Prithvi Narayan Campus

Institute of Science and Technology

Department of Computer Science and Information Technology

Tribhuvan University



Lab Report

On

Cryptography (CSC – 316)

Submitted to:

Mr. Dev Timilsina

Submitted by:

Sandesh Paudel

Roll No: 27571/077

# Lab Index

Name: Sandesh Paudel Roll. No: 27571/077

Level: Bachelor Year/Sem: 3rd/5th

Subject: Cryptography Course No.: CSC-316

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Experiment | Date of  Submission | Remarks |
| 1. | To encrypt and decrypt the plain text using Caesar cipher technique with the help of C program. | 2080/10/04 |  |
| 2. | To encrypt the plain text using rail-fence cipher technique with the help of C program. | 2080/10/04 |  |
| 3. | To encrypt and decrypt the plain text using hill cipher technique with the help of C++ program. | 2080/10/04 |  |
| 4. | To encrypt and decrypt the plain text using monoalphabetic cipher technique with the help of C program. | 2080/10/04 |  |
| 5. | To encrypt and decrypt the plain text using vigenere cipher technique with the help of C program. | 2080/10/04 |  |
| 6. |  |  |  |
| 7. |  |  |  |

**LAB 1**

**TITLE**:TO ENCRYPT AND DECRYPT THE PLAIN TEXT USING CAESAR CIPHER TECHNIQUE WITH THE HELP OF C PROGAM.

**THEORY**:

The Caesar Cipher technique is one of the earliest and simplest methods of encryption technique. It’s simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter with a fixed number of positions down the alphabet. For example, with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials.

Thus, to cipher a given text we need an integer value, known as a shift which indicates the number of positions each letter of the text has been moved down.

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.

Algorithm for Caesar Cipher:

Input:

* A String of lower-case letters, called Text.
* An Integer between 0-25 denoting the required shift.

Procedure:

* Traverse the given text one character at a time.
* For each character, transform the given character as per the rule, depending on whether we’re encrypting or decrypting the text.
* Return the new string generated.

**LAB WORK:**

Source code:

#include <stdio.h>

#include <string.h>

#include<ctype.h>

int main(){

char plain\_text[20], cipher\_text[20];

int key, i, length;

printf("Enter the plain text:");

scanf("%s", plain\_text);

printf("Enter the key value between 0 <= key <= 25:\t");

scanf("%d", &key);

printf("The plain text is %s", plain\_text);

printf("\nThe encrypted text is:\t");

length = strlen(plain\_text);

for (i = 0; i < length; i++){

if (isupper(plain\_text[i])){

cipher\_text[i] = (plain\_text[i] - 'A' + key) % 26 + 'A';

}

else if (islower(plain\_text[i])){

cipher\_text[i] = (plain\_text[i] - 'a' + key) % 26 + 'a';

}

else{

cipher\_text[i] = plain\_text[i];

}

printf("%c", cipher\_text[i]);

}

// Decryption

printf("\n\nThe decrypted text is:");

for (i = 0; i < length; i++){

if (isupper(cipher\_text[i])){

plain\_text[i] = (cipher\_text[i] - 'A' - key + 26) % 26 + 'A';

}

else if (islower(cipher\_text[i])){

plain\_text[i] = (cipher\_text[i] - 'a' - key + 26) % 26 + 'a';

}

else{

plain\_text[i] = cipher\_text[i];

}

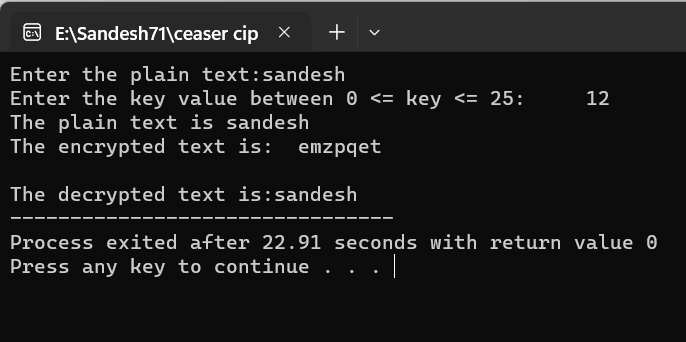
printf("%c", plain\_text[i]);

}

return 0;

}

**Output:**



**CONCLUSION**:

Hence, in this experiment we encrypted a plain text provided by the user, using Caesar cipher method. The plain text given was “sandesh” which was encrypted using a key value of 10 as “emzpqet” as cipher text. This cipher text was again decrypted to the original message.

# **LAB 2**

**TITLE**: TO ENCRYPT THE PLAIN TEXT USING RAIL-FENCE CIPHER TECHNIQUE WITH THE HELP OF C PROGRAM.

**THEORY**:

The rail fence cipher (sometimes called zigzag cipher) is a transposition cipher that jumbles up the order of the letters of a message using a basic algorithm. The key consists of the number of rows and the offset (starting place for the first plaintext character). It can encrypt any characters, including spaces, but security is increased if all characters are of equal caps. A rail fence cipher can very easily be broken, since there are only a very limited number of keys.

Encryption

* In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.  In the rail fence cipher, the plain-text is written downwards and diagonally on successive rails of an imaginary fence.
* When we reach the bottom rail, we traverse upwards moving diagonally, after reaching the top rail, the direction is changed again. Thus, the alphabets of the message are written in a zig-zag manner.
* After each alphabet has been written, the individual rows are combined to obtain the cipher-text.

**LAB WORK**:

Source code:

//Rail-Fence cipher

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

main()

{

int i, j, len, rails, count, code[100][1000];

char str[1000];

printf("Enter a Secret Message:");

gets(str);

len = strlen(str);

printf("Enter number of rails:");

scanf("%d", &rails);

for (i = 0; i < rails; i++){

for (j = 0; j < len; j++){

code[i][j] = 0;

}

}

count = 0;

j = 0;

while (j < len){

if (count % 2 == 0){

for (i = 0; i < rails; i++){

// strcpy(code[i][j],str[j]);

code[i][j] = (int)str[j];

j++;

}

}

else{

for (i = rails - 2; i > 0; i--){

code[i][j] = (int)str[j];

j++;

}

}

count++;

}

for (i = 0; i < rails; i++){

for (j = 0; j < len; j++){

if (code[i][j] != 0)

printf("%c", code[i][j]);

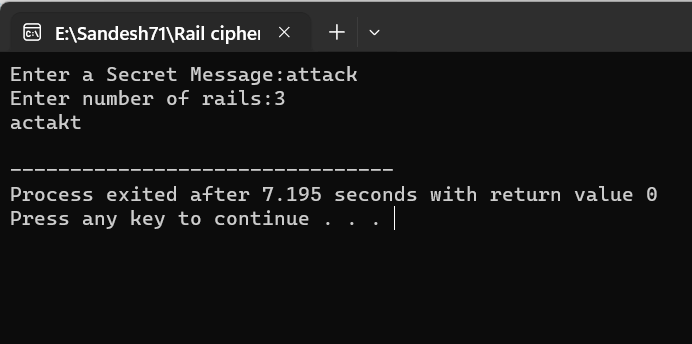
}

}

printf("\n");

}

**Output:**



**CONCLUSION**:

Hence, in this experiment we used Rail fence cipher technique to encrypt the plain text.

**LAB 3**

**TITLE**: TO ENCRYPT AND DECRYPT THE PLAIN TEXT USING HILL CIPHER TECHNIQUE WITH THE HELP OF C++ PROGRAM.

**THEORY**:

Hill cipher is a polygraphic substitution cipher based on linear algebra. Each letter is represented by a number modulo 26. Often the simple scheme A = 0, B = 1, …, Z = 25 is used, but this is not an essential feature of the cipher. To encrypt a message, each block of n letters (considered as an n-component vector) is multiplied by an invertible n × n matrix, against modulus 26. To decrypt the message, each block is multiplied by the inverse of the matrix used for encryption. The matrix used for encryption is the cipher key, and it should be chosen randomly from the set of invertible n × n matrices (modulo 26).

Encryption

To encrypt a message using the Hill Cipher we must first turn our keyword into a key matrix (a 2 x 2 matrix for working with digraphs, a 3 x 3 matrix for working with trigraphs, etc). We also turn the plaintext into digraphs (or trigraphs) 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.

E(K, P) = (K \* P) mod 26

where K is the key matrix and P is plain text in vector form. Matrix multiplication of K and P generates the encrypted 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.

D(K, C) = (K-1 \* C) mod 26

where K is the key matrix and C is the ciphertext in vector form. Matrix multiplication of inverse of key matrix K and ciphertext C generates the decrypted plain text.

LAB WORK:

Source code:

#include <iostream>

#include <math.h>

using namespace std;

float encrypt[3][1], decrypt[3][1], a[3][3],

b[3][3], mes[3][1], c[3][3];

void encryption();

void decryption();

void getKeyMessage();

void inverse();

int main()

{

getKeyMessage();

encryption();

decryption();

}

void encryption()

{

int i, j, k;

for (i = 0; i < 3; i++)

for (j = 0; j < 1; j++)

for (k = 0; k < 3; k++)

encrypt[i][j] = encrypt[i][j] + a[i][k] \* mes[k][j];

cout << "\nEncrypted string is: ";

for (i = 0; i < 3; i++)

cout << (char)(fmod(encrypt[i][0], 26) + 97);

}

void decryption()

{

int i, j, k;

inverse();

for (i = 0; i < 3; i++)

for (j = 0; j < 1; j++)

for (k = 0; k < 3; k++)

decrypt[i][j] = decrypt[i][j] + b[i][k] \*

encrypt[k][j];

cout << "\nDecrypted string is: ";

for (i = 0; i < 3; i++)

cout << (char)(fmod(decrypt[i][0], 26) + 97);

cout << "\n";

}

void getKeyMessage()

{

int i, j;

char msg[3];

cout << "Enter 3x3 matrix for key (It should be inversible):\n";

for (i = 0; i < 3; i++)

for (j = 0; j < 3; j++)

{

cin >> a[i][j];

c[i][j] = a[i][j];

}

cout << "\nEnter a 3 letter string: ";

cin >> msg;

for (i = 0; i < 3; i++)

mes[i][0] = msg[i] - 97;

}

void inverse()

{

int i, j, k;

float p, q;

for (i = 0; i < 3; i++)

for (j = 0; j < 3; j++)

{

if (i == j)

b[i][j] = 1;

else

b[i][j] = 0;

}

for (k = 0; k < 3; k++)

{

for (i = 0; i < 3; i++)

{

p = c[i][k];

q = c[k][k];

for (j = 0; j < 3; j++)

{

if (i != k)

{

c[i][j] = c[i][j] \* q - p \* c[k][j];

b[i][j] = b[i][j] \* q - p \* b[k][j];

}

}

}

}

for (i = 0; i < 3; i++)

for (j = 0; j < 3; j++)

b[i][j] = b[i][j] / c[i][i];

cout << "\n\nInverse Matrix is:\n";

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

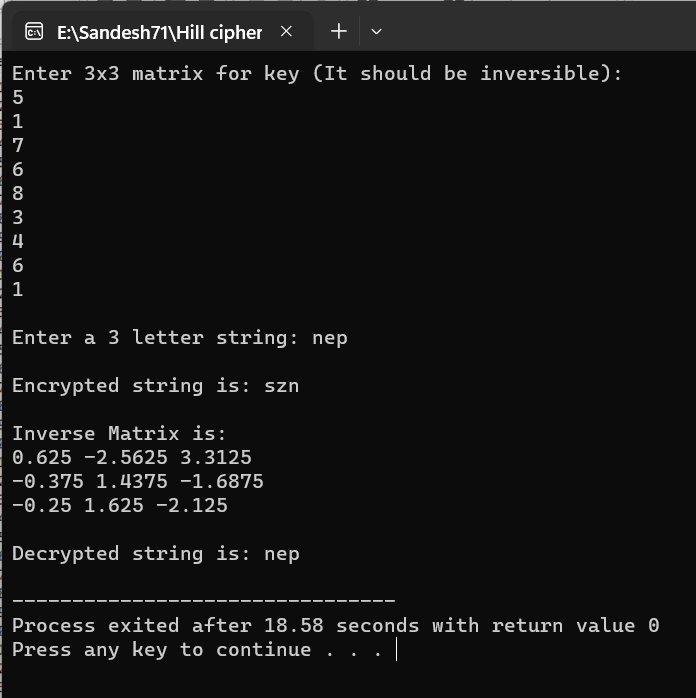
cout << b[i][j] << " ";

cout << "\n";

}

}

**Output**:



**CONCLUSION**:

Hence, in this experiment we used hill cipher technique to encrypt a 3-letter string “nep” using the 3x3 matrix provided by user as a key for encryption. The cipher text obtained from the original plain text is “szn”.

# LAB 4

TITLE: TO ENCRYPT AND DECRYPT THE PLAIN TEXT USING MONOALPHABETIC CIPHER TECHNIQUE WITH THE HELP OF C PROGRAM.

THEORY:

A mono-alphabetic substitution cipher is a type of substitution ciphers in which the equivalent letters of the plaintext are restored by the same letters of the ciphertext. Mono, which defines one, it signifies that each letter of the plaintext has a single substitute of the ciphertext. Caesar cipher is a type of Monoalphabetic cipher. It uses the similar substitution method to receive the cipher text characters for each plain text character.

In Monoalphabetic cipher, the substitute characters symbols support a random permutation of 26 letters of the alphabet. 26! Permutations of the alphabet go up to 4\*10^26. This creates it complex for the hacker to need brute force attack to gain the key. Mono-alphabetic cipher is a type of substitution where the relationship among a symbol in the plaintext and a symbol in the cipher text is continually one-to-one and it remains fixed throughout the encryption process. These ciphers are considered largely susceptible to cryptanalysis. For instance, if ‘T’ is encrypted by ‘J’ for any number of appearances in the plain text message, then ‘T’ will continually be encrypted to ‘J’. If the plaintext is “TREE”, thus the cipher text can be “ADOO” and this showcases that the cipher is possibly mono-alphabetic as both the “O” s in the plaintext are encrypted with “E” s in the cipher text.

LAB WORK:

Source code:

#include <stdio.h>

char monocipher\_encr(char);

char alpha[27][3] = {{'a', 'f'}, {'b', 'a'}, {'c', 'g'}, {'d', 'u'}, {'e', 'n'}, {'f', 'i'}, {'g', 'j'}, {'h', 'k'}, {'i', 'l'}, {'j', 'm'}, {'k', 'o'}, {'l', 'p'}, {'m', 'q'}, {'n', 'r'}, {'o', 's'}, {'p', 't'}, {'q', 'v'}, {'r', 'w'}, {'s', 'x'}, {'t', 'y'}, {'u', 'z'}, {'v', 'b'}, {'w', 'c'}, {'x', 'd'}, {'y', 'e'}, {'z', 'h'}};

char str[20];

int main(){

char str[20], str1[20];

int i;

printf("Enter the string:");

gets(str);

for (i = 0; str[i]; i++){

str1[i] = monocipher\_encr(str[i]);

}

str1[i] = '\0';

printf("After Encryption:%s\n", str1);

printf("Before Encryption:%s\n", str);

}

char monocipher\_encr(char a){

int i;

for (i = 0; i < 27; i++){

if (a == alpha[i][0])

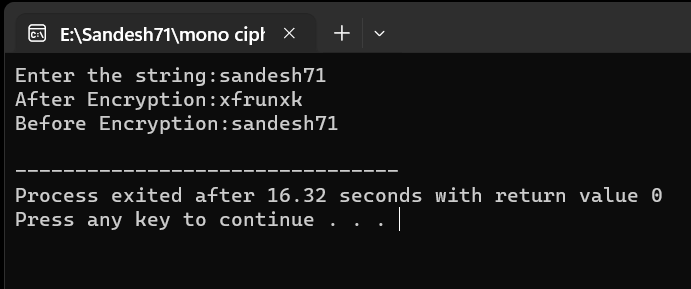
break;

}

return alpha[i][1];

}

Output:



CONCLUSION:

Hence, in this experiment we used monoalphabetic cipher technique to encrypt the given plain text “sandesh71” into cipher text “xfrunxk”.

# LAB 5

TITLE:TO ENCRYPT AND DECRYPT THE PLAIN TEXT USING VIGENERE CIPHER TECHNIQUE WITH THE HELP OF C PROGRAM.

THEORY:

Vigenère Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets. The encryption of the original text is done using the Vigenère square or Vigenère table.

* The table consists of the alphabets written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar Ciphers.
* At different points in the encryption process, the cipher uses a different alphabet from one of the rows.
* The alphabet used at each point depends on a repeating keyword.

Encryption

The plaintext(P) and key(K) are added modulo 26.

Ei = (Pi + Ki) mod 26

Decryption

Di = (Ei - Ki + 26) mod 26

Note: Di denotes the offset of the i-th character of the plaintext. Like offset of A is 0 and of B is 1 and so on.

LAB WORK:

Source code:

#include <stdio.h>

#include <ctype.h>

#include <conio.h>

#include <string.h>

#include <process.h>

void vigenere(char \*, char \*);

void encipher();

void decipher();

void main()

{

int option;

while (1)

{

printf("\n");

printf("\n 1. Encryption");

printf("\n 2. Decryption");

printf("\n 3. Exit");

printf("\n Enter your option: ");

scanf("%d", &option);

fflush(stdin);

if (option == 3)

exit(0);

else if (option == 1)

encipher();

else if (option == 2)

decipher();

else

printf("\n Invalid selection!!Try again!! ");

}

}

void encipher()

{

unsigned int i, j; // unsigned integer can represent only non negative integers

char plain[127], key[15];

printf("Enter the plaintext (maximum 128 characters): ");

gets(plain);

printf("Enter the key (maximum 16 characters): ");

gets(key);

printf("\n Encryption:");

for (i = 0, j = 0; i < strlen(plain); i++, j++)

{

if (j >= strlen(key))

{

j = 0;

}

printf("%c", 65 + (((toupper(plain[i]) - 65) + (toupper(key[j]) - 65)) % 26));

}

}

void decipher()

{

unsigned int i, j;

char plain[127], key[15];

int value;

printf("Enter the ciphertext: ");

gets(plain);

printf("Enter the key: ");

gets(key);

printf("\n Decryption:");

for (i = 0, j = 0; i < strlen(plain); i++, j++)

{

if (j >= strlen(key))

{

j = 0;

}

value = (toupper(plain[i]) - 64) - (toupper(key[j]) - 64);

if (value < 0)

{

value = value + 26;

}

printf("%c", 65 + (value % 26));

}

}

Output:



CONCLUSION:

Hence, in this experiment we use Vigenere cipher technique to encrypt the given plain text. Also we decipher the cipher text back to the original plain text using the same key.