Experiment 02

Design and Implement a product cipher using Substitution ciphers and Transposition Cipher..

Roll No.	19
Name	Manav Jawrani
Class	D15-A
Subject	Internet Security Lab
LO Mapped	LO1: To apply the knowledge of symmetric cryptography to implement classical ciphers.

Aim: Write a program to understand Implementation of a product cipher using Substitution ciphers and Transposition Cipher.

Introduction:

1. What is Cipher?

Ciphers, also called encryption algorithms, are systems for encrypting and decrypting data. A cipher converts the original message, called plaintext, into ciphertext using a key to determine how it is done.

2. What is a substitution cipher?

In cryptography, a **substitution cipher** is a method of encrypting in which units of plaintext are replaced with the ciphertext, in a defined manner, with the help of a key; the "units" may be single letters (the most common), pairs of letters, triplets of letters, mixtures of the above, and so forth. The receiver deciphers the text by performing the inverse substitution process to extract the original message.

3. What is a transposition cipher?

Transposition Cipher is a cryptographic algorithm where the order of alphabets in the plaintext is rearranged to form a cipher text. In this process, the actual plain text alphabets are not included.

Example:

A simple example for a transposition cipher is columnar transposition cipher where each character in the plain text is written horizontally with specified alphabet width. The cipher is written vertically, which creates an entirely different cipher text.

Consider the plain text hello world, and let us apply the simple columnar transposition technique as shown below

h	e	l	l
o	w	o	r
l	d		

The plain text characters are placed horizontally and the cipher text is created with vertical format as : holewdlo lr. Now, the receiver has to use the same table to decrypt the cipher text to plain text.

4. What is a product cipher?

A **product cipher**, data encryption scheme in which the ciphertext produced by encrypting a plaintext document is subjected to further encryption. By combining two or more simple transposition ciphers or substitution ciphers, a more secure encryption may result.

Algorithm:

STEP 1: Ask the user to enter a plain text.
STEP 2: Generate a random number for a key to denote the required shift.
STEP 3: Call the function to encrypt the given plain text.
STEP 4: Traverse the given plain text one character at a time.
STEP 5: For each character, transform the given character as per the above required shift.
STEP 6: Return the encrypted text generated.
STEP 7: Decrypt the encrypted text using the key.
STEP 8: Generate a key in a cyclic manner until it's length isn't equal to the length of the original text.
STEP 9: Returns the encrypted text generated with the help of the key.
STEP 10: Decrypts the encrypted text and returns the original text.

Code:

```
import java.util.*;
public class ProductCipher {
public static void main(String args[]) {
System.out.println("Enter the input to be encrypted:");
String substitutionInput = new Scanner(System.in).nextLine();
System.out.println("Enter a number:");
int n = new Scanner(System.in).nextInt();
// Substitution encryption
StringBuffer substitutionOutput = new StringBuffer();
for(int i=0 ; i<substitutionInput.length() ; i++) {
char c = substitutionInput.charAt(i);
substitutionOutput.append((char) (c+5));
```

```
}
System.out.println("\nSubstituted text:");
System.out.println(substitutionOutput);
// Transposition encryption
String transpositionInput = substitutionOutput.toString();
int modulus;
if((modulus = transpositionInput.length()%n) != 0) {
modulus = n-modulus;
// 'modulus' is now the number of blanks/padding (X) to be
appended
for( ; modulus!=0 ; modulus--) {
transpositionInput += "/";
}
}
StringBuffer transpositionOutput = new StringBuffer();
System.out.println("\nTransposition Matrix:");
for(int i=0 ; i<n ; i++) {
for(int j=0 ; j<transpositionInput.length()/n ; j++) {
char c = transpositionInput.charAt(i+(j*n));
System.out.print(c);
transpositionOutput.append(c);
}
System.out.println();
}
System.out.println("\nFinal encrypted text:");
System.out.println(transpositionOutput);
// Transposition decryption
n = transpositionOutput.length()/n;
StringBuffer transpositionPlaintext = new StringBuffer();
for(int i=0 ; i<n ; i++) {
for(int j=0 ; j<transpositionOutput.length()/n ; j++) {
```

```
char c = transpositionOutput.charAt(i+(j*n));
transpositionPlaintext.append(c);
}
}
// Substitution decryption
StringBuffer plaintext = new StringBuffer();
for(int i=0 ; i<transpositionPlaintext.length() ; i++) {
char c = transpositionPlaintext.charAt(i);
plaintext.append((char) (c-5));
}
System.out.println("\nPlaintext:");
System.out.println(plaintext);
}
}
```

Output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\MANAV> cd "e:\CNS Programs\" ; if ($?) { javac ProductCipher.java } ; if ($?) { java ProductCipher }
Enter the input to be encrypted:
Meet me tonight
Enter a number:
4

Substituted text:
Rjjy%rj%ytsnlmy

Transposition Matrix:
R%yl
jrtm
jjsy
y%n/

Final encrypted text:
R%yljrtmjjsyy%n/

Plaintext:
Meet me tonight*
PS E:\CNS Programs>
```

Conclusion:

We have understood the concept of product cipher and implemented it using a java program.