Cryptography.

Principles of Security:

* Confidentiality
* Authenticity
* Integrity

To overcome CIA there should be no,

* Interception
* Modification
* Fabrication

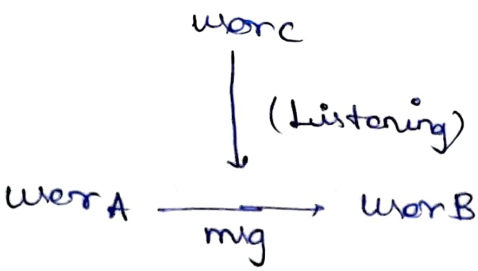
* Availability
* Access control
* Attacks
  - → Criminal
  - → Publicity
  - → Legal

Attacks :

→ Passive

→ Active

eg : DOS



(Passive attacks)

29/08/22

Vulnerability - Some problem in program explaiting.

Fuzzer - try different combination to find whether Vulnerable or not

Registers :

* General purpose registers
  - → EAX Accumulator
  - → EBX Base / offset
  - → ECX Counter
  - → EDX Data

* Segment
  - → CS Counter Segment
  - → DS Data Segment
  - → SS Stack Segment.

* Control register
  - → EIP Next Instruction

Every program will have,

* text → read only
* data, BSS → Global / Static variables.
* Heap → uninitialized.
  → FIFO.
* Stack → LIFO.

Dynamic Registor allocation → we use heap or stack.

To define word in Assembly : dw.

eg:

```
int number;
...
code
...
number++;
```

⇒

```
number dw 0;
...
...
mov eax, number;
inc eax;
move number, eax;
```

```
int number;
if (number < 0) {
...
}
```

⇒

```
number dw 0;
mov eax, number;
or, eax, eax;
jge label
<no code>

label
<yes code>
```

Debugging.

b main → Stops after main

b → Stops in current line

b N → Stops after N lines.

r → run till the first occurance of break points

s → same as next but if it is a block, it moves to next block.

next → move to next line

bt

info b → list all break points.

Buffer:

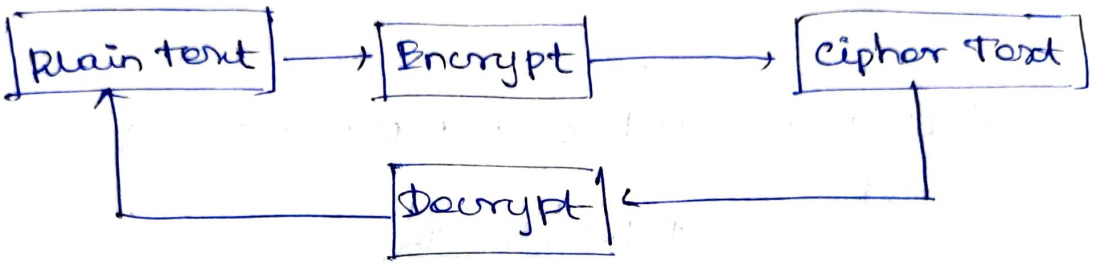
* Temporary Storage
* fast processing

Cryptology:

* Study of Cryptography.
* Plain text: human readable
* Cipher text: unreadable form

Encryption : Technique to convert plain text to cipher text

Decryption : to convert cipher text to plain text.



Substitution

Ceasor Cipher :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

P	Q	R	S	T	U	V	W	X	Y	Z
15	16	17	18	19	20	21	22	23	24	25

$$C = (P, K)$$

$$\text{Cipher text} = (P + K) \bmod 26.$$

$$P = (C, K)$$

$$\text{Plain text} = (C - K) \bmod 26$$

eg:

P = Meet Me.

Cipher = PHHW PH.

C = Nishant is a friend of Antony

Cipher text = QLVKDQW LV D IULHQG RI

DQWRQB

Cipher text = GBEXXEQQEXSRGI

DBUUBNLBU

EE

F

Monoalphabetic cipher:

* Each character is mapped to some random alphabet.

eg: P.T = CIPHER

Cipher = LVZSJT.

* Total combinations = 26!

* Can able be decipher it in 26! ways of combination.

Playfair ciphor:

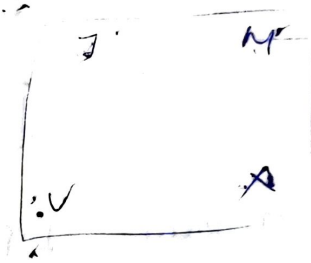
Key : APPLE

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

Rules:

* Same row →
* Same column ↓
* Rectangle / Square Swap

P.T	A	T	T	A	C	K
C.T	E	Q	Q	E	D	I



P.T =	E	N	C	R	Y	P	T	I	O	N
C.T =	G	T	D	Q	W	P	Q	N	I	O

Hill ciphor: (1929).

* Using mathematical operation

$$C.T = P.T \times K \bmod 26$$

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

Key $\begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}$ PT = HELP.

Taking HE :

$= \begin{bmatrix} 7 & 4 \end{bmatrix} \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix} \bmod 26$

$= \begin{bmatrix} 29 & 41 \end{bmatrix} \bmod 26$

$= \begin{bmatrix} 3 & 15 \end{bmatrix}$

$\begin{bmatrix} D & P \end{bmatrix}$

Taking LP :

$= \begin{bmatrix} 11 & 15 \end{bmatrix} \begin{bmatrix} 3 & 3 \\ 2 & 5 \end{bmatrix}$

$= \begin{bmatrix} 63 & 108 \end{bmatrix} \bmod 26$

$= \begin{bmatrix} 11 & 4 \end{bmatrix}$

$= \begin{bmatrix} L & E \end{bmatrix}$

$\therefore$ C.T = DPLE

Decryption :

Find K^{-1}.

$$\therefore K^{-1} = \frac{1}{9}\begin{bmatrix} 5 & -3 \\ -2 & 3 \end{bmatrix}$$

$9x \mod 26 = 1.$

C.T = DPLE

Taking DP :

$$= 3\begin{bmatrix} 3 & 15 \end{bmatrix}\begin{bmatrix} 5 & -3 \\ -2 & 3 \end{bmatrix} \mod 26.$$

$$= 3\begin{bmatrix} -15 & 66 \end{bmatrix} \mod 26$$

$$= \begin{bmatrix} -75 & 198 \end{bmatrix} \mod 26.$$

$$= \begin{bmatrix} -49 & 198 \end{bmatrix} \mod 26 \quad (+26 \text{ to make +ve})$$

=

Polyalphabetic cipher: Vigenere

$$C_i = (P_i + K_i) \bmod 26.$$

eg:

Key = Deceptive

PT = Pay once more for the class.

If P.T is larger than key size, repeat key until it becomes equal to PT.

Encryption:

= (3 + 15) mod 26 = 18 = S

= (4 + 0) mod 26 = 4 = E

= (2 + 24) mod 26 = 0 = A.

⋮

Instead of repeating key characters we can also repeat the characters of P.T also.

Vernam cipher:

$$c_i = (p_i \oplus k_i)$$

One Time Pad:

Same as polyalphabetic cipher but here we use a key only once.

Transposition:

Rail Fence cipher:

Depth => 2

P.T = Attack at once.

A		T		C		A		O		C	
	T		A		K		T		N		E

C.T = read characters row wise.

∴ C.T = ATCAOCTAKTNE

Depth = 3.

A				C				O			
	T		A		K		T		N		E
		T				A				C	

C.T = ACOTAKTNETAC.

Row column Transposition:

* Writing in row wise
* reading column wise using priority
* remaing cells can be filled with any combinations of alphabet.

PT = ATTACK AT ONCE

4	3	1	2
A	T	T	A
C	K	A	T
O	N	C	E

CT = TAC ATE TKN ACO.

Decryption:

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

1) RS|SR|DE

ATTACK.

ii) ON|TS|ML

MOSQUE.

Key = Caesar.

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

i) NT|MQ|KD|QF

MULTIPLE

ii) RL|AC|WT|OM|UI

ENCRYPTION.

10/9/22

Sample Space - S :

* Set of all possible events.

eg: coin toss S = {H, T}.

Event - E :

* Occurance of an action.

* Types :

→ Favorable event

→ non-Favorable event.

M - number of times the favorable event occurs.

$\frac{m}{n}$ → event
n → Sample Space.

Addition:

$P(A \cup B) = P(A) + P(B) - P(A \cap B)$.
↳ for dependent events

$P(A \cup B) = P(A) + P(B)$
↳ for independent events.

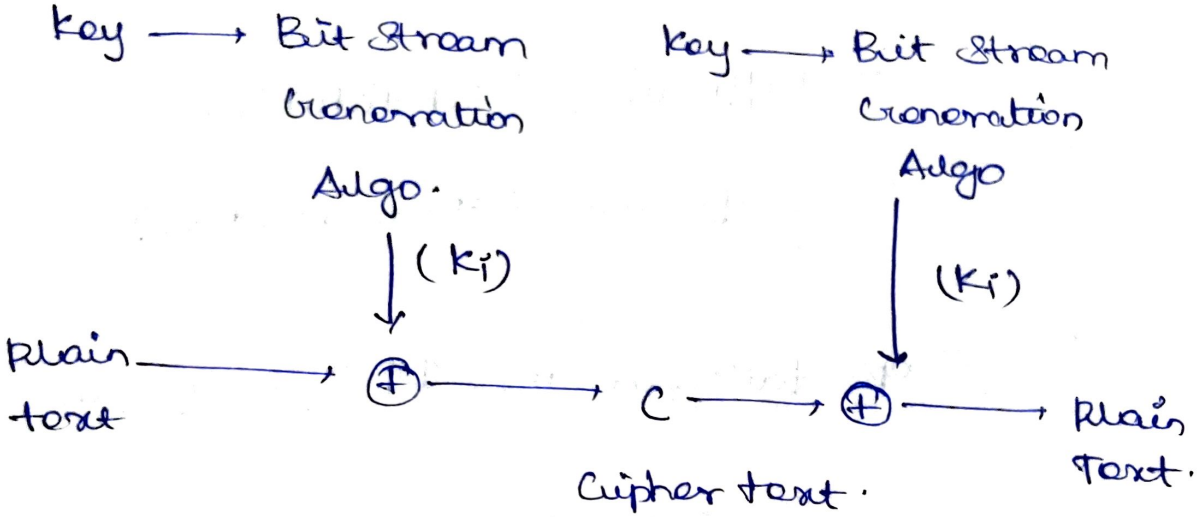
Binomial distribution = $\binom{n}{k} p^k (1-p)^{n-k}$.

DES: Algorithm: (Data Encryption Standards)

* based on Symmetric cipher.
* Stream cipher:
  → change P.T to C.T character by character
* Block cipher:
  → change PT to CT block by block

→ minimum 64 bit / 128 bit.

* DES → Feistel block structure.



* mapping:

→ reversible (eg. ceasar ciper)

→ irreversable.

Eg for irreversable:

P.T	C.T
00	10
01	11
10	00
11	11.

here 01 and 11 both are mapped to 11. So it requires 2^n combination to identify correct P.T.

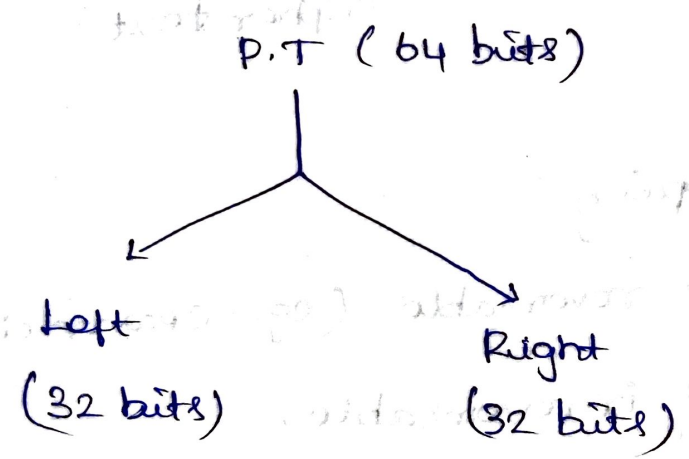
* Feistel block structure → Idle block structure.

* Concepts used in Feistel:

  → Diffusion. (Substitution)

  → Confusion. (Transposition)

* Steps followed in DES encryption



→ every 8th bit is removed from key

→ initially key is 64 bit after removing parity it becomes 56 bit.

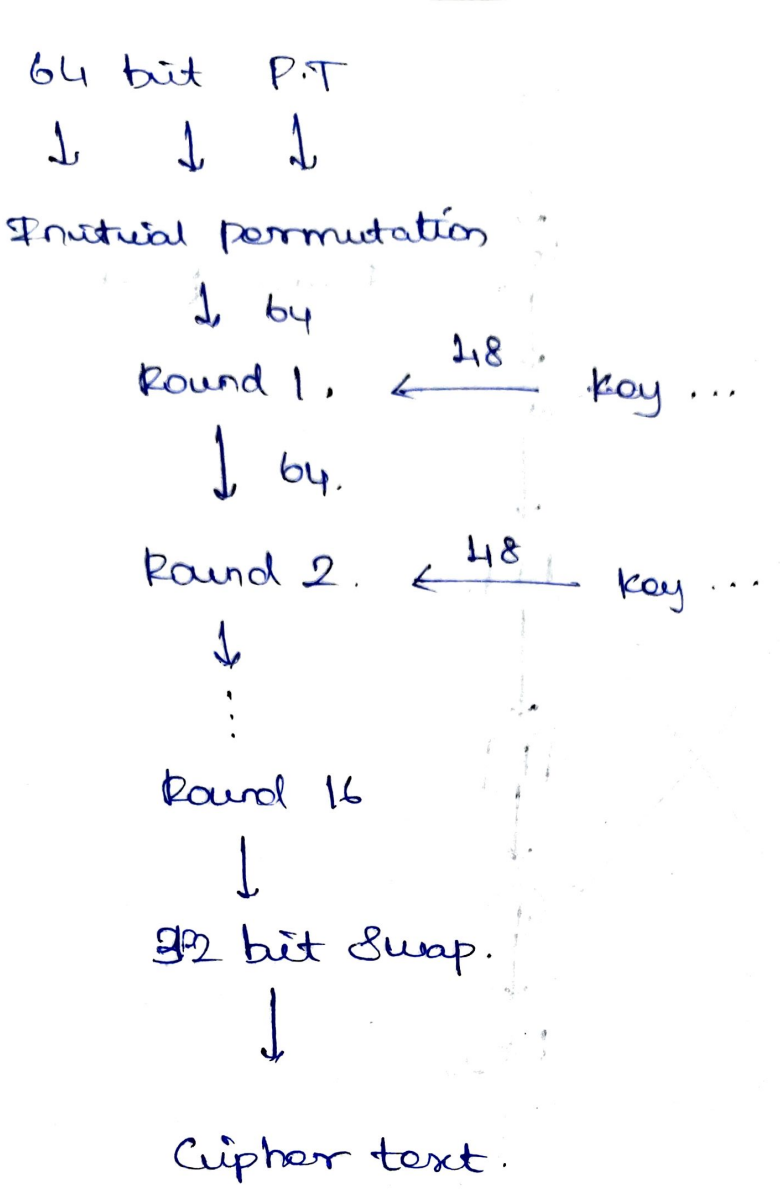
eg:

1 2 3 4 5 6 7 ~~8~~

. . . . . ~~16~~

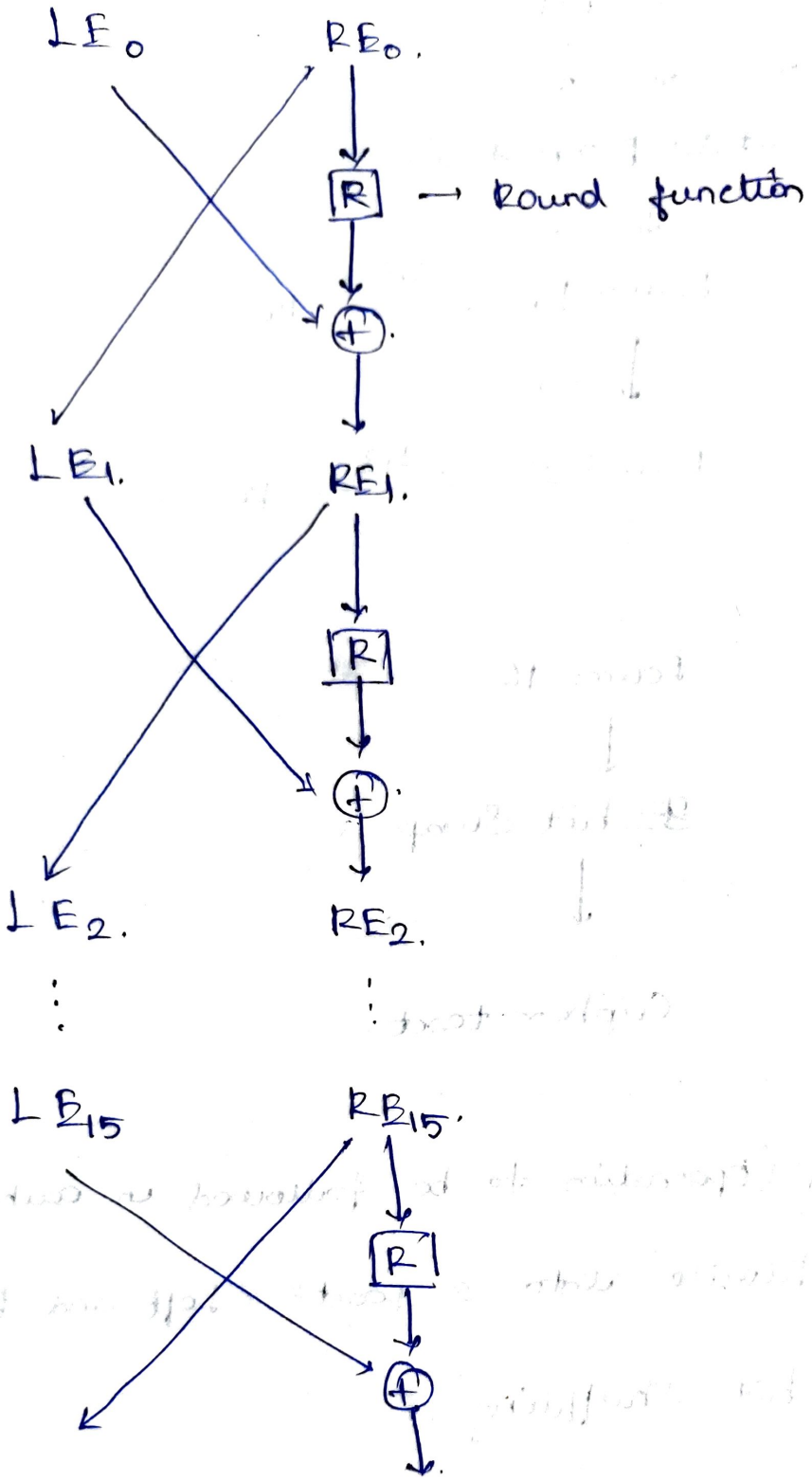
. 24

(remove 8, 16, 24, ...)



Round (Operation to be followed in each Round)

* Divide into 2 parts Left and Right
* bit Shuffling.
* Substitution (any Substitution can be applied)
* XOR key with input.



After completing 15 rounds Swap left and right part and apply Inverse Permutation.

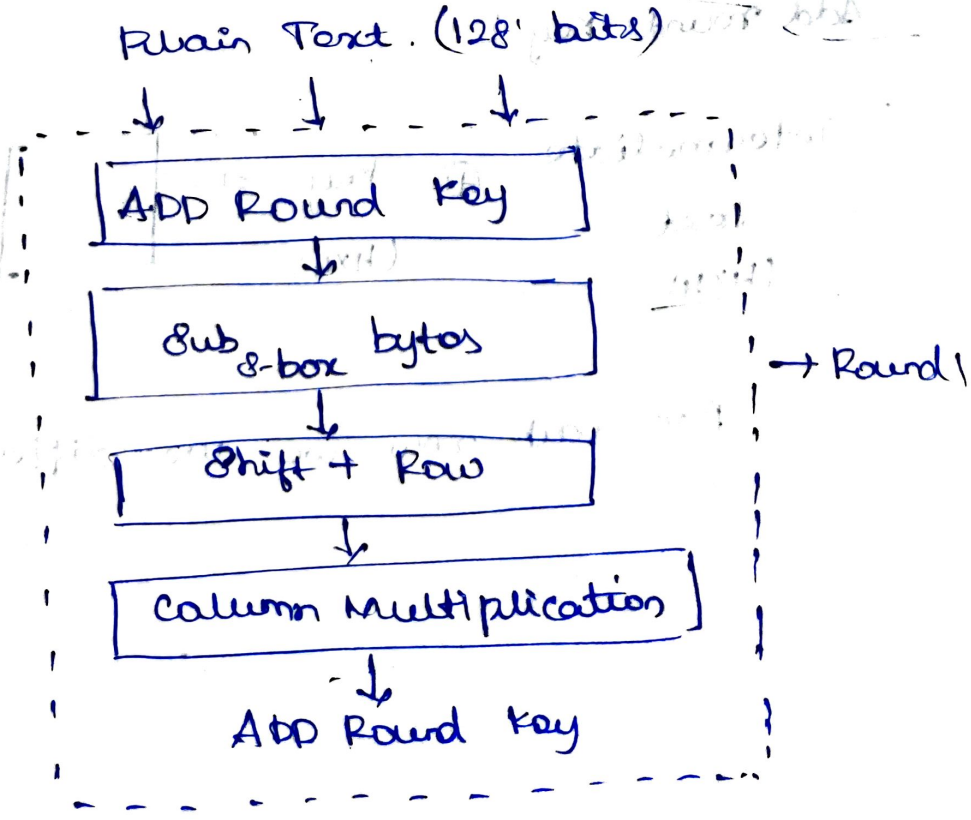
HW/: See problem. (any 1).

* Number of round is important
* Operation performed in round function is also very important.
* Initially DES is impossible to crack but later it become possible so we move to AES.

12/09/22

AES: (Advanced Encryption Standards)

* Initially proposed by National Institute of Standards.



* Repeat the same process for 9 rounds.

* For 10th round (last) alone column multiplication is not used.

State array : → Intermediate array

$$S = \begin{bmatrix} s_{0,0} & s_{0,1} & \cdots \\ s_{1,0} & s_{1,1} & \cdots \\ s_{2,0} & s_{2,1} & \cdots \\ s_{3,0} & s_{3,1} & \cdots & s_{3,3} \end{bmatrix}$$

$$O/p = \begin{bmatrix} o_1 & \cdots \\ o_2 & \cdots \\ o_3 & \cdots \\ \vdots & \cdots & o_{15} \end{bmatrix}$$

Add round key :

Intermediate text (4×4) ⊕ key (4×4) = [ ] → round.

For each row we use different key.

Root word:

X_1 69 6e 67 72

S-Box:

Y_1 F9 9f 85 40

Round const:

$[P_1 \longrightarrow R_{10}]$.

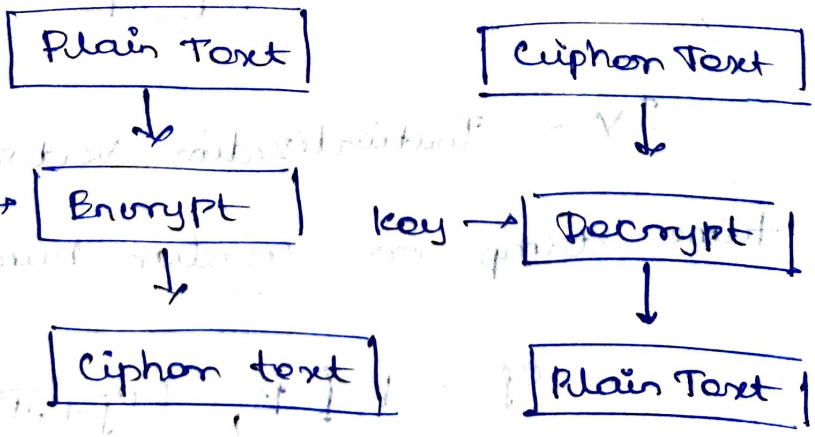
$\begin{bmatrix} 01 \\ 00 \\ 00 \\ 00 \end{bmatrix} \begin{bmatrix} 02 \\ 00 \\ 00 \\ 00 \end{bmatrix} \cdots \cdots \begin{bmatrix} 36 \\ 00 \\ 00 \\ 00 \end{bmatrix}$ ⇒ $g(w_3) \oplus$

$\begin{bmatrix} f9 & 01 \\ 9f & \oplus\ 00 \\ 85 & 00 \\ 40 & 00 \end{bmatrix}$

14/09/22.

Block cipher Mode:

* Electronic code book.



$C_i = E(K, P_i)$

$P_i = E(K, C_i)$

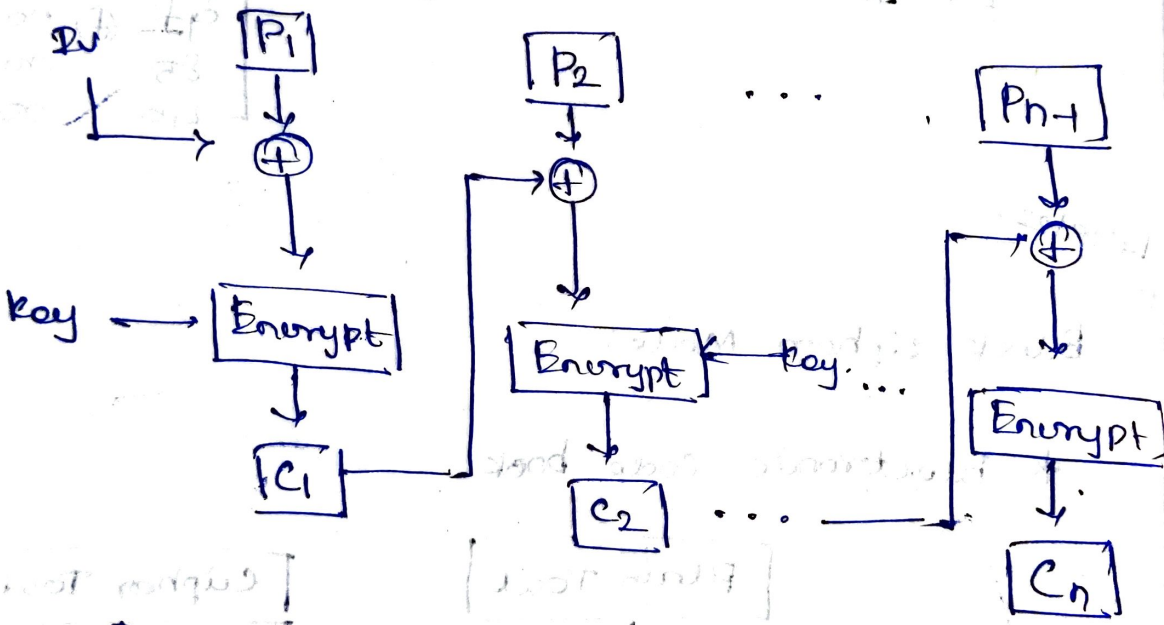
→ Used for small size text.

→ If there is repetition of words it is easier to crack.

* Other constraints to be maintained

1. Overhead
2. Error Recovery.
3. Error Propagation.
4. Diffusion.
5. Security.

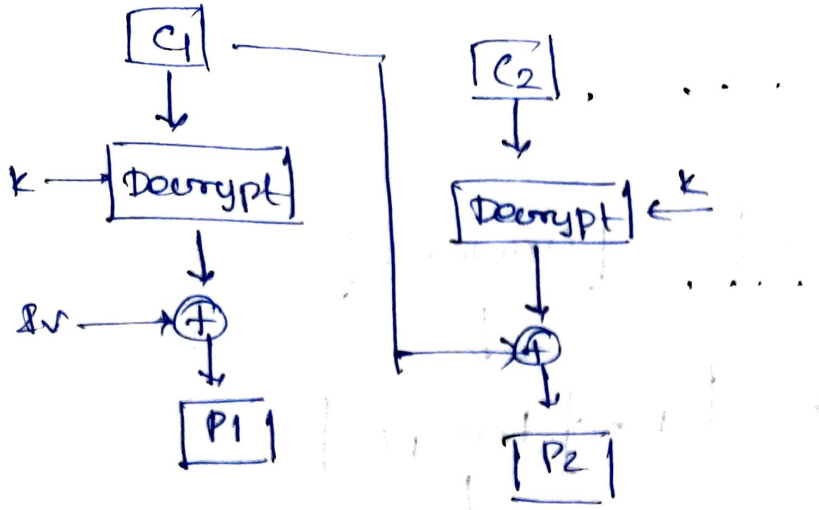
2. Cipher block chaining mode:



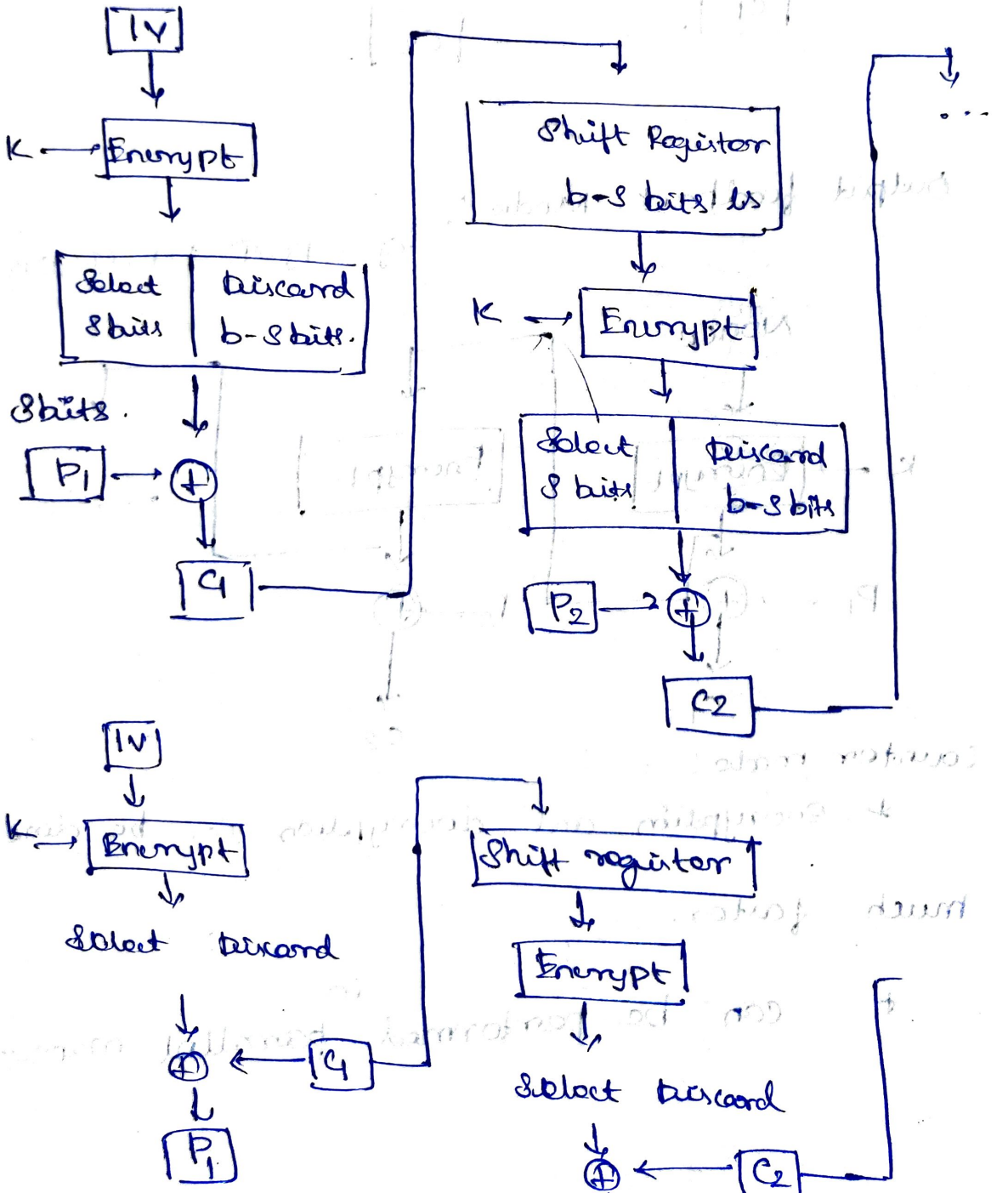
IV - Initialization vector. Can be time stamp or random number generated.

$$C_j = E[K, (C_{j-1} \oplus P_j)]$$

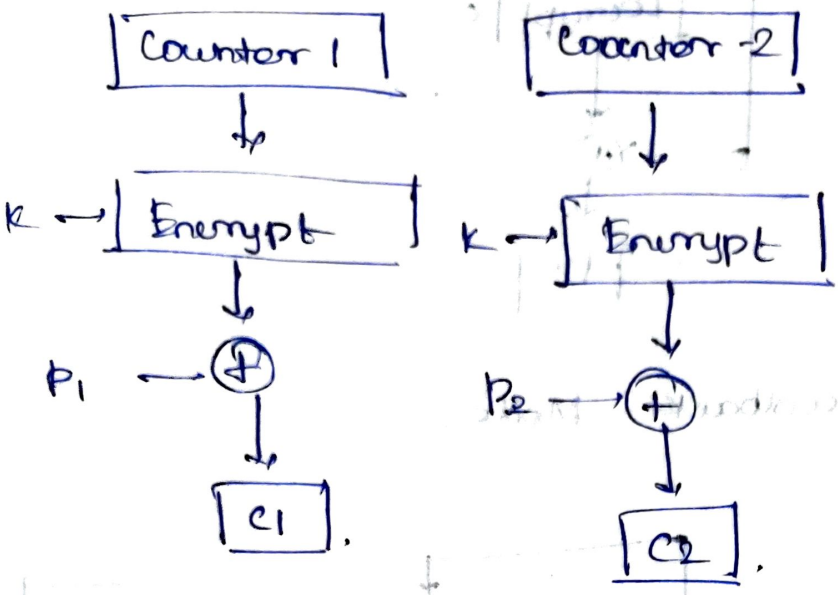
$$D = D[K, E(K, C_j \oplus P_j)]$$



3. Cipher Feedback Mode:

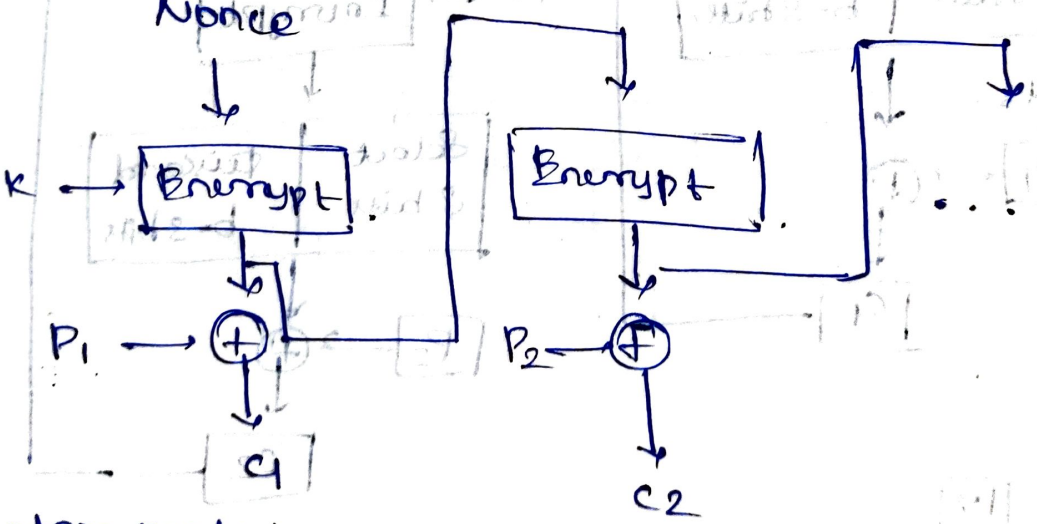


Counter Mode:



Output feedback - Mode:

$C_j = P_j \oplus E(K, C_{j-1} \oplus P_{j-1})$



Counter mode:

* encryption and decryption can be done much faster.
* can be performed in parallel manner.

Padding:

Simple - 10^*

Multirate - $10^* \ldots 1$.

20/09/2022.

Divisibility:

* A nonzero b divides a if $a = mb$ for some m, where a, b and m are integers.

* b and divides a if there is no remainder on division.

* notation $b|a$ is commonly used to mean b divides a.

Prime numbers:

Numbers that are divisible by only 1 and by the number itself.

Relative prime numbers:

a and b are relative prime numbers if $\gcd(a,b) = 1$.

eg: 3 and 4.

Congruent mod:

$a \bmod n = b \bmod n$.

Eg: $73 \bmod 23 = 4 \bmod 23$.

Properties of divisibility:

* If $a \mid 1$ then $a = \pm 1$
* If $a \mid b$ and $b \mid a$ then $a = \pm b$.
* Any $b \neq 0$ divides 0.
* If $a \mid b$ and $b \mid c$ then $a \mid c$.
* If $b \mid g$ and $b \mid h$ then $b \mid (mg + nh)$ for arbitrary integers m and n.

Division Algorithm:

Given any positive integer n and any nonnegative integer a, if we divide a by n

GCD:

GCD of a and b is the largest integer that divides both a and b.

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

i) gcd(72345, 43215).

ii) gcd(3486, 10292).

$72345 = 1 \times 43215 + 29130$

$43215 = 1 \times 29130 + 14085$

$29130 = 2 \times 14085 - 960$

$14085 = 14 \times 960 + 645$

$960 = 1 \times 645 + 315$

$645 = 2 \times 315 + 15$

$315 = 21 \times 15$

gcd(72345, 43215) = 15

iii) gcd(75, 32)

iv) gcd(117, 218).

$75 = 2 \times 32 + 11$

$32 = 2 \times 11 + 10$

$11 = 1 \times 10 + 1.$

$10 = 10 \times 1.$

gcd(75, 32) = 1

$218 = 1 \times 117 + 101.$

$117 = 1 \times 101 + 16$

$101 = 6 \times 16 + 5$

$16 = 3 \times 5 + 1.$

$5 = 5 \times 1.$

gcd(117, 218) = 1.

Modular Arithmetic:

* Congruent modulo n:

Two integers a and b are said to be Congruent modulo n if $(a \bmod n) = (b \bmod n)$

* Modulus:

If a is an integer and n is a positive integer we define a mod n to be the remainder when a is divided by n the integer is called modulus.

Properties:

1. $a = b \pmod n$ if $n(a-b)$
2. $a \equiv b \pmod n$ implies $b = a \pmod n$
3. $a = b \pmod n$ and $b = c \pmod n$ imply $a = c \pmod n$.

Let $a = 23,\ b = 8,\ n = 5$.

$$23 \equiv 8 \pmod 5$$

$$23 - 8 \equiv 15 = 5 \times 3.$$

$$\Rightarrow -11 \equiv 5 \pmod 8$$

$$-11 - 5 = -16 = -2 \times 8.$$

$[(a \bmod n) + (b \bmod n)] \bmod n = (a+b) \bmod n.$

$[(a \bmod n) - (b \bmod n)] \bmod n = (a-b) \bmod n$

$[(a \bmod n) * (b \bmod n)] \bmod n = (a*b) \bmod n.$

Commutative laws

Associative laws

Distributive law

Identities

Additive Inverse law

Fermat's Theorem:

If p is prime and a is a positive integer not divisible by p then.

$$a^{p-1} = 1 \pmod{p}.$$

Alternative form is:

If p is a prime and a is a positive integer then

$$a^p = a \pmod{p}.$$

$5^{19-1} = 1 \pmod{19} = 1$

$5^{19} = 5 \pmod{19} = 5$

$9^{794} \bmod 73$

$= 9^{(73 \times 10 + 64)} \bmod 73$

$= \left[(9^{73})^{10} \cdot 9^{64} \right] \bmod 73.$

$= \left[(9)^{10} \cdot 9^{64} \right] \bmod 73.$

$= 9^{74} \bmod 73 = (9^{73} \bmod 73)(9 \bmod 73)$

$= 9 \bmod 73 \times 9 \bmod 73$

$= 9 \times 9 \bmod 73 = 81 \bmod 73 = 8.$

$x^{86} = 6 \bmod 29.$

$x^{72} \cdot x^{14} = 6 \bmod 29.$

$(1 \bmod 73)\, x^{14} = 6 \bmod 29.$

$x^{14} = 6 \bmod 29.$

$x^{12} \cdot x^{2} = 6 \bmod 29$

$(1 \bmod 14)\, x^2 = 6 \bmod 29.$

$x^2 \bmod 29 = 6$

6, 35, 64, 93

$\therefore \quad x = 8.$

(iii) $4^{225} \mod 13.$

$= ((4^{17})^{13} \cdot 4^4) \mod 13$

$= (4^{17} \mod 13 \cdot 4^4) \mod 13.$

$= ((4^{13+4} \cdot \mod 13) \cdot 16) \mod 13.$

$= (4^{13} \cdot \mod 13 \cdot 4^4 \mod 13) \cdot (16) \mod 13.$

$= (4 \ (3)(16).) \mod 13.$

$= 102$

24/09/2022

Overflow Buffer:

array [30];

NOP - no operation;

```
fun() {
    gets()
    Printf()
}
main() {
    func()
}
```

26/09/2022.

Shell code (user):

* getting input
* accessing binary files
* for direct kernal access.

```
int main() {
    exit(0);
}
```

For Sys call : 0x80.

Call the function. It will be thrown into accumulator along with parameters and then execute Sys call. using 0x80.

```
int $0x80
mov 0x1
int $0x80.
```

store 0 into ebx

store 1 into eax

execute 0x80 syscall (not group exit).

section .text

global _start

_start

1. mov ebx storage bb 00 00 00 00
2. mov eax storage b8 01 00 00 00
3. int 0x80 storage cd. 80 . . . .

char shellcode [] = {"\xbb\x00\x00\x00\x00
\xb8\x01\x80};

1. xor ebx, ebx.
2. mov cd, 1.
3. int 0x80

* fork() + execve() → copy the program.
* execve() → replace with another program.
* execve() three arguments: function, argb and drgb endp.
* 4 registers used: 1 → sys call 3 → argument

* relative addressing used.

* placeholders.

→ Clibrary wrapper like

→ Sys call is loaded into ebx

→ placing arguments into register.

→ CPU is changed to kernel mode.

→ Sys call function is executed.

→ group exit is not needed, (optimization)

write desired code in high level language.

10/10/2022:

Format Strings:

→ modifying output.

Used in:

* Output Statements.

Different Format Strings

%d	Integer value.	%s - String.
%u	unsigned value	
%x	hexadecimal	
%n	number of bytes	

```
#include <stdio.h>
int main() {
    A = 5, B = 10, count_a, count_b;
    Pf ("No. of bytes X %n till now and no.
        of bytes X %n till pt", count_a,
        count_b);
    Pf ("%d", count_a);
    Pf ("%d", count_b);
    Pf ("Adds %0x80 for %d, %d is value of
        B, &A, A, B);
}
```

Format String Vulnerability:

%80 - 8 bytes in pack.

Reading from arbitary memory address:

Pass initial 4 bytes

Direct parameter parsing:

```
Pf("#: %#$d, 4: %.4$05d", 10, 20, 30, 40, 50
                          60, 70, 80, 90);
```

Output:

#: 70, 4: 00040.

EL:

Detours using : d tors

Short writes

Overwriting offset table modes.

node search.

12/10/2022

Euler's theorem: $a^{\phi(n)} \bmod n = 1$.

$\phi(n) = n-1$ if n is a prime number.

$\phi(n) = (P-1)(q-1)$ if $n = Pq$ where p and q are prime numbers.

$\phi(n) = n\left(1-\frac{1}{P_1}\right)\left(1-\frac{1}{P_2}\right)\cdots\left(1-\frac{1}{P_K}\right)$ if $n = ab$ where a and b or either one of them is composite

$\phi(7000) =$

$7000 = 7 \times 1000$

$= 7^1 \times 2^3 \times 5^3$

$\therefore \phi(7000) = 7000\left(1-\frac{1}{7}\right)\left(1-\frac{1}{2}\right)\left(1-\frac{1}{5}\right)$

$= 7 \times 4000\left(1-\frac{1}{\quad}\right)$

Using Euler theorem Solve:

$4^{99} \bmod 35$

$\phi(35) = 24.$

$= 4^{96} \cdot 4^{3} \bmod 35$

$= 35\left(1-\frac{1}{5}\right)\left(1-\frac{1}{7}\right)$

$= 35\left(\frac{4}{5}\right)\left(\frac{6}{7}\right)$

$= 24.$

$= (4^{24})^{4} \cdot 4^{3} \bmod 35.$

$= (1 \bmod 35)^{4} \cdot 4^{3} \bmod 35$

$= 1 \cdot 64 \bmod 35$

$= 29.$

$3^{202} \bmod 13.$

$\phi(13) = 12.$

$= (3^{12})^{16} \cdot 3^{10} \cdot \bmod 13$

$= (1 \bmod 13)^{16} \cdot 3^{10} \cdot \bmod 13$

$= 3^{10} \bmod 13.$

$= 3^{3} \cdot 3^{3} \cdot 3^{3} \cdot 3^{1} \bmod 13.$

$= 3 \bmod 13$

$= 3.$

Chinese remainder theorem:

Step 1: Identify or calculate m_1, m_2, m_3

Step 2: $M = m_1 \times m_2 \times m_3 \times \ldots m_i$

Step 3: $M_i = \frac{M}{m_i}$

Step 4: Calculate $a_1, a_2 \ldots a_i$

Step 5: Calculate M_i^{-1}

Step 6: $X = \sum_i a_i m_i M_i^{-1}$

$X = a_i \bmod m_i$

$X = 2 \bmod 3$

$X = 3 \bmod 5$ Using CRT.

$X = 2 \bmod 7$.

It is used when $m_i > n$ ($a_i^n \bmod m_i$)

Step 1: $m_1 = 3 \quad m_2 = 5 \quad m_3 = 7$

Step 2: $M = 3 \times 5 \times 7 = 105$

Step 3: $M_1 = \frac{105}{3} = 35 \quad M_2 = \frac{105}{5} = 21$

$M_3 = \frac{105}{7} = 15$

Step 4: $a_1 = 2$, $a_2 = 3$, $a_3 = 2$

Step 5: M_i^{-1}

$M_i M_i^{-1} = 1 \bmod m_i$.

$35 \times 1 \bmod 3 = 2$

$35 \times 2 \bmod 3 = 1$

$M_1^{-1} = 2$

$21 \times 1 \bmod 5 = 1$

$M_2^{-1} = 1$

$15 \times 1 \bmod 7 = 1$

$M_3^{-1} = 1$.

Step 6:

$X = 2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1$.

$= 142 + 153 + 14$.

$X = 233 \bmod 115 = 23$.

Hw:

1) $4X = 5 \bmod 9$

$2X = 6 \bmod 20$.

2) $X = 5 \bmod 3$

$X = 2 \bmod 5$

$X = 1 \bmod 11$.

Miller Rabin Primality test :

$n-1 = 2^k * m$

$a \quad 1 \leq a \leq (n-1).$

$b_0 = a^m \bmod n.$

$b_1 = b_0^2 \bmod n$

$b_2 = b_1^2 \bmod n.$

i) 561 a P. number

i.) $561 = (n+1) \Rightarrow 561-1 = 560 = 2^4 * 35.$

ii) $a \geq 2$

iii) $b_0 = 2^{35} \bmod 561.$

$= 263. \neq \pm 1$

$b_1 = (263)^2 \bmod 561.$

$= 166 \neq \pm 1$

$b_2 = (166)^2 \bmod 561.$

$= 67 \neq \pm 1.$

$b_3 = (67)^2 \bmod 561.$

$= 1.$

Composite number.

2) 53 a Prime number?

1) $53 \Rightarrow (n-1) = 52 = 2^2 \times 13$.

Addictive inverse of 53 is -1.

∴ It is prime number.

G.F Galois Fieldf of Prime number:

* order of finite field is given as p^n.
* used in polynomial prime
* Used in AES.

GF(2) =

$(0,1) = \{x, +, *\}$

$+_2$	0	1
0	0	1
1	1	0

$*_2$	0	1
0	0	0
1	0	1

a	0	1
-a	0	1
a^{-1}	-	1

GF(5):

$+_5$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	3	4	0	1	2
3	3	4	0	1	2
4	4	0	1	2	3

$*_5$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

a	-a	a^{-1}
0	0	-
1	4	1
2	3	3
3	2	2
4	1	4.

GF(7), GF(8).

$+_7$	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	2	3	4	5	6	0
2	2	3	4	5	6	0	1
3	3	4	5	6	0	1	2
4	4	5	6	0	1	2	3
5	5	6	0	1	2	3	4
6	6	0	1	2	3	4	5

$*_7$	0	1	2	3	4	5	6.
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4.
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1.

a	$-a$	a^{-1}
0	0	-
1	6	1
2	5	4
3	4	5
4	3	2
5	2	3
6	1	6

$+_8$	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

$*_8$	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

a	-a	a^{-1}
0	0	-
1	1	1
2	6	4
3	5	3
4	4	1
5	3	5
6	2	-
7	1	7

26/10/22

Not every element have inverse so we move to polynomial. (arx)

maximum degree = n-1. (eg: 2^3 degree = 3-1 = 2)

Polynomial addition ⇒ Xor.

0 - 000 → 0
1 - 001 → 1
2 - 010 → x
⋮
7 - 111 → x^2+x+1.

$$\begin{array}{r} x^2+x+1 \\ x^2+1 \\ \hline 0+x+0 \end{array}$$

$= x$ (2).

Polynomial multiplication ⇒ multiply and then Xor.

eg:

$$\begin{array}{r} x^2+x+1 \\ * \quad x^2+1 \\ \hline x^4+x^2+x^3+x+x^2+1 \end{array}$$

$m(x) = x^3 + x + 1.$

$$
\begin{array}{r|l}
 & x + 1 \\
\hline
x^3 + x + 1 & x^4 + x^3 + x + 1 \\
 & x^4 + x^2 + x \\
\hline
 & x^3 + x^2 + 1 \\
 & x^3 + x + 1 \\
\hline
 & \boxed{x^2 + x} = 6.
\end{array}
$$

$m(x) = x^8 + x^4 + x^3 + x + 1$ for $GF(2^8)$.

(2)

$$
\begin{array}{r}
x + 1 \\
\times \quad x^2 + x \\
\hline
x^3 + x^2 + x^2 + x \\
x^3 + x.
\end{array}
$$

$$
\begin{array}{r|l}
 & 1 \\
\hline
x^3 + x + 1 & x^3 + x \\
 & x^3 + x + 1 \\
\hline
 & \boxed{1}
\end{array}
$$

$f(x) = x^6 + x^4 + x^2 + x + 1$

$g(x) = x^7 + x + 1$

$m(x) = x^8 + x^4 + x^3 + x + 1.$

$f(x) + g(x) = x^7 + x^6 + x^4 + x^2$

multiplication:

$$= x^{13} + x^7 + x^6 + x^{11} + x^5 + x^4 + x^9 + x^3 + x^2 + x^8 + x^2 + x + x^7 + x + 1.$$

$$= x^{13} + x^{11} + x^9 + x^6 + x^5 + x^4 + x^3 + 1$$

$x^8 + x^4 + x^3 + x + 1$) $x^{13} + x^{11} + x^9 + x^6 + x^5 + x^4 + x^3 + 1$ ($x^5 + x^3 + 1$

$x^{13} + x^9 + x^8 + x^6 + x^5$

$x^{11} + x^8 + x^4 + x^3 + 1.$

$x^{11} + x^7 + x^6 + x^4 + x^3$

$x^8 + x^7 + x^6 + 1.$

$x^8 + x^4 + x^3 + x + 1$

$x^7 + x^6 + x^4 + x^3 + x$

Assignment:

Take one application apply any hashing function to get output. (eg: SHA-256)

$f(x) = x^2 + x + 1$

$g(x) = x^2$.

Addition: $x + 1$ (3)

multiplication: $x^4 + x^3 + x^2$

$$\begin{array}{r|l} & x+1 \\ \hline x^3+x+1 & x^4+x^3+x^2 \\ & x^4+x^2+x \\ \hline & x^3+x \\ & x^3+x+1 \\ \hline & 1 \end{array}$$

2. $f(x) = x^7 + x^6$

$g(x) = x^6 + x^5 + x^3 + 1$.

$m(x) = x^8 + x^4 + x^3 + x + 1$

Addition: $x^7 + x^5 + x^3 + 1$.

multiplication: $x^{13} + x^{12} + x^{10} + x^7 + x^{12} + x^{11} + x^9 + x^6$

$$\begin{array}{r|l} & x^5+x^3+x^2+1 \\ \hline x^8+x^4+x^3+x+1 & x^{13}+x^{12}+x^{11}+x^9+x^7+x^6 \\ & x^{13}+x^9+x^8+x^6+x^5 \\ \hline & x^{12}+x^{11}+x^8+x^7+x^6+x^5 \\ & x^{12}+x^8+x^7+x^5+x^4 \\ \hline & x^{11}+x^6+x^4+x^3 \\ & x^{11}+x^7+x^6+x^4+x^3 \\ \hline & x^7+x^6+x^5+x^4+x^3 \\ & x^7+x^3+x^2+x \\ \hline & x^6+x^5+x^4+x^2+x \end{array}$$

$x^8 + x^4 + x^2$

$x^8 + x^4 + x^3 + x + 1$

$x^3 + x^2 + x + 1$

$x^{10} + x^8 + x^6 + x^5 + x^4 + x^3$

$x^{10} + x^6 + x^5 + x^3 + x^2$

9/11/2022

Affine Cipher:

key : (α, β)

$f(x) = \alpha x + \beta$
↳ P.T

constraints:

(i) $GCD(\alpha, 26) = 1$

(ii) $1 \leq \alpha \leq 25$

(iii) $0 \leq \beta \leq 25$

Problem:

1) $\alpha = 7$ $\beta = 2$.

P.T = G.

$\therefore f(x) = (7(6) + 2)$

$= 44$. $(44 > 26) \Rightarrow 44 \mod 26$

$f(x) = 18$

$\therefore$ C.T = S

2). P.T = Hello.

For H:

$f(x) = 7(7) + 2$

$f(x) = 51 \bmod 26 = 25$

$\therefore CT = Z$

For E:

$f(x) = 7(4) + 2$

$= 30 \bmod 26 = 4$

$CT = E$

For L:

$f(x) = 7(11) + 2$

$= 79 \bmod 26 = 1$

$CT = B$

For L:

$CT = B$

For O:

$f(x) = 7(14) + 2$

$= 98 + 2$

$= 100 \bmod 26 = 22$

$CT = W$

$\therefore CT = ZEBBW.$

Decryption:

$$x = \alpha^{-1}(y - B) \qquad [\alpha \cdot \alpha^{-1} \bmod 26 = 1]$$

Finding Inverse:

A mod B : 11 mod 26.

A	B	Q	R	T1	T2	T = T1 − T2 × Q
26	11	2	4	0	1	−2
11	4	2	3	1	−2	5
4	3	1	1	−2	5	−7
3	1	3	0	5	−7	26
1	0			−7	26	

$$A^{-1} = -7 \bmod 26$$

$$A^{-1} = 19$$

7 mod 26

Q	A	B	R	T_1	T_2	T_1
3	26	7	5	0	1	-3
1	7	5	2	1	-3	4.
2	5	2	1	-3	4	-11
2	2	1	0	4	-11	26.
-	1	0	-	-11	26.	-

$\therefore \alpha^{-1} = -11 \bmod 26$

$\alpha^{-1} = 15$

C.T = ZEBBW.

For Z :

$x = 15(25-2)$

$= 15 \times 23.$

$= 345 \bmod 26$

$= 7 \quad (H)$

For E :

$x = 15(4-2)$

$= 30 \bmod 26$

$= 4. \quad (E)$

For B :

$x = 15(1-2)$

$= -15 \bmod 26$

$= 11 \ (L)$

For W :

$x = 15(22-20)$

$= 300 \bmod 26$

$= 14. \ (O)$

$\therefore$ P.T = HELLO.

Public key cryptography:

* Consist of 2 keys (Public and Private)
* Encryption done using public key
* Decryption done using Private key

$E \longrightarrow (P_{Ub} \times X)$ P_{Ub} - Public

$D \longrightarrow (P_{rb} \times E)$ P_{rb} - Private.

Terminologies used:

* Encryption Algorithm (Symmetric)
* Decryption Algorithm

RSA:

* Not very Secure
* Easily prone to attack

3 Steps:

* key generation
* Encryption
* Decryption

key generation :

* 7 Steps process

(i) Select prime numbers P and q

(ii) Calculate $n = P \times q$

(iii) Find your $\phi(n) = (P-1)(q-1)$
↑ Euler function

(iv) choose this value 'e' such that,

1. $1 < e < \phi(n)$
2. $\gcd(e, \phi(n)) = 1$.

(v) Calculate 'd' $= e^{-1} \bmod \phi(n)$.

can be rewritten, $ed \bmod \phi(n) = 1$.

(vi) Public key $P_{ub} = \{e, n\}$

Private key $P_{rb} = \{d, n\}$

Enc $\Rightarrow$ $C.T = M^e \bmod n$, $M < n$ ↳ length of the message.

Dec $\Rightarrow$ $P.T = C^d \bmod n$

Problem: (Commonly asked problem ☆.)

$p = 3, \quad q = 11, \quad n = 33.$

(i) $p = 3, \quad q = 11$

(ii) $n = pq = 3 \times 11 = 33$

(iii) $\phi(n) = (p-1)(q-1) = (2)(10) = 20.$

(iv) $e = 7 \quad \because \gcd(7, 20) = 1$

$1 < 7 < 20$

(v) $d = 7^{-1} \bmod 20.$

$d = 3$

$7 \cdot (7)^{-1} \bmod 20 = 1$

$7 \times 3 \bmod 20 = 1$

$\therefore 7^{-1} = 3$

(vi) $P_{Ub} = \{7, 33\}$

$P_{rb} = \{3, 33\}.$

(vii) $P.T = M \quad \ldots \quad (M = 31)$

Enc $\Rightarrow$ $C.T = 31^{7} \bmod 33.$

$C.T = 4.$

Dec $\Rightarrow$ $P.T = (4)^{3} \bmod 33$

$P.T = 31.$

* Brute force (trying with all possible combinations)
* Mathematical
* Timing
* Chosen ciphor text

Mathematical:

* Using factors of public key. we can find private key

Timing:

* Add constant time
* Add random delay

Chosen ciphor text:

* Based on properties.

1). $P = 17 \quad q = 11$.

(i) $P = 17, q = 11$

(ii) $n = 17 \times 11 = 187$.

(iii) $\phi(n) = 16 \times 10 = 160$

(iv) $e = 3$.

(v) $d = e^{-1} \bmod 160$

$d = 107$.

(vi) $P_{ub} = \{3, 187\}$

$P_{rb} = \{107, 187\}$.

$M = 2$

$C.T = M^e \bmod n$

$= (2)^3 \bmod 187$

$C.T = 8$

$P.T = 8^d \bmod n$

$= 8^{107} \bmod 187$

$P.T = 2$

28/11/2022

Diffie Hellman

* Private key x_A — value or number, less than q (prime number),

* α - primitive root q.

$a \bmod q, a^2 \bmod q, a^3 \bmod q, \ldots, a^{p-1} \bmod q.$

$\Downarrow$

$1, 2, 3, \ldots, p-1.$

* Only public key is exchanged,

$Y_A = \alpha^{x_A} \bmod q.$

* Key generation :

i) $K = (Y_B)^{X_A} \bmod q$.

ii) $K = (Y_A)^{X_B} \bmod q$.

$\Rightarrow K = (Y_A)^{X_B} \bmod q$

$= (\alpha^{X_A} \bmod q)^{X_B} \bmod q$

$= (\alpha^{X_A X_B}) \bmod q$.

$= (\alpha^{X_B})^{X_A} \bmod q$.

$= (\alpha^{X_B} \bmod q)^{X_A} \bmod q$

$= (Y_B)^{X_A} \bmod q$.

Problem :

$q = 11 \quad \alpha = 2 \quad X_A = 6 \quad X_B = 8$.

$Y_A = 2^6 \bmod 11$.

$= 9$.

$Y_B = 2^8 \bmod 11$

$= 3$.

$\therefore K = (3)^6 \bmod 11 = 3$

$K = (9)^8 \bmod 11 = 3$.

23.
25 6
11

Module - 3

* ELF

* SQL Injections.

3/12/2022.

* Fuzzing
* Port Scanning.
* ARP.

7/12/2022

Authentication :

* to determine authenticity.

* Traffic analysis :

* Masquerade :

* Content modification.

* Sequence modification:

* Timing (delay).

* Source repudiation.

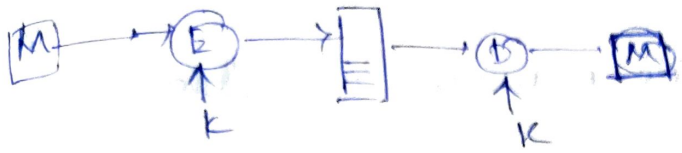
Message authentication function:

* 2 levels of auth.
* Low level → Authentication
* High / MAC

* Frame check sequence.

* Cryptographic check sum.

MAC = C(K, M)



uses:

* Network broadcasting

Disadv:

* can decrypt using brute force attack

HMAC:

12/12/2022.

PKI. (Public key Infrastructure).

Certificate authority:

* Collect all public keys.
* Should be trustable
* Secure communication
* Provide authorized access.

Example:

Uniformal PKI:

* Highly impossible to exist.

Everyone trust this PKI which is the center to all. only person to hold all PKI

* to remotely access resources through Virtual location.

* Person outside network can also access resources.

Electronic Banking:

* Financial transaction

* Person making transaction should be authoriz

* need PKI to secure and accept authorized transaction.

* need multiple level of Certificate authorit

* Individual CA for each bank.

Refinary Sensors:

* every process is automated.

* need PKI to prevent hacker from modifyi the information from sensors.

* CA to validate info from sensors.

Credit Card Validation:

* authorization of transaction.

* CA to validate authorized transaction.

Multilevel certification :

* Split CA into multiple levels.

eg: Bank transaction.

* PKI uses clock through which a particular digital signature will get expired. This improves security.

PKI reality :

* Name :
* Authority
* Trust

Indirect authentication :

* access control list.
* which is a drawback.

Direct authorization :

* For every key, authorization is provided,

Credential system ;

* Contains additional information..
* who can access along with time to access.
* micromanaging the entire system.
* very complex.

Revocation:

* Very difficult task.
* online certification protocol. for verification of Certificate
*
  - → Speed : max. time to revoke
  - → reliability: effective or not
  - → Connectivity: verify certificate valid or not
  - →

* Expiration can be used to overcome difficulties in revocation

Life of a key:

* Create a key
* obtain a certificate.
* Broadcast certificate details using PKI
* Active use / Passive use
* Expiration (must be stored in CL)

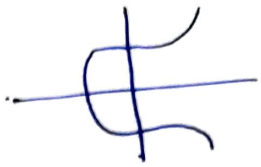
Certification format:

* required + optional details.
* Permission
* Root key,

Module - 9.

Elliptic curve cryptography:

$$y^2 = x^3 + ax + b.$$



Characteristics:

* Faster
* Efficient

Disadv:

* Difficult to implement
* Secure but slow.

Algorithm for Ecc : ECDSA.

Digital signature operation:

* key generation
* Signing algo.
* Signature verification

Digital certificates: