

Practical No.5

Aim: To implement Columnar Cipher transposition techniques.

Date :

Practical No. 5



Aim : To implement columnar cipher transposition techniques.

Theory :

Given a plain-text message and a numeric key, cipher / de-cipher the given text columnar transposition cipher. The columnar transposition cipher is the form of transposition cipher just like Rail Fence cipher. Columnar transposition involves writing the plaintext out in rows, and then reading the cipher off in columns one by one.

Steps to be included :

In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.

- ① The message is written out in rows of a fixed length, and then read out again column by column and the columns are chosen in some scrambled order.
- ② Width of the rows and the permutation of the columns are usually defined by a keyword.
- ③ For example, the word HALL is a length 4, and the permutation is defined by the alphabetical order of the letters in the keyword. In this case, the order would be "3 1 2 4".
- ④ Any space spaces are filled with nulls or left blank or placed by a character (examples: ...)
- ⑤ Finally, the message is read off in columns, in the order specified by the keyword.

Example :

Columnar Transposition method

Given : Key - 4312567

plaintext : attack postponed until two am xyz Page No. _____

Date :



4	3	1	2	5	6	7
q	t	t	q	c	k	p
o	s	t	p	o	n	e
d	u	n	t	i	d	t
w	o	a	m	x	y	z

columns : 1 2 3 4 5 6 7

Cipher text : ttng aptm tsuo aodw coix knly pzt

Program :

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

```
public class Columnar {
```

```
    private static Scanner in;
```

```
    public static void main (String[] args) {
```

```
        System.out.println("Columnar Transposition Cipher");
```

```
        in = new Scanner (System.in);
```

```
        System.out.print("1. Encryption \n 2. Decryption \n choose (1,2): ");
```

```
        int choice = in.nextInt();
```

```
        in.nextLine();
```

```
        if (choice == 1) {
```

```
            System.out.println("Encryption");
```

```
            encryption();
```

```
        } else if (choice == 2) {
```

```
            System.out.println("Decryption");
```

```
            decryption();
```

```
        } else {
```

Page No. _____



```
System.out.println("Invalid choice");  
System.exit(0);  
}
```

```
}
```

```
private static void encryption () {  
    System.out.print("Enter message : ");  
    String plaintext = in.nextLine().toUpperCase().replace(" ", "");  
    StringBuilder msg = new StringBuilder(plaintext);  
    System.out.print("Enter keyword : ");  
    String keyword = in.nextLine().toUpperCase();
```

```
int[] keywordNumList = keywordNumAssign(keyword);
```

```
for (int i=0; j=1; i<keyword.length(); i++, j++) {  
    System.out.print(keyword.substring(i, j) + " ");  
}
```

```
System.out.println();
```

```
for (int i : keywordNumList) {  
    System.out.print(i + " ");  
}
```

```
System.out.println();  
System.out.println("-----");
```

```
int extraLetters = msg.length() / keyword.length();  
int dummyCharacters = keyword.length() - extraLetters;
```

```
if (extraLetters != 0) {  
    for (int i=0; i<dummyCharacters; i++) {  
        msg.append(" ");  
    }
```

```
}
```


Name of Practical

```
int numofRows = msg.length() / keyword.length();
char[][] arr = new char[numofRows][keyword.length()];
```

```
int z=0;
for (int i=0; i<numofRows; i++) {
    for (int j=0; j<keyword.length(); j++) {
        arr[i][j] = msg.charAt(z);
        z++;
    }
}
```

```
for (int i=0; i<numofRows; i++) {
    for (int j=0; j<keyword.length(); j++) {
        System.out.print(arr[i][j] + " ");
    }
    System.out.println();
}
```

```
StringBuilder cipherText = new StringBuilder();
```

```
System.out.println();
String numLoc = getNumberLocation(keyword, keywordNumList);
System.out.println("Location of numbers: " + numLoc);
System.out.println();
```

```
for (int j=0; j<numofRows; j++) {
    cipherText.append(arr[j][0]);
}
```

Teacher's Signature

Name of Practical

```
System.out.println("Cipher Text:" + cipherText);
}
```

```
private static void decryption() {
    System.out.print("Enter message: ");
    String msg = in.nextLine().toUpperCase().replace(" ", "");
    System.out.print("Enter keyword: ");
    String keyword = in.nextLine().toUpperCase();
    int numRows = msg.length() / keyword.length();
    char[][] arr = new char[numRows][keyword.length()];
    int[] keywordNumList = keywordNumAssign(keyword);
    String number = getNumLocation(keyword, keywordNumList);

    for (int i = 0; i < msg.length(); i += k++) {
        int d = 0;
        if (k == keyword.length()) {
            k = 0;
        } else {
            d = character.getNumericValue(number, charAt(k));
        }
        for (int j = 0; j < numRows; j++) {
            arr[j][d] = msg.charAt(i);
        }
    }
}
```

```
StringBuilder plaintext = new StringBuilder();
for (int i = 0; i < numRows; i++) {
    for (int j = 0; j < keyword.length(); j++) {
```

Teacher's Signature

ne of Practical

```
        plaintext.append(char[i][j]);  
    }  
    System.out.println("plain text : " + plaintext);  
}
```

```
private static String getNumberLocation (String keyword, int []  
keywordNumList) {  
    String numLoc = " ";  
    for (int i=1; i<keyword.length()+1; i++) {  
        for (int j=0; j<keyword.length(); j++) {  
            if (keywordNumList[j] == i) {  
                numLoc += j;  
            }  
        }  
    }  
    return numLoc;  
}
```

```
private static int[] keywordNumAssign (String keyword) {  
    String alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
    int[] keywordNumList = new int [keyword.length()];  
  
    int init = 0;  
    for (int i=0; i<alpha.length(); i++) {  
        for (int j=0; j<keyword.length(); j++) {  
            if (alpha.charAt(i) == keyword.charAt(j)) {  
                init++;  
            }  
        }  
    }  
}
```

Teacher's Signature

Columnar Transposition Cipher

1. Encryption

2. Decryption

Choose(1,2): 1

Encryption

Enter Message: Tomato and Potato

Enter Keyword: Buy

B U Y

1 2 3

T O M

A T O

A N D

P O T

A T O

Location of numbers: 012

Cipher Text: TAAPAOITNOTMODTO

Columnar Transposition Cipher

1. Encryption

2. Decryption

Choose(1,2): 2

Decryption

Enter Message: TAAPAOITNOTMODTO

Enter Keyword: Buy

Plain Text: TOMATOANDPOTATO

Conclusion:

Columnar cipher implemented successfully.

Name of Practical

```
        keywordNumList[i] = init;
    }
}

return keywordNumList;
}
```

Conclusion:

Columnar cipher implemented successfully.

Teacher's Signature

Date :



Viva Question :

① Columnar cipher falls under the category of ?
→ Transposition cipher.

② Which cipher is formed by applying columnar transposition cipher twice ?
→ Double transposition cipher.