



# SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMKUR-572103

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
CRYPTOGRAPHY AND NETWORK SECURITY LABORATORY (7RCSL01)

Student Name: Pragati Shankar	USN: 1SI19CS090	Batch No: B2	Date: 06-12-2022
-------------------------------	-----------------	--------------	------------------

## Evaluation:

Write Up (10 marks)	Clarity in concepts (10 marks)	Implementation and execution of the algorithms (10 marks)	Viva (05 marks)	Total (35 marks)

Sl.No	Name of the Faculty In-Charge	Signature
1.	H K Vedamurthy	
2.	Gururaj S P	

## Question No: 4

Write a program to perform encryption and decryption using transposition technique with column permutation given as key.

## Transposition technique:

### Encryption:

- Construct a matrix/rectangle in which column is represented by key.
- Message/ plain text must be filled row by row over a specified number of columns.
- Read the content of the matrix column by column in order of the given key to get the cipher text.

### Decryption :

- Find the number of rows and columns in a matrix :  
**No of Rows= Length of the Cipher/ Length of the Key.**  
**No of Columns is represented by digits in the key.**
- Fill the cipher text in a matrix column by column orderly.
- Read the content of the matrix column by column in order to get the plain text.
-

## CODE:

```
#include<bits/stdc++.h>
using namespace std ;

string encrypt(string pt , string key)
{
    string ct = "";
    int k = 0;

    int num_row = ceil((float) pt.length()/key.length());
    int num_col = key.length();
    char mat[num_row][num_col];

    cout << "\nEncryption Matrix :" << endl;
    cout << "-----" << endl;
    for(int i=0; i<num_row ; i++)
    {
        for(int j=0; j<num_col; j++)
        {
            if(k < pt.length())
            {
                cout << (mat[i][j] = pt[k++]) << " ";
            }
            else
            {
                cout << (mat[i][j] = 'x') << " ";
            }
        }
        cout << endl;
    }

    for(int i=0; i<num_col; i++)
    {
        for(int j=0; j<num_row; j++)
        {
            ct += mat[j][key.find(i+'1')];
        }
    }
    return ct;
}

string decrypt(string ct , string key)
{
    string pt = "";
    int k = 0;

    int num_row = ceil((float)ct.length() / key.length());
    int num_col = key.length();
    char mat[num_row][num_col];
```

```

for(int i=0; i<num_col; i++)
{
    for(int j=0; j<num_row; j++)
    {
        mat[j][key.find(i+'1')] = ct[k++];
    }
}

cout << "\nDecryption Matrix :" << endl;
cout << "-----" << endl;

for(int i=0; i<num_row ; i++)
{
    for(int j=0; j<num_col; j++)
    {
        cout << mat[i][j] << " ";
        pt += mat[i][j];
    }
    cout << endl;
}
return pt;
}

int main()
{
    string plaintext , key , ciphertext , decrypttext;

    cout << "Enter text : ";
    cin >> plaintext;

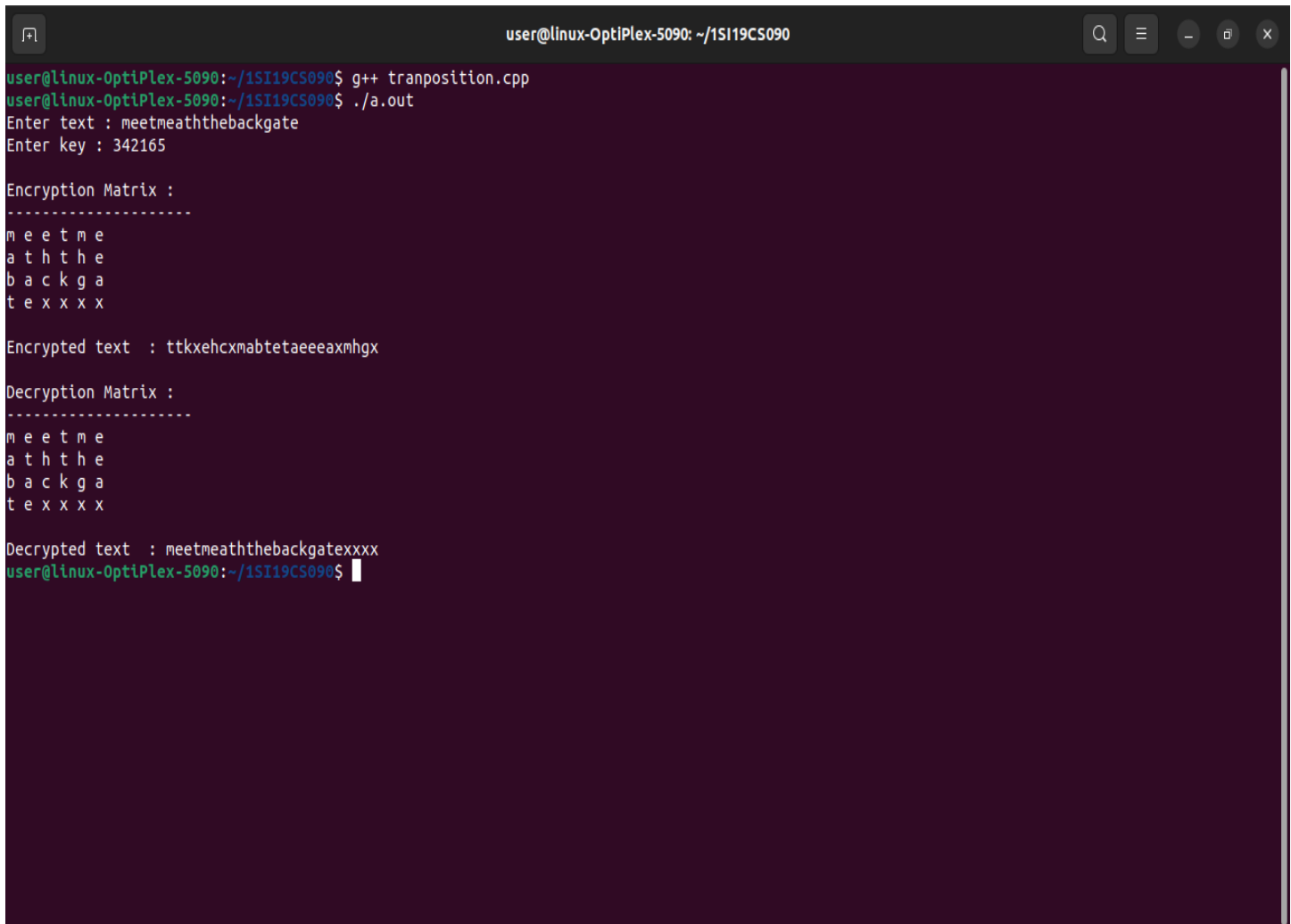
    cout << "Enter key : ";
    cin >> key;

    ciphertext = encrypt(plaintext , key);
    cout << "\nEncrypted text \t: " << ciphertext << endl;

    decrypttext = decrypt(ciphertext , key);
    cout << "\nDecrypted text \t: " << decrypttext << endl;
}

```

## Output Screenshot:



```
user@linux-OptiPlex-5090: ~/1SI19CS090
user@linux-OptiPlex-5090:~/1SI19CS090$ g++ tranposition.cpp
user@linux-OptiPlex-5090:~/1SI19CS090$ ./a.out
Enter text : meetmeaththebackgate
Enter key : 342165

Encryption Matrix :
-----
m e e t m e
a t h t h e
b a c k g a
t e x x x x

Encrypted text : ttkxehcxmabtetaeeexnhgx

Decryption Matrix :
-----
m e e t m e
a t h t h e
b a c k g a
t e x x x x

Decrypted text : meetmeaththebackgatexxxx
user@linux-OptiPlex-5090:~/1SI19CS090$
```