Experiment 02

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

Roll No.	70
Name	Ishika Sanjay Devare
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:

Product Cipher:

In cryptography, a product cipher combines two or more transformations in a manner intending that the resulting cipher is more secure than the individual components to make it resistant to cryptanalysis. The product cipher combines a sequence of simple transformations such as substitution (S-box), permutation (P-box), and modular arithmetic. The concept of product ciphers is due to Claude Shannon, who presented the idea in his foundational paper, Communication Theory of Secrecy Systems.

For transformation involving a reasonable number of n message symbols, both of the foregoing cipher systems (the S-box and P-box) are by themselves wanting. Shannon suggested using a combination of S-box and P-box transformation—a product cipher. The combination could yield a cipher system more powerful than either one alone. This approach of alternatively applying substitution and permutation transformation has been used by IBM in the Lucifer cipher system and has become the standard for national data encryption standards such as the Data Encryption Standard and the Advanced Encryption Standard. A product cipher that uses only substitutions and permutations is called an SP network. Feistel ciphers are an important class of product ciphers.

Substitution Cipher:

Hiding some data is known as encryption. When plain text is encrypted it becomes unreadable and is known as ciphertext. In a Substitution cipher, any character of plain text from the given fixed set of characters is substituted by another character from the same set depending on a key. For example with a shift of 1, A would be replaced by B, B would become C, and so on. Note: Special case of Substitution cipher is known as Caesar cipher where the key is taken as 3.

Mathematical representation The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,…, Z = 25. Encryption of a letter by a shift n can be described mathematically as.
E_n(x)=(x+n)mod\ 26
(Encryption Phase with shift n)

D_n(x)=(x-n)mod\ 26
(Decryption Phase with shift n)

Examples: 1. Plain Text: I am studying Data Encryption
Key: 4
Output: M eq wxyhCmrk Hexe IrgvCtxmsr

2. Plain Text: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Key: 4
Output: EFGHIJKLMNOPQRSTUVWXYZabcd

Input:
A String of both lower and upper case letters, called PlainText.
An Integer denoting the required key.

Procedure:
1. Create a list of all the characters.
2. Create a dictionary to store the substitution for all characters.
3. For each character, transform the given character as per the rule, depending on whether we're encrypting or decrypting the text.
4. Print the new string generated.

Transposition Cipher:
In cryptography, a transposition cipher is a method of encryption which scrambles the positions of characters (*transposition*) without changing the characters themselves. Transposition ciphers reorder units of plaintext (typically characters or groups of characters) according to a regular system to produce a ciphertext which is a permutation of the plaintext. They differ from substitution ciphers, which do not change the position of units of plaintext but instead change the units themselves.
Plaintexts can be rearranged into ciphertext using a key, scrambling the order of characters like the shuffled pieces of a jigsaw puzzle. The resulting message is hard to decipher without the key because there are many ways the characters can be arranged.

For example, the plaintext "THIS IS WIKIPEDIA" could be encrypted to "TWDIP SIHII IKASE". To decipher the encrypted message without the key, an attacker could try to guess possible words and phrases like DIATHESIS, DISSIPATE, WIDTH, etc., but it would take them some time to reconstruct the plaintext because there are many combinations of letters and words. By contrast, someone with the key could reconstruct the message easily:

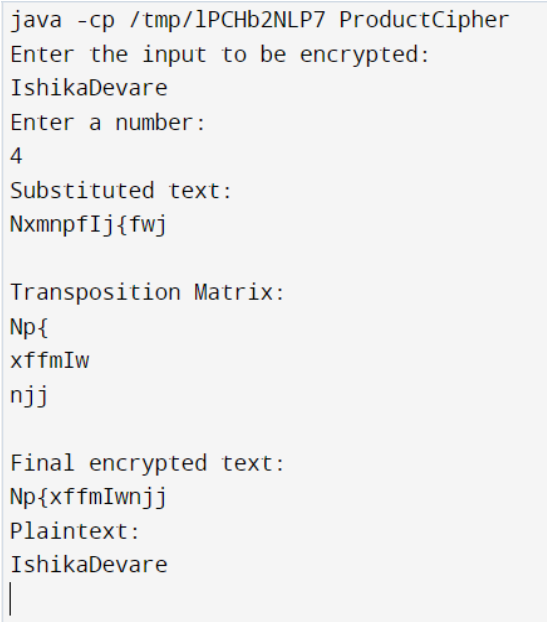
Code:

```
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 = 0;
if ((modulus = transpositionInput.length() % n)
!= 0) {
modulus = n - modulus;
// 'modulus' is now the number of
blanks/padding (X) to be

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);
}
```

Results:



```
java -cp /tmp/lPCHb2NLP7 ProductCipher
Enter the input to be encrypted:
IshikaDevare
Enter a number:
4
Substituted text:
NxmnpfIj{fwj

Transposition Matrix:
Np{
xffmIw
njj

Final encrypted text:
Np{xffmIwnjj
Plaintext:
IshikaDevare
```

Conclusion:

In this practical we learned about design and implementation of a product cipher using Substitution ciphers and Transposition Cipher.