

信息安全作业（一）

计算机 1202 张艺瀚
学号:20123852

March 27, 2015

程序使用说明：在 Linux 下当前目录执行可执行文件，按照提示操作即可，具体见运行结果截图。

1 编程实现双轨算法

1.1 代码清单

```
1 int main(int argc, char** argv){  
2     cout << "Input plaintext: ";  
3     string plaintext, ciphertext;  
4     cin >> plaintext;  
5     int t = floor((plaintext.size() - 1) / 2);  
6     for(unsigned int i = 0; i < plaintext.size(); ++i){  
7         if(i <= t){  
8             ciphertext.push_back(plaintext[2 * i]);  
9         }else{  
10            ciphertext.push_back(plaintext[2 * (i - t) - 1]);  
11        }  
12    }  
13    cout << "Ciphertext is: " << ciphertext << endl;  
14    return 0;  
15 }
```

Listing 1: 双轨算法的 C++ 实现

1.2 说明

从键盘读入用户的明文，输出密文。

若同一字符在明文中的下标为 i ，在密文中的下标为 i' ，则有

$$i = \begin{cases} 2i', & i' \leq \lfloor \frac{l-1}{2} \rfloor \\ 2(i' - \lfloor \frac{l-1}{2} \rfloor) - 1, & \text{else} \end{cases}$$

其中 $l = \text{len}(\text{plaintext})$

1.3 运行结果

```
zephyr@ubuntu:~/code/cpp/information_security/1/1.1$ ./1-1
Input plaintext: discreteandsystem
Ciphertext is: dsrtadytmiceensse
```

Figure 1: 双轨算法的运行结果

2 编程实现钥控算法

2.1 代码清单

```
1 template<typename T>
2 void display(const vector<T>& v){
3     for_each(v.begin(), v.end(),
4         [](T t){
5             cout << t << " ";
6         });
7     cout << endl;
8 }
9
10 int main(int argc, char** argv){
11     string plaintext, ciphertext, key;
12     cout << "Input plaintext: ";
13     cin >> plaintext;
14     cout << "Input key: ";
15     cin >> key;
16     int dist = 26;
17     vector<int> v(key.size());
18     for(size_t i = 0; i < v.size(); ++i){
19         v[i] = key[i] - 'a';
```

```
20 }
21 int r = plaintext.size() / key.size();
22 if(r * key.size() != plaintext.size()){
23     ++r;
24 }
25 int n = r * key.size();
26 for(size_t i = 0; i < key.size(); ++i){
27     int m = 26, idx;
28     for(size_t j = 0; j < v.size(); ++j){
29         if(v[j] >= 0 && v[j] < m){
30             idx = j;
31             m = v[j];
32         }
33     }
34     v[idx] = -1;
35     for(auto p = plaintext.begin() + idx;;){
36         int d = p - plaintext.begin();
37         if(d < plaintext.size()){
38             ciphertext.push_back(*p);
39         }else if(d < n){
40             ciphertext.push_back('z');
41         }
42         if(d < n){
43             p += key.size();
44         }else{
45             break;
46         }
47     }
48 }
49 cout << "Ciphertext is: " << ciphertext << endl;
50 return 0;
51 }
```

Listing 2: 钥控算法的 C++ 实现

2.2 说明

从键盘读入用户的明文和密钥，输出密文。

若 $l = \text{len}(\text{key})$ ，将明文写成一个 l 列矩阵，按照 key 中个字符的字典序将每一列写出即为密文。

2.3 运行结果

```
zephyr@ubuntu:~/code/cpp/information_security/1/1.2$ ./1-2
Input plaintext: thenormaldeisionablerepresentationhasfourseparatepartsinaspec
ficformat
Input key: computer
tltriutni
moeafrsia
eebsnspsf
hdaeoreac
nileheapo
anptoaift
rirtsatcm
osenaprer
Ciphertext is: tltriutnimoeafrsiaeebsnspsfhdaeoreacnileheapoanptoaiftirtsatcmo
enaprer
```

Figure 2: 钥控算法的运行结果