**Roll Number : 21BCE006**

**Course code : 2CSDE54**

**Course Name : Information and Network Security**

**PRACTICAL 2**

Perform encryption, decryption using the following substitution techniques

1. Playfair
2. Hill Cipher

The **Playfair Cipher** Encryption Algorithm:

The Algorithm consists of 2 steps:

* + - 1. Generate the key square:

1. The key square is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table (as the table can hold only 25 alphabets). If the plaintext contains J, then it is replaced by I.
2. The initial alphabets in the key square are the unique alphabets of the key in the order in which they appear followed by the remaining letters of the alphabet in order.
3. Algorithm to encrypt the plain text: The plaintext is split into pairs of two letters (digraphs). If there is an odd number of letters, a Z is added to the last letter.

For example:

**PlainText: "instruments"**

**After Split: 'in' 'st' 'ru' 'me' 'nt' 'sz'**

Plain Text: **“hello”**

After Split: **‘he’ ‘lx’ ‘lo’**

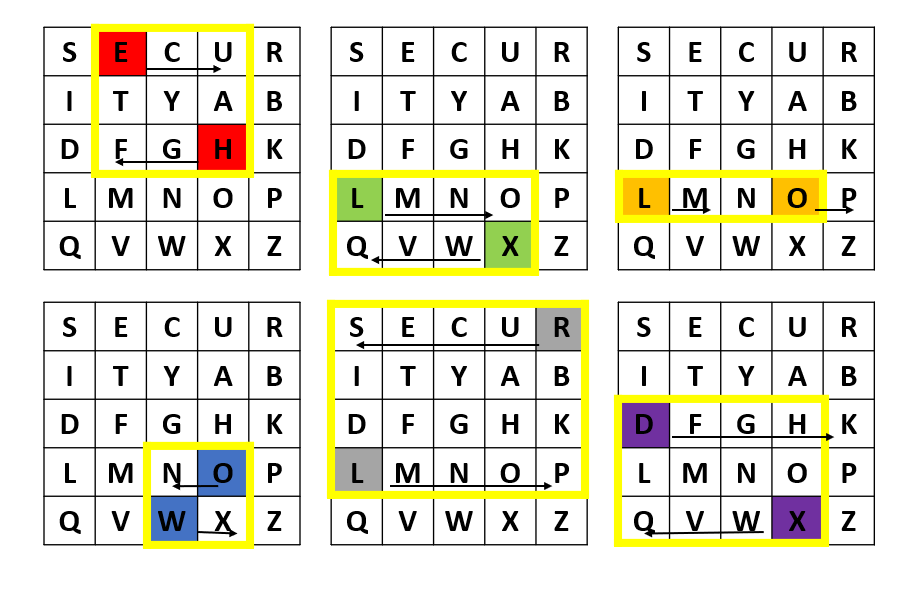
Plain Text: **“helloe”**

AfterSplit: **‘he’ ‘lx’ ‘lo’ ‘ex’**

**Rules for Encryption:**

1. If both letters in the pair are in the same row of the key square, we replace each letter with the letter to its right (wrapping around if necessary).
2. If both letters in the pair are in the same column of the key square, we replace each letter with the letter below it (wrapping around if necessary).
3. If the letters are in different rows and columns, we form a rectangle with the pair and replace each letter with the letter at the rectangle’s opposite corner (moving only left or right).

Using the matrix with the keyword SECURITY, let?s find the row and column of each pair and apply the encryption rules to HELLOWORLD, whose pairs are HE LX LO WO RL DX:

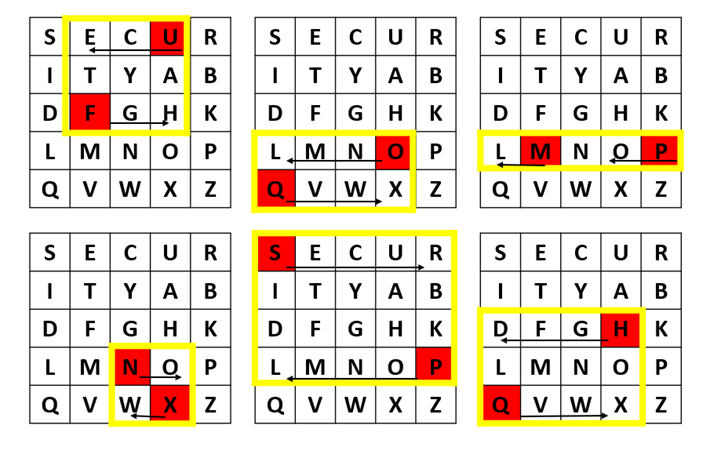


After applying the encryption rules to all the letter pairs, we get FUOQMPXNSPHQ.

**Decryption rule:** The decryption rules are reverse encryption rules.

Let’s decrypt the message FUOQMPXNSPHQ using these decryption rules.

The pairs are:



At this point, we have HELXLOWORLDX. After dealing with Xs, we get HELLOWORLD

CODE :

#include<bits/stdc++.h>

using namespace std;

vector<vector<char>> playfair(5,vector<char>(5,'-'));

map<char,pair<int,int>> storeIndex;

void Initialise\_Matrix(string a,string b)

{

    map<char,bool> m;

    string temp;

    int index=0;

    for(auto it:b){

        it=tolower(it);

        if(it=='j')

            it='i';

        if(m.find(it)==m.end()){

            temp+=it;

            m[it]=1;

        }

    }

    for(auto it:a){

        it=tolower(it);

        if(it=='j')

            it='i';

        if(m.find(it)==m.end()){

            temp+=it;

            m[it]=1;

        }

    }

    for(char ch='a';ch<='z';++ch)

        if(m.find(ch)==m.end() && ch!='j')

            temp+=ch;

    // cout<<temp.size()<<" "<<temp<<endl;

    for(int i=0;i<5;++i){

        for(int j=0;j<5;++j){

            playfair[i][j]=temp[index++];

        }

    }

    cout<<"Matrix used for Encryption is : "<<endl;

    for(int i=0;i<5;++i){

        for(int j=0;j<5;++j){

            storeIndex[playfair[i][j]]={i,j};

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

        }

        cout<<endl;

    }

    cout<<endl;

}

string convert(string s)

{

    string s1="";

    for(int i=0;i<s.size();i++){

        if(i+1==s.size()){

            s1.push\_back(tolower(s[i]));

            s1.push\_back('x');

            continue;

        }

        if(s[i]!=s[i+1]){

            s1.push\_back(tolower(s[i]));

            s1.push\_back(tolower(s[i+1]));

            i++;

        }if(s[i]==s[i+1]){

            s1.push\_back(tolower(s[i]));

            s1.push\_back('x');

        }

    }

    return s1;

}

string encrypt(string s,string k)

{

    string s1=convert(s);

    cout<<"Original String : "<<s1<<endl;

    string s2="";

    for(int i=0;i<s1.size();++i){

        pair<int,int> a=storeIndex[s1[i]];

        pair<int,int> b=storeIndex[s1[i+1]];

        if(a.first==b.first){

            s2+=playfair[(a.first+1)%5][(a.second)];

            s2+=playfair[(b.first+1)%5][(b.second)];

            i++;

        }else if(a.second==b.second){

            s2+=playfair[(a.first)][(a.second+1)%5];

            s2+=playfair[(b.first)][(b.second+1)%5];

            i++;

        }else{

            s2+=playfair[b.first][a.second];

            s2+=playfair[a.first][b.second];

            i++;

        }

    }

    // cout<<"String After Conversion : "<<s2<<endl;

    return s2;

}

string decrypt(string s,string k)

{

    // string s1=convert(s);

    // cout<<"Original String : "<<s1<<endl;

    string s1=s;

    string s2="";

    for(int i=0;i<s1.size();++i){

        pair<int,int> a=storeIndex[s1[i]];

        pair<int,int> b=storeIndex[s1[i+1]];

        if(a.first==b.first){

            s2+=playfair[(a.first+4)%5][(a.second)];

            s2+=playfair[(b.first+4)%5][(b.second)];

            i++;

        }else if(a.second==b.second){

            s2+=playfair[(a.first)][(a.second+4)%5];

            s2+=playfair[(b.first)][(b.second+4)%5];

            i++;

        }else{

            s2+=playfair[b.first][a.second];

            s2+=playfair[a.first][b.second];

            i++;

        }

    }

    // cout<<"String After Conversion : "<<s2<<endl;

    return s2;

}

int main()

{

    string s,k;

    // cout<<"Enter string : ";

    // cin>>s;

    ifstream f1("./input2a.txt");

    ofstream f2("./output2a.txt");

    f1>>s;

    cout<<"String is : "<<s<<endl;

    cout<<"Enter key : ";

    cin>>k;

    Initialise\_Matrix(s,k);

    string s1=encrypt(s,k);

    f2<<"String After Encryption : "<<s1<<endl;

    string s2=decrypt(s1,k);

    f2<<"String after decryption : "<<s2<<endl;

    return 0;

}

INPUT / OUTPUT :

Input file content :

Varad

Output file content :

String After Encryption : gkidbz

String after decryption : varadx

Terminal Output :

PS C:\Users\Varad Acharya\Desktop\Nirma 6th sem Study Material\INS all practicals\INS practical 2> ./p2a

String is : Varad

Enter key : HelloSecurity

Matrix used for Encryption is :

h e l o s

c u r i t

y v a d b

f g k m n

p q w x z

Original String : varadx

**Instruction:** You need to take the **input.txt** file and generate the cipher text in the **output.txt**

**Hill cipher:** 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).

**Input :** Plaintext: ACT

Key: GYBNQKURP

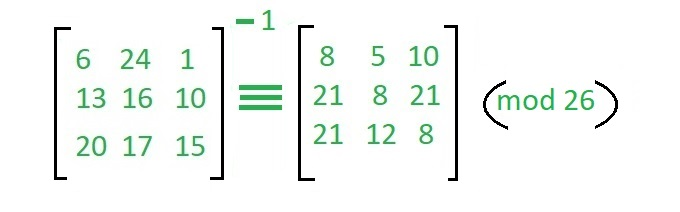
**Output :** Ciphertext: POH

We have to encrypt the message ‘ACT’ (n=3).

|  |  |
| --- | --- |
| The key is ‘GYBNQKURP’ which can be written as the nxn matrix Lightbox | The message ‘ACT’ is written as vector: Lightbox |
| The enciphered vector is given as: which corresponds to ciphertext of ‘POH’ Lightbox | |

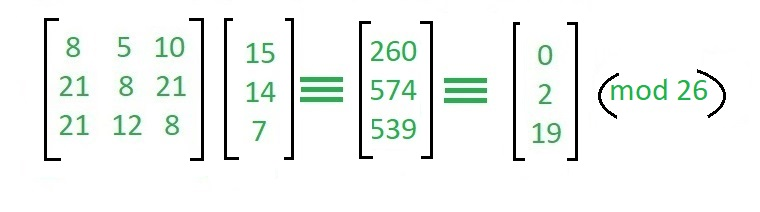
**Decryption :**

To decrypt the message, we turn the ciphertext back into a vector, then simply multiply by the inverse matrix of the key matrix (IFKVIVVMI in letters).The inverse of the matrix used in the previous example is:



For the previous Ciphertext ‘POH’: which gives us back ‘ACT’.

Assume that all the alphabets are in upper case.



CODE :

#include<bits/stdc++.h>

using namespace std;

vector<int> multiply(int n,vector<int> A,vector<vector<int>> B)

{

    vector<int> C(n,0);

    for(int i=0;i<n;++i){

        for(int j=0;j<n;++j){

            C[i]+=(A[j]\*B[i][j]);

        }

        C[i]%=26;

    }

    return C;

}

string encrypt(int n,vector<int> s,vector<vector<int>> v)

{

    vector<int> V=multiply(n,s,v);

    string res;

    for(auto it : V)

        res+=char('A'+it);

    return res;

}

int modInverse(int a,int m)

{

    a=a%m;

    for(int x=1;x<m;x++)

        if((a\*x)%m==1)

            return x;

    return 1;

}

int determinant(int a,int b,int c,int d)

{

    return ((a\*d)-(b\*c))%26;

}

vector<vector<int>> modInverseMatrix(vector<vector<int>> key)

{

    int det=(key[0][0]\*key[1][1]-key[0][1]\*key[1][0])%26;

    int invDet=modInverse(det,26);

    vector<vector<int>> invMatrix(2,vector<int>(2));

    invMatrix[0][0]=key[1][1]\*invDet%26;

    invMatrix[0][1]=(-key[0][1]\*invDet)%26;

    invMatrix[1][0]=(-key[1][0]\*invDet)%26;

    invMatrix[1][1]=key[0][0]\*invDet%26;

    return invMatrix;

}

string decrypt(int n,vector<int> s,vector<vector<int>> key)

{

    vector<vector<int>> invKey=modInverseMatrix(key);

    vector<int> decrypted=multiply(n,s,invKey);

    string res;

    for(auto it : decrypted)

        res+=char('A'+it%26);

    return res;

}

int main()

{

    string s,k;

    ifstream f1("./input2b.txt");

    ofstream f2("./output2b.txt");

    f1>>s;

    vector<int> s1;

    int n=s.size();

    for (auto it : s)

        s1.push\_back(toupper(it) - 'A');

    cout<<"Enter key : ";

    cin>>k;

    int keySize=ceil(sqrt(k.size())),index=0;;

    while(k.size()<keySize\*keySize)

        k+='A';

    vector<vector<int>> v(keySize,vector<int>(keySize));

    for(int i=0;i<keySize;++i)

        for(int j=0;j<keySize;++j)

            v[i][j]=toupper(k[index++])-'A';

    string res=encrypt(n,s1,v);

    f2<<res<<endl;

    vector<int> resvec;

    for(auto it:res)

        resvec.push\_back(it+'A');

    string original=decrypt(n,resvec,v);

    f2<<original<<endl;

    return 0;

}

INPUT / OUTPUT

Input file content :

AFC

Output file Content :

MMU

AFC

Terminal Output :

PS C:\Users\Varad Acharya\Desktop\Nirma 6th sem Study Material\INS all practicals\INS practical 2> g++ -o ./p2b ./Practical2b.cpp

PS C:\Users\Varad Acharya\Desktop\Nirma 6th sem Study Material\INS all practicals\INS practical 2> ./p2b

Enter key : ABCfgidgE