INFORMATION SECURITY AND CRYPTOGRAPHY
ASSIGNMENT- 4

U20CS005
BANSI MARAKANA

1. Implement columnar transposition cipher.
2. Implement Vernam cipher.

```
#include <iostream>
#include <string>
#include <cmath>
#include <cstdlib>
#include <ctime>
#include <algorithm>
#include <fstream>
using namespace std;
typedef long long ll;
string encrypt_transposition(string plain_text, string key)
{
    int rows = ceil((float)plain_text.length() / key.length()), cols =
key.length();
    char transposition_matrix[rows][cols];
    int k = 0;
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
        {
            if (k >= plain_text.size())
                transposition_matrix[i][j] = 'z';
            else
                transposition_matrix[i][j] = plain_text[k];
            k++;
        }
    }
    string ref_key = key;
    sort(ref_key.begin(), ref_key.end());
    string encrypted;
    int i = 0, j = 0;
    while (i < key.length())
    {
        if (key[i] == ref_key[j])
        {
```

```
            for (int k = 0; k < rows; k++)
                encrypted += transposition_matrix[k][i];
            j++;
            i = -1;
        }
        i++;
    }
    for (int i = 0; i < key.length(); i++)
        cout << "\t" << key[i];
    cout <<
"\n\t-----------------------------------------------------------------\n";
    for (int i = 0; i < rows; i++)
    {
        for (int j = 0; j < cols; j++)
            cout << "\t" << (char)transposition_matrix[i][j];
        cout << endl;
    }
    cout <<
"\t-----------------------------------------------------------------\n";
    return encrypted;
}
string decrypt_transposition(string encrypted, string key)
{
    string ref_key = key;
    sort(ref_key.begin(), ref_key.end());
    int rows = key.length(), cols = encrypted.length() / key.length();
    char transposition_matrix[rows][cols];
    int i = 0, k = 0;
    while (i < key.length())
    {
        if (key[i] == ref_key[k])
        {
            k = k * cols;
            for (int j = 0; j < cols; j++)
                transposition_matrix[i][j] = encrypted[k++];
            k = -1;
            i++;
        }
        k++;
    }
```

```
    cout <<
"\n\t-------------------------------------------------------------------------
----------------------------------------\n";
    for (int i = 0; i < rows; i++)
    {
        cout << "\t" << key[i] << "|";
        for (int j = 0; j < cols; j++)
            cout << "\t" << (char)transposition_matrix[i][j];
        cout << endl;
    }
    cout <<
"\t-------------------------------------------------------------------------
----------------------------------------\n";
    string decrypted;
    for (int i = 0; i < cols; i++)
        for (int j = 0; j < rows; j++)
            decrypted += transposition_matrix[j][i];
    return decrypted;
}
string convertToUpper(string text)
{
    string str = "";
    ll length = text.size();
    for (int i = 0; i < length; i++)
        if (isalpha(text[i]))
            str += toupper(text[i]);
    return str;
}
string generate_key(ll size)
{
    char alphabet[26] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J',
'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y',
'Z'};
    string key = "";
    srand((unsigned)time(NULL));
    for (ll i = 0; i < size; i++)
        key += alphabet[rand() % 26];
    return key;
}
string encrypt_vernam(string plain_text, string key)
```

```
{
    string cipher_text;
    for (int i = 0; i < plain_text.length(); i++)
    {
        int xorr = (plain_text[i] - 'A') ^ (key[i] - 'A');
        if (xorr > 25)
            xorr = xorr - 26;
        cipher_text += xorr + 'A';
    }
    return cipher_text;
}
void decrypt_vernam(string cipher_text, string key)
{
    string decrypted1, decrypted2;
    for (int i = 0; i < cipher_text.length(); i++)
    {
        int xorr1 = (cipher_text[i] - 'A') ^ (key[i] - 'A');
        int xorr2 = (cipher_text[i] + 26 - 'A') ^ (key[i] - 'A');
        decrypted1 += (xorr1 + 'A');
        decrypted2 += (xorr2 + 'A');
    }
    cout << "\t" << decrypted1 << endl;
    cout << "\t" << decrypted2 << endl;
}
int main()
{
    int choice;
    string key;
    cout << "1. Encryption using columnar transposition cipher \n2.
Decryption using columnar transposition cipher";
    cout << "\n3. Vernam cipher\n5. Exit";
    while (1)
    {
        cout << "\nEnter your choice: ";
        cin >> choice;
        switch (choice)
        {
        case 1:
        {
```

```
            string fname, fname1, key, plain_text = "", cipher_text = "",
text = "";
            cout << "\tEnter file name to read plain text: ";
            cin >> fname;
            cout << "\tEnter file name to write cipher text: ";
            cin >> fname1;
            cout << "\tEnter encryption key: ";
            cin >> key;
            ifstream fin;
            ofstream fout;
            fin.open(fname + ".txt");
            if (!fin.is_open())
            {
                cout << "\tFile does not exist!!";
                return 0;
            }
            fout.open(fname1 + ".txt");
            cout << "\tEncrypted text is: " << endl;
            while (getline(fin, text))
                plain_text += text;

            cipher_text += encrypt_transposition(plain_text, key);
            cout << "\t" << cipher_text << endl;
            fout << cipher_text;
            fout.close();
            fin.close();
            break;
        }
        case 2:
        {
            string fname, fname1, key, plain_text, cipher_text, text;
            cout << "\tEnter file name to read cipher text: ";
            cin >> fname;
            cout << "\tEnter file name to write plain text: ";
            cin >> fname1;
            cout << "\tEnter decryption key: ";
            cin >> key;
            ifstream fin;
            ofstream fout;
            fin.open(fname + ".txt");
```

```
            if (!fin.is_open())
            {
                cout << "\tFile does not exist!!";
                return 0;
            }
            fout.open(fname1 + ".txt");
            while (getline(fin, text))
                cipher_text += text;
            plain_text += decrypt_transposition(cipher_text, key);
            cout << "\t" << plain_text << endl;
            fout << plain_text;
            fout.close();
            fin.close();
            break;
        }
        case 3:
        {
            string fname, fname1, text, txt, key, plain_text, cipher_text;
            ll leng;
            cout << "\tEnter file name to read plain text: ";
            cin >> fname;
            ifstream fin;
            fin.open(fname + ".txt");
            if (!fin.is_open())
            {
                cout << "\tFile does not exist!!";
                return 0;
            }
            while (getline(fin, text))
                plain_text += text;
            fin.close();
            text = convertToUpper(text);
            leng = text.length();
            cout << "\tEnter file name to previous keys: ";
            cin >> fname1;
            fin.open(fname1 + ".txt");
            if (!fin.is_open())
            {
                cout << "\tFile does not exist!!";
                return 0;
```

```
            }
            key = generate_key(leng);
            while (getline(fin, txt))
            {
                if (0 == key.compare(txt))
                {
                    key = generate_key(leng);
                    fin.seekg(0);
                }
                else
                    continue;
            }
            fin.close();
            ofstream fout;
            fout.open(fname1 + ".txt", ios::app);
            fout << key << endl;
            fout.close();
            cout << "\n\tPlain text is: \n\t" << text << endl;
            cout << "\n\tKey is: \n\t" << key << endl;
            cout << "\n\tEncrypted text is: " << endl;
            cipher_text = encrypt_vernam(text, key);
            cout << "\t" << cipher_text << endl;
            cout << "\n\tTwo possible decrypted text are: " << endl;
            decrypt_vernam(cipher_text, key);
            break;
        }
        case 4:
            exit(0);
            break;
        default:
            cout << "Please enter valid choice!!";
            break;
        }
    }
}
```

1. Columnar Transposition Cipher:

p.txt

```
p.txt
Hello World!! I am learning Cryptography.
```

c.txt

```
c.txt
eW!aagpazH d enyr.olIlirgyzlo!mr tpzlr  nCohz
```

Encryption:

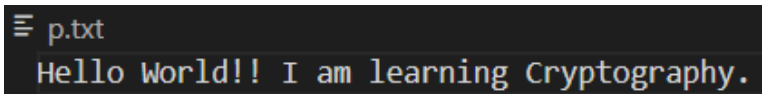
```
PS D:\BANSI MARAKANA\ISC> ./a
1. Encryption using columnar transposition cipher
2. Decryption using columnar transposition cipher
3. Vernam cipher
5. Exit
Enter your choice: 1
        Enter file name to read plain text: p
        Enter file name to write cipher text: c
        Enter encryption key: bansi
        Encrypted text is:
        b       a       n       s       i
        ------------------------------------------------------------
        H       e       l       l       o
                W       o       r       l
        d       !       !               I
                a       m               l
        e       a       r       n       i
        n       g               C       r
        y       p       t       o       g
        r       a       p       h       y
        .       z       z       z       z
        ------------------------------------------------------------
        eW!aagpazH d enyr.olIlirgyzlo!mr tpzlr  nCohz
```

Decryption:

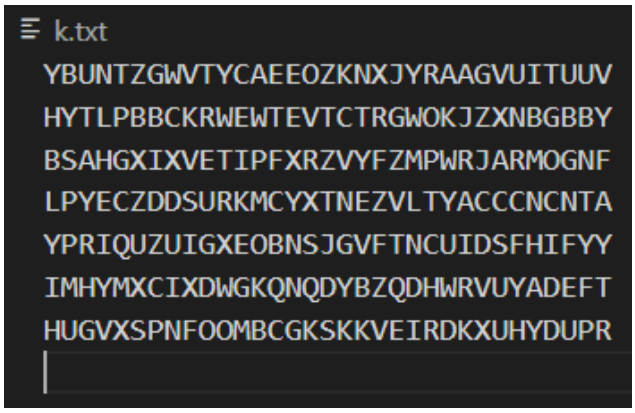
```
PS D:\BANSI MARAKANA\ISC> ./a
1. Encryption using columnar transposition cipher
2. Decryption using columnar transposition cipher
3. Vernam cipher
5. Exit
Enter your choice: 2
        Enter file name to read cipher text: c
        Enter file name to write plain text: p
        Enter decryption key: bansi

        ------------------------------------------------------------------------------------------------------------
        b|      H               d               e       n       y       r       .
        a|      e       W       !       a       a       g       p       a       z
        n|      l       o       !       m       r               t       p       z
        s|      l       r                       n       C       o       h       z
        i|      o       l       I       l       i       r       g       y       z
        ------------------------------------------------------------------------------------------------------------
        Hello World!! I am learning Cryptography.zzzz
```

2. Vernam Cipher:

p.txt



k.txt




```
PS D:\BANSI MARAKANA\ISC> ./a
1. Encryption using columnar transposition cipher
2. Decryption using columnar transposition cipher
3. Vernam cipher
5. Exit
Enter your choice: 3
        Enter file name to read plain text: p
        Enter file name to previous keys: k

        Plain text is:
        HELLOWORLDIAMLEARNINGCRYPTOGRAPHY

        Key is:
        RKUSKWGDQMZAQZRWNMNJCSJOSFHSPCBVJ

        Encrypted text is:
        WOFZEAISBPRACSVWCBFEEQYWDWJUECOSR

        Two possible decrypted text are:
        HERLOWORRDIASLEAPNINGCRYRTOGLAPHY
        bcLbUMepLfs[MvagRXSX]y|aPve}R_jzc
```