Date:
Page:

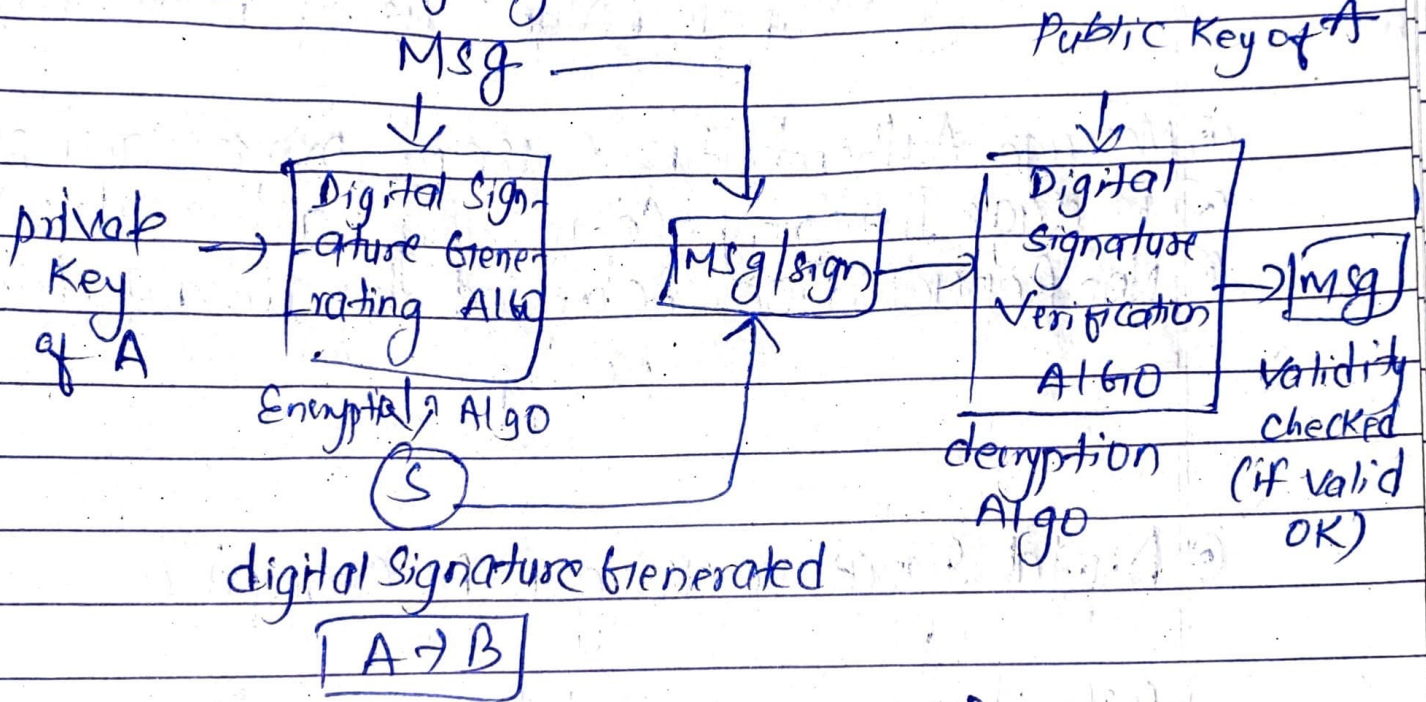
Digital signature:

→ Imp role in e-commerce, online transaction etc
→ based on asymmetric cryptography
(*) encryption: private key
(*) decryption: public key

→ Used for:

(I) MSg Authentication
(II) Non Repudiation
(III) MSg Integrity

→ Not for Confidentiality



A → B

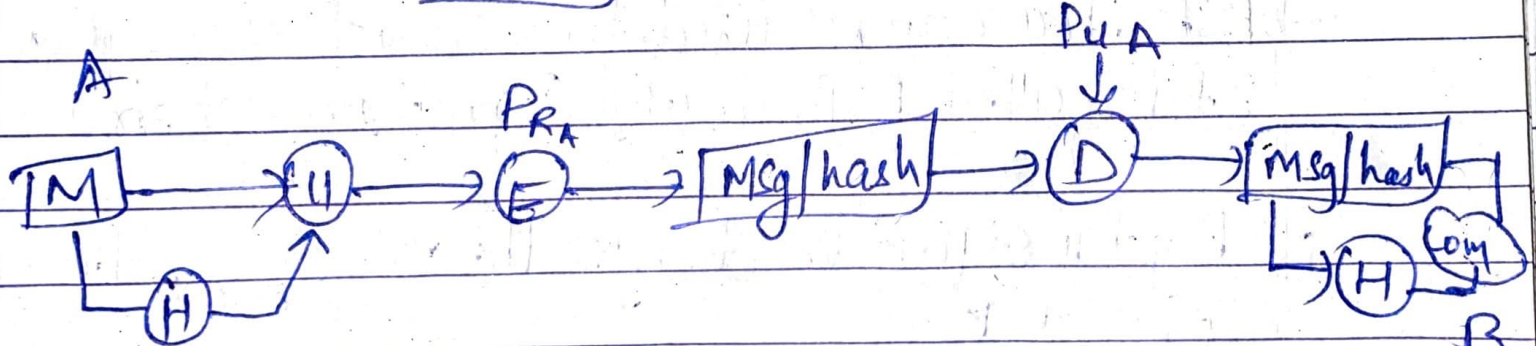


Fig: General Concept of signature.

Date:
Page:

→ must use some info unique to the sender, to prevent forgery and denial

→ must be easy to produce digital signature

→ must be easy to verify and recognize digital signature

We need : ① Key Generation Algo : to generate private key
② Signing Algo : i/p : M and private key
o/p : Digital Sign
③ Verifying Algo : using public key and sign

⊙ Message Authenticity : Use of private and public keys
⊙ Message Integrity : Comparing hash values
⊙ Non Repudiation : Achieved by using a trusted third party

⊙ Digital Signature using RSA Concept :

→ RSA idea can be used to sign and verify a msg. It is called RSA digital signature scheme.

→ Digital signature changes the role of public and private keys

① → Private and public keys of sender is used

Date:
Page:

(11) → the sender uses his/her private key to sign the document
→ and receiver uses the sender's public key to verify it

(6) Key Generation: (Same as RSA Algorithm)

Here,

$S = M^d \bmod n$ / In Normal RSA, $C = M^e \bmod n$

(6) Verification:

$M' = S^e \bmod n$ / $M = C^d \bmod n$

Compares M' and M

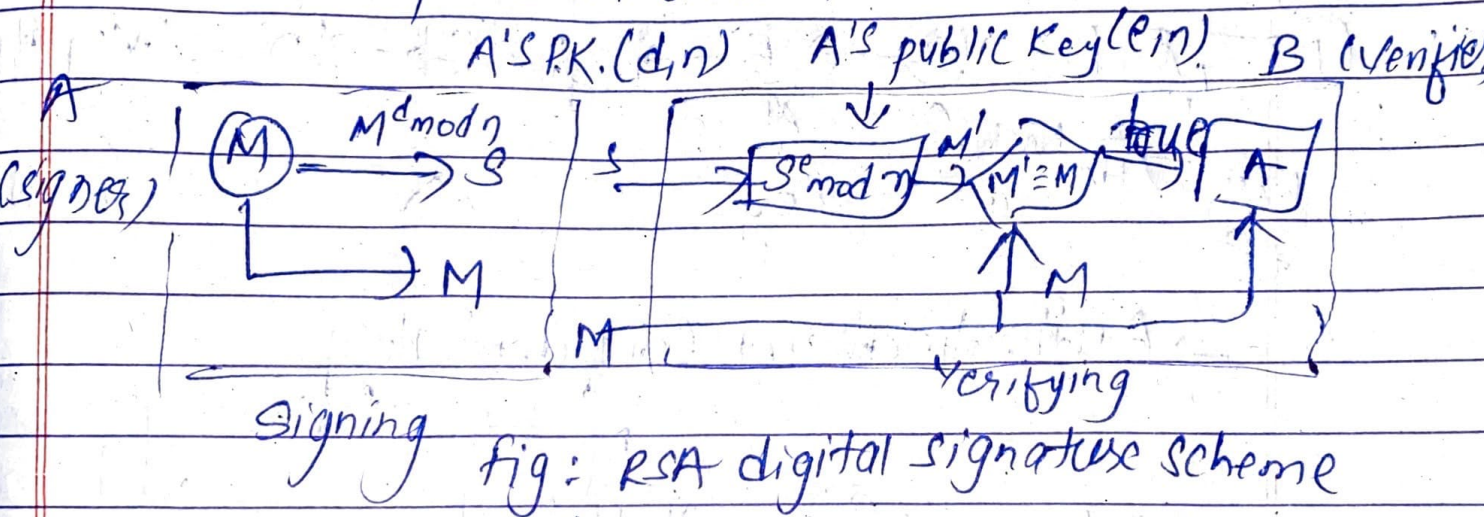


fig: RSA digital signature scheme

→ Achieve Authenticity:

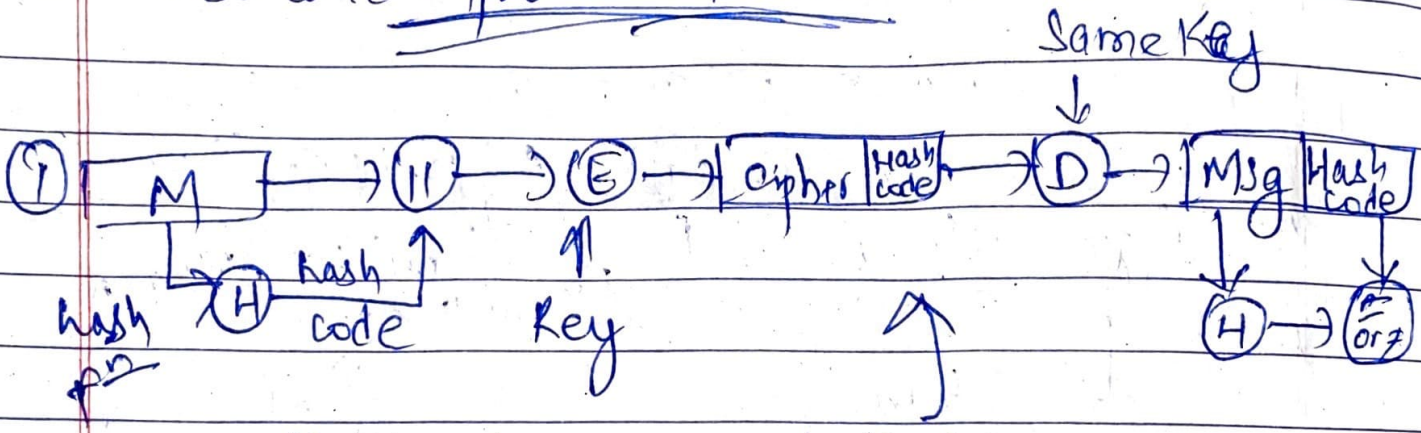
Date:
Page:

(III) ⊙ Hash functions in Cryptography:

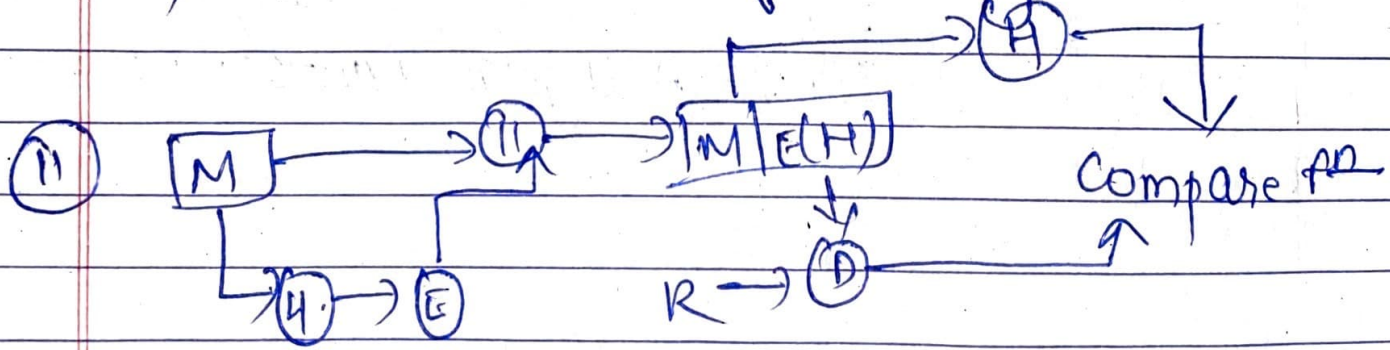
→ Takes in variable size message and produces a fixed output
↳ called hash code / hash value / or message digest

→ H(M.)= fixed length code 'h'
↑
variable length message

→ also called compression functions

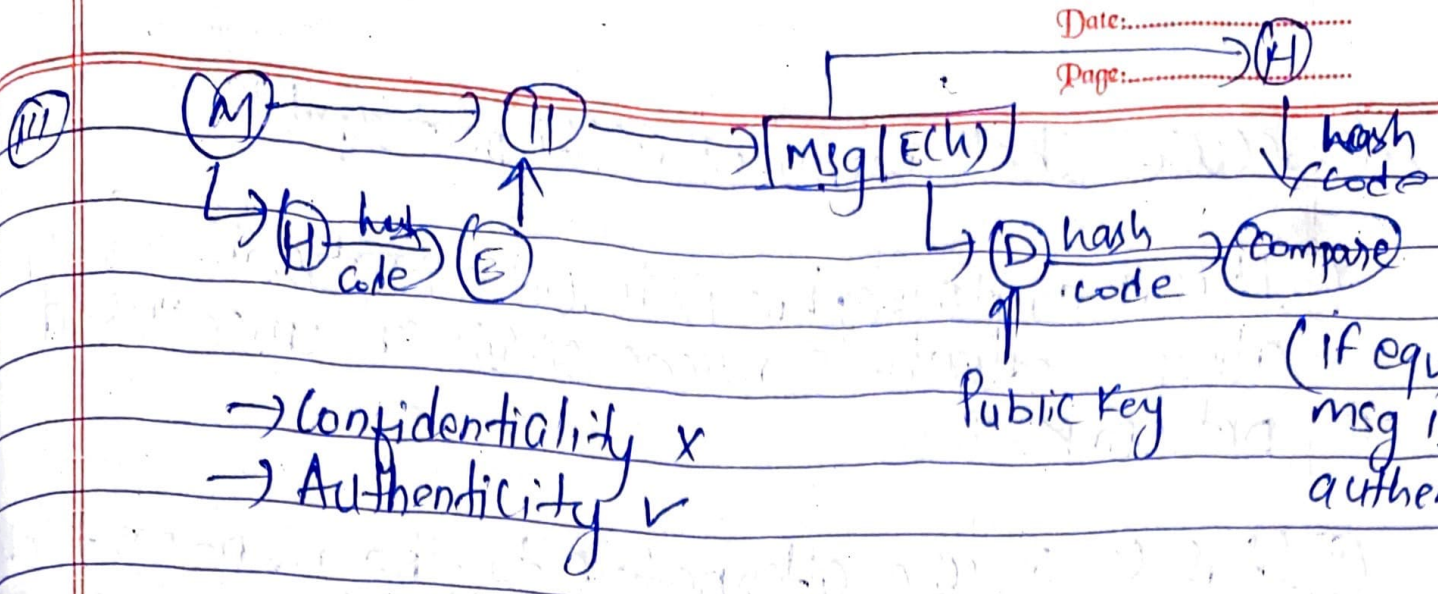


→ Authentication + Confidentiality

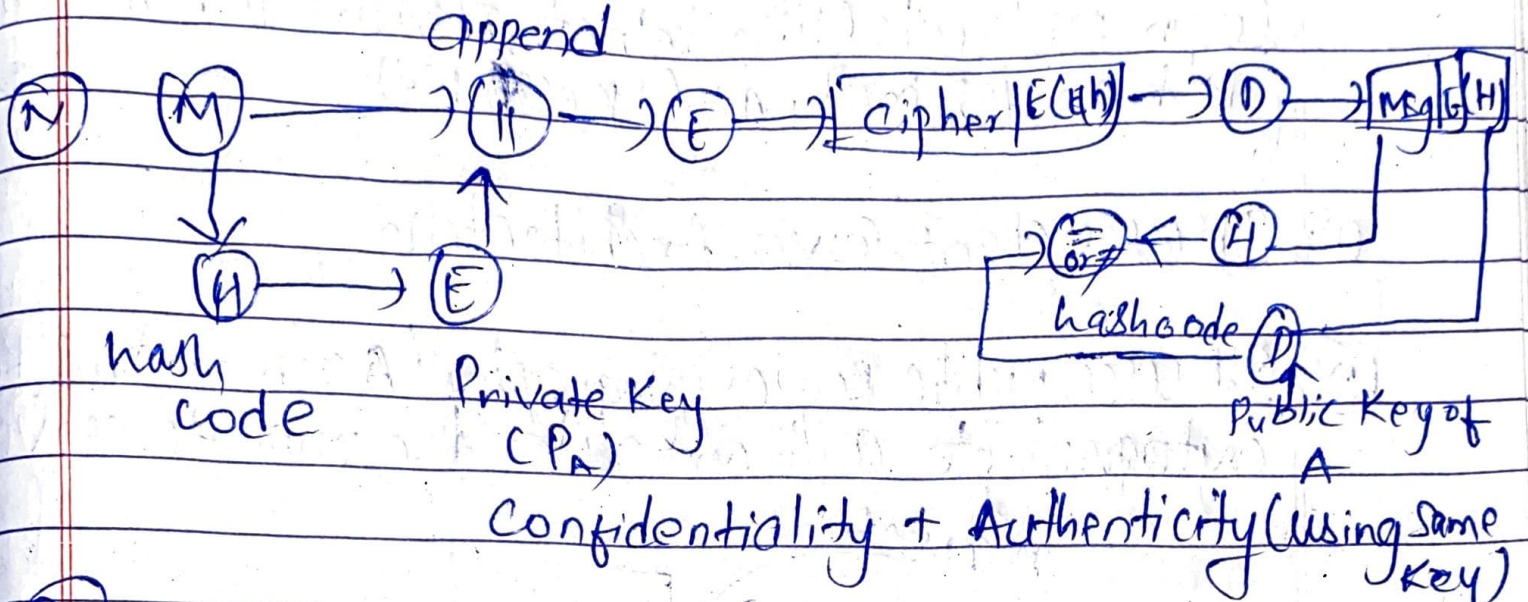


→ Authenticity but no confidentiality

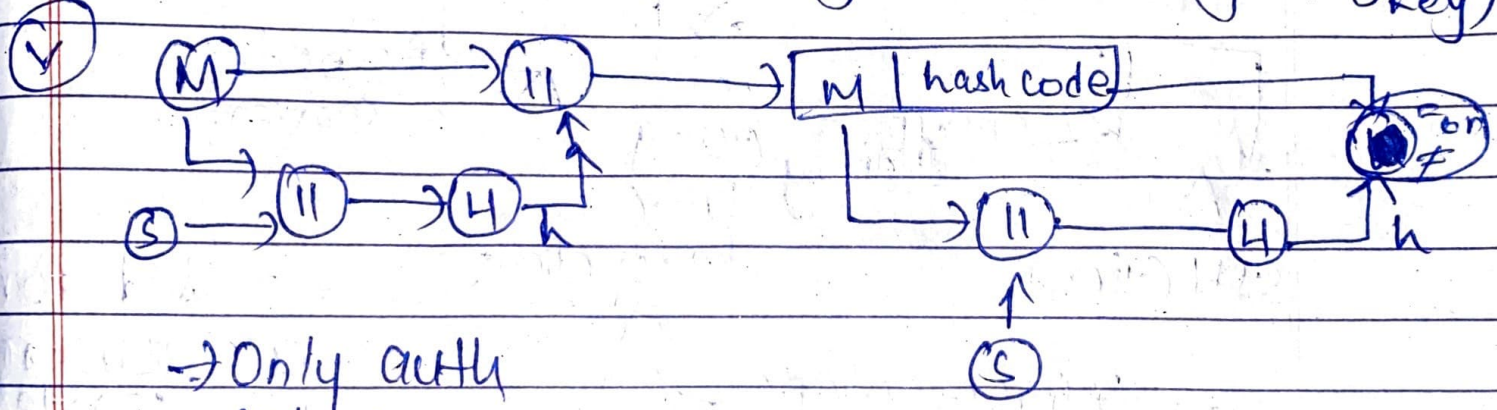
Date:
Page:



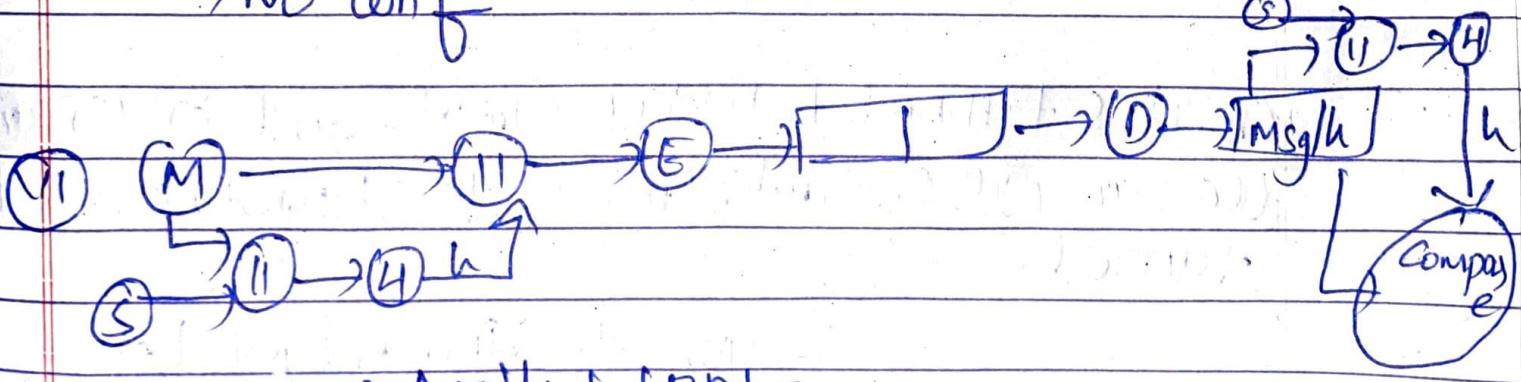
→ Confidentiality ✗
→ Authenticity ✓



Confidentiality + Authenticity (using same key)



→ Only auth
→ No conf



→ Auth + conf

→ MAC/Linux: Terminal
→ Windows: Putty

Date:
Page:

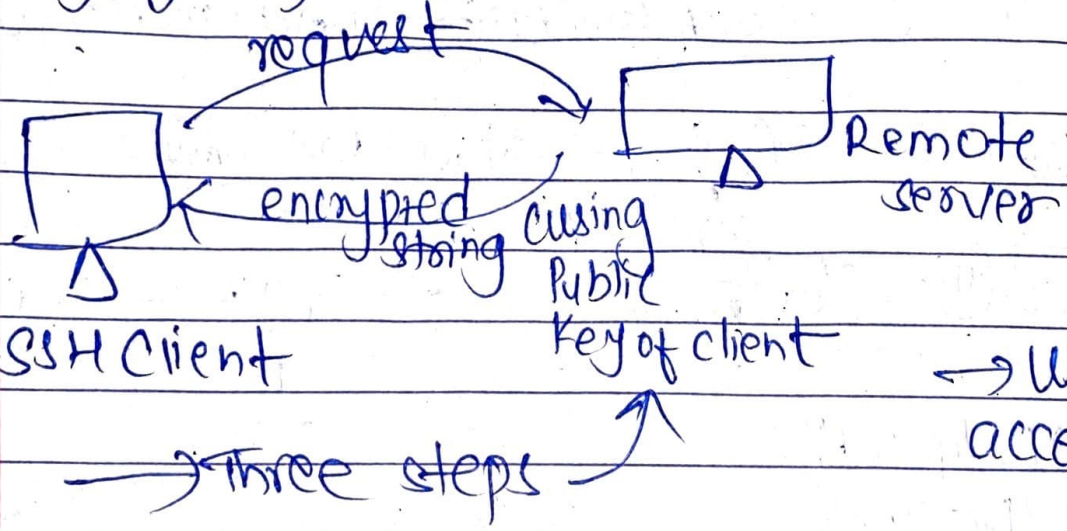
⓪ Secure Shell protocol (SSH protocol):

(i) It is a cryptographic network protocol for operating network services over an unsecured network.

(ii) It is a secure alternative to the non-protected login protocols (like Telnet, rlogin) and insecure file transfer methods (like FTP)

(iii) It uses Client Server Architecture

(iv) It uses public key cryptography / Asymmetric key cryptography to authenticate the remote server



→ Used for accessing remote servers

→ creates tunnel between Client and server and we can pass our data (in encrypted form) over this tunnel.

→ provides: (i) Confidentiality
(ii) Integrity of data

Date:
Page:

⑥ PGP: (pretty Good privacy):

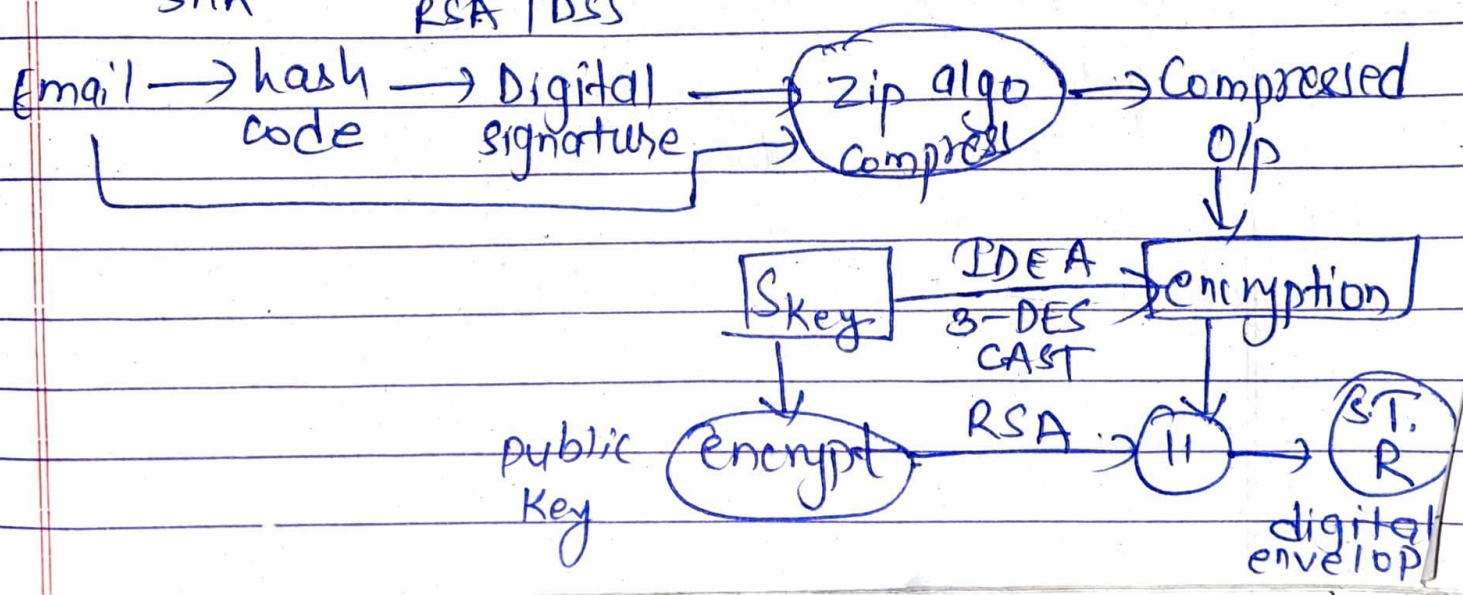
→ N/w security concept which provides email security

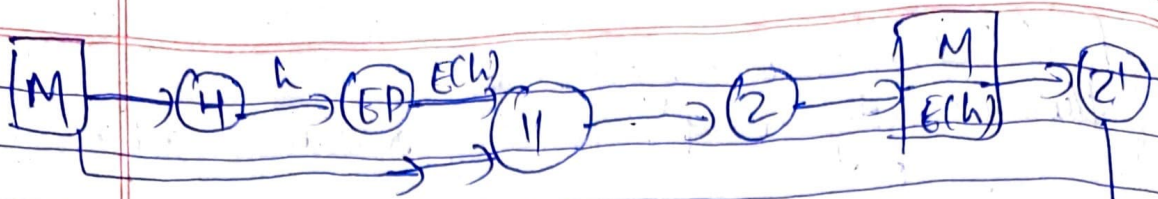
→ Encryption program that provides cryptographic privacy i.e. confidentiality and authentication for data communication

→ PGP is used for signing, encrypting and decrypting texts, email files, directories and to increase the security of email communication.

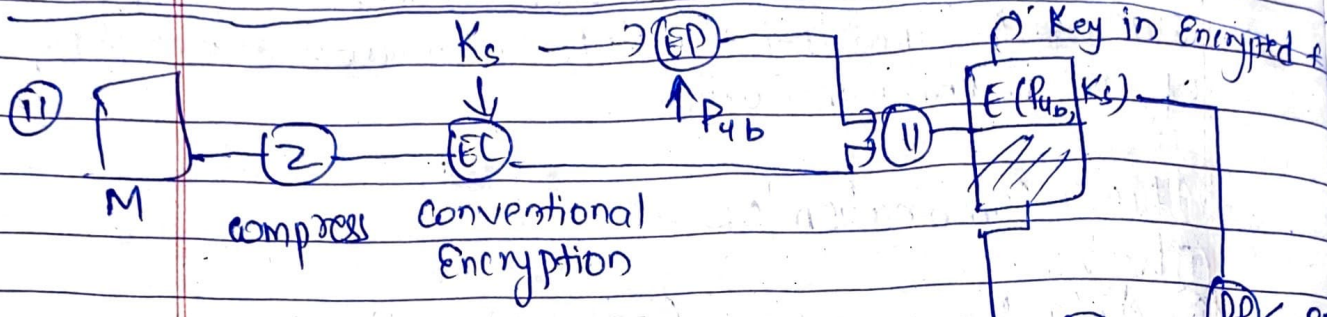
→ It is a combination of:
(i) hashing
(ii) data compression (Zip Algorithms)
(iii) Symmetric Key Cryptography
(iv) Asymmetric " "

→ email Compatibility using radix-64 encoding scheme





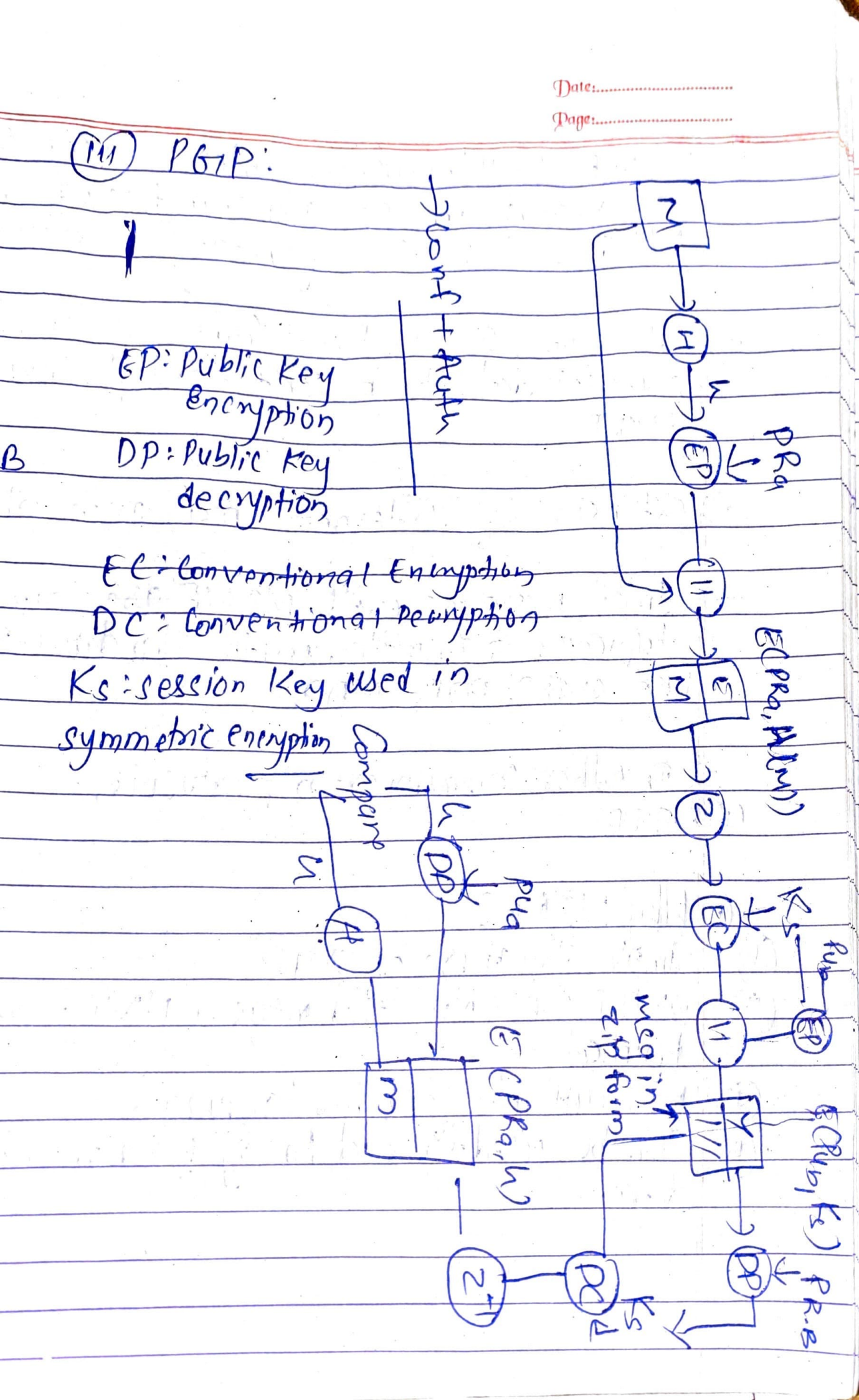
① Auth + D.S (No Conf)



→ Only Conf (NO Auth / No sign)

Date:
Page:

(11) PGP:



Date:
Page:

(*) MIME: (Multipurpose Internet Mail Extension)
→ to expand limited capabilities of email
→ NVT 7-bit ASCII format (previously)

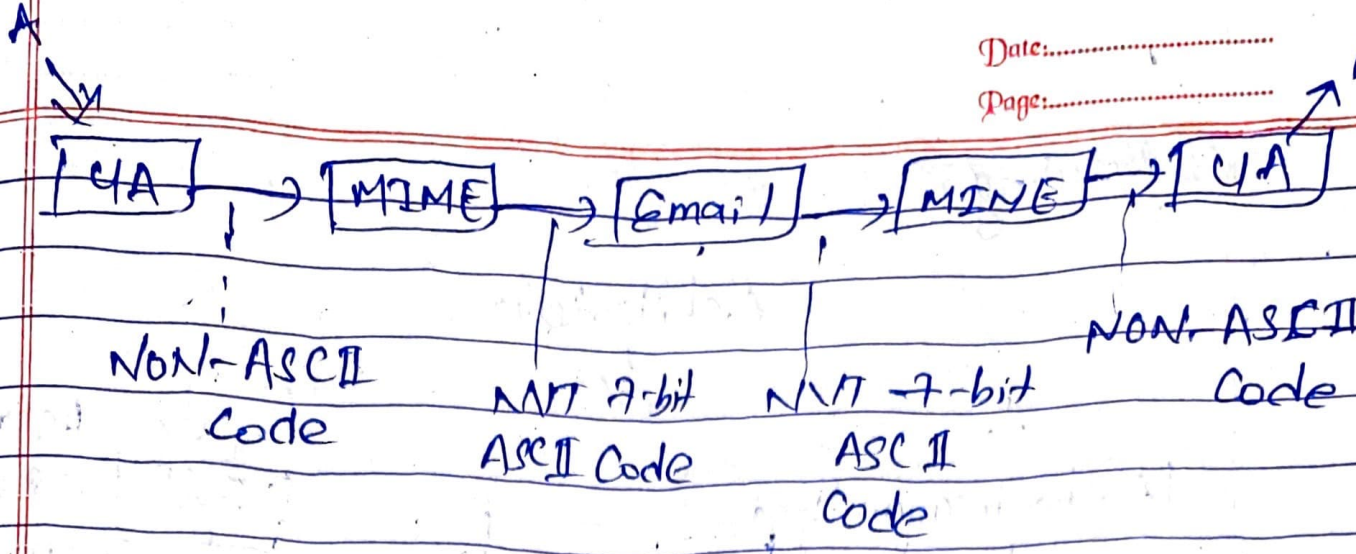
→ supplementary protocol / add on which allows non ascii data to be sent through email using SMTP

→ extension of Internet Email protocol

→ Email messages with MIME ~~are~~ formatting are typically transmitted with standard protocols like SMTP, POP, IMAP

→ In other communication protocols,
(1) HTTP

(*) Why MIME?
(i) Simple structure of SMTP
(ii) SMTP can only send msgs in NVT 7-bit ascii format
(iii) Not suitable for other languages
(iv) Can be used to send binary files or video or audio data.



→ MIME Header:

→ Added to original E-mail header section to define transformation

→ 5 headers:

(i) MIME Version
(ii) Content type
(iii) Content transfer encoding (8 bit / 7 bit)
(iv) Content Id
(v) Content description

→ Email header
(i)
(ii)
(iii) (iv) (v)
Email body

→ S/MIME:
- Secure
- Encrypting mails
- based on asymmetric encryption

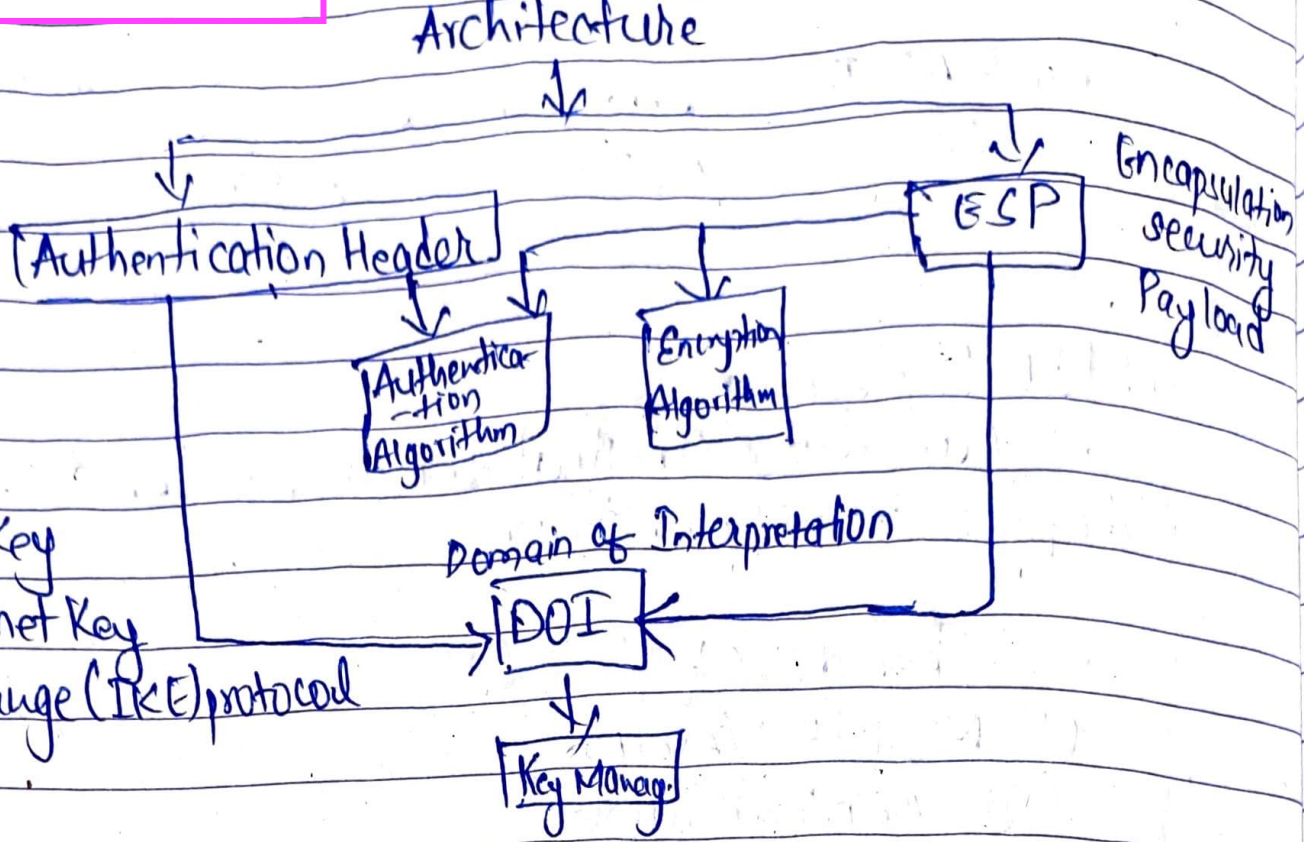
→ function:
(i) authentication
(ii) Message Integrity
(iii) Non-repudiation of Origin
(iv) privacy
(v) data security

→ Two services: (i) DS (Auth) (ii) Msg Encryption (Confi. Int.)

→ S/MIME is used now.
→ 1995, 1998, 1999
→ Outlook Express 5-01 & later
→ Exchange 5.5 & later

Date:
Page:

① IP Security :



- ipsecKey
- Internet Key Exchange (IKE) protocol

② Security Association :
(i) Security parameter Index
(ii) Security protocol Identifier
(iii) Sequence Index Counter (0 to $2^{32}-1$)
(iv) AH Information
(v) ESP Information
(vi) Life time of SA
(vii) IPsec Protocol mode
- Transport
- Tunnel

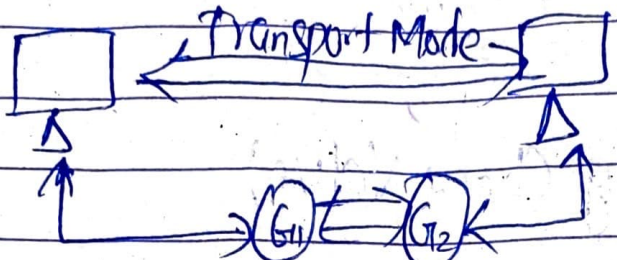
Date:
Page:

① Transport Mode

→ payload is encrypted but not IP header

→ | IP header | IPsec header | payload |

AH or ESP — encrypted



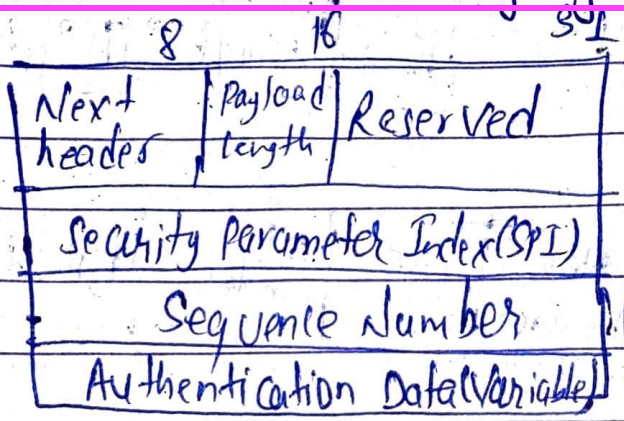
Tunnel Mode

Tunnel Mode

→ | New header | IPsec header | IP header | Payload |

Encrypted

② Authentication Header: (Integrity and Auth)



(0 to $2^{32}-1$)

IPV4 | IP header | TCP | Data |

IPV6 | IP header | Extension header | TCP | Data |

Before AH

| IP header | AH | TCP | Data |

| IP header | Ext header | AH | TCP | Data |

Transport mode

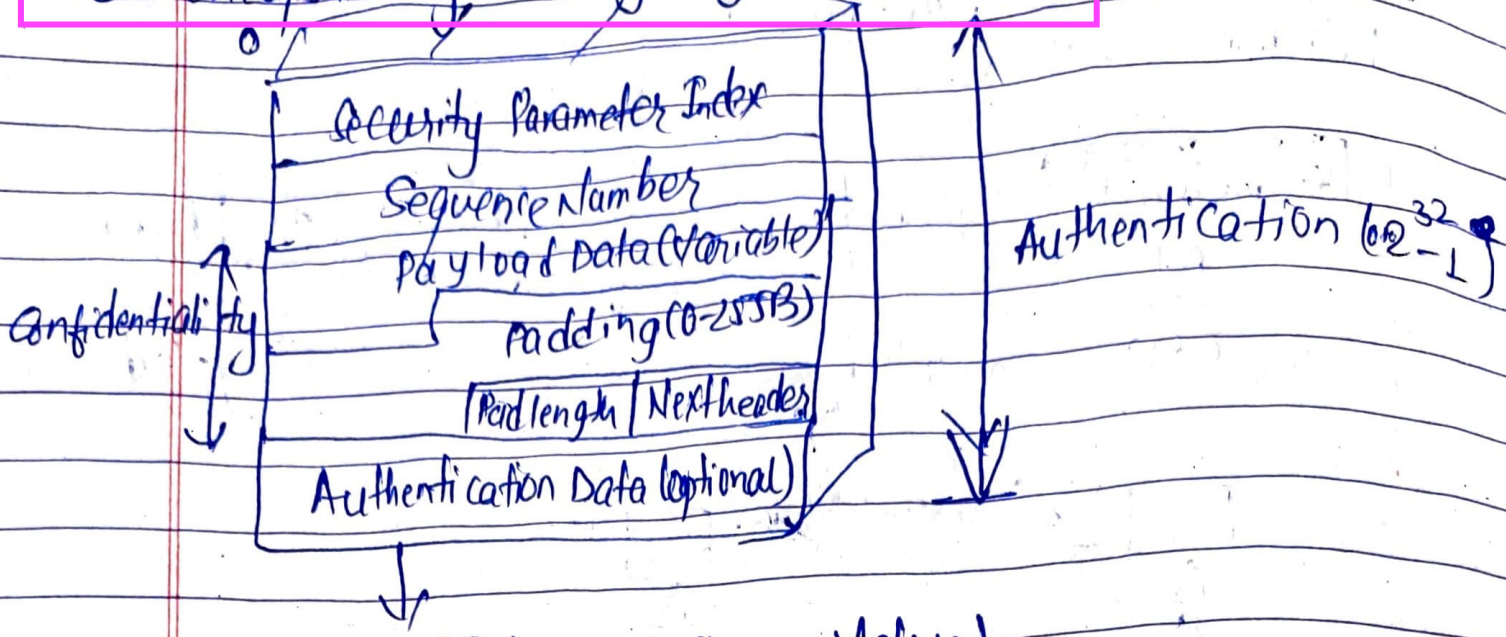
After AH

| New IP | AH | IP header | TCP | Data |

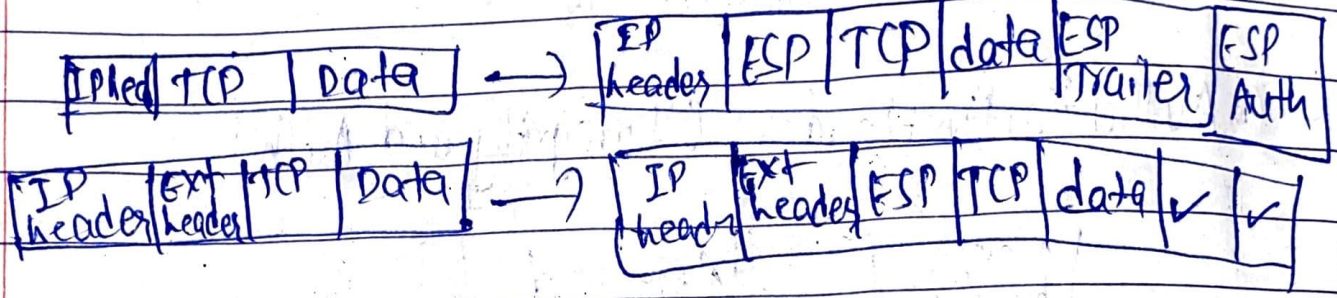
Tunnel mode { | New IP header | AH | | | | |

Date:
Page:

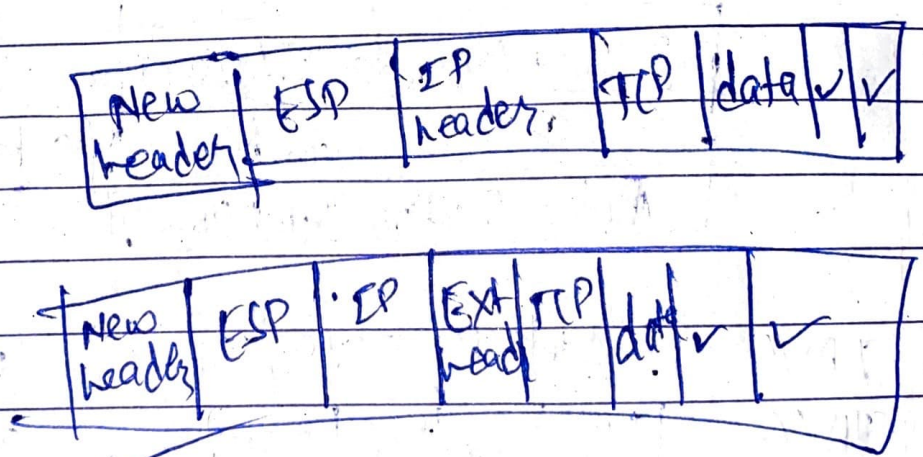
⑥ Encapsulating Security Payload:



Transport mode

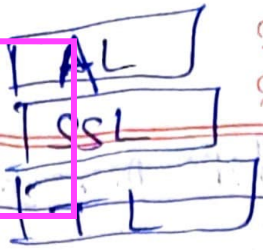


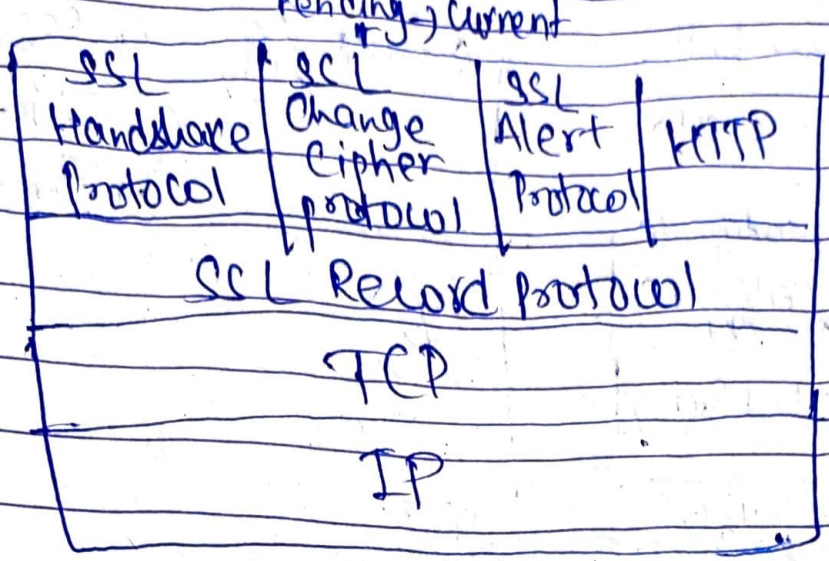
Tunnel



Date:
Page:

⓪ Secured Socket layer (SSL): (CIA)

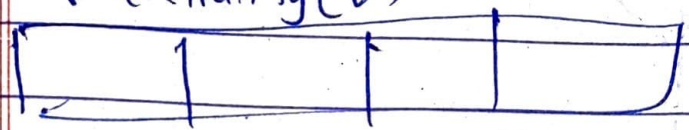


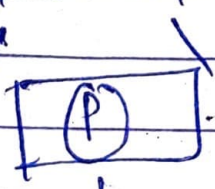


SSL protocol stack

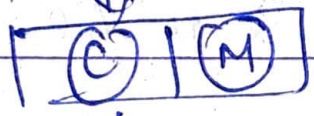
① SSL Record Protocol:

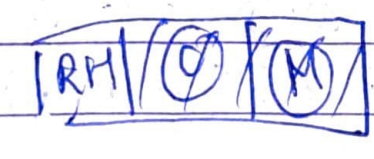
→ Confidentiality (✓)

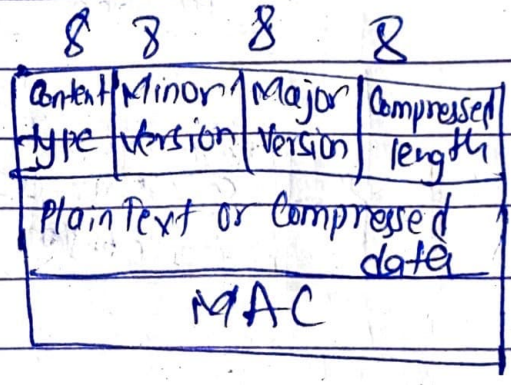




2^{14} byte







⊙ Content-type: The higher layer protocol to process the enclosed fragment

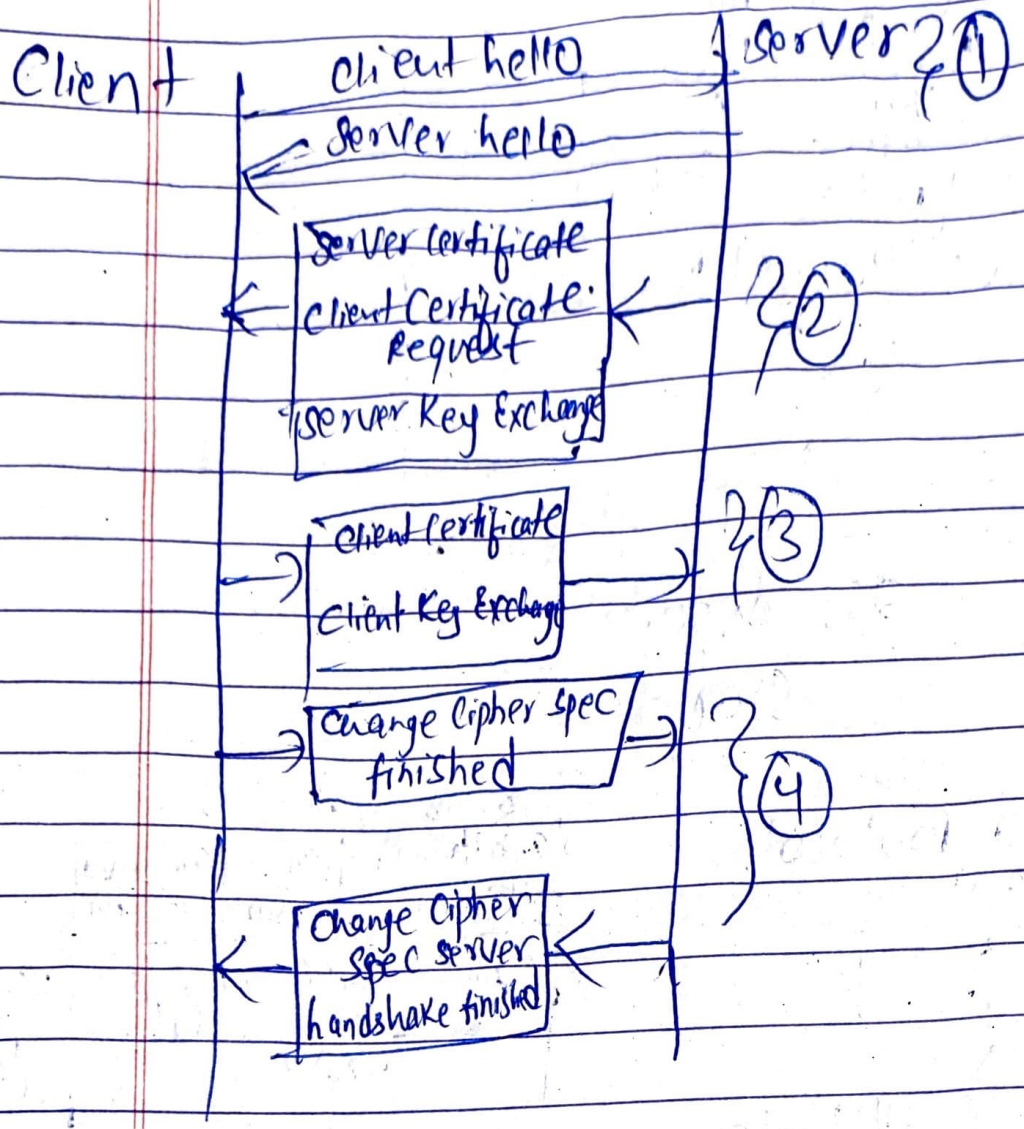
⊙ Major version: SSLV3 → 3

⊙ Minor version: value is '0'

⊙ Compressed: length of compressed fragment in bytes

Date:
Page:

① SSL handshake protocol: (Auth)



① SSL version
② session ID
③ Cipher Suite
④ Compression Metho

② SSL Alert Protocol:

1B	1B
level	Alert

(contains code that indicate the specific Alert)

level → Warning, fatal

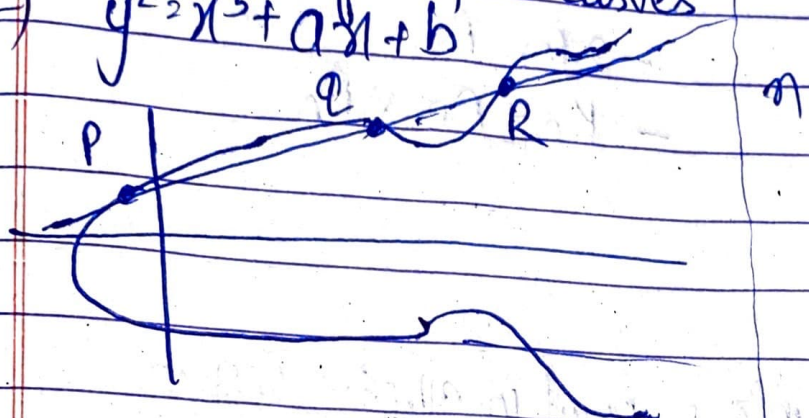
Alert → Types of Alert

Alert-Msg	Description
(i) Close_notify	→ No more message sender
(ii) Unexpected_message	→ Incorrect message received

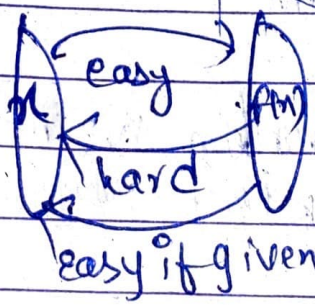
(iii) bad_record_mac → wrong mac received
(iv) bad_certificate → Received a corrupted certificate
(v) certificate expired

⑥ Elliptic Curve Cryptography:

→ It is asymmetric / public key cryptography
→ It provides equal security with smaller key size
→ makes use of elliptic curves
→ $y^2 = x^3 + ax + b$



→ Trapdoor function :



→ let $E_p(a,b) \Rightarrow Q = kP$ (discrete logarithm problem)

Date:
Page:

(112 → 512)
(160 → 1024)

⓪ Key-Exchange:

→ Global Public Elements:

$Eq(a,b)$ → q is prime no. of the 2^m or any prime integer

G → Point on the curve whose value is $> n$

→ User A Key Generation:
- → Private Key: n_A
- → public Key: $P_A = n_A * G$

→ For B
- → Private Key: n_B
- → public Key: $P_B = n_B * G$

→ Secret Key for A:
$K_A = n_A \times P_B$

→ For B
→ $K_B = n_B \times P_A$

⓪ For Encryption:

→ Let message be M.
→ First encode message into a point in elliptic curve
→ let this point be P_m.

For Encryption:
choose random +ve integer K;

$C_m = \{KG, P_m + KP_B\}$

→ For Decryption:
→ Multiply 1st point with receiver's secret key: $KG * n_B$

then,
$P_m + KP_B - (KG * n_B)$ But $P_B = n_B * G$

$= P_m + K P_B * G - K G * n_B$

→ P_m (original point)

Date:
Page:

ⓐ Euclid's Algorithm:

$gcd(a,b) = gcd(b, a\%b)$

$g(a,0) = a$

q	r_1	r_2	r
1	2740	1760	980
1	1760	980	780
1	980	780	200
3	780	200	180
1	200	180	20
9	180	20	0
	20	0	

Ans: 20

ⓐ Extended Euclidean Algorithm:

→ Given two integers 'a' and 'b' we often need to find two integers s and t such that

$s*a + t*b = gcd(a,b)$ $\quad s = s_1 - q s_2$, $t = t_1 - q t_2$

(Coeff of Bezout's identity)

				1	0		0	1	
q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	-5
1	28	21	7	0	1	-1	1	-5	6
3	21	7	0	1	-1	4	-5	6	-2
	7	0		-1	4		6	-2	
				s			t		

Date:
Page:

Multiplicative Inverse:

Find MI of 11 in Z_{26}

$t = t_1 - q t_2$

q	r_1	r_2	r	t_1	t_2	t
	26	11				

(i) if (-ve), then add Z's subscript

$11 * 9 \bmod 26 = 1$

$\gcd(m, n) = 1$

✓

MI (possible)

otherwise not