

SSN College of Engineering
Anna University Practical Examination
IT 8761 – Security Laboratory

Name: Sneha V

Register Number: 312217104160

Date: 21/12/2020

Aim, procedure, sample input-output and result -

IT 8761 - Security Laboratory

Sneha V
312217104160
21/12/2020

Question / : Develop a java program to implement the Row & column Transformation cipher twice to generate the cipher text

Aim

Procedure:

(*) Encryption :

- (1) Read the plain text and key value (of size n)
- (2) In the form of matrix of (m x n), write the plaintext row by row.
- (3) Read the With key values as column header, ~~two~~ generate the cipher text with respect to the order of column header.
- (4) Display the cipher text generated.

(*) Decryption :

- (1) Read cipher text as input
- (2) generate the matrix (n x m) with ciphertext written row by row.
- (3) The decryption key is generated from encryption key as following example:

Encryption: 4 3 1 2

Decryption: 3 4 2 1

(4) With key value as the column header, generate the plaintext with respect to the value of the column header.

(5) Display plaintext generated.

(6) Repeat encryption function twice then decryption function twice.

Function used:

(1) encrypt(text, key)

- Generates matrix
- Performs encryption
- Returns ciphertext

(2) decrypt(text, key)

- Generates matrix
- Performs decryption
- Returns plaintext

Sample I/O:

Plain text: my cipher text

Key: 3412

1st encryption

Matrix:

m	y	c	i
p	h	e	r
t	e	x	t

ciphertext: cexirtmptyhe

2nd encryption:

Matrix:

c	e	x	i
r	t	m	p
t	y	h	e

ciphertext: xmhipecrtety

1st decryption

Matrix:

x m h

i p e

c r t

e t y

ciphertext: cexlrmptyhe

2nd decryption

Matrix:

c e x

i r t

m p t

y h e

ciphertext: myciphertext

Result:

Row-column transposition cipher is executed
successfully.

Code -

```
import java.util.Scanner;
```

```
class Main {
```

```

//encryption
private static String encrypt(String plainText, String key){
    int keylen = key.length();
    char[][] plainTextMatrix = new char[10][keylen];
    System.out.println("Key:");
    for(int i=0; i<keylen;i++){
        System.out.print(key.charAt(i)+" \t");
    }
    System.out.println("\nPlain Text Matrix:");
    int c=0,cnt=0,row=0;
    for(int i=0;i<10;i++){
        for(int j=0;j<keylen;j++){

            if(cnt==plainText.length()){
                plainTextMatrix[i][j] = (char) ((char)c++ +
                    'a');
            }
            else{
                plainTextMatrix[i][j]= plainText.charAt(cnt++);
            }
            System.out.print(plainTextMatrix[i][j)+"\t");

        }
        System.out.println();
        if(cnt==plainText.length()){
            row=i+1;
            break;
        }
    }
    System.out.println("Rows: "+row);
    String dkey="";
    for(int i=0;i<keylen;i++){
        for(int j=0;j<keylen;j++){
            if((i+1)==Character.getNumericValue((key.charAt(j)))){
                dkey+=(j+1);
            }
        }
    }
    String cipherText="";
    for(int j=0;j<keylen;j++){
        for(int i=0; i<row;i++){
            cipherText+=
plainTextMatrix[i][Character.getNumericValue(dkey.charAt(j))-1];
        }
    }
}

```

```

        return cipherText;
    }
    //decryption
    private static String decrypt(String cipherText, String key)
    {
        int keylen = key.length();
        int c=0,cnt=0,row=cipherText.length()/keylen;
        char[][] cipherTextMatrix= new char[keylen][row];
        System.out.println("Cipher Text Matrix:");
        for(int i=0; i<keylen;i++){
            for(int j=0;j<row;j++){
                cipherTextMatrix[i][j]=cipherText.charAt(cnt++);
                System.out.print(cipherTextMatrix[i][j]+"\\t");
            }
            System.out.println();
        }
        String plainText="";
        for(int i=0;i<row;i++){
            for(int j=0;j<keylen;j++){
                plainText+=cipherTextMatrix[Character.getNumericValue(key.charAt(j))-
1][i];
            }
        }
        return plainText;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String plainText,key;
        int flag=0;
        System.out.println("-----");
        System.out.println("\\t\\tROW-COLUMN CIPHER\\n");
        System.out.println("-----");
        System.out.println("\\t\\tEncryption\\n");
        System.out.println("-----");

        do{
            int keylen,c=0;
            flag=0;
            System.out.println("Enter plain text:");
            String text = sc.nextLine();
            System.out.println("Enter key:");
            key= sc.nextLine();
            keylen=key.length();
            plainText="";

```

```

        for(int i=0;i<text.length();i++){
            if(text.charAt(i)!=' '){
                plainText+= text.charAt(i);
            }
        }
        for(int i=0;i<keylen;i++){
            for(int j=0;j<keylen;j++){
                if((j+1)==Character.getNumericValue(key.charAt(i)))
                    c++;
            }
        }
        if(c!=keylen || !( (plainText!=null) && (!plainText.equals("")) &&
(plainText.matches("[a-zA-Z]*$")))){
            flag=1;
            System.out.println("Incorrect format of plain text or key. Please try
again.\n");
        }
    }while(flag==1);

    System.out.println("\n\t\t1st Encryption:");
    String cipherText = encrypt(plainText,key);
    System.out.println("Text after 1st encryption: "+ cipherText);

    System.out.println("\n\t\t2nd Encryption:");
    cipherText = encrypt(cipherText,key);
    System.out.println("Text after 2nd encryption: "+ cipherText);
    System.out.println("\nCipher Text: "+ cipherText);
    System.out.println("-----");
    System.out.println("\t\tDecryption");
    System.out.println("-----");

    System.out.println("\n\t\t1st Decryption:");
    cipherText = decrypt(cipherText,key);
    System.out.println("Text after 1st decryption: "+ cipherText);

    System.out.println("\n\t\t2nd Decryption:");
    plainText = decrypt(cipherText,key);
    System.out.println("Text after 2nd decryption: "+ plainText);
    sc.close();
}
}

```

Output -

ROW-COLUMN CIPHER

Encryption

Enter plain text:

columnar transposition

Enter key:

451326

1st Encryption:

Key:

4 5 1 3 2 6

Plain Text Matrix:

c o l u m n

a r t r a n

s p o s i t

i o n a b c

Rows: 4

Text after 1st encryption: ltonmaibursacasiorporntc

2nd Encryption:

Key:

4 5 1 3 2 6

Plain Text Matrix:

l t o n m a

i b u r s a

c a s i o r

p o n n t c

Rows: 4

Text after 2nd encryption: ousnmsotnrinlicptbaoaarc

Cipher Text: ousnmsotnrinlicptbaoaarc

Decryption

1st Decryption:

Cipher Text Matrix:

o u s n

m s o t

n r i n

l i c p
t b a o
a a r c

Text after 1st decryption: ltonmaibursacasi orpnntc

2nd Decryption:

Cipher Text Matrix:

l t o n
m a i b
u r s a
c a s i
o r p o
n n t c

Text after 2nd decryption: columnartranspositionabc