4) Playfair Cipher Algorithm

Cipher - Algo for encrypting & decrypting
Ciphertext - Process which applies different types of algos to convert ~~cip~~ plain text to coded text

Algorithm

1) Create 5×5 matrix called grid of letters
2) The matrix is made by inserting the values of key and remaining alphabets into the matrix (row wise from left to right) where letter I & J are combined together
3) Convert the text into pairs of alphabets
   eg → Heya → He ya
   a) Pair cannot be of same letters, add 'x' in between.
      eg → Hello → He lx lo
      Helloe → He lx lo ce ← alone
   b) For alone letters, add 'z'
      eg → Helloe → He lx lo ez
      Hexxoe → He xz xo ez ← for alone e
      ↑ x already paired, so add z

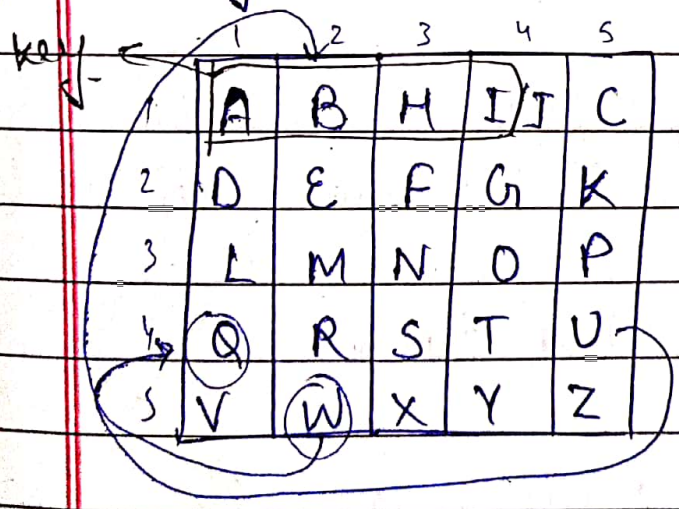
4) Code will be formed using 3 rules

i) If both the alphabets are in the same row, replace them with alphabets to their immediate right.

ii) If both the alphabets are in same column, replace them with alphabets to their immediate below

iii) If not in same row/column, replace them with alphabets in the same row respectively, but at other pair of corners.

eg → key = AABHI.



Convert

BM → ER

RW → WB

FG → GK

UQ → QR

QW → RV

FL → DN

Fiestel Cipher Structure

Most of the block cipher technique follows this structure.

i) The plain text is processed (divided) into two equal halves L_0 and R_0

→ The two halves of the data pass through N rounds of processing and then combine to produce the cipher text block of equal length.

→ The right half we apply a function & in the function we will need a subkey generated from the master key. The o/p of this will be XOR with the left half and then their o/p will be swapped.
(for one single round)

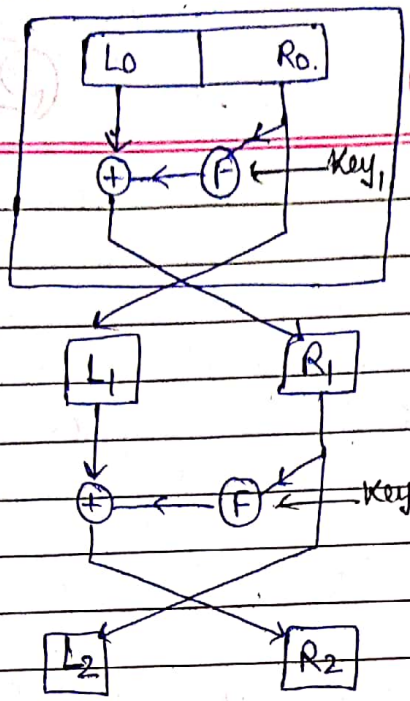
Main key

↓

In n rounds

key1 key2 . . . key(n) → subkeys generated by some algo in each round.

Plain text divided into 2 halves



Some terms,

1) Block size - larger block size, more security
2) Key size - larger key size, more security but speed of encryption/decryption decreases
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 - more complex fun^c, harder for cryptanalyst to attack.

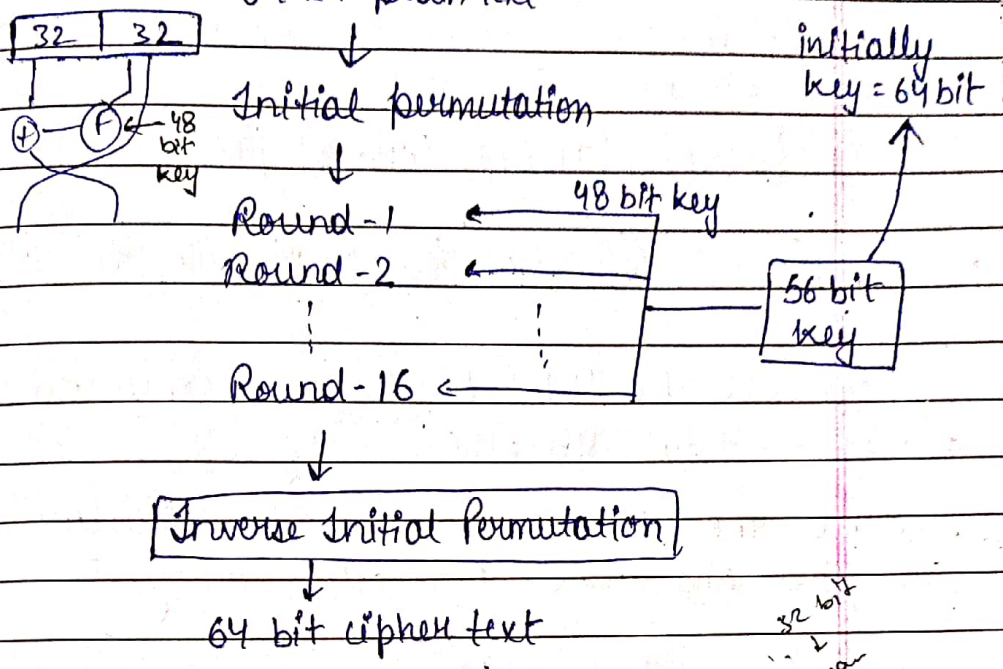
DES (Data Encryption Standard)

i) block cipher
ii) symmetric cipher
iii) 64 bit plain text block
iv) 16 rounds, each round is a Feistel round

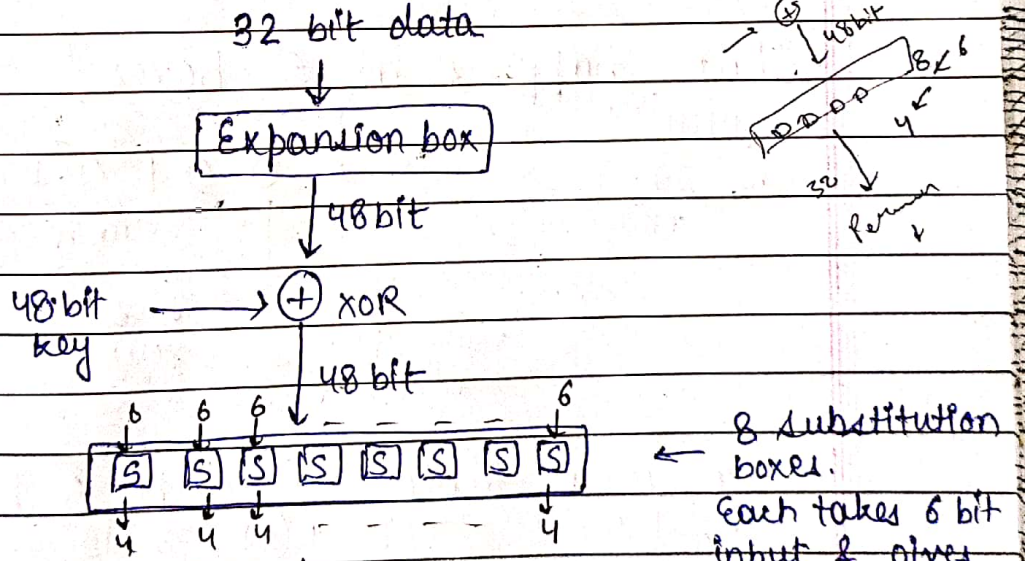
Steps

1) Initial permutation
2) 16 Feistel rounds
3) swapping / left right swap
4) Final permutation / Inverse Initial Permutation

Basic structure

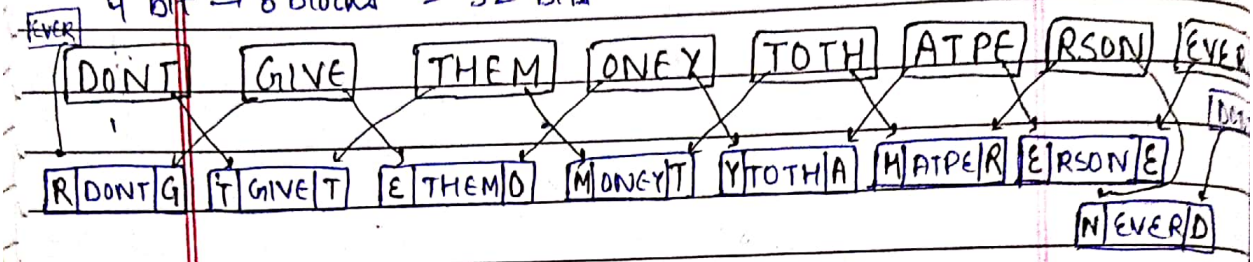


Function definition



What happens in expansion box?
32 bit data will be 1's & 0's but for explanation, let us consider a text-

4 bit → 8 blocks = 32 bits

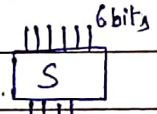


So, now all the 4 bit blocks are converted to 6 bit blocks → total 48 bits.

* Assume the 32 bit data to be circular so we take the first and last bit accordingly.

So, input = 32 bit
o/p = 48 bits

What happens in S-boxes?



eg → 0 0 1 0 1 1

column

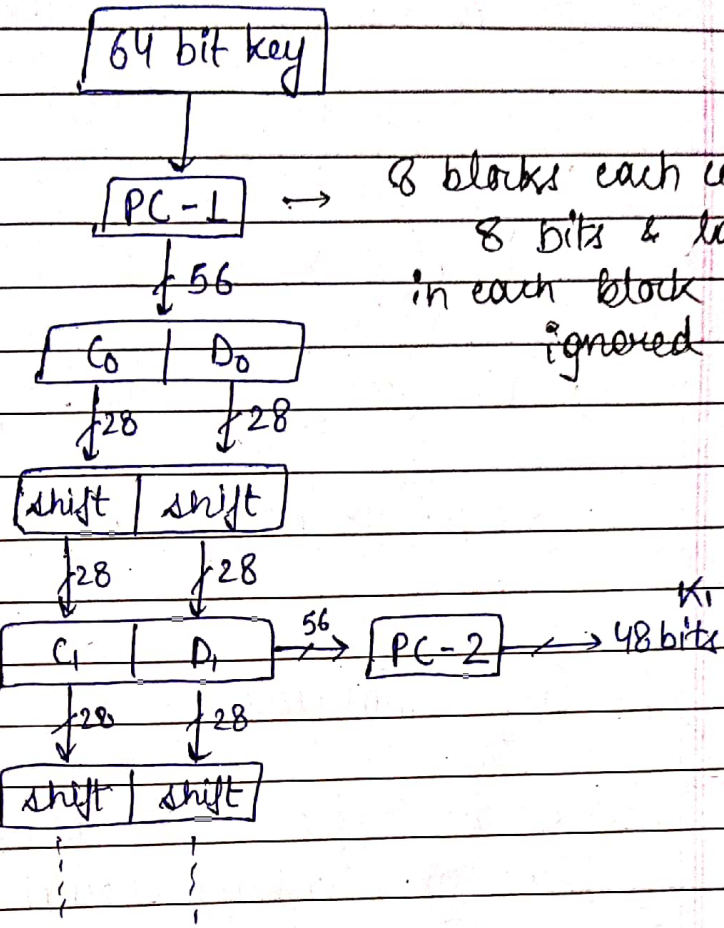
row = 01 → 1 row
column = 0101 → 5

There is a table for each S box

	0	1	2	3	4	5	...	15
0								
→ 1						✱		
2								
3								

The intersection element converted to its binary equivalent gives the 4 bit o/p.

How are sub keys generated?



1) In PC-1 (~~Permutation~~ Permuted Choice - 1)

In each part, last bit → discarded
8 parts × 7 bits = 56 bits

2) Divide into two halves (28 bit each)

3) Now, each part bits are shift by left in each round.
in rounds → 1, 2, 9, 16 → 1 bit left shift
in other rounds → 3, 4, 5, 6, 7, 8, 10, ... 15 → 2 bits left shift

4) In PC-2, 56 bits → 48 bits using a predefined table

C_1 → 28 bits (1-28)

D_1 → 28 bits (29-56)

not necessarily the same position bits

Now, in C_1 (9, 8, 22, 25) → position bits discarded → 24 bits

in D_1 (35, 38, 43, 54) → position bits discarded → 24 bits

↳ o/p = 48 bits (Key 1).

DES Analysis

(1) Avalanche Effect - It means a small change in plain text (or key) should create a significant change in the cipher text.

→ DES is proved to be strong with ~~strong~~ regard to this property.

eg→ Plain → 0000000000000000
Cipher → 4789FD476E82A5F1

Plain → 0000000000000001
Cipher → 0A4ED5C15A63FEA3

} key used is same, say Key = 22234512987ABB23

☆ changing 1 bit changes the cipher text significantly

2) Completeness Effect - It means that each bit of the cipher text needs to depend on many bits on the plain text.

→ Confusion & Diffusion produced by P-boxes & S-boxes in DES, show a very strong completeness effect.

Multiple DES

Double DES Triple DES.

→ Since DES attack was vulnerable to Brute force attack, we use multiple DES

I Double DES (2DES)

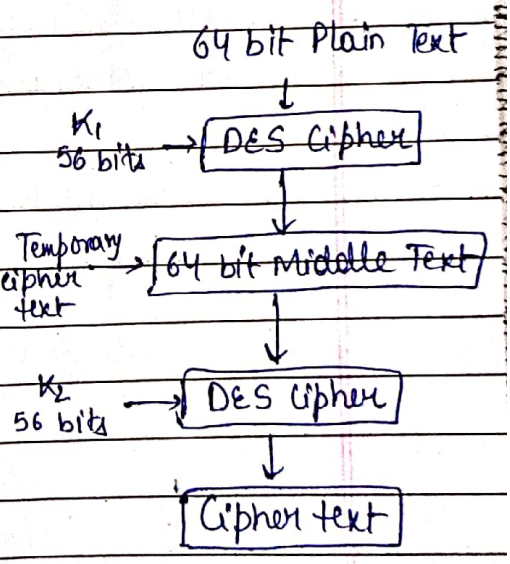
→ uses 2 diff keys
56 + 56 = 112 bit

→ Double Encryption,

$P \rightarrow E(K_1, P) = X$

$E(K_2, X)$

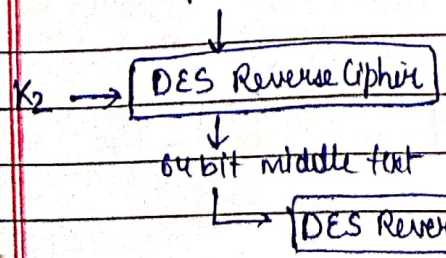
Cipher.



Cipher = $E(K_2, E(K_1, P))$

Plain = $D(K_1, D(K_2, C))$

→ For decryption,



Drawback of Double DES.

Meet-in-the-middle attack

This attack involves encryption from 1 end & decryption from the other end and then "matching the results in the middle"

(Sort them) Decrypt (K_2, C) = Encrypt (K_1, P)

(2^{56} combinations)

We will get some common middle cipher texts, $\Rightarrow$ (K_1, K_2) will be the key pair used.

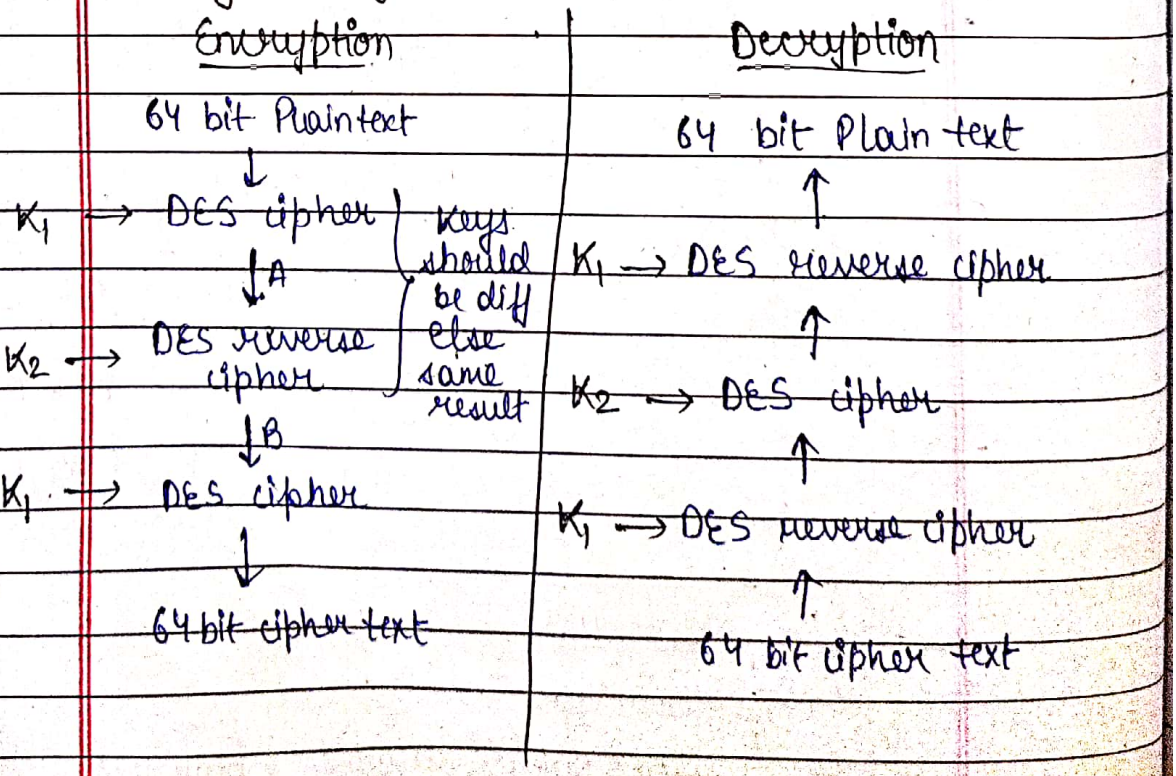
* It takes twice as long to break double DES using brute force than DES.

II Triple DES (3DES).

→ 2 or 3 keys used

→ much stronger than double DES.

a) using 2 keys.



b) using 3 keys.
same process, only keys used = $K_1 / K_2 / K_3$

DES Weaknesses

1) Key size: 56 bit key is easy to crack by today's technology.

So, we use ~~double~~ triple DES = 2 key → 112 bits
3 key → 168 bits

2) Weak keys: 4 out of 2^{56} keys are weak keys which are all 0's and all 1's, then half 0's half 1's

Disadvantage - If we encrypt a block with a weak key & subsequently encrypt the result with the same weak key, we get original block
→ (same for decryption)

So, if after 2 decryptions, the result is same - the attacker is successful.

3) Semi weak keys: 6 out of 2^{56} are semi weak keys.
A semi weak key creates only two different round keys and thus each of them is repeated 8 times.

4) Possible Weak keys: 48 keys out of 2^{56}
It produces only 4 different keys which are repeated in 16 rounds.

5) Key clustering
Means 2 or more diff. keys can create the same cipher text from plain text.

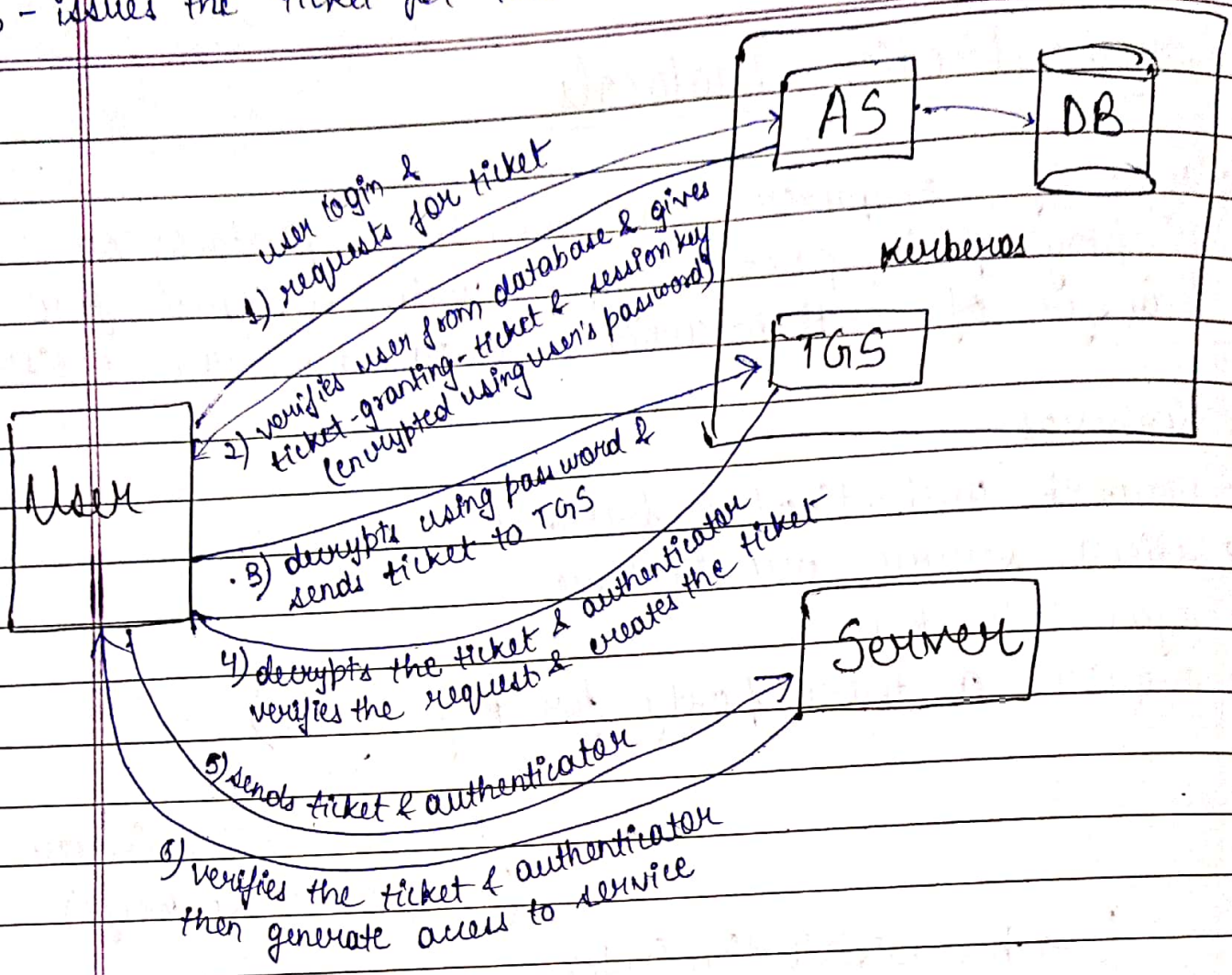
Weakness in cipher design.
1) Two specifically chosen i/ps to S-box array can create the same o/p.

Authentication Protocols

→ Type of computer communications protocol or cryptographic protocol specifically designed for transfer of authentication data b/w two entities.

1) Kerberos
→ network authentication protocol
→ client server architecture
→ symmetric key
→ requires a third party for key (KDC)
↓
Key Distribution Center
(db for secret keys)

AS – performs initial authentication & ticket for TGS
Database – AS verifies the access rights of users in the db
TGS – issues the ticket for the server



Advantages

1) supports various operating systems
2) authentication key is shared much efficiently than public sharing

Disadvantages

1) used only to authenticate clients & services used by them
2) shows vulnerability to soft or weak passwords.

X.509 Authentication Service

→ X.509 digital certificate is a certificate based authentication security framework that can be used for providing secure transaction processing and private information

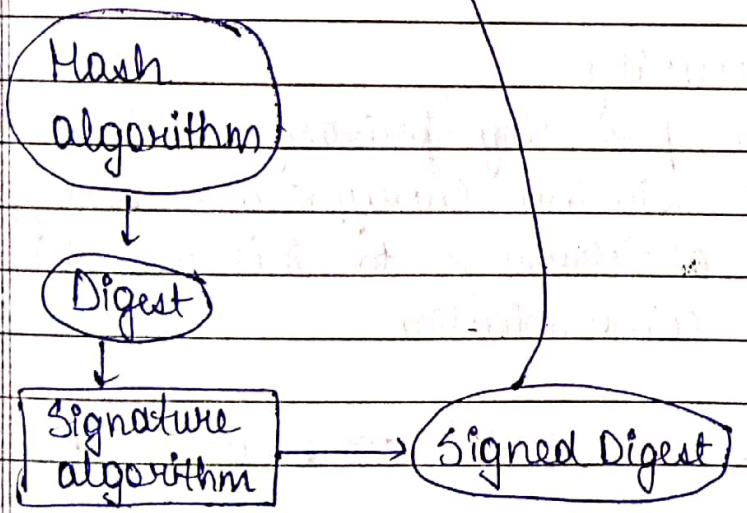
→ handle security & identity in computer networking & internet based communications

→ provides a way to access public keys but does not generate keys.
It provides a certificate stating that he is a authorized user & can get access to key

→ The certificate is basically presented like an 'identity' at the resource that requires authentication

Version Number
Serial Number (unique no.)
Signature Algorithm Identifier → algorithm used for signing the certificate
Issuer name → name of the certified authority issued
Validity period
Subject name → name of the user to whom certificate is
Subject Public Key → defines subject's public key along with an identifier of the algorithm for which key is to be used
Issuer Unique Identifier
Subject Unique Identifier
Extensions → contains additional standard information
Signature → contains the hash code

(Issuer Unique Identifier, Subject Unique Identifier, Extensions — optional)



Applications

1) Document signing and Digital Signature
2) Email certificates
3) Code signing
4) Digital identities

Directory Services

→ IT infrastructure for storing and mapping of resource information that are accessed frequently on daily basis.

Features → 1) centralized storage
2) Access speed relatively faster

Examples → 1) Employee profile details
2) Domain Name ~~System~~ (DNS) server

Pretty good privacy (PGP) in Cryptography

→ provides email security
→ encryption program providing privacy + authentication
→ PGP is used for signing, encrypting & decrypting texts, emails, files, directories & to increase the security of email communication

PGP encryption uses a serial combination of
i) hashing
ii) data compression
iii) symmetric key cryptography
iv) asymmetric key cryptography

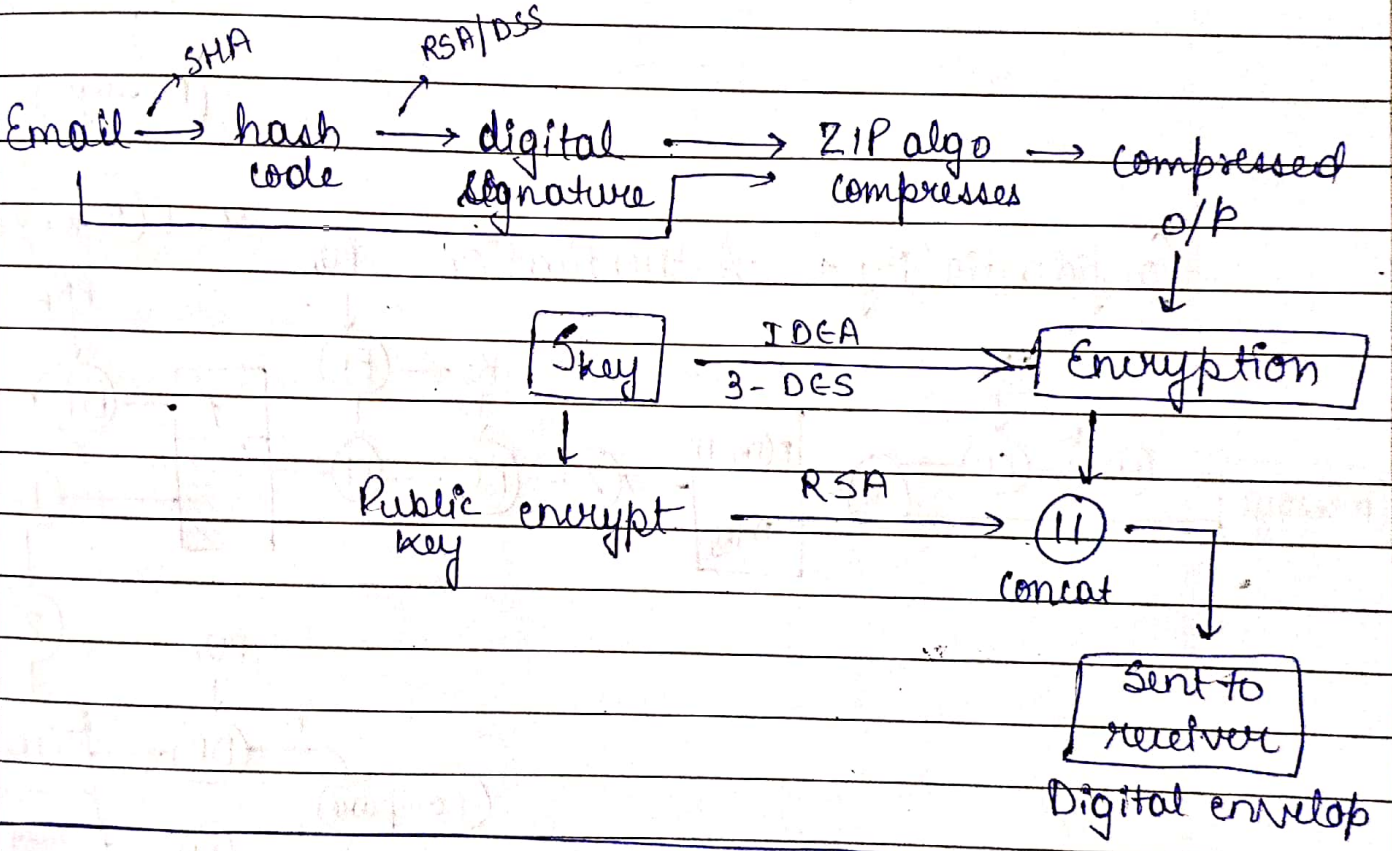
& each step uses one of the several supported algorithms like RSA, IDEA, SHA etc

Services provided:

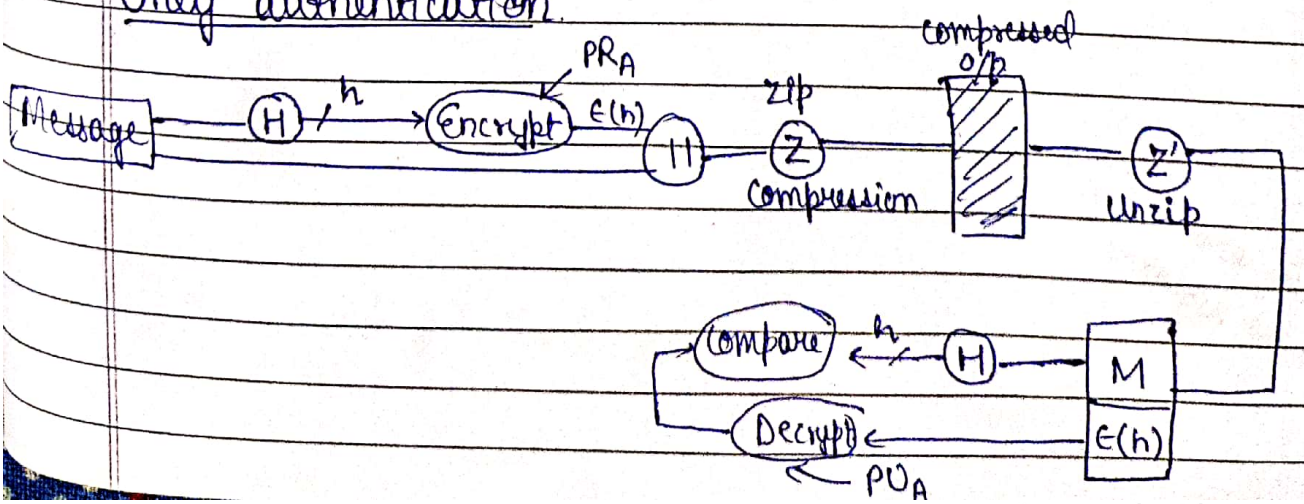
* authentication (using digital signature)
* confidentiality

We do compression using the ZIP algorithm

PGP provides email compatibility using the radix-64 encoding scheme.

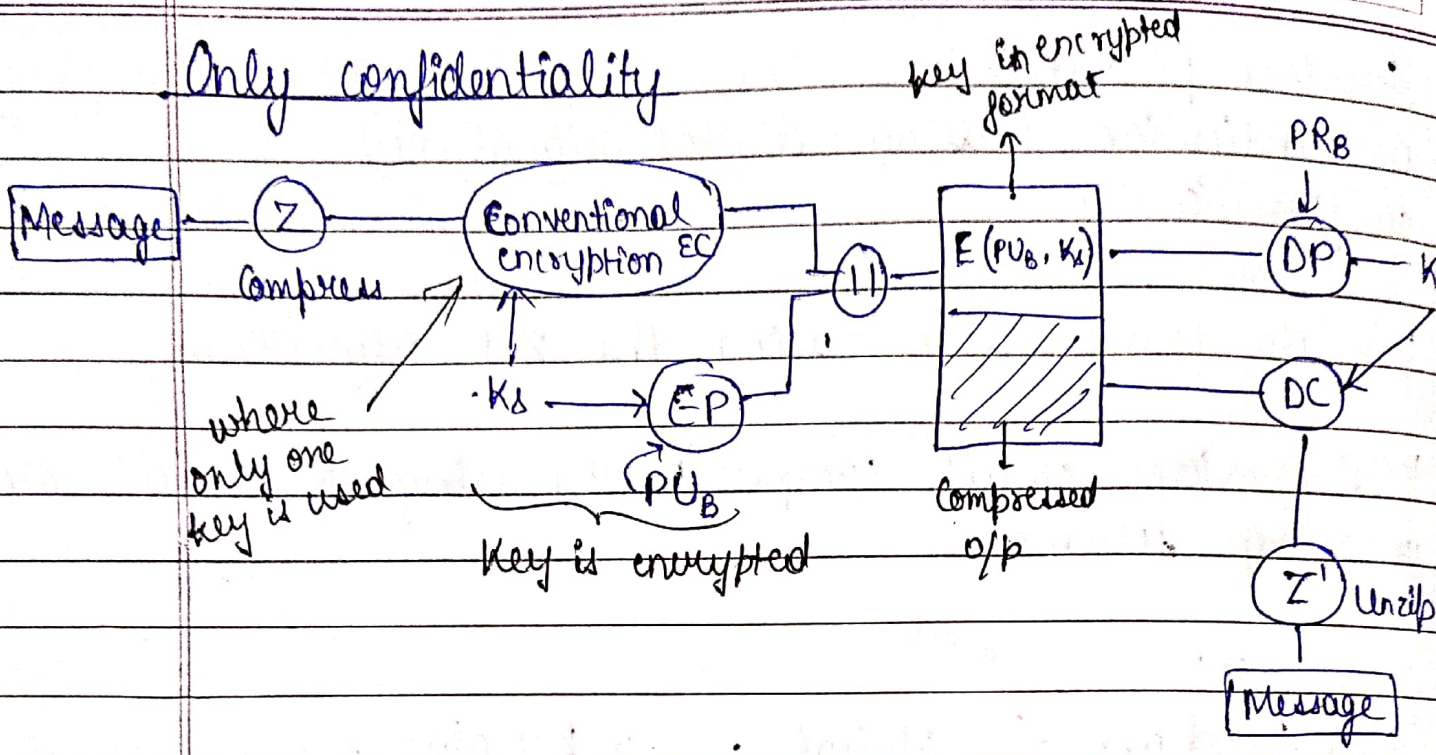


Only authentication



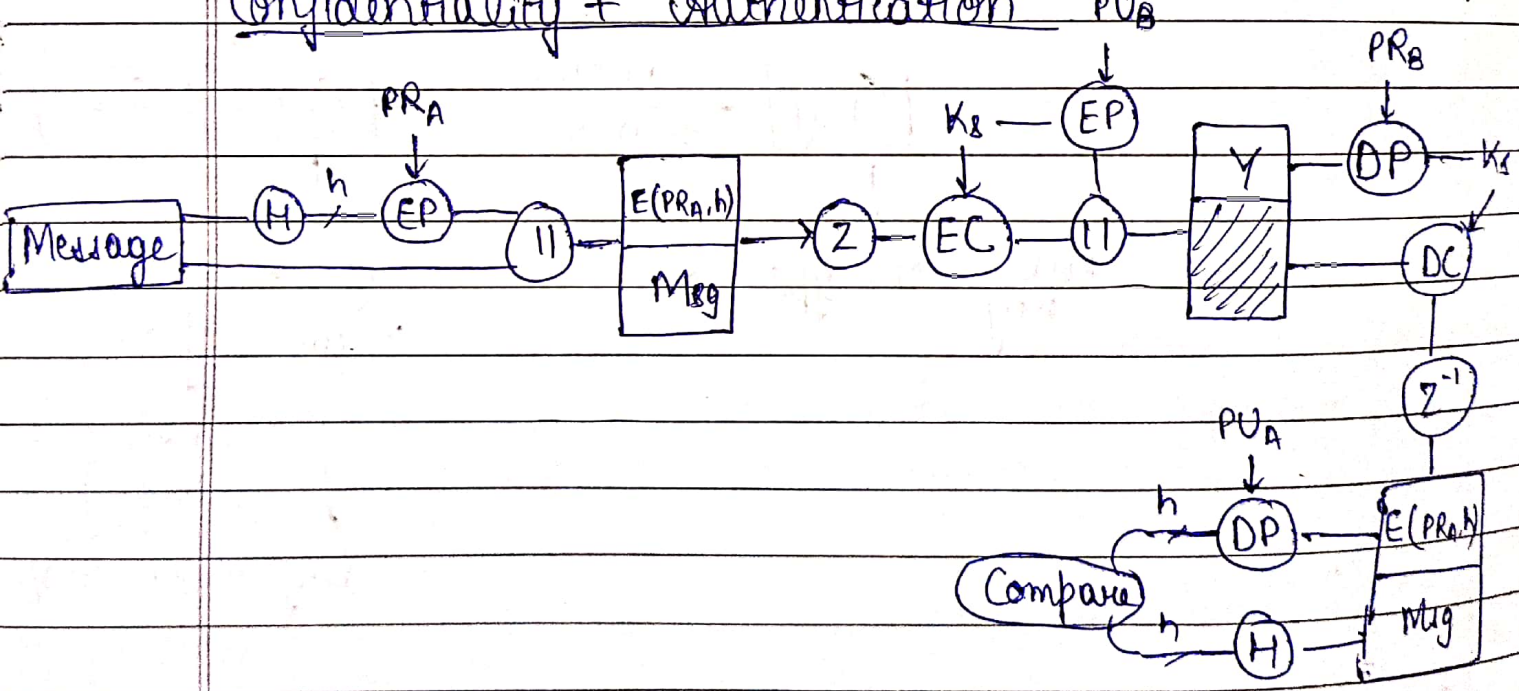
EP → encryption using public key

Only confidentiality



Confidentiality + Authentication

$Y = E(PU_B, K_s)$



Date
Page

S/MIME in Cryptography

→ Secure / Multipurpose Internet Mail Extension
→ provides security for commercial emails
↳ by encrypting emails
→ extension of MIME protocol
→ It is a widely accepted method for sending digitally signed & encrypted msgs
→ based on asymmetric key encryption

Func
i) authentication
ii) Message integrity
iii) Non-repudiation (using digital signature)
iv) Privacy
v) Data Security