(1) Caesar cipher

encrypt "hello academy"

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

h	e	l	l	o	a	c	a	d	e	m	y
Q	H	V	R	D	F	D	G	H	P	B	

$C = (P + K) \bmod 26$ — Ciphertext

$= (13 + 3) \bmod 26$

$= 16 \bmod 26$

$= 16 \simeq Q$

decrypt "Q H V R D F D G H P B" using Caesar cipher

Plaintext

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

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

$= (16 - 3) \bmod 26$

$= 13 \bmod 26$

$= 13 \simeq N$

Q	H	V	R	D	F	D	G	H	P	B
N	e	S	O	a	C	R	d	R	m	y

① Polyalphabetic Cipher

Encryption :- $E(P_i) = (P_i + K_i) \bmod 26$

P_i is position of i^{th} letter in plaintext
K_i is position of i^{th} letter in key

Decryption :- $D(C_i) = (C_i - K_i) \bmod 26$

C_i is position of i^{th} letter in ciphertext
K_i is position of i^{th} letter in the key

Message :- HELLO Key: "KEY"

Key: KEYKE (repeated to match the length of message)

Encryption

① H (position 7) with K (position 10)
(7 + 10) mod 26 = 17 → R

② E (4) with E (4)
(4 + 4) mod 26 = 8 → I

③ L (11) with Y (24)
(11 + 24) mod 26 = 9 → J

④ L (11) with K (10)
(11 + 10) mod 26 = 21 → V

⑤ O (14) with E (4)
(14 + 4) mod 26 = 18 → S

HELLO → RIJVS

Decryption

Message :- RIJVS
Key :- KEYKE

① R (position 17) with K (position 10)
(17 - 10) mod 26 = 7 → H

② I (8) with E (4)
(8 - 4) mod 26 = 4 → E

③ J (9) with Y (24)
(9 - 24) mod 26 = 11 → L

④ V (21) with K (10)
(21 - 10) mod 26 = 11 → L

⑤ S (18) with E (4)
(18 - 4) mod 26 = 14 → O

RIJVS → HELLO

(2) Affine Cipher

Encryption

$$E(x) = (a.x + b) \bmod 26$$

x is position of letter in alphabet (starting from 0 for 'A' or 'a')

a & b are keys used in cipher

a must be coprime with 26 ie (gcd(a, 26) = 1)

Decryption

$$D(y) = a^{-1}.(y - b) \bmod 26$$

y is ~~encrypted~~ position of letter in ciphertext

a^{-1} is modular inverse of a mod 26, finding ~~no~~ a' such that

$$(a.a^{-1}) \bmod 26 = 1$$

Message "HELLO"

keys: a = 5 b = 8

Encryption

H → 7

$E = (5.7+8) \bmod 26 = 43 \bmod 26$
$= 17 \rightarrow R$

E → 4

$E = (5.4+8) \bmod 26 = 28 \bmod 26$
$= 2 \rightarrow C$

L → 11

$E = (5.11+8) \bmod 26 = 63 \bmod 26$
$= 11 \rightarrow L$

L → 11

$E = 11 \rightarrow L$

O → 14

$E = (5.14+8) \bmod 26 = 78 \bmod 26$
$= 0 \rightarrow A.$

HELLO → RCLLA

Decryption

RCLLA

a = 5 , b = 8 , we need to find modular inverse of a mod 26

$5^{-1} \bmod 26 = 21$

R → 17

$D = (21.(17-8)) \bmod 26$
$= 7 \rightarrow H$

C → 2

$D = (21.(2-8)) \bmod 26$
$= 4 \rightarrow E$

L → 11

$D = (21(11-8)) \bmod 26 = 11 \rightarrow L$

L → 11

$D = 11 \rightarrow L$

A → 0

$D = (21.(0-8)) \bmod 26 = 14 \rightarrow O$

RCLLA → HELLO

③ Playfair Cipher

Eg:- Monarchy

Construct 5×5 matrix using keyword, don't repeat letters

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

Plaintext :- attack.

Rules

① Repeating letter - Filler letter

Eg:- Plaintext :- balloon

Digrams: ba ll oo n

ba l① lo on

② Filler letter.

Eg:- plaintext :- wrape

Digrams:- wr ap e

wr ap e②

③ Same column | ↓ | wrap around

④ Same row | → | wrap round

⑤ Rectangle | ↙↗ | swap

Eg:- Plaintext : attack

at ta ck

at	ta	CK
RS	SR	DE

Plaintext :- attack

Digrams:- RS SR DE

M	O	N	④ A	R
① C	H	Y	B	D
E	F	G	I/J	② K
L	P	Q	S	③ T
U	V	W	X	Z

(14) Additive Ciphers

Similar to Ceasar Cipher where instead of fixed shift of 3 like Ceasar Ciphers

$C = (P + k) \bmod 26$ $\quad P = (C - k) \bmod 26$

Plaintext : HELLO

Key : 5

Ciphertext : M JQQT

(15) Diffie-Hellman

All should be prime number

$p = 7$, $g = 3$ } given

$a = 11$, $b = 9$ } given (private key)

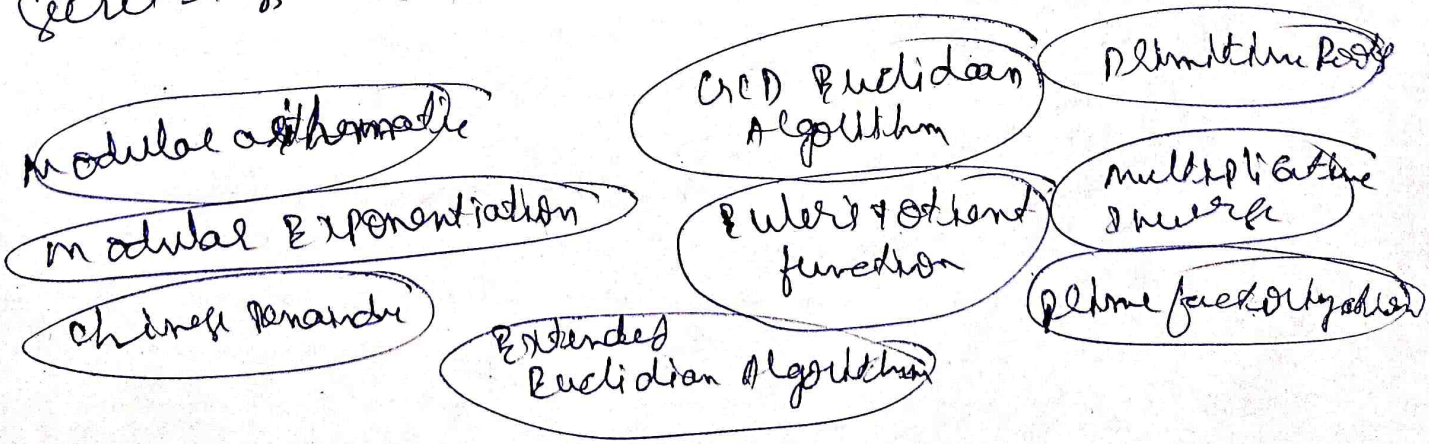
$A = g^a \bmod p \Rightarrow 3^{11} \bmod 7$ } calculating public key

$B = g^b \bmod p \Rightarrow 3^9 \bmod 7$

secret_A $= B^a \bmod p \Rightarrow B^{11} \bmod 7$ } calculating secret_key

secret_B $= A^b \bmod p \Rightarrow A^9 \bmod 7$

secret_keys must be same then the data transfer happens



5) Autokey cipher

Same as polyalphabetic substitution cipher, after the keyword is exhausted, the cipher begins using the plaintext itself as the key

Encryption $E = (P_i + K_i) \bmod 26$

Decryption $D = (C_i - K_i) \bmod 26$

message:- HELLO **key**:- "KEY"

key :- "KEYHEL" (key will be extended by adding plaintext)

Encryption

1. H (position 7) with K (position 10)
   $(7+10) \bmod 26 = 17 \rightarrow R$
2. E (4) with E (4)
   $(4+4) \bmod 26 = 8 \rightarrow I$
3. L (11) with Y (24)
   $(11+24) \bmod 26 = 9 \rightarrow J$
4. L (11) with H (7)
   $(11+7) \bmod 26 = 18 \rightarrow S$
5. O (14) with E (4)
   $(14+4) \bmod 26 = 18 \rightarrow S$

HELLO becomes RIJSS

Decryption

message:- RIJSS

key :- KEYHEL

1. R (position 17) with K (position 10)
   $(17-10) \bmod 26 = 7 \rightarrow H$
2. I (8) with E (4)
   $(8-4) \bmod 26 = 4 \rightarrow E$
3. J (9) with Y (24)
   $(9-24) \bmod 26 = 11 \rightarrow L$
4. S (18) with H (7)
   $(18-7) \bmod 26 = 11 \rightarrow L$
5. S (18) with E (4)
   $(18-4) \bmod 26 = 14 \rightarrow O$

RIJSS becomes HELLO

⑧ Hill Cipher

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

$$P = D(K, C) = C K^{-1} \bmod 26$$

$$= P \times K \times K^{-1} \bmod 26$$

Encrypt "pay more money" using Hill Cipher with key $\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$

Q) Encrypt "pay more money" using Hill Cipher with key

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

Soln

p	a	y	m	o	r	e	m	o	n	e	y
15	0	24	12	14	17	4	12	14	13	4	24

key = 3×3 matrix

PT = pay mor emo ney

① Encrypting :- pay

$$(C_1 C_2 C_3) = (P_1 P_2 P_3) \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix} \bmod 26$$

$$= (15\ 0\ 24) \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \bmod 26$$

$$= (15\times17+0\times21+24\times2 \quad 15\times17+0\times18+24\times2 \quad 15\times5+0\times21+24\times19) \bmod 26$$

$$= (303\ 303\ 531) \bmod 26$$

$$= (17\ 17\ 11)$$

$$= (R\ R\ L) \leftarrow \text{ciphertext}$$

② Encrypting :- mor

$$(C_1\ C_2\ C_3) = (12\ 14\ 17)\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \bmod 26$$

$$(12\times17+14\times21+17\times2 \quad 12\times17+14\times18+17\times2 \quad 12\times5+14\times21+17\times19) \bmod 26$$

$$= (532\ \ 490\ \ 677) \bmod 26$$

$$= (12\ \ 22\ \ 1)$$

$$= (M\ W\ B)$$

③ Encrypting :- end

$$(C_1\ C_2\ C_3) = (4\ \ 12\ \ 14)\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \bmod 26$$

$$= (348\ \ 312\ \ 258) \bmod 26$$

$$= (10\ \ 0\ \ 18)$$

$$= (K\ A\ S)$$

④ Encrypting :- my

$$(C_1\ C_2\ C_3) = (13\ \ 4\ \ 24)\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \bmod 26$$

$$= (15\ \ 3\ \ 7)$$

$$= (P\ D\ H)$$

PT	p	a	y	m	o	r	e	m	o	n	e	y
CT	R	R	L	M	W	B	K	A	S	P	D	H

Decryption:

$P = D(K, C) = C \times K^{-1} \bmod 26$

Decryption requires K^{-1}, the inverse matrix K

$$K^{-1} = \frac{1}{\det K} \times \text{Adj } K$$

$$K = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

Step1

$$\det \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \bmod 26$$

$= -9439 \bmod 26$

$= -3 \bmod 26$ (when we get -ve value add 26 both)

$= 23$

Step2

$K^{-1} = \frac{1}{\det K} \times \text{Adjoint } K$

$\text{Adj}(K) = K^{-1} \det(K)$

$$= \begin{bmatrix} +300 & -313 & 267 \\ -357 & 313 & -252 \\ 6 & 0 & -51 \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 14 & -1 & 7 \\ -19 & 1 & -18 \\ 6 & 0 & -25 \end{bmatrix} \bmod 26$$

now divide matrix by 26 and Put remindy

$$= \begin{bmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{bmatrix} \bmod 26$$

convert -ve to +ve add 26

step 3

$$k^{-1} = \frac{1}{23} \times \begin{pmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{pmatrix} \bmod 26$$

$$= (23^{-1}) \begin{pmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{pmatrix} \bmod 26$$

multiplicative inverse

$$= 17 \times \begin{pmatrix} 14 & 25 & 7 \\ 7 & 1 & 8 \\ 6 & 0 & 1 \end{pmatrix} \bmod 26$$

$$= \begin{pmatrix} 238 & 425 & 119 \\ 119 & 17 & 136 \\ 102 & 0 & 17 \end{pmatrix} \bmod 26$$

$$= \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

Decrypt "RRLMWBKASPDH" using HW Cipher with key

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

Soln

$P = C K^{-1} \bmod 26$

R	R	L	M	W	B	K	A	S	P	D	H
17	17	11	12	22	1	10	0	18	15	3	7

Decrypting RRL

$$(P_1 P_2 P_3) = (RRL)\begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$$= (17\ 17\ 11)\begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$$= (587 \quad 442 \quad \text{[illegible]}) \bmod 26$$

$$= (15 \quad 0 \quad 24)$$

$$= (P \quad A \quad Y)$$

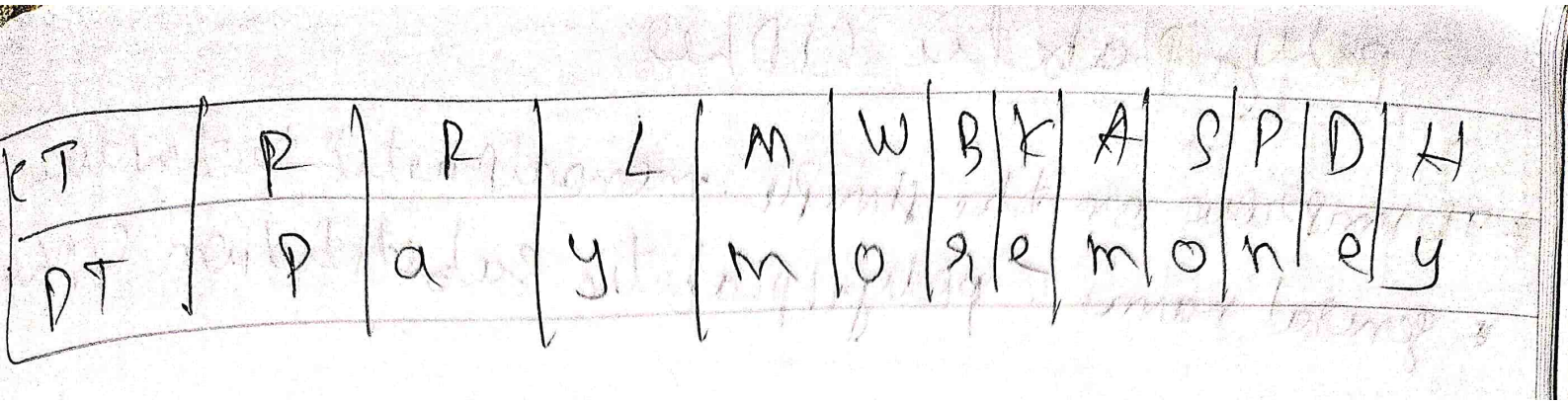
Similarly

$$(P_1 P_2 P_3) = (MWB)\begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$$= (12 \quad 22 \quad 1)\begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} \bmod 26$$

$$= (12\ 14\ 17)$$

$$= (M\ O\ R)$$


ET
DT
P
P
R
a
L
y
M
m
W
o
B
r
K
e
A
m
S
o
P
n
D
e
H
y

RSA

Algorithm

(1) Choose two prime numbers P & Q

(2) Compute n as $n = P \times Q$ this n is part of the public & private key

(3) Compute $\phi(n)$ (Euler's totient function)

$\phi(n) = (P-1) \times (Q-1)$

(4) Choose a public exponent e such that $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$. This e is part of public key

(5) Find private exponent d such that $(d \times e) \bmod \phi(n) = 1$
This is done by finding the modular inverse of e mod $\phi(n)$
d is part of private key

(6) Encrypt a message m as $c = m^e \bmod n$, where c is ciphertext

(7) Decrypt the ciphertext c as $m = c^d \bmod n$, where m is original message

Manual

(1) Choose primes $P = 7$ & $Q = 11$

$n = P \times Q = 7 \times 11 = 77$

$\phi(n) = (P-1) \times (Q-1) = 6 \times 10 = 60$

(2) Find public exponent e:

we need $1 < e < 60$ and $\gcd(e, 60) = 1$

Lets try $e = 7$ (it satisfied $\gcd(7, 60) = 1$)

(3) Find private exponent d:

we need $(d \times e) \bmod 60 = 1$ this means value for d in $(d \times 7) \bmod 60$.

.. using trial & error

(4) Encryption

- Let message $m = 13$

encrypt, compute $c = m^e \bmod n$

$c = 13^7 \bmod 77 = 17$

encrypted message c = 17

(5) Decryption

to decrypt, compute $m = c^d \bmod n$

$m = 17^{43} \bmod 77$

- using modular exponentiation,

$m = 13$ (original message)

(8) Digital Signature Standard

(1) Key generation :- generate a private key & its corresponding public key

(2) message signing :- sign the message using the private key, after encoding it into bytes

(3) Signature verification :- verify the signed message using the public key, ensuring signature is correct

SHA-512

(1) Appending Padding bits
(range of 1 to 1024)
single 1 bit & followed by necessary no of 0 bits

(2) Append length
(128 bits is appended to message)
multiple of 1024 bits
N × 1024

(3) Initialize hash buffer
(Buffers = 64 bit registers named as a, b, c, d, e, f, g & h

(4) Process the message in 1024 bit (128 word) blocks
(Round functions)

(5) Output

(9) DES

Steps:

(1) Feistel Round Function (feistel_round):
- Split the 8-bit block into two halves left & right
- XOR the left half with the key (Feistel function)
- XOR the result with the right half to produce a new right half
- Swap the halves for next round.

(2) Encryption function (simplified_des encrypt)
- Ensure the plaintext length is multiple of 8 bits by padding with zeros if needed
- Divide the plaintext into 8-bit blocks
- For each block, repeat the Feistel rounds for a specified number of iterations (rounds)
- Combine the blocks to form the ciphertext

(3) main function (main)
- get input for the plaintext, key & number of rounds from the user
- call the encryption function to encrypt the plaintext using the given key & rounds
- Display the original plaintext & the encrypted ciphertext

Left Half i-1 → 32 → Left Half i
Right Half i-1 → 32 → Right Half i
Right Half i-1 → Expansion permutation → 48 → ⊕ Round Key → Key mixed Substitution (8 S-Boxes) → 48 → Transposition (P-Box) → 32 → ⊕ Left Half i-1 → 32