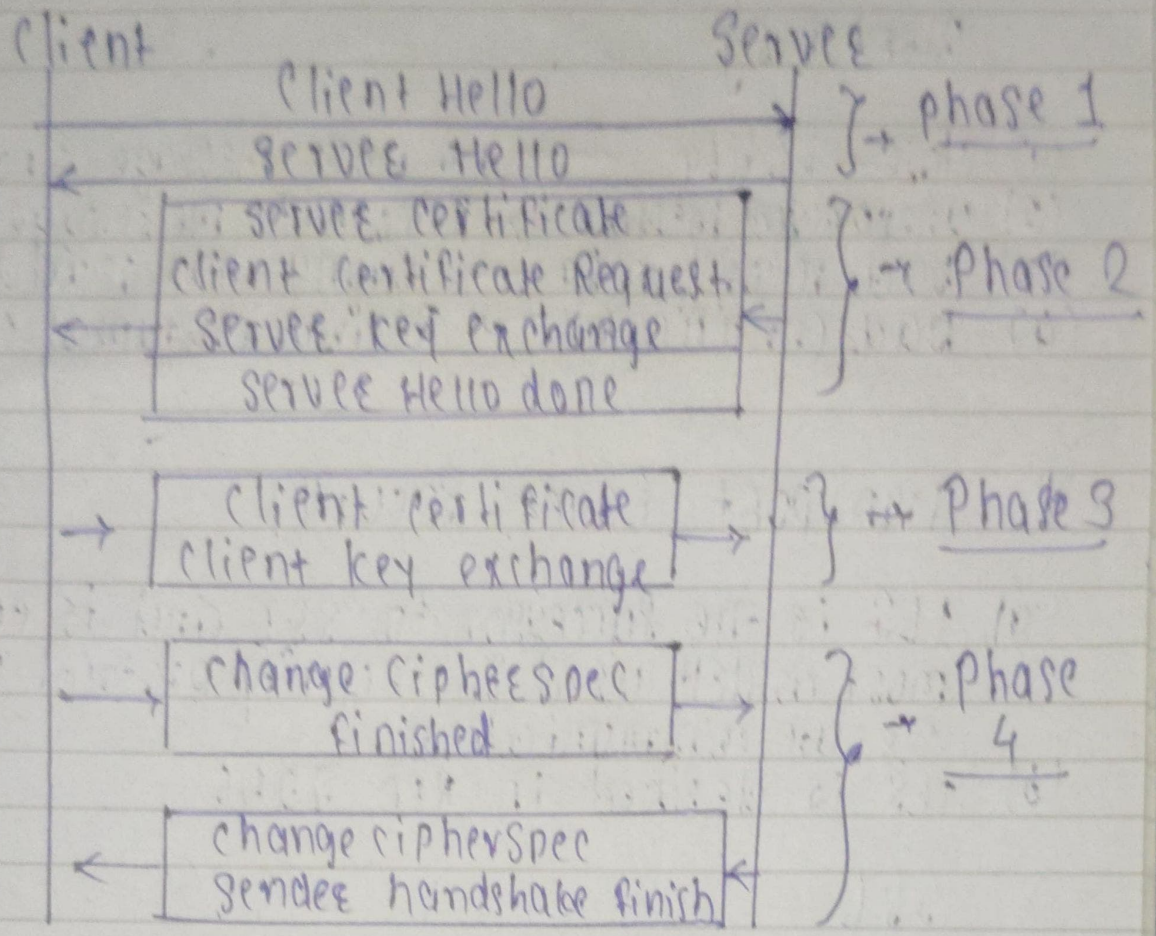
(d) SSL Architecture



(1) Handshake protocols:

It is first sub protocol used by client and the server to communicate using SSL enabled connection.

(a) SSL version
(b) session id
(c) cipher suite
(d) compression method.



Type	length	content
1 byte	3 byte	1 byte or more

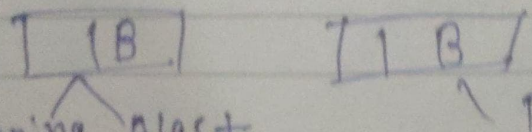
(2) Record Protocol

(a) Record protocols come after Successful Handshake is completed.

(b) the protocols provides two Services:
(a) confidentiality (2) Integrity

(c) Refer PDT (Diagram)

(3) Alert protocols: When either client or Server detects an error, the detecting party Sends an alert message to another party



Alert msg

(1) close_notify → no more message sendee
(2) unexpected message → Incorrect message recived
(3) bad_record_msg → wrong mac recived
(4) bad_certificate → Recived corrupted certificate

* Transport Layer Security

(a) TLS is the successor to SSL and is currently most widely used in security protocol for secure communication.
(b) TLS is defined in RFC 2246

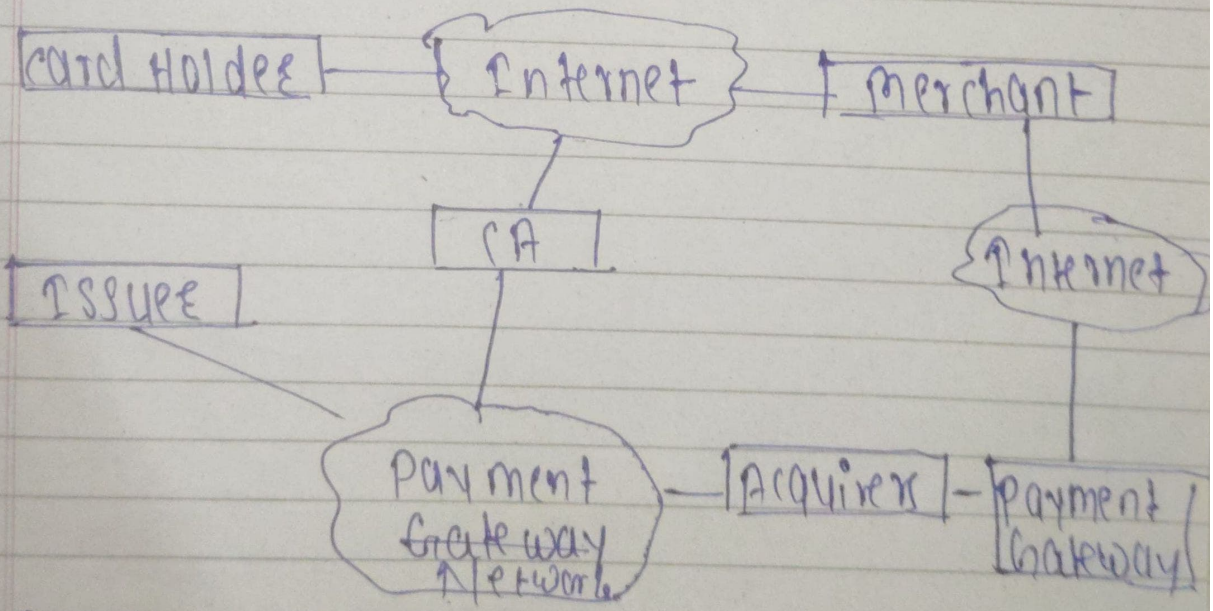
SSL	TLS
version 3.0	version 1.0
ciphee suite → Fortezza	NO
cry. secret → M.D. to generate master secret	Pseudo-random fuction mathee
Record protocol MAC	HMAC
Complex	Simple

* SECURE ELECTRONIC TRANSCATION (SET)

(a) open encryption and security specification designed to protect credit-card transcation over the internt

(b) services of SET:
- (1) provides secure comm^{n} chanel
- (2) provides auth^{n} → D.C.
- (3) ensures confidentality

(c) SET Participants:
- (1) cardholder
- (2) merchant
- (3) Issuer
- (4) Acquirer
- (5) payment Gateway
- (6) certificate Authority



Requirment in Set

(1) mutual authencation
(2) payment / order Info
(3) NO message modification
(4) Inter peraBility