The OSI Model →

- Application Layer
- Presentation Layer
- Section Layer
- Transport Layer
- Network Layer
- Data-Link Layer
- Physical Layer

The OSI Security Architecture →

focuses on Security attacks, mechanisms and services.

- **Security Attacks →**

Any action that compromises the security of information owned by organization.

- **Security Mechanism →**

A process that is designed to detect/prevent/recover from attack.

- **Security Services →**

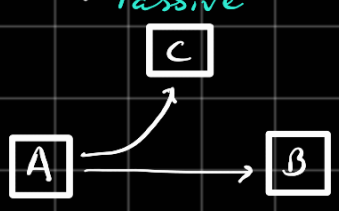
A processing service that enhances the security of the data processing systems and information transfers of an organization.

- **Attacks Types →**

→ Active

- Data modification happens.

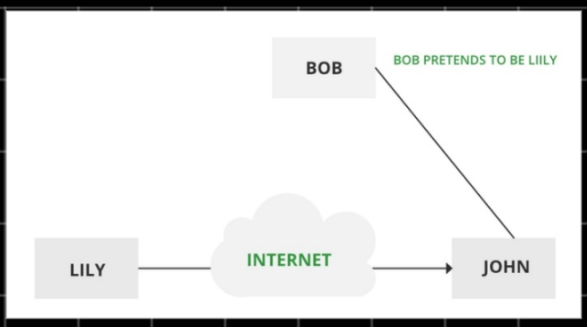
• Passive



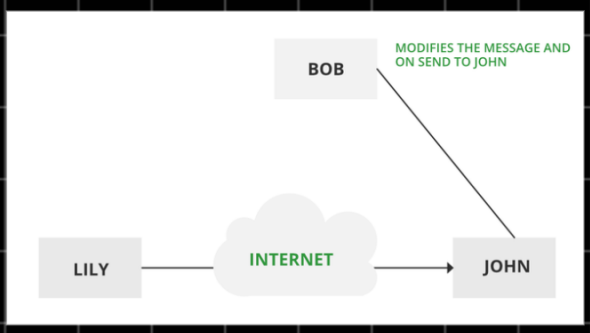
- One data is read. it cant be manipulated.

→ **Masquerade →**

One entity pretends to be a different entity.



→ Modification of message →

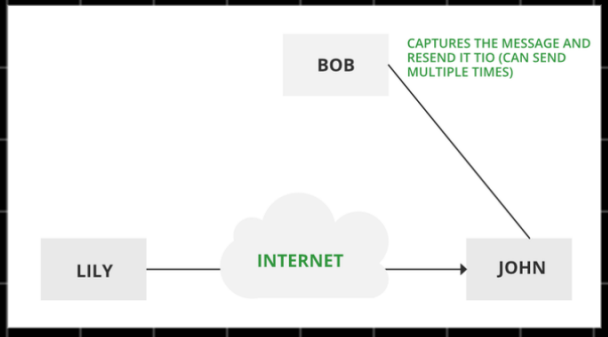


→ Repudiation →

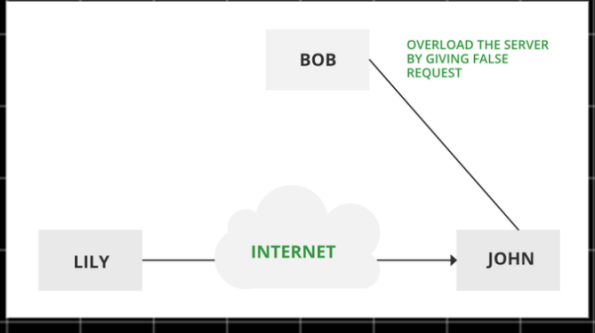
Tampering with the login control, manipulation of data on behalf of others.

→ Replay →

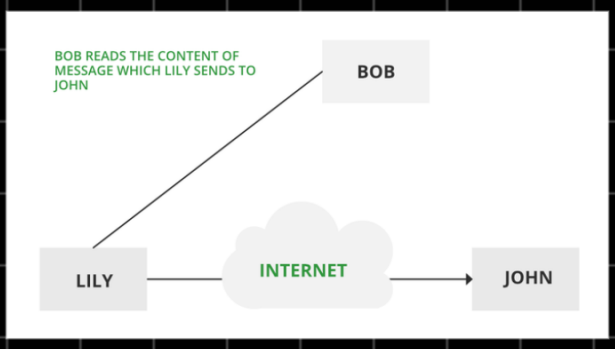
Recording/saving data present on particular network and use for personal use.



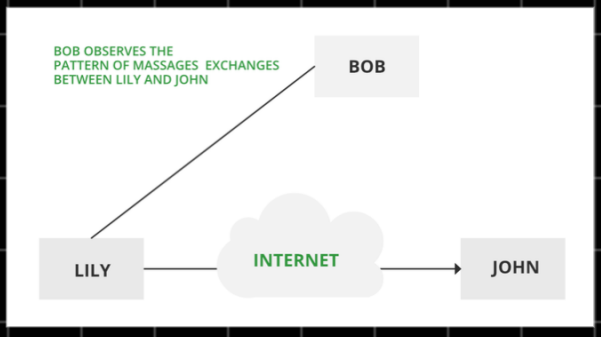
→ Denial of service →



→ Release of message content →

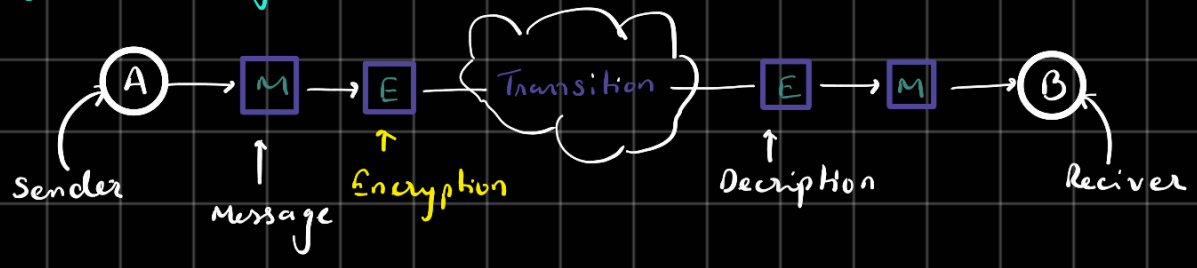


→ Traffic Analysis →

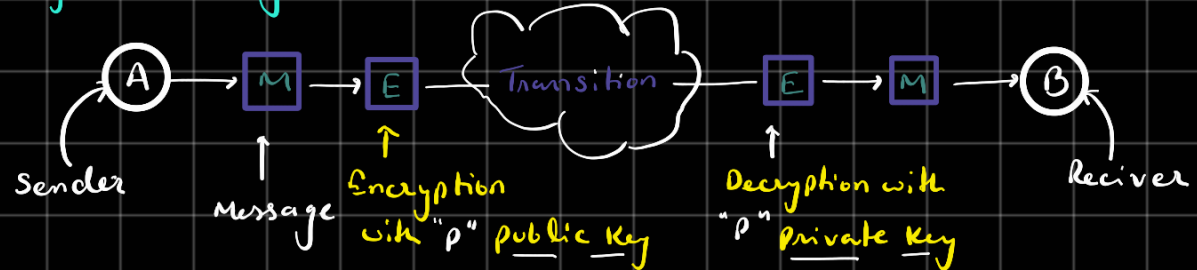


Cyptography Techniques →

- Symetric Key →



- Asymetric key →



Classical Encryption Approach →

1) Substitution Technique

Caesas Ciples

2) Transposition Technique

Rail fence

Monoalphabatic Ciphes

Playfair's Ciphes

Hill Ciphes

Polyalphabetic Ciphes

One-Time Pad / Vernam Ciphes

Row column Transposition

1. Caesar Ciphes →

$C = E(K, P) = (P+K) \bmod 26$

E → encryption K → Key

P → Plain Text C → Ciphes

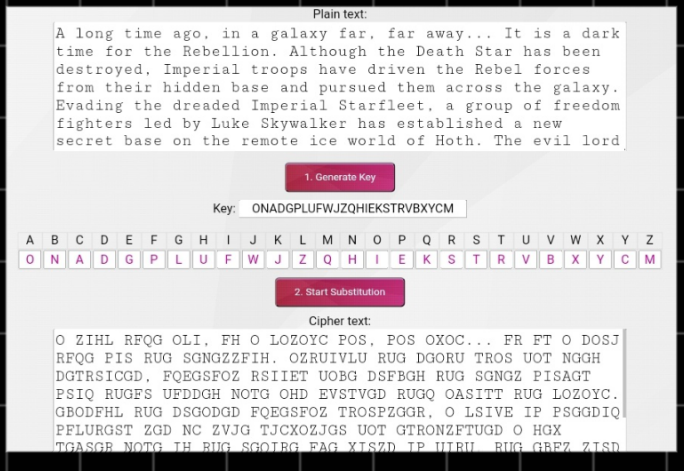
$P = D(K, C) = (C-K) \bmod 26$

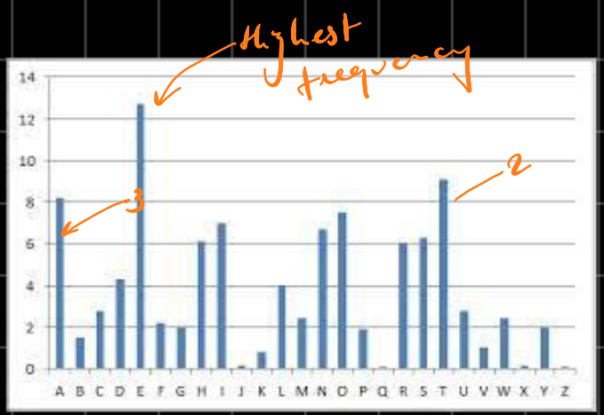
D → decryption K → Key

C → Ciphes text P → Plain text

2. Monoalphabatic Ciphes →

A unique key is generated for the alphabets.





Can be decoded by city letter repetation frequency.

3. Playfair's Cipher →

- If both the alphabets are in the same row, replace with alphabets to immediate right.

- Alphabets in same column, replace with alphabet immediatly below them.

- If not in same row/column, replace with alphabet at (x, y) coordinate format.

- If same alphabet come introduce 'x' in between them.

- If alphabet is single, add bonus alphabet 'z'.

Key → Yash

Y	A	S	H ← B	
C	D	E	F → G	
I/J	K	L	M → N	
O	P	Q ←	R	T
U	V	W →	X	Z

BM → HN

RW → QX

FG → GC

CJ → JO

because of repeatation

- Happy → Ha px py (bonus)

Ciphers → BS RV OA

because it was in pairs.

- The → Th ez (bonus)

Ciphers → RB GW

4. Hill Ciphers →

- Form matrix with Key.
- Divide the plaintext into (n×1) matrix form.

$$C = [Key]_{(n,n)} [Plaintext]_{(n,1)} \bmod 26$$

$$Plaintext = [Key]^{-1}_{n \times n} [C] \bmod 26$$

"Hill" like

"Nice"

"MYUT"

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$(a \times b - c \times b)\,|A|$

A^{-1}

5. One Time Pad / Vernam Ciphers

- Length of plaintext = length of Key.

eg Plain Text → RAMSWARUPK

key → RANCHOBABA

Soln → Plain text →

key →	17	0	13	2	7	14	1	0	1	0
(T+key) after adding →	34	0	25	20	29	14	18	20	16	10
subt →	8	0	25	20	3	14	18	20	16	10

Cipher → I A Z U D O S U Q K

Now, for Decryption,

cipher →	8	0	25	20	3	14	18	20	16	10
key →	17	0	13	2	7	14	1	0	1	0
CT-key →	-9	0	12	18	-4	0	17	20	15	10
Plain →	17	0	12	18	22	0	17	20	15	10

R A M S W A R U P K

d −6 8 9
−c a 7 7

11

$\frac{1}{11}\begin{bmatrix} 11 & -8 \\ -11 & 7 \end{bmatrix} \bmod 26$

26 − 8

26 − 11

$11^{-1}[\ ] \bmod 26$

$11x \equiv 1 \bmod 26$

$26 \overline{)132}\ (5$, -130, remainder 2

130

6. Polyalphabatic →

Key → Lock

Plaintext → Give money

Plain → G i v e m o n e y

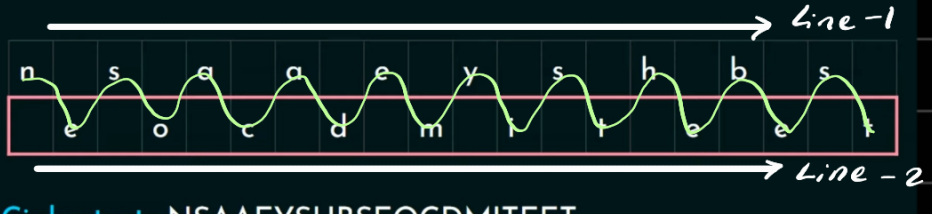
Key → L O C K L O C K L

1. Rail Fence

Rail Fence Technique

Plaintext : neso academy is the best.

Depth : 2



Ciphertext : NSAAEYSHBSEOCDMITEET

2. Row Column Transposition →

1) Columnar Transposition technique:

: FIVE
MINUTES
ENGINEERING

1	2	3	4	5
F	I	V	E	M
I	N	U	T	E
S	E	N	G	I
N	E	E	R	I
N	G			

Key = 43512

C : ETGRVUNEMFTTFTSNNINEEG

Algebric Structure
Semi Group
Monoid
Group
Abelian Group / Commutative

Hack → key len - 4

Chandigarh

c h a n

d i g a

r h _ _

H a c K

1 1 1 1

3 1 2 4 ← position in alphabatical order.

h i h a g _ c d r n a _

① ② ③ ④

h a c n

i g d a

h _ r _

H A C K

1 1 1 1

3 1 2 4

A C H K

1 1 1 1

c h a n

d i g a

r h _ _

sun - key len - 3

Shdy_ - plain text

s t u

d y i

n g _

S U N

2 3 1

u i _ s d n t y g } cypher - text

u s f
i d y
– n g
n s u

s t u
d y i
n g –

→ studying ✓

Playfair key matrix →

5 x 5 matrix

A O C D E
F G H (I/J) K
L M N O P
Q R S T U
V W X Y Z

As alphabets are 26 but '25' blocks are available

Plain text → INSTRUMENTS

Key → MONARCHY

(M) O (N) A (R)
C H Y B D
(E) F G (I/J) K
L P Q (S) (T)
(U) V W X Z

IN ST RU ME NT S
GA TL MZ CL RQ SZ

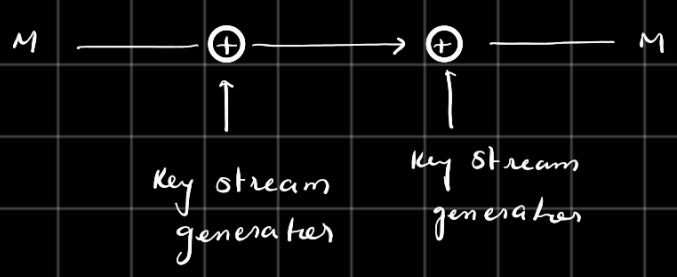
Hill cipher →

Key → (n x n)

Happy

Stream Ciphers →

- Converts into bits.



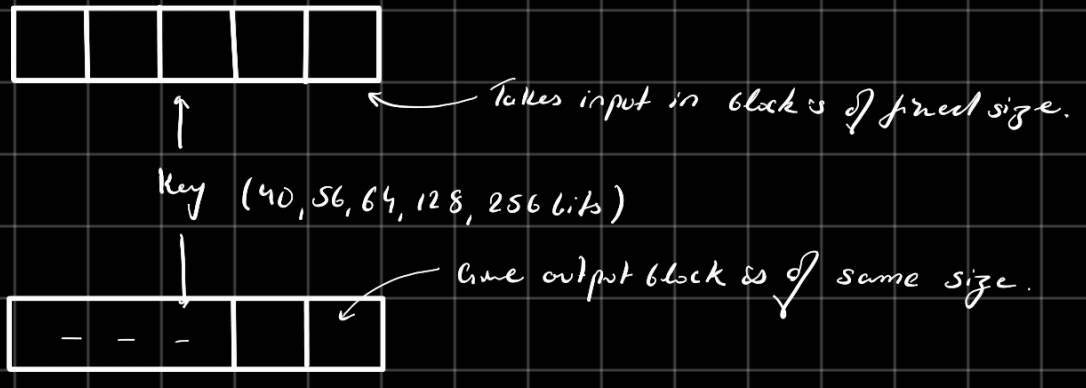
Message → 0100101111

Key → Hello → 0110111 0

⊕ Message ⊕ Key

XOR

Block Ciphers →



Block Ciphers Principle →

1) Number of Rounds
2) Design of function F
3) Key generation

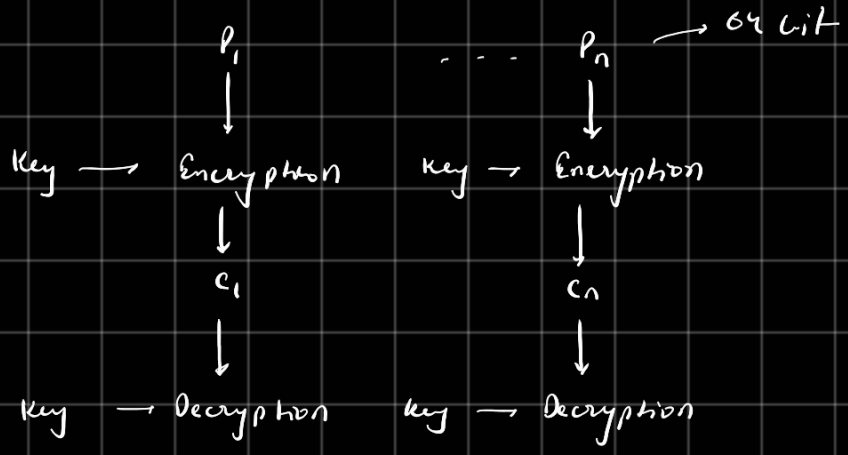
Block Ciphers → Modes of Operation

ECB, CBC, OFB, CTC modes
CFM, CTR

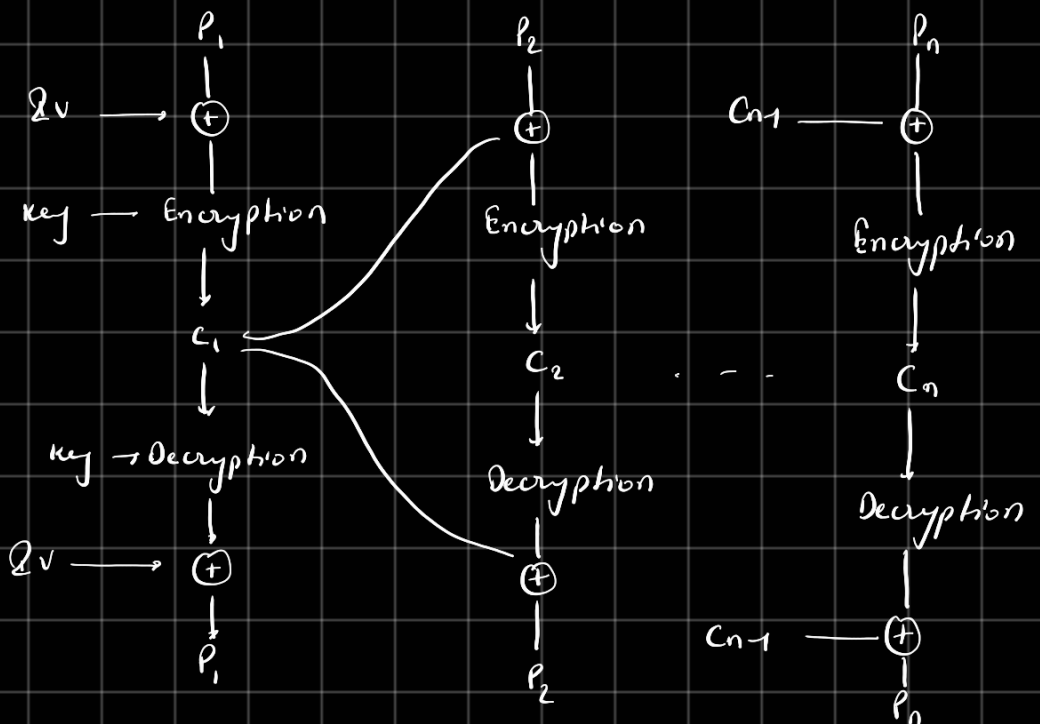
Block Ciphers Algorithm →

1) DES
2) AES
3) Blowfish

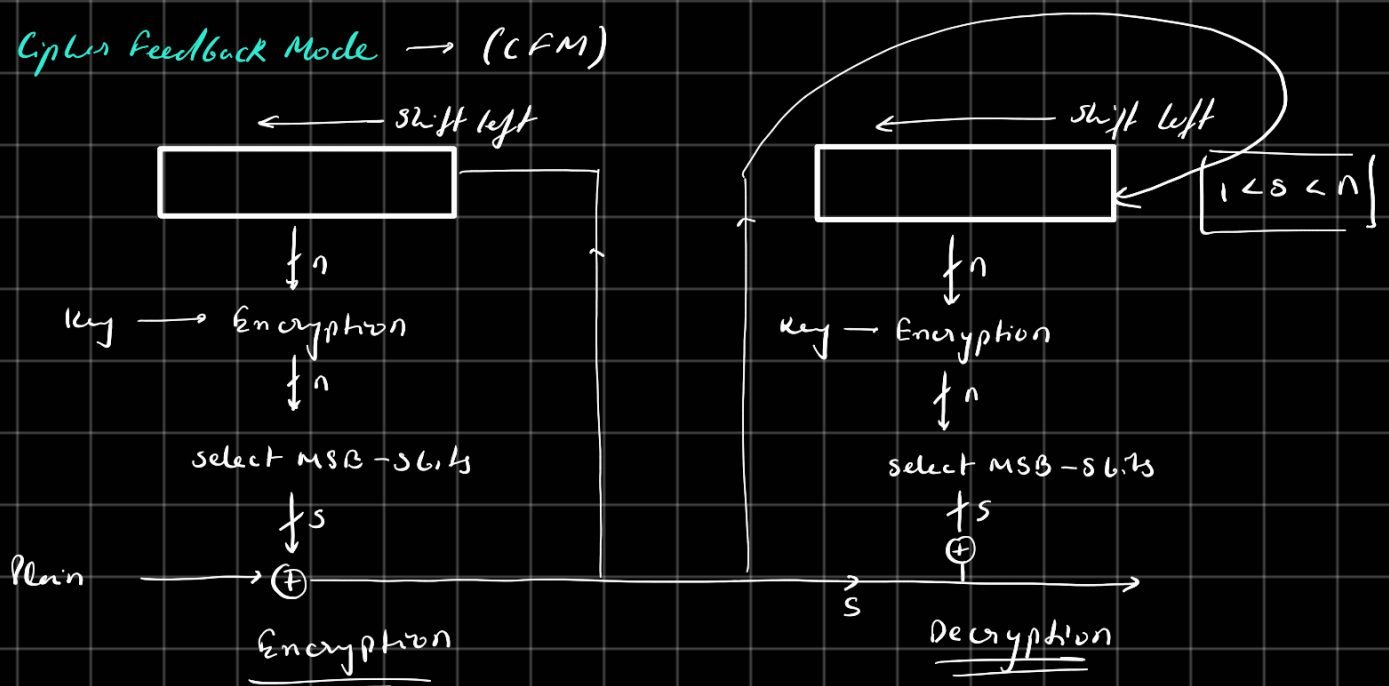
1) Electronic Code Book (ECB) →



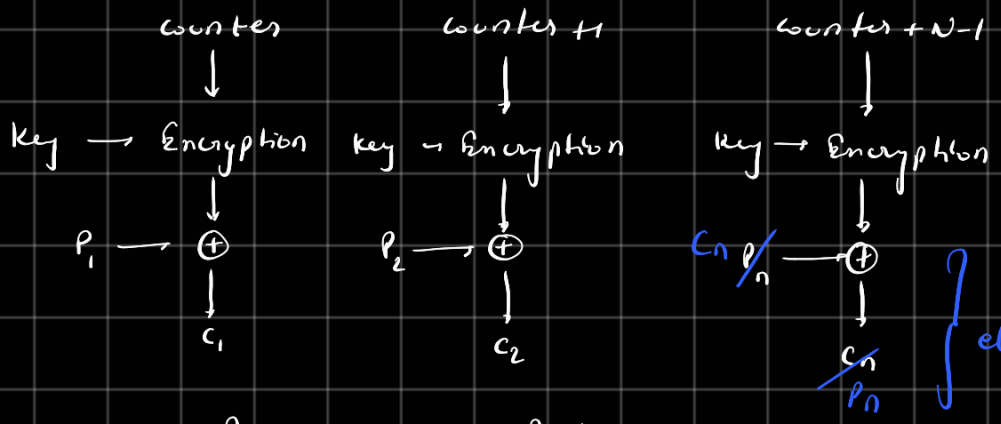
2) Cipher Block Chaining (CBC) →



3) Cipher Feedback Mode → (CFM)



Counter Mode → (CTR)



Decryption

else every thing is the same.

size of counter = size of plain text

Steganography →

Stegano	graphy
covered or concealed	writing

Various forms of steganography →

- Text
- Audio
- Vedio
- Images

Steganography can be used to "conceal" any type of digital content.

Cryptography	Steganography
Known form of communication.	Hidden form of communication.
It converts the message into unreadable form.	Hide the existance of communication.
Alters the overall structure of the data.	Does not alters with the structure.
Key is necessary.	Not necessary.
Final obtained result is called Ciphertext.	final result obtained is called stego media. (image + secret info)
Difficult to crack.	Can be cracked with less effort than cryptography.
Confidentiality, authentication, data integrity and non repudiation.	Confidentiality, authentication.

3 Techniques used in Steganography →

- Least Significant Bit

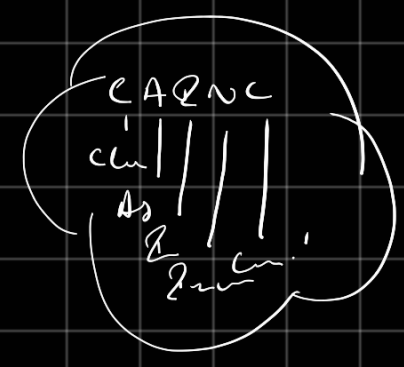
  Is changed with the secret message

- Palette Based Technique

- Secure Cover Selection

Groups →

1. Closure $a, b \in G \quad (a.b) \in G$
2. Associative $a.(b.c) = (a.b).c \quad a, b, c \in G$
3. Identity $a.e = e.a = a \quad a \in G$
4. Inverse $a.a^{-1} = a^{-1}.a = e \quad a, a^{-1} \in G$
5. Commutative $a.b = b.a \quad a, b \in G$



Algebraic Structure – (1)

Semi Group – (1,2)

Monoid – (1,2,3)

Group – (1,2,3,4)

Abelian Group / Commutative group – (1,2,3,4,5)

Ques → $S = \{1, -1, i, -i\}$?

Ans → ① Closure →

	1	-1	i	-i
1	1	-1	i	-i
-1	-1	1	-i	i
i	i	-i	-1	1
-i	-i	i	1	-1

} $\in S$

Rings →

R denoted by $\{R, +, *\}$ $a, b, c \in R$, follows axioms are obeyed →

- $\{CA2NC\}$, Abelian Group $A_1 - A_4$

M_1 ∘ Closure under multiplication $a, b \in R$ then $ab \in R$

M_2 ∘ Associativity of multiplication $a(bc) = (ab)c$ $a, b, c \in R$

M_3 ∘ Distributive laws

$$\left.\begin{aligned} (a+b)c &= ac+bc \\ a(b+c) &= ab+ac \end{aligned}\right\} a, b, c \in R$$

Commutative Rings →

If satisfies the condition →

M_4 ∘ Commutativity of Multiplication $ab = ba$ $a, b \in R$

Integral domain →

It is a commutative ring that obeys the following axioms →

M_5 ∘ Multiplicative Identity $1 \in R$ such that $a1 = 1a = a$ $a \in R$

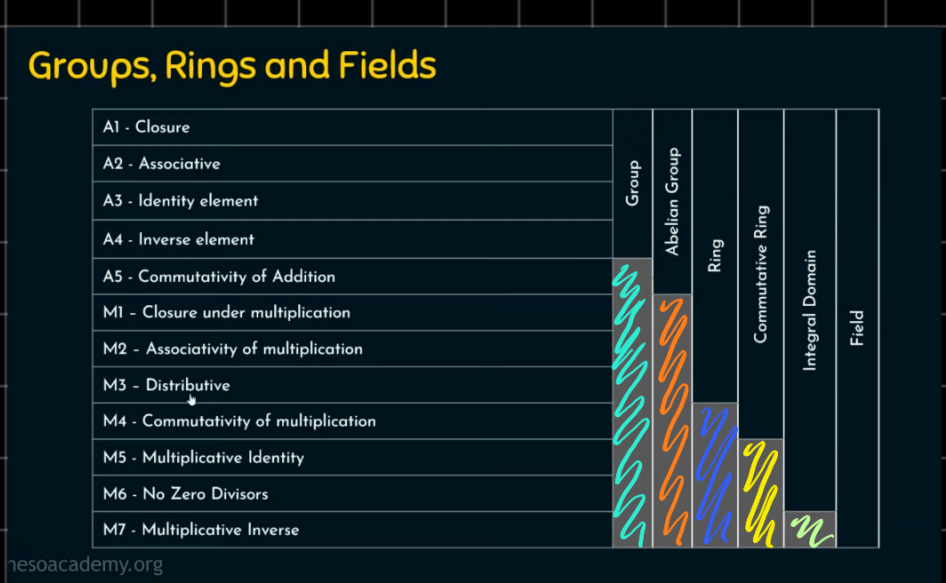
M_6 ∘ No zero divisors If $a, b \in R$ $ab = 0$ either $a = 0$ or $b = 0$.

Fields →

Field F denoted by $\{F, +, *\}$ $a, b, c \in F$, the axioms are →

- F is an integral domain, satisfies $A_1 - A_5$ and $M_1 - M_6$.

M_7 ∘ Multiplicative inverse $aa^{-1} = (a^{-1})a = 1$ $a \in F$



Modular Arithmatic ⟶

① $x \equiv y \bmod n$

② $x \equiv y \bmod n$ (If n divides $(x-y)$)

③ If $x \equiv y \bmod n$ and $a \equiv b \bmod n$ then

$(x+a) \equiv (y+b) \bmod n$

④ If $x \equiv y \bmod n$ and $a \equiv b \bmod n$ then

$(x-a) \equiv (y-b) \bmod n$

⑤ If $x \equiv y \bmod n$ and $a \equiv b \bmod n$ then

$(x \times a) \equiv (y \times b) \bmod n$

⑥ If $x \equiv (y \times z) \bmod n$ then

$x \equiv (y \bmod n \times z \bmod n) \bmod n$

⑦ If $x \equiv (y+z) \bmod n$ then

$x \equiv (y \bmod n + z \bmod n) \bmod n$

Eulers Totient Function ⟶ $\phi(n)$

- $\phi(n)$ for "$n \geq 1$" is defined as the number of +ve integers less than 'n' that are coprime to 'n'.

$\phi(5) = \{1, 2, 3, 4\} \rightarrow ④$ ⟶ Total number of elements which are coprime to 'n'.

$\phi(6) = \{1, 5\} \rightarrow ②$ → Totient number.

→ Totient function.

- When 'n' is a prime number

$\phi(n) = (n-1)$

$\phi(23) = 22$

$\phi(a * b) = \phi(a) * \phi(b)$ where [a and b are coprime]

$\phi(35) = \phi(7 * 5)$

$= \phi(7) * \phi(5)$

$= 6 * 4$

$= 24$

Fermat-Eulas Theorem ⟶

$x^{\phi(n)} \equiv 1 \bmod n$ — where x and n are coprime.

$11^{\phi(10)} \equiv 1 \bmod 10$

$\phi(10) = \phi(2 * 5)$

$= \phi(2) * \phi(5)$ ⇒ $11^4 \equiv 1 \bmod 10$

$= 1 * 4$

$= 4$

$x^{\phi(n) \cdot a} \equiv 1 \bmod n$

$11^4 \equiv 1 \bmod 10$

$11^{4 \times 2} \equiv 1 \bmod 10$

$11 \equiv 1 \bmod 26$

$11^{\phi(26)} \equiv 1 \bmod 26$

$\phi(13 \times 2)$

$\phi(13) \times \phi(2)$

12×1

∴ 12

$11^{12} = 1 \bmod 26$

Fermat's Theorem ⟶

$x^{n-1} \equiv 1 \bmod n$ — where "x" is +ve, "n" is prime.

$x = 3 \quad n = 5$

$3^{5-1} \equiv 1 \bmod 5$

$3^4 = 81$ ⇒ $81 \equiv 1 \bmod 5$

Testing for Primality ⟶

$a^p - a$ ⟶ 'p' is prime if this is a multiple 'p' for all $1 \le a < p$.

If $p = 5$, then $a < 5$ and $a \ge 1$

$1^5 - 1 = 0$

$2^5 - 2 = 32 - 2 = 30$

$3^5 - 3 = 243 - 3 = 240$

$4^5 - 4 = 1024 - 4 = 1020$

} These here all the results are multiple of 5.

∴ Hence 5 is prime

Chinese Remainder Theorem ⟶

$x = a_1 \pmod{m_1}$

$x = a_2 \pmod{m_2}$

$x = a_3 \pmod{m_3}$

$$x = (a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} \ldots) \bmod M$$

$$(a_1 M_1 M_1^{-1} + \ldots) \bmod M$$

$m_1 m_2 m_3 = M$

$M_1 = M/m_1$

$M_1 M_1^{-1} \bmod m = 1$

$x = 2 \bmod 3$

$x = 3 \bmod 5$

$x = 2 \bmod 7$

a_1 2	m_1 3	M_1 35	M_1^{-1} 2	
a_2 3	m_2 5	M_2 21	M_2^{-1} 1	M 105
a_3 2	m_3 7	M_3 15	M_3^{-1} 1	

$M = m_1 m_2 m_3$

$= 3 \times 5 \times 7$

$= 105$

$M_1 = M/m_1$ ⇒ $105/3$ ⇒ 35

$M_2 = M/m_2$ ⇒ $105/5$ ⇒ 21

$M_3 = M/m_3$ ⇒ $105/7$ ⇒ 15

$M M^{-1} = 1 \bmod m$

$35 \cdot M_1^{-1} = 1 \bmod 3$

$3 \overline{)\,70}$ (23, 6, 10, −9, 1) ⇒ $M_1^{-1} = 2$

$21 \cdot M_2^{-1} = 1 \bmod 5$

$5 \overline{)\,21}$ (4, 20, 1) ⇒ $M_2^{-1} = 1$

$15 \cdot M_3^{-1} = 1 \bmod 7$

$7 \overline{)\,15}$ (2, −14, 1) ⇒ $M_3^{-1} = 1$

$x = (2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1) \bmod 105$

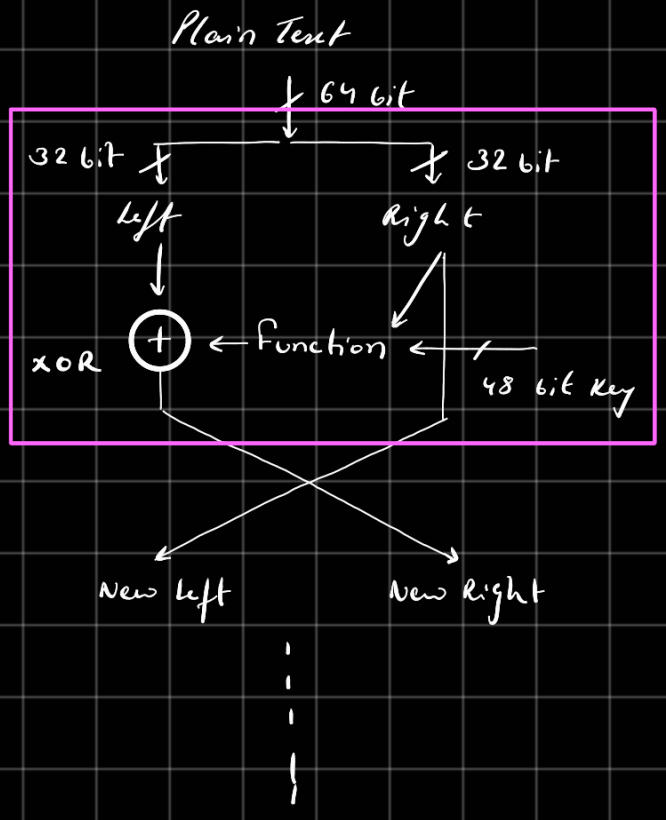
$= (140 + 63 + 30) \bmod 105$

$= 233 \bmod 105$

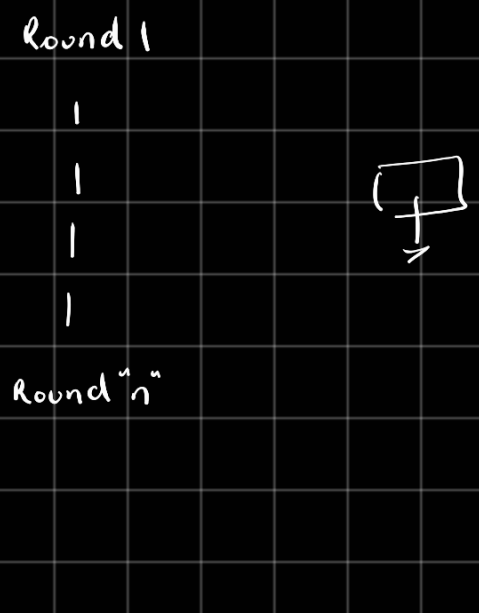
$x = 23$

$105 \overline{)\,233}$ (2, −210, 23)

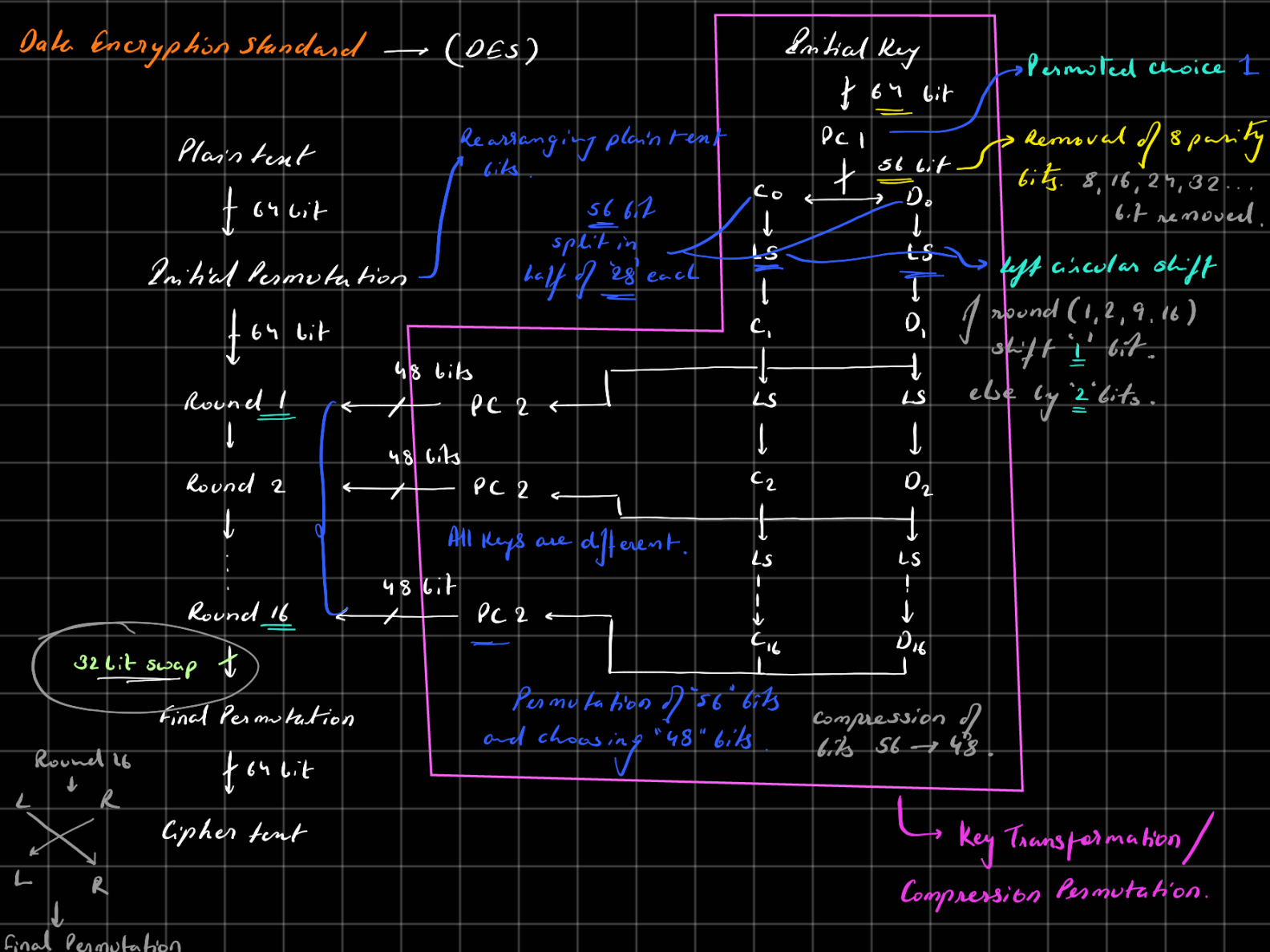
Feistel Cipher →



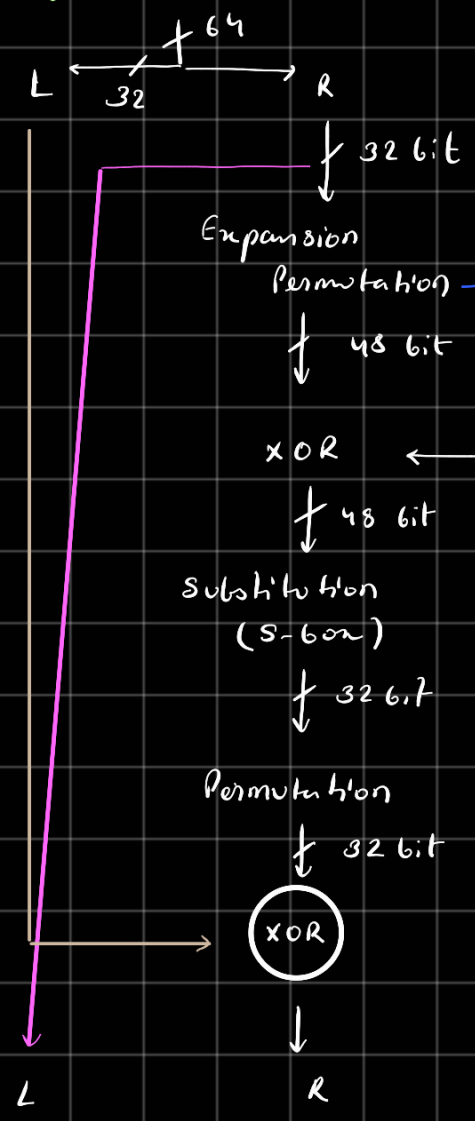
DES is based on Feistal Cipher.



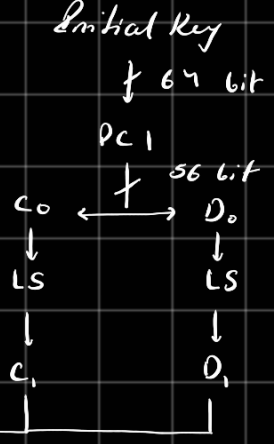
Data Encryption Standard → (DES)

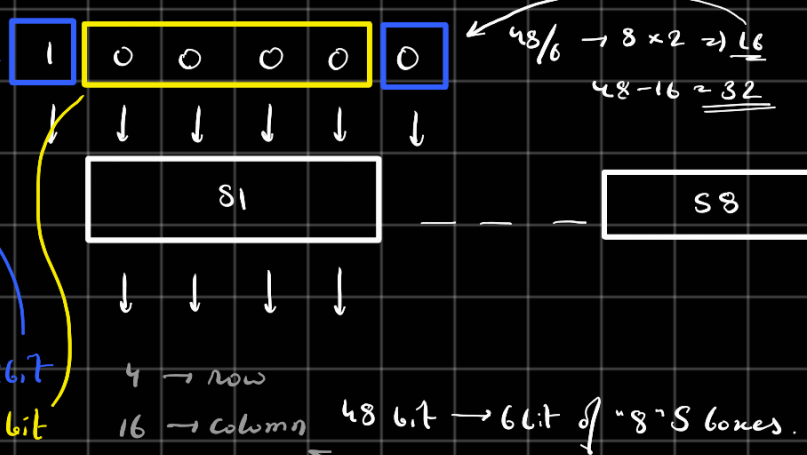


Inside every single round →

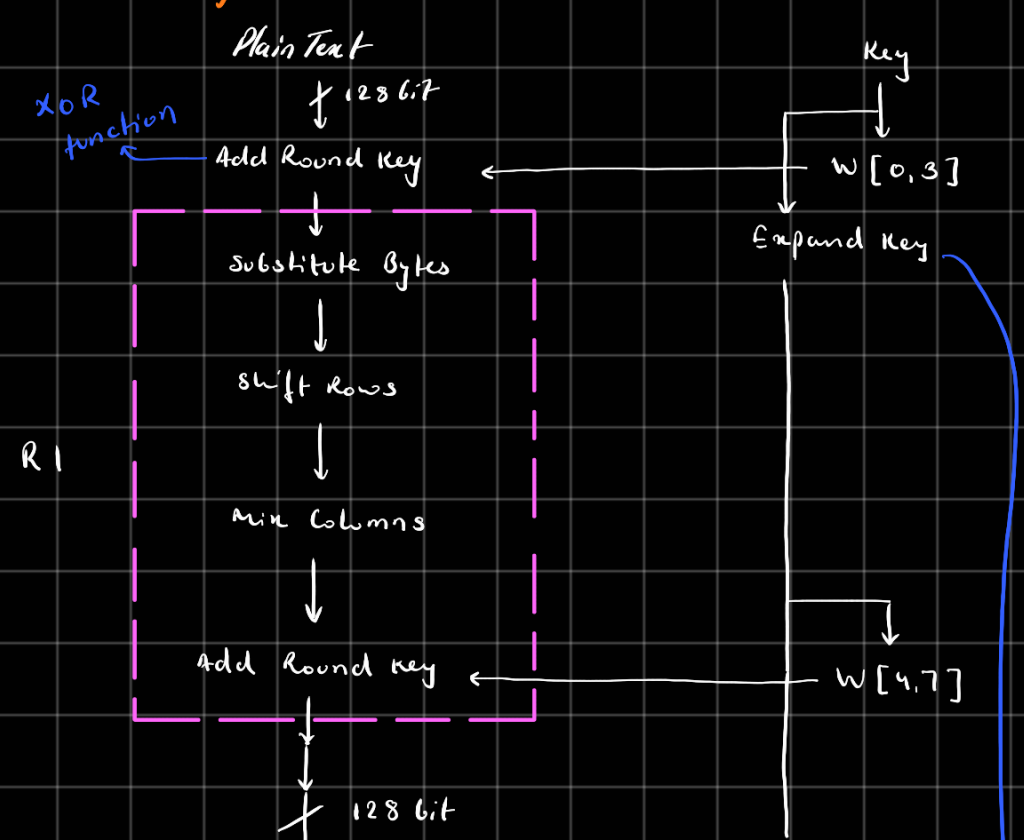


Adding additional "16" bit.





Advance Encryption Standard → (AES)

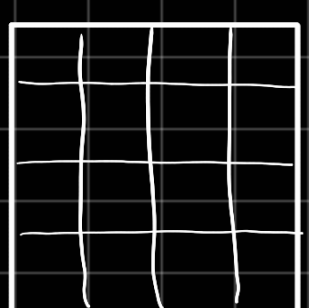


128 → 10 rounds
198 → 12 rounds
256 → 14 rounds

1 word → 32 bit / 4 byte

Input Array (4×4)

16 byte, 128 bit / 4 word



Substitute Bytes

↓

Shift Rows

↓

Add Round Key

↓

128 bit

Cipher Text (128 bit)

R10

final round.

W[40, 43]

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}

4 word

after expansion ⇒

w_0	w_1	....	w_{43}

44 words

State Array (4x4)

16 byte / 4 words

$S_{2,3}$

Index word

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

After every round data is stored in state Array.

4x4

↓

16x16 → Sub Bytes

S-box

↓

4x4

↓

Shift Rows

↓

4x4

↓

[any 4x4 matrix] → Mix column.

↓

4x4

↓

Add round key

↓

4x4

$S_{0,1}$

10100100

5 4

4x4

1 byte → 8 bits

16x16

$S'_{0,0}$

4x4

$S_{0,0}$ . . . .

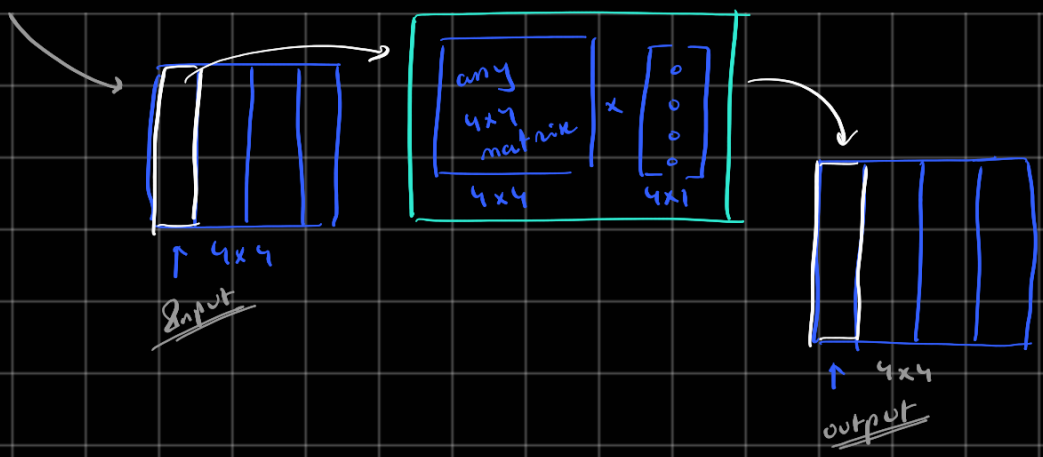
4x4 input

0	$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
1	$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
2	$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{1,3}$
3	$S_{3,0}$	$S_{3,1}$	$S_{3,3}$	$S_{3,3}$

shift left in circular form towards left with respect to index.

4x4 matrix

$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,0}$	→ 1
$S_{2,2}$	$S_{2,3}$	$S_{2,0}$	$S_{2,1}$	→ 2



Blowfish Algorithm →

- Symmetric Key
- Block cipher (64 bit)

DES (Data Encryption Standard) alternative

Block size / Plain text → 64 bit

Key size → variable (32 – 448) bit

sub keys → 18 (P arrays, $P_0, P_1 \dots P_{17}$) of 32 bit

No of sbox (substitution box) → 4 sbox

16 Rounds — same as DES

fast, simple, compact, secure

- fast: XOR and add
- simple, compact: executes in less memory
- secure → because of variable size key

18 - sub key are stored in 'P' array each of 32 bit

P[0] = "243a f631"

⋮

P[17] = "a243 bcc"

All in hexadecimal format

each 32 bit

converting 'P' array →

as variable key [32 – 448]

if 448, then

32 32 — ×14 times

P[0] = P[0] XOR 1st 32 bit of i/p key

⋮

P[13] = P[13] XOR 14th bit of i/p key

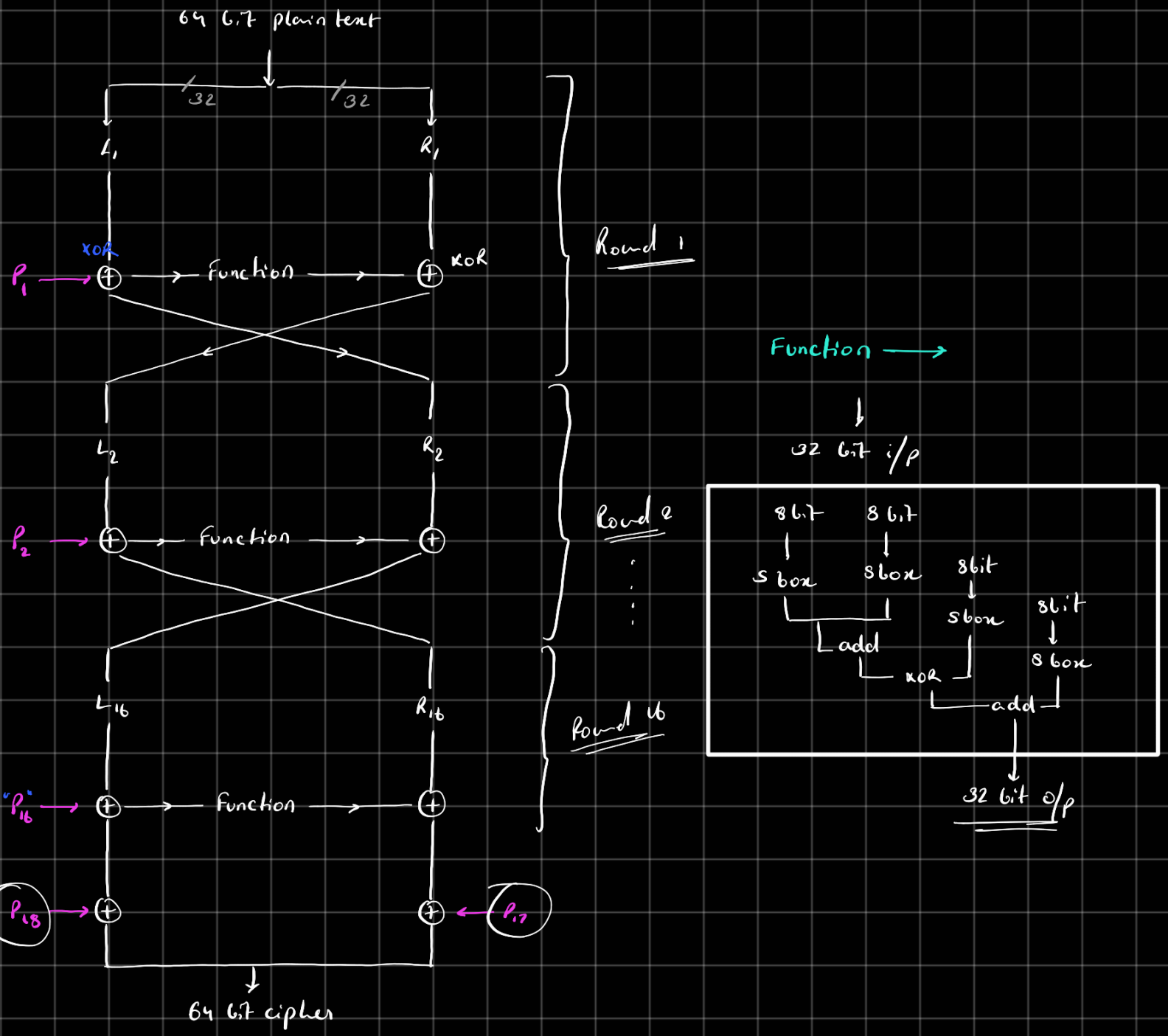
P[14] = P[14] XOR 1st bit of i/p key ← Restart from i.

P[18] . . . . .

New P array | P[0] | P[1] | . . . . . | P[17] |

S box →

- 4 boxes S[0], S[1], S[2], S[3] for encryption + decryption
- 256 entries (32-bit each)



Function →

32 bit i/p

32 bit o/p

Number of Rounds — 16

Sub Key — 18

S-box — 4 of 256

Key size — 32-448 / max number of Key — 14

$32 \times 14 = 448$

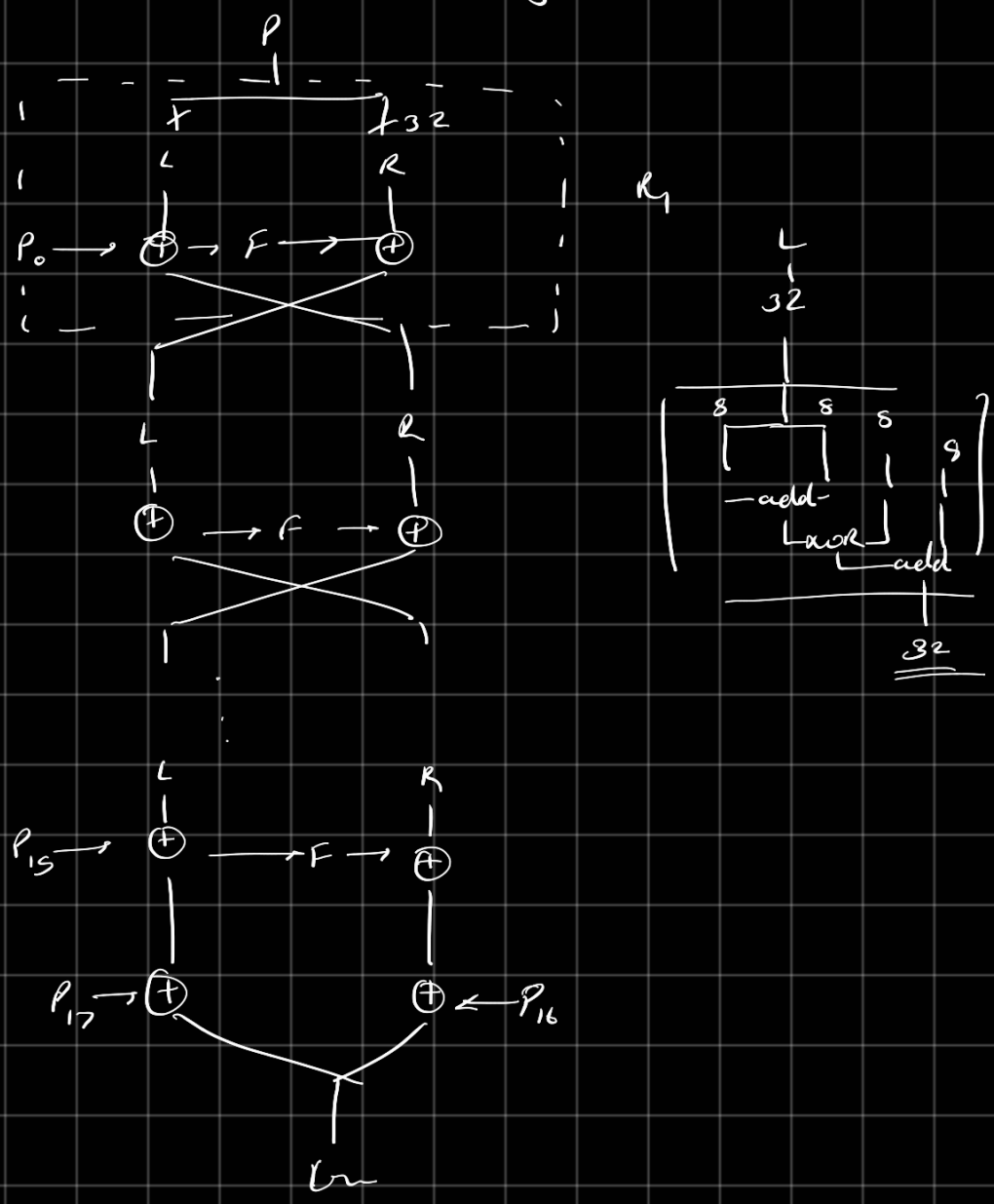
Parray = " "

} all of in hexadecimal with 8 place value

P[0] = P[0] XOR K_1

P[17] = P[17] XOR K_{14}

P[— —] New Parray



Tripped DES →

2 Key →

Plain text
↓ 64 bit
K_1 → DES cypher
↓ 64 bit
K_2 → DES decripting cypher
↓ 64 bit
K_1 → DES cypher
↓ 64 bit
Cypher Text

3 key →

Plain Text
↓ 64 bit
K_1 → DES Cypher
↓ 64 bit
K_2 → DES decryption
↓ 64 bit
K_3 → DES Cypher
↓ 64 bit
Cypher Text

'RC5' — Not done

Unit - 2

Principle of Public Key cryptosystems →

Public Key Cryptosystem —

Asymmetric Algorithm depends on one key for encryption and a different but related key for decryption.

Public Key encryption scheme →

- Plain Text
- Encryption Algorithm
- Public and Private Key
- Ciphertext
- Decryption Algorithm

RSA Algorithm →

- Asymmetric cryptography algorithm
- Block Cipher

1) Choose 2 Prime Number P and Q

$P = 61 \quad Q = 53$

To choose (e, n) and (d, n)

Public Key — Private Key

In both 'n' so common so we compute n first

2) Compute $n = P \times Q = 61 \times 53$

$= 3233$

3) $\phi(n) = \phi(P \times Q)$

$= \phi(P) \times \phi(Q)$

$= (P-1) \times (Q-1) = 3120$

4) Choose 'e' ; $1 < e < \phi(n)$, Coprime to $\phi(n)$.

$e = 17$

- Public Key $(e, n) = (17, 3233)$

5) Determine 'd' as $ed \equiv 1 \bmod \phi(n)$

$d = e^{-1} \bmod \phi(n)$ (d is Multiplicative Inverse of e)

$17 \times d = 1 \bmod 3120$

$d = 2753$

- Private Key $(d, n) = (2753, 3233)$

- finding 'd'

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

$d = \frac{(\phi(n) \times i) + 1}{e}$

$d = \frac{(3120 \times 1) + 1}{17} = 183.58$

until get int format / non decimal.

$d = \frac{(3120 \times 15) + 1}{17} = 2753$

→ Encryption (13, 143)

$C = P^e \bmod n \; ; \; P < n$

$C = 13^{13} \bmod 143$

- $13 \bmod 143 = 13$
- $13^4 \bmod 143 = 104$
- $13^8 \bmod 143 = 91$

$C = [(13^8 \bmod 143)(13^4 \bmod 143)(13 \bmod 143)] \bmod 143$

$C = (91 \times 104 \times 13) \bmod 143$

$C = 52$

→ Decryption (37, 143)

$P = C^d \bmod n$

$P = 52^{37} \bmod 143$

- $52 \bmod 143 = 52$
- $52^4 \bmod 143 = 26$
- $52^{32} \bmod 143 = 130$

$P = [(52^{32} \bmod 143)(52^4 \bmod 143)(52 \bmod 143)] \bmod 143$

$P = (130 \times 26 \times 52) \bmod 143$

$P = 13$

Four possible approaches to attacking the RSA algorithm's →

1. Brute force attack →
   - Trying all possible private keys
   - Defense against the brute-force, use larger number of bits in d.

2. Mathematical attacks →

   Effort of factoring the product of two primes.

3. Timing attacks →

Depends on the running time of the decryption algorithm.

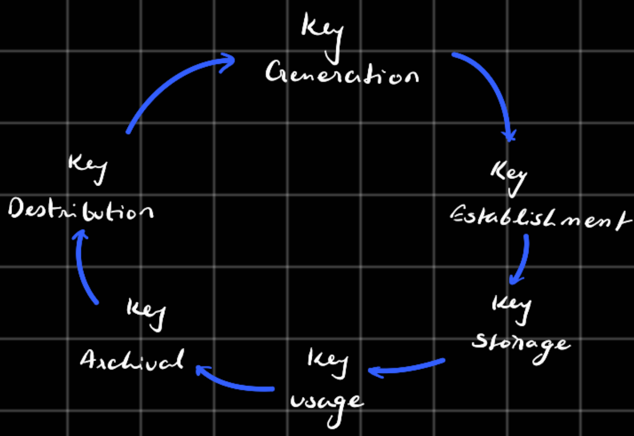
4. Chosen ciphertext attacks →

This type of attacks exploits properties of the RSA algorithm.

Key Management →

- Requirements of Key Management
  - Secrecy of private keys.
  - Assurance of public keys
- Public Key Infrastructure (PKI)
- Digital Certificate
- Certifying Authority (CA)

Key Life Cycle →

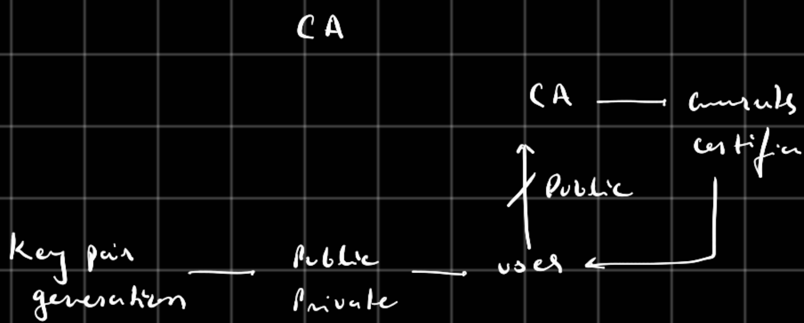


Requirement of Key Management →

- Two specific requirement of key management →
  - Security of private Keys
  - Assurance of public Keys

Public Key Infrastructure (PKI) →

- Assurance of public Keys
- Secrecy of private keys.



- Gen Key
- [illegible]
- [illegible]
- Pub of Public Key

Components of PKI →

- Digital Certificate
- Private key token
- Certification Authority
- Registration Authority
- Certificate Management system.

Digital Certificate →

- Is issued by a third party which proves sender's identity to the receiver and reciever's identity to the sender.

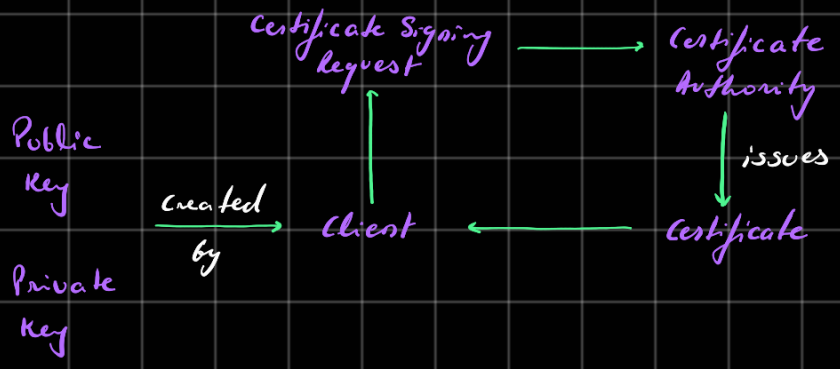
- Digital certificate is issued by Certificate Authority (CA).

- CA issues an encrypted digital certificate containing user's Public Key and other information

Digital Certificate contains →

- Name of certificate holder.
- Serial number used to uniquely identify a certificate.
- Expiration date.

Process of Obtaining Digital Certificate →



Key functions of CA →

1. Generating Key pairs :-
   CA may generate key pair independently or jointly with the client.

2. Issuing digital certificates :-
   CA provides certificate after providing required credentials.

3. Publishing Certificates :-
   CA need to publish certificates so that users can find them.

4. Verifying Certificates :-
   Verification of CA signature on clients digital certificate.

5. Revocation of Certificates :-
   CA revokes the certificate issued due to compromise of Private Key;

or less of trust in the client.

Classes of Certificate →

- Class 1

  These certificates can be easily acquired by supplying an email address.

- Class 2

  These required additional personal information to be supplied.

- Class 3

  These can only be purchased after checks have been made about requestor's identity.

- Class 4

  They may be used by governments and financial organizations needing very high levels of trust.

Diffie-Hellman Algorithm →

- Not an encryption algorithm.
- Used to exchange secret key by the help of asymetric encryption.

→ Algorithm

1) Consider a prime number 'q'

2) Select 'α' such that it must be primitive root of 'q' and [α < q]

'a' is primitive root of 'q' if

$a \bmod q$

$a^2 \bmod q$

$a^3 \bmod q$

⋮

$a^{q-1} \bmod q$ gives result → {1, 2, 3, ... q-1}

→ No value should be repeated.

3) assume 'X_A' (private key) of A and $X_A < q$

$Y_A = \alpha^{X_A} \bmod q$

↓

Public Key of A

4) assume 'X_B' (private key) of B and $X_B < q$

$Y_B = \alpha^{X_B} \bmod q$

↓

Public Key of B

5) Now we will calculate secret key

" To calculate the secret key, both the sender's and receiver will use public keys.

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

Y_B → public key ← Y_A

Known to all

except (X_A, X_B) all are global / known to all

→

If $q = 7$ is prime,

$\alpha = 5$ is primitive root of 'q'

① Private key $X_A = 4$

$$Y_A = \alpha^{X_A} \bmod q$$
$$= 5^4 \bmod 7$$
$$= 2$$

$=$ Public key of 'A'

② Private key $X_B = 3$

$$Y_B = \alpha^{X_B} \bmod q$$
$$= 5^3 \bmod q$$
$$= 6$$

$=$ Public key of 'B'

③ Calculating Secreet Key

$$K_1 = (Y_A)^{X_B} \bmod q$$
$$= 2^3 \bmod 7$$
$$= 8 \bmod 7$$
$$= 1$$

$$K_2 = (Y_B)^{X_A} \bmod q$$
$$= 6^4 \bmod 7$$
$$= 1$$

MAC and SHA

```
   17
7)125
 -7
  55
 -49
   6
```

```
   1
   3
  36
  36
 216
1080
1296
```

```
   185
7)1296
 -7
  59
 -56
  36
  35
   1
```

Authentication Requirements →

1. Disclosure →

   Release of message contents to any person

2. Traffic Analysis →

   Discovery of the pattern of traffic between parties.

3. Masquerade →

   The creation of messages by an opponent that are purported to come from an authorized entity.

4. Content Modification →

   Changes to the contents of a message.

5. Sequence modification →

6. Timing modification →

   Delay or replay of messages

7. Repudiation →

   Denial of transmission of message by source.

Message Authentication →

- Procedure to verify that recieved messages is from verified source
- Also verify timeliness and sequencing
- Digital Signature is an authentication technique also contain denial of service from source and destination.

Types of Authentication function →

1) Message Encryption →

2) Message Authentication Code (MAC) →

$C(M, K)$ = fixed length code (MAC)

- C ↑ Authentication function
- M ↑ Message
- K ↑ Key
- MAC ↑ This will act as an authenticator here.

→ example message → 1 MB
MAC compresses → 1 KB

3) Hash function (H)

- Independent of key.

$H(M)$ = fixed length code (Hash code 'h')

- H ↓ Authentication
- M ↓ Message
- Message Digest

1) Message Encryption →

① M → E → Ciphertext → D → M

- Sender: E ↑ K_1
- D ↑ K_1: Receiver

② M → E → Ciphertext → D → M

- Sender: E ↑ Public B
- D ↑ Private B: Receiver

- ✓ Data confidentiality is maintained.
- ✗ Authentication is not maintained, because we can't verify the source/sender.

③ M → E → Cyphertext → D → M

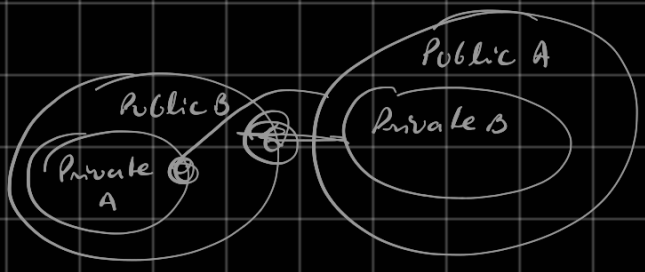
Sender — Private B — Public B — Receiver

- ✗ Confidentiality, because any one decrypt the message
- ○ Authentication, the reciever know the source, verified.

④ M → E → M, KP_A → E → Ciphes → D → M, KP_B → D → M

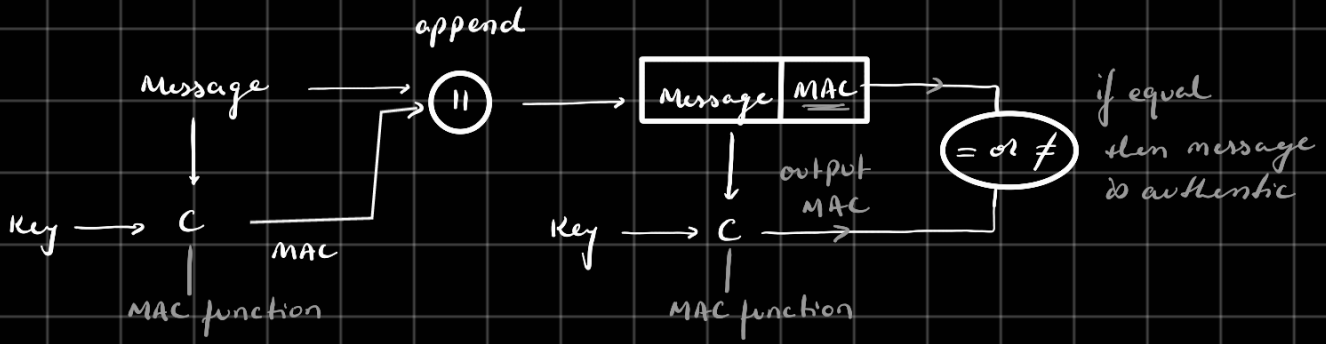
Sender — Private A — Public B — Private B — Public A — Reciver

- ○ Confidentiality
- ○ Authentication



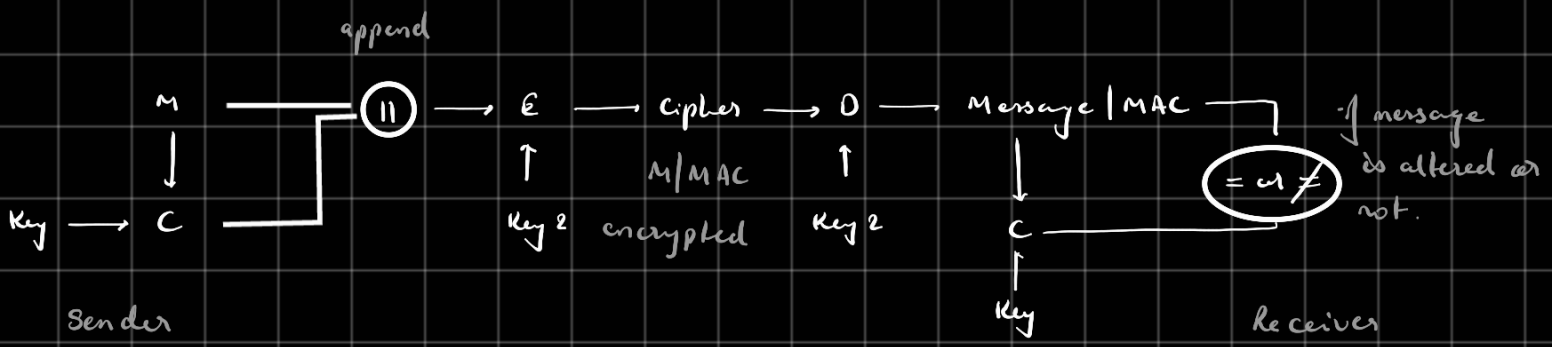
2) Method Authentication Code (MAC) →

① MAC for authentication

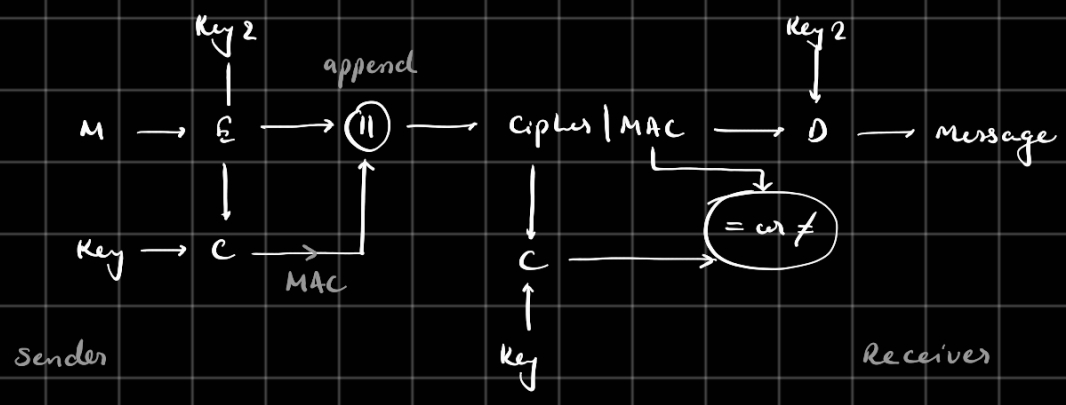


- ✗ No confidentiality, because no sort of encryption is taking place.
- ○ Authentication, if both MCA are not same, can conclude data breach has takes place.

② Authentication tied to plain text

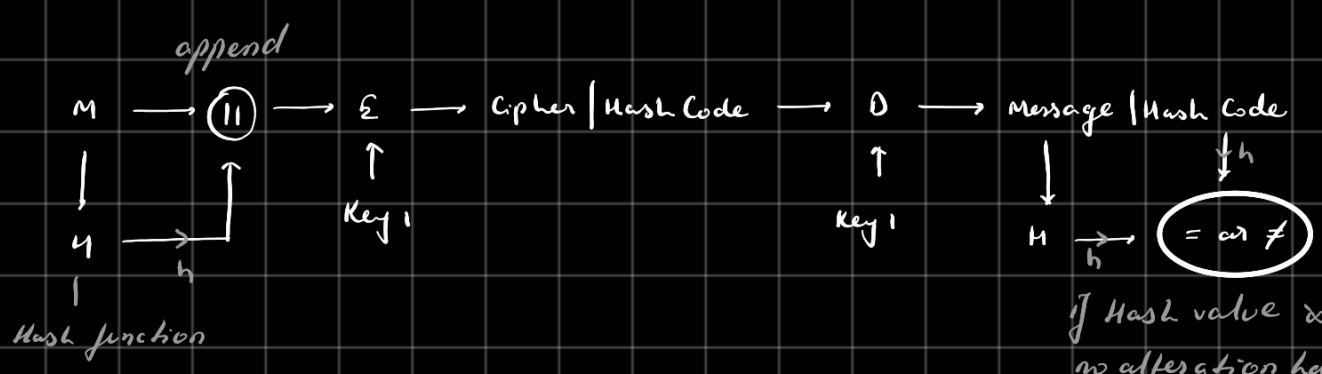


③ Authentication tied to cipher text



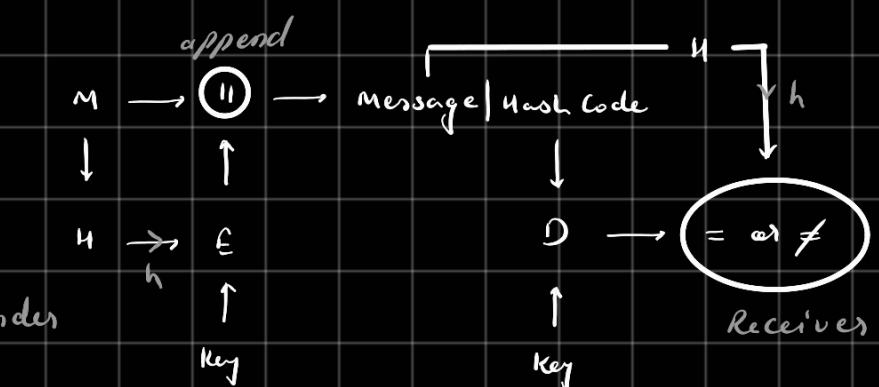
3. Hash Function →

①



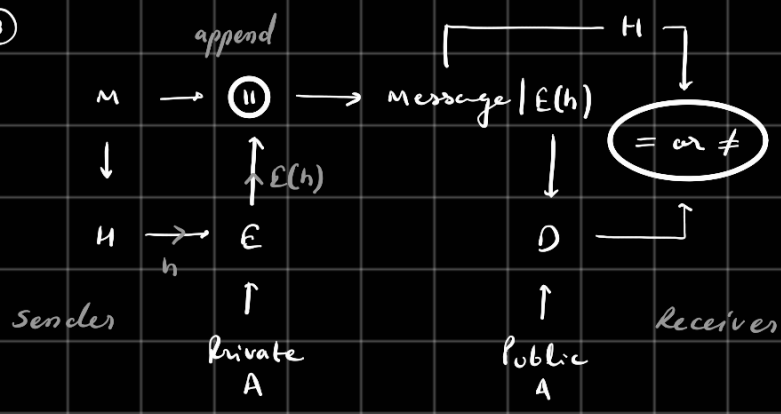
- Confidentiality, by the help of encryption. (use symmetric key)
- Authentication, by matching the hash value.

②



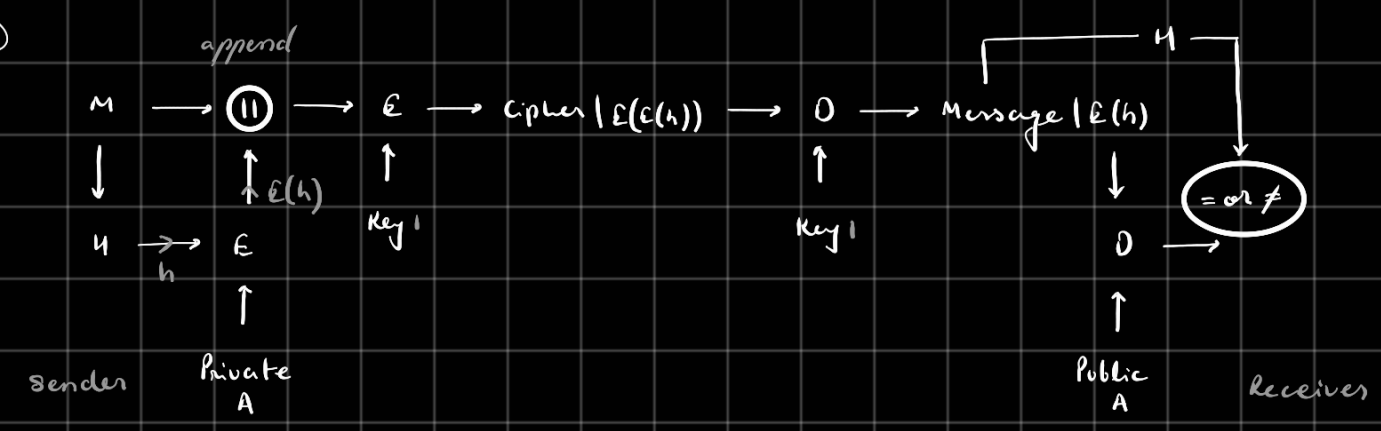
- ✗ Confidentiality, as the message is not encrypted.
- ○ Authentication, the Hash code is encryped. (Symmetric Key)

③



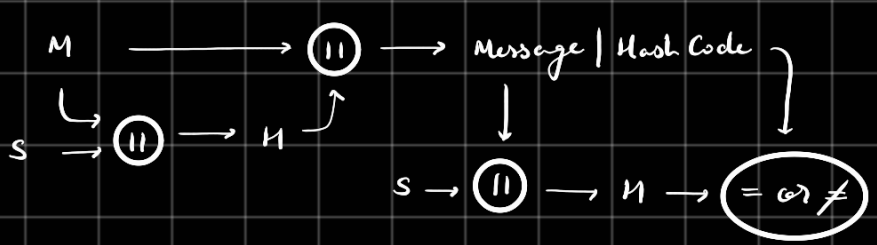
- ✗ Confidentiality, the message is not encrypted
- ○ Authentication, Hash value is encrypted by source private key.

④



- ○ Confidentiality (Symmetric Key)
- ○ Authentication

⑤ Sender and Receiver will have a secret code 'S'.



- ✗ No confidentiality.
- ○ Authentication, as only sender and receiver has the secret code.

(6)

Features of Hash function →

1) Input → H → Hash value — easy

   Hash value → H → Input — difficult

2) $Input_A \rightarrow H \rightarrow Hash\ value_A$

   $Input_B \rightarrow H \rightarrow Hash\ value_A$

   } No different inputs can have the same hash value.

3) $Input_A \rightarrow H \rightarrow$ Hash value A

   $Input_B \rightarrow H \rightarrow$ Hash value B

   } every input have unique output

4) Input A (10 MB) → H → Hash value (1KB)

   Input C (7 MB) → H → Hash value (1KB)

   } length is fixed

   Input length varies.

5) Input A → H → Hash value

   Input A → H → Hash value

   } both Hash values are the same

Advantages →

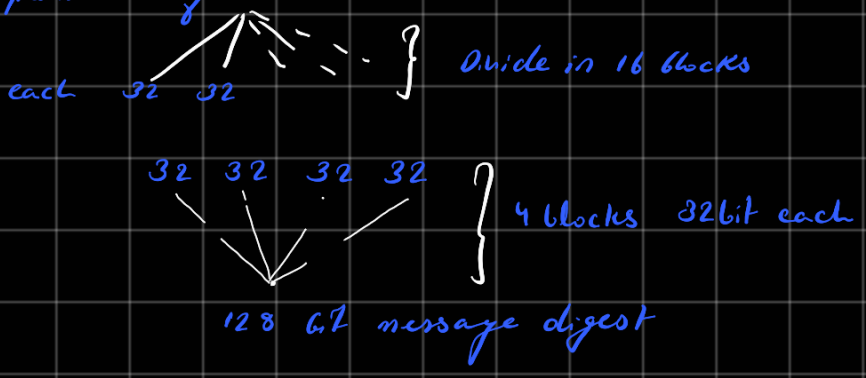
1) Password storage
2) Data integrity
3) Message Authentication
4) Fast computation

Disadvantages →

1) Finding inputs with same hash value
2) Limited input size
3) Rainbow table attack

MD5 →

- Takes plain text of 512 bit block



- Message Digest is produced through five steps →
  1. Padding
  2. Append length
  3. Divide input in 512 bit block
  4. Initialize chaining variables & process blocks and 4 rounds

- Use of MD5 Algorithm →
  1) Security

     input of any size → 128 bit Hash value

  2) Hash value stored / parity bit (detection of alteration and missing bit)

1. Append Padding Bits →
   - message ≡ 448 (modulo 512)
   - message + ≤64 multiple of 512

   - Padding is done even if message ≡ 448 (modulo 512)

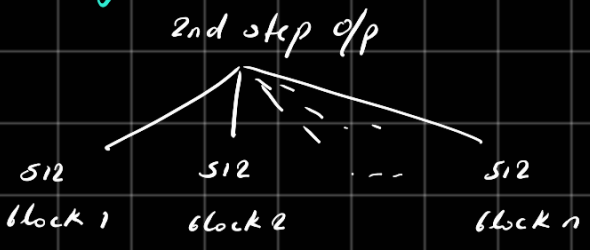
     where 1st element is 1 followed by 0s.

2. Append length →

[message + padding] + 64 = multiple of 512

- after padding +64 bits (at the end) } used to record the length of original input.

3. Deviding →



4. Initialising →

'4' chaing variables →

(A), (B), (C), (D) → value predefined (Hexadecimal)

each of 32 bit

5. Processing →

(1) Copy '4' chaing variables into some corresponding variable

(A) = a

(B) = b

(C) = c

(D) = d

(2) Divide 512 bit blocks into 16 block of 32 bit blocks

(3) "Four Rounds"

16 subblocks and Constant (K)

Calculated in 0st round ② nd ③ ④ round

for ①st round

$$a = b + ((a + \text{Process } P(b,c,d) + m[i] + T[k]))$$

②nd

$$b = a + ((b + \text{Process } P(a,c,d) + m[i] + T[k]))$$

Advantages →

- Used in file management
- Linux and Unix command promp.
- 'Help' the table shows up.

Disadvantages →

- Can't deal with large message

SHA-512 (Secure Hash Algorithm) →

128 bit → SHA 128

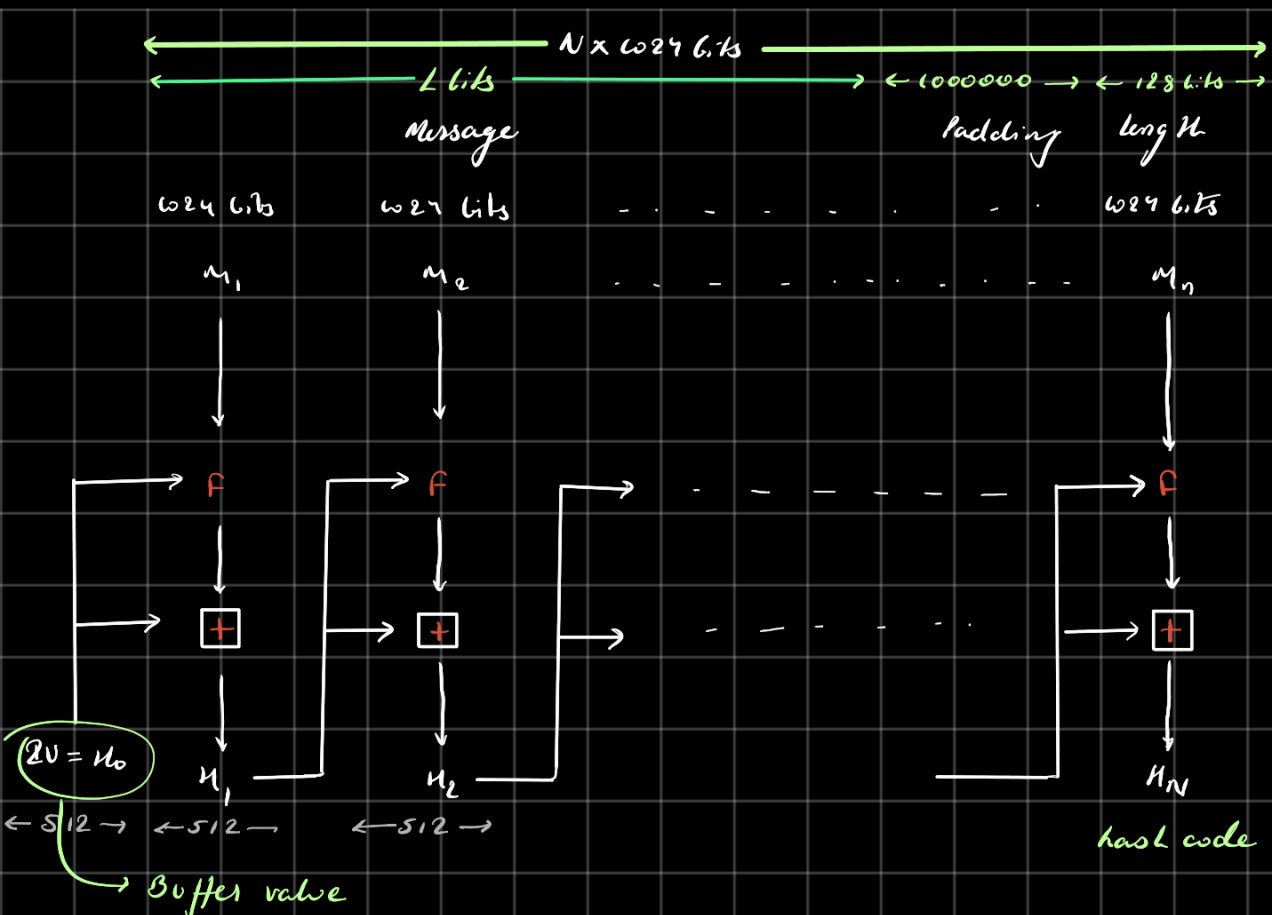
256 bit → SHA 256

512 bit → SHA 512

1. Plain text - 1024 bits each block
2. Number of Rounds - 80
3. Each round will produce a word & word (w) → 64 bits

→ Used for →

- Email Addressing
- Password Hashing
- Digital Record



1. Append Padding bits →
   - Message is padded so length ≡ 896 modulo 1024
     → message = total length of message − 128 (L).
   - Also if length ≡ 1024
   - Number of padding bits in range 1 – 1024
   - Padding consists '1' followed by '0's.

2. Append Length →
   - Block of 128 bits appended to the message.
   - Representing the original length.

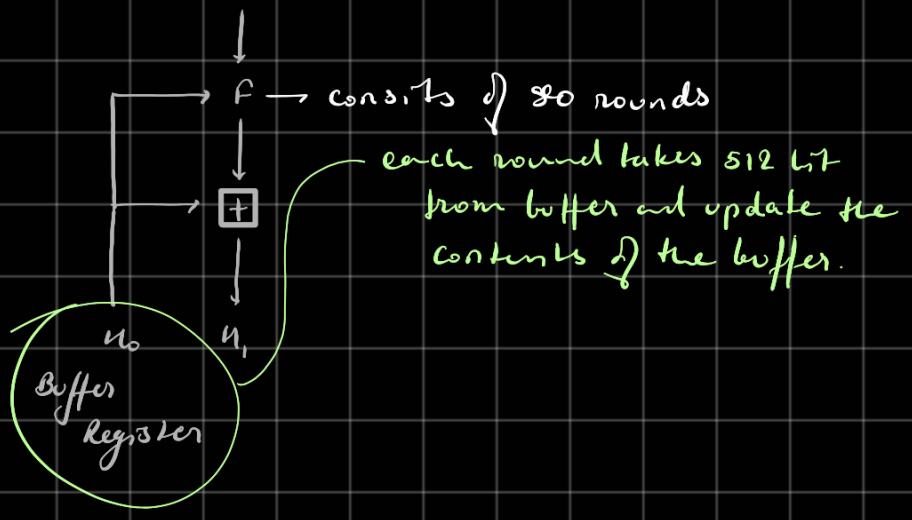
3. Initializing Hash Buffer →
   - 512 bit buffer
   - hold intermediate and final result of hash function

   (A) (B) (C) (D) (E) (F) (G) (H) } 8 registers

   64 bit each

   (Hexadecimal values)

4. Message Process of 1024 bit



5. Output →

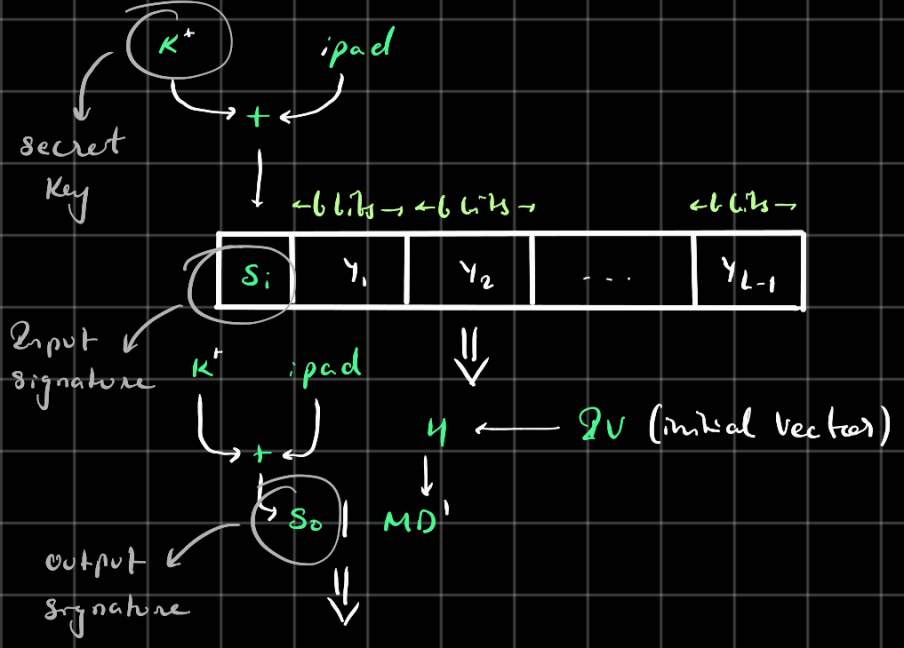
- After all blocks are processed the output from Nth stage ⇒ 512 bit message digest.

HMAC → (Hashed based Message Authentication Code)

- Based on MAC
- MAC + Hash → HMAC

used in IP security.

- Message —easy→ Hash
  Hash —difficult→ Message
- Less effect from collision



K^+ → padding with "0"

$S_i = K^+ \oplus ipad$

$S_o = K^+ \oplus ipad$

$MD' = H(S_i \| MD)$

$MD = H(S_o \| MD')$

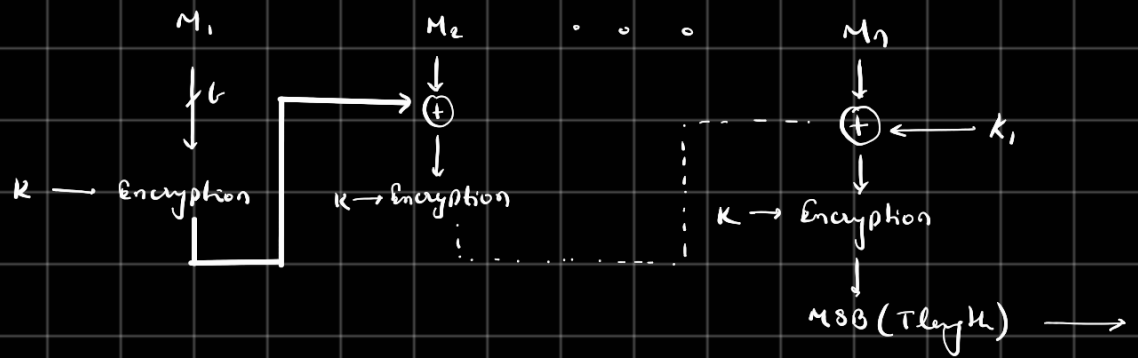
H ← Input Vector

↓

MD

CMAC (Ciphers - based Message Authentication Code) →

- Block ciphers based on MAC



$C_1 = E(K, M_1)$

$C_2 = E(K, [M_1 \oplus C_1])$

⋮

$C_n = E(K, [M_n \oplus K_2 \oplus C_{n-1}])$

Advantages →

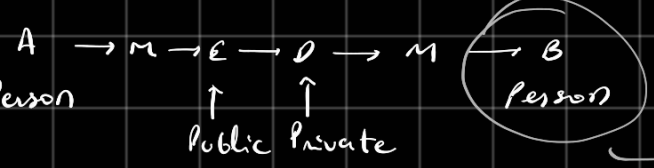
- Use existing functions
- Encryption algorithm helps in resolving collision.

Disadvantages →

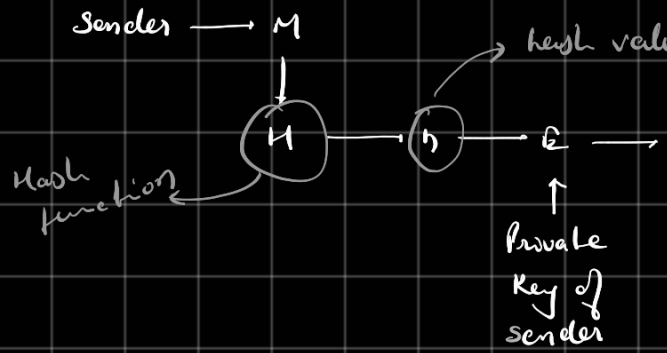
- Encryption Algorithm can be much slower than hash function.

Digital Signature →

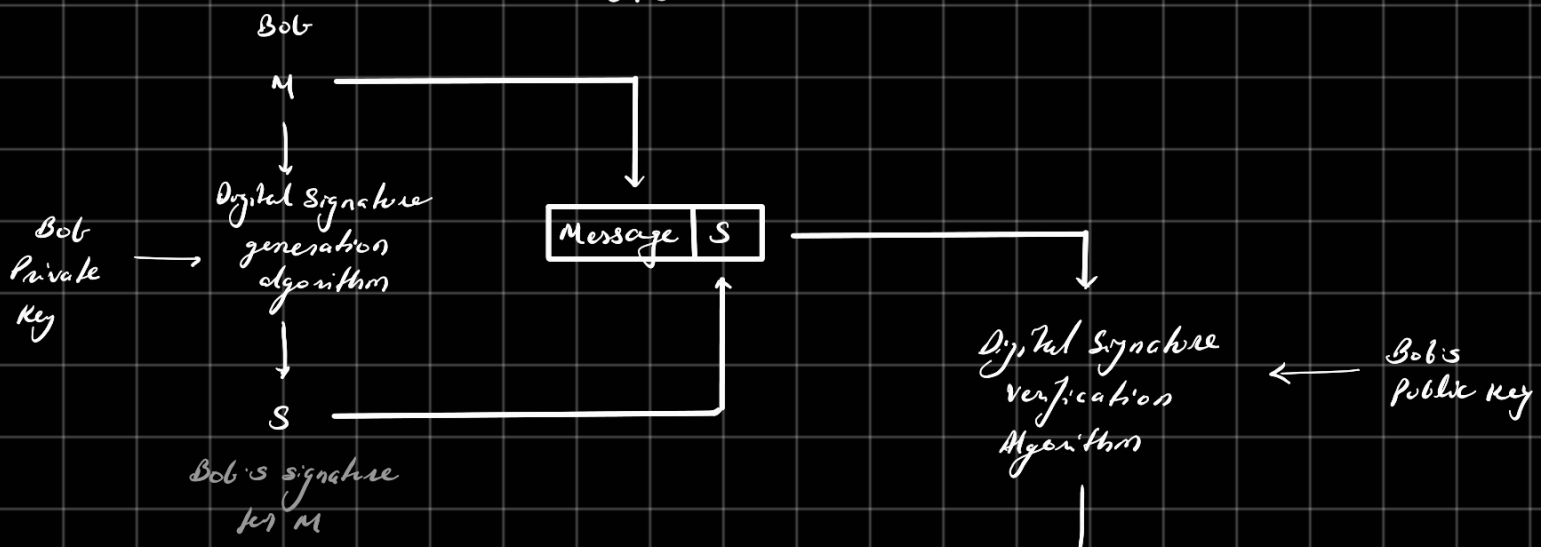
Is a mathematical technique used to validate the authenticity and integrity of message, software or any digital document.



→ Can't authenticate who the sender is if 'A' or some one else.

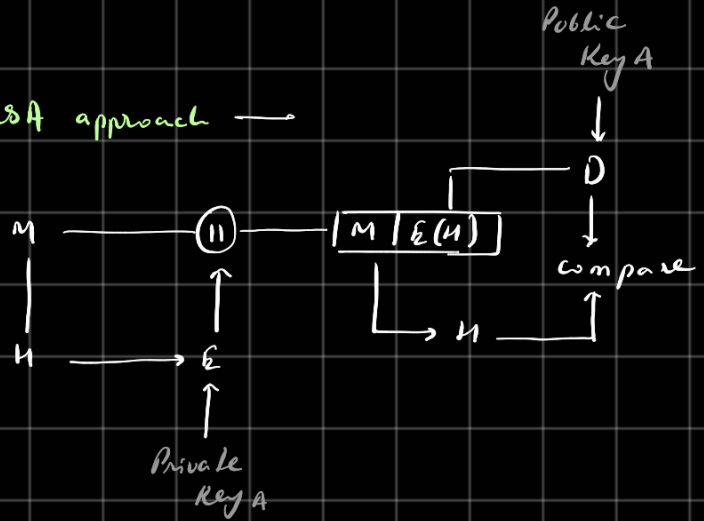


• To authenticate the sender

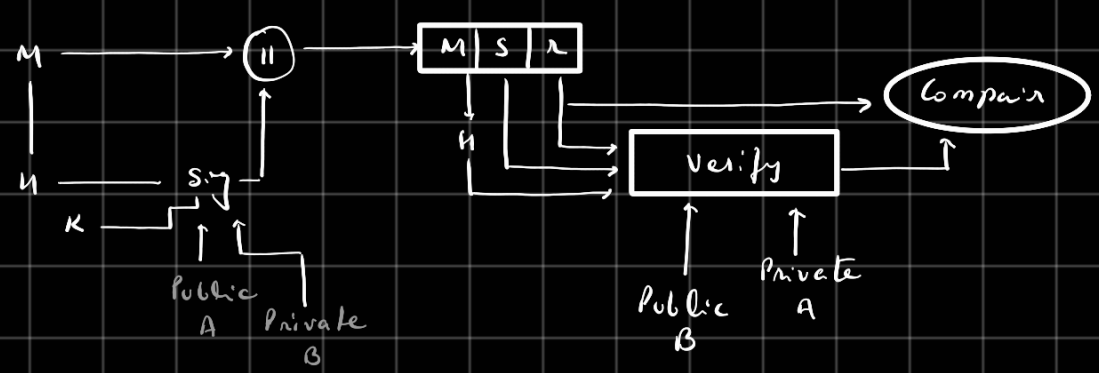


→ DSS (Digital Signature standard uses SHA)

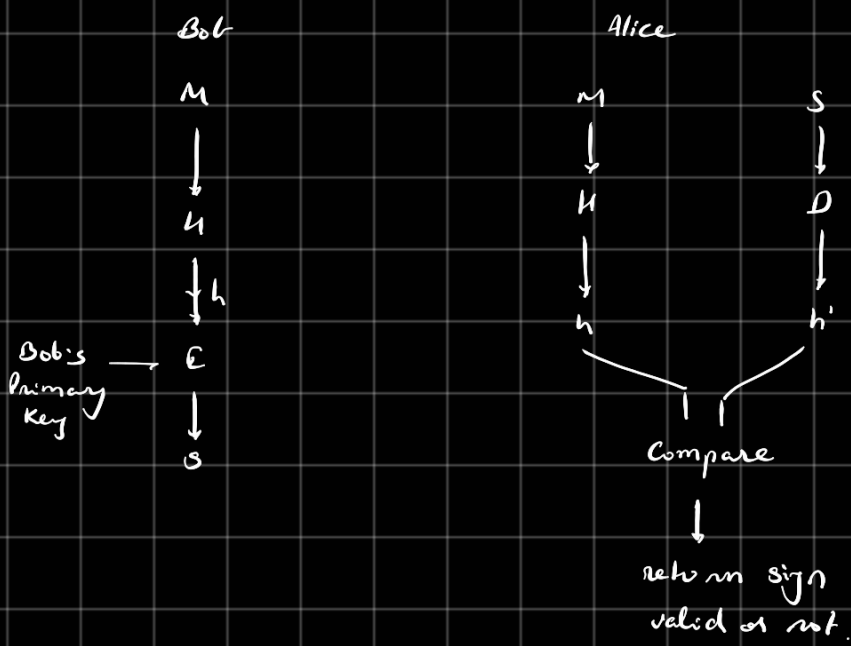
① RSA approach →



② DSS approach



Elements of Digital Signature Process →



Properties of Digital Signature →

- Date and Time of signature.
- Authenticate the contents of signature.
- Must be verifiable by third party.

Digital Signature Certificate →

DS ✓
DSS ✓
DC

Elgamal Digital Signature —

- Asymmetric encryption algorithm

1) Select a prime number 'q'
2) Select a primitive root 'α' of 'q'
3) Generate a random integer 'X_A', $1 < X_A < q-1$
4) Compute $Y_A = \alpha^{X_A} \bmod q$
5) Private key = X_A

   Public Key = $\{q, \alpha, Y_A\}$
6) Generate hash code (m) for the plain Text (P)

   $m = H(P) \quad 0 \leq m \leq q-1$
7) Generate a random Integer K

   $1 \leq K \leq q-1$ and $\gcd(K, q-1)$
8) Calculate S_1 and S_2

   $S_1 = \alpha^K \bmod q$

   $S_2 = K^{-1}(m - X_A S_1) \bmod (q-1)$

9) Signature Pair (s_1, s_2)

10) Now at B's side,

Calculate V_1 and V_2

11) $V_1 = \alpha^m \bmod q$

$V_2 = (Y_A)^{s_1} \cdot (s_1)^{s_2} \bmod q$

If $V_1 = V_2$

⟶ Signature valid

Digital Signature Scheme ⟶

- ElGamal Scheme
- Schnorr Scheme
- Digital Signature Algorithm (DSA algorithm)

Schnorr Digital Signature ⟶

Advantage : Speed

Parameters : p, q, a, s, v, r, x, y

p : prime number

q : factor of $p-1$

a : $a^q = 1 \bmod p$

s : $0 < s < q$ ⟶ secret

v : $a^{-s} \bmod q$ ⟶ Public Key

Signing ⟶

- Choose random 'r'
- Compute $x = a^r \bmod p$
- Concatenate $e = H(M \| x)$
- Compute $y = (r + se) \bmod q$

Send ⟶

Message, M

Signature (e, y)

Verification →

- Received
- Known Publicly

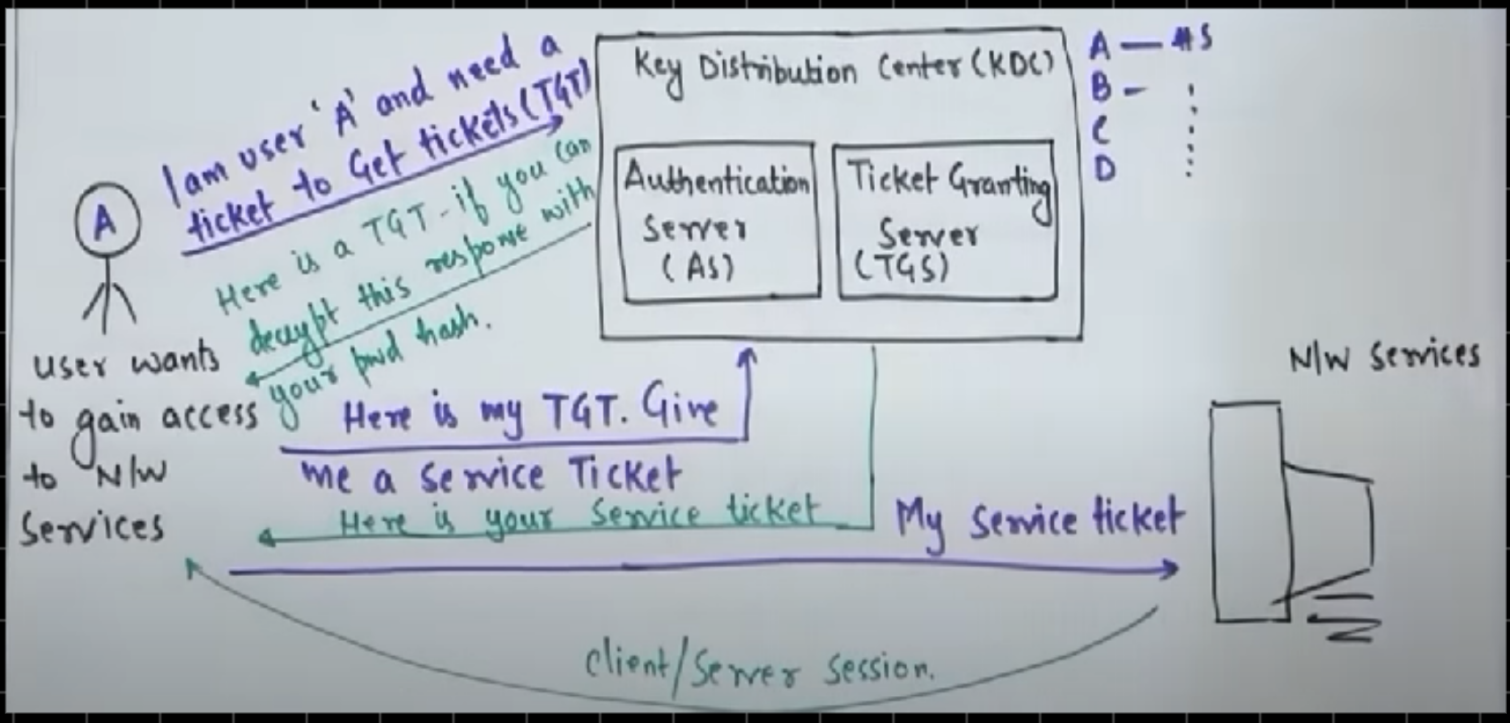
- Compute :

$$x' = a^r v^e \bmod p$$

- Check

Kerberos →

- Computer networking authentication Protocol works on the basis of "tickets".
- Allow nodes to communicate over a "non secure" network to prove their identity to one another in a secure manner.
- Symmetric Key
- Client - server
- Requires a Trusted Third Party



- Limitation —

1. All the keys of users and client are stored in KDC any compromise may lead to losse in data confidentiality.

2. KDC od TGS are one point of faitue may cause is no new secession goreration

It provides →

- Authentication
- Authorization
- Confidentiality

Fire wall →

Protects unauthorised access of your internal data from external network.

- It stops any threatening activity to enter inside of your private network.
- Acts as an traffic controller.

Risks of not having a firewall →

1. Open Access
2. Compromise of Data
3. Network crashes

Functions of Firewall →

1. Network Threat
2. Identity Protection
3. Network traffic controll
4. Records and Repoarts all events.

Limitation of firewall →

1. Can't stop user from accessing malacious sites.
2. Can't scan any virus fitted file of software.
3. Can't secure infected system.
4. Some times may encrypted files are passed with out scanning them due to peformance related issue (it takes time to decrypt → scan → encrypt it and send it to its designated location.

Types of Firewall →

1. Application level
2. Packet Filter
3. Unified Threat Management
4. Next generation Firewall } both TCP and Deep packet tracing
5. Multi level inspection

Firewall VS Antivirus →

Intruders →

- Unauthorised personal attacking and stealing confidential data and sealing it to third party.
- Which aims to misuse the information against you.

3 - Types of Intruders →

1. Masquerade →
2. Misfeasor →
   - Authorised Individual misusing granted permission
3. Clandestine User →
   - Those who have supervision / administrative control over system.

Intrusion Detection System → (IDS)

- Reads network traffic for malacious activity.
- IDS learns and is capable to detect good and bad connections.
- If any activity is found the activity is recorded and notified to authorities.
- Looks for signs and abnormal behaviour in the network.

PGP → Preety Good Privacy

- Simple to use
- 1990's
- Documentation, source code freely available.

"Confidentiality + Authentication"

Growth of PGP →

- Available freely world wide. Windows, DOS, UNIX and more.
- Based on popular and successful algorithm. RSA, SHA-1, Deffi Helman.
- Easy to use.
- Not controlled by any government
- Wide applicability.

PGP services →

- Authentication
- Confidentiality
- Compression
- Email-capability
- Segmentation