VISVESVARAYA TECHNOLOGICAL UNIVERSITY
Jnana Sangama, Belagavi-590010

MINI PROJECT REPORT
ON

" IMPLEMENTATION OF CRYPTOGRAPHIC HILL CIPHER ALGORITHM "

Submitted in partial fulfillment for the requirements for the sixth semester assignment

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

For the Academic year 2019-2020

Submitted by:

1MV17CS079- Prakruthi M
1MV17CS106- Somya
1MV17CS110- Srishti Nema
1MV17CS128- Pratilipi Aich

Project carried out at:
Sir M. Visvesvaraya Institute Of Technology
Bangalore - 562157

Sir M. Visvesvaraya Institute Of Technology ,Bangalore
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY
HUNASAMARANAHALLI, BANGALORE- 562157

CONTENTS

INTRODUCTION AND PROBLEM STATEMENT

Polygraphic substitution is a cipher in which a uniform substitution is performed on blocks of letters. When the length of the block is specifically known, more precise terms are used: for instance, a cipher in which pairs of letters are substituted is bigraphic.

In classical cryptography, the Hill cipher is a polygraphic substitution cipher based on linear algebra. Invented by Lester S. Hill in 1929, it was the first polygraphic cipher in which it was practical (though barely) to operate on more than three symbols at once. This cipher is usually based on matrices and linear algebra. Each letter is represented by a number modulo 26. Often the simple scheme A = 0, B = 1, …, Z = 25 is used, but this is not an essential feature of the cipher.

To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n × n matrix, against modulus 26. To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption.

The matrix used for encryption is the cipher key, and it should be chosen randomly from the set of invertible n × n matrices (modulo 26). The cipher can, of course, be adapted to an alphabet with any number of letters; all arithmetic just needs to be done modulo the number of letters instead of modulo 26.

Idea and Problem Statement tackled- Implementation of Hill cipher in a programming language and a user friendly web page where easily a plaintext can be encrypted to ciphertext and ciphertext can be decrypted back to plaintext easily with the help of textboxes and buttons.

ALGORITHM

The Hill cipher algorithm progresses in the following stages:

- Organize character alphabetically with numeric A→1, B→2,….,Z→26 or in ASCII characters (256 characters).
- Create a key matrix measuring m x m.
- Matrix K is an invertible matrix that has multiplicative inverse K^{-1} so that KK^{-1}.
- Plaintext P=p1 p2…. Pn, blocked with the same size as the row or column K.
- Transpose matrix P which is denoted by P' and multiply matix K with transposed P in modulo 26 or 256 and denote it by C'. This means. C'=KP'.
- Transpose C' to get back C.
- Change the result of the previous step into alphabet using alphabetical correspondence with numeric in the first step to obtain ciphertext.
- To understand this the following example can be considered.

Let the message to be enciphered be "ATTACK AT DAWN". First three characters are taken and a 3 x 3 matrix is considered. To consider first three characters we change them to number matrix, so, ['A' 'T' 'T'] becomes [0 19 19]. Let the key be as follows:

$$\begin{bmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 17 & 7 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 4 & 5 \\ 9 & 2 & 1 \\ 3 & 17 & 7 \end{bmatrix} \begin{bmatrix} 0 \\ 19 \\ 19 \end{bmatrix} = \begin{bmatrix} 171 \\ 57 \\ 456 \end{bmatrix} \pmod{26} = \begin{bmatrix} 15 \\ 5 \\ 14 \end{bmatrix} = \text{'PFO'}$$

So 'A T T' changes 'P F O'. To get back the 'A T T', the decryption key is K^{-1}.

$$K^{-1} \begin{bmatrix} 15 \\ 5 \\ 14 \end{bmatrix} \pmod{26} = \begin{bmatrix} 0 \\ 19 \\ 19 \end{bmatrix} = \text{'ATT'}$$

Thus we get back this 'ATT' after decryption.

IMPLEMENTATION AND CODE SNIPPETS

The implementation of the algorithm is done in Java and the web interface is made using HTML,CSS, Javascript, Bootstrap and Jquery. The code snippets for HillCipherExample.java, HillCipher.java and index.html showing the successful implementation .

1. HillCipherExample.java

```
package
college;
            import java.util.*;
            import java.io.BufferedReader;
            import java.io.IOException;
            import java.io.InputStreamReader;
            public class HillCipherExample {
                int[] lm;
                int[][] km;
                int[] rm;
                static int choice;
                int [][] invK;
                public void performDivision(String temp, int s)
                {
                    while (temp.length() > s)
                    {
                        String line = temp.substring(0, s);
                        temp = temp.substring(s, temp.length());
                        calLineMatrix(line);
                        if(choice ==1){
                            multiplyLineByKey(line.length());
                        }else{
                            multiplyLineByInvKey(line.length());
                        }
                        showResult(line.length());
                    }
                    if (temp.length() == s){
                        if(choice ==1){
                            calLineMatrix(temp);
                            multiplyLineByKey(temp.length());
                            showResult(temp.length());
                        }
                        else{
                            calLineMatrix(temp);
```

```
            this.multiplyLineByInvKey(temp.length());
            showResult(temp.length());
        }
    }
    else if (temp.length() < s)
    {
        for (int i = temp.length(); i < s; i++)
            temp = temp + 'x';
        if(choice ==1){
            calLineMatrix(temp);
            multiplyLineByKey(temp.length());
            showResult(temp.length());
        }
        else{
            calLineMatrix(temp);
            multiplyLineByInvKey(temp.length());
            showResult(temp.length());
        }
    }
}
public void calKeyMatrix(String key, int len)
{
    km = new int[len][len];
    int k = 0;
    for (int i = 0; i < len; i++)
    {
        for (int j = 0; j < len; j++)
        {
            km[i][j] = ((int) key.charAt(k)) - 97;
            k++;
        }
    }
}
public void calLineMatrix(String line)
{
    lm = new int[line.length()];
    for (int i = 0; i < line.length(); i++)
    {
        lm[i] = ((int) line.charAt(i)) - 97;
    }
}
public void multiplyLineByKey(int len)
{
    rm = new int[len];
    for (int i = 0; i < len; i++)
    {
        for (int j = 0; j < len; j++)
```

```
        {
            rm[i] += km[i][j] * lm[j];
        }
        rm[i] %= 26;
    }
}
public void multiplyLineByInvKey(int len)
{
    rm = new int[len];
    for (int i = 0; i < len; i++)
    {
        for (int j = 0; j < len; j++)
        {
            rm[i] += invK[i][j] * lm[j];
        }
        rm[i] %= 26;
    }
}
public void showResult(int len)
{
    String result = "";
    for (int i = 0; i < len; i++)
    {
        result += (char) (rm[i] + 97);
    }
    System.out.print(result);
}
public int calDeterminant(int A[][], int N)
{
    int resultOfDet;
    switch (N) {
        case 1:
            resultOfDet = A[0][0];
            break;
        case 2:
            resultOfDet = A[0][0] * A[1][1] - A[1][0] * A[0][1];
            break;
        default:
            resultOfDet = 0;
            for (int j1 = 0; j1 < N; j1++)
            {
                int m[][] = new int[N - 1][N - 1];
                for (int i = 1; i < N; i++)
                {
                    int j2 = 0;
                    for (int j = 0; j < N; j++)
                    {
```

```
                        if (j == j1)
                            continue;
                        m[i - 1][j2] = A[i][j];
                        j2++;
                    }
                }
                resultOfDet += Math.pow(-1.0, 1.0 + j1 + 1.0) * A[0][j1]
                        * calDeterminant(m, N - 1);
            }   break;
        }
        return resultOfDet;
    }
    public void cofact(int num[][], int f)
    {
        int b[][], fac[][];
        b = new int[f][f];
        fac = new int[f][f];
        int p, q, m, n, i, j;
        for (q = 0; q < f; q++)
        {
            for (p = 0; p < f; p++)
            {
                m = 0;
                n = 0;
                for (i = 0; i < f; i++)
                {
                    for (j = 0; j < f; j++)
                    {
                        b[i][j] = 0;
                        if (i != q && j != p)
                        {
                            b[m][n] = num[i][j];
                            if (n < (f - 2))
                                n++;
                            else
                            {
                                n = 0;
                                m++;
                            }
                        }
                    }
                }
                fac[q][p] = (int) Math.pow(-1, q + p) * calDeterminant(b, f - 1);
            }
        }
        trans(fac, f);
    }
```

```
void trans(int fac[][], int r)
{
    int i, j;
    int b[][], inv[][];
    b = new int[r][r];
    inv = new int[r][r];
    int d = calDeterminant(km, r);
    int mi = mi(d % 26);
    mi %= 26;
    if (mi < 0)
        mi += 26;
    for (i = 0; i < r; i++)
    {
        for (j = 0; j < r; j++)
        {
            b[i][j] = fac[j][i];
        }
    }
    for (i = 0; i < r; i++)
    {
        for (j = 0; j < r; j++)
        {
            inv[i][j] = b[i][j] % 26;
            if (inv[i][j] < 0)
                inv[i][j] += 26;
            inv[i][j] *= mi;
            inv[i][j] %= 26;
        }
    }
    invK = inv;
}
public int mi(int d)
{
    int q, r1, r2, r, t1, t2, t;
    r1 = 26;
    r2 = d;
    t1 = 0;
    t2 = 1;
    while (r1 != 1 && r2 != 0)
    {
        q = r1 / r2;
        r = r1 % r2;
        t = t1 - (t2 * q);
        r1 = r2;
        r2 = r;
        t1 = t2;
        t2 = t;
```

```
    }
    return (t1 + t2);
}
public boolean check(String key, int len)
{
    calKeyMatrix(key, len);
    int d = calDeterminant(km, len);
    d = d % 26;
    if (d == 0)
    {
        System.out.println("Key is not invertible");
        return false;
    }
    else if (d % 2 == 0 || d % 13 == 0)
    {
        System.out.println("Key is not invertible");
        return false;
    }
    else
    {
        return true;
    }
}
public static void main(String args[]) throws IOException
{
    HillCipherExample obj = new HillCipherExample();
    BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Menu:\n1: Encryption\n2: Decryption");
    choice = Integer.parseInt(in.readLine());
    System.out.println("Enter the line: ");
    String line = in.readLine();
    System.out.println("Enter the key: ");
    String key = in.readLine();
    double sq = Math.sqrt(key.length());
    if (sq != (long) sq)
        System.out.println("Cannot Form a square matrix");
    else
    {
        int size = (int) sq;
        if (obj.check(key, size))
        {
            System.out.println("Result:");
            obj.cofact(obj.km, size);
            obj.performDivision(line, size);
        }
    }
}
```

```
}
```

2. Front end code- Index.html

The landing page of the implementation is given by the following code:

```
<html>
        <head>
            <meta charset="utf-8">
            <meta http-equiv="X-UA-Compatible" content="IE=edge">
            <meta name="viewport" content="width=device-width, initial-scale=1">
            <title>HILL CIPHER </title>
        <!--
        Strategy Template
        http://www.templatemo.com/tm-489-strategy
        -->
            <!-- load stylesheets -->
            <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open+Sans:300,400">  <!-
        - Google web font "Open Sans" -->
            <link rel="stylesheet" href="font-awesome-4.5.0/css/font-awesome.min.css">                <!-
        - Font Awesome -->
            <link rel="stylesheet" href="css/bootstrap.min.css">                                      <!-
        - Bootstrap style -->
           <link rel="stylesheet" href="css/templatemo-style.css">                                   <!--
        Templatemo style -->
            <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
            <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
                <!--[if lt IE 9]>
                  <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
                  <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
                  <![endif]-->
        </head>
        <script type="text/javascript">
            function Encrypt(){
                plaintext = document.getElementById("p").value.toLowerCase().replace(/[^a-z]/g, "");
                k = document.getElementById("k").value.toLowerCase().replace(/[^0-9 ]/g, "");
                keys = k.split(" ");
                // do some error checking
                if(plaintext.length < 1){ alert("please enter some plaintext (letters and numbers
        only)"); return; }
                if(plaintext.length % 2 == 1){ plaintext = plaintext + "x"; }
                if(keys.length != 4){ alert("key should consist of 4 integers"); return; }
                for(i=0;i<4;i++) keys[i] = keys[i]%26;
                ciphertext="";
                for(i=0; i<plaintext.length; i+=2){
                    ciphertext += String.fromCharCode((keys[0]*(plaintext.charCodeAt(i)-97) +
```

```
keys[1]*(plaintext.charCodeAt(i+1)-97))%26 + 97);
            ciphertext += String.fromCharCode((keys[2]*(plaintext.charCodeAt(i)-97) +
keys[3]*(plaintext.charCodeAt(i+1)-97))%26 + 97);
        }
        document.getElementById("c").value = ciphertext;
    }
function Decrypt(){
    ciphertext = document.getElementById("c").value.toLowerCase().replace(/[^a-z]/g, "");
    k = document.getElementById("k").value.toLowerCase().replace(/[^0-9 ]/g, "");
    keys = k.split(" ");
    // do some error checking
    if(ciphertext.length < 1){ alert("please enter some ciphertext (letters only, numbers should
be spelled)"); return; }
    if(ciphertext.length % 2 == 1){ alert("ciphertext is not divisible by 2 (wrong algorithm?)");
return; }
    if(keys.length != 4){ alert("key should consist of 4 integers"); return; }
    for(i=0;i<4;i++) keys[i] = keys[i]%26;
    // calc inv matrix
    det = keys[0]*keys[3] - keys[1]*keys[2];
    det = ((det%26)+26)%26;
    di=0;
    for(i=0;i<26;i++){ if((det*i)%26 == 1) di = i; }
    if(di == 0){alert("could not invert, try different key"); return; }
    ikeys = new Array(4);
    ikeys[0] = (di*keys[3])%26; ikeys[1] = (-1*di*keys[1])%26;
    ikeys[2] = (-1*di*keys[2])%26; ikeys[3] = di*keys[0];
    for(i=0;i<4;i++){ if(ikeys[i] < 0) ikeys[i] += 26; }
    plaintext="";
    for(i=0; i<ciphertext.length; i+=2){
        plaintext += String.fromCharCode((ikeys[0]*(ciphertext.charCodeAt(i)-97) +
ikeys[1]*(ciphertext.charCodeAt(i+1)-97))%26 + 97);
        plaintext += String.fromCharCode((ikeys[2]*(ciphertext.charCodeAt(i)-97) +
ikeys[3]*(ciphertext.charCodeAt(i+1)-97))%26 + 97);
    }
    document.getElementById("p").value = plaintext;
}
</script>
    <body>
        <section class="cd-hero">
            <ul class="cd-hero-slider autoplay">
            <!--
                <ul class="cd-hero-slider autoplay"> for slider auto play
                <ul class="cd-hero-slider"> for disabled auto play
            -->
                <li class="selected">
                    <div class="cd-full-width">
                        <div class="tm-slide-content-div">
```

```
                        <form action="index.html" id="search-form">

                            <h2 class="text-uppercase">HILL CHIPHER</h2>
                            <p class="m-b-mid">Demonstration of Encryption and Decryption</p>
                            <div class="form-group">

                                    <p>PLAIN TEXT<br>
                                    <textarea id="p" name="p" rows="4" cols="50"></textarea>
                                    <p>KEY = <input id="k" name="k" size="40" value="5 17 4
15" type="text">

                                        <p><input name="e" value="ENCRYPT"
onclick="Encrypt()" type="button">

                                            <input name="d" value="DECRYPT"
onclick="Decrypt()" type="button"></p>

                                                <p>CIPHER TEXT<br>
                                                <textarea id="c" name="c" rows="4"
cols="50"></textarea> </p>


                        </form>
                    </div>
                </div> <!-- .cd-full-width -->
            </li>
            </div>




        </section>
        <!-- load JS files -->
        <script src="js/jquery-1.11.3.min.js"></script>             <!-- jQuery
(https://jquery.com/download/) -->
        <script src="https://www.atlasestateagents.co.uk/javascript/tether.min.js"></script> <!--
Tether for Bootstrap (http://stackoverflow.com/questions/34567939/how-to-fix-the-error-error-
bootstrap-tooltips-require-tether-http-github-h) -->
        <script src="js/bootstrap.min.js"></script>                 <!-- Bootstrap js (v4-
alpha.getbootstrap.com/) -->
        <script src="js/hero-slider-script.js"></script>            <!-- Hero slider
(https://codyhouse.co/gem/hero-slider/) -->
        <script src="js/jquery.touchSwipe.min.js"></script>         <!--
http://labs.rampinteractive.co.uk/touchSwipe/demos/ -->
        <script>

            $(document).ready(function(){
                /* Auto play bootstrap carousel
                * http://stackoverflow.com/questions/13525258/twitter-bootstrap-carousel-
autoplay-on-load
```

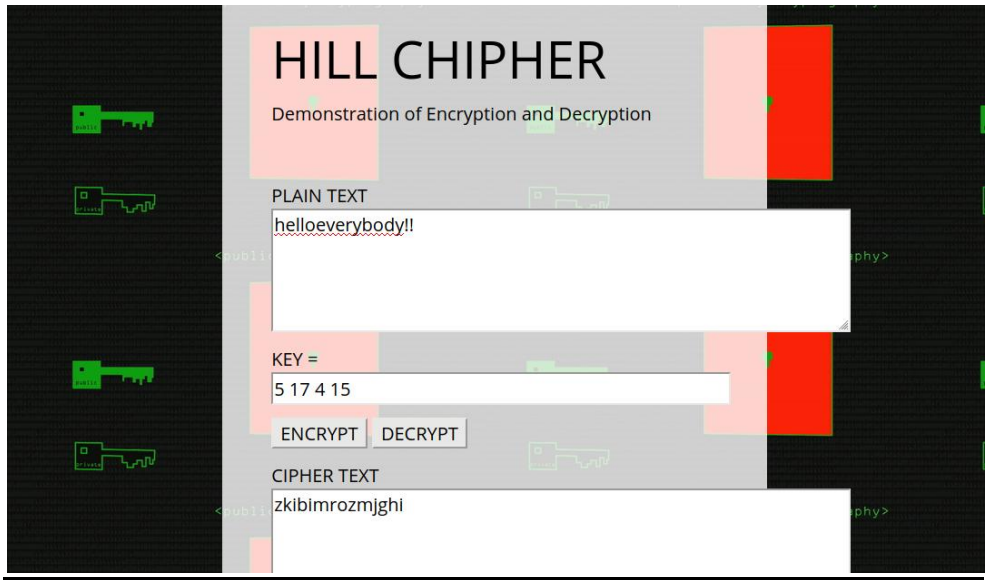
```
                ------------------------------------------------------------------------------
--------*/
                $('.carousel').carousel({
                  interval: 3000
                })
                /* Enable swiping carousel for tablets and mobile
                 * http://lazcreative.com/blog/adding-swipe-support-to-bootstrap-carousel-3-0/
                 ------------------------------------------------------------------------------
-*/
                if($(window).width() <= 991) {
                    $(".carousel-inner").swipe( {
                        //Generic swipe handler for all directions
                        swipeLeft:function(event, direction, distance, duration, fingerCount) {
                            $(this).parent().carousel('next');
                        },
                        swipeRight: function() {
                            $(this).parent().carousel('prev');
                        },
                        //Default is 75px, set to 0 for demo so any distance triggers swipe
                        threshold:0
                    });
                }
                /* Handle window resize */
                $(window).resize(function(){
                    if($(window).width() <= 991) {
                        $(".carousel-inner").swipe( {
                            //Generic swipe handler for all directions
                            swipeLeft:function(event, direction, distance, duration, fingerCount)
{

                                $(this).parent().carousel('next');
                            },
                            swipeRight: function() {
                                $(this).parent().carousel('prev');
                            },
                            //Default is 75px, set to 0 for demo so any distance triggers swipe
                            threshold:0
                        });
                     }
                });
            });
        </script>
</body>
</html>
```

The entire code is updated on the GitHub and is open sourced here: https://github.com/prakruthi3/Hill-Cipher .

RESULTS AND SCREENSHOTS

1. Encryption



2. Decryption