

Cryptography and Network Security

Name: Piyush Mhaske
PRN : 2019BTECS00089

Batch: B3

Assignment 5

Objective: Columnar Transposition

Theory:

The Columnar Transposition Cipher is a form of transposition cipher just like Rail Fence Cipher. Columnar Transposition involves writing the plaintext out in rows, and then reading the ciphertext off in columns one by one.

Code:

```
//code by :- Piyush Mhaske
#include <bits/stdc++.h>
#define ll long long
#define ull unsigned long long
#define pb emplace_back
#define po pop_back
#define vi vector<ll>
#define vii vector<vector<ll>>
using namespace std;
void file(){
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);}
ll M = 1e9 + 7;
string Decrypt(string CipherText, string key){
    unordered_map<int, vector<char>> mp;
    int n = key.size();
    int m = CipherText.size();
    string ans="";

    int col = m/n;
    int rem = m%n;
    vector<int> len(key.size(), col);
    if(rem!=0){
        for(int i=0; i<n; i++){
            if(rem>0){
                len[i] = col+1;
            }
        }
    }
    for(int i=0; i<m; i++){
        int col = i/n;
        int rem = i%n;
        ans += CipherText[rem];
    }
    return ans;
}
```



```

        temp.push_back({key[i], i});
    }

    sort(temp.begin(), temp.end());

    int j=0;
    int m = key.size();
    for(int i=0; i<PlainText.size(); i++){
        mp[j].push_back(PlainText[i]);
        j = (j+1)%m;
    }

    for(auto x:temp){
        if(mp.count(x.second)){
            cout<<x.first<<": ";
            for(auto y:mp[x.second]){
                cout<<y;
                ans+=y;
            }
            cout<<"\n";
        }
    }

    return ans;
}



int main()
{
    file();
    string PlainText, CipherText, key;
    cin>>PlainText>>key;

    CipherText = Columnar(PlainText, key);
    cout<<"The Encrypted text is: "<<CipherText<<"\n";

    cout<<"Decrypting\n";
    string decrypt = Decrypt(CipherText, key);
    decrypt = decrypt.substr(0, CipherText.size());
    cout<<"The decrypt text is : "<<decrypt<<"\n";
    return 0;
}

```

Output:

^ Testcase 1 Passed 30ms  

Input: Copy



thisisthekey
get

Expected Output: Copy

e: hihe
g: tstk
t: isey
hihetstkisey

Received Output: Copy

e: hihe
g: tstk
t: isey
hihetstkisey

^ Testcase 2 Passed 32ms  

Input: Copy

success
key

Expected Output: Copy

e: ue
k: scs
y: cs
uescscs

Received Output: Copy

e: ue
k: scs
y: cs
uescscs

Conclusion:

Easy to Crack the message can be predicted if it is small.