Traditional Substitution Cipher – A Mathematical Overview and its Implementation

Rupali Bhakkad[1] *, Nilesh Kajale*[2] *, Mohan Reddy Palugulla* [3]

[1] Department of MCA ,Marathwada Institute of technology,
Aurangabad, Maharashtra, India
rupali.bhakkad@gmail.com

[2] Department of MCA ,Marathwada Institute of technology,
Aurangabad, Maharashtra, India
nileshkkajale@gmail.com

[3] Department of MCA ,Marathwada Institute of technology,
Aurangabad, Maharashtra, India
mohan.reddy.pal @gmail.com

Abstract*: Exchanging information over internet become crucial part of our day today life,which includes e-banking, online transactions, online shopping and exchange of official and personal information, as this type of information's are sensitive , security of such critical information is becoming increasingly important. Information Security can be achieved with the help of cryptography. Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication. Cryptography has been divided in two types, Symmetric key cryptosystem and asymmetric key cryptosystem. In order to work with the current era's cryptosystem one should become familiars with the working of classical cryptosystem such as Shift cipher, affine cipher, substitution cipher and hill cipher techniques etc. This paper highlights the concept of classical symmetric key cryptosystem and its type and explains in detail elementary substitution cryptosystem (substitution Cipher) and polyalphabetic substitution cryptosystem (Hill Cipher) example with its mathematical overview and implementation in C++.*

Keywords: Modular Arithmetic, Modular Multiplicative Inverse, Matrix Inverse, Substitution Cipher, Hill Cipher

1. Introduction

Today's world is digital world, in this digital age confidential and sensitive data are frequently exchanged over the network. Such a sensitive and confidential data can be easily acquire by the advisory because the most of the information they acquire from a system or over the network is in a form that they can read and comprehend, advisory may reveal the information to other , modify it to misrepresent an individual or organization, or use it to launch an attack. Such problems can be reduced to the greater extent with the help of cryptography[1][15][16][20], which prevent advisory from being able to use the information that they get.

Cryptography is art of hiding data. In technical terms, Cryptography is the science of using mathematics to encrypt and decrypt data. Cryptography enables you to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. Data is made secure by converting the original data (referred as plain text) in to an unintelligent code (referred as cipher text). Sender generates the cipher text and sends it to the receiver. At the other end receiver generate the plain text from the cipher text. Here both sender and receiver use a key/key's to covert the plain text in to cipher text and vice versa.(Fig. 1.)[1][6][15][16][20]

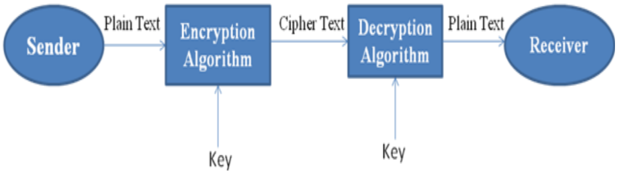


Figure 1: Cryptographic System

The process of converting plain text into cipher text by applying key on text is called as encryption and the process of converting cipher text in to plain text again by applying the inverse of key on the cipher text is called as decryption.[1][6][16][20] Key is a piece of information that determines the functional output of the algorithm.

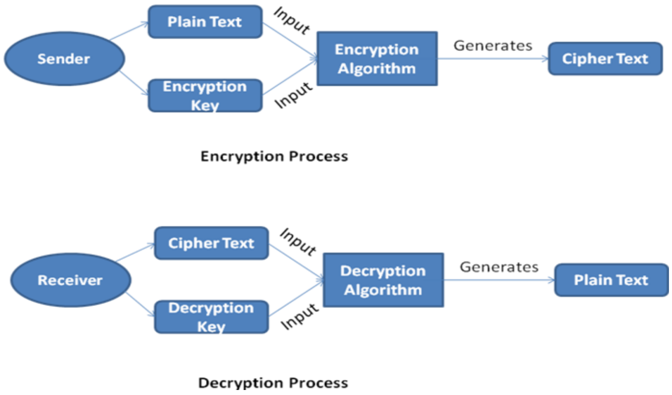


Figure 2: Encryption and Decryption Process

A cryptosystem (Cryptography System) is a pair of algorithm that takes a key and text as input and coverts plain text in to cipher text and back (i.e. encryption and decryption).Cryptosystem is classified into two, symmetric-key cryptosystem and Public-key cryptosystem/Asymmetric-key Cryptosystem.[2][7][13][20]

Symmetric cryptosystem uses a shared key for encryption and decryption process i.e. sender and receiver uses the same key for encryption and decryption process. In symmetric key cryptosystem sender and receiver either agreed upon a key before communication or sender send akey to the receiver by means of secure communication channel[1][3][16][20]. In symmetric-key cryptosystem security of data is totally depend upon the difficulty in finding key. Size of a key, type of a key and way of communication of a key plays an important role in providing security to a key. Example of Symmetric-key cryptosystem is AES (Advance Encryption Standard), BlowFish[4], TwoFish[5], etc.

Public-key cryptosystem uses a public key for encryption and private key for decryption i.e. the sender and receiver uses different key for encryption and decryption process. In Asymmetric-key cryptosystem public key is publicly made available which is used by the sender for encryption only where as receiver uses private key (does not known to anyone other than receiver) for decryption.[1][2][3][7][16][19][20] Examples of Public-key cryptosystems are RSA, Deffie Hellman Key Exchange Algorithm, Taher Elgamal Algorithm, Elliptical Curve, etc.

Classical cryptosystem are based on symmetric-key cryptosystem. Classical cryptosystem are broadly categorized in to following two category[1][3][16][18]

1. Substitution
2. Transposition

In substitution cipher plaintext characters are replaced by the other character(s) to generate cipher text and transposition cipher shuffles, rearranges or permutes the bit of block of plaintext to generate a cipher text.

Substitution ciphers are of two types[1][3][16][18]

1. Elementary Substitution cipher :
2. Polyalphabetic Substitution cipher :

In elementary substitution cipher one character is replaced at a time with one another character. Examples are Substitution cipher, Shift cipher and Affine cipher etc, whereas in polyalphabetic cipher more a set of characters are replaced at a time by another set of characters. Examples are Hill cipher, Vigenere cipher etc.

In this paper we are going to focus on substitution cipher technique from elementary substitution ciphers with its implementation in C++ and Hill cipher from polyalphabetic substitution cipher with its implementation in C++.

2. Number Theory

Before starting with Substitution techniques we will first overview the mathematical concept used for the implementation of these algorithms.

2.1 Divisibility

let a and b are two integers then a divides b (a|b) if there exists one integer c such that b=a*c and c=b/a here a is divisor of b.[7][20]

Example :

Let a = 5 and b =15
15=5 * 3 (here c = 3)
3=15/5

2.2 Prime numbers:

A number is said to be a prime number if it is divisible by 1 and itself only.[7][20]

Example :

The number 59 is divisible by 1 and itself so it is prime number.

2.3 Relatively Prime numbers:

Two numbers a and b are called relatively prime/co-prime if the greatest common divisor of a and b is 1 i.e. GCD(a,b)=1[7][20]

Example :

GCD(15,17)=1

The numbers 15 and 17 has only one common divisor and that is 1, so this numbers are called as relatively prime numbers

2.4 Prime Factorization:

Prime factorization is the decomposition of a composite number into smaller prime numbers, which when multiplied together equals the original integer .i.e.[7][20]

$$n=p_1^{e1}*p_2^{e2}*p_3^{e3}*\ldots\ldots.*p_n^{en} \quad (1)$$

Example :

Given m=100 find the prime factors of m.
100= 2*2*5*5 (Which can be represented as)
$100 = 2^2 * 5^2$

2.5 Modular Arithmetic:

Modulus” (abbreviated as "mod") is the Latin word for “remainder, residue” or more in “what is left after parts of the whole are taken”. Thus, "modular" or "mod arithmetic" is really "remainder arithmetic". More precise: We are looking for the integer that occurs as a remainder (or the "left-over") when one integers is divided by another integer[7][12][17][20]. Notation used for modular arithmetic (a is dividend, n is divisor and r is remainder) is

Example :

$$a \bmod n = r \quad (2)$$

1	17 mod 5 = 2	When 17 is divided by 5 then we get the remainder of 2
2	100 mod 6 = 4	When 100 is divided by 6 then we get the remainder of 4

2.6 Congruence:

Integers which leaves the same remainder after divided by m and if m completely divides the result of subtraction of those integers then such integers are called as congruent numbers[7][12][17][20]. Notation used for congruence i

$a \equiv b \pmod{m}$ (3)

Given an integer $m \geq 2$, we say that, 'a' is congruent to b modulo m, if

1. r1=r2
2. a-b mod m = 0

Example :

Given a=759, b=10074 and m=5 find whether $a \equiv b \pmod{m}$?

First we will find a (mod m), b (mod m) and a-b (mod m) as

r1 =a mod m	r2 = b mod m	(a-b) mod m
=759 mod 5	=10074 mod 5	=(759-10074) mod 5
= 4	= 4	= -9315 mod 5
		= 0

As we can see that r1=r2 and m completely divides a-b so we can say that 759 and 10074 are congruent numbers modulo m.

2.7 Greatest common divisor :

The greatest common divisor of two positive integers a and b is the largest divisor common to a and b[7][20] . It is Denoted as GCD(a,b)

Example :

GCD(100,70) =10

Here 10 is the highest common number which completely divides 100 and 70.

2.8 *Euler phi Function* :

Euler Phi function determines the count of numbers relatively prime to a given number. If m>=2 euler phi function is denoted as φ(m)[7][20].Formula to compute φ(m) is

$$\varphi(m) = m * \left(1 - \frac{1}{P_1}\right) * \left(1 - \frac{1}{P_2}\right) * \dots\dots\dots * \left(1 - \frac{1}{P_1}\right) \quad (4)$$

Example

Given m=100 find the numbers relatively prime to m.

Step1 Find the Prime factors of 100

$100 = 2^2 * 5^2$

Step2 Put values in Formula of Euler Phi

$$\begin{aligned}\varphi(100) &= 100 * \left(1 - \frac{1}{2}\right) * \left(1 - \frac{1}{5}\right)\\ &= 100 * \left(1 - \frac{1}{2}\right) * \left(1 - \frac{1}{5}\right)\\ &= 40\end{aligned}$$

2.9 Modular multiplicative inverse :

The modular multiplicative inverse of an integer a modulo m is an integer x such that $a^{-1} \equiv x \pmod{m}$ i.e it is the multiplicative inverse in the ring of integers modulo m. This is equivalent to $ax \equiv aa^{-1} \equiv 1 \pmod{m}$.The multiplicative inverse of a modulo m exists if and only if a and m are coprime (i.e., if gcd(a, m) = 1).[7][8][20]

Finding Modular Multiplicative Inverse using Euler phi function

$a * a^{-1} \equiv 1 \mod m$ (5)

$a^{-1} = a^{\varphi(m)-1} \mod m$ (6)

Where φ(m) is Euler phi function

Example:-

Given a=11 and m=26 find the modular multiplicative inverse of a.

$a*a^{-1} \pmod{m} = 11*11^{-1} \pmod{m}$ -----------------I

Step1 First step is to find numbers relatively prime to φ(26) = 13

Step2 $11^{-1} = 11^{\boldsymbol{\varphi(26)-1}} \mod 26$

$= 11^{13-1} \mod 26$

$= 11^{12} \mod 26$

$= 19$

Step3 Putting values of 11^{-1} in equation I

$1*19 \mod 26 \equiv 1$

Hence the Modular Multiplicative inverse of 11 mod 26 is 19

2.10 Permutation:

Permutation is the study of different arrangement created from the group of people or items where the order of people or items plays very important role .[10][20] This is usually written as ${}_nP_r$.

$${}_nP_r = \frac{n!}{(n-r)!} \quad (7)$$

Example:

Given group of 10 items out of which three items has to choose then find the number of combination in which item can be arranged.

Here r=3 and n=10

Putting this values in formula

$$\begin{aligned}10P_3 &= \frac{10!}{(10-3)!}\\ &= \frac{10!}{(7)!}\\ &= 720\end{aligned}$$

Hence the 3 item can be arranged in 720 combinations.

2.11 Matrix Inverse:

A n x n square matrix 'A' is called invertible if there exists an another n x n square matrix A^{-1} such that when A is multiplied to A^{-1} we will get n x n identity matrix I[9][11][20] . i.e.

Let **A=** $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ **and** $\mathbf{A^{-1}}$ = $\begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix}$ be 2* 2 square matrix, then

$A \text{ X } A^{-1} = I$ (8)

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \text{X} \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Here matrix A^{-1} is called as inverse of matrix A

Note: All square matrix are not invertible

Following are the steps to find the inverse of square matrix

Step1 Determinatant of matrix $|A| = (a * d - b * c)$ mod 26

Step2 Inverse of determinant of matrix

$|A|^{-1} = |A| * |A|^{-1}$ mod 26
(by the definition of modular inverse)

Step3 Adjoint Matrix = $\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$ mod 26

Step4 Inverse of matrix = Inverse of determinant of matrix * Adjoint of Matrix

Inverse of matrix $A^{-1}=|A|^{-1} * \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$ mod 26

Example :

Let $A\equiv\begin{bmatrix} 3 & 2 \\ 5 & 7 \end{bmatrix}$ be 2* 2 matrix, by using following steps we can find inverse of matrix

Step1 Determinatant of matrix $|A| = (3*7-2*5)$ mod 26
= 11 mod 26
= 11

Step2 Find $|A|^{-1}$ by using Euler Phi function

$|A|^{-1} = |A|^{\varphi(m)-1}$

$11^{-1} = 11^{\varphi(26)-1}$ mod 26
$= 11^{13-1}$ mod 26
$= 11^{12}$ mod 26
$= 19$

$|A| * |A|^{-1} \equiv 1$ mod 26
$11 * 19 \equiv 1$ mod 26

Step3 Adjoint Martix= $\begin{bmatrix} 7 & -2 \\ -5 & 3 \end{bmatrix}$ mod 26

Step4 Inverse of matrix =Inverse of determinant Of matrix*Adjoint of Matrix

Inverse of matrix A =19 * $\begin{bmatrix} 7 & -2 \\ -5 & 3 \end{bmatrix}$ mod 26

$=\begin{bmatrix} 133 & -38 \\ -95 & 57 \end{bmatrix}$ mod 26

$= \begin{bmatrix} 3 & 14 \\ 9 & 5 \end{bmatrix}$

3 SUBSTITUTION CIPHER TECHNIQUES :

3.1 Substitution Cipher

A substitution cipher is one in which letters are represented by other letters. In substitution cipher we can think of encryption and decryption as permutation of alphabetic character .In this we use lower case letters to represent plain text and upper case letter to represent cipher text. Puzzles cryptograms in news papers are example of substitution cipher[1][2][3][6][14][16][19][20].

3.1.1Algorithm Substitution Cipher (P, $\mathbb{Z}_{26}$)

Step1 Take P and C both to be the 26-letter English alphabet i.e. P = C = $\mathbb{Z}_{26}$.

Step2 K consists of all possible permutations of alphabetic characters for encryption and decryption of the 26 symbols 0, 1, 2, …, 25. Here lowercase letters are used to represent plain text and uppercase letters are used to represent cipher text.

Step3: For each permutation $\pi \in$ K, define Encryption rule as

$$e_\pi(x) = \pi(x),$$

Step4: For each π, π^{-1} is the inverse permutation to $\pi \in$ K, define Decryption rule as

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

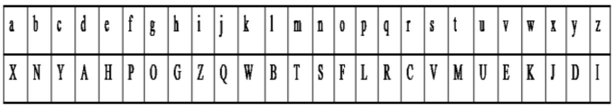
(Here inverse of permutation can be formed by writing the second line first and then sorting in alphabetical order.)

Example:

Consider our plaint text: is “secretemessage”

Encryption:

Encryption Key: Upper row represent plain text and lower row represent cipher text.



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

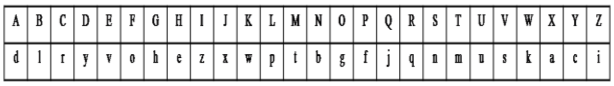
By using above key, each alphabet in plain text is replaced by its respective alphabet in lower row of encryption key (cipher text) so the encryption is

s	e	c	r	e	t	e	m	e	s	s	a	g	e
V	H	Y	C	H	M	H	T	H	V	V	X	O	H

Encrypted text: - VHYCHMHTHVVXOH

Decryption*:*

Decryption Key: Upper row represent cipher text and Lower row represent plain text. Decryption key is formed by writing second row of encryption key first and sorting it in alphabetical order.



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
d	l	r	y	v	o	h	e	z	x	w	p	t	b	g	f	j	q	n	m	u	s	k	a	c	i

By using above key, each alphabet in cipher text is replaced by its respective alphabet in lower row of decryption key (plain text) so the decryption is

V	H	Y	C	H	M	H	T	H	V	V	X	O	H
s	e	c	r	e	t	e	m	e	s	s	a	g	e

Decrypted text: - secretemessage

3.1.2 Implementation of Substitution Cipher in C++

```
//Program for substitution cipher
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()  //Main function
{
   char ch;
  //array to find position for encryption
   char alpha[]="abcdefghijklmnopqrstuvwxyz";
  //array to find position for decryption
char alpha1[]="XNYAHPOGZQWBTSFLRCVMUEKJDI";
 int i,j,choice,temp[100];
 clrscr();
 do
       {
          char input[100];
          cout<<"\nEnter the text:";
            //Input string
           gets(input);
          cout<<"\n1.Encrypt\n2.Decrypt\nEnter choice:";
          .
cin>>choice
switch(choice)
              {
        case 1:
           {
 //Encryption
                cout<<"\nEncrypted text : ";
                           //Encrypted Text
                for(i=0;i<input[i]!='\0';i++)
               cout<<alpha1[temp[i]];
                           break;
             }
        case 2:
                 {
                   //Decryption
                cout<<"\nDecrypted text : ";
                //Finding position of input string
                     for(i=0;i<input[i]!='\0';i++)
                        {
                   for(j=0;j<26;j++)
              {
                                  if(input[i]== alpha1[j])
               temp[i]=j;
              }   //Decrypted text
                    for(i=0;i<input[i]!='\0';i++)
                  cout<<alpha[temp[i]];
     break;      }     }
      cout<<"\nEnter choice to continue(Y/N):";
      cin>>ch;
      }while(ch=='y' || ch=='Y');
   getch();
}
```

```
Enter the text : secretemessage
1.Encrypt
2.Decrypt
Enter choice : 1
Encrypted  text : VHYCHMHTHVVXOH
Enter choice to continue(Y/N):y
Enter the text : VHYCHMHTHVVXOH
1.Encrypt
2.Decrypt
Enter choice : 1
Encrypted  text :secretemessage
```

3.2 Hill Cipher

This cipher was invented by Lesters S.Hill. Let m be a positive integer, and define P = e = $(\mathbb{Z}_{26})^m$,here m is the linear combinations of the m alphabetic charcaters in one plaintext elements and thus producing the m alphabettic characters in one ciphertext element For Example if m=2 then we could write plain text element as x=(x_1,x_2) and a cipher text element as y=(y_1,y2).[9][14][16][19][20]

3.2.1 Algorithm HillCipher(P, Z_{26}, m)

Step1 Take m X m matrix K as our key.This m X m encryption matrix must be invertible (Determinant of encryption matrix must not be zero, it must be relatively prime to size of alphabet i.e, 26). If ‘entry in row i and column j of K is $k_{i,j}$’ Then we write K=($k_{i,j}$)

Step2 Split the plain text P into blocks of m size. If ‘length of P is not divisible by m’ Then Add any character to the end of the string untill length of P is divisible by m.

Step3 Encryption
cipher text is obtained from the plain text by means of a linear tranformation i.e for x=(x1,…, x_m)∈ Pand K ∈ K we ,We compute y = $e_K(x)$ = xK as follows

$$(y_1, y_2, \ldots, y_m) = (x_1, x_2, \ldots, x_m)\begin{pmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,m} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,m} \\ \vdots & \vdots & & \vdots \\ k_{m,1} & k_{m,2} & \cdots & k_{m,m} \end{pmatrix} \bmod 26$$

Step4 Decryption –
For y = (y1,. . . ym) ∈ C
x = d_K(y) = yK^{-1} where,

$$K^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Example:
Consider our plain text is “secretemessage” and m=2

Choose a 2 * 2 encryption matrix i.e, K = $\begin{vmatrix} 3 & 2 \\ 5 & 7 \end{vmatrix}$

$\begin{vmatrix} 3 & 2 \\ 5 & 7 \end{vmatrix}$ = (3 * 7) - (5 * 2) = 11

Since, 11 is not equal to zero, so this matrix is invertible and 11 is also relatively coprime to mod 26.

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Now, by using above chart change the plain text alphabets into their respective numeric values and group them into block of size 2.

s	e	c	r	e	t	e	m	e	s	s	a	g	e
18	4	2	17	4	19	4	12	4	18	18	0	6	4

Encryption **:**
Now, multiplying each of these column vectors by the encryption matrix and take (mod 26) of the result; we get the

1.	$\begin{vmatrix} 3 & 2 \\ 5 & 7 \end{vmatrix}\begin{vmatrix} 18 \\ 4 \end{vmatrix} = \begin{vmatrix} 62\ (mod26) \\ 118(mod26) \end{vmatrix} = \begin{vmatrix} 10 \\ 14 \end{vmatrix} = \begin{vmatrix} K \\ O \end{vmatrix}$
2.	$\begin{vmatrix} 3 & 2 \\ 5 & 7 \end{vmatrix}\begin{vmatrix} 2 \\ 17 \end{vmatrix} = \begin{vmatrix} 40\ (mod26) \\ 129(mod26) \end{vmatrix} = \begin{vmatrix} 14 \\ 25 \end{vmatrix} = \begin{vmatrix} O \\ Z \end{vmatrix}$
3.	$\begin{vmatrix} 3 & 2 \\ 5 & 7 \end{vmatrix}\begin{vmatrix} 4 \\ 19 \end{vmatrix} = \begin{vmatrix} 50\ (mod26) \\ 153(mod26) \end{vmatrix} = \begin{vmatrix} 24 \\ 23 \end{vmatrix} = \begin{vmatrix} Y \\ X \end{vmatrix}$
4.	$\begin{vmatrix} 3 & 2 \\ 5 & 7 \end{vmatrix}\begin{vmatrix} 4 \\ 12 \end{vmatrix} = \begin{vmatrix} 36\ (mod26) \\ 104(mod26) \end{vmatrix} = \begin{vmatrix} 10 \\ 0 \end{vmatrix} = \begin{vmatrix} K \\ A \end{vmatrix}$
5.	$\begin{vmatrix} 3 & 2 \\ 5 & 7 \end{vmatrix}\begin{vmatrix} 4 \\ 18 \end{vmatrix} = \begin{vmatrix} 48\ (mod26) \\ 146\ (mod26) \end{vmatrix} = \begin{vmatrix} 22 \\ 16 \end{vmatrix} = \begin{vmatrix} W \\ Q \end{vmatrix}$
6.	$\begin{vmatrix} 3 & 2 \\ 5 & 7 \end{vmatrix}\begin{vmatrix} 18 \\ 0 \end{vmatrix} = \begin{vmatrix} 54\ (mod26) \\ 90\ (mod26) \end{vmatrix} = \begin{vmatrix} 2 \\ 12 \end{vmatrix} = \begin{vmatrix} C \\ M \end{vmatrix}$
7.	$\begin{vmatrix} 3 & 2 \\ 5 & 7 \end{vmatrix}\begin{vmatrix} 6 \\ 4 \end{vmatrix} = \begin{vmatrix} 26\ (mod26) \\ 58\ (mod26) \end{vmatrix} = \begin{vmatrix} 0 \\ 6 \end{vmatrix} = \begin{vmatrix} A \\ G \end{vmatrix}$

Encrypted text: KOOZYXKAWQCMAG
Decryption :

$$K^{-1}=\frac{1}{3X7-2X5}\begin{bmatrix}7 & -2\\ -5 & 3\end{bmatrix}=\frac{1}{11}\begin{bmatrix}7 & -2\\ -5 & 3\end{bmatrix}=11^{-1}\begin{bmatrix}7 & -2\\ -5 & 3\end{bmatrix}=19\begin{bmatrix}7 & -2\\ -5 & 3\end{bmatrix}=\begin{bmatrix}133 & -38\\ -95 & 57\end{bmatrix}=\begin{bmatrix}133\,(\text{mod}26) & -38(\text{mod}26)\\ -95\,(\text{mod}26) & 57(\text{mod}26)\end{bmatrix}=\begin{bmatrix}3 & 14\\ 9 & 5\end{bmatrix}$$

1.	$\begin{bmatrix}3 & 14\\ 9 & 5\end{bmatrix}\begin{bmatrix}10\\ 14\end{bmatrix}=\begin{vmatrix}226\ (\text{mod } 26)\\ 160\ (\text{mod } 26)\end{vmatrix}=\begin{vmatrix}18\\ 4\end{vmatrix}=\begin{vmatrix}s\\ e\end{vmatrix}$
2.	$\begin{bmatrix}3 & 14\\ 9 & 5\end{bmatrix}\begin{bmatrix}14\\ 25\end{bmatrix}=\begin{vmatrix}392\ (\text{mod } 26)\\ 251\ (\text{mod } 26)\end{vmatrix}=\begin{vmatrix}2\\ 17\end{vmatrix}=\begin{vmatrix}c\\ r\end{vmatrix}$
3.	$\begin{bmatrix}3 & 14\\ 9 & 5\end{bmatrix}\begin{bmatrix}24\\ 23\end{bmatrix}=\begin{vmatrix}394\ (\text{mod } 26)\\ 331\ (\text{mod } 26)\end{vmatrix}=\begin{vmatrix}4\\ 19\end{vmatrix}=\begin{vmatrix}e\\ t\end{vmatrix}$
4.	$\begin{bmatrix}3 & 14\\ 9 & 5\end{bmatrix}\begin{bmatrix}10\\ 0\end{bmatrix}=\begin{vmatrix}30\ (\text{mod } 26)\\ 90\ (\text{mod } 26)\end{vmatrix}=\begin{vmatrix}4\\ 12\end{vmatrix}=\begin{vmatrix}e\\ m\end{vmatrix}$
5.	$\begin{bmatrix}3 & 14\\ 9 & 5\end{bmatrix}\begin{bmatrix}22\\ 16\end{bmatrix}=\begin{vmatrix}290\ (\text{mod } 26)\\ 278\ (\text{mod } 26)\end{vmatrix}=\begin{vmatrix}4\\ 18\end{vmatrix}=\begin{vmatrix}e\\ s\end{vmatrix}$
6.	$\begin{bmatrix}3 & 14\\ 9 & 5\end{bmatrix}\begin{bmatrix}2\\ 12\end{bmatrix}=\begin{vmatrix}174\ (\text{mod } 26)\\ 78\ (\text{mod } 26)\end{vmatrix}=\begin{vmatrix}18\\ 0\end{vmatrix}=\begin{vmatrix}s\\ a\end{vmatrix}$
7.	$\begin{bmatrix}3 & 14\\ 9 & 5\end{bmatrix}\begin{bmatrix}0\\ 6\end{bmatrix}=\begin{vmatrix}84\ (\text{mod } 26)\\ 30\ (\text{mod } 26)\end{vmatrix}=\begin{vmatrix}6\\ 4\end{vmatrix}=\begin{vmatrix}g\\ e\end{vmatrix}$

Decrypted text: secretemessage

3.2.2 Implementation of Hill Cipher in C++

```
//Program for Hill Cipher
#include<iostream.h>   #include<stdio.h>
#include<conio.h> #include<string.h>
#include<stdlib.h>
int  Euclidean(int a,int b)     {
        int gcd;  int count = 0;
        if(a>b)       {
               int temp = a;
               a = b;
               b = temp;       }
        int r = 1;
        while(r >=1 && gcd >=1 )          {
               gcd = r;     r = a % b;
               a = b;    b = r;
              if(r <=0)        {
                     count++;
             break;     }        }
         if(count > 0
            return gcd;
        else
            return r;       }/Function to calculate mod
int mod(int x,int y)       {
      int q,rem;      q=x/y;
      if(q<0)     q=q*(-1);
      if(x>=0 && q>=0)
         rem=x-(q*y);
     else       {
          q=q+1;    rem=x-(-q*y);}
          return rem;            }
     for(j=0;j<2;j++)       {
           cin>>array[i][j];
```

```
  ain[1][1]=array[0][0];
  temp=array[0][0]*array[1][1]-array[0][1]*array[1][0];
 /determinant    temp=modInv(temp,26);
                     if(temp==-1)
                     cout<<"\n Inverse cannot be found";
                     else    {
                      for(int i=0;i<2;i++)    {
                      for(int j=0;j<2;j++)   {
                      ain[i][j]*=temp;
                      ain[i][j]=mod(ain[i][j],
                      for(i=0;i<input[i]!='\0';i++)
                      for(j=0;j<alpha[j]!='\0';j++) {
                      if(alpha[j]==input[i])     {
                     encryp[k]=j;
                      k++;
                           break;   }  }  }
                     int nk = k;
                     if(k%2!=0)      {
                     encryp[k]=25;   // Same sized block
                     k++;    }
                     for(i=0;i<k;i=i+2)   {
                    int arr[2];
                  arr[0]=encryp[i];    arr[1]=encryp[i+1];
                 encryp[i]=arr[0]*ain[0][0]+arr[1] *ain[0][1];
                 encryp[i]=encryp[i]%26;
                 encryp[i+1]=arr[0]*ain[1][0]+arr[1]*ain[1][1];
                 encryp[i+1]=encryp[i+1]%26;     }
                 if(nk%2 !=0)     nk = nk-1;
                 for(i=0;i<nk;i++)   {
                 cout<<alpha[encryp[i]];
                 break;  }    }
            cout<<"\nEnter choice to continue(Y/N):";
            cin>>ch;   }
          else
          cout<<"\nEntered matrix is invalid "; //exit(0);
                }while(ch=='y' || ch=='Y');
                getch();     }
```

Output of Hill Cipher

```
//Finding modular inverse
int modInv(int n,int m)
 int i,inv=1,count=0;
 for(i=1;i<=m;i++)             {
 if(((n%i)==0) && (m%i==0))
 count++;
 if(count==1)                {
  i=1;
                do    {
                      if((n*i)%m==1)
                      break;
                       i++;
                 }while(inv==1);
                return i;              }
            else
                return -1;       }
void main()    { //Main funct
char ch;
/array to find position for encryption
 charalpha[]="abcdefghijklmnopqrstuvwxyz”;
/array to find position for decryption
 char b[]="ABCDEFGHIJKLMNOPQRSTUVWXTZ";
  int i,j,choice,k=0;
  int *encryp;
  int determinent = 0;
  int array[2][2];
   clrscr();
    do               {
   char input[100];
   cout<<"\nEnter the text:";
   gets(input);      //Input string
   cout<<"\nEnter the 2 * 2 multiplication  matrix:";
    for(i=0;i<2;i++)                  {    //Multiplication matrix
 determinent=(array[0][0]*array[1][1])-array[0][1]*array[1][0])
//Calculating determinent must not be zero
   if(determinent != 0 &&
  Euclidean(determinent,26)==1)       {
   cout<<"\n1.Encrypt\n2.Decrypt\nEnter choice:";
   cin>>choice;
   switch(choice)    {
   case 1:   {   //Encryption
    cout<<"\nEncrypted text : ";
                       for(i=0;i<input[i]!='\0';i++)    {
                        for(j=0;j<alpha[j]!='\0';j++)   {
                          if(alpha[j]==input[i])    {
                              k++;
                              break;  }   }    }
                      if(k%2!=0)  { //Same  sized block
                        encryp[k]=25;
                        k++;   }
                      for(i=0;i<k;i=i+2)  //Encryption logic       {
                        int arr[2];
                        arr[0]=encryp[i];    arr[1]=encryp[i+1];
                        encryp[i] = arr[0]*array[0][0] + arr[1] *
                                   array[0][1];
                        encryp[i]=encryp[i]%26;
encryp[i+1]=arr[0]*array[1][0]+arr[1]*array[1][1];
                         encryp[i+1]=encryp[i+1]%26;  }
                           for(i=0;i<k;i++)
                             cout<<b[encryp[i]];
                      break;   }   }
                     case 2:    {   //Decryption
                        cout<<"\nDecrypted text : ";
                        int ain[2][2],temp; //Matrix inverse
                        ain[0][0]=array[1][1 ];
                        ain[0][1]=-array[0][1];
                       ain[1][0]=-array[1][0];
```

```
Enter the text : secretemessage
Enter the 2*2 multiplication matrix :
3 2
5 7
1.Encrypt
2.Decrypt
Enter choice : 1
Encrypted text KOOZYXKAWQCMAG
Enter choice to continue(Y/N):y
Enter the text : KOOZYXKAWQCMAG
Enter the 2*2 multiplication matrix :
3 2
5 7
1.Encrypt
2.Decrypt
Enter choice : 2
Decrypted text : secretemessage
Enter choice to continue(Y/N):y
```

4 Conclusion

Data security is achieved with the help of the cryptography. In this paper we tried to focus on working of traditional substitution cipher technique and explained it with the help of examples and its implementation in c++, which will help us to understand the modern era’s cryptosystem. In this paper we also tried to explain some mathematical concept from the number theory on which this algorithm works.

References

[1] Atul Kahate “Cryptography and Network Security” 2nd Edition

[2] Avi Kak (kak@purdue.edu) ”Lecture Notes on “Computer and Network Security”

[3] Bernard Menezes “Network Security and Cryptography” CENGAGE Learning

[4] Bruce Schneier,”*The Blowfish Encryption Algorithm”*, Dr. Dobb’s Journal of Software Tools, pp. 4, 38, 40, 98, 99, 1994.

[5] Bruce Schneier∗ John Kelsey† Doug Whiting‡ David Wagner§ Chris Hall¶ Niels Ferguson k 15 June 1998 “Twofish: A 128-Bit Block Cipher ,”

[6] Bruice Schineier ,“Applied Cryptography”.

[7] Chitra Desai,” *A Novel Approach for Digital Signature Scheme Based on Solving Two Hard Problems”* IJCMSA: Vol. 6, No. 3-4, July-December 2012, pp. 95– 100.

[8] Chitra G.Desai ,Rupali Bhakkad ,Sonal Sarnaik, ”Identifying Quadratic Residuity Using Legendre-Jacobi Symbol”.

[9] Chris Christensen ,Fall 2006 ,MAT/CSC 483 ,The Hill cipher

[10] Dr. S.A.M Rizvi, Neeta Wadhwa ,Dept. of Comp. Sc., Jamia Milia Islamia, NewDelhi , “Analysis of Substitution and Permutation from Cryptanalysis Perspective”

[11] Graybill, Franklin A. (1969). Introduction to Matrices with Applications in Statistics, second edition, Belmont, CA: Wadsworth.

[12] http://www2.sunysuffolk.edu/fultonj/MA22/Modular%20Arithmetic%20&%20Cryptography.pdf

[13] Joan Daemen and Vincent Rijmen,” *Rijndael for AES”.,* AES Candidate Conference, pp. 343–348, 2000.

[14] Jonathan Katz, Yehuda Lindel ,”Introduction to modern cryptography”, l, CRC Press

[15] Luca Trevisan ,“Cryptography “,Lecture Notes from CS276, Spring 2009 ,Stanford University.

[16] Menezes, Alfred J; van Oorschot, Paul C.; Vanstone, Scott A. (2001), “Handbook of Applied Cryptography”,[Online]

[17] Miguel a. Lerma “Modular Arithmetic”

[18] Monika Agrawal, Pradeep Mishra, ”A Comparative Survey on Symmetric Key Encryption Techniques”

[19] Shafi Goldwasser, Mihir Bellare July 2008,”Lecture Notes on Cryptography”.

[20] Stinson, Douglas Robert (2006),"*Cryptography: Theory and Practice (3rd ed.)*", London: CRC Press

Author Profile

Ms.Rupali Bhakkad is basically a science graduate and completed her Masters of Computer Applicationfrom Institute of Management studies and Information Technology Aurangabad in 2009. She is presently working as the Assistant Professor in the Department of MCA at Marathwada Institute of Technology (Engineering) from last four and half years. Her area of interest includes Operating System,Linux and Cryptography. She has 1 international publications in journals

Nilesh K. Kajale has completed his bachelor’s degree in Computer Science and pursuing Masters of Computer Application (Final Year)from Marathwada Institute of Technology (Engineering), Aurangabad. His area of interest includes Network Security and Cryptography.

Mohan Reddy Palugulla completed his bachelor’s degree in Computer Science and pursuing Masters of Computer Application (Final Year) from Marathwada Institute of Technology (Engineering), Aurangabad. His research interest includes Information system, cloud computing and Cryptography. He has one international publications in journals