Date:- 26/08/19

Message Authentication

Approaches to Message Authentication:-

Authentication Requirements.

Attacks ~~across~~ can be identified across a network those are!-

1) Traffic Analysis:- passive attack (connectionless)

2) Mask rate :- Active attack
unauthorized accys

3) Content modification :- original content is modified.

4) Sequence modification:-

5) Timing modifications:-

6) Source repudation :- Deniel of message transmiss by source.

Destination repudation :- Denial of reciept of message by destination.

Authentication functions

- these are divided into 2 levels
  - (i) Low level Authentication (sort of function)
  - (ii) High level Authentication
    (protocols are used to identify
    Authentication of source and destination)

(1) Encryption

(2) MAC

(3) Hash function.

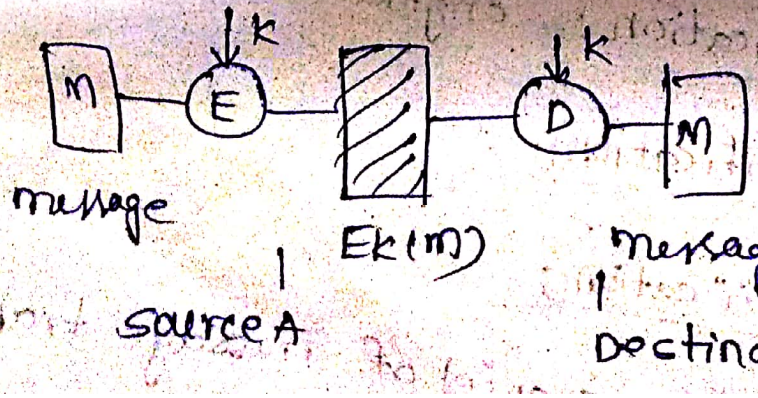
Authentication using encryption (~~confidentialit~~

→ Assumes sender and reciver knows key.

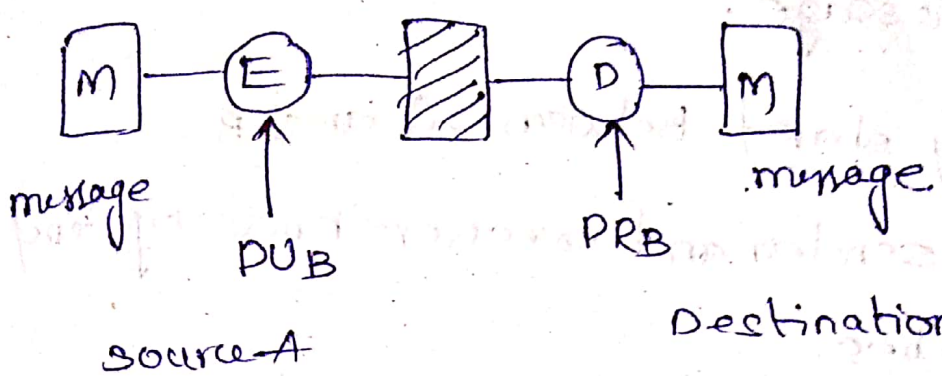
→ the Analysis is differ for symmetric encryption and public key encryption.

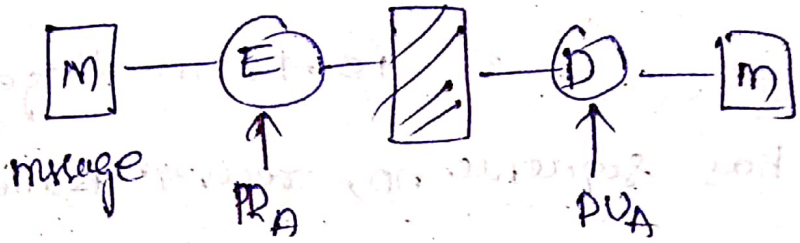
→ Encrypted message itself provides authentication

(1) Symmetric encryption:- (confidentiality)



(2) public key encryption (confidentiality)

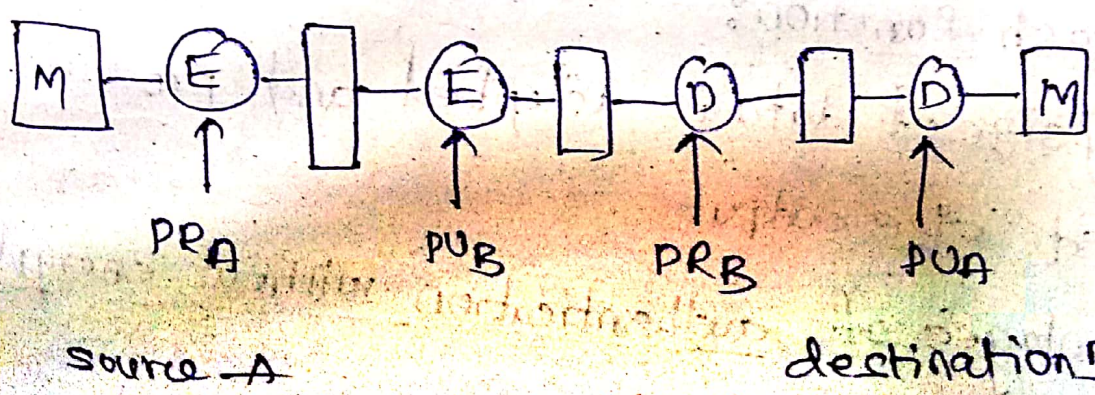




message includy

i) error detection code
ii) sequence number
iii) timestamp

→ Authentication along with confidentiality



(3) Message Authentication code (MAC)

→ Generate Authentication code based on shared key and message;

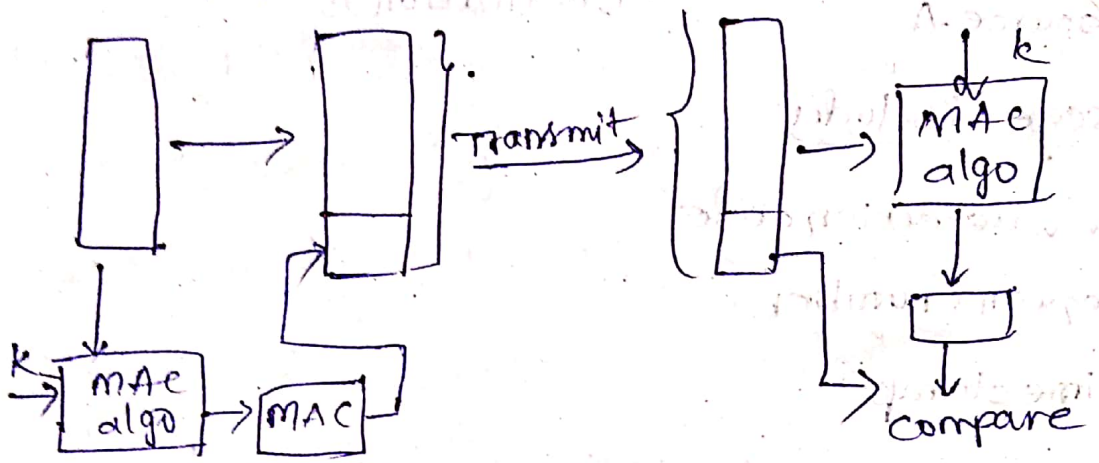
→ common key shared between A and B.

→ If only sender and receiver know key and code matches.

→ Receiver assured message has not altered.

→ " " " is from alleged.

→ If message has sequence no, receiver assured of proper sequence.



⇒ oneway Hash function:-

→ Any size of data is accepted and produces fixed size output.

Advantages of authentication without encryption:-

→ Encryption is slow

→ Encryption hardware expensive

→ Encryption hardware is optimized to large data.

* secure Hash Functions:-

properties :-

- can be applied to any size of data block.
- produs fixed length output
- Easy to compute
- Not feasible to reverse
- Not feasible to find two message that give the same hash.

* public key cryptographic systems:-

services :-

provides confidentiality, Authentication,

→ classified into 3 types based on

- Encryption, decryption
- Digital signature
- key exchange

Algorithm	Encryption/ Decryption	Digital signature	key exchange
RSA	✓	✓	✓
Eliptic curve	✓	✓	✓
Diffe-Hellman	X	X	✓
DSS	X	✓	X

RSA Algorithm :-

→ Generating public key and private key

- Select p, q — (prime no's)
- calculate $n = p \times q$
- Select integer $\gcd(\phi(n), e) = 1$; $1 < e < \phi(n)$
- calculate d $\quad d = e^{-1} \bmod \phi(n)$

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

public key $KU = \{e, n\}$

private key $KR = \{d, n\}$

Encryption:- Decryption

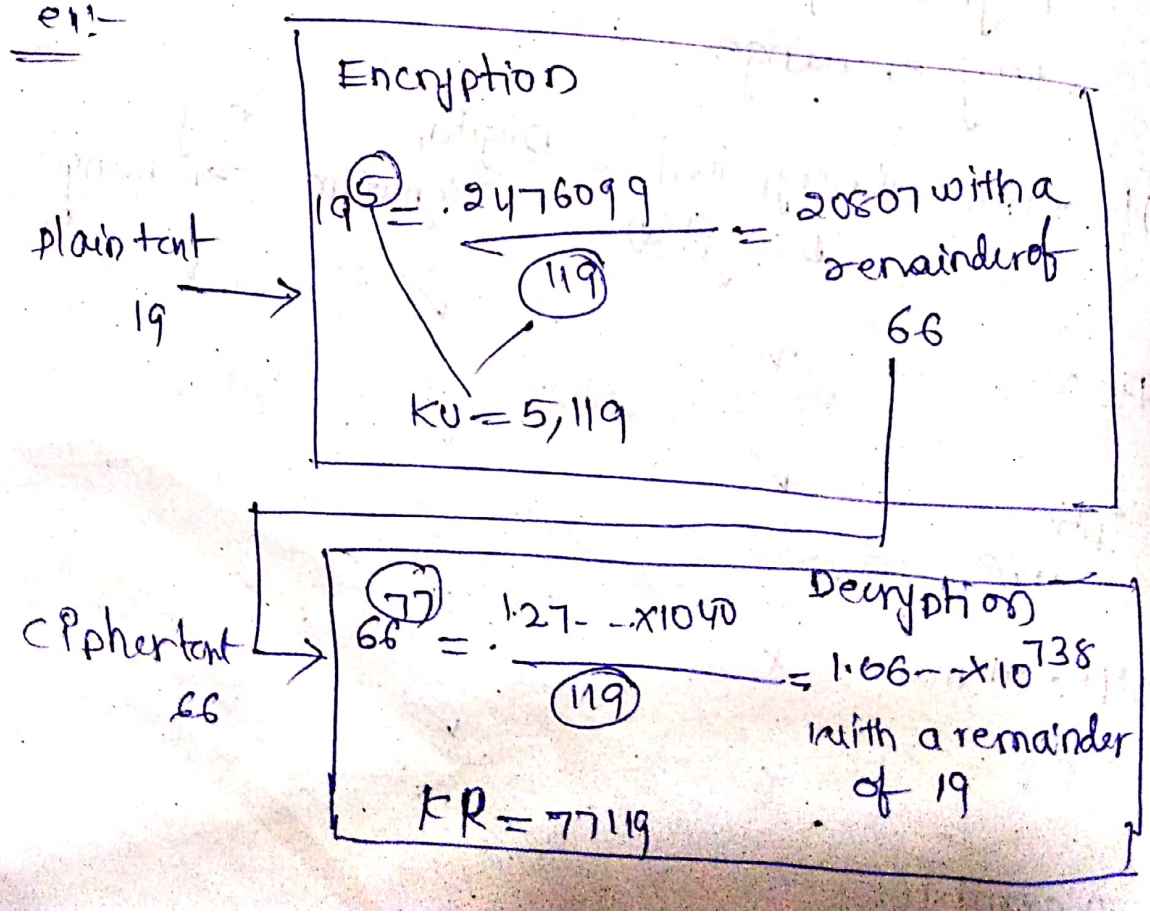
plaintext: $M < n$ (message)

plaintext: $M = C^d \pmod n$

ciphertext: $C = M^e \pmod n$

→ plain text and cipher text are bits. these values are from 0 to 1024.

ex:-



step1 :-

- select 2 prime numbers $p=17$, $q=11$
- calculate $n = p \times q = 17 \times 11 = 187$
- calculate $\phi(n) = (p-1)(q-1) = 16 \times 10 = 160$
- select "e" such that "e" is relatively prime to $\phi(n)$ i.e., 160 and less than $\phi(n)$.

we choose 'e' is 7.

$\gcd(\phi(n), e) \Rightarrow \gcd(160, 7) = 1$.

- Determine d such that $de = 1 \bmod(160)$ and.

Note:- d is less than 160

$d = 23$.

public key $KU = [7, 187]$

private key $KR = [23, 187]$

step2 :- Encryption

plain tent: 88 ($88 < 187$)

cipher tent $C = M^e \pmod n$

$88^7 \bmod 187 = 11$

step3 :- Decryption

plain tent: $M = C^d \pmod n$

$= 11^{23} \bmod 187$

$= 88$

Date:- 29/08/19

Diffe-Hellman key exchange algorithm:-

1) Global elements
   - q any prime no
   - α $\alpha < q$ and a primitive root of q

2) User A key Generation
   Select private x_A $x_A < q$
   calculate y_A $y_A = \alpha^{x_A} \bmod q$

3) User B key generation
   Select private x_B $x_B < q$
   calculate public y_B $y_B = \alpha^{x_B} \bmod q$

4) Generation of secret key by user A
   $k = (y_B)^{x_A} \bmod q$

5) Generation of secret key by user B
   $k = (y_A)^{x_B} \bmod q$

→ Common key for both sender and receiver.
k (secret key).

Ex:- $q = 7$ — any prime no.

$\alpha = ?$

$1^i \bmod 7 \quad i = 1, \dots, 6$

$1 \bmod 7$

$1^2 \bmod 7 = \{1\}$

$a^i \bmod n \quad i = 1, 2, \dots$

$a \bmod n$

$a^2 \bmod n$

$a^3 \bmod n$

$a^{n-1} \bmod n$

$\{1, 2, 3, \dots n-1\}$

a is primitive root of

$2^i \bmod 7$ $i = 1, 2, \ldots 6$

$2 \bmod 7 = 2$

$2^2 \bmod 7 = 4$

$2^3 \bmod 7 = 1$

$2^4 \bmod 7 = 2$

$2^5 \bmod 7 = 4$

$2^6 \bmod 7 = 1$

$\{1, 2, 4\}$

$3^i \bmod 7$ $i = 1, 2, \ldots 6$

$3 \bmod 7 = 3$

$3^2 \bmod 7 = 2$

$3^3 \bmod 7 = 6$

$3^4 \bmod 7 = 4$

$3^5 \bmod 7 = 5$

$3^6 \bmod 7 = 1$

$\{1, 2, 3, 4, 5, 6\}$

3 is the primitive root of 7.

$q = 7$

$\alpha = 3$

2) $x_A = 4 \rightarrow$ private

$y_A = 3^4 \bmod 7 \rightarrow$ public.

$= 4 \rightarrow$ public.

3) $x_B = 6$

$y_B = 3^6 \bmod 7$

$y_B = 1$

4) $k = (4)^4 \bmod 7$

$= 1$

5) $k = 4^6 \bmod 7$

$4096 \bmod 7$

$k = 1$

Elkptic curve cryptographic Algorithm.

- → Encryption
- → Decryption
- → message authentication
- → Hash function.

Key Exchange

Step1 :-

User A selects a random integer n_A which will become his/her private key. public key which is a point in $E_p(a,b)$ is calculated as follows.

$$P_A = n_A \times G$$

where G is a generator point in $E_p(a,b)$

Step2 :-

User B similarly selects a random integer n_B which will become his/her private key. public key is given by $P_B = n_B \times G$

Step3 :-

User A generates the secret key 'K' as follows

$$K = n_A \times P_B$$

Step4 :-

User B generates the secret key 'K' as follows

$$K = n_B \times P_A$$

Encryption & Decryption :-

→ To encrypt a message P_m, user A chooses a random positive integer k and produces the cipher text C_m as follows

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

→ To decrypt the cipher text the user B does the following calculations.

$$P_m = \phi * P_B - n_B(kQ)$$

$$P_m + k n_B G) \phi - n_B(kQ)$$

$$= P_m$$

Assignment

ADDITIVE RULES of Eleptic curve cryptograph algo

(2) SHA1, security aspects of RSE algo.

(3) Hacks on hash → diff Hus (words)

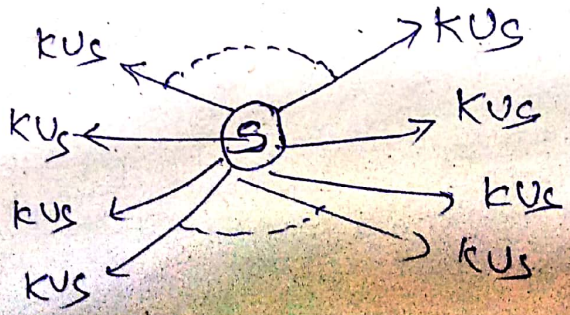
DSA (Digital signature Algorithm)

public key distribution :-

There are several approaches for distribution of public keys some of them are!

1) public announcement of public keys
2) public available directly.
3) public key Authority
4) public key certificates.

1) public announcement of public keys:-



2) public available directory

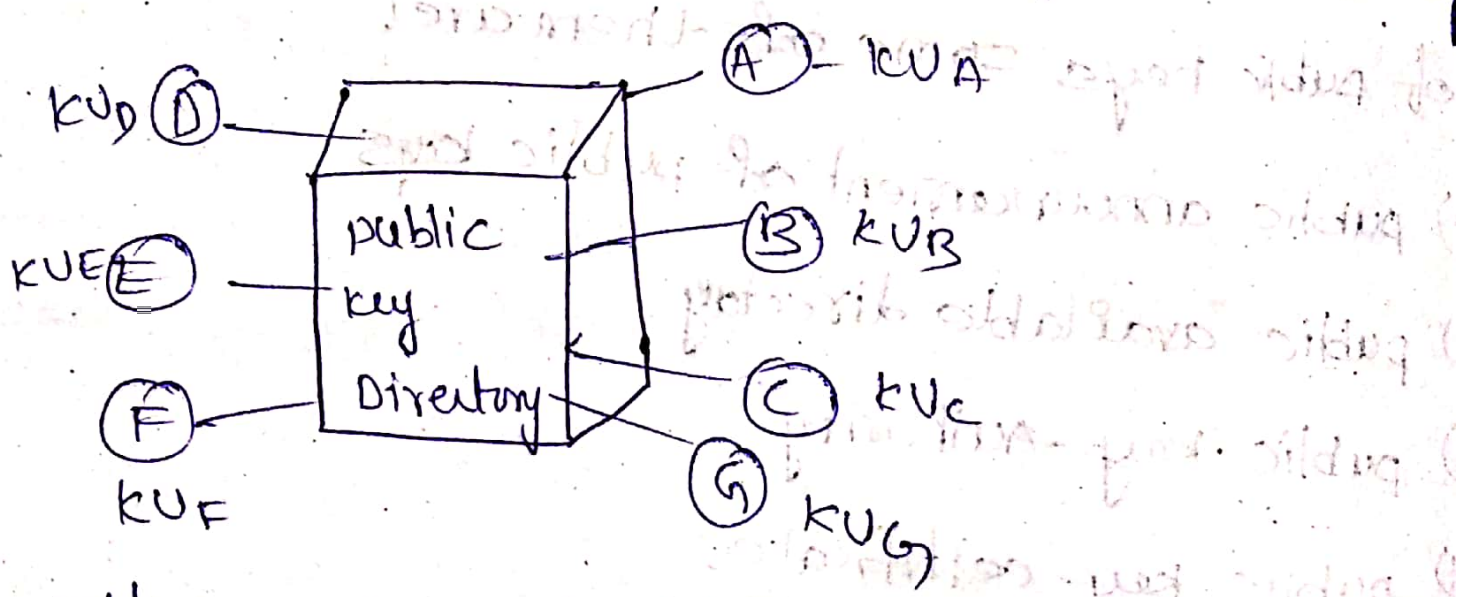
The directory must be trusted with the following properties

i) can contain names and public key entries

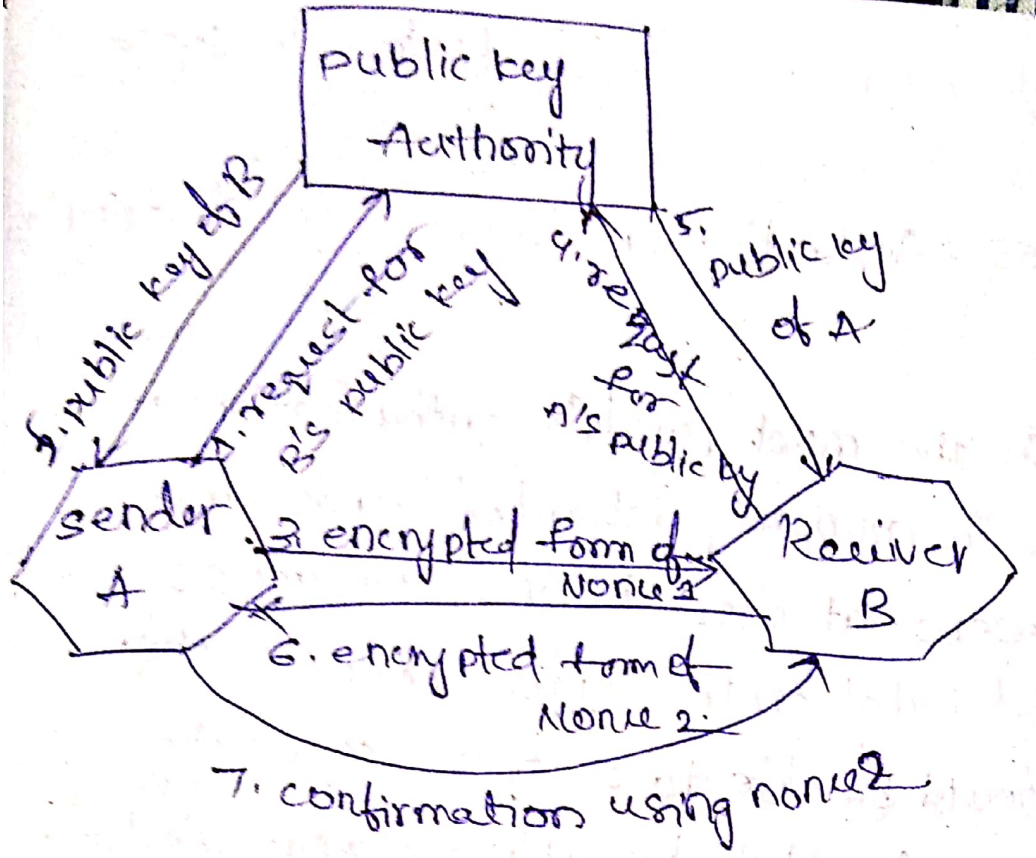
ii) participants register securely with the public directory.

iii) All participants has a chance to replace keys at any time. This directory is periodically published so that modifications can be seen by others.

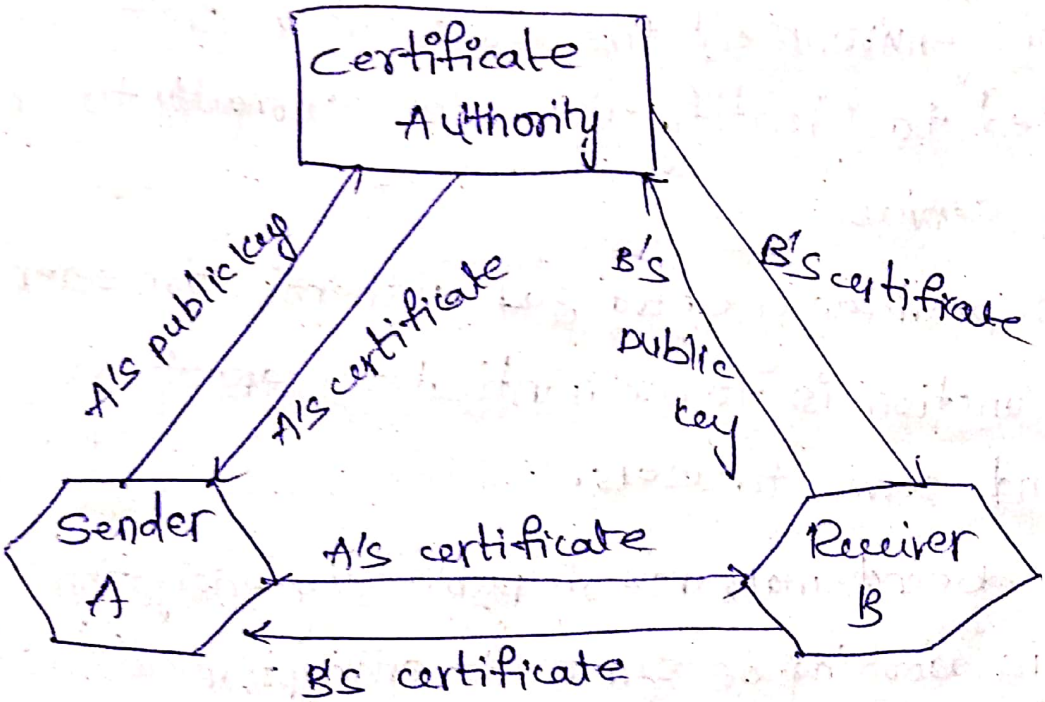
iv) public directory should be available to all users and can be accessed by electronically



3) public key Authority:



4) public key certificates



Date:- 09/8/19

Date:- 09/Sept/19

Kerberos:-

→ It provides Authentication services in a distributed network.

→ Kerberos is the most popular authentication service. This deals in an open distributed environment in which users at workstations can access servers distributed on the entire network the servers should be able to restrict access to authorized users and to be able to authenticate request for service.

→ So, in this environment the workstation can't be trusted to identify the users correctly to network services.

→ Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users.

→ Kerberos doesn't make use of public key encryption and it is working on symmetric encryption.

Requirements for Kerberos:-

1) Secure

2) reliable :- Backup maintenance

3) Transparency :- Except entering a password the user should not be aware of the authentication taking place.

4) Scalable :- There should be modular, distributed architecture i.e., the system should be capable of supporting large no. of clients & servers.

Kerberos version 4 :-

→ It uses DES algorithm.
→ It has 3 dialogs.
1) A simple Authentication dialogue
2) A more secure Authentication dialogue
3) Version 4 authentication dialogue.

Simple Authentication Dialog :-

→ In an unprotected network environment any client can interact with any server for a service. In order to avoid an unauthorized client to obtain unauthorized privileges on server machines.
→ To use an authentication server (AS) that knows the passwords of all users and stores in a centralized database.
→ AS share an unique secret key with each server. These keys have been distributed physically or in some secure method.
→ So, in simple authentication dialog a process will be occured

1) C → AS : IDc || Pc || IDv
2) AS → C : ticket
3) C → V : IDc || ticket

where c – client
AS – Authentication server
V – server.

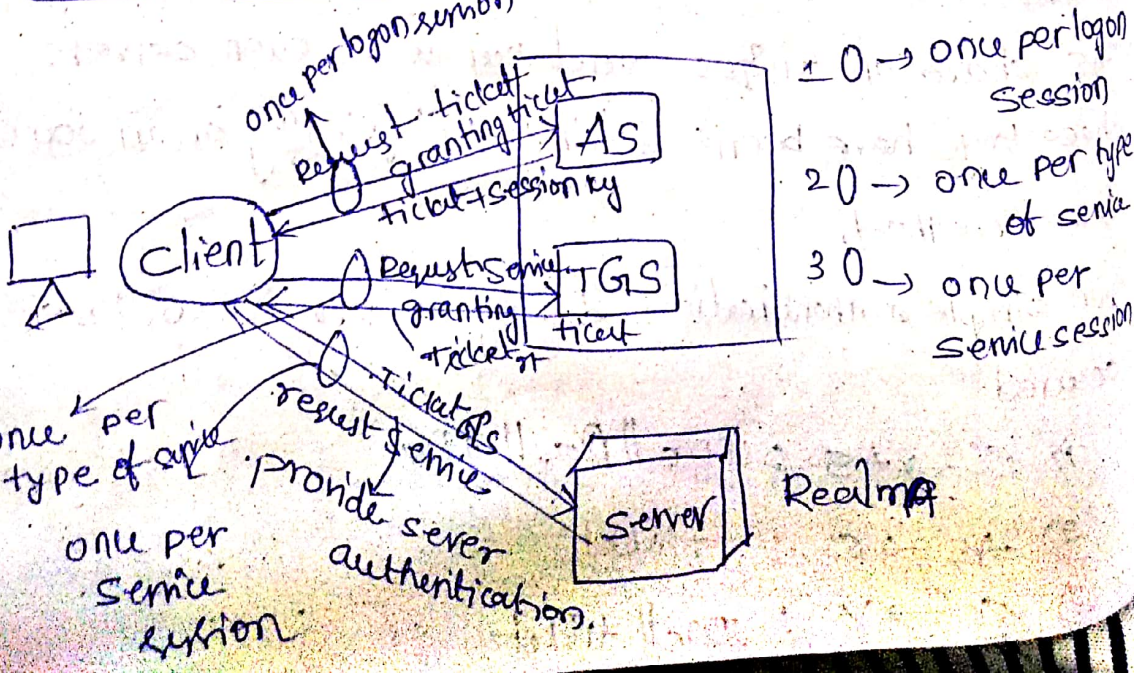
1) In the first case user will locks onto a work station and request to server 'V'. The client 'c' sends a message to the "AS" that includes users ID, servers ID and user's password.

2) The AS checks its database which is supplied by user then user is permitted to access server. For this the AS creates a ticket that contains user's id and network address and server's ID.
→ this is encrypted using secret key shared by the AS and this is send back to client.

3) Now, client can access request to server, now the ticket is valid only if at all it is transmitted from the same workstation which is initially requested for service.

Date:- 12-09-19

2) A more secure Authentication dialog:-



Request ticket granting ticket → AS
ticket + session key ←

Request servic granting ticket → TGS
+ Ticket_tgs ←

Ticket_gs – Request service
provide server authentication → Server ←

messages

a) Authentication servic exchange to obtain ticket – Granting ticket.

(i) C → AS : options || IDc || Realmc || IDtgs || Times || Nonce1.

(ii) AS → C : Realmc || IDc || Ticket_tgs || Ekc [Kc,tgs || Times || Nonce1 || Realm_cgs || IDtgs]

E_kc → Encrypt by using key of C.

b) Ticket granting servrice exchange to obtain service granting ticket.

(i) C → TGS : options || IDv || Times || Nonce2 || Ticket_tgs || Authenticatorc

(ii) TGS → C : Realmc || IDc || Ticket_v || Ekcv [Ekcv || Times || Nonce2 || Realmv || IDv]

Ticket_tgs = Ektgs [Flags || Kc,tgs || Realmc || IDc || ADc || Times]

Ticket_v = Ekv [Flags || Kcv || Realmc || IDc || ADc || Times]

Authenticator$_c$ = $E_{K_{c,tgs}}$[ID$_c$ || Realm$_c$ || TS$_1$]

c) client server Authentication exchange to obtain service.

(i) C → V : options || Ticket$_v$ || Authenticator$_c$

(ii) V → C : $E_{K_{c,v}}$[TS$_2$ || subkey || Seq #]

- Ticket$_v$ = E_{K_v}[flags || $K_{c,v}$ || Realm$_c$ || ID$_c$ || AD$_c$ || Times]

Authenticator$_c$ = $E_{K_{c,v}}$[ID$_c$ || Realm$_c$ || TS$_2$ || subkey || Seq #]

Date:- 18/09/19

Expand X.509 Directory Authentication Service:-

→ X.509 certificate format is used in S/MIME, IP security and SSL/TLS and SET.

→ X.509 uses public key cryptography and digital signatures. The standard doesn't decide the use to any specific algorithm but recommended to use DSEL algorithm and the digital signature scheme is assumed to require the use of the Hash function.

parameters

Z <<b>> : Z {v, s, val.}

↓ CA — Issuer ↓ Subject

Y <<y>>

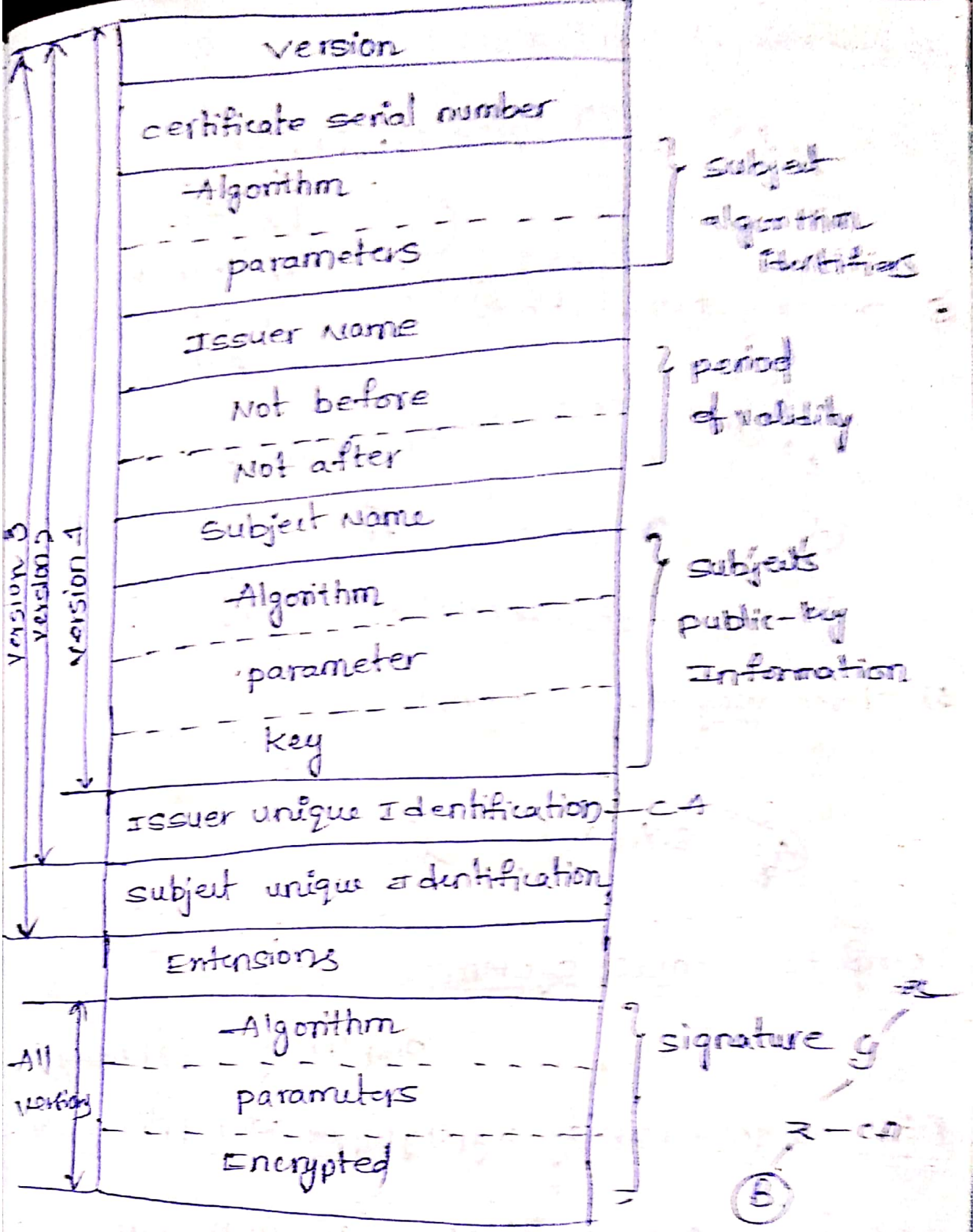
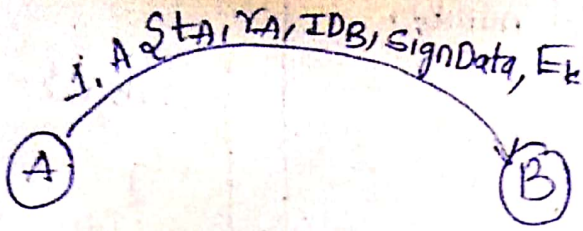


Fig:- X.509 certificate format

Authentication procedures in X.509

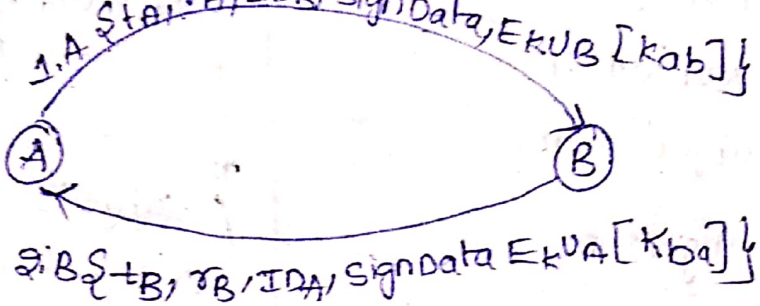
1) one way Authentication
2) Two way Authentication
3) three way Authentication

1] One way Authentication :-

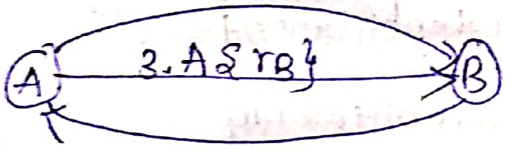


t_A → time period
r_A → random/nonce
ID_B → identification
U_B → public key of B

2] Two way Authentication :-



3] Three way Authentication :-



Complete E-mail system :-

PGP (pretty good privacy)

PGP :- high security cryptography software application

PGP was developed by Zimmerman in 1980 and the 1st version was released on internet in 1991. Because of legal issues for usage of RSA. It was purchased by via crypt and RSA licensed company in 1993 and released again in 1994.

→ In 1998 it was purchased by network associates

→ Zimmer was developed PGP and he has done the following things.

- Best available cryptographic algorithms used as building blocks.
- Integrate these algorithms into a general purpose application.
- It is independent of os & processor.
- simple commands to use.
- made total documentation including source code freely available via the internet.

Functions/operations

1) Authentication
2) confidentiality
3) compression — speed up process
4) compatibility
5) segmentation.

Date:- 17/09/19

PGP operations:-

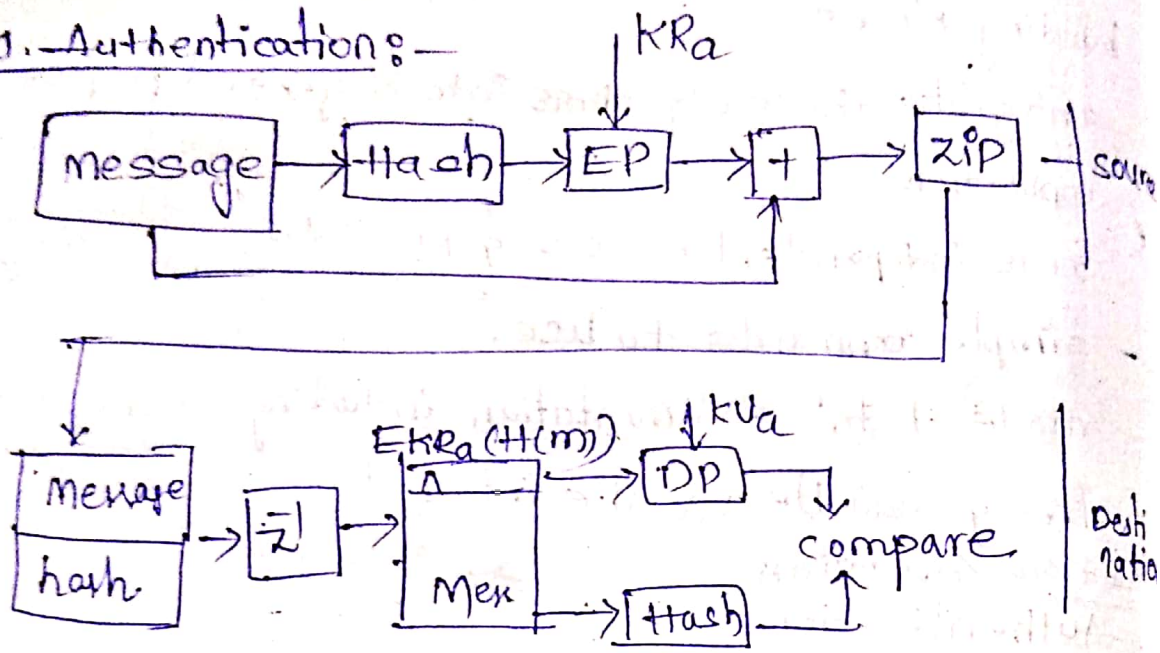
Service	function	Algorithms used.
1. Authentication	Digital signature	DSS with SHA or RSA/SHA
2. confidentiality	message encryption	CAST/IDEA/triple DES along with DH/RSA

3. Speed compression ZIP Algorithm.

4. transmission size email compatability RADIX 64 conversion

5. Limitations segmentation —

1. Authentication:—



1) — 160 bit hash code is generated by using SHA-1 algo.

2) - EkRa(H(m))

3) EkRa(H(m))||M

4) z(EkRa(H(m))||M).

Destination

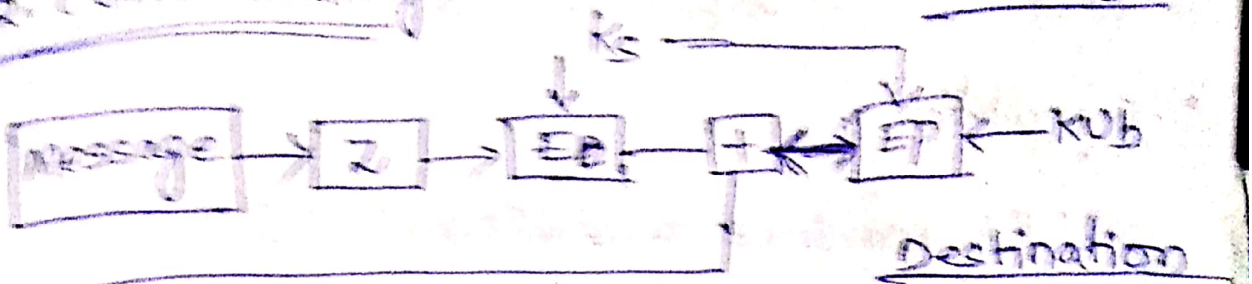
1) unzip — EkRa(H(m)), M

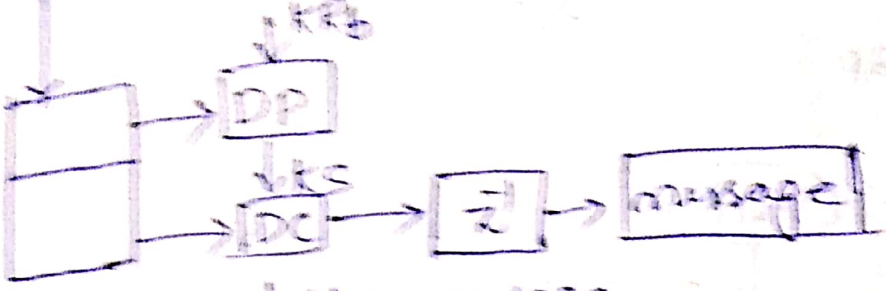
2) Decrypt — DKUa(EkRa(H(m)))

3) M → hash H(m)

4) comparesion.

2. Confidentiality :-





EC - Encrypt the message

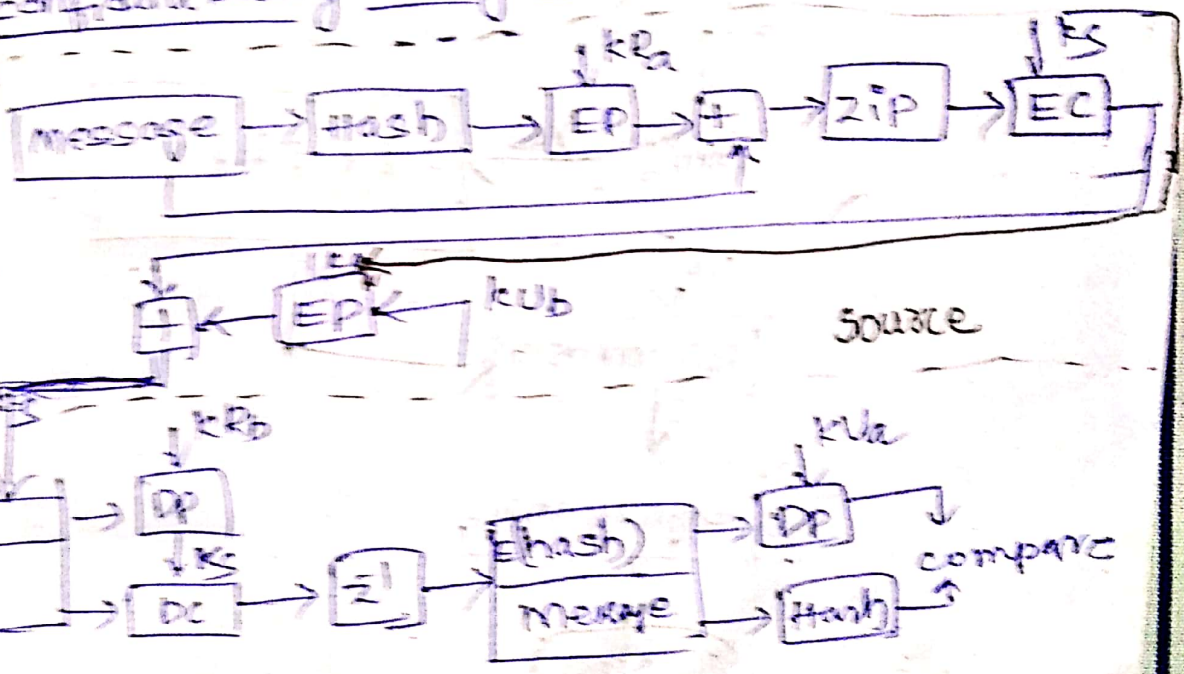
EP - encrypt the session key.

DP - Decrypt the " "

DC - " " message.

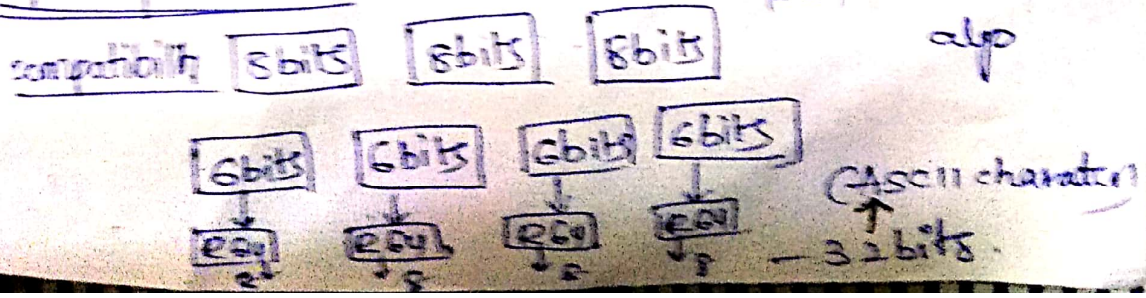
Ks length - 128 bit

confidentiality along with authentication.



3. ~~compression~~ :-

RLE - Radix64 alp



3. compression :- zipping the data.

5. Segmentation :-

MTU - maximum ~~transmission~~ unit
transfer

Flowchart PGP