Page No.
Date : / /

Chapter 1
Introduction

* Three Security Goals:

1. Confidentiality.
2. Integrity.
3. Availability.

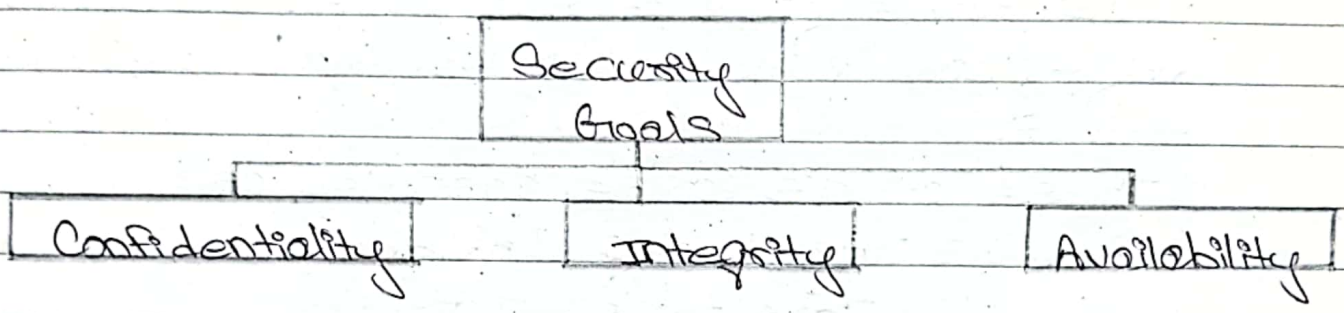


Fig 0 : Taxonomy of Security goals.

① Confidentiality :

Confidentiality refers to secrecy or privacy of data. It is probably the most common aspect of information security. We need to protect our confidential information.

An organization needs to guard against those malicious actions that endanger the confidentiality of its information.

② Integrity :

Information shouldnot be modified on the way in the digital world. Integrity ensures that the message received is the same as the message that was sent. In case, of message change it should be done only by authorized entities and through authorized mechanism.

Page No.
Date : / /

③ Availability:

The information created and stored by an organization needs to be available to authorized entities. Information needs to be constantly changed, which means it must be accessible to authorized entities.

* Security attacks that threaten Security Goals:

The three goals of security – Confidentiality, Integrity and Availability can be threatened by Security attacks.

These attacks can be classified as:

① Attacks Threatening Confidentiality,
② Attacks Threatening Integrity,
③ Attacks Threatening Availability and
④ Passive versus Active Attacks.

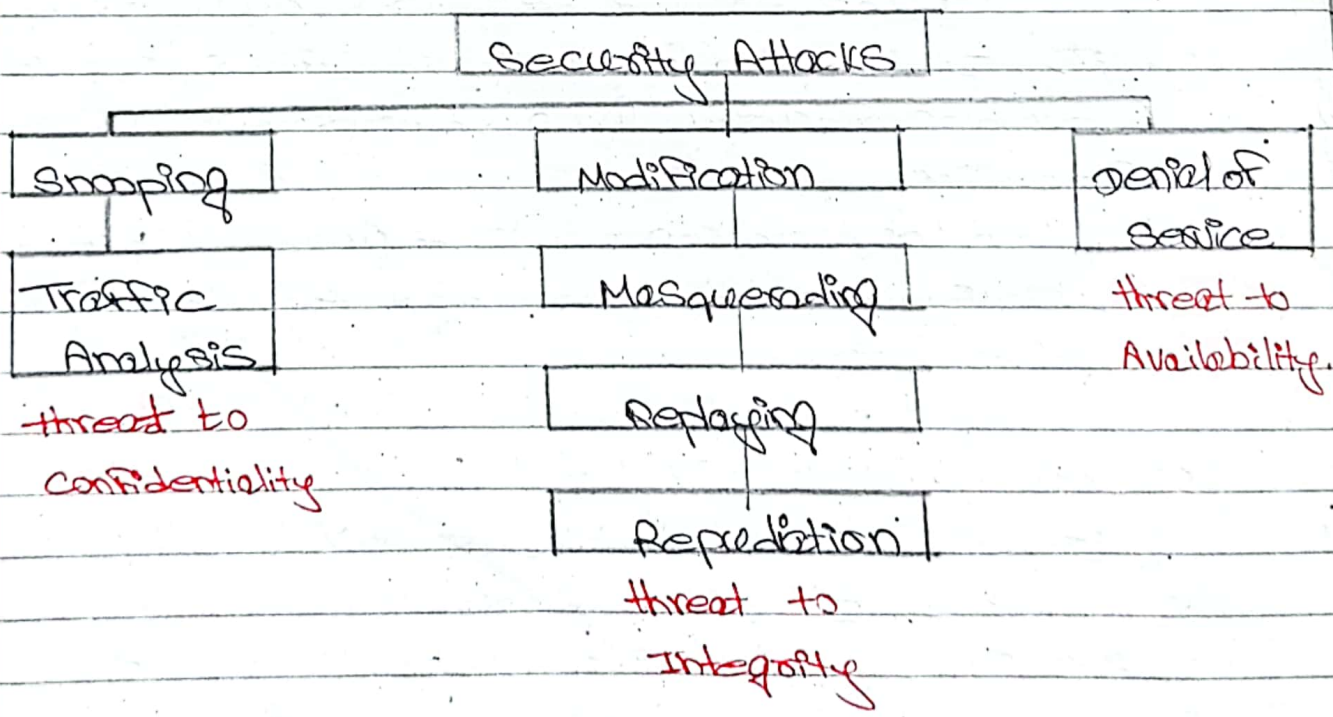


Fig: Taxonomy of attacks with relation to Security goals.

Page No.
Date : / /

① Threat to Confidentiality :

① Snooping:

Snooping, in security context, is unauthorized access to another person's or company's data. The practice is similar to eavesdropping which refers to unauthorized access to or interception of data.

② Traffic Analysis:

Traffic Analysis refers to obtaining information by monitoring online traffic.

② Threat to Integrity:

① Modification:

It means that the attacker intercepts the message and modifies or changes it.

② Masquerading:

A type of threat action whereby an unauthorized entity gains access to a system or performs a malicious act by illegitimately posing as an authorized entity. It happens when the attacker impersonates somebody else.

③ Replaying:

It means the attacker obtains a copy of a message sent by a user and later tries to replay it.

Page No.
Date : / /

④ Repudiation:

It means that Sender of the message might later deny that she has sent the message or the receiver of the message might later deny that he has received the message. Two kinds:- ① Source repudiation ② Destination repudiation

③ Threat to Availability:

① Denial of Service (DoS):

DoS is a very common attack. It may slow down or totally interrupt the service of a system.

* Passive versus Active Attacks:

Attacks	Passive/Active	Threatening
Snooping Traffic Analysis	Passive	confidentiality
Modification Masquerading Replaying Repudiation	Active	Integrity
Denial of Service	Active	Availability

* Services and Mechanisms:

ITU-T (International Telecommunications Union) provides some Security Services and some mechanisms to implement those services.

Security services and mechanisms are closely related because a mechanism or combination of mechanisms are used to provide a service.

Page No.
Date : / /

① Security Services:

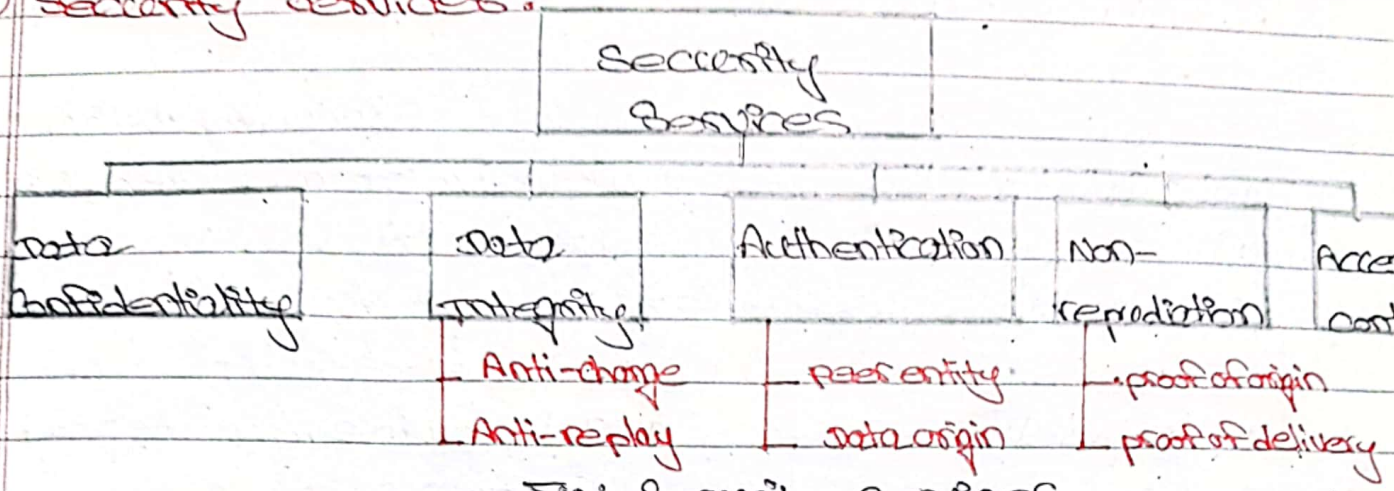


Fig:- Security Services

② Security Mechanism:

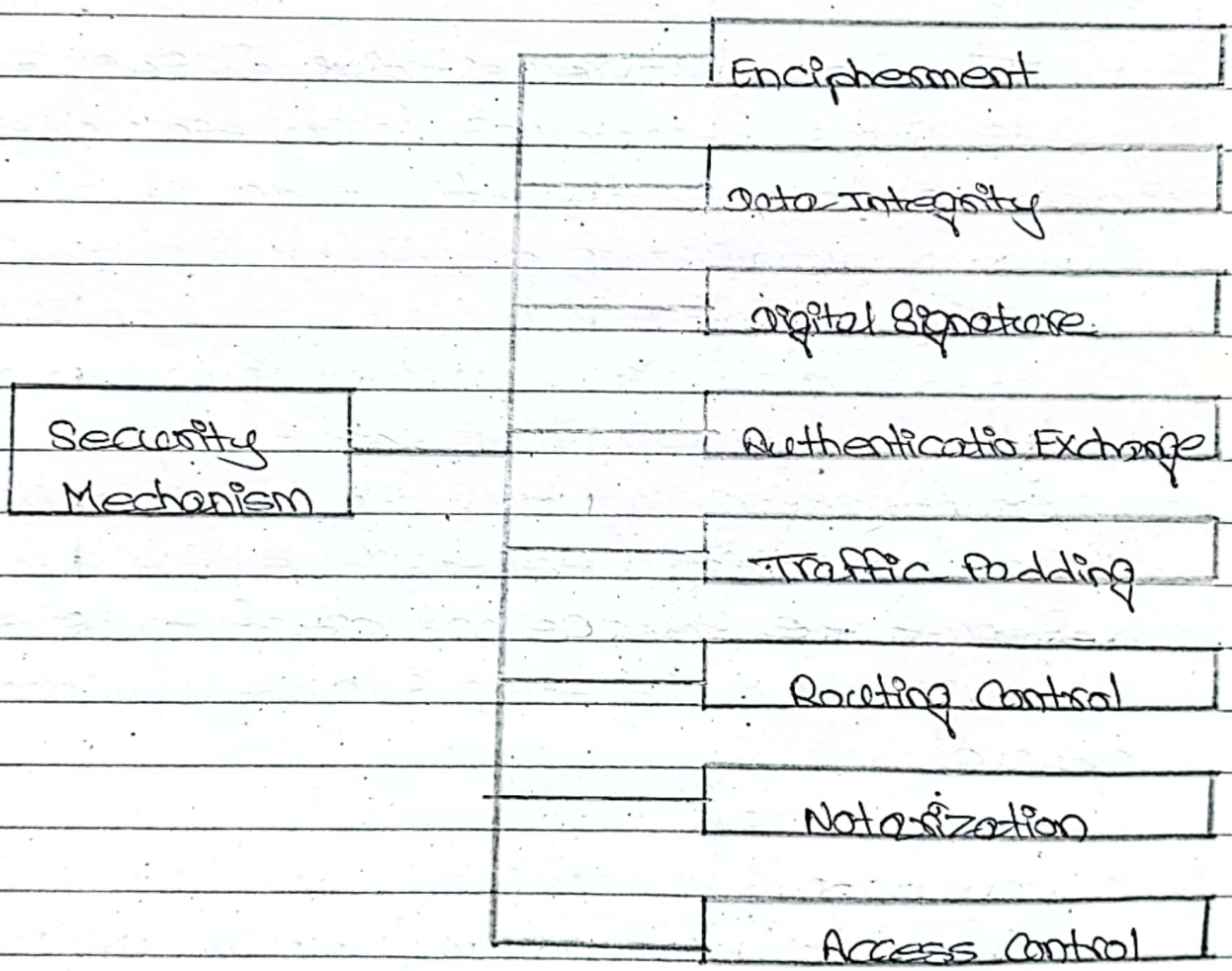


Fig:- Security Mechanism.

Page No.
Date : / /

③ Relation between Security Services and Mechanisms:

Security Service	Security Mechanism
Data Confidentiality	Encipherment and routing control
Data Integrity	Encipherment, digital signature, data integrity
Authentication	Encipherment, digital signature, Authentication Exchanges
Non-repudiation	Digital signature, data integrity, and notarization
Access control	Access Control Mechanism

The Actual implementation of security goals needs some techniques. Two techniques are prevalent today:

① Cryptography,
② Steganography.

① Cryptography:

Cryptography, a word with Greek origins, means "Secret writing". However, we use the term to refer to the science and art of transforming messages to make them secure and immune to attacks.

② Steganography:

The word steganography, with origin in Greek, means "Covered writing", in contrast with cryptography, which means "secret writing".

Page No.
Date : / /

eg:① Covering data with text.

This book is mostly about cryptography, not steganography.

□ □ □ □ □ □ □

0 1 0 0 0 0 1

eg:② Using dictionary

A friend called a doctor

0 10010 0001 0 01001

Chapter 2
Mathematics of Cryptography

* Integer Arithmetic:

In integer arithmetic, we use a set and a few operations. Some of the familiar set and the corresponding operations are:

1. Set of Integers,
2. Binary operations,
3. Integer Divison,
4. Divisibility
5. Linear Diophantine Equation.

(1) Set of Integers:

The set of Integers, denoted by Z, contains all integral numbers (with no fraction) from negative infinity to positive infinity. i.e.

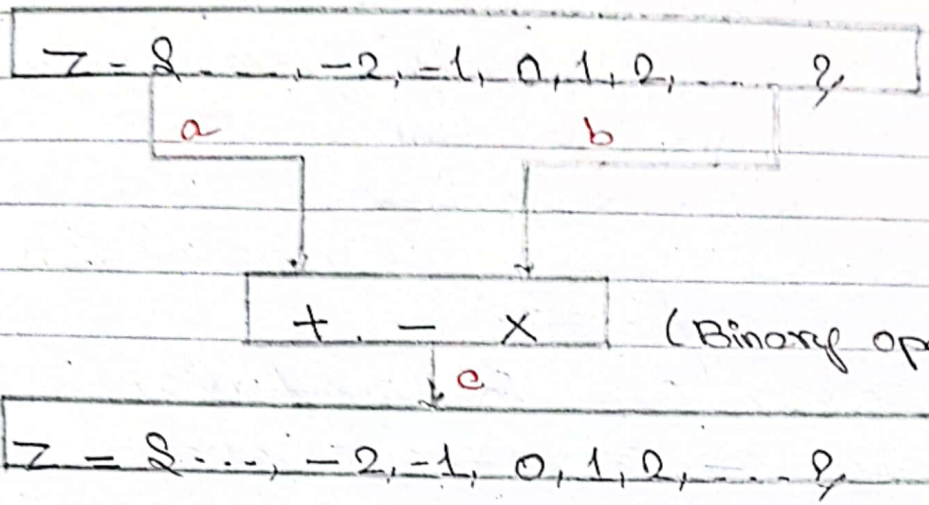
$$Z = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$$

(2) Binary operations:

In cryptography, we are interested in three binary operations (Addition, Subtraction and Multiplication) applied to the set of integers.

A binary operations takes two inputs and creates one output.

i.e.



Example:

The following shows the results of the three binary operations on two integers. Since, each input integer can be either positive or negative, we can have four cases for each operation.

Add: $5+9=14$ $(-5)+9=4$ $5+(-9)=-4$ $(-5)+(-9)=-14$

Subtract: $5-9=-4$ $(-5)-9=-14$ $5-(-9)=14$ $(-5)-(-9)=4$

Multiply: $5\times9=45$ $(-5)\times9=-45$ $5\times(-9)=-45$ $(-5)\times(-9)=45$

Integer Division:

In integer arithmetic, if we divide a by n, we can get q and r. The relationship between these four integers can be shown as

$$a = q \times n + r$$

Page No.
Date : / /

Where,

n = divisor
a = dividend
q = qutionent
r = reminder

i.e,

(dividend)
(divisor) n) a (q (qutionent)
—
r (remainder

① Example:

Assume that a = 255 and n = 11, we can find q = 23 and R = 2 using the division algorithm.

i.e, n → 11) 255 (23 ← q
a
-22
35
-33
2 ← r

② Example 2:

When we use a compiter or a calculator, r and q are negative when a is negative. So how can we apply the restriction that r needs to be positive?

The solution is simple, we decrement the value of q by 1 and we add the value of n to r to make it positive.

i.e. $-255 = (-23 \times 11) + (-2)$

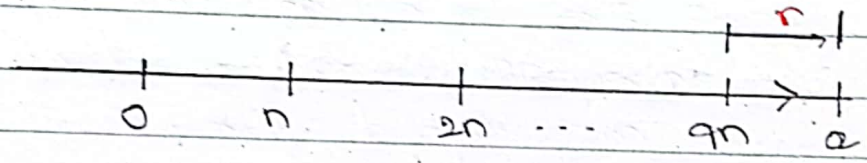
$a = qn + r$

Decrement value of q by 1 and add n to r i.e,

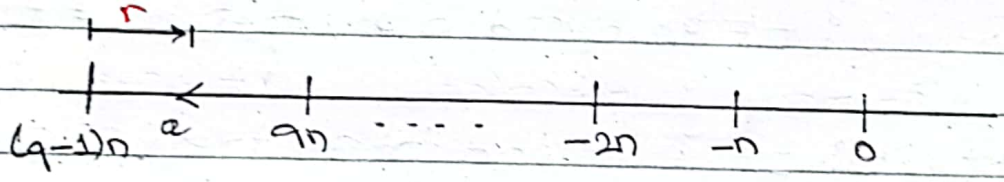
$-255 = (-24 \times 11) + (11 - 2)$

i.e, $-255 = (-24 \times 11) + 9$

* Graph of divison algorithm:



Case of positive a



Case of negative a

If a is not zero and we let $r = 0$ in the divison relation we get

$$\boxed{a = qn}$$

① If the remainder is zero, $a \mid n$

② If the remainder is not zero, $a \nmid n$

Page No.
Date: / /

Case (i):

The integer 4 divides the integer 32 because. $32 = 8 \times 4$. We show this as

$$4 \mid 32$$

Case (ii):

The number 8 doesnot divide the number 42 because. $42 = 5 \times 8 + 2$. There is a reminder, the number 2, in the equation. We show this as

$$8 \nmid 42$$

* Properties of Integer Divison:

① Property 1: If $a \mid 1$, then $a = \pm 1$

② Property 2: If $a \mid b$ and $b \mid a$, then $a = \pm b$

③ Property 3: If $a \mid b$ and $b \mid c$, then $a \mid c$

④ Property 4: If $a \mid b$ and $a \mid c$, then

$a \mid (m \times b + n \times c)$ where, m and n are arbitrary integers

eg:- $3 \mid 15$ and $3 \mid 9$

$3 \mid (15 \times 2 + 9 \times 4)$, i.e, $3 \mid 66$.

(b) (c)

Arbitrary integers

Page No.
Date : / /

Note: ① The integer 1 has only one divisor, itself.
② Any positive integer has at least two divisors, 1 and itself. (but it can have more).

* Greatest Common Divisor (GCD):

The greatest common divisor of two positive integers is the largest integer that can divide both integers.

* Euclidean Algorithm:

Fact 1: $gcd(a,0) = a$

Fact 2: $gcd(a,b) = gcd(b,r)$, where r is the remainder of dividing a by b

eg:-
Find the greatest common divisor of 2740 & 1760.

Sol $gcd(2740, 1760)$ (∵ $gcd(a,b) = gcd(b,r)$)
$gcd(1760, 980)$
$gcd(980, 780)$
$gcd(780, 200)$
$gcd(200, 180)$
$gcd(180, 20)$
GCD
$= 20$,,

eg:- Find the greatest common divisor of 25 & 60.

Sol $gcd(25, 60)$
$gcd(60, 25)$ (∵ switch(a,b))
$gcd(25, 10)$
$gcd(10, 5)$
GCD
$= 5$,,

Page No.
Date: / /

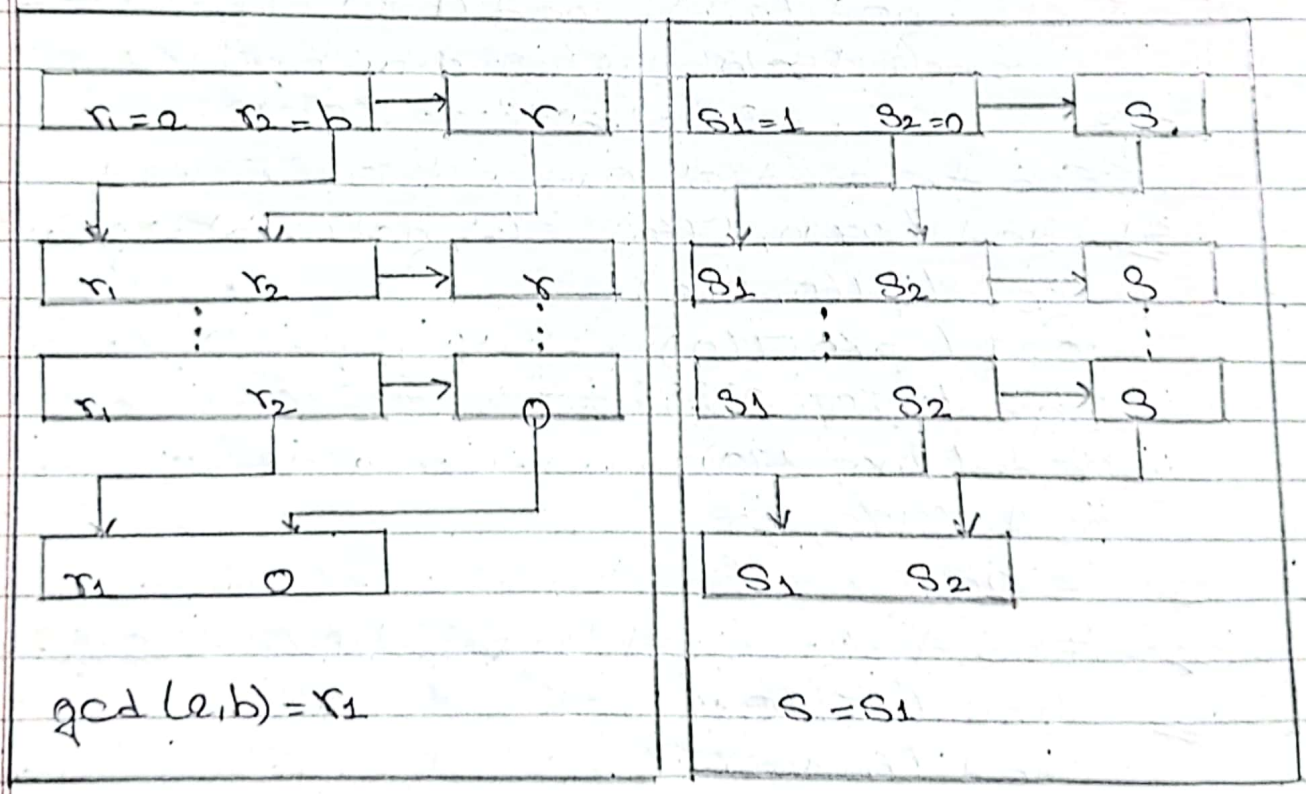
Note: When gcd (a,b) = 1, we say that a & b are relatively prime.

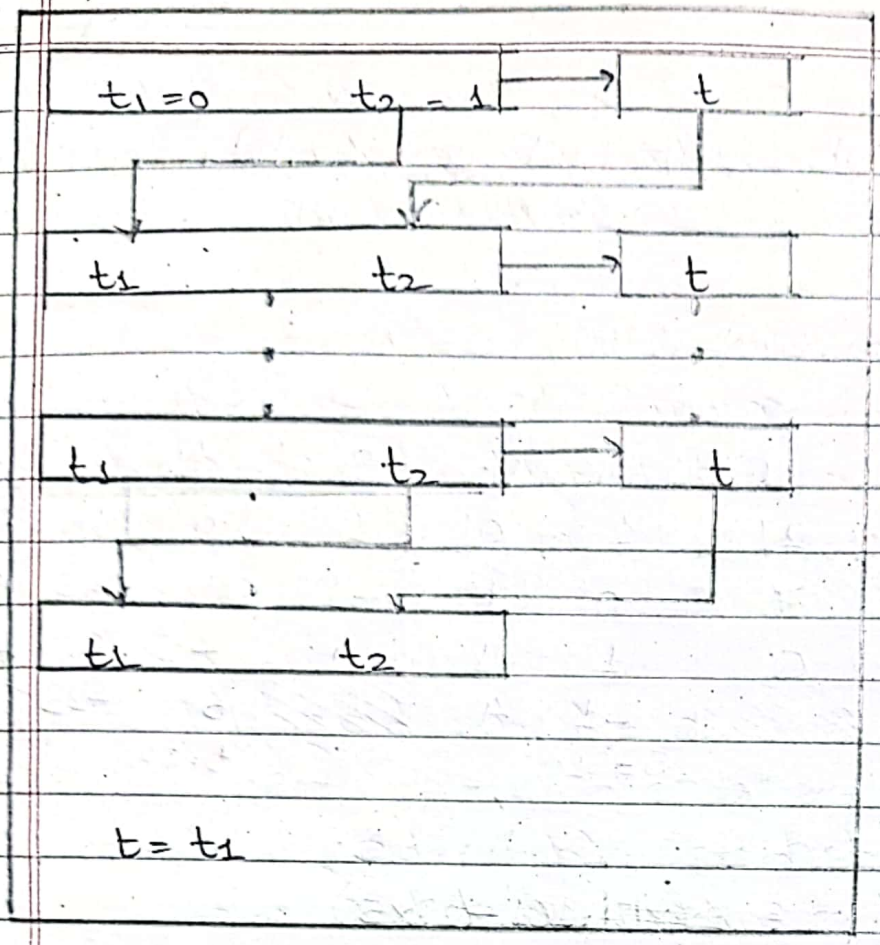
Extended Euclidean Algorithm:

Given, two integers a and b, we often need to find other two integers, S and t, such that

$$S \times a + t \times b = \gcd(a, b)$$

So, the Extended Euclidean algorithm can calculate the gcd (a,b) and at the same time calculate the values of S and t.





a) Process:

Initialization:

$r_1 \leftarrow a$; $r_2 \leftarrow b$;

$s_1 \leftarrow 1$; $s_2 \leftarrow 0$;

$t_1 \leftarrow 0$; $t_2 \leftarrow 1$;

while ($r_2 > 0$)

{

$q \leftarrow r_1 / r_2$;

updating r's

$r \leftarrow r_1 - q \times r_2$;

$r_1 \leftarrow r_2$; $r_2 \leftarrow r$;

updating s's

$s \leftarrow s_1 - q \times s_2$;

$s_1 \leftarrow s_2$; $s_2 \leftarrow s$;

updating t's

$t \leftarrow t_1 - q \times t_2$;

$t_1 \leftarrow t_2$; $t_2 \leftarrow t$;

}

gcd $(a, b) \leftarrow r_1$; $s \leftarrow s_1$; $t \leftarrow t_1$

Page No.
Date : / /

* Example:
Given a = 161 & b = 28, find gcd (a,b) & the values of s & t.

Sol

We use following table:

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	-5
1	28	21	7	0	1	-1	1	-5	6
3	21	7	0	1	-1	4	-5	6	-23
	7	0		-1	4		6	-23	

gcd = r_1 S = S_1 t = t_1

1) For 1st row:

$q = r_1 / r_2$ $r = 161 - (28 \times 5)$

$= 161 / 28$ $= 21$

$= 5.75$

So, $q = 5$

$s = s_1 - q \times s_2$ $t = t_1 - q \times t_2$

$= 1 - 5 \times 0$ $= 0 - 5 \times 1$

$= 1$ $= -5$

2) For 2nd row:

$q = r_1 / r_2$ $r = 28 - (21 \times 1)$

$= 28 / 21$ $= 7$

$= 1.33$

So, $q = 1$

$s = 0 - 1 \times 1$ $t = 1 - (1 \times -5)$

$= -1$ $= 6$

3) For 3rd row:

$q = r_1 / r_2$ $\quad r = 21 - (7 \times 3)$

$= 21/7$ $\quad = 0$

So, $q = 3$

$s = 1 - (3 \times -1)$ $\quad t = -5 - (3 \times 6)$

$= 4$ $\quad = -5 - 18$

$= -23$

∴ We get $\gcd(a,b) = \gcd(161, 28)$

$= 7$

$s = -1$

$t = 6$

* Example:

Given $a = 17$, and $b = 0$, find $\gcd(a,b)$ and the value of s and t.

Sol

We get

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
	17	0		1	0		0	1	

$\gcd(17, 0) = 17$,

$s = 1$

$t = 0$

Page No.
Date : / /

* Example:
Given a = 0 and b = 45, find gcd (a, b) and the values of s and t.

= Sol.
We get gcd (0, 45) = 45, s = 0, and t = 1.

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
0	0	45	0	1	0	1	0	1	0
	45	0		0	1		1	0	

* Modular Arithmetic:
The division relationship ($a = q \times n + r$) discussed in previous section has two inputs (a and n) and two outputs (q and r).

In Modular arithmetic, we are interested in only one of the outputs, i.e the remainder r.

* Modulo operator.
The modulo operator is shown as mod. The second input (n) is called the modulus. The output r is called the residue.

example.
a) 27 mod 5
b) 36 mod 12
c) −18 mod 14
d) −7 mod 10.

= Sol
a) 27 mod 5
= dividing 27 by 5 results in r = 2

Page No.
Date : / /

b) $36 \bmod 12$ (a = 36, n = 12)

= dividing 36 by 12 results in $r = 0$

c) $-18 \bmod 14$ (a = −18, n = 14)

= dividing −18 by 14 results in $r = -4$.

now, adding modulus (n) $r = -4 + 14$

$= 10$.

d) $-7 \bmod 10$

= dividing −7 by 10 results in $r = -7$

now, adding modulus (n), $r = -7 + 10$

$= 3$.

* Set of Residues:

The modulo operation creates a set, which in modular arithmetic is referred as the set of least residues modulo n, or Z_n.

i.e.

$$Z_n = \{0, 1, 2, 3, \ldots, (n-1)\}$$

for eg:-

$Z_2 = \{0, 1\}$ $\xrightarrow{eg}$ $3 \bmod 2 = 1$ or $2 \bmod 2 = 0$

$1 \in \{0, 1\}$ $\quad 0 \in \{0, 1\}$

$Z_6 = \{0, 1, 2, 3, 4, 5\}$

* Congruence:

To show that two integers are congruent, we use the congruence operator (≡). For eg:-

① $2 \equiv 12 \pmod{10}$

② $13 \equiv 23 \pmod{10}$

③ $3 \equiv 8 \pmod{5}$

④ $8 \equiv 13 \pmod{5}$

i.e,

$$-8 \equiv 2 \equiv 12 \equiv 22 \pmod{10}$$

* Example:-

We use modular arithmetic in our daily life; for example, we use a clock to measure time. our clock system use modulo 12 arithmetic.

However, instead of a 0 we use the number 12.

* Inverses:

When we are working in modular arithmetic, we often need to find the inverse of a number relative to an operation. (Additive or Multiplicative inverse).

We are normally looking for an additive inverse (relative to an addition operation) or a multiplicative inverse (relative to a multiplication operation).

① Additive Inverse:

In Z_n, two numbers a and b are additive inverses of each other if

$$a + b \equiv 0 \pmod{n}$$

Note: In modular arithmetic, each integer has an additive inverse. i.e The sum of an integer and its additive inverse is Congurent to 0 module n.

For example: Find all additive inverse pairs in Z_{10}.

Sol

The six possible pairs of additive inverses are (0,0), (1,9), (2,8), (3,7), (4,6) and (5,5).

② Multiplicative Inverse:

In Z_n, two numbers a and b are the multiplicative inverse of each other if

$$a \times b \equiv 1 \pmod{n}$$

Note: In modular arithmetic, an integer may or may not have a multiplicative inverse. When it does the product of the integer and its multiplicative inverse is congruent to 1 mod n.

* Example: Find the multiplicative inverse of 8 in Z_{10}.

There is no multiplicative inverse because gcd (10, 8) = 2 ≠ 1.

i.e, In other words, we cannot find any number between 0 and 9 such that when multiplied by 8, the result is congruent to 1.

* Example:

Find all multiplicative inverses in Z_{10}.

Sol

There are only three pairs: (1,1), (3,7), and (9,9). The numbers 0, 2, 4, 5, 6, and 8 do not have a multiplicative inverse.

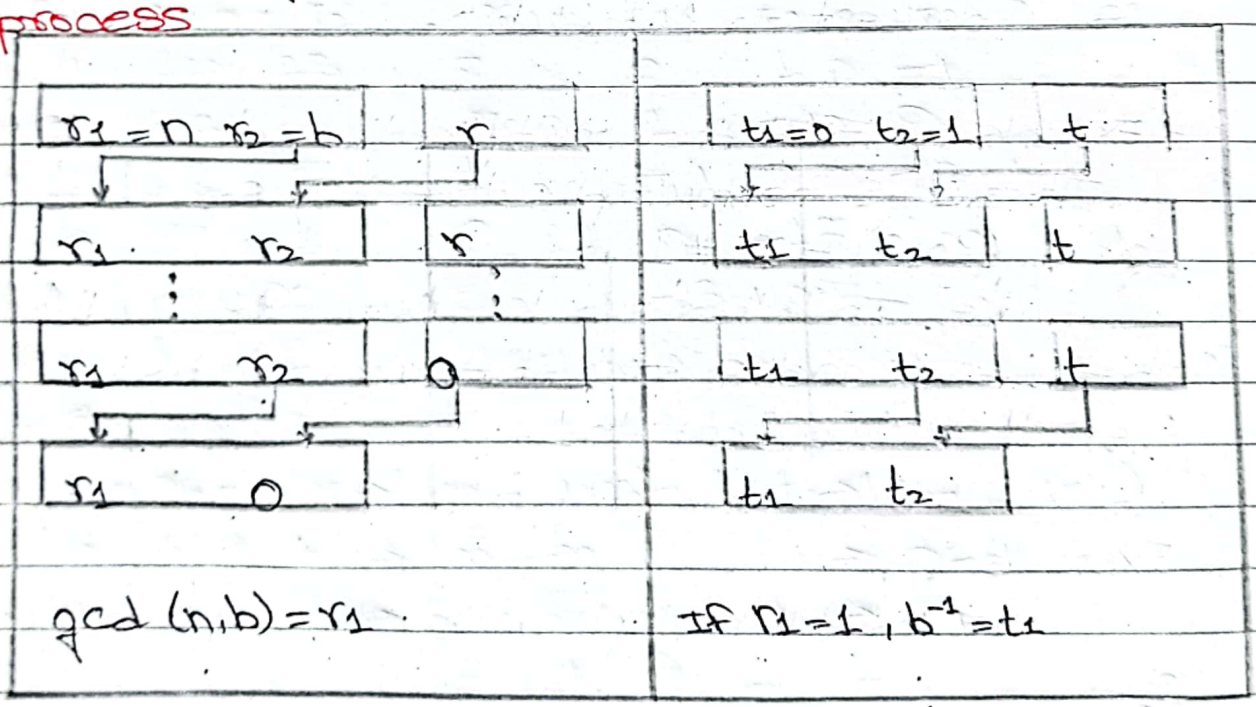
* Example:
Find all multiplicative inverse pairs in Z_{11}.

Sol
We have ~~seven~~ six pairs:
(1,1), (2,6), (3,4), (5,9), (7,8), (10,10).

* Extended Euclidean to find multiplicative inverse
Note: The extended Euclidean algorithm finds the multiplicative inverses of b in Z_n
i.e, when n and b are given and
$$\gcd(n,b) = 1$$

The multiplicative inverse of b is the value of t after being mapped to Z_n.

i) process



ii) Algorithm:

$r_1 \leftarrow n$; $r_2 \leftarrow b$;
$t_1 \leftarrow 0$; $t_2 \leftarrow 1$;

while ($r_2 > 0$)
{

$q \leftarrow r_1 / r_2$;

$r \leftarrow r_1 - q \times r_2$;
$r_1 \leftarrow r_2$; $r_2 \leftarrow r$;

$t \leftarrow t_1 - q \times t_2$;
$t_1 \leftarrow t_2$; $t_2 \leftarrow t$;
}

if ($r_1 = 1$) then $b^{-1} \leftarrow t_1$

* Find the Multiplicative inverse of 11 in Z_{26}.

Sol

q	r_1	r_2	r	t_1	t_2	t
2	26	11	4	0	1	-2
2	11	4	3	1	-2	5
1	4	3	1	-2	5	-7
3	3	1	0	5	-7	26
	1	0		-7	26	

① For 1st row:

$q \rightarrow r_1 / r_2$ $\quad r = 26 - (2 \times 11)$ $\quad t = 0 - (2 \times 1)$
$\rightarrow 26/11$ $\quad = 4$ $\quad = -2$
$\rightarrow 2$

② For 2nd row:

$q \rightarrow 11/4$ $\quad r = 11 - (2 \times 4)$ $\quad t = 1 - (2 \times -2)$
$\rightarrow 2$ $\quad = 3$ $\quad = 5$

③ For 3rd row:

$q \to 4/3$ $\quad r \to 4-(1\times3)$ $\quad t = -2-(1\times5)$

$\to 1$ $\quad = 1$ $\quad = -7$

④ For 4th row:

$q \to r_1/r_2$ $\quad r = 3-(3\times1)$ $\quad t = 5-(3\times-7)$

$\to 3/1$ $\quad = 0.$ $\quad = 5+21$

$\to 3$

∴ The gcd (26, 11) is 1 & the inverse of 11 is −7 or 19.

* Find the Multiplicative inverse of 23 in Z_{100}.

Sol

q	r_1	r_2	r	t_1	t_2	t
4	100	23	8	0	1	−4
2	23	8	7	1	−4	9
1	8	7	1	−4	9	−13
7	7	1	0	9	−13	100
	1	0		−13	100	

① For 1st row:

$q \to r_1/r_2$ $\quad r \to 100-(4\times23)$ $\quad t \to 0-(4\times1)$

$\to 100/23$ $\quad \to 100-92$ $\quad \to 0-4$

$\to 4$ $\quad \to 8$ $\quad \to -4$

② For 2nd row:

$q \to 23/8$ $\quad r \to 23-(2\times8)$ $\quad t \to 1-(2\times-4)$

$\to 2$ $\quad \to 7$ $\quad \to 1+8$

$\to 9$

③ For 3rd row:

$q \to 8/7$ $r \to 8-(1\times7)$ $t \to -1-(1\times9)$

$\to 1$ $\to 1$ $\to -1-9$

$\to -13$

④ For 4th row:

$q \to 7/1$ $r \to 7-(7\times1)$ $t \to 9-(7\times-13)$

$\to 7$ $\to 0$ $\to 9+91$

$\to 100$

∴ The gcd (100, 23) is 1; the inverse of 23 is −13 or 87.

* Find the inverse of 12 in Z_{26}.

= Sol

q	r_1	r_2	r	t_1	t_2	t
2	26	12	2	0	1	−2
6	12	2	0	1	−2	13
	2	0		−2	13	

① For 1st row:

$q \to 26/12$ $r \to 26-(2\times12)$ $t \to 0-(2\times1)$

$\to 2$ $\to 2$ $\to -2$

② For 2nd row:

$q \to 12/2$ $r \to 12-(6\times2)$ $t \to 1-(6\times-2)$

$\to 6$ $\to 0$ $\to 1+12$

$\to 13$

∴ The gcd (26, 12) is 2; the inverse doesnot exist.

Page No.
Date : / /

* Addition and Multiplication Tables:

① Addition Table in Z_{10}.

	0	1	2	3	4	5	6	7	8	9
0	0*	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	0*
2	2	3	4	5	6	7	8	9	0*	1
3	3	4	5	6	7	8	9	0*	1	2
4	4	5	6	7	8	9	0*	1	2	3
5	5	6	7	8	9	0*	1	2	3	4
6	6	7	8	9	0*	1	2	3	4	5
7	7	8	9	0*	1	2	3	4	5	6
8	8	9	0*	1	2	3	4	5	6	7
9	9	0*	1	2	3	4	5	6	7	8

② Multiplication Table in Z_{10}.

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1*	2	3	4	5	6	7	8	9
2	0	2	4	6	8	0	12	4	6	8
3	0	3	6	9	2	5	8	1*	4	7
4	0	4	8	2	6	0	4	8	2	6
5	0	5	0	5	0	5	0	5	0	5
6	0	6	2	8	4	0	6	2	8	4
7	0	7	4	1*	8	5	2	9	6	3
8	0	8	6	4	2	0	8	6	4	2
9	0	9	8	7	6	5	4	3	2	1*

Note: We need to use Z_n when additive inverses are needed, we need to use Z_n^* when multiplicative inverses are needed.

Page No.
Date : / /

Mathematics of Cryptography
Algebraic Structures:

Cryptography requires sets of integers and specific operations that are defined for those sets. The combination of the set and the operations that are applied to the elements of the set is called an algebraic structure.

We will define three common algebraic structures:
1. Groups
2. Rings
3. Fields.

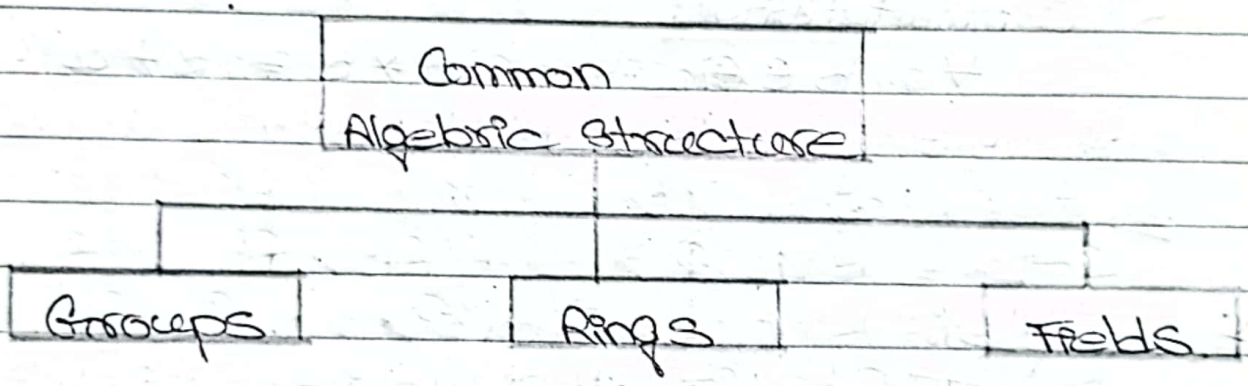


Fig: Common algebraic structure.

1. Group:

A group (G) is a set of elements with a binary operation (·) that satisfies four properties (or axioms). It is denoted as $< G, * >$

A commutative group satisfies an extra property, commutativity.

1) closure :
$\forall a, b \in G$ then $a * b \in G$

2) Associativity :
$\forall a, b, c \in G$ then $(a*b)*c = a*(b*c)$

3) Existence of identity :
$\forall a \in G \; \exists \; e \in G$ s.t $a*e = e*a = a$

4) Existence of Inverse :
$\forall a \in G \; \exists \; a^{-1} \in G$ s.t $a*a^{-1} = a^{-1}*a = e$

5) Commutativity :
$\forall a, b \in G$ then $a*b = b*a$
then G is called Abelian group

* Example;
The set of residue integers with the addition operator,
$G = \langle Z_n, + \rangle$,
where, $Z = \{ \ldots, -3, -2, -1, 0, 1, 2, 3, \ldots \}$

① closure :
$\forall a, b \in Z_n$ then $a*b \in Z$ satisfy closure, ✓

② Associativity :
$\forall a, b, c \in Z_n$ then $(a*b)*c = a*(b*c)$
satisfy Associativity.

③ Existence of Identity:

for Addition operation 0 is the identity element i.e.
$a+0 = 0+a = a$ so, satisfy Existence of identity.

④ Existence of Inverse:

We know that in modular arithmetic, each integer has an Additive inverse, therefore,

$$\forall a \in Z_n \; \exists \; a^{-1} \in Z \text{ s.t } a+(-a)=0$$
$$(-a)+a=0$$

satisfies Existence of Inverse

⑤ Commutativity:

$\forall a, b \in Z_n$ then $a*b = b*a$

therefore satisfies commutativity.

Therefore $\langle Z_n, + \rangle$ is a Abelian group.

Note: The set Z_n^* with the multiplication operator, $G_1 = \langle Z_n^*, * \rangle$, is also an abelian group.

Also,

$\langle Z_n, + \rangle$, Identity (e) is 0
$\langle Z_n, * \rangle$, Identity (e) is 1

* Example:

The set of residue integers with the Multiplication operator.

$G_1 = \langle Z_n, \times \rangle$

Sol

$\langle Z_n, \times \rangle$ satisfies the closure & Associative property.

Page No.
Date : / /

Let's Analyze the group $(Z, \times)$. First we need an Identity element. In this group, 1 would be our identity element. However, Problem arises with Inverses.

i.e. For any integer a, $a \times \frac{1}{a} = 1$.

However, for most integers, $\frac{1}{a}$ is not an element of Z. For eg:- $3 \times \frac{1}{3} = 1$.

but $\frac{1}{3}$ isn't an element of Z

$\therefore$ $\not<Z, \times>$ is not a group.

* Example: Is ~~$<Z_6, +>$~~ : $<Z_6, +>$ is a group.

Sol

$Z_n = \{0, 1, 2, 3, \ldots, n-1\}$

$Z_6 = \{0, 1, 2, 3, 4, 5\}$

$+_6$	0	1	2	3	4	5
0	0*	1	2	3	4	5
1	1	2	3	4	5	0*
2	2	3	4	5	0*	1
3	3	4	5	0*	1	2
4	4	5	0*	1	2	3
5	5	0*	1	2	3	4

From Above table we can say:

① It satisfies the closure property.

② Addition is always Associative.

③ Existence of Identity :- For Addition Identity element is 0

④ Existence of Inverse : for Addition there always exists a Inverse i.e. $a \times a^{-1} = e$ (from above table)

⑤ Commutative to $a * b = b * a$

$2 + 1 = 1 + 2$

∴ It is a abelian group.

* Example : IS $\langle Z_6, \times \rangle$ is a group.

$\times_6$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1*	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1*

Sol

$\langle Z_6, \times \rangle$ satisfies other properties but there isn't Existence of Inverse for 2, 3, 4, 2, so it is not a group.

Page No.
Date : / /

* Example: IS $\langle Z_7, \times \rangle$ is a group.

$\times_7$	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*

Sol $\langle Z_7, \times \rangle$ satisfies other properties and also Every Integer has its relative inverse so, it is a Group.

Note: $\langle Z_n, \times \rangle$ is not a group but $\langle Z_p, \times \rangle$ is a group.
where, p is prime no.

Q.n. $\langle Z_n, \times \rangle$ is not a group but $\langle Z_p, \times \rangle$ is a group where p is Prime no. Why?

Sol

In $\langle Z_n, \times \rangle$ not Every elements has it's relative inverse. but in $\langle Z_p, \times \rangle$ Every elements has it's relative inverse which helps to satisfy all the condition of a Group.

$$\gcd(a, b) = 1 \text{ in } \langle Z_p, \times \rangle$$

where, a and b are the integers $\in Z_p$.

Page No.
Date : / . /

* Cyclic group:

If a subgroup of a group can be generated using the power of an element, the subgroup is called the cyclic sub-group.

$$a^n \rightarrow a \cdot a \cdot \ldots \cdot a \text{ (n times)}$$

cube root of unity = $\{1 + \omega + \omega^2 = 0\}$

$\{1, \omega, \omega^2\}$

Q.No. Is the group $H = \langle Z_{10}, + \rangle$ a subgroup of the group $G_1 = \langle Z_{12}, + \rangle$?

Sol. The answer is no. Although H is a subset of G_1, the operations defined for these two groups are different.

The operation H is addition modulo 10, the operation in G_1 is addition modulo 12.

* Ring:

A ring, $R = \langle \{\ldots\}, +, \cdot \rangle$, is an algebric structure with two operations. i.e.

For +

1. closure
2. Associative
3. Existence of Identity
4. Existence of inverse
5. Commutativity

For .

1. closure
2. Associative
3. Distributive $(a \cdot (b+c) = a \cdot b + a \cdot c)$
4. Commutativity

Set = $\{a, b, c, \ldots\}$ operations = +, .

Ring.

* Example:

The Set Z with two operations, addition and multiplication, is a commutative ring. We show it by $R = \langle Z, +, \times \rangle$.

① Addition satisfies all of the FIVE properties,
② Multiplication satisfies only three properties.

* Field:

A Field, denoted by $F = \langle \{\ldots\}, +, \cdot \rangle$ is a Commutative ring in which the second operation satisfies all five properties defined for the first operation except that the identity element (i.e. 0) of the first operation has no inverse with respect to the second operation.

For +

① closure,
② Associativity,
③ Commutativity,
④ Existence of Identity,
⑤ Existence of Inverse.

For ·

① closure,
② Associativity,
③ Commutativity,
④ Existence of Identity,
⑤ Existence of Inverse.
⑥ No zero divisor.
⑦ Distributive.

set = {a, b, c, ...}
operation = {+, ×}

Note: The identity element of the first operation has no inverse with respect to the second operation.

* Example:

① Set of rational number Q

(2) Set of real numbers R
(3) set of complex numbers C

* Example: Set of rational number Q is a field.

Sol

The set rational number Q is an abelian group under the binary operation of "+"

Also, the set of rational numbers Q omitting "0" is an abelian group under binary operation "."

Now, to show that the set of rational numbers Q is distributive under the binary operation of "+" over "." i.e is.

$$a\cdot(b+c) = (a\cdot b)+(a\cdot c),\ \forall a,b,c \in Q$$

Let $a = \frac{2}{3}$, $b = \frac{-11}{4}$, $c = \frac{1}{9}$

$$a\cdot(b+c) = \frac{2}{3}\left(\frac{-11}{4}+\frac{1}{9}\right)$$

$$= \frac{2}{3}\left(\frac{-99+4}{36}\right)$$

$$= \frac{2}{3} \times \frac{-95}{36}$$

$$= \frac{-95}{54}$$

$$(a\cdot b)+(a\cdot c) = \frac{2}{3}\times\frac{-11}{4} + \frac{2}{3}\times\frac{1}{9}$$

$$= \frac{-22}{12} + \frac{2}{27}$$

$$= \frac{-95}{54}$$

∴ The set of rational number Q is a Field.

* Z_n is not a field but Z_p is a Field why?
* Give one example of group, abelian group, ring, Commutative ring, Integral domain & Field?
* Give one example of ring but not Field.

Galois Field:

Galois showed that for a field to be Finite, the number of elements should be p^n, where p is a prime and n is a positive integer.

Note: A Galois Field, $GF(p^n)$, is a Finite field with p^n elements.

When $n=1$, we have $GF(p)$ Field.

This Field can be the set Z_p, $\{0,1,\ldots,p-1\}$, with two arithmetic operations.

Example:

A very Common Field in this category is $GF(2)$ with the set $\{0,1\}$ and two operations, addition & Multiplication i.e.

$GF(2)$ where, $p=2$ & $n=1$

$\{0,1\}$ $\boxed{+ \times}$

+	0	1
0	0	1
1	1	0

Addition

×	0	1
0	0	0
1	0	1

Multiplication

Page No.
Date: / /

Inverses:

a	0	1
-a	0	1

Addition Inverse

a	0	1
a^{-1}	—	1

Multiplication Inverse.

Summary:

Algebric Structure	Supported Typical operations	Supported Typical Sets of Integers.
Group	(+ −) or (× ÷)	Z_n or Z_n^*
Ring	(+ −) and (×)	Z
Field	(+ −) and (× ÷)	Z_p

* GF(2^n) Fields:

In cryptography, we often need to use four operations (addition, subtraction, multiplication, and division). In other words, we need to use fields. We can work in GF(2^n) and uses a set of 2^n elements. The elements in this set are n-bit words.

eg:- Let us define GF(2^2) fields in which the set has four 2-bit words: {00, 01, 10, 11}.

We can redefine addition and multiplication for this field in such a way that all properties of these operations are satisfied - i.e.

+	00	01	10	11
00	00*	01	10	11
01	01	00*	11	10
10	10	11	00*	01
11	11	10	01	00*

Identity: 00

×	00	01	10	11
00	00	00	00	00
01	00	01*	10	11
10	00	10	11	01*
11	00	11	01*	10

Identity: 01

Page No.
Date : / /

* Polynomials:

A polynomial of degree n is an expression of the form.

$$f(x) = a_0 + a_1x^1 + a_2x^2 + \ldots + a_nx^n$$

where,

x^n is called the n^{th} term

a^n is the coefficient of the i^{th} term.

* example: We can represent the 8-bit word 10011001 using a polynomials.

n-bit word = 1 0 0 1 1 0 0 1

polynomial = $1x^7\ 0x^6\ 0x^5\ 1x^4\ 1x^3\ 0x^2\ 0x^1\ 1x^0$

First simplification = $1x^7 + 1x^4 + 1x^3 + 1x^0$

Second Simplication = $x^7 + x^4 + x^3 + 1$

* Example:

To Find the 8-bit word related to the polynomial $x^5 + x^2 + x$, we first supply the omitted terms. Since n = 8, it means the polynomial is degree 7.

The expanded polynomial is

$$0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x + 0x^0$$

8-bit word: 00100110.

Page No.
Date : / /

Note: Polynomials representing n-bit words use two fields. GF(2) and GF(2^n).

* Irreducible polynomials:

For the set of polynomials in GF(2^n), a group of polynomials of degree n is defined as the modulus. Such polynomials are referred to as irreducible polynomials.

Degree	Irreducible polynomials.
1	$(x+1), (x)$
2	(x^2+x+1)
3	$(x^3+x^2+1), (x^3+x+1)$
4	$(x^4+x^3+x^2+x+1), (x^4+x^3+1), (x^4+x+1)$
5	$(x^5+x^4+x^3+x^2+x+1), (x^5+x^4+x^3+x^2+1),$ $(x^5+x^4+x^3+x+1), (x^5+x^3+x^2+x+1),$ $(x^5+x^2+1),$

* Example:

Multiplication of $(x+1) \otimes (x^2+x)$ in GF(2^n) with irreducible polynomial of degree 3 (x^3+x+1)

Sol

$(x+1) \otimes (x^2+x)$

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

$= x^3+x^2+x^2+x$

$= x^3 + 2x^2 + x$

$= x^3 + x$

$x^3+x+1 \;\overline{)\; x^3+x \;(\; 1}$

x^3+x+1

1

Page No.
Date : / /

* Example:
Find the result of $(x^5+x^2+x) \otimes (x^7+x^4+x^3+x^2+x)$ in GF(2^8) with irreducible polynomial $(x^8+x^4+x^3+x+1)$

=Sol

$= x^5(x^7+x^4+x^3+x^2+x) + x^2(x^7+x^4+x^3+x^2+x) + x(x^7+x^4+x^3+x^2+x)$

$= x^{12}+x^9+x^8+x^7+x^6+x^9+x^6+x^5+x^4+x^3+x^8+x^5+x^4+x^3+x^2$

$= x^{12}+2x^9+2x^8+x^7+2x^6+2x^5+2x^4+2x^3+x^2$

$= x^{12}+x^7+x^2$

Now,

$$x^8+x^4+x^3+x+1 \;\overline{)\; x^{12}+x^7+x^2 \;(\; x^4+1}$$

$$x^{12}+x^8+x^7+x^5+x^4$$

$$x^8+x^5+x^4+x^2$$

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

$$x^5+x^3+x^2+x+1$$

Page No.
Date : / /

You can start from here .....
topics included before this are only prerequisites.

Page No.
Date : / /

Chapter 3
Traditional Symmetric-Key Ciphers

* Simple Cryptosystems:

The fundamental objective of cryptography is to enable two people to communicate over an insecure channel in such a way that an opponent cannot understand what is being said. This channel could be a telephone line or computer network.

For example:- The original message from Alice to Bob is called plaintext, the message that is sent through the channel is called the ciphertext. To create the ciphertext from the plaintext, Alice use an encryption algorithm and a shared secret key. To create the plaintext from ciphertext, Bob uses a decryption algorithm and the same secret key.

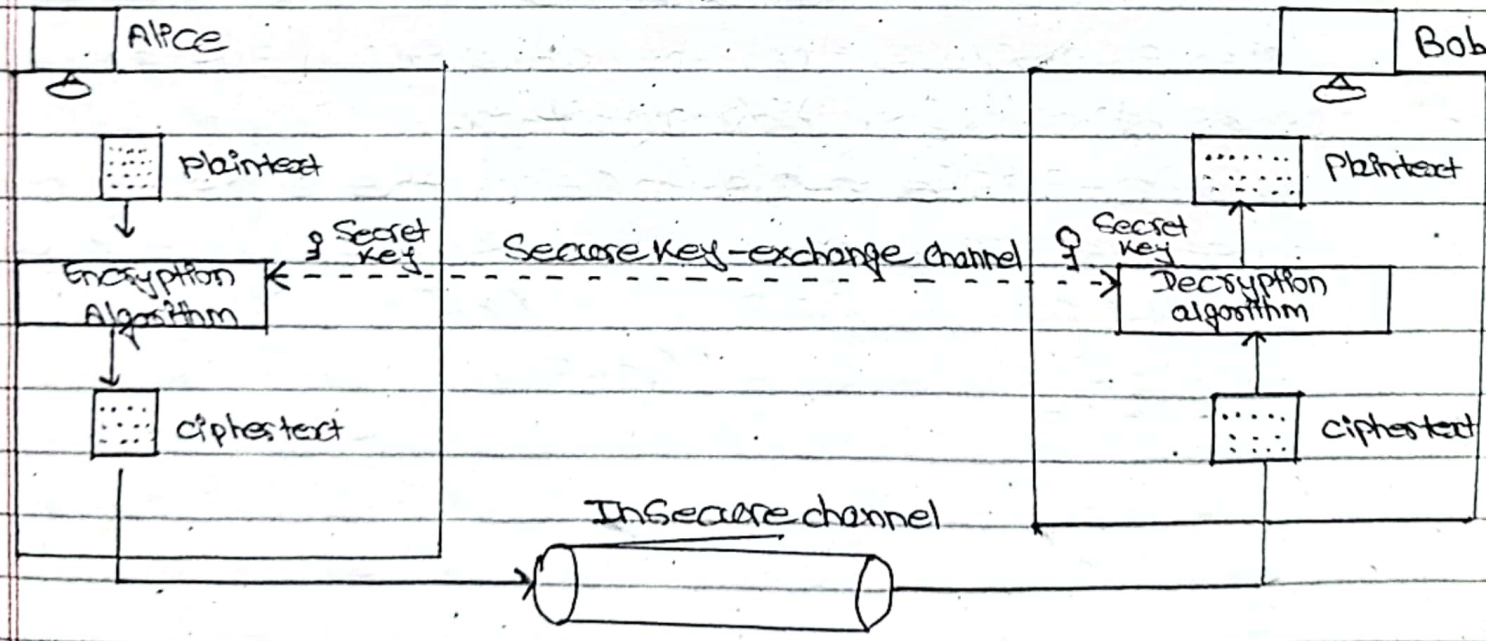


Fig of:- General idea of Symmetric-Key Cipher.

Page No.
Date : / /

A cryptosystem is a five-tuple (P, C, K, E, D), where the following conditions are satisfied.

1. P is a finit set of possible plaintexts;
2. C is a finite set of possible cipher texts;
3. K, the key space, is a finite set of possible keys;
4. For each $k \in K$,
   There is encryption rule : $e_k \in E$
and,
   Corresponding decryption rule: $d_k \in D$.

Each $e_k : P \to C$ and $d_k : C \to P$

These are functions such that $d_k(e_k(x)) = x$ for every plaintext element $x \in P$.

Kerchoff's Principle:

Based on Kerckhoff's Principle, one should always assume that the adversary, Eve, knows the encryption / decryption algorithm.

The resistance of the cipher to attack must be based only on the secrecy of the key.

Page No.
Date : / /

* Shift Cipher:

Shift cipher are basically additive ciphers, which is based on modular arithmetic.

Let $P = C = K = Z_{26}$. For $0 \le K \le 25$, define

$$e_K(x) = (x + K) \bmod 26$$
$$d_K(x) = (y - K) \bmod 26$$

where,

$$x, y \in Z_{26}$$

Example:

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

O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25

* Encrypt & Decrypt plaintext "hello" & K = 15

plaintext	Index	Encryption	Ciphertext
h	7	(7 + 15) mod 26 = 22	W
e	4	(4+15) mod 26 = 19	T
l	11	(11+15) mod 26 = 0	A
l	11	(11+15) mod 26 = 0	A
o	14	(14+15) mod 26 = 3	D

To decrypt the ciphertext, we need to subtract same key = 15 from each value (reducing modulo 26), & convert the sequence of integers to alphabetic characters.

Ciphertext	Index	Decryption	Plaintext
W	22	(22-15) mod 26 = 7	H
T	19	(19-15) mod 26 = 4	E
A	0	(0-15) mod 26 = 11	L
A	0	(0-15) mod 26 = 11	L
D	3	(3-15) mod 26 = 14	O

Note:- While subtracting if result is -ve add 26 to it.

Note: For the particular key = 3, the cryptosystem is often called the Caesar cipher, which was proposed by Julius Caesar.

* Substitution Techniques:

A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols. If the plaintext is viewed as a sequence of bits, then substitution techniques replaces plaintext bit pattern with cipher text bit pattern.

eg:- Caesar cipher

* Caesar Cipher:

The earliest known, and the simplest, use of a substitution cipher is Caesar cipher. It involves replacing each letter of the alphabet with the letter standing three places further down the alphabet.

Page No.
Date : / /

* for example:

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

plain text : meet me
cipher text : PHHW PH

This algorithm can be expressed as follow:

$$C = E(3, p) = (p+3) \bmod 26$$
$$p = D(3, C) = (C-3) \bmod 26$$

A shift may be of any amount, so the general Caesar algorithm is:

$$C = E(K, p) = (p+K) \bmod 26$$
$$p = D(K, C) = (C-K) \bmod 26$$

where K takes on a value in the range 1 to 25.

If it is known that a given ciphertext is a Caesar cipher then brute-force cryptanalysis is easily performed. Simply try all the 25 possible keys.

The major drawbacks of these cipher are:

1. The encryption & decryption algorithms are known,
2. There are only 25 keys to try.
3. The language of the plaintext is known and easily recognizable.

So, with only 25 possible keys, the Caesar cipher is far from secure. So, a dramatic increase in the key space can be achieved by allowing an arbitrary

Page No.
Date : / /

Substitution: i.e It is more convenient to think of encryption and decryption as Permutations of alphabetic characters.

In general, there are n! permutations of a set of n elements, because the first element can be chosen in one of n ways, the second in n-1 ways, the third in n-2 ways & so on.

If, Instead, the "cipher" line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than 4×10^{26} possible keys. which would seem to eliminate brute-force techniques for cryptanalysis.

Mathmatically,

Let $P = C = Z_{26}$, K Consists of all possible permutation of the 26 Symbols 0, 1, ... 25. for each permutation $\pi \in K$, define

$$e_\pi(x) = \pi(x),$$
$$d_\pi(y) = \pi^{-1}(y),$$

where π^{-1} is the inverse permutation to π.

Example:- An example of a random permutation, π, which could Comprise an encryption function. (plaintext characters are written in lower case and ciphertext characters are written in upper case).

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
X	N	Y	A	H	P	O	G	Z	Q	W	B	T	S	F	L

q	r	s	t	u	v	w	x	y	z
R	C	V	M	U	E	K	J	D	I

Thus, $e_K(a) = X$, $e_K(b) = N$, etc.

The decryption function is the inverse permutation. which is formed by writing second line first & sorting alphabetically.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
d	l	r	y	v	o	h	e	z	x	w	p	t	b	q	f	j	q	n

T	U	V	W	X	Y	Z
m	u	s	k	a	c	i

Hence, $d_K(A) = d$, $d_K(B) = l$, etc.

Note: However, It seem very secure by using permutation of the 26 alphabetic characters. we shall see later that a substitution cipher can easily be cryptanalyzed by other methods.

* Program to Convert lower case into upper case alphabetic?

```
for (i=0; i< strlen(p); i++)
& p[i] = to upper (p[i])
 c[i] = (p[i] - 65 + k) mod 26 + 65
&
```

Page No.
Date: / /

* Affine Cipher:

Affine Cipher is a special case of the substitution cipher. In Affine Cipher we use two different keys. It also provides the idea that every key is not valid to encrypt.

In affine cipher inorder to decryption to be feasible, it is necessary to ask when an affine function $((ax+b) \bmod 26)$ is Injective.

Injective : many to one is not or i.e one to one property.

If it violets the injective property, it is difficult to encrypt & decrypt.

In $(ax+b) \bmod 26$ affine function, a, b are keys $\in Z_{26}$

If $\gcd(a, 26) = 1$ then Affine is possible other it violets injective property & Affine cipher is not possible.

for eg:- $(a, b) = (4, 7)$

$\gcd(a, 26) = 1$

$\gcd(4, 26) = 2 \neq 1$

Sol.

$e(x) = ax + b \bmod 26$

$= 4x + 7 \bmod 26$

$e(a) = 4 \times 0 + 7 \bmod 26 = 7$ * H

$e(h) = 4 \times 7 + 7 \bmod 26 = 9$ J

$e(n) = 4 \times 13 + 7 \bmod 26 = 7$ * H

while encrypting a & n the answer is same H which

violets the injective property, so Affine font feasible.

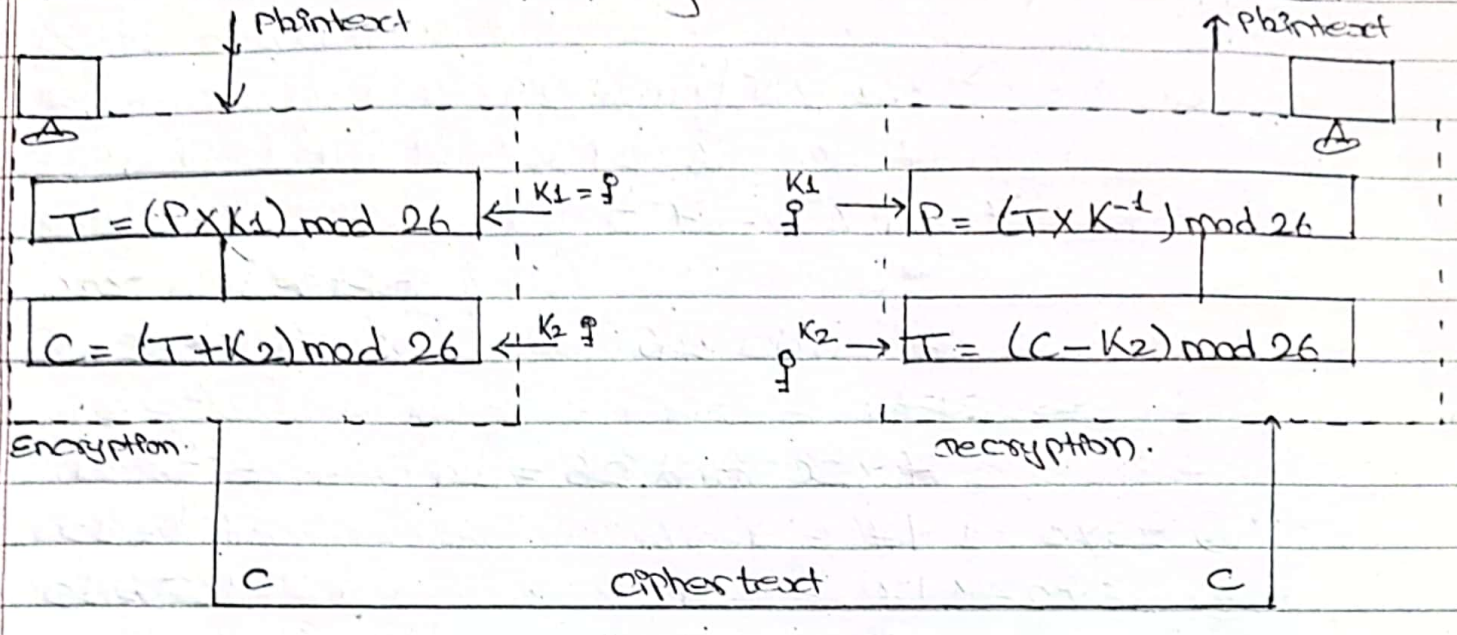


Fig:- Affine cipher.

$C = (P \times K_1 + K_2) \bmod 26$

$P = (C - K_2) \times K^{-1} \bmod 26$

where,

K_1^{-1} is the multiplicative inverse of K_1

$-K_2$ is the additive inverse of K_2.

* Example: Encrypt & Decrypt "hello" with key $(\overset{a}{7}, \overset{b}{2})$

Sol: gcd (a, 26) = 1

gcd (7, 26) = 1 So, Affine cipher is possible.

plaintext	index	encryption	ciphertext
h	7	(7X7+2) mod 26 = 25	Z
e	4	(4X7+2) mod 26 = 4	E
l	11	(11X7+2) mod 26 = 1	B
l	11	(11X7+2) mod 26 = 1	B
o	14	(14X7+2) mod 26 = 22	W

Page No.
Date: / /

For Decryption.

$P = (C - K_2) \times K^{-1} \bmod 26$

$K_1 = 7$, $K^{-1} = 7^{-1}$

$7^{-1} \bmod 26$

$7 \times x \bmod 26 = 1$

① $26 \times 1 + 1 / 7 = 3.58$
② $26 \times 2 + 1 / 7 = 7.57$
③ $26 \times 3 + 1 / 7 = 11.28$
④ $26 \times 4 + 1 / 7 = 15$

Note: Multiplicative Inverses in Z_{26}

$1^{-1} = 1$
$3^{-1} = 9$
$5^{-1} = 21$,
$7^{-1} = 15$
$11^{-1} = 19$
$17^{-1} = 23$
$25^{-1} = 25$

i.e, $7^{-1} = 15$

$7 \times x \bmod 26 = 1$

$7 \times 15 \bmod 26 = 1$

ciphertext	Index	Decryption	plaintext
Z	25	$(25-2) \times 15 \bmod 26 = 7$	H
E	4	$(4-2) \times 15 \bmod 26 = 4$	E
B	1	$(1-2) \times 15 \bmod 26 = 11$	L
B	1	$(1-2) \times 15 \bmod 26 = 11$	L
W	22	$(22-2) \times 15 \bmod 26 = 14$	O

Note: In case of -ve add 26 to it i.e,

$(1-2) = -1 + 26 = 25$

$25 \times 15 \bmod 26 = 11$

Monoalphabetic cipher } shift cipher
substitution cipher.

Page No.
Date : / /

* The Vigenere Cipher:

In both the shift cipher and the substitution cipher, once a key is chosen, each alphabetic character is mapped to a unique alphabetic character, these cryptosystem are called mono alphabetic cryptosystems.

Vigenere cipher is a polyalphabetic cryptosystem i.e polyalphabetic substitution cipher.

Mathematically,

Let m be a positive integer. Define $P = C = K = (Z_{26})^m$. For a key $K = (K_1, K_2, \ldots, K_m)$, we define.

$$e_K(x_1, x_2, \ldots, x_m) = (x_1 + K_1, x_2 + K_2, \ldots, x_m + K_m)$$
$$d_K(y_1, y_2, \ldots, y_m) = (y_1 - K_1, y_2 - K_2, \ldots, y_m - K_m).$$

where, all operations are performed in Z_{26}

example: Suppose m = 6 and the keyword is CIPHER. This corresponds to numerical equivalent K = (2, 8, 15, 7, 4, 17). Suppose the plaintext is string:

This cryptosystem ~~is not secure~~.

plaintext	T	h	i	s		c	r	y	p	t	o		s	y	s	t	e	m
PS value	19	7	8	18		2	17	24	15	19	14		18	24	18	19	4	12
key stream	2	8	15	7		4	17	2	8	15	7		4	17	2	8	15	7
ciphervalue	21	15	23	25		6	8	0	23	8	21		22	15	20	1	19	19
ciphertext	V	P	X	Z		G	I	A	X	I	V		W	P	U	B	T	T

now,

Similarly decryption can be done as:

Ciphertext	V	P	X	Z	G	I	A	X	I	V	W	P	U	B	T	T
Ciphervalue	21	15	23	25	6	8	0	23	8	21	22	15	20	1	19	19
key stem	2	8	15	7	4	17	2	8	15	7	4	17	2	8	15	7
p's value	19	7	8	18	2	5	24	15	19	14	18	24	18	19	4	12
plaintext	T	h	i	s	c	r	y	p	t	o	s	y	s	t	e	m

The number of possible keywords of length m in Vigenere Cipher is 26^m, so even for relatively small value of m, an exhaustive key search would require a long time.

In Vigenere cipher having keyword length m, an alphabetic character can be mapped to one of m possible alphabetic characters. (therefore, it is polyalphabetic cryptosystem)

* Hill cipher:

Hill cipher is another poly alphabetic substitution cipher. This cipher was invented in 1929 by Lester S. Hill.

Mathmatically,

Let $m \geq 2$ be an integer. Let $P = C = (Z_{26})^m$ and let

$K = \{m \times m$ invertible matrices over $Z_{26}\}$

For a key K, we define.

$$e_K(p) = (K, p) = pK \bmod 26$$
$$d_K(c) = (K, C) = cK^{-1} \bmod 26$$
$$= pK^{-1}K.$$
$$= p$$

where, all operations are performed in Z_{26}

Page No.
Date: / /

Note:- ① The key K can be a square matrix of size m×m
② m×m matrix should be invertible matrix i.e, $K \cdot K^{-1} = I$.
③ m×m matrix of key should not be a singular matrix. i.e A square matrix is singular if & only if its determinant is "0".
④ We need to find the multiplicative inverse of K to decrypt the cipher text.

Example: Suppose the key matrix is 2×2 square matrix $K = \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix}$ & we need to encrypt the plain text july.

=Sol

July is corresponding to words (9, 20, 11, 24)$_{1\times4}$ since we are taking K matrix of form 2×2 we need to convert given plaintext into 1×2 matrix to ~~for~~ make them multiplicable i.e. (9 20), (11, 24)

now,

$$ek(p) = PK \bmod 26$$
$$= [9\ 20] \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} \bmod 26$$
$$= [9\times11 + 20\times3 \quad 9\times8 + 20\times7]$$
$$= [159\ 212] \bmod 26$$
$$= [3\ \ 4] = [D\ E]$$

again, for (11 24) $= [11\ 24] \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} \bmod 26$

$$= [121 + 72 \quad 88 + 168] \bmod 26$$
$$= [11\ \ 22]$$
$$= [L\ W]$$

Page No.
Date : / /

So, the Ciphertext of "July" = DELW

now, before decryption we need to find K^{-1}

ie,

$$K^{-1} = \frac{1}{\det.K} \begin{bmatrix} 7 & -8 \\ -3 & 11 \end{bmatrix}$$

$$= \frac{1}{(77-24)} \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix}$$

$-8+26 = 18$
$-3+26 = 23$

$$= \frac{1}{53} \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix}$$

$$= \frac{1}{1^{0}} \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix}$$

$$= 1^{-1} \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix}$$

Since multiplicative inverse of 1 is 1 itself K^{-1} is

$$\begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix}$$

now,

$$dK(c) = CK^{-1} \bmod 26$$

$$= [3 \ 4] \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} \bmod 26$$

$$= [21+92 \quad 54+44] \bmod 26$$

$$= [9 \ 20] = [J \ U]$$

Page No.
Date : / /

for; $[11 \quad 22] \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} \bmod 26$

$= [77+506 \quad 198+242] \bmod 26$

$= [11 \quad 24] = [L \quad Y]$

we can also use $dk(c) = P\,KK^{-1}$

i.e.

(i) $= [9 \quad 20] \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix}$ $\because KK^{-1} = I$

Invertible Matrix.

$= [9 \quad 20] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$= [9 \quad 20]$.

(ii) $[11 \quad 24] \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} 7 & 18 \\ 23 & 18 \end{bmatrix}$

$= [11 \quad 24] \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$= [11 \quad 24]$

Page No.
Date: / /

Let's see a example where decryption is not feable.

Example:

plaintext = [February] = [5 4 1 17 20 0 17 24]

key = July = $\begin{bmatrix} J & u \\ L & y \end{bmatrix} = \begin{bmatrix} 9 & 20 \\ 11 & 24 \end{bmatrix}$

now, Encryption = $[5 \quad 4] \begin{bmatrix} 9 & 20 \\ 11 & 24 \end{bmatrix}$

$= [45+44 \quad 100+96] \bmod 26$

$= [89 \quad 196] \bmod 26$

$= [11 \quad 14]$

$= [L \quad O]$

now, decryption we need to find K^{-1}

$$K = \begin{bmatrix} 9 & 20 \\ 11 & 24 \end{bmatrix}$$

$$K^{-1} = \frac{1}{\det K} \begin{bmatrix} 24 & -20 \\ -11 & 9 \end{bmatrix}$$

$$= \frac{1}{(216-220)} \begin{bmatrix} 24 & 6 \\ 15 & 9 \end{bmatrix}$$

$$= \frac{1}{4} \begin{bmatrix} 24 & 6 \\ 15 & 9 \end{bmatrix}$$

Page No.
Date : / /

$$A^{-1}\begin{bmatrix} 24 & 6 \\ 15 & 9 \end{bmatrix}$$

Since, there is not multiplicative inverse of 4 we can't decrypt these cipher text.

Note:- To check whether a number has it's multiplicative inverse we use gcd i.e.

gcd (4, 26) = 2 doesn't exist

gcd (9, 26) = 1 multiplicative inverse exist

* Permutation cipher: (Transposition Techniques)

All of the cryptosystems we discussed so far involved substitution i.e plaintext characters are replaced by different ciphertext characters.

The idea of a permutation cipher is to keep the plaintext characters unchanged, but to alter their positions by rearranging them using a permutation.

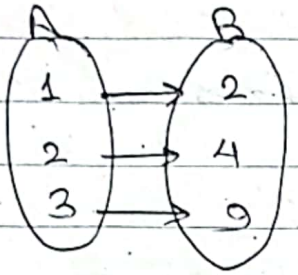
Before looking at How permutation cipher works, we need to define 3 Function: ① Injective Function
② Surjective Function
③ Bijective Function.

Page No.
Date : / /

① Injective Function:

- A function that always maps the distinct element of its domain to the distinct element of its co-domain.
- It is also known as one-to-one function.

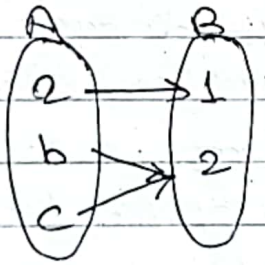
eg:-



② Surjective Function:

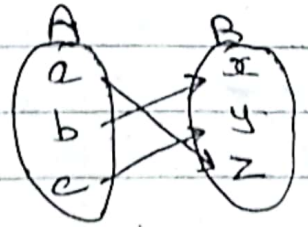
- A function that maps one or more elements of A to the same element of B.
- It is also known as onto function.

eg:-



③ Bijective function:

- A function that is both injective and surjective means that every element "b" in the co-domain B, there is exactly one element "a" in the domain A. such that $f(a) = b$.
- It is also known as one-to-one correspondence.



Page No.
Date : / /

now,

A permutation of a finite set X is a bijective function $\pi : X \to X$. In other words, the function π is one-to-one (injective) and onto (surjective).

i.e, for every $x \in X$, there is a unique element $x' \in X$ such that

$$\pi(x') = x$$

which allow to define inverse permutation,

$$\pi^{-1}(x) = x'$$

eg:- Suppose $m = 6$ and the key is the following permutation π.

x	1	2	3	4	5	6
$\pi(x)$	3	5	1	6	4	2

now, The inverse permutation π^{-1} can be constructed by interchanging the two rows. and making first row in increasing order with its respective value in π^{-1} as:

x	1	2	3	4	5	6
$\pi^{-1}(x)$	3	6	1	5	2	4

Suppose the plaintext is "crypto" then, partition into group of 6 we get

"crypto"

each group of six letters is re arranged as permutation π (ciphertext): "yptroc"

The ciphertext can be decrypted using inverse permutation as π^{-1} (plain text): "crypto".

Stream cipher & Block cipher:

Symmetric ciphers are divided into two broad categories : ① Stream cipher,
② Block cipher,

Although the definitions are normally applied to modern ciphers, this categorization also applies to traditional ciphers.

① Stream ciphers:

The basic idea is to generate a key stream $K = \{k_1, k_2, \ldots\}$ and use it to encrypt a plaintext stream $P = \{P_1 P_2 P_3 \ldots\}$ to create a ciphertext stream $C = \{C_1 C_2 C_3, \ldots\}$

i.e,

$P = P_1 P_2 P_3 \ldots$ $\quad C = C_1 C_2 C_3 \ldots$ $\quad K = \{k_1, k_2, k_3, \ldots\}$

$C_1 = E_{k_1}(P_1)$, $C_2 = E_{k_2}(P_2)$, $C_3 = E_{k_3}(P_3) \ldots$

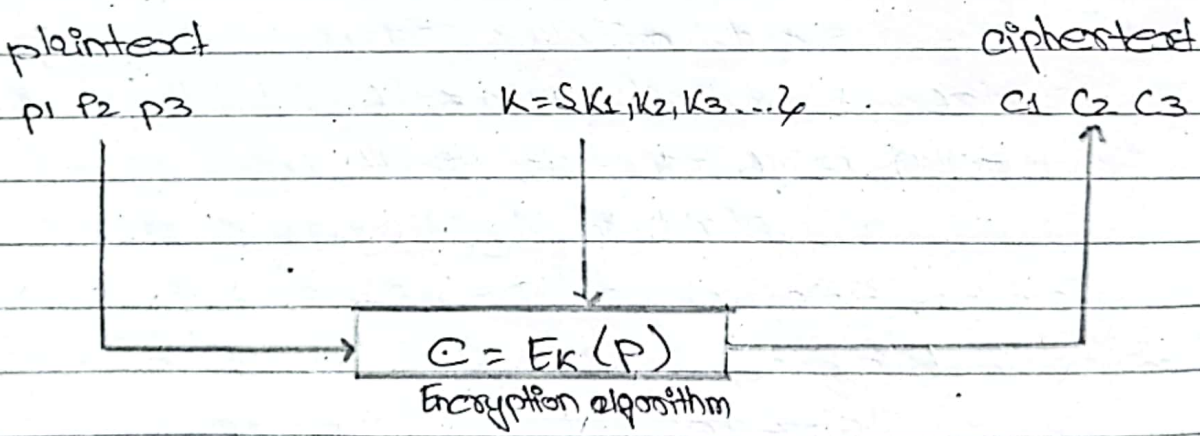


Fig:- Stream cipher.

e.g:- Additive cipher, monoalphabetic substitution ciphers, Vigenere ciphers.

② Block Cipher:

In block cipher, a group of plaintext symbol of size m (m > 1) are encrypted together creating a group of ciphertext of the same size.

A single key is used to encrypt the whole block even if the key is made of multiple values.

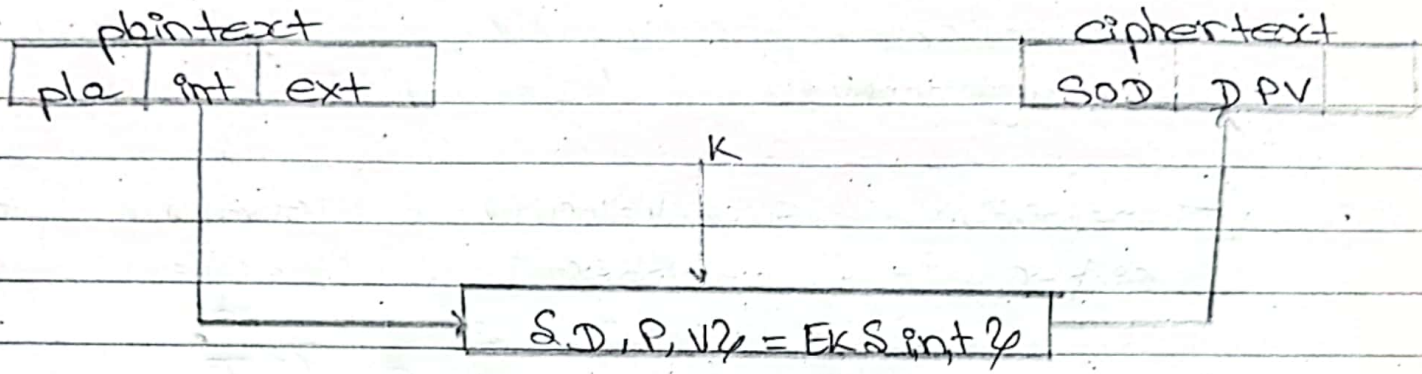


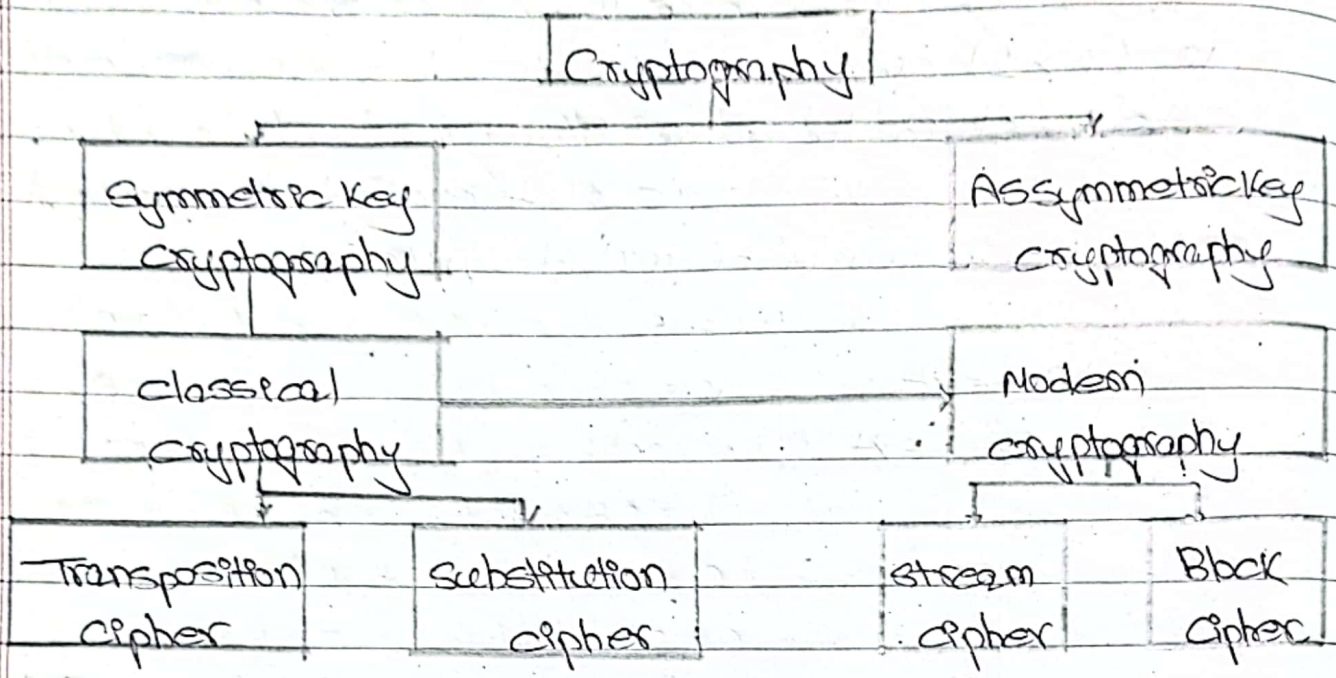
Fig:- Block Cipher

eg:- playfair cipher (block size m = 2), Hill ciphers, polyalphabetic cipher.

Note:- In practice, blocks of plaintext are encrypted individually, but they use a stream of keys to encrypt the whole message block by block. So, the block cipher itself is a stream cipher when looking at the individuals blocks as a single unit.

Page No.
Date : / /

TYPES OF CRYPTOGRAPHY



* A Linear Feedback shift register:

A linear feedback shift register is a special type of recurrence relation where,

① The input bit is a linear function of its previous state,
② The preceding terms are not raised to powers,
③ There are no added constants.

– A LFSR is used to generate a keystream efficiently

for eg:-

$S_n = S_{n-1} + 1$; is not LFSR as there is +1 constant

where,

$S_n = S_{n-1} + S_{n-4} + S_{n-5}$; is a LFSR

- Since, there's no added constant, a seed value of all 0s will produce nothing but 0s. i.e if all the registers fill with all 0s, the output will always be 0s.
- The period of a LFSR with K registers is at most $2^K - 1$.

*example:

Let $S_n = S_{n-1} + S_{n-2} \bmod 2$

Find the first terms of the sequence with $S_0 = 1$, $S_1 = 1$ = Sol

The first term are: S_0, S_1 i.e,

~~S_0~~ 1, 1, 0, 1, 1, 0

$S_2 = S_0 + S_1 \bmod 2$

$= 1 + 1 \bmod 2$

$= 0$

$S_3 = S_1 + S_2 \bmod 2$

$= 1 + 0 \bmod 2$

$= 1$

$S_4 = S_2 + S_3 \bmod 2$

$= 0 + 1 \bmod 2$

$= 1$

$S_5 = S_4 + S_3 \bmod 2$

$= 1 + 1 \bmod 2$

$= 0$

Since, we use 2 registers the period of LFSR is $2^2 - 1 = 3$ i.e,

1, 1, 0, 1, 1, 0, 1, 1, 0,

1st per 2nd per 3rd per The periods gets repeated.

* Example:

Let, $S_n = S_{n-3} + S_{n-4} \bmod 2$

Find period / starting with $S_0 = 1, S_1 = 1, S_2 = 1, S_3 = 1$.

The sequence is

$S_4 = S_1 + S_0 = 1+1 \bmod 2 = 0$

$S_5 = S_2 + S_1 = 1+1 \bmod 2 = 0$

$S_6 = S_3 + S_2 = 1+1 \bmod 2 = 0$

$S_7 = S_4 + S_3 = 0+1 \bmod 2 = 1$

$S_8 = S_5 + S_4 = 0+0 \bmod 2 = 0$

$S_9 = S_6 + S_5 = 0+0 \bmod 2 = 0$

$S_{10} = S_7 + S_6 = 1+0 \bmod 2 = 1$

$S_{11} = S_8 + S_7 = 0+1 \bmod 2 = 1$

$S_{12} = S_9 + S_8 = 0+0 \bmod 2 = 0$

$S_{13} = S_{10} + S_9 = 1+0 \bmod 2 = 1$

$S_{14} = S_{11} + S_{10} = 1+1 \bmod 2 = 0$

$S_{15} = S_{12} + S_{11} = 0+1 \bmod 2 = 1$

$S_{16} = S_{13} + S_{12} = 1+0 \bmod 2 = 1$

$S_{17} = S_{14} + S_{13} = 0+1 \bmod 2 = 1$

$S_{18} = S_{15} + S_{14} = 1+0 \bmod 2 = 1$

$S_{19} = S_{16} + S_{15} = 1+1 \bmod 2 = 0$

Initial term repeated so sequence will be repeat

i.e,

The sequence will be

15th terms till repeat

1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1,

There are 15 terms, & the maximum possible period for a 4th order recurrence relation is $2^4 - 1 = 15$, this is the largest

possible period.

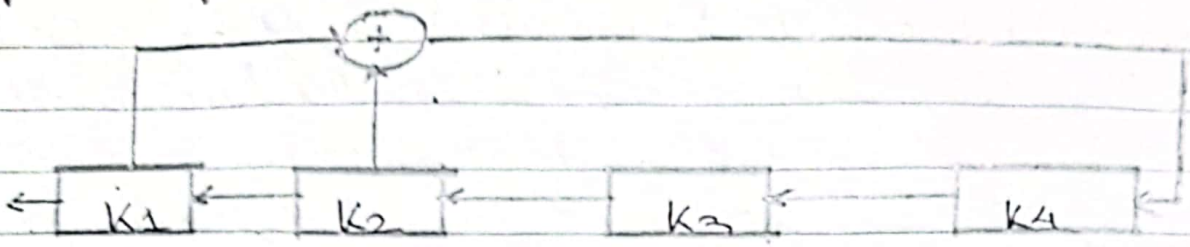


Fig of :- Linear Feedback shift register.

In LSFR, we would use a shift register with m stages. The vector $(K_1, \ldots, K_m)$ would be used to initialize the shift register. At each time unit, the following operations would be performed concurrently:

i) K_1 would be tapped as the next keystream bit
ii) $K_2, \ldots, K_m$ would each be shifted one stage to the left.
iii) The new value of K_m would be computed as per recurrence relation defined.

* Non-linear Feedback Shift Registers (NLFSRs):

Feedback shift registers are the basic components of many keystream generators used in stream ciphers. Each time the system is clocked, the internal state is shifted right, outputting one symbol, and the new left bit is computed from the previous state by a function f.

- The linear Feedback shift registers use a linear Function F, and are the most common in used.
- The Non linear Feedback Shift Registers (NLFSRs) use a non-linear function f. It can be viewed as finite state automata.

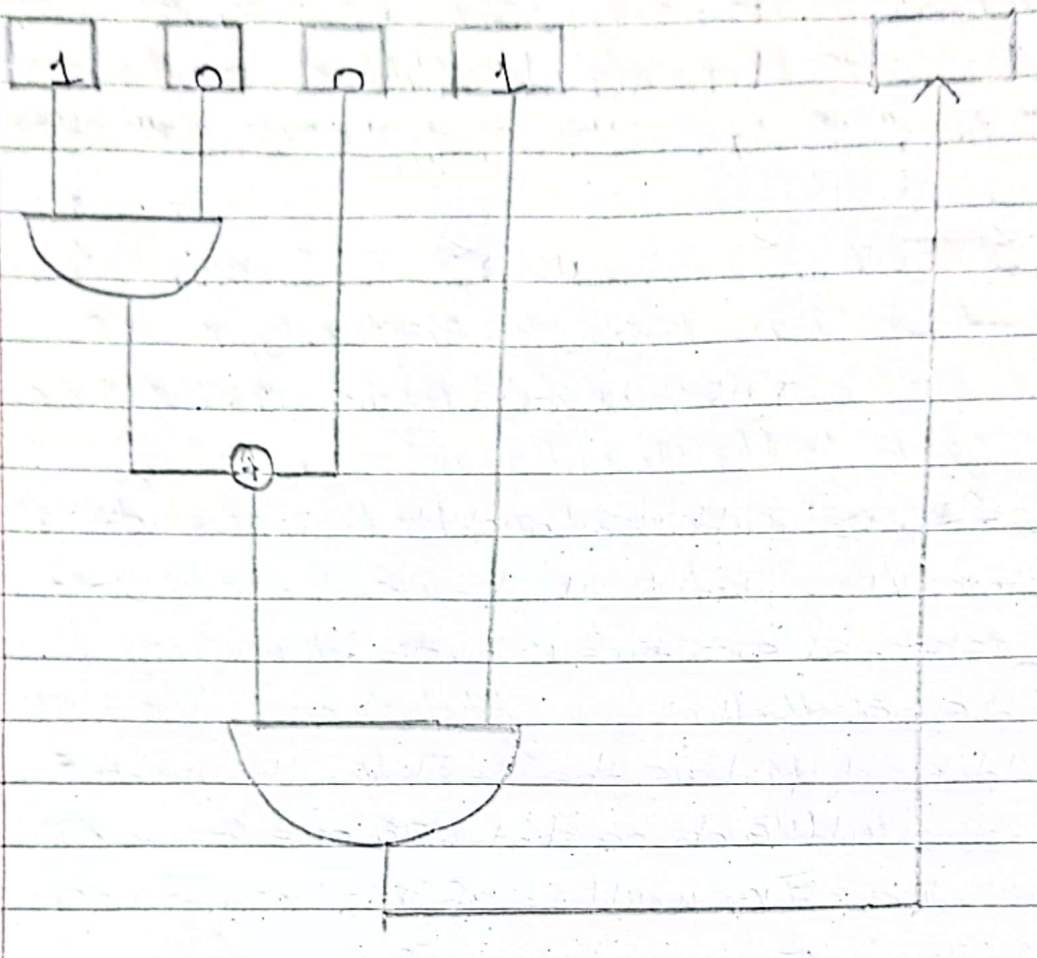


Fig of :- ANLFSR (Non-linear Feedback shift register)

* Cryptanalysis:

- As cryptography is the science and art of creating secret codes, cryptanalysis is the science and art of breaking those codes.
- It is a Analytical process of finding plaintext from cipher text without knowing the key.
- The general assumption that is usually made is that the opponent, Oscar, knows the cryptosystem being used, which is referred usually as Kerchoff's principle

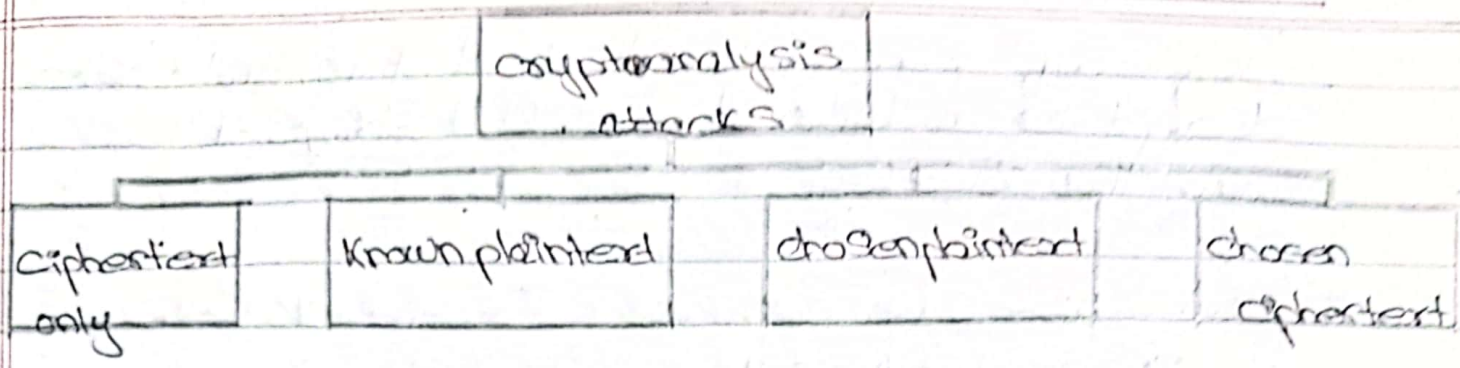


Fig:- cryptanalysis attacks

① Ciphertext only attack:
The opponent possesses a string of ciphertext, y & the algorithm.

② Known plaintext attack:
The opponent possesses a string of plaintext, X, and the corresponding ciphertext, y.

③ choosen plaintext attack:
The opponent has obtained temporary access to the encryption machinery. Hence he can choose a plaintext string, X, and construct the corresponding ciphertext string, y.

④ choosen ciphertext attack:
The opponent has obtained temporary access to the decryption machinery. Hence, he can choose a ciphertext string, y, and construct the corresponding plaintext string, X.

In each case, the objective of the adversary is to determine the key that was used which allow the opponent

to decrypt a specific ciphertext string, and further to decrypt any additional ciphertext strings that are encrypted using the same key.

* Probabilities of occurrence of the 26 letters:

In cryptanalysis, we first consider the weakest type of attack, namely a ~~cipher attack~~ ciphertext-only attack. We also assume that the plaintext string is ordinary English text, without punctuation or "space".

now,

Letters	probability	Letters	probability
E	·127	M	.0.024
T	· 091	W	0·023
A	· 082	F	0·022
O	· 075	G	·020
I	· 070	Y	·020
N	· 067	P	·019
S	· 063	B	·015
H	· 061	V	· 010
R	· 060	K	·008
D	· 043	J	· 002
L	· 040	X	· 001
C	· 028	Q	0·001
U	· 028	Z	·001

Many techniques of cryptanalysis use statistical properties of the English Language. The relative frequency & probabilities of occurence is mentioned above according to descending order, which can be divided into five groups.

as follow:

1) E having probability about 0.120
2) T, A, O, I, N, S, H, R each having probability between 0.06 & 0.09
3) D, L each having probability around 0.04
4) C, U, M, W, F, G, Y, P, B each having probability between 0.015 & 0.028
5) V, K, J, X, Q, Z, each having probability less than 0.01.

It is also useful to consider sequences of two (digrams) and three (Trigrams) consecutive letters.

① Digrams are: (Decreasing order) 30

TH	AN	ES	TO	OU	OR	IT	HI
HE	RE	ST	NT	EA	TI	AR	OF.
IN	ED	EN	HA	NG	IS	TE	
ER	ON	AT	ND	AS	ET	SE	

② Trigram : 12

THE	ERE	WAS
ING	ENT	ETH
AND	THA	FOR
HER	NTH	OTH

Page No.
Date : / /

* Cryptanalysis of the Affine Cipher:

Since, we are considering ciphertext only attack, Suppose Oscar has intercepted the following ciphertext.

eg:- Ciphertext obtained from an Affine cipher

FMXVEDKAPHFERBNDKRXRSREFMORUDSDKDVSH
VUFEDKAPRKDLYEVLRHHRH

We will be using statistical data to solve this problem So, the frequency analysis of this ciphertext is given below:

Letter	Frequency	Letter	Frequency
A	2	N	1
B	1	O	1
C	0	P	2
D	7	Q	0
E	5	R	8
F	4	S	3
G	0	T	0
H	5	U	2
I	0	V	4
J	0	W	0
K	5	X	2
L	2	Y	1
M	2	Z	0

There are total 57 characters in given ciphertext with their no. of repetion represented in above table.

From above table the most frequent ciphertext characters are:

R → 8
D → 7
E → 5
H → 5
K → 5

So, using statistical data, we can assume that

E(e) = R
E(t) = D

since e & t are the two most common letters i.e,

et → RD

i.e,

E(4) = 17
E(19) = 3

we also know that in affine cipher

$C = (PXK1 + K2) \bmod 26$

i.e,

$4K1 + K2 = 17$ — (i)
$19K1 + K2 = 3$ — (ii)

now, we need to find K1 & K2 i.e.

$$\begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix} \begin{bmatrix} K1 \\ K2 \end{bmatrix} = \begin{bmatrix} 17 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} K1 \\ K2 \end{bmatrix} = \begin{bmatrix} 17 \\ 3 \end{bmatrix} \begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix}^{-1}$$

Page No.
Date : / /

now,

$$\begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix}^{-1} = \frac{1}{|4-19|} \begin{bmatrix} 1 & -1 \\ -19 & 4 \end{bmatrix}$$

$$= \frac{1}{-15} \begin{bmatrix} 1 & -1 \\ -19 & 4 \end{bmatrix}$$

$$= \frac{1}{11} \begin{bmatrix} 1 & 25 \\ 7 & 4 \end{bmatrix}$$

$$= 19 \begin{bmatrix} 1 & 25 \\ 7 & 4 \end{bmatrix} \quad (\because 11^{-1} = 19)$$

$$= \begin{bmatrix} 19 & 475 \\ 133 & 76 \end{bmatrix}$$

$$= \begin{bmatrix} 19 & 7 \\ 3 & 24 \end{bmatrix}$$

now,

$$\begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} 17 \\ 3 \end{bmatrix} \begin{bmatrix} 19 & 7 \\ 3 & 24 \end{bmatrix} \begin{bmatrix} 17 \\ 3 \end{bmatrix}$$

$$= \begin{bmatrix} 323+21 \\ 51+72 \end{bmatrix}$$

$$= \begin{bmatrix} 344 \\ 123 \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 6 \\ 19 \end{bmatrix}$$

So, $K_1 = 6$ & $K_2 = 19$

Also, for these keys to be valid $\overset{\text{gcd}}{\uparrow}(K_1, 26) = 1$

$\gcd(6, 26) = 2 > 1$, not valid key

Again, let

$E(e) = R \rightarrow E(4) = 17$

$E(t) = E \rightarrow E(19) = 4$

So, $4K_1 + K_2 = 17$

$19K_1 + K_2 = 4$

again, find valid K_1 & K_2 i.e,

$$\begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix} \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} 17 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 17 \\ 4 \end{bmatrix}$$

now,

$$\begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 19 & 7 \\ 3 & 24 \end{bmatrix}$$

$$\begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} 19 & 7 \\ 3 & 24 \end{bmatrix} \begin{bmatrix} 17 \\ 4 \end{bmatrix} = \begin{bmatrix} 13 \\ 17 \end{bmatrix}$$

again, $\gcd(13, 26) = 2 > 1$ not valid

now again consider,

$$E(e) = R \rightarrow E(4) = 17$$
$$E(t) = K \rightarrow E(19) = 10$$

So, $4K_1 + K_2 = 17$

$19K_1 + K_2 = 10$

So, $\begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix} \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} 17 \\ 10 \end{bmatrix}$

$$\begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 19 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 17 \\ 10 \end{bmatrix}$$

$$= \begin{bmatrix} 19 & 7 \\ 8 & 24 \end{bmatrix} \begin{bmatrix} 17 \\ 10 \end{bmatrix}$$

$$= \begin{bmatrix} 323 + 70 \\ 51 + 240 \end{bmatrix} \bmod 26$$

$$= \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

$K_1 = 3, K_2 = 5$

Also,

$\gcd(K_1, 26)$ i.e $\gcd(3, 26) = 1$

Page No.
Date : / /

So, These are our valid keys i.e, $K_1 = 3$

$K_2 = 5$

now in Affine Cipher for decryption

$$P = (C - K_2) \times K_1^{-1} \bmod 26$$

So, $K_1 = 3$

$K_1^{-1} = 9$

now, we perform decryption for given letters i.e,

$F = 5 = (5 - 5) \times 9 \bmod 26 = 0 = A$

$M = 12 = (12 - 5) \times 9 \bmod 26 = 11 = L$

$X = 23 = (23 - 5) \times 9 \bmod 26 = 6 = G$

$V = 21 = (21 - 5) \times 9 \bmod 26 = 14 = O$

$E = 4 = (4 - 5) \times 9 \bmod 26 = 17 = R$

$D = 3 = (3 - 5) \times 9 \bmod 26 = 8 = I$

$K = 10 = (10 - 5) \times 9 \bmod 26 = 19 = T$

$A = 0 = (0 - 5) \times 9 \bmod 26 = 7 = H$

$P = 15 = (15 - 5) \times 9 \bmod 26 = 12 = M$

Since, we are getting meaningful String of English, this confirm the validity of (3,5) as keys similarly continue to decrypt for all letters.

Page No.
Date : / /

* Cryptanalysis of the substitution cipher.

- Substitution cipher being a monoalphabetic cipher we use frequency analysis method i.e. guessing & using intuitions for guessing or decipher the ciphertext.
- The possible keys or the key space for the substitution cipher is $26! = 4 \times 10^{26} = 2^{88}$ which is impractical so, use Frequency Analysis.
- The possibility or Accuracy solely depends on user who tries to decrypt the ciphertext mainly depending on his guesses.

Example: Ciphertext obtained from a substitution cipher.

YIFQFMZRWQFYVECFNDZPCVMRZWNNDZVEJBTXCD
UNJ NDIFEFNDZCDMQZKCEYFCJMYRNCWJCSZREX
CHZUNNMXZNZUCDRJXYYSMRTMEYIFZWDYVZVYFZUM
RZCRWNZDZJJXZWGCHSMRNMDHNCMFQCHZJMXJZ
IEJYUCFWDJNZDTR.

The Frequency Analysis of this ciphertext is given as:

Letter	Frequency		Letter	Frequency	
A	0		I	5	15
B	1		J	11	6
C	15	3	K	1	
D	13	4	L	0	
E	7	11	M	16	2
F	11	5	N	9	9
G	1		O	0	
H	4		P	1	

Letter	Frequency	
Q	4	
R	10	7
S	3	
T	2	
U	5	14
V	5	15
W	8	10
X	6	12
Y	10	8
Z	20	1

since, Z occcers significantly the most we can guess that $dk(z) = e$.

but there is confusion as the remaining ciphertext characters that occcers at least ten times each are:

C, D, F, J, M, R, Y.

we might guess these letter are encryptions of (a subset of) & t, a, o, i, n, s, h, r & but frequencies doesn't vary enough so, less chances of decryption.

At this stage we use digrams i.e especially those of the form _Z or Z_ because we know that $dk(z) = e$.

ie. MZ → 1 ZC → 1+1 → 2 ZN → 1

ZR → 1+1 = 2 QZ → 1 NZ → 1+1+1 = (3)*

DZ → 1+1+1+1 = (4) ZK → 1 FZ → 1+1 = 2

ZP → 1 SZ → 1 VZ → 1

RZ → 1+1 = 2 HZ → 1+1 → 2 ZD → 1+1 = 2

ZW → 1+1+1+1 → (4) ZV → 1+1+1 → (3)* ZJ → 1+1 = 2

ZV → 1+1 = 2 XZ → 1+1 → 2 JZ → 1

Page No.
Date: / /

We find:

① DZ & ZW = 4
② NZ & ZU = 3
③ RZ, HZ, XZ, FZ, ZR, ZV, ZC, ZD & ZJ = 2

Since, ZW occurs four times & WZ not at all, the possible digrams are: Z → E
W → ?

ES
EA
ED
EN
ER
ET

also, W occurs less often than many other characters.
i.e, W occurs less than (C, D, F, J, M, R, Y)
(t, a, i, o, n, s, h, r)

we, guess dk(W) = d.

now, for, DZ occurs four times & ZD occurs twice, we would think that dk(D) ∈ {r, s, t, h}

i.e, possibilities are:- NE
RE
TE
SE

but it is not clear which of 4 possibilities is the correct one.

Page No.
Date : / /

Until now, we assume $d_K(z) = e$

$d_K(w) = d$

With these, when we look back at the ciphertext, we notice ZRW occuring in the beginning together & RW occurs again together letter. From which we guess common diagrams with d at end are : Ed & Nd (possible digram)

So, we assume $d_K(R) = n$

exchange ciphertext with assumed plaintext we get,

```
      end          e     ned    e
YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ
      e    e          n  d  en   e    e
NDIFEFNDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ
e   n    n     ed   e    e  ne  nd  e  e
NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUNRZCRWNZDZJJ
e   n          e    Rd        d   e  n
XZWGRCHSNRNMDMNCMFQCHZJMXJZWIEJYUCFWDJNZDIR
```

Now, NZ occcers 8 times & ZN not at all the possible digram are {HE, RE, TE, SE} So, we assume

$d_K(N) = h$,

So, if our assumption is correct then the segment of plaintext ne-nd he suggests $d_K(C) = a$. (guess)

Until we assume,

$d_K(z) = e$

$d_K(w) = d$

$d_K(R) = n$

$d_K(n) = h$

$d_K(c) = a$ we get the ciphertext as.

Page No.
Date : / /

end a e a nedh e a
YIFQFNZRWQFYUECFNDZPCVNRZWNNDZVEJBTXCDDUNJ
h ea e a a nhad a en a e h e
NJIFEFNDZCJNQZKCEYFCJNYRNCWJCSZREXCHZUNNXZ
he a n n ed e e neandhe e
NZUCDRJXYYSNRTNEYIFZWDYVZVYFZUNRZCRWNZDZJJ
ed a nh ha a e ed a d he r
XZWGCHSNRNNDHNCMFQCHZJNXJZWIEJYUCFWDJNZDIR

(10)

now, we guess plaintext for letter M, which is most common ciphertext character.

We take RNN from ciphertext where, RNN decrypts to nh. It suggests h_ begins a word, so, we have 3 possibilities i.e, HI, HE, HA
we have already accounted for a & e, so,

$$dk(M) = I$$

Next, we try to determine the which letter is the encryption of O. Since O is a common plaintext character we have the most frequent ciphertext character i.e

C, D, F, J, M, R, Y

where, C, M, R are already accounted so, the option are,

D, F, J, Y.

We choose Y for encryption of O. i.e. if we choose F or J it would gives a long string of vowels which not be valid i.e.

CFM → aOi or CJM → aOi

So, $d_K(Y) = 0$.

now, The remaining 3 most frequent ciphertext letters are D, F, J. which consecutive could decrypt to r, s, t in some order.

Two occurrences of trigram NMD suggests that $d_K(D) = S$, giving NMD → hiS in the plaintext.
which also proves our earlier hypothesis i.e, $d_K(D) \in \{r, s, t\}$.
The segment of ciphertext HNCMF could be an encryption for chair

$d_K(HNCMF) \rightarrow$ chair

which would give:

$d_K(F) = r$

$d_K(H) = c$

$d_K(J) = t$ (only remaining possibility).

we have, until now,

$d_K(z) = e$

$d_K(W) = d$

$d_K(R) = n$

$d_K(N) = h$

$d_K(C) = a$

$d_K(M) = i$

$d_K(Y) = 0$

$d_K(D) = s$

$d_K(H) = c$

$d_K(F) = r$

$d_K(J) = t$

Page No.
Date : / /

ndw, the ciphertext look like:

o_r_riend_ro__arise_a_inedhise_t___ess_it
YIFQFMZRWQFYVECFMDZPCVMRZWNMDZVEJBTXCDDUMJ

hs_r_riseasi_e_a_orationhadta_en_ace_hi_e
NDIFEFMDZCDMQZKCEYFCJMYRNCWJCSZREXCHZUNMXZ

he_asnt_oo_in_i_o_redsoo_e_ore_ineandhese
NZUCDRJXYYSMRTMEY IFZWDYVZVYFZUMRZCRWNZDZ

_ed_ac_inhischair_aceti_ted__to_ardsthes_
XZWGCHSMRNMDHNCMFQCHZJMXJZWIEJYUCFWDJNZDR

- now, the remaining decryption of ciphertext text letter is similar to fill in the gaps with appropriate english ~~let word~~

The complete decryption is as follow:

our friend from paris examined his empty glass with surprise, as if ~~a~~ evaporation had taken place while he wasn't looking. I poured some more wine ~~and~~ he settled back in his chair, face ~~tilt~~ tilted up towards the sun.

Cryptanalysis of the Vigenere cipher:

Vigenere cipher is a polyalphabetic cipher where each unique alphabetic character can be mapped to different alphabetic characters.

Cryptanalysis of the Vigenere cipher starts from determining the keyword length, which is denoted by m. So, the possible key size in polyalphabetic cipher is given by:

$$p = c = K = (Z_{26})^m.$$

which provides a very large number of keys. So, it is more complex than monoalphabetic cipher.

There are couple of techniques that can be employed. The first of these is the so-called Kasiski test & the second uses the index of coincidence.

Cryptanalysis of the Hill cipher:

The Hill cipher can be difficult to break with a ciphertext-only attack.

But it can be easily ~~decrypt~~ be cryptanalysed to a known plaintext attack.

We assume that, the opponent has determined the value of m (key) being used. Also, we assume that we know the proper plaintext-ciphertexts pairs too.

Let's see an example:

Suppose the plaintext 'friday' is encrypted using Hill cipher with $m = 2$, to give ciphertext PQCFKU.

Page No.
Date : / /

We have,

$e_K(F, r) = (P, Q)$ i.e, $e_K(5, 17) = (15, 16)$

$e_K(i, d) = (C, F)$ i.e, $e_K(8, 3) = (2, 5)$

$e_K(a, y) = (K, V)$. i.e, $e_K(0, 24) = (10, 20)$

Taking first two plaintext-ciphertext pairs, we get the matrix equation as;

$$\begin{bmatrix} 15 & 16 \\ 2 & 5 \end{bmatrix} = \begin{bmatrix} 5 & 17 \\ 8 & 3 \end{bmatrix} K$$

now,

$$P^{-1} = \begin{bmatrix} 5 & 17 \\ 8 & 3 \end{bmatrix}$$

$$= \frac{1}{\text{det. P}} \begin{bmatrix} 3 & -17 \\ -8 & 5 \end{bmatrix}$$

$$= \frac{1}{(15 - 136)} \begin{bmatrix} 3 & 9 \\ 18 & 5 \end{bmatrix}$$

$$= \frac{1}{-121} \begin{bmatrix} 3 & 9 \\ 18 & 5 \end{bmatrix}$$

$$= \frac{1}{9} \begin{bmatrix} 3 & 9 \\ 18 & 5 \end{bmatrix}$$

$$= 9^{-1} \begin{bmatrix} 3 & 9 \\ 18 & 5 \end{bmatrix}$$

$$= 3 \begin{bmatrix} 3 & 9 \\ 18 & 5 \end{bmatrix} \bmod 26 \quad (\because \text{Multiplicative inverse of } 9 \text{ is } 3)$$

$$= \begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix}$$

now,

$$K = \begin{bmatrix} 9 & 1 \\ 2 & 15 \end{bmatrix} \begin{bmatrix} 15 & 16 \\ 2 & 5 \end{bmatrix}$$

$$= \begin{bmatrix} 135+2 & 144+5 \\ 30+30 & 32+75 \end{bmatrix}$$

$$= \begin{bmatrix} 7 & 19 \\ 8 & 3 \end{bmatrix}$$

now, to check the validity of key use 3rd plaintext-ciphertext pair i.e,

$$e_K(0, 24) = (10, 20)$$

i.e,

$$(0, 24) \begin{bmatrix} 7 & 19 \\ 8 & 3 \end{bmatrix} = (10, 20)$$

$$\Rightarrow (0+192 \quad 0+72) \bmod 26 = (10, 20)$$

Page No.
Date : / /

Or, $(10, 20) = (10, 20)$

So, our key $\begin{bmatrix} 7 & 19 \\ 8 & 3 \end{bmatrix}$ is a valid key.

– We already mentioned that Hill cipher can be difficult to break with a cipher-text only attack although let's see an Example of cipher-text only attack.

i.e.

Given Cyphertext to decrypt is

LMQETXYEAGTXCTUIEWNCTXLZEWUAISPZYVAPE WLMGTQWYAXFTCJMSQCADAGTXLNDXNXSNPJQSYVAPR IQSMHNOCVAXFV

Suppose M=2. Break the cipher-text into block of length of two letters diagrams. Each such diagrams are the encryption of a plaintext diagrams. I assume it in the encryption of a common diagram for Example TH or ST.

= Sol

Given, m=2, we break the given ciphertext into blocks of length, the two-digram and their frequencies in the ciphertext are as follows:

Digrams	Frequency	Digrams	Frequency
LM	3	EW	3
QE	1	NC	1
TX	4	LZ	1
YE	1	UA	1
AG	2	IS	1
CT	1	PZ	1
UI	1	YV	2

Digrams	Frequency	Digrams	Frequency
AP	2	NX	1
GQ	1	BN	1
WY	1	PJ	1
AX	1	QS	2
FT	1	RJ	1
CJ	1	MH	1
MS	1	NO	1
QC	1	CV	1
AD	1	AX	1
DX	1	FV	1

From above table we can assume that the most common ciphertext digram 'TX' corresponds to the common plaintext digram 'TH'

We need to determine the plaintext so, firstly we need to determine the key inverse.

Let us consider 2×2 matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ be the inverse of the key.

and, TX corresponds to TH then,

$$\begin{bmatrix} T \\ X \end{bmatrix} = \begin{bmatrix} 19 \\ 23 \end{bmatrix}$$

$$\begin{bmatrix} T \\ H \end{bmatrix} = \begin{bmatrix} 19 \\ 7 \end{bmatrix}$$

then,

$$\begin{bmatrix} a & b \\ c & 1 \end{bmatrix} \begin{bmatrix} 19 \\ 23 \end{bmatrix} = \begin{bmatrix} 19 \\ 7 \end{bmatrix}$$

Page No.
Date : / /

Converting the above matrix into the linear equation. We get

$$19a + 23b = 19 \quad \text{—(I)}$$
$$19c + 23d = 7 \quad \text{—(II)}$$

To solve the above equation, let's assume that the next common cipher diagram 'LM' corresponds to diagram 'he'

$$\begin{bmatrix} L \\ M \end{bmatrix} = \begin{bmatrix} 11 \\ 12 \end{bmatrix}$$

$$\begin{bmatrix} H \\ E \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \end{bmatrix}$$

then,

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 11 \\ 12 \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \end{bmatrix}$$

$$11a + 12b = 7 \quad \text{—(III)}$$
$$11c + 12d = 4 \quad \text{—(IV)}$$

now,

solving the eqn (I), (II), (III) & (IV) to find the value of a, b, c, d.

$$19a + 23b = 19 \quad \times 12 \quad \text{Mod } 26.$$
$$11a + 12b = 7 \quad \times 23$$

$$20a + 16b = 20$$
$$19a + 16b = 5$$
$$\underline{\quad - \quad - \quad -}$$
$$a = 15.$$

Page No.
Date : / /

again. for finding the value of b.

$19a + 23b = 19 \quad \times 11$

$11a + 12b = 7 \quad \times 19$

$a + 19b = 1$

$a + 20b = 3$

$-b = -2$

$\therefore b = 2$

$19c + 23d = 7$

$11c + 12d = 4$

for finding the value of c i.e.

$19c + 23d = 7 \quad \times 12 \quad \text{mod } 26.$

$11c + 12d = 4 \quad \times 23$

$20c + 16d = 6$

$19c + 16d = 14$

$c = -8 \quad \text{mod } 26$

$= 18$

for finding the value of d we get

$19c + 23d = 7 \quad \times 11$

$11c + 12d = 4 \quad \times 19$

$c + 19d = 25$

$c + 20d = 24$

$-d = 1$

Page No.
Date : / /

$$d = -1 \mod 26$$
$$= 25$$

So, $a = 15, b = 2, c = 18, d = 25$

i.e, Key inverse $= \begin{bmatrix} 15 & 2 \\ 18 & 25 \end{bmatrix}$

now, Let decrypt using the above matrix.

$$LM = \begin{bmatrix} 15 & 2 \\ 18 & 25 \end{bmatrix} \begin{bmatrix} 11 \\ 12 \end{bmatrix} = \begin{bmatrix} 189 \\ 498 \end{bmatrix} \mod 26 = \begin{bmatrix} 7 \\ 4 \end{bmatrix} = he$$

$$QE = \begin{bmatrix} 15 & 2 \\ 18 & 25 \end{bmatrix} \begin{bmatrix} 16 \\ 4 \end{bmatrix} = \begin{bmatrix} 248 \\ 388 \end{bmatrix} \mod 26 = \begin{bmatrix} 14 \\ 24 \end{bmatrix} = OY$$

$$TX = \begin{bmatrix} 15 & 2 \\ 18 & 25 \end{bmatrix} \begin{bmatrix} 19 \\ 23 \end{bmatrix} = \begin{bmatrix} 331 \\ 917 \end{bmatrix} \mod 26 = \begin{bmatrix} 19 \\ 7 \end{bmatrix} = TH$$

In similar fashion we use the above key & decrypt the ciphertext. The resulting plaintext will be.

h eoy them muth qreo oysy thhray owa cwpl mvela yhe 30 oiedit whiq kaq x muth hen fhd k zih q k mvell m q k mbpm upudnx.

Since, the resulting plaintext has no meaning, we try to find another key inverse.

Page No.
Date : / /

We assume that the ciphertext digram 'TX' corresponds to the plaintext diagram 'in' & 'LM' corresponds to 'th' now

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 19 \\ 23 \end{bmatrix} = \begin{bmatrix} 8 \\ 13 \end{bmatrix}$$

$$19a + 23b = 8 \quad — (V)$$
$$19c + 23d = 13 \quad — (VI)$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 11 \\ 12 \end{bmatrix} = \begin{bmatrix} 19 \\ 7 \end{bmatrix}$$

$$11a + 12b = 19 \quad — (VII)$$
$$11c + 12d = 7 \quad — (VIII)$$

Solving above eqn (V), (VI), (VII) & (VIII) we get,

$$a = 23, \ b = 13, \ c = 21, \ d = 16$$

(Similarly as we did in first)

The inverse key is $\begin{bmatrix} 23 & 13 \\ 21 & 16 \end{bmatrix}$

Also,

$$\det. = 368 - 273 \mod 26$$
$$= 17.$$

now, decrypt using the above key

(1) $LM = \begin{bmatrix} 23 & 13 \\ 21 & 16 \end{bmatrix} \begin{bmatrix} 11 \\ 12 \end{bmatrix} = \begin{bmatrix} 409 \\ 423 \end{bmatrix} \mod 26 = \begin{bmatrix} 19 \\ 7 \end{bmatrix} = TH$

(2) $QF = \begin{bmatrix} 23 & 13 \\ 21 & 16 \end{bmatrix} \begin{bmatrix} 16 \\ 4 \end{bmatrix} = \begin{bmatrix} 420 \\ 400 \end{bmatrix} \mod 26 = \begin{bmatrix} 4 \\ 10 \end{bmatrix} = EK$

(iii) $TX = \begin{bmatrix} 23 & 13 \\ 21 & 16 \end{bmatrix} \begin{bmatrix} 19 \\ 23 \end{bmatrix} = \begin{bmatrix} 736 \\ 767 \end{bmatrix} \bmod 26 = \begin{bmatrix} 8 \\ 13 \end{bmatrix} = in$

Similarly, use inverse key to decrypt the ciphertext, finally we get the plaintext as:

'The King was in his counting house counting out his money The queen was in the parlour eating bread and honey.'

which is a meaningful sentence.

* Some Important Terminology:

① Computational Security:

Some ciphers are easy to crack and others are very difficult. In practise, perfect security is impossible to achieve. Most ciphers used today rely on computational security. This means that they rely on the fact that there's no computer system powerful enough to crack the cipher in a reasonable amount of time.

It also refers to the computational effort required to break a cryptosystem. We might define a cryptosystem to be computational secure if while using the best algorithm for breaking the cryptosystem requires at least N operations, where N is a very large number.

② Provable Security:

In Provable Security, we show that if the cryptosystem can be broken in some specific way, then it would be

possible to efficiently solve some well-studied problem that is thought to be difficult. For example, it may be possible to prove a statement of a type "a given cryptosystem is secure if a given integer n cannot be factored".

But it must be understood that this approach only provides a proof of security relative to some other problem, not an absolute proof of security.
i.e. Similar to proving that a problem is NP-complete i.e, it proves that the given problem is at as difficult as any other NP-complete problem, but it doesnot provide an absolute proof of the computational difficulty of the problem.

Unconditional security:

A cryptosystem is defined to be unconditionally secure if it cannot be broken, even with infinite computational resources. i.e, there is no bound on the amount of computation.

Perfect Secrecy:

Perfect Secrecy means that the Ciphertext Conveys no information about content of the plaintext.
i.e, no matter how much Ciphertext is available, no information about plaintext or key can be found.
Probability distribution of the possible Plaintext is independent of ciphertext.

Page No.
Date : / /

* Properties of Perfect Secrecy:

① For a ciphertext, if given choice of two plaintext, out of which one is real plaintext, it should be impossible to find the real plaintext.

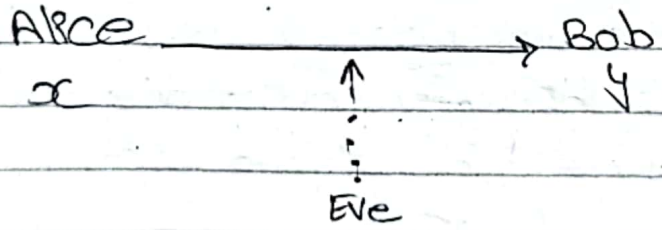
② If a plaintext is encrypted with a key exactly ones. i.e, the size of plaintext is equal with the size of key.
e.g:- one-time padding is an example of perfectly secret cipher.

```
  00110101  (plain-text)
⊕ 11100011  (one-time secret key)
  11010110  (cipher-text)

  11010110  (cipher-text)
⊕ 11100011  (key)
  00110101  (plain-text)
```

- This idea is made precise by formulating it in terms of the probability distribution, as:

Definition: A cryptosystem has perfect secrecy if $Pr[x|y] = Pr[x]$ for all $x \in P, y \in C$
Idea: Eve can obtain no information about the plainte by observing the ciphertext.



Page No.
Date: / /

Problem: Let the message space be $M = \{a, b\}$ with probability distribution $Pr[a] = 3/4$, $Pr[b] = 1/4$. Let the key space be $K = \{K1, K2, K3\}$ with probability distribution $Pr[K1] = 1/2$, $Pr[K2] = Pr[K3] = 1/4$. Let ciphertext space $C = \{1, 2, 3, 4\}$ was produced using an encryption scheme given in the below matrix. for example $Enc(K1, a) = 1$ or $Dec(K2, 3) = b$.

	a	b
K1	1	2
K2	2	3
K3	3	4

prove or disprove that this encryption schemes satisfies the definition of perfect secrecy.

Sol

Let See an example, where, $m = a$ & $K1 = 1$ then according to Baye's theorem

$$p(m=a \mid C=1) = \frac{p(C=1 \mid m=a)\, p(m=a)}{P(C=1)}$$

So,

$$P(C=1) = p(K=K1) \cdot p(m=a)$$
$$= \frac{1}{2} \cdot \frac{3}{4} \quad \text{(given)}$$
$$= \frac{3}{8}$$

now, $p(C=1 \mid m=a) = p[K=K1]$

$$= \frac{1}{2}$$

So, $p(m=a \mid C=1) = \frac{p(C=1 \mid m=a)\, p(m=a)}{P(C=1)}$

$$= \frac{\frac{1}{2} \cdot \frac{3}{4}}{\frac{3}{8}}$$

$$= 1$$

but we have given that $P(m=a) = \frac{3}{4}$

$\frac{3}{4} \neq 1$ So we disprove the scheme for perfect secrecy.

* **Theorem:** Shift Cipher provides perfect Secrecy.

Suppose the 26 keys in the shift cipher are used with equal probability $\frac{1}{26}$. Then for any plaintext probability distribution, the shift cipher has perfect secrecy.

Proof: Recall that $P = C = K = Z_{26}$, and for $0 \leq K \leq 25$, the encryption rule e_K is defined as $e_K(x) = (x+K) \bmod 26$ $(x \in Z_{26})$

First, we compute the probability distribution on C. Let $y \in Z_{26}$ then.

$$Pr[x|y] = \frac{Pr[y|x] \cdot Pr[x]}{Pr[y]}$$

So,

① $$Pr[y] = Pr[y=y] = \sum_{K \in Z_{26}} Pr[K=K] \, Pr[x = d_K(y)]$$

$$= \sum_{K \in Z_{26}} \frac{1}{26} \, Pr[x = y-K]$$

$$= \frac{1}{26} \sum_{K \in Z_{26}} Pr[x = y - K]$$

now, for fixed y, the values of $(y-K) \bmod 26$ comprise a permutation of Z_{26}.

Hence,

$$\sum_{K \in Z_{26}} Pr[x = y-K] = Pr[x = x]$$
$$= 1$$

So,

$$Pr[y] = \frac{1}{26} \cdot 1$$
$$= \frac{1}{26}$$

now,

② $Pr[y|x] = Pr[K = (y-x) \bmod 26]$
$$= \frac{1}{26}$$

This is true because, for every x, y, the unique key K such that $e_K(x) = y$ is $K = (y-x) \bmod 26$.

now,

by Bayes' theorem,

$$Pr[x|y] = \frac{Pr[x] \cdot Pr[y|x]}{Pr[y]}$$
$$= \frac{Pr[x] \cdot \frac{1}{26}}{\frac{1}{26}}$$
$$= Pr[x]$$

We have perfect secrecy.

Hence, the shift cipher is "unbreakable" provided that a new random key is used to encrypt every plaintext character

Page No.
Date : / /

* Entropy: (Randomness)

Definition: Suppose x is a discrete random variable which takes on value from a finite set X. Then, the entropy of the random variable x is defined as:

$$H(x) = -\sum_{x \in X} Pr[x] \log_2 Pr[x].$$

Page No.
Date : / /

* Modern CryptoSystem: (Stalling slide - Chap-3)

- Before entering Modern CryptoSystem we need to understand the Concept of Block and Stream Cipher and we use Product Cipher which is the combination of Substitution and transposition Ciphers.
- We have already describe about Block and stream ciphers (refer to 'initial definition' in notebook). Let's see simple definition though:

i) **Block Cipher:** Block Ciphers process messages in blocks, each of which is then encrypted & decrypted.
e.g:- blocks of 64-bits or more.

ii) **Stream Cipher:** Stream Ciphers process messages a bit or byte at a time when encryption & decryption.

- Many Current Ciphers are block Ciphers,
- Because it provides broader range of applications.

* Modern Block Ciphers:

- One of the most widely used types of Cryptographic algorithms.
- It provides Secrecy / authentication Services
- To illustrate block cipher design principles we focus on DES (Data Encryption Standard) which is the ~~most~~ Symmetric block ciphers.
- Most Symmetric bbck Ciphers are based on a Feistel Cipher structure

Page No.
Date : / /

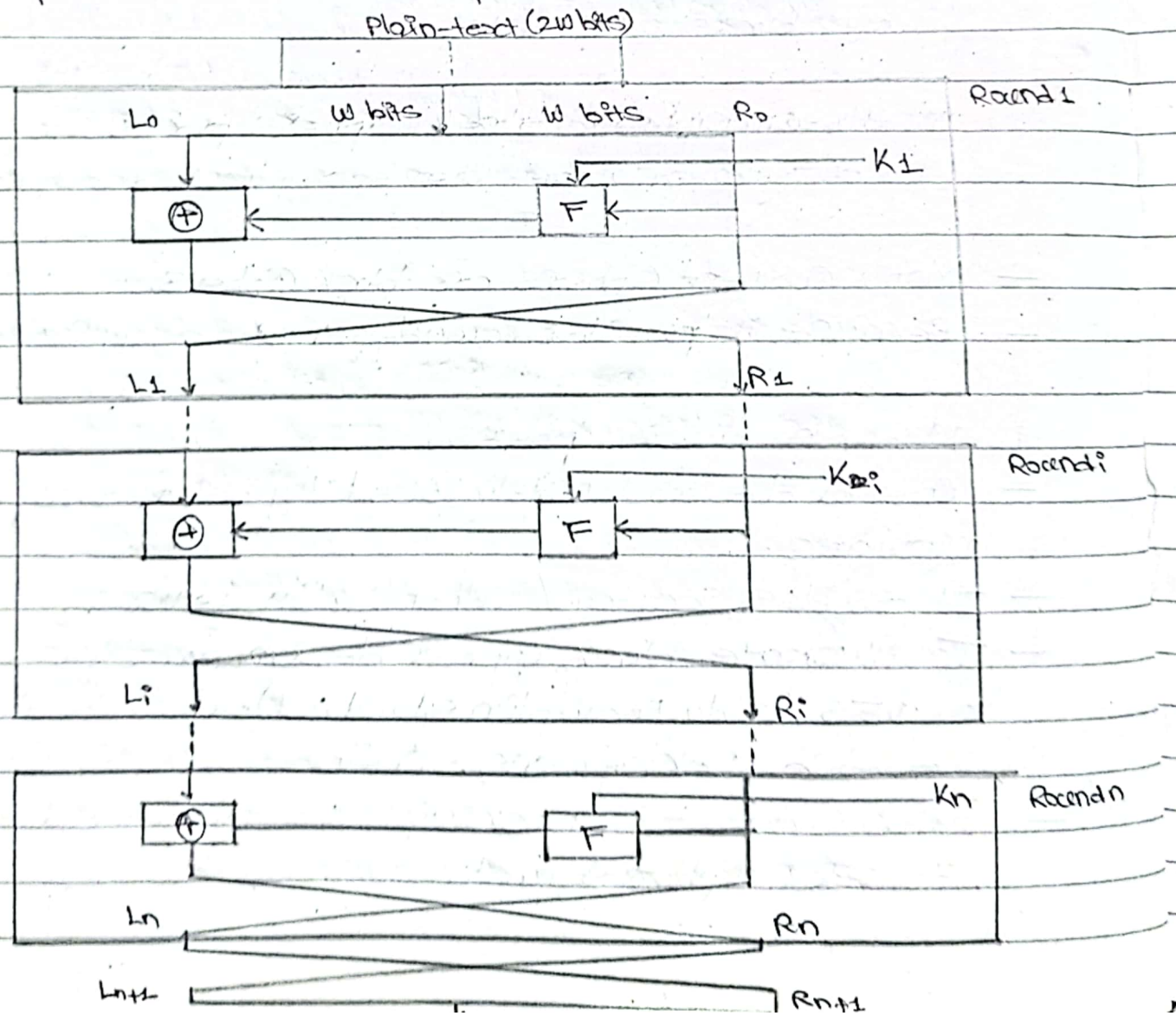
* Feistel cipher Structure:

- Horst Feistel devised the Feistel cipher based on concept of invertible product cipher.

How it works:

① Partitions input block into two halves:
- Process through multiple rounds which
- perform a substitution on left data half
- based on round function of right half & subkey
- Then have permutation swapping halves.

- It implements Shanon's S-P net concept.



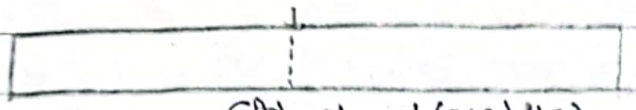


Fig of :- block diagram of Feistel Structure.

Note:- Feistel uses even bit size so, we use 2w bits to makes bit size even.

Feistel cipher Design Elements:

The exact realization of a Feistel network depends on the choice of the following parameters & design features.

Block size
Key size
number of rounds.
Subkey generation algorithms.
round function
Hardware & software.
Ease of Analysis

Decryption is similar as encryption but only difference is the application of sub-key is in reverse order. The block diagram is in slide (Stalling slides. chap-3. S.n→12).

DES (Data Encryption Standard):

It is the most widely used block cipher in world. Adopted in 1977 by National Bureau of Statistics (NBS) (now National Institute of Standards and Technology (NIST) as Federal Information processing standard (FIPS) PUB 46. encrypts 64-bit data using 64-bit key.

- has widespread use,
- It has been considerable controversy over its security and new version are released as DES-II, DES-III.

Block size : 64-bit
Key-size :- 64-bit (56-bits)
Rounds :- 16 round.

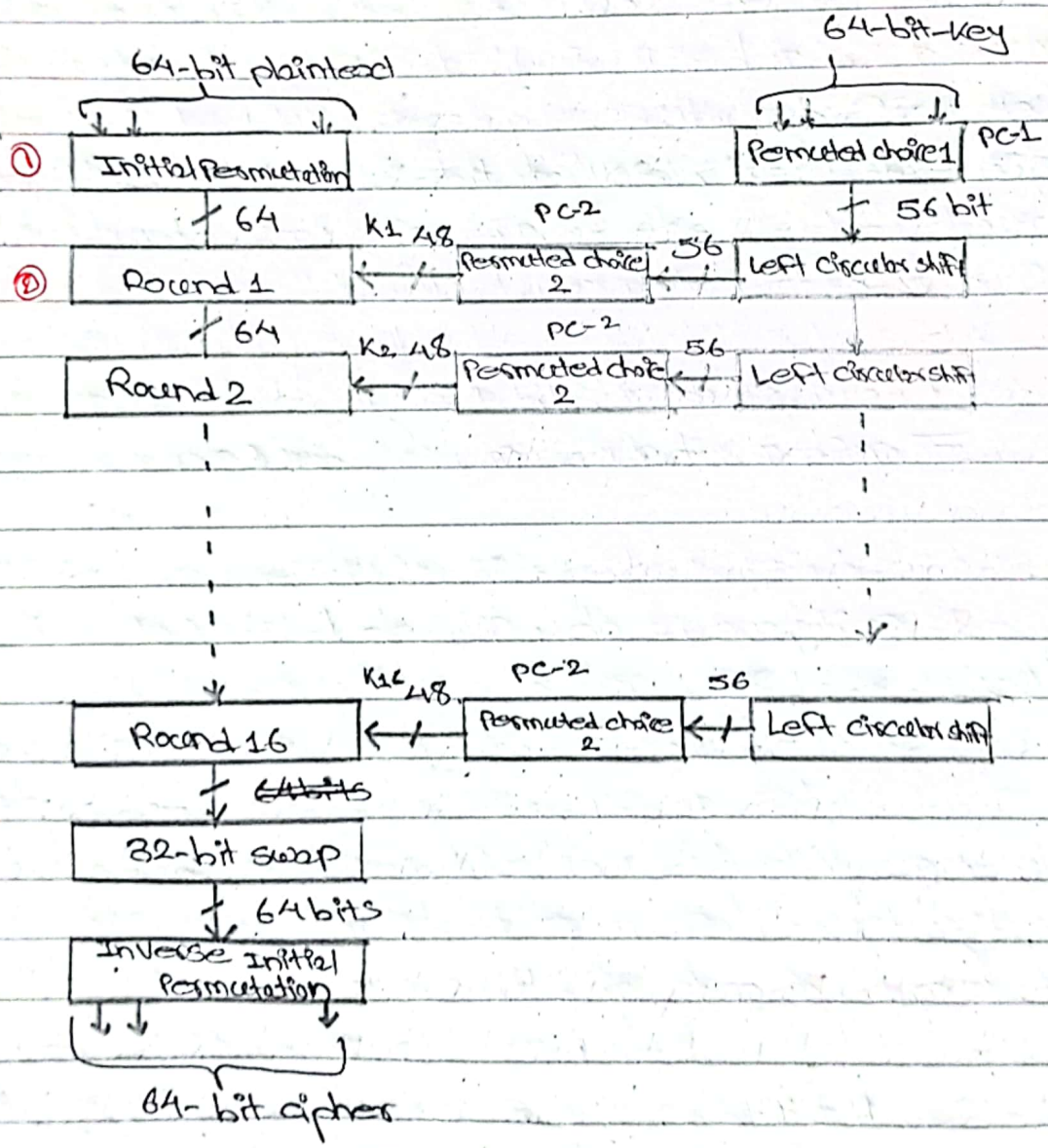


Fig of :- Block diagram of DES

(DES supplementary material wikipedia)

Page No.
Date : / /

① Initial Permutation:

- It is the first step of the data computation.
- Initial permutation reorders or swiftes the input data bits
- Even bits to upper half, odd bits of lower half.
- Easy in h/w, i.e. standard table (refer to DES supplementary materials in wikipedia).

② DES Round Structure:

- It follows the Feistel structures so,
- It divides the 64-bit into two 32-bit Left & Right halves.
- According to Feistel structure:

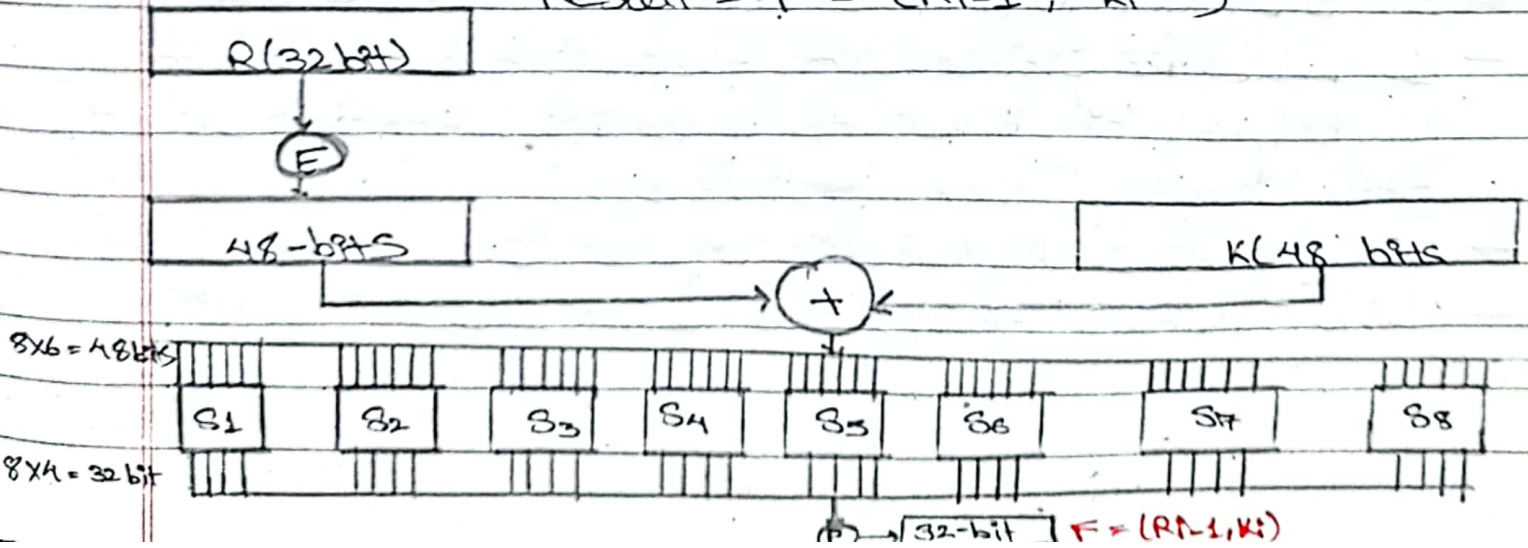
$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

- F takes 32-bit Right half and 48-bit sub key:
  - i) Expands R to 48-bits using permutation or Expansion Function (E) (refer to wikipedia).
  - ii) Adds to subkey using XOR
  - iii) passes through 8 S-boxes to get 32-bit result
  - iv) finally permutes using 32-bit perm P (permutation) (refer to wikipedia)

32-bit 48-bit

$$\text{Result} = F = (R_{i-1}, K_i)$$



Page No.
Date : / /

Note: 6 bits that goes through S-box, 1st & 6th bit represents row (2-5) remaining middle bit represent column. S-box table (refers to wikipedia Substitution boxes (S-boxes)) gives 4 bits i.e 8 S-box = 8 × 4 = 32 bits result of Function.

③ DES Key Schedule:
- It helps to form Sub-keys used in each round.
- The initial 64-bit keys passes through PC-1 to privid provide 56-bits of two 28-bit halves (PC-1 table wikipedia)
- 16 Rounds consisting of:
  ① rotating each half separately either 1 or 2 places depending on the key rotation schedule K (wikipedia)
  ② now, selecting 24-bits from each half & permuting them by PC2 for use in round function F.

* DES Decryption:
Due to Feistel design, do encryption steps again using subkeys in reverse order (SK16 - - - SK1).

* Avalanche Effect:
- It is a key desirable property of encryption algorithm,
- In Avalance Effect when there is a change in one input bit or key bit it results in changing approx half output bits.
- So, guessing the keys is impossible.
- DES Exhibits strong avalanche effect.

Page No.
Date : / /

* Strength of DES (Key Size):

- DES has 56-bit keys i.e the possible key combination is $2^{56} = 7.2 \times 10^{16}$ values.
- In past, when DES is used brute force search looks hard.
- But the recent advances have shown that it is possible to break DES:
  1. In 1997 on internet in a few months,
  2. In 1998 on dedicated h/w (EFF) in a few days.
  3. In 1999 above combined in 22 hrs!
- Above are possibility that the key can be find, but we still must be able to recognize plaintext.
- So, we must now consider alternatives of DES.

* Analytic attacks on DES:

- There are several analytic attacks on DES that utilize some structure of the cipher like:
  1. by gathering information about encryption,
  2. can eventually recover some/all of the sub-key bits
  3. Then exhaustively search for the rest sub-key if necessary.
- Generally, these are statistical attacks and includes
  - i) Differential cryptanalysis,
  - ii) Linear Crypt Analysis
  - iii) related key attacks.

Page No.
Date : / /

i) Differential cryptanalysis: [chosen plaintext attack]

- Differential cryptanalysis is a statistical attack against Feistel ciphers which uses cipher structure (Feistel structure)
- Differential cryptanalysis observe the behavior of pairs of text blocks evolving along each pair of ciphers, instead of observing the evolution of single text block.
- Consider, the original plaintext block m which consists of two halves m_0, m_1.
- In Feistel structure, Each round maps the left-hand input to the right-hand output & sets the right hand output to be a function of the left-hand input and the sub-key of the round.
- So, at each round, only new 32-bit block is created.
- If we label each new block m_i then intermediate message halves are related as:

$$m_{i+1} = m_{i-1} \oplus f(m_i, K_i) \text{ where, } i = 1, 2, \ldots 16$$

In differential cryptanalysis, we start with two messages, m and m' with known XOR difference

$$\Delta m = m \oplus m',$$

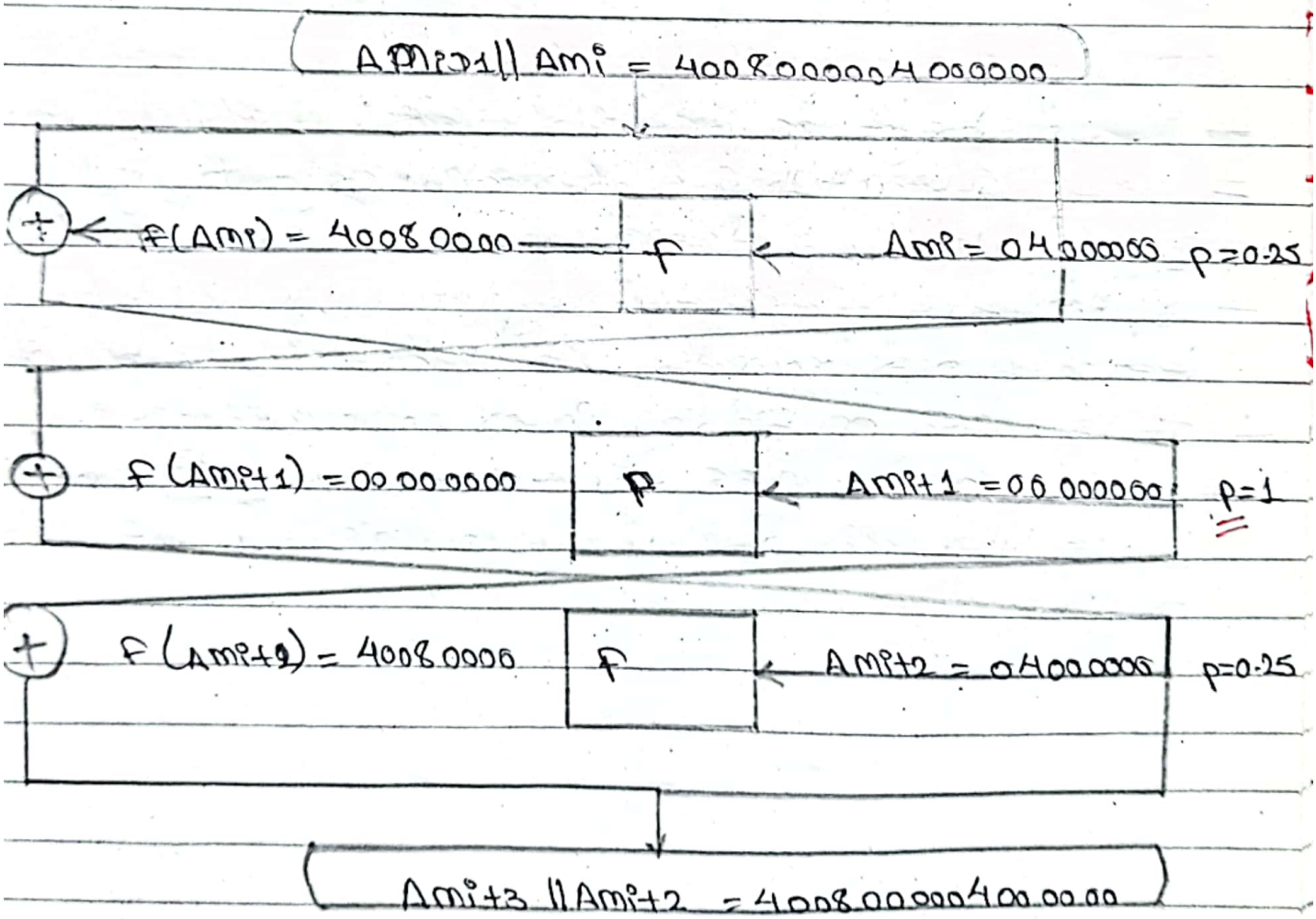
Consider the difference between the intermediate message halves $m_i = m_i \oplus m_i'$ Then we have:

$$\begin{aligned} \Delta m_{i+1} &= m_{i+1} \oplus m'_{i+1} \\ &= [m_{i-1} \oplus f(m_i, K_i)] \oplus [m'_{i-1} \oplus f(m'_i, K_i)] \\ &= \Delta m_{i-1} \oplus [f(m_i, K_i) \oplus f(m'_i, K_i)] \end{aligned}$$

So, with known difference in the input, we are searching for a known difference in output when some subkeys are used.

Steps:

Let, we have some known input difference which gives some output difference with probability p.
So, if we find instances of some higher probability input/output difference pairs occuring we can infer sub-key that was used in round.
Also, we must iterate these above process over many rounds (with decreasing probabilities) if desirable sub-keys probability is not obtained.



Page No.
Date : / /

- Perform attack by repeatedly encrypting plaintext pairs with known input XOR until obtain desired output XOR.

* When the desired XOR is found:
- If intermediate rounds match, required XOR have right pair,
- If not then have a wrong pair,

→ We can then deduce keys values for the rounds.
  i) right pairs suggests same key bits,
  ii) wrong pairs give random values.

- For large numbers of rounds, probability is so low that more input/output difference pairs are required than exist with 64-bit inputs.

ii) Linear cryptanalysis for DES: [Known plaintext Attack]
- More difficult than differential cryptanalysis,
- It is also a statistical method,
- Similar to differential cryptanalysis i.e it must be iterated over rounds, with decreasing probabilities,
- It is based on finding linear approximations,
- It can attack DES structure with 2^{43} known plaintexts easier than differential cryptanalysis but infeasible in practise

Page No.
Date : / /

AES (Advanced Encryption Standard):

Origin:

As we see DES can be decipher, there is a clear need of replacement for DES.
We can use Triple DES instead of DES but using DES structure 3 time makes it more complicated & slow. Also Attacks are feasible but only time will increase.
So, US NIST issued call for ciphers in 1997.
where 15 candidates accepted in JUN 98 but only 5 were shortlisted.

i) MARS (IBM) – Complex, Fast, high Security Margin
ii) RC6 (USA) – V. simple, V. Fast, low Security Margin
iii) Rijndael (Belgium) – clean, fast, good security Margin
iv) Serpent (Euro) – slow, clean, V. high Security margin
v) Two Fish (USA) – Complex, V. Fast, high Security Margin.

Rijndael was selected as the AES in Oct-2000
Issued as FIPS PUB 197 Standard in Nov-2001

AES Requirements:

Private key Symmetric block cipher (Encryption-decryption using same key).
128-bit data, 128/192/256-bit keys
Must be stronger & Faster than Triple DES.
Active life of 20-30 years (+ archival use)
Provide full specification & design details.
both C & Java implementations.
NIST have released all submissions & unclassified analyses.

Page No.
Date : / /

* AES Evaluation Criteria:

① Initial criteria:

- Security: effort for practical cryptanalysis.
- Cost: In term of computational efficiency.
- algorithm & implementation characteristics

② Final criteria:

- General security,
- Ease of software & hardware implementation,
- Implementation attacks.
- Flexibility (in en/decrypt, keying, other factors)

* The AES Cipher - Rijndael:

- Designed by Rijmen - Daemen in Belgium
- has 128/192/256 bit keys, 128 bit data.
  (10R, 12R, 14R)
- It is an Iterative rather than Feistel cipher.
  i) Processes data as block of 4 columns of 4 bytes.
  ii) operates on entire data block in every round.

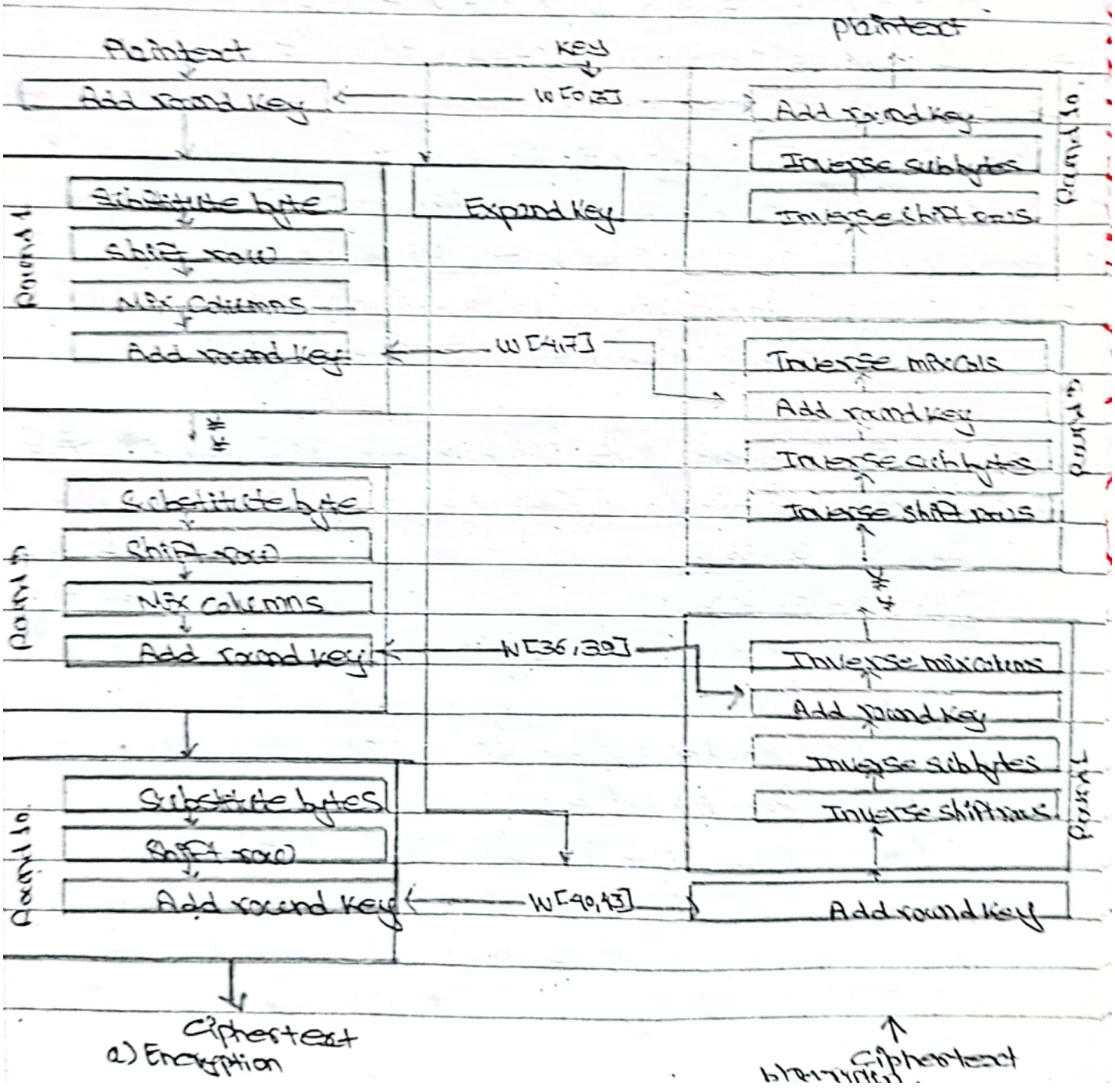
- Designed to be:
  1) Resistant against known attacks,
  2) design simplicity

How it works?

Data block of 4 columns of 4 bytes is state.

has 9/11/13 + one last extra rounds in which state undergoes:

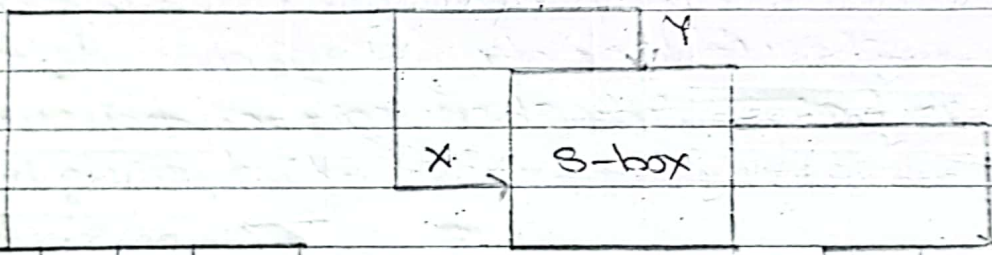
1) byte substitution (1 S-box used on every bytes)
2) shift rows (permute bytes between groups/column)
3) mix columns (scbs using matrix multiply of groups)
4) add round key (XOR state with key material)



Page No.
Date : / /

i) Byte Substitution:

→ a simple substitution of each byte,
→ uses a S-box of 16×16 bytes tables containing a permutation of all 256 8-bit values.
→ each byte of state is replaced by byte indexed by:
Left-4bits → row
Right 4-bits → columns.
i.e byte $95 is replaced by byte in row 9 & column 5 which has value $2A in S-box.
→ S-box is constructed using $GF(2^8)$
→ It is designed to be resistant to all known attacks.

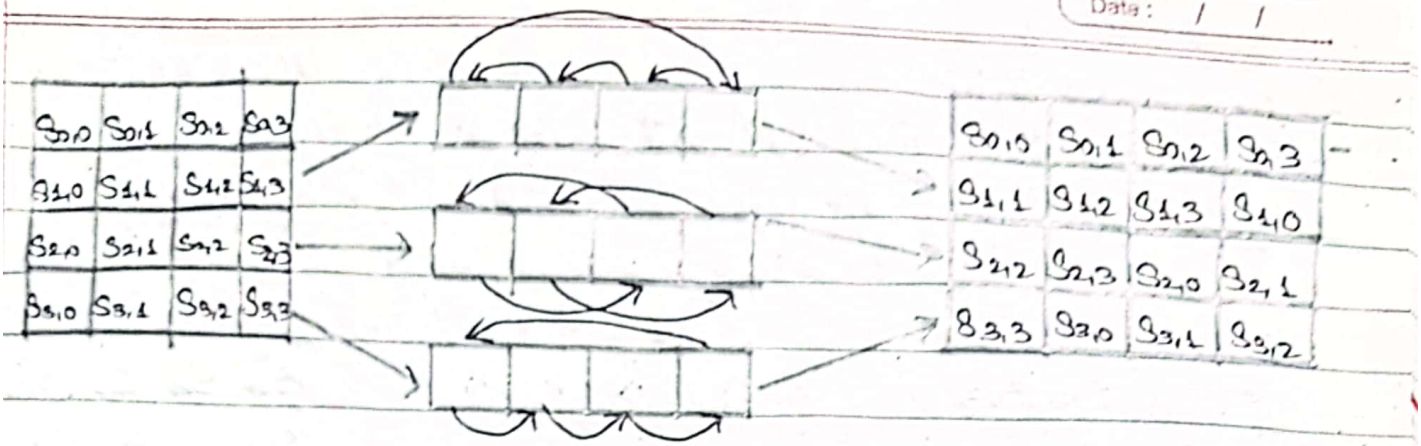


$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

$S'_{0,0}$	$S'_{0,1}$	$S'_{0,2}$	$S'_{0,3}$
$S'_{1,0}$	$S'_{1,1}$	$S'_{1,2}$	$S'_{1,3}$
$S'_{2,0}$	$S'_{2,1}$	$S'_{2,2}$	$S'_{2,3}$
$S'_{3,0}$	$S'_{3,1}$	$S'_{3,2}$	$S'_{3,3}$

ii) Shift Rows

→ A circular byte shift in each.
i) 1st row is unchanged
ii) 2nd row does 1 byte circular shift to left
iii) 3rd row does 2 byte circular shift to left.
iv) 4th row does 3 byte circular shift to left.



MIX Columns:

Each column is processed separately, each byte is replaced by a value dependent on all 4 bytes in the column.

A matrix multiplication in $GF(2^8)$ using prime poly $m(x) = x^8+x^4+x^3+x+1$

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{00} & s_{01} & s_{02} & s_{03} \\ s_{10} & s_{11} & s_{12} & s_{13} \\ s_{20} & s_{21} & s_{22} & s_{23} \\ s_{30} & s_{31} & s_{32} & s_{33} \end{bmatrix} = \begin{bmatrix} s'_{00} & s'_{01} & s'_{02} & s'_{03} \\ s'_{10} & s'_{11} & s'_{12} & s'_{13} \\ s'_{20} & s'_{21} & s'_{22} & s'_{23} \\ s'_{30} & s'_{31} & s'_{32} & s'_{33} \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} \quad \\ \quad \\ \quad \\ \quad \end{bmatrix} = \begin{bmatrix} \quad \\ \quad \\ \quad \\ \quad \end{bmatrix}$$

Page No.
Date : / /

iv) Add Round Key:

→ XOR state with 128-bits of the round key.

S_{00}	S_{01}	S_{02}	S_{03}
S_{10}	S_{11}	S_{12}	S_{13}
S_{20}	S_{21}	S_{22}	S_{23}
S_{30}	S_{31}	S_{32}	S_{33}

$\oplus$

W_0	W_{i+1}	W_{i+2}	W_{i+3}

$=$

S'_{00}	S'_{01}	S'_{02}	S'_{03}
S'_{10}	S'_{11}	S'_{12}	S'_{13}
S'_{20}	S'_{21}	S'_{22}	S'_{23}
S'_{30}	S'_{31}	S'_{32}	S'_{33}

* AES Key Expansion:

→ Takes 128-bit key and expands into array of 44/52/60

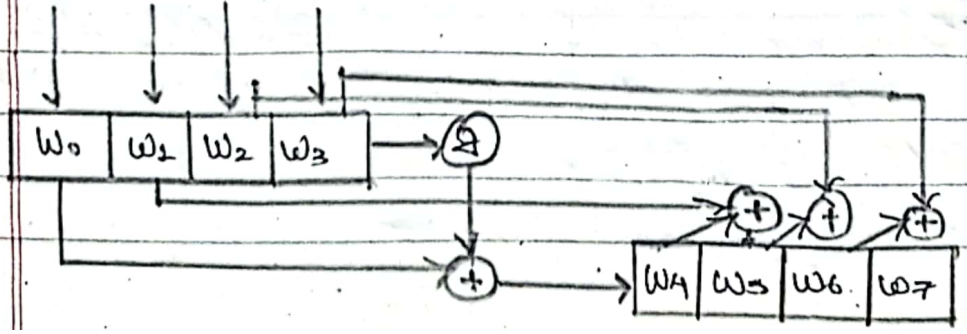
→ Start by copying key into first 4 words.

→ then loop creating words that depend on values in previous & 4 places back.

i) In 3 of 4 cases just XOR these together.

ii) 1st word in 4 has rotate + S-box + XOR round constant on previous

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}



Page No.
Date : / /

* AES Decryption:

→ AES decryption is not identical to encryption since steps done in reverse

→ For decryption we define an equivalent inverse cipher with steps as for encryption
  i) but using inverses of each step
  ii) with a different key schedule.

* IDEA (International Data Encryption Algorithm):

→ In cryptography, the IDEA is a symmetric block cipher by Xuejia Lai and James Massey and was first described in 1991.

→ The algorithm was intended as a replacement for the DES (Data Encryption Standard).

How it works:

→ IDEA operates on 64-bit blocks using a 128-bit key and consists of a series of eight identical rounds + an out round (the half-round, see in figure below).

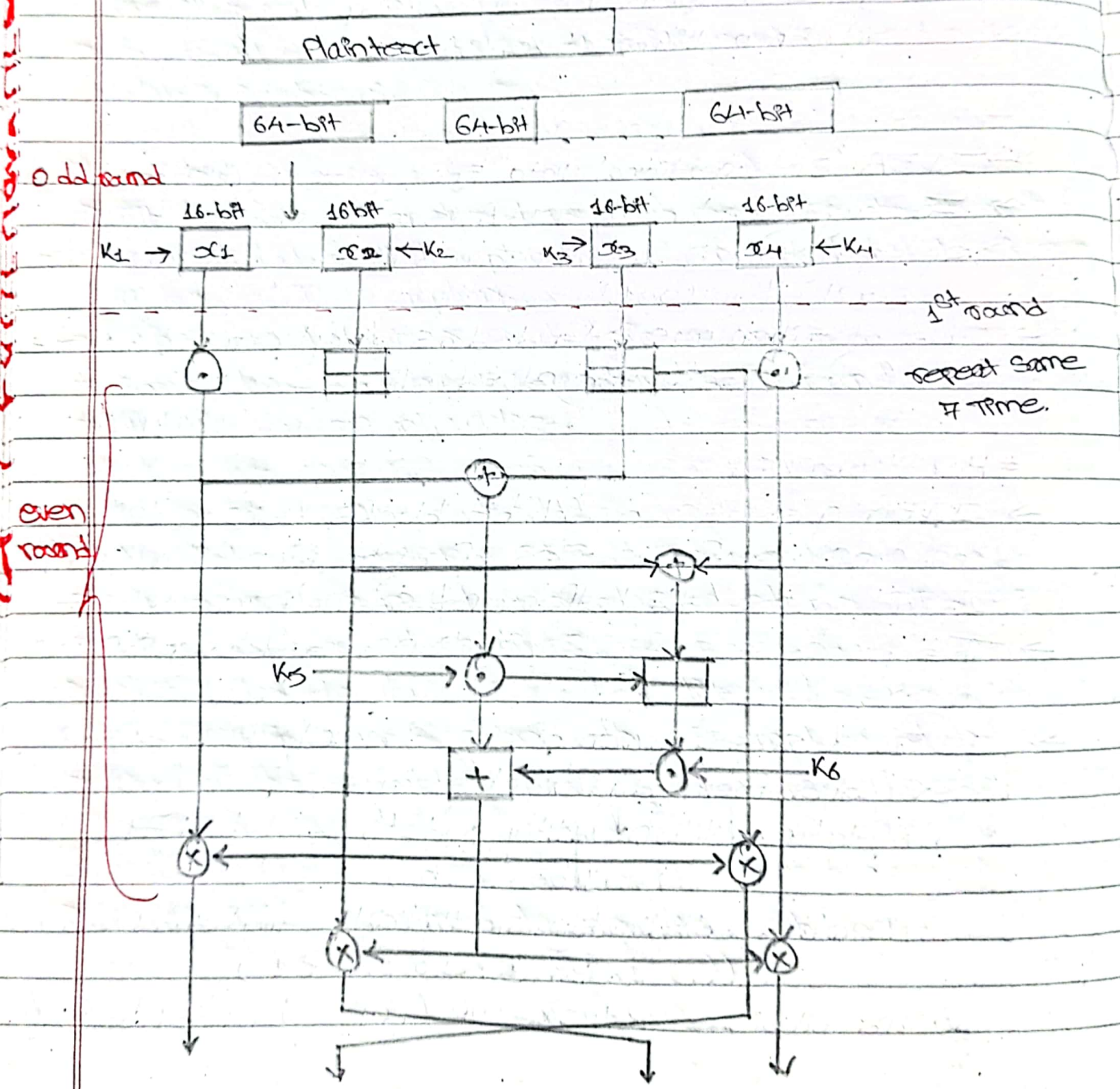
→ The processes for encryption and decryption are similar.

→ IDEA derives much of its security by performing operations from different groups.
  i) Modular addition (addition mod 2^{16}, denoted by ⊞ ($1000\ldots0$, 16)
  ii) Modular multiplication (multiplication mod $2^{16}+1$, denoted by ⊙, $1000\ldots01$, 15)
  iii) and bitwise exclusive OR (XOR) denoted by ⊕ &

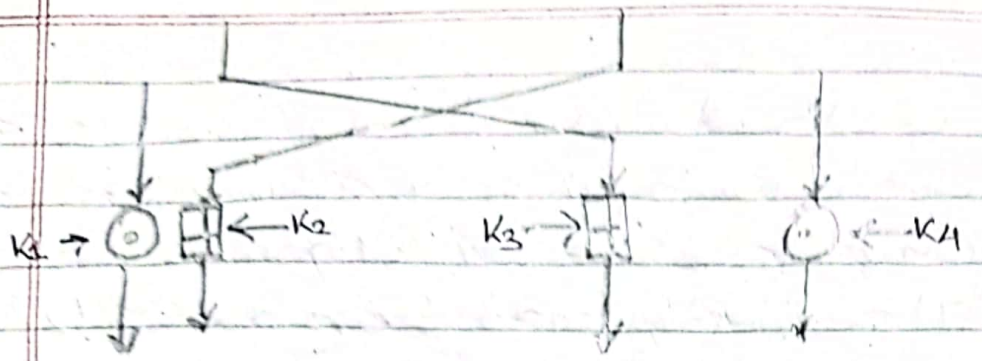
Page No.
Date : / /

→ The figure below in left shows a round (1-8) and in right shows final ½ half round.

→ Initially, the plaintext is divided into 64-bit block and again these 64-bit blocks are divided into 4 - 16 bit blocks which are used as input to the IDEA structure.



Last 5 round



* Key schedule:

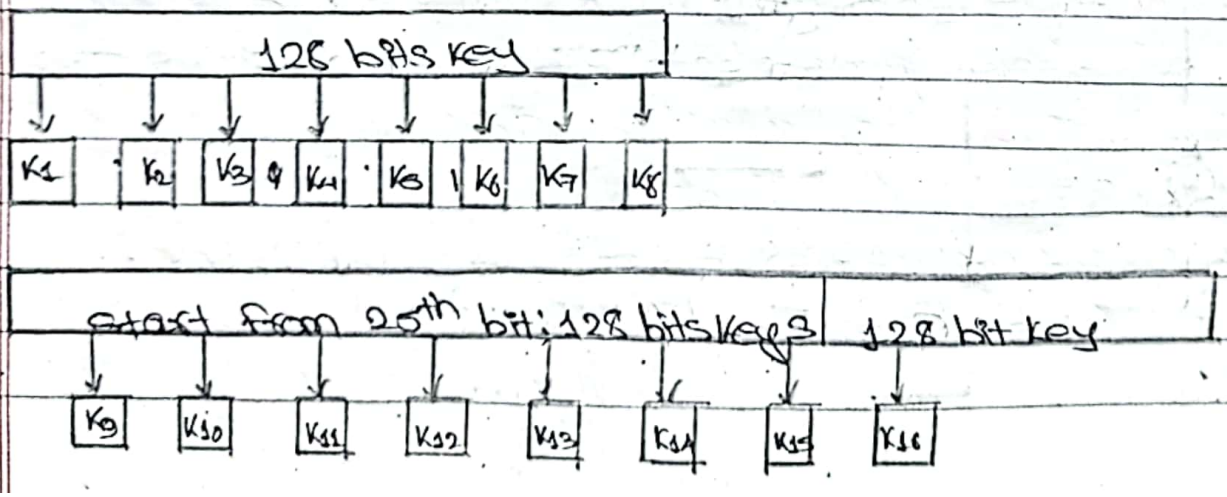
→ 6 keys are used for each round and remaining 4 keys are used in final half round.

So, for 8 round we need = 8×6 +4
= 48+4
= 52 subkeys.

→ The 128 bit key is used to generate 52 subkeys of length 16 bits.

→ The figure below shows the key generation mechanism where we start getting the subkeys from the starting position of the 128 bits keys, so in first round we get 8 subkeys.

→ In each of the round of subkey generation 128 bits keys undergoes a 25 bit position right to generate next sets of keys i.e. start with bit position 0, 25, 50 & soon until all 52 key are derived.



Page No.
Date : / /

* Round operations:

It has been mentioned above that IDEA uses 8 full rounds and 1 half round. We now break the 8 full round and make it 16 rounds such that there are total 17 rounds where 9 odd rounds (1,3,...,17) are identical and 8 even rounds (2,4,...,16) are identical.

Each odd round takes 4 subkeys where each even round takes 2 subkeys.

* Security:

→ The designers analyzed IDEA to measure its strength against differential cryptanalysis & concluded that it is immune under certain assumptions.

→ No successful linear or algebric weakness have been reported.

⇒ Some classes of weak keys have been found but these are of little concern in practice.

* Multiple Encryption & DES: (Contemporary cipher):

→ After a clear need of replacement for DES, AES is a new cipher alternatives.

→ But prior to this AES Alternative, there is a concept of use multiple encryption with DES implementations.

→ one of these implementations idea results Triple-DES

① Double DES:

→ Using DES two times to encrypts each block.

i.e; $C = E_{K_2}(E_{K_1}(P))$

Page No.
Date : / /

We have "meet-in-the-middle" attack in double DES.
i.e

$$E_{K1}(P) = X = D_{K2}(C)$$

Attack on double-DES by encrypting P with all keys and store then decrypt C with keys & match X value
Can take $O(2^{56})$ steps
Double DES with 2 PC's is similar to single DES

Triple DES with two-keys:

DES that uses 3 encryptions.
- would seem to need 3 distinct keys

But can use 2 keys with E-D-E sequence
i) $C = E_{K1}(D_{K2}(E_{K1}(P)))$
ii) but if $K1 = K2$ then it work similar to single DES
i.e,

$$C = E_{K1}(D_{K2}(E_{K1}(P)))$$

no current known practical attacks

Triple DES with Three-keys:

Although there are no practical attacks on two keys, Triple-DES have of form with three-keys to avoid even some possible attacks.
- $C = E_{K3}(D_{K2}(E_{K1}(P)))$

Adopted by some Internet Applications, eg:- PGP, S/MIME

Page No.
Date : / /

* Modes of operation:

→ Till now, we know that a block ciphers encrypts fixed size blocks.
e.g:- DES encrypts 64-bit blocks with 56-bit key.

→ But, we need some ways to encrypt & decrypts arbitrary amounts of data in practise

→ So, FIPS (Federal Information Processing standard) defines 5 possible modes:
i) Electronic Codebook Book (ECB),
ii) cipher Block chaining (CBC),
iii) cipher Feedback (CFB),
iv) Output Feedback (OFB),
v) Counter,

i) Electronic Codebook Book (ECB):

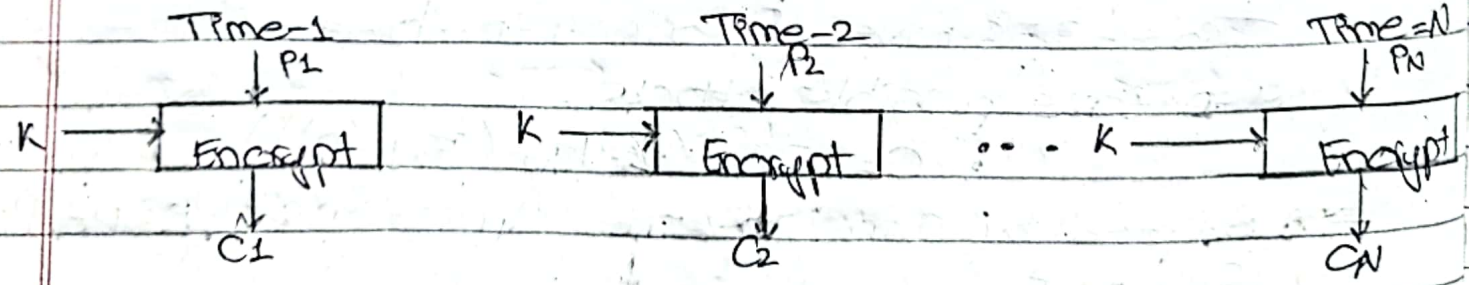
→ These are the traditional mode of encryption & decryption which we are used to. i.e using key to encrypt the plaintext to give ciphertext.

→ Messages is broken into independent blocks which are encrypted

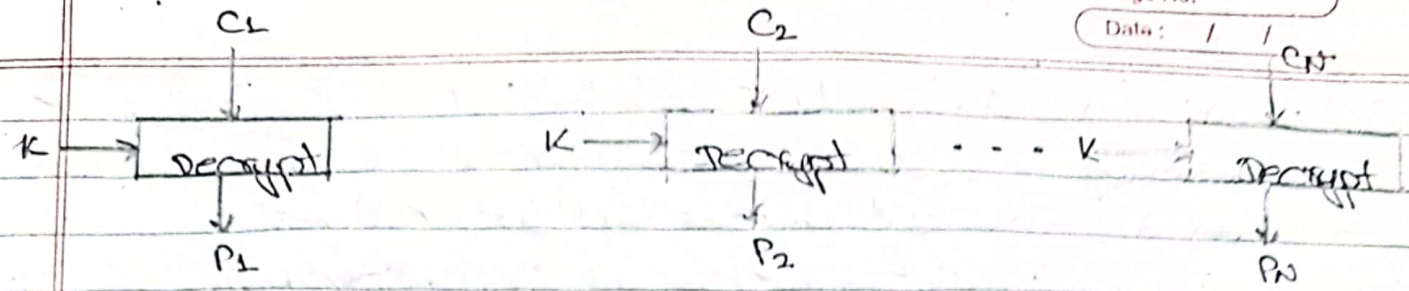
→ each block is encoded independently of the other blocks.

→ If Parallel processing is needed with less security it is used

$$C_i = DES_{K_1}(P_i)$$



a) Encryption



b) Decryption

ii) Cipher Block chaining (CBC):

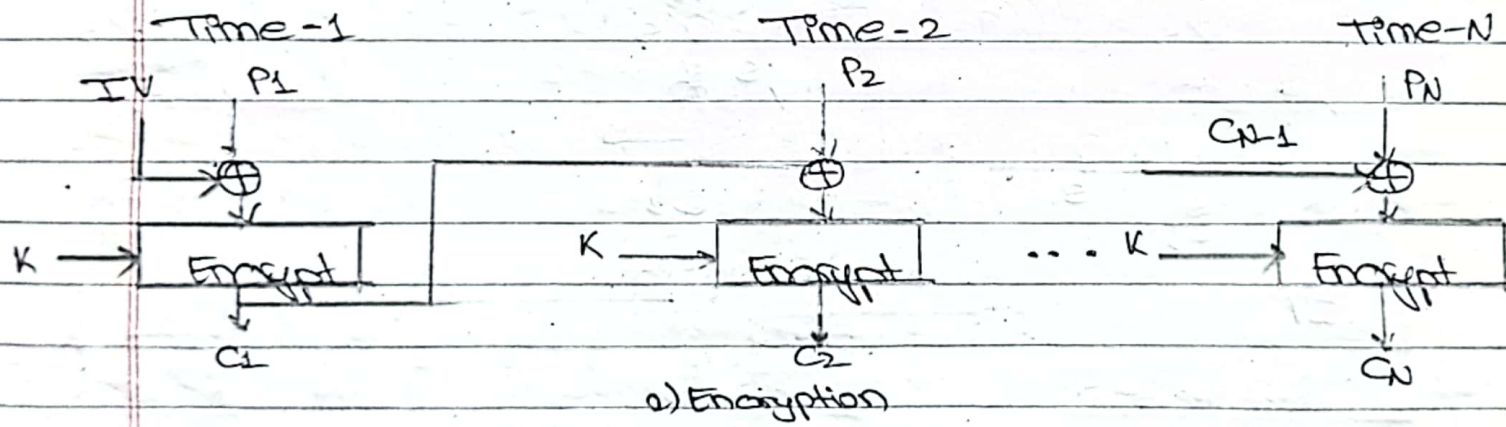
→ Message is broken into blocks,

→ Linked together in encryption operation i.e. each previous cipher blocks is chained with current plaintext block, as name suggests.

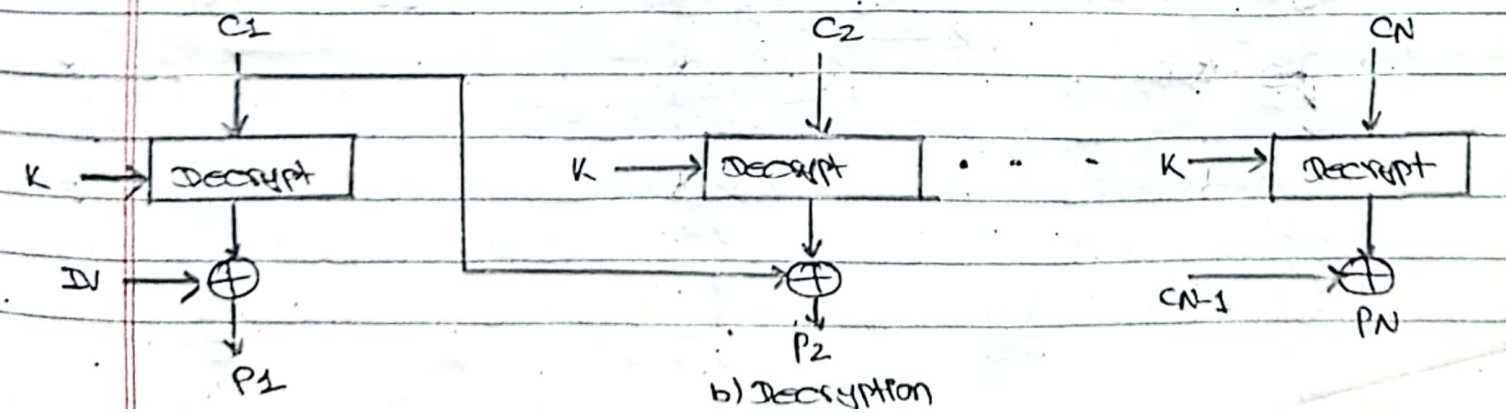
→ but Initially, we need to define Initial Vector (IV) to XOR with Plaintext to start process.

$C_i = DES_{K_1}(P_i \text{ XOR } C_{i-1})$

$C_{-1} = IV$ (Initial Vector)



a) Encryption



b) Decryption

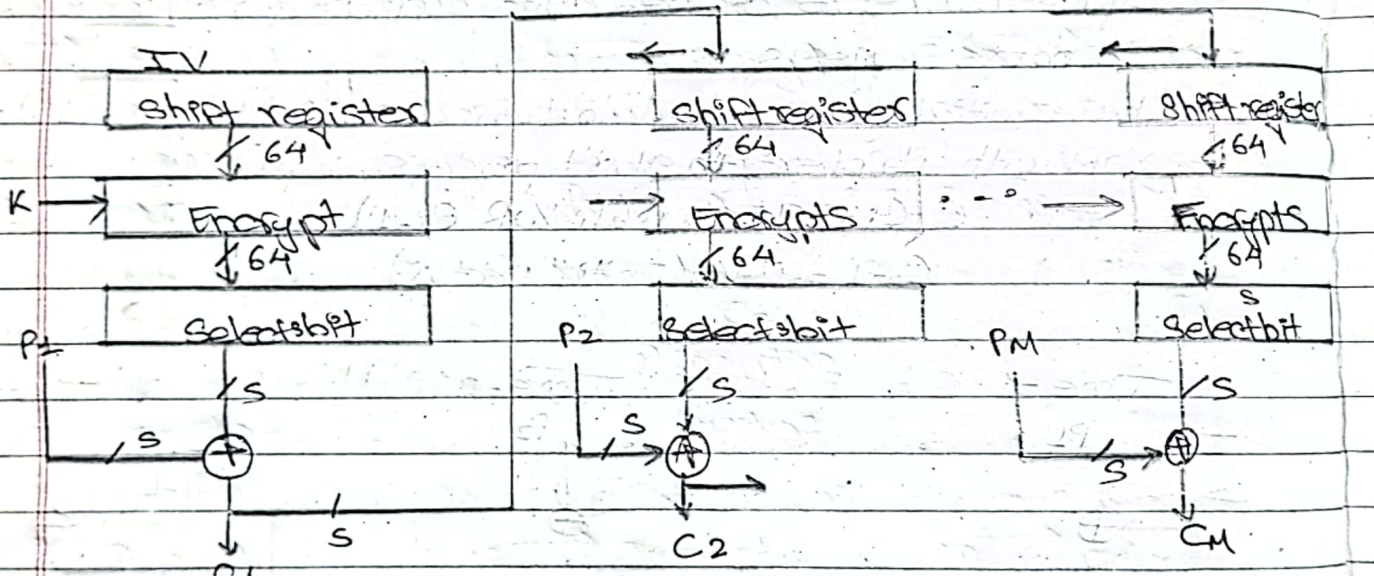
Page No.
Date : / /

iii) Cipher Feedback (CFB):

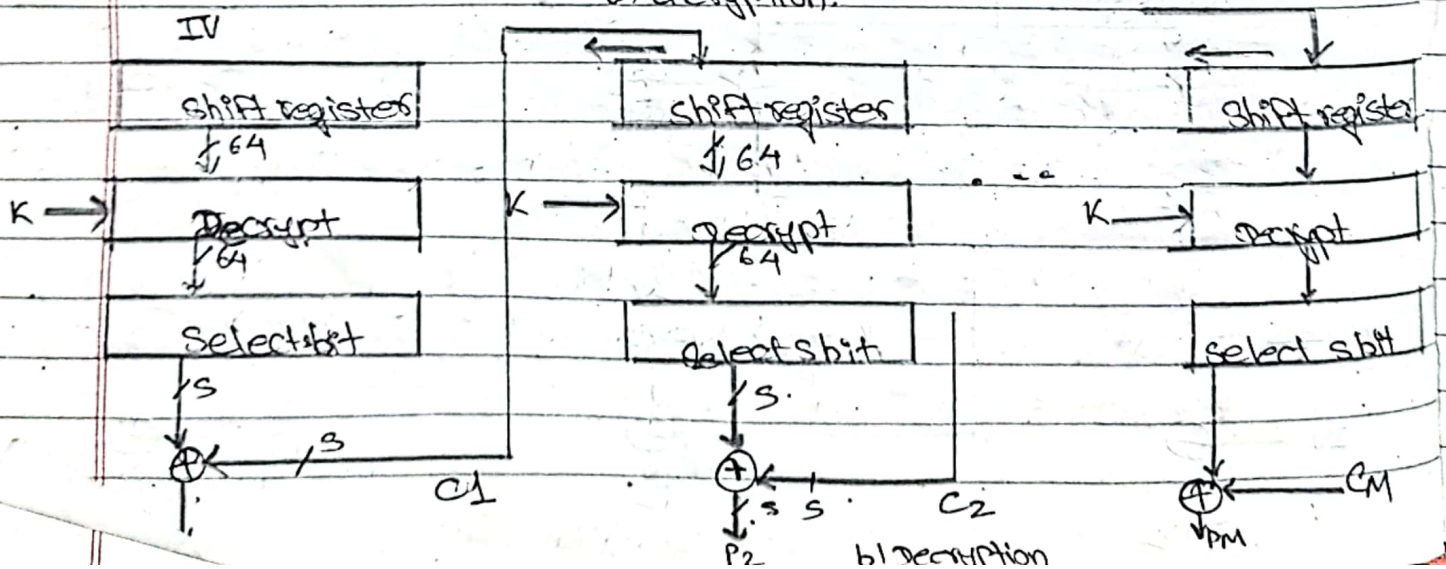
→ Message is treated as a streams of bits
→ Added to the output of the block cipher.
→ Most efficient to use all bits in blocks.

$$C_i = P_i \text{ XOR } DES_{K1}(C_{i-1})$$
$$C_{-1} = IV$$

→ Stream data encryption



a) Encryption.



b) Decryption

Page No.
Date: / /

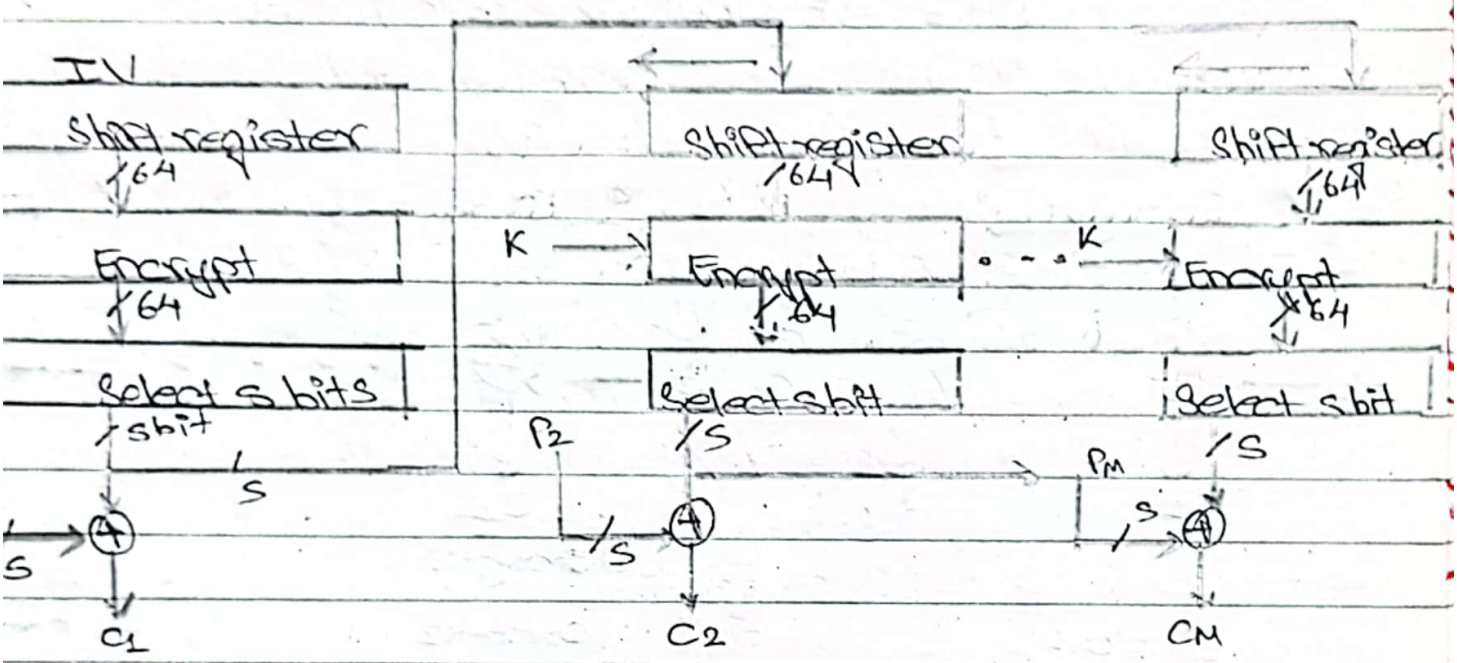
Output Feed Back (OFB):

Message is treated as a stream of bits.
Output of cipher is added to message
It can be done in parallel i.e, we can computed in advance

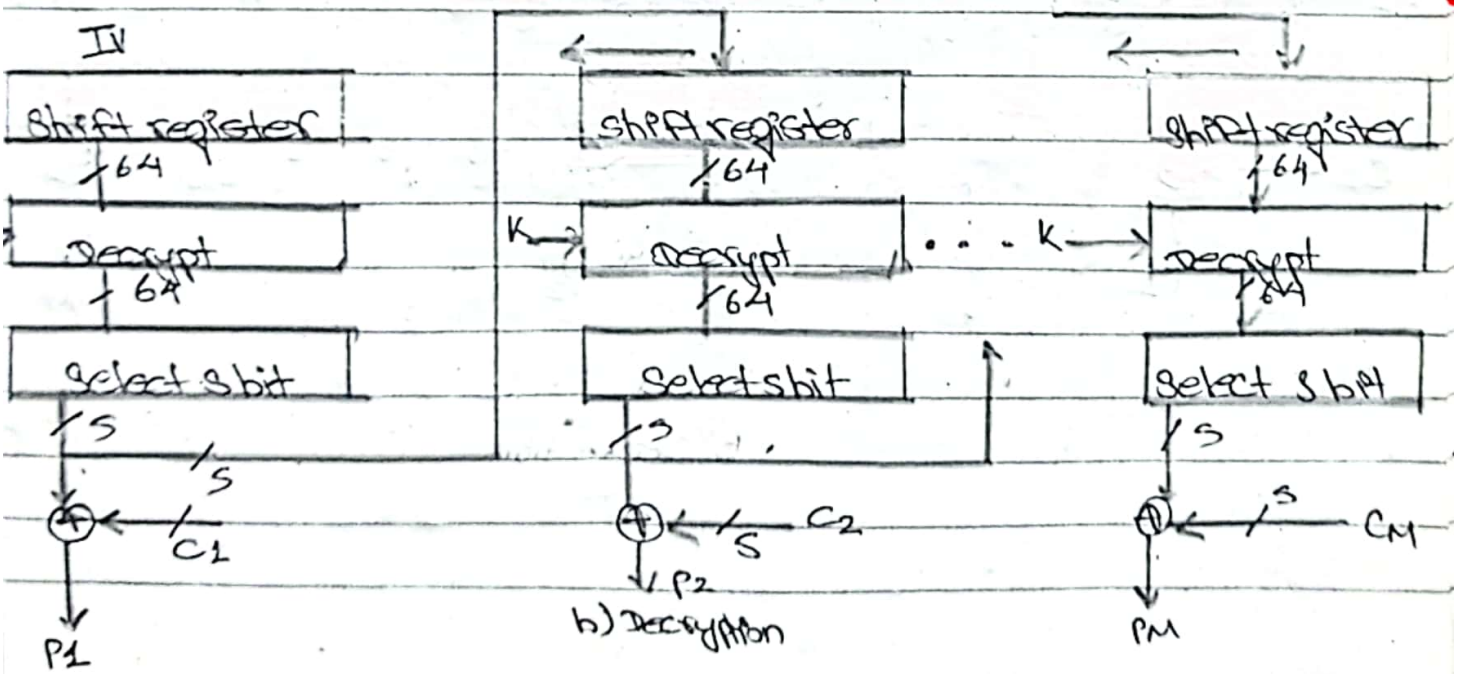
$C_i = P_i \text{ XOR } O_i$

$O_i = DES_{K1}(O_{i-1})$

$O_{-1} = IV$



a) Encryption.



b) Decryption

v) Counter (CTR):

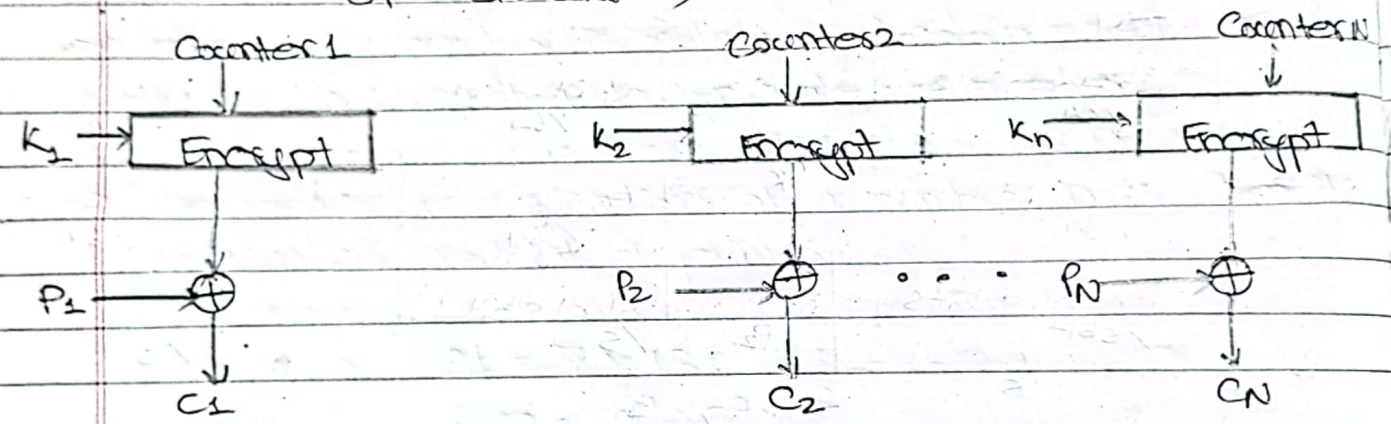
→ A "new" mode similar to OFB but encrypts counter value rather than any feedback value.

→ Must have different key & counter value for every plaintext block (never reused)

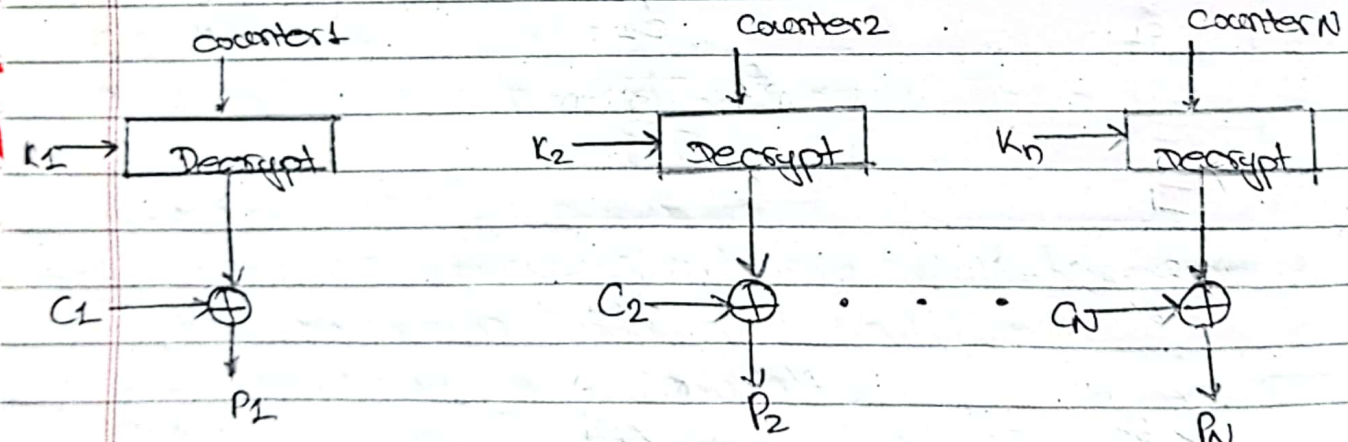
i.e,

$C_i = P_i \text{ XOR } O_i$

$O_i = DES_{K1}(i)$ i = Counter.



a) Encryption



b) Decryption

Page No.
Date : / /

* Message padding:

→ To perform encryption with a block cipher in ECB or CBC, when the length of the data to be encrypted is not an exact multiple of Block size B.S, it must be padded to make it.

→ After decrypting, the padding needs to be removed.

→ For other modes of encryption, such as Counter, CFB, OFB padding is not required i.e In these cases the cipher text is always the same length as the plaintext.

→ Padding can be done by 2 ways:

i) Pad either with known non-data value (eg 0's or Null)

ii) or pad last block along with count of pad size.

eg [b1 b2 b3 0 0 0 0 5]

padding

means have 3 bytes of data, then 5 bytes Pad + Count

Page No.
Date : / /

Unit: 3

Public key cryptography and Discrete Logarithms:

* Prime Factorization:

→ Finding the factor of a number n is to write it as a product of other numbers.

i.e, $n = a \times b \times c$

→ Factoring a number is relatively hard compared to multiplying the factors together to generate the number.

→ The prime factorization of a number n is when its written as product of primes

for eg:-

$91 = 7 \times 13;$

$3600 = 2^4 \times 3^2 \times 5^2$

we can denote it as;

$$n = \prod_{p \in P} p^{a_p}$$

#Note: Two numbers a, b are relatively prime if they have no common divisor apart from 1 ie

$$GCD(a,b) = 1$$

eg:- $a = 300 = 2^1 \times 3^1 \times 5^2$

$b = 18 = 2^1 \times 3^2$

So,

$GCD(18, 300) = 2 \times 3 = 6$

Page No.
Date : / /

* Fermat's theorem:

→ $a^{p-1} \bmod p = 1$

where, p is prime and gcd (a, p) = 1

→ Also known as Fermat's little Theorem

→ We can also say it as;

$$a^{p-1} \cdot a^1 \bmod p = 1 \cdot a^1$$

$$a^p \bmod p = a$$

$$a^p \bmod p = a$$

→ Useful in public key & primality testing.

eg:

let p = 5 then co-prime of 5 a = 4

So, that

gcd (a, p) = 1

So,

$a^{p-1} \bmod p = 1$

$4^{5-1} \bmod 5 = 1$

$256 \bmod 5 = 1$

$1 = 1$

Also,

$a^p \bmod p = a$

$4^5 \bmod 5 = 4$

$1024 \bmod 5 = 4$

$4 = 4$

* Euler Totient Function $\phi(n)$:

→ It is a Function which provides the total no. of co-prime less than n. where n is the given number.

eg:- n = 10

$\phi(n) = 4$

i.e,

Co-prime of 10 less than 10 = {1, 3, 5, 7}

Page No.
Date : / /

Similarly, $n = 15$

$\phi(n) = 8$

i.e. Co-prime (15) = {1, 2, 4, 7, 8, 11, 13, 14}

= 8

→ We can also find the Euler's Totient Function as:

i) For p (prime number) $\phi(p) = p-1$

eg $p = 17$

$\phi(p) = 16$

ii) For p·q (p, q prime), $\phi(pq) = (p-1) \times (q-1)$

i.e.

eg $n = 21$

then, $n = p \times q$

$n = 7 \times 3$

So, $\phi(n) = (7-1) \times (3-1)$

$= 12$

✳ Euler's Theorem:

→ It is a generalisation of Fermat's Theorem.

i.e.

$$a^{\phi(n)} \bmod n = 1$$

where, a, n are random number where.

$$\gcd(a, n) = 1$$

For eg:- Let $a = 3$ & $n = 10$.

then,

$\phi(n) = 4$

So,

$a^{\phi(n)} \bmod n = 1$

$3^4 \bmod 10 = 1$

$81 \bmod 10 = 1$

$1 = 1$

Page No.
Date: / /

Primality Testing:

→ In cryptography, we always need large prime numbers.
→ Traditionally, we determine whether the number is prime or not by using trial division. i.e.
  i) divide by all numbers (primes) in turn less than the square root of the number
  ii) But it only works for small numbers.
→ Alternative way of primality test is using statistical primality tests based on the properties of primes.
  i) We use Pseudo-prime number concepts i.e Those number which have all property of prime number but may or may not be prime.
→ To determine whether the pseudo-prime number is prime or composite we use Miller-Rabin Algorithm which gives 99.999% probability that the number is prime.

Miller-Rabin Algorithm:

→ Determines whether a given pseudo-prime number is a prime number or composite number.
→ A Test based on Fermat's Theorem

Algorithm:

i) Test (n) is:
ii) Find integers $k, q, k>0$, q odd, so that $(n-1) = 2^k \cdot q$
iii) Select a random integer a, $1<a<n-1$
iv) if $a^q \bmod n = 1$ then return ("may be prime"); 1st condition
v) for $j = 0$ to $k-1$ do
    if $(a^{2^j \cdot q} \bmod n = n-1)$ 2nd condition.
    then return ("may be Prime")
vi) return ("composite")

Page No.
Date : / /

eg:- $a = 45$

So, $n = 45$ Case (1):-

or $n-1 = 44 = 2^2 \cdot 11$

So, $2^k \cdot q$ compare with $2^2 \cdot 11$

$k = 2, \ q = 11$

now,

Let select $a = 2$ where $1 < a < 45$

$a^q \bmod n = 1$

$2^{11} \bmod 45 = 1$

$23 \neq 1$ case (1) is False

again, for case (2):

select. $j = 1$, where $i = 0$ to $k-1$ $(2-1)$

$= 0$ to 1

when $i = 0$

$2^{2^0 \cdot 11} \bmod n = n-1$

$2^{11} \bmod n = n-1$

$23 = 45-1$

$23 = 44$ (False)

when $i = 1$

$2^{2^1 \cdot 11} \bmod 45 = n-1$

$34 = 45-1$

$34 = 44$ False.

So, both case (1) & (2) are false. So the pseudo number is "composite"

* Probabilistic Considerations:

→ If Miller-Rabin returns "composite" the number is definitely not prime.

→ Otherwise it may be prime or a pseudo-prime.

→ The chances it detects a pseudo-prime is $\frac{1}{4}$ i.e 25%.

→ Hence, if we repeat test with different random values of a ($1 < a < n-1$), then the chances of n is prime after t test is:

i) n is prime after t-tests: $1 - \left(\frac{1}{4}\right)^t$

ii) if $t = 10$ this probability is $\approx$ 99.9999%.

≈ 1 i.e the number is prime

Note:- Prime Distribution: Prime number occurs roughly after $\ln(n)$.

→ but we can immediately ignore evens.

→ So in practice we only need to test $0.5 \times \ln(n)$ numbers of size n to locate a prime.

→ This is only the "average" but sometimes primes are close together or sometimes are quite far apart.

* Chinese Remainder theorem:

→ It is used to speed up the modulo computations.

→ If the working modulo is a product of numbers.

eg:- $a \bmod M = x$

$M = m_1 \cdot m_2 \cdot \ldots \cdot m_k$

$a \bmod m_1 \cdot m_2 \cdot \ldots \cdot m_k$

→ This theorem lets us work in each moduli m_i seperately.

→ Since computational cost is proportional to size, this is faster than working in the full Modulus M.

Page No.
Date : / /

→ This chinese remainder theorem can be used to find value of x in following type of situation where,

$x \equiv 1 \mod 5$
$x \equiv 3 \mod 7$
} where, these 5,7 should be co-prime

* Explaination:

$x \equiv a_1 \pmod{m_1}$
$x \equiv a_2 \pmod{m_2}$
$x \equiv a_3 \pmod{m_3}$

where, $\gcd(m_1, m_2) = \gcd(m_2, m_3) = \gcd(m_3, m_1) = 1$
i.e. all co-prime

then, $$x = (N_1 X_1 a_1 + N_2 X_2 a_2 + N_3 X_3 a_3 + \cdots + N_n X_n a_n) \bmod M$$

where, $M = m_1 * m_2 * m_3 \cdots m_n$

$M_i = \frac{M}{m_i}$ e.g:- $M_1 = \frac{M}{m_1} = \frac{m_1 * m_2 * m_3}{m_1}$

$= m_2 \times m_3$

Similarly, $M_2 = \frac{M}{m_2} = m_1 \times m_3$

$M_3 = \frac{M}{m_3} = m_1 * m_2$

To calculate X_i

Multiplicative Inverse of M_i

$$M_i X_i = 1 \bmod m_i$$

e.g $M_1 X_1 \equiv 1 \bmod m_1$

=

Page No.
Date : / /

now, let's see an example:

we have, $x \equiv 1 \pmod 5$

$x \equiv 1 \pmod 7$

$x \equiv 3 \pmod{11}$

So,

$a_1 = 1,\ a_2 = 1,\ a_3 = 3$ & $m_1 = 5,\ m_2 = 7$ & $m_3 = 11$

So,

Since 5, 7 & 11 all are relatively prime to one another. So, we can find x

i.e. $\gcd(5,7) = \gcd(7,11) = \gcd(5,11) = 1$

$N = m_1 * m_2 * m_3$

$= 5 \times 7 \times 11$

$= 385$

now,

$M_1 = \frac{m_1 * m_2 * m_3}{m_1} = m_2 \times m_3$

$= 7 \times 11$

$= 77$

$M_2 = m_1 m_3 = 5 \times 11 = 55$

$M_3 = m_1 . m_2 = 5 \times 7 = 35$

$M_1 = 77$

$M_2 = 55$

$M_3 = 35$

now we need to find

X_1, X_2, X_3

For X_1

$M_1 X_1 \equiv 1 \bmod m_1$

$77 . X_1 \bmod 5 = 1$

$2 . X_1 \bmod 5 = 1$

$=$

$\because \frac{77}{5} = 2$ Remainder

So,

$X_1 = 3$

ii) For X_2:

$M_2 X_2 \bmod m_2 = 1$
$55 X_2 \bmod 7 = 1$
$6 X_2 \bmod 7 = 1$

So, $X_2 = 6$

iii) For X_3:

$M_3 X_3 \bmod m_3 = 1$
$35 X_3 \bmod 11 = 1$
$2 X_3 \bmod 11 = 1$

So, $X_3 = 6$

Now,
we have,
$X_1 = 3, X_2 = 6, X_3 = 6$
$a_1 = 1, a_2 = 1, a_3 = 3$
$M_1 = 77, M_2 = 55, M_3 = 35$
$M = 385$ ($m_1 * m_2 * m_2$
$5 \times 7 \times 11$)

So, $x = (M_1 X_1 a_1 + M_2 X_2 a_2 + M_3 X_3 a_3) \bmod M$
$= (77 * 3 * 1 + 55 * 6 * 1 + 35 * 6 * 3) \bmod 385$
$= 1191 \bmod 385$
$= 36$

$\therefore x = 36$ using chinese remainder theorem

Page No.
Date : / /

Primitive Roots:

We study Euler's Theorem which is:

$$a^{\phi(n)} \bmod n = 1$$

Consider $a^m \bmod n = 1$, where a & n are co-prime i.e GCD(a, n) = 1

If the smallest value of $m = \phi(n)$, then we can say a is primitive root of n. satisfying above condition $a^m \bmod n = 1$

But there may exists some value for $m < \phi(n)$ such that

$a^{m<(\phi(n))} \bmod n = 1$ then a is not primitive root of n

eg:-

let $a = 3$

$n = 10$

then, $\phi(n) = 4$ {1, 3, 7, 9}

now, $m = 1$ to 4

when, $m = 1$ $a^m \bmod n = 1$

$3^1 \bmod 10 = 3 \neq 1$

when, $m = 2$ $3^2 \bmod 10 = 9 \neq 1$

when $m = 3$ $3^3 \bmod 10 = 7 \neq 1$

when $m = 4$ $3^4 \bmod 10 = 1 = 1$

We can see that the smallest value of m which is $4 = \phi(n)$

So, 3 is primitive root of 10.

now, let's see another example where this doesn't satisfy

$a = 2$ & $n = 15$

So,

$\phi(n) = 8$ {1, 2, 4, 7, 8, 11, 13, 14}

now, m = 1 to 8

So,

when, m=1. then $a^m \bmod n = 1$

$2^1 \bmod 15 = 2 \neq 1$

when, m=2 then $2^2 \bmod 15 = 4 \neq 1$

when. m=3 then $2^3 \bmod 15 = 8 \neq 1$

when m=4 then $2^4 \bmod 15 = 1 = 1$

So, here, when m = 4 which is $< \phi(15) = 8$

$4 < 8$ satisfy conditions.

2 is not the primitive root of 15.

* Discrete Logarithms:

→ The ~~Discrete~~ logarithms is the inverse problem to exponentiation problem.

i.e,

$x^y = z$ (Exponential problem)

$10^2 = 100$

where,

$\log_{10} 100 = y$

$\log_x z = y$ (Logarithm problem)

$\log_{10} 100 = 2$

→ Similarly, The inverse problem to modular exponentiation is known as discrete logarithm of a number modulo p.

Page No.
Date : / /

i.e,

$x^y \bmod n = z$ (modular exponentiation)

$\log x^z \bmod y\ n = y$ (Discrete logarithms)

~~$\log_x n^z \bmod$~~

$\rightarrow \log_{x,n}^{z} = y$ (we write discrete logarithms this way)

$\rightarrow$

$\log_2^3 \bmod 13 = 4$

If (2,13) are primitive root only the answer exists.

$\log_3^4 \bmod 13 = ?$ Answer doesn't exists.

They are not the primitive roots

$\Rightarrow$ So, $X = \log_g^Y \pmod p$

Must be primitive root

If g is a primitive root of p then only discrete logarithm exists otherwise Answer doesnot exists.

$\Rightarrow$ So, exponentiation is relatively easy, but find discrete logarithms is generally a hard problem.

$\Rightarrow$ Prime factorisation is also another example of a hard problem.

* Public-key cryptography:

→ Traditional approach of cryptography uses private/secret single key which is shared between sender & receiver.

→ If the key is disclosed communications are compromised.

→ A symmetric key cryptography or private-key cryptography doesn't protect message from forging & claiming that it is sent by sender at receiver's end.

→ It uses two keys :- ① a public key
② a private key

→ The main reason behind the use of public-key cryptography are:

i) Key distribution:

In private key cryptography the main issue came due to large number of key's & it's management. The private cryptosystem ~~system~~ is secure itself but the drawback is due to key management. So, public-key gives answer to the question of How to have secure communications in general without having to trust a KDC (Key distribution Center) with your key.

ii) Digital signature:

It also provides the concept of digital signature which verify a message intact from the claimed sender. i.e message Authenticity.

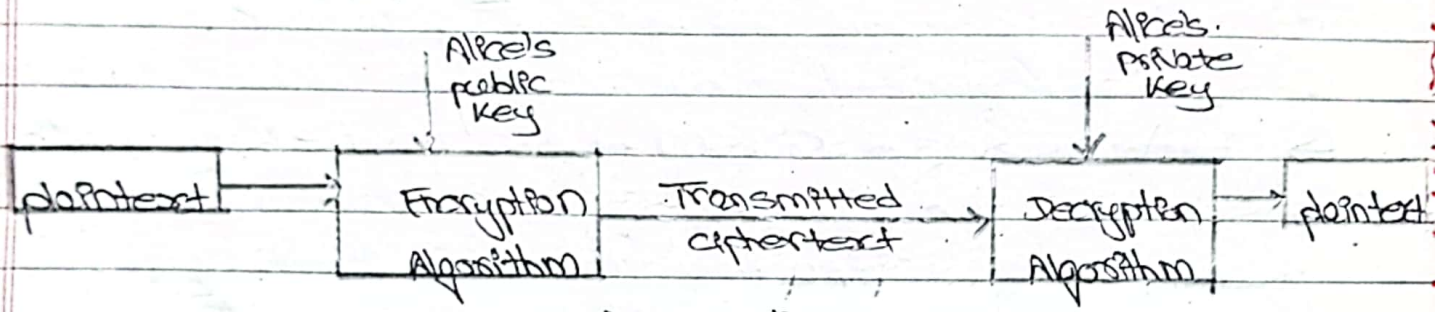
→ Public-key / two-key / Asymmetric cryptography involves the use of two keys:

i) Public key: which may be known by anybody, and can be used to encrypt messages, & verify signatures.

ii) Private key:

Known only to the recipient, used to decrypt messages, and sign signatures.

→ It is known as Asymmetric because those who encrypts messages or verify signatures cannot decrypt messages or create signatures.



a) Encryption.

* Characteristics:

→ public-key algorithms rely on two keys where:

i) It is computationally infeasible to find decryption key knowing only algorithm & encryption key, which is also known as Trap door function or one-way problem.

ii) It is computationally easy to en/decrypt messages when the relevant key is known.

iii) either of the two related keys can be used for encryption, with the other used for decryption (for some algorithm)

* Applications:

→ encryption / decryption (provide security)

→ digital signatures (provide authentication)

→ key exchange (of session keys)

Some algorithms are suitable for all uses, others are specific to one.

Page No.
Date : / /

* RSA :

→ Developed by Rivest, Shamir & Adleman of MIT in 1977
→ It is one of the best known & widely used public-key scheme
→ It uses large integers (eg 1024 bits)
→ Secure due to cost of factoring large numbers.
i.e factorization takes $O(e^{\log n \log\log n})$ operations (hard)

* Algorithm:

→ Each user generates a public/private key pair by:
→ Selecting two large primes at random : p, q
→ Calculate :

$$n = p \times q$$
$$\phi(n) = (p-1) \times (q-1)$$

→ Selecting at random the encryption key e :
Where;

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

→ Solve the following equation to find decryption key d
i.e,

$$e \cdot d \equiv 1 \bmod \phi(n) \text{ and } 0 \leq d \leq n$$

→ Publish the public encryption key : $PU = \{e, n\}$
→ Keep secret private decryption key : $PR = \{d, n\}$

Page No.
Date : / /

① Encryption:

to encrypt a Message M the sender:

① obtains publickey of recipient $PU = \{e, n\}$

$C = M^e \bmod n$, where $0 < M < n$

② Decryption:

to decrypt the ciphertext C the owner(Receiver)

① uses their private key $PR = \{d, n\}$

$M = C^d \bmod n$

#Note: The message M must be smaller than the modulus n.

* let see an example:

① Generating two large prime numbers:

prime number 1 (p) = 7

prime number 2 (q) = 17

now,

$n = p \times q$

$= 7 \times 17$

$= 119$

② Generating e:

Compute totient of n, $\phi(n) = (p-1) \times (q-1)$

$= 6 \times 16$

$= 96$

choosing a number that has gcd = 1 with $\phi(n)$ i.e

prime numbers between 1 & 96 are:

{ 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89}

Page No.
Date : / /

now, use bruteforce method to find the prime number that has gcd = 1 with $\phi(n)$ i.e.

$e = 2$, $\gcd(2, 96) = 2$ ✗
$e = 3$, $\gcd(3, 96) = 3$ ✗
$e = 5$, $\gcd(5, 96) = 1$ (but small number so not secure)
$e = 7$, $\gcd(7, 96) = 1$ (but not secure)
⋮
$e = 89$ $\gcd(89, 96) = 1$

We can consider any number for e which is
$1 < e < \phi(n)$, $\gcd(e, \phi(n)) = 1$

We assume $e = 89$

③ Generating of d:
Where, d is the multiplicative inverse of e with modulo $\phi(n)$.
$e = 89$
$\phi(n) = 96$
$d = ?$
such that, $d \times e \equiv 1 \bmod \phi(n)$
$d \times 89 = 1 \bmod 96$

$d = 41$ (using extended euclidean algorithm)

$41 \times 89 = 1 \bmod 96$

④ public key $(e, n) = (89, 119)$
private key $(d, n) = (41, 119)$

Page No.
Date : / /

* RSA Security:

→ Possible approaches to attacking RSA are:

(1) Brute force key search (infeasible given size of numbers),
(2) Mathematical attacks (based on difficulty of computing $\phi(n)$, by factoring modulus n).
(3) timing attacks (on running of decryption)
(4) Chosen ciphertext attacks (given properties of RSA)

(1) Brute force attack:

→ Brute force attack in RSA is simply finding the prime factorisation of n i.e

if we can find $n = p \times q$

we can break RSA using p & q.

(2) Mathematical attack:

→ Mathematical attack in RSA is finding the value of $\phi(n)$. We known the public key (e, n) pu. using these if we can find $\phi(n) = (p-1)(q-1)$

we can break RSA.

(3) Timing attacks:

→ Uses the concept of Exponentiation to decrypt the RSA.

(a) Exponentiation:

→ use the Square & Multiply Algorithm

→ based on the concept of repeatedly squaring base

for eg:

$7^{123} \bmod 11$ can also be calculated using exponentiation as

1 2 4 8 16 32 64 128

→ $(7 \times 7^{2})^{2})^{2})^{2})^{2})^{2} \times 7 \bmod 11$

Page No.
Date : / /

Also, we can convert 128 into binary

10000000 → multiply when 1
→ square when 0

so,

$(7 \times 7)^2)^2)^2)^2)^2)^2)^2 \bmod 11$

→ ~~Attackers~~ Computer takes extra time & resources while perfo
it's operation in the sequence.
→ Attacker may guess power consumption & Resources taken
so can practically guess the key.

④ Chosen ciphertext Attacks:
→ Attackers chooses ciphertexts & gets decrypted plaintext back
→ choose ciphertext to exploit properties of RSA to provide info to help cryptanalysis.

* Rabin CryptoSystem:
→ The Rabin cryptosystem can be thought of as an RSA cryptosystem in which the value of e & d are fixed.
→ i.e.

encryption = $C = P^2 \pmod n$

Decryption = $P = C^{1/2} \pmod n$

(i) Key Generation:
→ choose two large primes p and q in the form of $4k+3$ & $p \neq q$.

$p \bmod 4 = 3$
$q \bmod 4 = 3$

Page No.
Date : / /

calculate: $n = p \times q$

public-key = n

private-key = (p, q)

Encryption:

using public-key → (n, p)

where P is plain-text

$$C \rightarrow P^2 \bmod n$$

Decryption:

The Rabin Cryptosystem is not deterministic.

Decryption creates four plaintexts.

i.e, private key (p, q) & cipher text C.

We calculate,

$$a_1 \rightarrow +\left(C^{(p+1)/4}\right) \bmod p$$
$$a_2 \rightarrow -\left(C^{(p+1)/4}\right) \bmod p$$
$$a_3 \rightarrow +\left(C^{(q+1)/4}\right) \bmod q$$
$$a_4 \rightarrow -\left(C^{(q+1)/4}\right) \bmod q$$

Then using the values of a_1, a_2, a_3, a_4 and Chinese remainder theorem four plaintext is generated among which one is the valid plaintext.

$P_1 \rightarrow$ chinese-Remainder (a_1, b_1, p, q)
$P_2 \rightarrow$ chinese-Remainder (a_1, b_2, p, q)
$P_3 \rightarrow$ chinese-Remainder (a_2, b_1, p, q)
$P_4 \rightarrow$ chinese-Remainder (a_2, b_2, p, q)

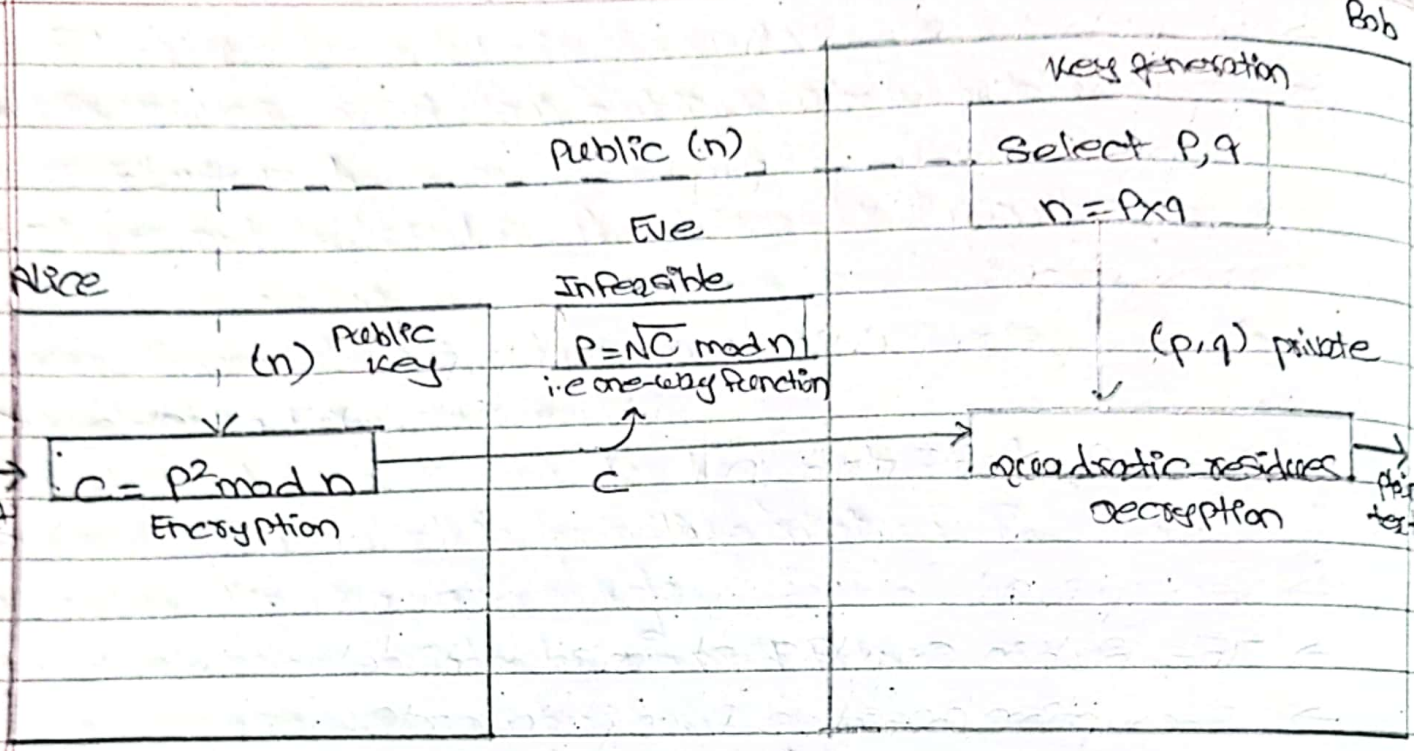


Fig :- block diagram of Rabin cryptosystem

* Let's see an example:

Let, $p = 7$, $q = 11$

both must be congerent to 3 mod 4 i.e,

$p = 7 \bmod 4 = 3$

$q = 11 \bmod 4 = 3$

now,

$n = p \times q = 77$ (public key)

Let our plaintext $P = 24$.

also $\gcd(p, n) = 1$

$\gcd(24, 77) = 1$

now,

① Encryption $(C) = P^2 \bmod n$

$= 24^2 \bmod 77$

$= 37$

Decryption:

$$a_1 = +(c^{(p+1)/4}) \bmod p = +(37^{(7+1)/4}) \bmod 7$$
$$= 4$$

$$a_2 = -(c^{(p+1)/4}) \bmod p = -(37^{(7+1)/4}) \bmod 7$$
$$= 3$$

$$b_1 = +(c^{(q+1)/4}) \bmod q = +(37^{(11+1)/4}) \bmod 11$$
$$= 9$$

$$b_2 = -(c^{(q+1)/4}) \bmod q = -(37^{(11+1)/4}) \bmod 11$$
$$= 2$$

So, we get

$a_1 = 4$

$a_2 = 3$

$b_1 = 9$

$b_2 = 2$

now, using (a_1, b_1, p, q)

(a_1, b_2, p, q)

(a_2, b_1, p, q)

(a_2, b_2, p, q)

& using chinese remainder theorem we get possible four plaintext as: P_1, P_2, P_3, P_4 where any one of the plaintext will be 24.

Page No.
Date : / /

✱ ELGAMAL Cryptosystem:

→ Besides RSA and Rabin, another public-key cryptography is EL-Gamal.

→ EL-Gamal is based on the discrete logarithm problem

i) Key Generation:

→ Select a large prime P.

→ Select d which is private key such that $1 \leq d \leq P-2$

→ Select e_1 which is primitive root of P

→ calculate e_2 as

$$e_2 = e_1^d \bmod P$$

So,

Public key → (e_1, e_2, P)
Private key → d

ii) Encryption:

→ Select a random integer r.

→ $C_1 \leftarrow e_1^r \bmod p$

$C_2 \leftarrow (P \times e_2^r) \bmod p$

use of public-key (e_1, e_2, p)

We create two ciphertext

iii) Decryption:

→ use private key (d)

i.e.

$$P = [C_2 (C_1^d)^{-1}] \bmod P$$

$$= p \times e_2^r [(e_1^r)^d]^{-1} \quad (\because \text{from above } C_1 \text{ & } C_2)$$

$$= p \times e_2^r [(e_1^d)^r]^{-1}$$

$$= p \times e_2^r [(e_2)^{-r}] \quad (\because e_2 = e_1^d)$$

$$= P \quad \text{(plain-text)}$$

Bob

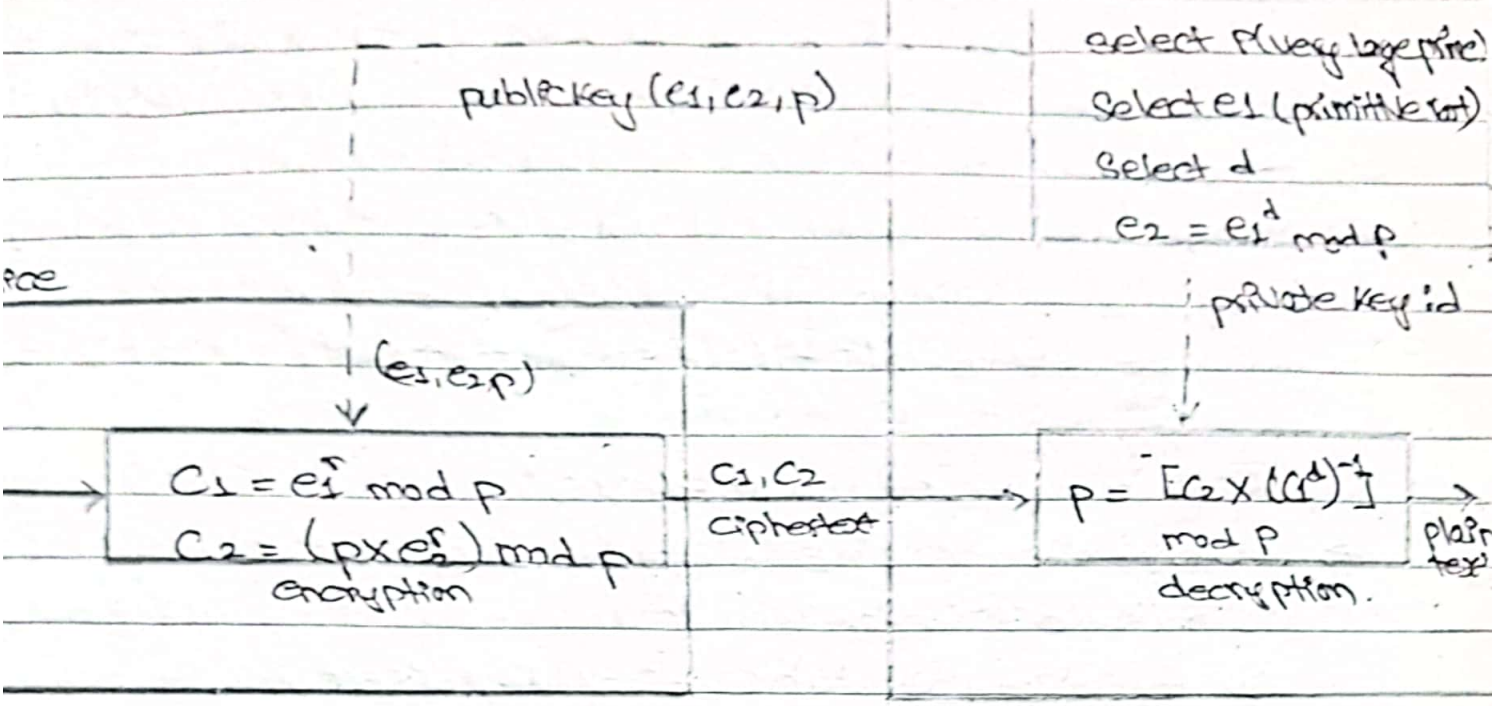


Fig of :- block diagram of Elgamal.

Let's see an example:

Let P = 11

where primitive root of P = e1 = 2

& d = 3

now, $e_2 = e_1^d = 2^3 = 8$

So public key = (e1, e2, p)

= (2, 8, 11)

let private key (d) = 3

& r = 4 & plaintext (P) = 7

So,

Encryption:

$C_1 = e_1^r \bmod 11$

$= 2^4 \bmod 11$

$= 5$

$C_2 = (P \times e_2^r) \bmod 11$

$= (7 \times 8^4) \bmod 11$

$= 6$

$(C_1, C_2) = (5, 6)$

Decryption:

$P = [C_2 (C_1^d)^{-1}] \bmod P$

$= [6 (5^3)^{-1}] \bmod 11$

$= 6 \times 5 \bmod 11$

$= 7$

Key Management & Public key Distribution:

→ Public-key encryption helps address key distribution problems.

→ There are basically two aspects of this:

i) distribution of public-keys,

ii) use of public-key encryption to distribute secret keys.

i) Distribution of Public-keys:

→ Public-key can be ~~distrib~~ distributed using one of:

1) public announcement,

2) publicly available ~~directo~~ directory,

3) public-key authority,

4) public-key certificates.

1) public announcement:

→ Users distribute public keys to recipients or broadcast to community at large

→ The major weakness is forgery i.e anyone can create a public key claiming to be someone else and broadcast it

2) Publicly Available Directory:

→ We can obtain greater security by registering keys with a public directory,

→ But it is still vulnerable to tampering or forgery

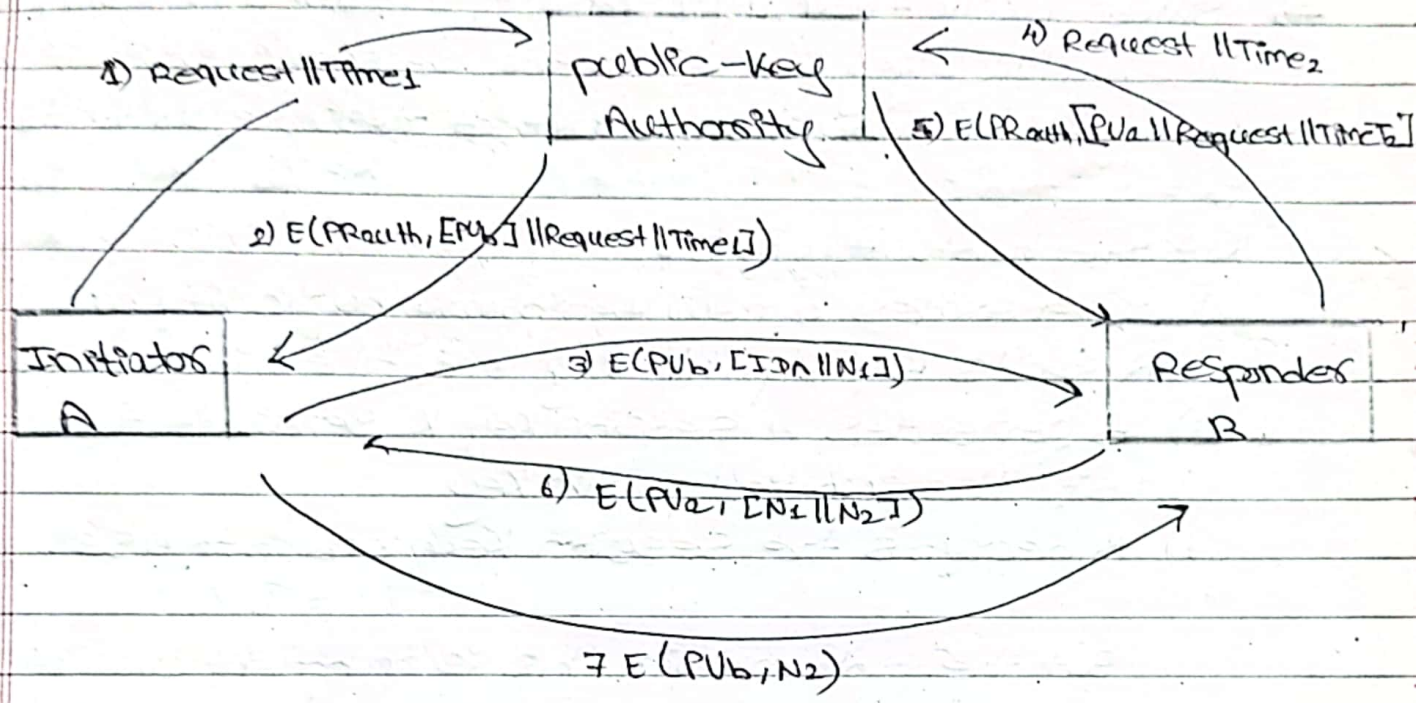
3) Public-key Authority:

→ It improves the security by tightening control over distribution of keys from directory

→ The users interact with public-key Authority/directory

to obtain any desired public key securely.

→ But it does require real-time access to directory when keys are needed.

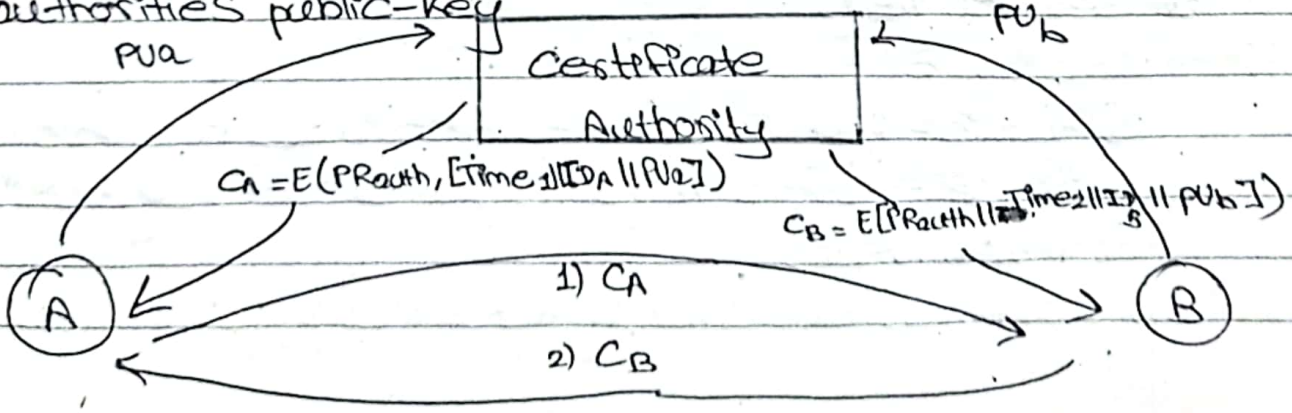


4) Public-key Certificates:

→ Certificates allow key exchange without real-time access to public-key authority.

→ A certificate binds identity to public key with all contents signed by a trusted public-key or certificate Authority (CA)

→ It can be verified by anyone who knowns the public-key authorities public-key



ii) Use of Public-key encryption to distribute of Secret key:

→ Use previous methods to obtain public key,
→ can use for secrecy or authentication,
→ Since public-key algorithms are slow. eg:- RSA, we use public-key encryption for session creation & use private-key encryption to protect message contents.

Let see how's it works:

i) A generates a new temporary public key pair,
ii) A sends B the public key and their identity,
iii) B generates a session key K sends it to A encrypted using the supplied public key
iv) A decrypts the session key and both use.

But the problem with this is an oppent can intercept and impersonate both halves of protocol (man-in-middle) attack.

So, the solution to this is;

Hybrid key distribution:

* Hybrid key distribution:

→ It uses of private-key KDC (Key Distribution Center).
→ It helps to share secret master key with each user.
→ User uses these secret master key to secure distribution of session key.

· Note:

① public-key is used to distribute masters keys
② These masters keys help in session key distribution between users
③ Messages are securely transmitted by help of session keys

There are different approaches of these using key distribution center.

i) A simple protocol using KDC,
ii) Needham-Schroeder Protocol (Use of Nonce)
iii) Denning-Sacco Protocol (Same as N-S Protocol but make use of Timestamp)
iv) Otway-Rees protocol

Note:- Refer to Forouzan slides chapter 15 for Block diagram & working Mechanism.

* Diffie-Hellman Key Exchange:

→ It is a practical method for public Exchange of a secret key
→ It cannot be used to exchange an arbitrary message,
→ rather used to establish a common key, which is only known to the two participants.

* How it works:

Choose a large Prime integer P
Find the primitive root of P. Let say α
Now, Let Assume two users A & B then

A	B
choose private key : X_A	choose private key : X_B
Compute public key $(Y_A) = \alpha^{X_A} \bmod P$	compute public key $(Y_B) = \alpha^{X_B} \bmod P$
Shared session key is generated as:	Session key generated as:
$K = Y_B^{X_A} \bmod P$	$K = Y_A^{X_B} \bmod p$
$= (\alpha^{X_B})^{X_A} \bmod p$	$= (\alpha^{X_A})^{X_B} \bmod p$
$\equiv$	

— But there is a possibility of man-in-the-middle attack in Diffie-Hellman i.e,

Man-in-the-middle Attack.

M (intruder)

A
X_A
$Y_A = a^{X_A} \bmod p$
$K = Y_m^{X_A} \bmod p$

X_m
$Y_m = a^{X_m} \bmod p$
$K = Y_A^{X_m} \bmod p$

$X_{m'}$
$Y_{m'} = a^{X_{m'}} \bmod p$
$K = Y_B^{X_{m'}} \bmod p$

B
X_B
$Y_B = a^{X_B} \bmod p$
$K = Y_{m'} \bmod p$