SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMKUR-572103

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CRYPTOGRAPHY AND NETWORK SECURITY LAB (7RCSL01)

Student Name: Khalid Farooq	USN: 1SI18CS046	Batch No: A3	Date:

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.	Sunitha N R	
2.	A H Shanthakumara	

Question No: 4

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

Transposition technique:

Write the message in a rectangle, row by row, and read the message off, column by column, but permute the order of the columns. The order of the columns then becomes the key to the algorithm.

Example:

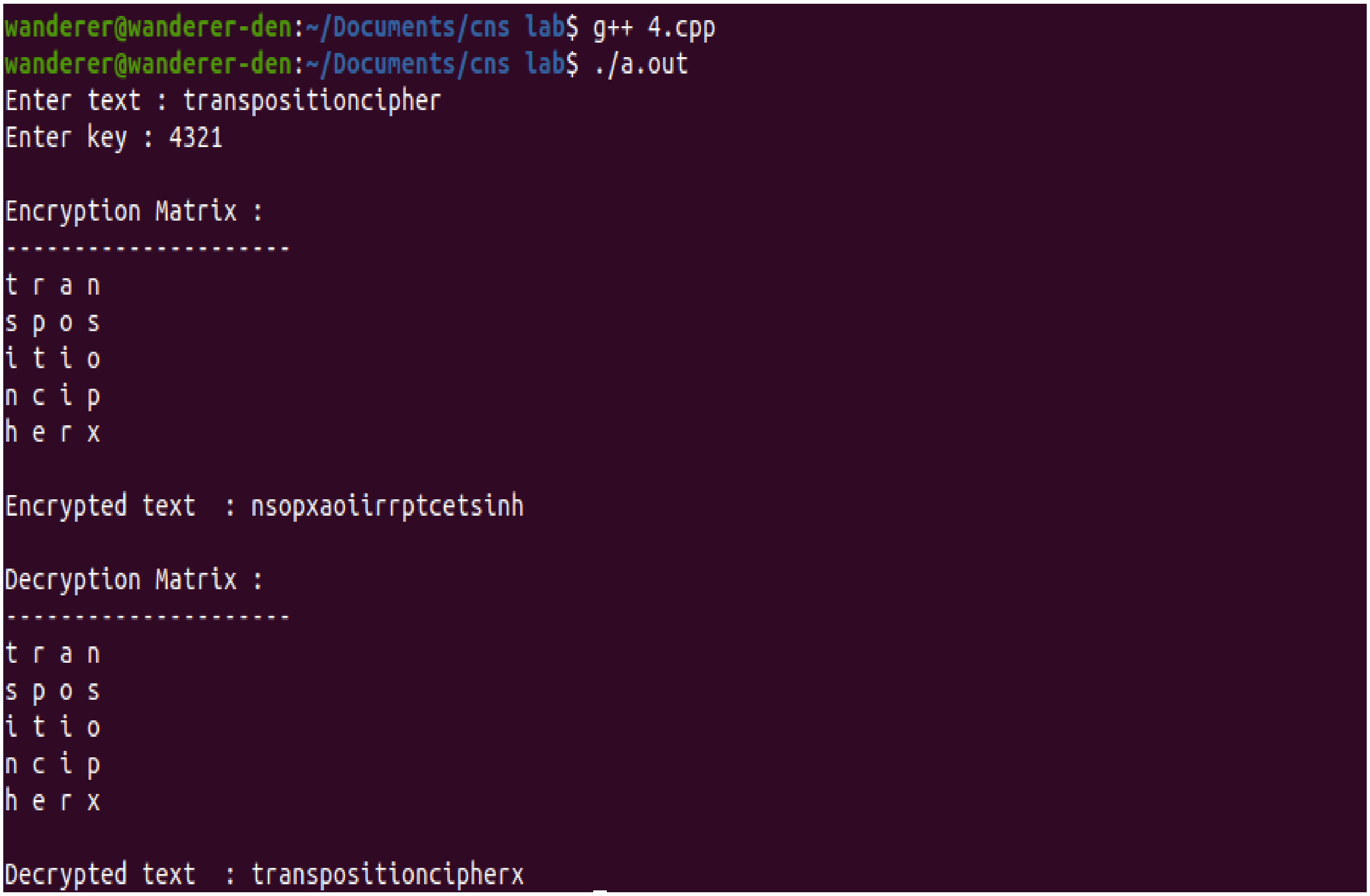
```
Key:           4 3 1 2 5 6 7
Plain text:    a t t a c k p
               o s t p o n e
               d u n t i l t
               w o a m x y z
Cipher text: TTNAAPTMTSUOAODWCOIXKNLYPETZ
```

Let us consider the key is 4312567. To encrypt, start with the column that is labeled 1, in this case column 3. Write down all the letters in that column. Proceed to column 4, which is labeled 2, then column 2, then column 1, then columns 5, 6, and 7.

Program:

```
#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 , decryptext;

    cout << "Enter text : ";
    getline(cin,plaintext);

    cout << "Enter key : ";
    getline(cin,key);

    plaintext.erase(remove(plaintext.begin(), plaintext.end(), ' '),
plaintext.end());

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

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

Output:



```
wanderer@wanderer-den:~/Documents/cns lab$ g++ 4.cpp
wanderer@wanderer-den:~/Documents/cns lab$ ./a.out
Enter text : transpositioncipher
Enter key : 4321

Encryption Matrix :
---------------------
t r a n
s p o s
i t i o
n c i p
h e r x

Encrypted text  : nsopxaoiirrptcetsinh

Decryption Matrix :
---------------------
t r a n
s p o s
i t i o
n c i p
h e r x

Decrypted text  : transpositioncipherx
```