IT 8761 NETWORK SECURITY LABORATORY

SEMESTER PRACTICAL EXAMINATION

QUESTION:

Develop a java program to implement the Rail fence Cipher with depth 5 and reapply the same algorithm with depth 3 on the intermediate cipher and generate the final cipher text.

21-12-20 (1).

IT 8761 - Network Security Laboratory

Semester Practical Examination

NAME : VAISHALI.R

REG-NO : 312217104181

DATE : 21-12-2020

SESSION: AN

Aim:

To develop a Java program to implement the Rail fence cipher with depth 5 and reapply the same algorithm with depth 3 on the intermediate cipher and generate the final cipher text.

Algorithm:

i) Read the plaintext.
ii) Initialize a matrix with no. of rows equal to the depth 5 and no. of columns equal to length of the plaintext.
iii) Fill the matrix with the plaintext characters in the downward diagonal direction.
iv) On reaching the last row, fill in the upward diagonal direction.
v) Repeat steps (iii) and (iv) until all the plaintext characters have been filled into the matrix.

(vi) Read the matrix entries row-wise and concatenate each entry with the cipher text.

(vii) The cipher text (with depth 5) is obtained.

(viii) Initialize another matrix arr2 with no. of rows equal to 3 and no. of columns equal to the length of the intermediate cipher text

ix) Fill the matrix 'arr2' in the downward diagonal direction using the intermediate cipher text.

x) On reaching the last row, move diagonally upward.

xi) Repeat steps (viii) and (ix) till the entire matrix is filled.

xii) Read the entries of the matrix 'arr2' row-wise and concatenate each entry with the cipher text.

xiii) Display the final cipher (obtained with depth 3).

Methods used:

i) initializeMatrix (char[][] mat, int row, int col)

- mat : 2D char array
  row : no. of rows in matrix
  col : no. of columns in matrix
- This method is used to initialize a given matrix with dummy entries (eg '-').

ii) String encrypt (char[][] mat, int r, int c, String plaintext)

- mat: 2D character array
- r: no. of rows in the matrix
- c: no. of columns in the matrix
- plaintext: The plaintext to be encrypted.

Return type: String
- returns the encrypted string (cipher text)

This method:

- Fills the matrix with the plaintext

```
dir_down ← true
row ← 0
for col = 0 to c
        mat[row][col] = plainText[col]
        if row is last row
                dir_down ← false
        else if row is first row
                dir_down ← true
        if dir_down
                row++
        else
                row--
```

- Read the cipher text

```
for i: 0 to r
    for j: 0 to c
        if mat[i][j] != '_'
            cipherText += mat[i][j]
```

Sample input and output:

RAIL-FENCE CIPHER

Enter the plaintext: HELLOWORLD

Encrypting using depth 5....

The matrix of depth 5 is

```
H - - - - - - - L - -
- E - - - - - R - D
- - L - - - O - - -
- - - L - W - - - -
- - - - O - - - - -
```

The intermediate cipher text after encrypting with depth 5 is: HLERDLOLWO

Encrypting intermediate cipher text using depth 3....

The matrix of depth 3 is:

```
H       D       W
  L   R   L   L   O
    E       O
```

The final cipher text is HDWLRLLOEO

Result:

Thus the given message has been encrypted using rail fence cipher with depth 5 and then again encrypted with depth 3 to get the final cipher text.

CODE:

```
import java.util.Scanner;

class Main {

  public static void main(String[] args) {

   String plainText;
   String cipherText1,cipherText2;

   Scanner sc = new Scanner(System.in);

   System.out.println("\nRAIL-FENCE CIPHER");
   System.out.println("-----------------");

   System.out.print("\nEnter the plaintext: ");
   plainText=sc.nextLine();

   int len=plainText.length();

   char arr1[][] = new char[5][len];
   char arr2[][] = new char[3][len];

   initializeMatrix(arr1,5,len);
   initializeMatrix(arr2,3,len);

   System.out.println("\nEncrypting using depth 5...");

   cipherText1=encrypt(arr1,5,len,plainText);

   System.out.println("\nThe intermediate cipher text after encrypting with depth 5 is:
"+cipherText1);

   System.out.println("\nEncrypting intermediate cipher text using depth 3...");
   cipherText2=encrypt(arr2, 3, len, cipherText1);
   System.out.println("\nThe final cipher text is: "+cipherText2);
  }

  static void initializeMatrix(char[][] mat,int row,int col){

    for(int i=0;i<row;i++){

     for(int j=0;j<col;j++){

       mat[i][j]='-';

     }
    }
  }
```

```
static String encrypt(char[][]mat,int r,int c,String plainText){

   boolean dir_down=true;
   int row=0;

 for(int col=0;col<c;col++){

     mat[row][col]=plainText.charAt(col);

     if(row==r-1){
       dir_down=false;
     }
     else if(row==0){
       dir_down=true;
     }

     if(dir_down){
       row++;
     }
     else{
       row--;
     }
   }

   String cipherText="";
   System.out.println("The matrix of depth "+r+" is \n");

   for(int i=0;i<r;i++){

     for(int j=0;j<c;j++){

       System.out.print(mat[i][j]+" ");

       if(mat[i][j]!='-'){
         cipherText+=mat[i][j];
       }
     }
     System.out.println();
   }
   return cipherText;
  }

}
```

OUTPUT:

Example1:

PLAINTEXT: DECODE ZIGZAG

CIPHERTEXT: DGOGIZCZEDE A

RAIL-FENCE CIPHER

Enter the plaintext: DECODE ZIGZAG

Encrypting using depth 5...

The matrix of depth 5 is

D - - - - - - - I - - - -

- E - - - - - Z - G - - -

- - C - - - - - - Z - -

- - - O - E - - - - - A -

- - - - D - - - - - - - G

The intermediate cipher text after encrypting with depth 5 is: DIEZGC ZOEADG

Encrypting intermediate cipher text using depth 3...

The matrix of depth 3 is

D - - - G - - - O - - - G

- I - Z - C - Z - E - D -

- - E - - - - - - A - -

The final cipher text is: DGOGIZCZEDE A

Example 2:

PLAINTEXT: HELLOWORLD

CIPHERTEXT: HDWLRLLOEO

RAIL-FENCE CIPHER

Enter the plaintext: HELLOWORLD

Encrypting using depth 5...

The matrix of depth 5 is

H - - - - - - - L -

- E - - - - - R - D

- - L - - - O - - -

- - - L - W - - - -

- - - - O - - - - -

The intermediate cipher text after encrypting with depth 5 is: HLERDLOLWO

Encrypting intermediate cipher text using depth 3...

The matrix of depth 3 is

H - - - D - - - W -

- L - R - L - L - O

- - E - - - O - - -

The final cipher text is: HDWLRLLOEO