Principle of Information Security--HW1

张溢弛 3180103772

Part 1：

- **What are the differences between Transposition Cipher and Substitution Cipher? Please give some examples of them.**

Answer：Substitution Cipher is a kind of cipher that substitude each word of the plaintext with another word using a substitution-table, and the key of substitution cipher is the substitution table.

Transposition cipher is a kind of cipher that change the permutation of the plaintext

Examples:

1. The Vigenère Square
2. Caesar Shift
3. Queen Mary‘s Cipher

Part 2：

- Programming Environment
  - OS: win 10
  - Editor: Visual Studio Code
  - Compiler: g++

- Algorithm Introduce

  Encryption

  - input the plaintext(only digits and letters are allowed)
  - create a **key** with the cycle sequence of "3180103772"
  - for each word of the plaintext, use such method to get the substitution cipher

    $$substiotution[i] = (ASCIIcode + key[i]) * f(key[i])$$

  - and f(key[i]) is defined as `a large number - key[i]`,like 3429
  - change the order of the substitution sequence in the group of 3, change group change the order 1,2,3 to 2,3,1
  - reverse the substiotution sequence
  - output thr result of encryption

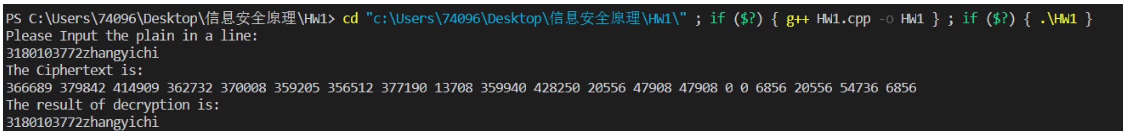
  Decryption

  - reverse the substiotution sequence
  - change the order of the substitution sequence in the group of 3, change the order 1,2,3 to 3,1,2
  - compute the ASCII code of each one of the sequence using the public ke

    $$ASCIIcode = (substitution[i]/f(key[i]) - key[i])$$

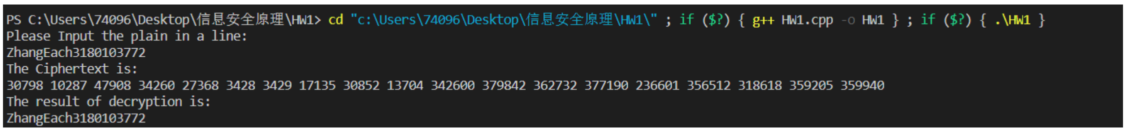
  - find the plaintext by search the ASCII code list
  - output the result of decryption

- Experiment Result
  - test data 1: `3180103772zhangyichi`,测试结果正确



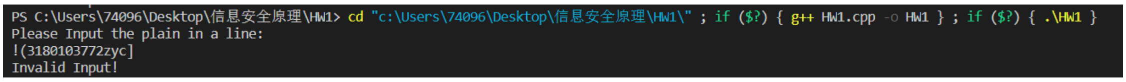
```
PS C:\Users\74096\Desktop\信息安全原理\HW1> cd "c:\Users\74096\Desktop\信息安全原理\HW1\" ; if ($?) { g++ HW1.cpp -o HW1 } ; if ($?) { .\HW1 }
Please Input the plain in a line:
3180103772zhangyichi
The Ciphertext is:
366689 379842 414909 362732 370008 359205 356512 377190 13708 359940 428250 20556 47908 47908 0 0 6856 20556 54736 6856
The result of decryption is:
3180103772zhangyichi
```

  - test data 2:(有大写字母) `ZhangEach3180103772`，测试结果正确



```
PS C:\Users\74096\Desktop\信息安全原理\HW1> cd "c:\Users\74096\Desktop\信息安全原理\HW1\" ; if ($?) { g++ HW1.cpp -o HW1 } ; if ($?) { .\HW1 }
Please Input the plain in a line:
ZhangEach3180103772
The Ciphertext is:
30798 10287 47908 34260 27368 3428 3429 17135 30852 13704 342600 379842 362732 377190 236601 356512 318618 359205 359940
The result of decryption is:
ZhangEach3180103772
```

  - test data 3:(Invalid Input) `!(3180103772zyc]`,测试结果正确



```
PS C:\Users\74096\Desktop\信息安全原理\HW1> cd "c:\Users\74096\Desktop\信息安全原理\HW1\" ; if ($?) { g++ HW1.cpp -o HW1 } ; if ($?) { .\HW1 }
Please Input the plain in a line:
!(3180103772zyc]
Invalid Input!
```

- Summary and experience

  信息安全原理的第一次作业，总体难度不是很大，在设计自己的加密算法时，因为主要用了代替法和置换法的传统加密方法，因此加密算法的总体复杂度较为浅显，但我也尽可能将加密算法的复杂度提高，想要进一步提高自己设计的密码算法的难度，还需要继续深入学习密码学的相关知识，将跟多复杂的数学方法引入加密算法中。

- Source code

```cpp
#include<iostream>
#include<string>
#include<vector>
using namespace std;

//author:zhangyichi-3180103772
string CreateKey(int n);
int checkplain(string s);
int f(string s,int i);
void OutPutCode(vector<int> s);

int main()
{
    int i;
    string plain,key;

    cout<<"Please Input the plain in a line:"<<endl;
    cin>>plain;
    if(!checkplain(plain)){
        cout<<"Invalid Input!"<<endl;
        return 0;
    }
    key=CreateKey(plain.size());

    //Encryption
    vector<int> cipher1,cipher;
    //step1:change to ASCII code
    for(i=0;i<plain.size();i++){
        int unit;
        if(plain[i]>='0' && plain[i]<='9'){
            unit=plain[i]-'0'+key[i]-'0';
        }
        else if(plain[i]>='A' && plain[i]<='Z'){
            unit=plain[i]-'A'+65+key[i]-'0';
        }
        else if(plain[i]>='a' && plain[i]<='z'){
            unit=plain[i]-'a'+97+key[i]-'0';
        }
        unit*=f(key,i);
        cipher1.push_back(unit);
    }

    //OutPutCode(cipher1);
    //step 2: a swap of each 3
    for(i=0;i<=cipher1.size()-3;i+=3){
        int t=cipher1[i];
        cipher1[i]=cipher1[i+1];
        cipher1[i+1]=cipher1[i+2];
        cipher1[i+2]=t;
    }


    //OutPutCode(cipher1);

    //step 3: reverse
    for(i=0;i<cipher1.size();i++){
        cipher.push_back(cipher1[cipher1.size()-i-1]);
    }
    //step 4:show the final cipher
    cout<<"The Ciphertext is:"<<endl;
    OutPutCode(cipher);

    //Decryption
    vector<int> cipher2;
    string origin;
    for(i=0;i<cipher.size();i++){
        cipher2.push_back(cipher[cipher1.size()-i-1]);
    }
    for(i=0;i<=cipher2.size()-3;i+=3){
        int t=cipher2[i+2];
        cipher2[i+2]=cipher2[i+1];
        cipher2[i+1]=cipher2[i];
        cipher2[i]=t;
    }

    for(i=0;i<cipher2.size();i++){
        cipher2[i]/=f(key,i);
        cipher2[i]-=(key[i]-'0');
        if(cipher2[i]>=0&&cipher2[i]<=9){
            origin+='0'+cipher2[i];
        }
        else if(cipher2[i]>=65&&cipher2[i]<=90){
            origin+='A'+cipher2[i]-65;
        }
        else if(cipher2[i]>=97&&cipher2[i]<=122){
            origin+='a'+cipher2[i]-97;
        }
    }
    cout<<"The result of decryption is:\n";
    cout<<origin<<endl;
    system("pause");
    return 0;
}


int checkplain(string s)
{
    for(int i=0; i<s.length();i++){
        int x,y,z;
        x=(s[i]>='0' && s[i]<='9');
        y=(s[i]>='A' && s[i]<='Z');
        z=(s[i]>='a' && s[i]<='z');
        if(!(x==1||y==1||z==1)){
            return 0;
        }
    }
    return 1;
}

string CreateKey(int n)
{
    string result;
    for(int i=0;i<n;i++){
        if(i%10==0||i%10==6){
            result+='3';
        }
        else if(i%10==1||i%10==4){
            result+='1';
        }
        else if(i%10==2){
            result+='8';
        }
        else if(i%10==3||i%10==5){
            result+='0';
        }
        else if(i%10==7||i%10==8){
            result+='7';
        }
        else if(i%10==9){
            result+='2';
        }
    }
    return result;
}

void OutPutCode(vector<int> s)
{
    for(int i=0;i<s.size();i++){
        if(i==0){
            cout<<s[i];
        }
        else{
            cout<<" "<<s[i];
        }
    }
    cout<<endl;
}

int f(string s,int i)
{
    return 3429-(s[i]-'0');
}
```