

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING
Department of Computer Engineering

1. Course , Subject & Experiment Details

Academic Year	2024-25	Estimated Time	03 - Hours
Course & Semester	T.E. (CMPN)- Sem VI	Subject Name & Code	CSS - (CSC602))
Chapter No.	02 – Mapped to CO- 1	Chapter Title	Basics of Cryptography

Practical No:	1
Title:	Design and Implementation of a product cipher using Substitution and Transposition ciphers
Date of Performance:	30/01/2025
Date of Submission:	20/02/2025
Roll No:	9913
Name of the Student:	Mark Lopes

Evaluation:

Sr. No	Rubric	Grade
1	On time submission Or completion (2)	
2	Preparedness(2)	
3	Skill (4)	
4	Output (2)	

Signature of the Teacher:

Date:

Lab Manual: Prepared by Prof. Monali Shetty

Title: Design and Implementation of a product cipher using Substitution and Transposition Ciphers.

Lab Objective :

This lab provides insight into:

- How different types of Substitution Ciphers and Transposition Ciphers like Hill cipher, Verman cipher, Playfair cipher, Vigenere cipher works and their advantages and disadvantages.

Reference : “Cryptography and Network Security” B. A. Forouzan
“Cryptography and Network Security” Atul Kahate

Pre-requisite : Any Programming language and Knowledge of Ciphering .

Theory:

Cryptography is the practice and study of hiding information. It is the process of converting ordinary information (plain text) into cipher text and converting cipher text again to plain text, A cipher is a pair of algorithms which create the encryption and decryption.

Substitution Cipher: In cryptography, a **substitution cipher** is a method of encryption by which units of plaintext are replaced with cipher text according to a regular system; 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 an inverse substitution.

Types of substitution cipher:

- **Monoalphabetic Cipher:** A *monoalphabetic substitution cipher*, also known as a simple substitution cipher, relies on a fixed replacement structure. That is, the substitution is fixed for each letter of the alphabet. Thus, if "a" is encrypted to "R", then every time we see the letter "a" in the plaintext, we replace it with the letter "R" in the ciphertext.

Ex. If **a** is substituted by ‘**x**’ and **b** is substituted by ‘**y**’ and so on than

“starbucks at three” encrypted as
PQXOYRHPXQQEOBB

- 1) Caesar Cipher/Additive/ Shift: It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 3, A would be replaced by D, B would become E, and so on. The method is named after Julius Caesar, who used it to communicate with his generals.

Lab Manual: Prepared by Prof. Monali Shetty

Example

Plain: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Cipher: DEFGHIJKLMNOPQRSTUVWXYZABC

Like Plaintext: the quick brown fox jumps over the lazy dog

Cipher text: WKH TXLFN EURZQ IRA MXPSV RYHU WKH ODCB GRJ

- **Polyalphabetic Cipher** : In this cipher, we are using no. of substitutions at different positions in the message.

- 1) Vigenere Cipher
- 2) Hill Cipher

- 1) **Hill Cipher** : It is a block cipher.

Key: An invertible $m \times m$ matrix (where m is the block length) i.e the sender & receiver must first agree upon a key matrix A of size $m \times m$. A must be invertible mod 26.

Encryption: To encrypt a message using a message using the Hill Cipher we must first turn our keyword into a key matrix (a 2×2 matrix for working with digraphs). We also turn the PT into digraphs and each of these into a column vector. We then perform matrix multiplication modulo the length of the alphabet (i.e 26) on each vector. These vectors are then converted back into letters to produce the ciphertext.

Decryption:

To decrypt a ciphertext encoded using the Hill Cipher, we must find the inverse matrix. Once we have the inverse matrix, the process is the same as encrypting. That is we multiply the inverse key matrix by the column vectors that the ciphertext is split into, take the results modulo the length of the alphabet, and finally convert the numbers back to letters.

- 2) **Vernam Cipher:** it is a stream, polyalphabetic cipher in which the plaintext is XORed with a random or pseudorandom stream of data to generate the ciphertext. If the stream of data is truly random and used only once, this is the one-time pad.

Ex. H E L L O message

7 (H) 4 (E) 11 (L) 11 (L) 14 (O) message

+ 23 (X) 12 (M) 2 (C) 10 (K) 11 (L) key

= 30 16 13 21 25 message + key

= 4 (E) 16 (Q) 13 (N) 21 (V) 25 (Z) message + key (mod 26)

E Q N V Z \rightarrow ciphertext

A **transposition cipher** is a method of encryption by which the positions held by units of plaintext (which are commonly characters or groups of characters) are shifted according to a regular system, so that the ciphertext constitutes a permutation of the plaintext. That is, the order of the units is changed.

Lab Manual: Prepared by Prof. Monali Shetty

1. **Rail Fence cipher:** The Rail Fence cipher is a form of transposition cipher that gets its name from the way in which it is encoded. In the rail fence cipher, the plaintext is written downwards on successive "rails" of an imaginary fence, then moving up when we get to the bottom. The message is then read off in rows. For example, using three "rails" and a message of 'WE ARE DISCOVERED. FLEE AT ONCE', the cipher writes out:

Example:

```
W . . . E . . . C . . . R . . . L . . . T . . . E
. E . R . D . S . O . E . E . F . E . A . O . C .
. . A . . . I . . . V . . . D . . . E . . . N . .
```

Then reads off:

WECRL TEERD SOEEF EAOCA IVDEN

2. **Single Columnar transposition:** In a columnar transposition, 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. Both the width of the rows and the permutation of the columns are usually defined by a keyword. For example, the word ZEBRAS is of length 6 (so the rows are of length 6), and the permutation is defined by the alphabetical order of the letters in the keyword. In this case, the order would be "6 3 2 4 1 5".

In a regular columnar transposition cipher, any spare spaces are filled with nulls; in an irregular columnar transposition cipher, the spaces are left blank. Finally, the message is read off in columns, in the order specified by the keyword. For example, suppose we use the keyword ZEBRAS and the message WE ARE DISCOVERED. FLEE AT ONCE. In a regular columnar transposition, we write this into the grid as:

Example:

ZEBRAS - 632415

```
6 3 2 4 1 5
W E A R E D
I S C O V E
R E D F L E
E A T O N C
E Q K J E U
```

The ciphertext is then read off as:

EVLNE ACDTK ESEAQ ROFOJ DEECU WIREE

3. Double Columnar transposition: A single columnar transposition could be attacked by guessing possible column lengths, writing the message out in its columns (but in the wrong order, as the key is not yet known), and then looking for possible anagrams. Thus to make it

Lab Manual: Prepared by Prof. Monali Shetty

stronger, a double transposition was often used. This is simply a columnar transposition applied twice. The same key can be used for both transpositions, or two different keys can be used.

As an example, we can take the result of the irregular columnar transposition in the previous section, and perform a second encryption with a different keyword, *STRIFE*, which gives the permutation "564231"

Example:

5 6 4 2 3 1
E V L N A C
D T E S E A
R O F O D E
E C W I R E
E

This is read off column wise to give the cipher text.

CAEEN SOIAE DRLEF WEDRE EVTOC

Algorithm of Proposed Product Cipher :

```
// Single Columnar Transposition Cipher with Affine Encryption
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

// Affine Encryption Function
void encrypt_affine(char word[], char cipher[], int A, int B) {
    int i;
    for(i = 0; word[i] != '\0'; i++) {
        if(word[i] != ' ') {
            cipher[i] = ((A * (word[i] - 'A') + B) % 26) + 'A';
        } else {
            cipher[i] = ' ';
        }
    }
    cipher[i] = '\0';
}

// Columnar Transposition Encryption Function
char* Encryption(int no_rows, int len_key, int len_msg, char msg[], int
col_val[]) {
    int x = 0;
    char enc_mat[no_rows + 1][len_key];
    char* cipher = (char*)malloc((len_msg + (no_rows + 1) * len_key -
len_msg + 1) * sizeof(char));
    int cipher_index = 0;

    // Creating the matrix
    for (int i = 0; i < no_rows + 1; i++) {
        for (int j = 0; j < len_key; j++) {
            if (x >= len_msg) {
                enc_mat[i][j] = '_';
            } else {
                enc_mat[i][j] = msg[x];
            }
            x++;
        }
    }
}
```

```

    int t = 1;
    // Finding the cipher text according to the value of col_val array
    while (t <= len_key) {
        for (int i = 0; i < len_key; i++) {
            int k = col_val[i];
            if (k == t) {
                for (int j = 0; j < no_rows + 1; j++) {
                    cipher[cipher_index++] = enc_mat[j][i];
                }
                t++;
            }
        }
        cipher[cipher_index] = '\0'; // Null-terminate the string
        return cipher;
    }

int main() {
    // Original Message
    // char msg[] = "Mark Lopes";
    char msg[100];
    printf("Enter your string: ");
    fgets(msg, sizeof(msg), stdin);

    char affine_cipher[100];

    int A,B;
    printf("Enter the value of A and B: ");
    scanf("%d %d", &A, &B);
    getchar(); // Clear the newline character from the buffer

    // Apply Affine Encryption first
    encrypt_affine(msg, affine_cipher, A, B);

    // Key for Columnar Transposition
    //char key[] = "HACK";

```

```

char key[100];
printf("Enter the key for columnar transposition: ");
fgets(key, sizeof(key), stdin);

int len_key = strlen(key);
int len_msg = strlen(affine_cipher);

int val = 1, count = 0, ind;
int col_val[len_key];

// Initializing col_val array with 0
memset(col_val, 0, sizeof(col_val));

// Numbering the key alphabets according to their ASCII values
while (count < len_key) {
    int min = 999;
    for (int i = 0; i < len_key; i++) {
        if ((min > (int)key[i]) && (col_val[i] == 0)) {
            min = (int)key[i];
            ind = i;
        }
    }
    col_val[ind] = val;
    count++;
    val++;
}

int no_rows = len_msg / len_key;
if (len_msg % len_key != 0) no_rows++; // Adjusting for leftover
characters

// Apply Columnar Transposition Encryption on Affine Cipher Text
char* cipher_text = Encryption(no_rows, len_key, len_msg,
affine_cipher, col_val);

// Display Results
printf("Affine Ciphered Text: %s\n", affine_cipher);
printf("Final Encrypted Message: %s\n", cipher_text);

```



```
    free(cipher_text); // Free allocated memory

    return 1;
}
```

Lab Manual: Prepared by Prof. Monali Shetty

Example of Product Cipher

```
Enter your string: Mark Lopes
Enter the value of A and B: 17 19
Enter the key for columnar transposition: person
Affine Ciphered Text: PRUF YVMHL;
Final Encrypted Message: V__RH_Y__ __PM_UL_F;_
PS C:\Users\Mark Lopes\Desktop\college\Sem_6\css\lab_1>
```

Lab Manual: Prepared by Prof. Monali Shetty

Practical & Real-Time Application

- encryption and decryption of stream cipher and block cipher

Conclusion:

The program was tested for different sets of inputs.
Program is working SATISFACTORY NOT SATISFACTORY (Tick appropriate outcome)

Post Lab :

1. To break the Caesar cipher using brute force attack, how many attempts are needed?
2. Compare Substitution and Transposition techniques.

TE comps - A
9913
Mark Lopez CSS - ep1 postlab

Q. 1 To break the Caesar cipher using brute force attack, how many brute attempts are required?

→ In a cipher, each letter in the message, is shifted by a fixed number of places in the alphabet. Since there are only 26 possible shifts (0-25), a brute force attack would require at most 26 attempts to find the correct one.

Q. 2 Compare Substitution and transposition techniques?

Substitution cipher	Transposition cipher
1. Plain text characters are replaced with other char, numbers and symbols.	1. Plain text characters are rearranged with respect to the position.
2. They are monoalphabetic substitution cipher and poly alphabetic substitution cipher.	2. They are key - less transposition cipher and keyed transposition cipher.
3. Involves replacing plaintext letters or groups of letters with cipher text according to a specific algorithm or key.	3. Involves rearranging the order of plaintext letters according to a specific algorithm or key.
4. Ex:- Caesar cipher, monoalphabetic cipher, polyalphabetic cipher.	4. Ex:- Rail fence cipher, columnar transposition cipher and route cipher.

Lab Manual: Prepared by Prof. Monali Shetty