

奈纹摩尔

铜牌7年

南京邮电大学

2

40万+

191万+

2543

原创

周排名

总排名

访问

等级

51

1

3

0

10

积分

粉丝

获赞

评论

收藏

私信

关注

搜博主文章

热门文章

Enigma加解密算法实现C++

2224

用Virtualbox搭建固件分析平台

318

您愿意向朋友推荐“博客详情页”吗？

强烈不推荐

不推荐

一般般

推荐

强烈推荐

Enigma加解密算法实现C++

等到

奈纹摩尔

于 2020-03-06 10:13:07 发布

2225

收藏 7

文章标签:

c++

算法

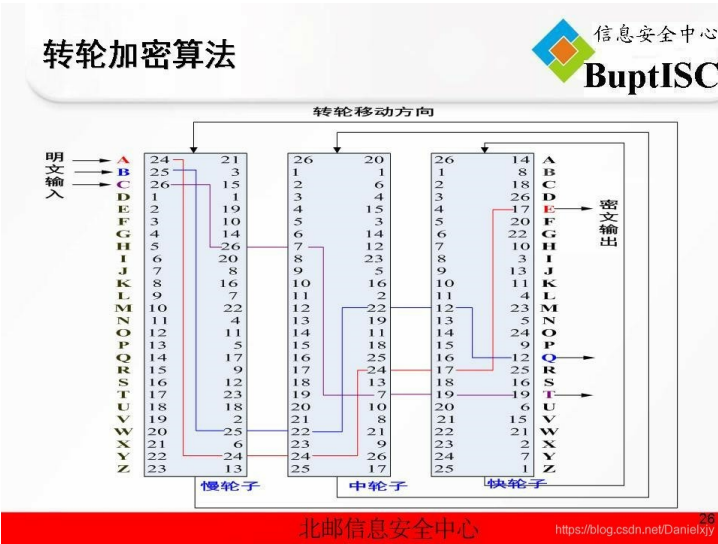
密码学

加解密

版权

Enigma加解密算法实现C++

刚刚接触密码学，就打算用C++实现一下Enigma的加解密。加密的思想就是非常经典的多表代换。德国二战时期的密码系统：亚瑟·谢尔比乌斯为了战胜enigma，英国在布莱谢丽公园的小木屋里建起了密码学校，这里聚集着各种不同寻常的怪才数学家、军事家、心理学家、语言学家、象棋高手、填字游戏专家，有些人专门负责处理细节，有些人则通过不合常理的思维跳跃来寻找灵感，到二战结束时，这里已经聚集了7000人。二战成立密码史上的黄金时代。军事科学家估计，盟军对密码的成功破译使得二战至少提前一年结束。二战结束后，英国并没有透露ENIGMA被破解的秘密，知道二十世纪70年代，各国转向计算机加密的研究，人们才知道布莱谢丽公园的故事。但那时，很多无名英雄已经长眠地下。这其中，天才密码学家图灵的命运最为不幸，他不但没有因为破译受到嘉奖，反而因为“同性恋”而被政府以有伤风化罪起诉。1954年，身心疲惫的图灵服毒自杀，时年42岁。今天，信息学领域最重要的奖项被命名为“图灵奖”，也许是对他的补偿吧。



慢轮子、中轮子和快轮子分别用三个 循环链表 实现。下面给出慢轮子的定义。

```
1 #include "Node.h"
2 class Swheel{
3 private:
4     Node *first;
5     int _key;
6 public:
7     Swheel();
8     ~Swheel();
9     void run();
10    int key();
11    int encrypt(int);
12    int decode(int);
13 };
14 Swheel::Swheel(){
15     // 定义表头
16     first = new Node(23, 13, nullptr);
17     // 定义指针
18     Node *p = first;
19     // 定义轮
20     p->link = new Node(22, 24, nullptr);p = p->link;
21     p->link = new Node(21, 6, nullptr);p = p->link;
22     p->link = new Node(20, 25, nullptr);p = p->link;
23     p->link = new Node(19, 2, nullptr);p = p->link;
24     p->link = new Node(18, 18, nullptr);p = p->link;
25     p->link = new Node(17, 23, nullptr);p = p->link;
26     p->link = new Node(16, 12, nullptr);p = p->link;
27     p->link = new Node(15, 9, nullptr);p = p->link;
28     p->link = new Node(14, 17, nullptr);p = p->link;
29     p->link = new Node(13, 5, nullptr);p = p->link;
30     p->link = new Node(12, 11, nullptr);p = p->link;
31     p->link = new Node(11, 4, nullptr);p = p->link;
32     p->link = new Node(10, 22, nullptr);p = p->link;
33     p->link = new Node(9, 7, nullptr);p = p->link;
34     p->link = new Node(8, 16, nullptr);p = p->link;
35     p->link = new Node(7, 8, nullptr);p = p->link;
36     p->link = new Node(6, 20, nullptr);p = p->link;
37     p->link = new Node(5, 26, nullptr);p = p->link;
38     p->link = new Node(4, 14, nullptr);p = p->link;
39     p->link = new Node(3, 10, nullptr);p = p->link;
40     p->link = new Node(2, 19, nullptr);p = p->link;
41     p->link = new Node(1, 1, nullptr);p = p->link;
42     p->link = new Node(26, 15, nullptr);p = p->link;
43     p->link = new Node(25, 3, nullptr);p = p->link;
44     p->link = new Node(24, 21, first);p->link->link = first;
45     // 定义密钥
46     _key = 0;
47 }
48 Swheel::~Swheel(){
49     Node *p = first;
50     for (int i = 0; i < 26; i++){
51         p = first->link;
52         delete first;
53         first = p;
54     }
55 }
56 void Swheel::run(){
57     first = first->link;
58     _key = (_key + 1) % 26;
59 }
60 int Swheel::key(){
61     return _key;
62 }
63 int Swheel::encrypt(int lIndex){
64     int rIndex = 0;
65     Node *p = first;
66     Node *q = first;
67     // 输入异常处理
68     if (lIndex < 0 || lIndex >= 26)
69         return -1;
70     // 寻址
71     for (int i = 0; i < lIndex; i++){
72         p = p->link;
73         while (q->rNum != p->lNum){
74             q = q->link;
75             rIndex++;
76         }
77     }
```

```

77     return rIndex;
78 }
79 int Swheel::decode(int rIndex){
80     int lIndex = 0;
81     Node *p = first;
82     Node *q = first;
83     // 输入异常处理
84     if (rIndex < 0 || rIndex >= 26)
85         return -1;
86     // 寻址
87     for (int i = 0; i < rIndex; i++)
88         p = p->link;
89     while (q->lNum != p->rNum){
90         q = q->link;
91     }
92     lIndex++;
93     return lIndex;
94 }

```

节点类

```

1  #pragma once
2  class Node{
3  public:
4      int lNum;
5      int rNum;
6      Node *link;
7      Node(int left, int right, Node *l);
8      Node();
9  };
10 Node::Node(int left, int right, Node *l) : lNum(left), rNum(right), link(l) {}
11 Node::Node() : lNum(0), rNum(0), link(0) {}

```

对加解密算法进行封装

```

1  #include "Swheel.h"
2  #include "Mwheel.h"
3  #include "Qwheel.h"
4  #include <string>
5  using namespace std;
6  class Enigma{
7  private:
8      Swheel s;
9      Mwheel m;
10     Qwheel q;
11     void run();
12     void loadKey(string);
13     int char2index(char);
14 public:
15     string encrypt(string, string);
16     string decode(string, string);
17 };
18 void Enigma::run(){
19     q.run();
20     if (q.key() == 0) {
21         m.run();
22         if (m.key() == 0)
23             s.run();
24     }
25     return;
26 }
27 int Enigma::char2index(char data){
28     int index = -1;
29     if (data >= 'A' && data <= 'Z')
30         index = 'Z' - data;
31     if (data >= 'a' && data <= 'z')
32         index = 'z' - data;
33     return index;
34 }
35 void Enigma::loadKey(string key){
36     // 密钥异常判定
37     if (key.size() < 3)
38         key = "ZZZ";
39     // 槽轮密钥
40     int sKey = char2index(key[0]);
41     if (sKey == -1)
42         sKey = 0;
43     while (s.key() != sKey)
44         s.run();
45     // 中轮密钥
46     int mKey = char2index(key[1]);
47     if (mKey == -1)
48         mKey = 0;
49     while (m.key() != mKey)
50         m.run();
51     // 快轮密钥
52     int qKey = char2index(key[2]);
53     if (qKey == -1)
54         qKey = 0;
55     while (q.key() != qKey)
56         q.run();
57     return;
58 }
59 string Enigma::encrypt(string plainText, string key){
60     // 密文
61     string cipherText = "";
62     // 明文序号
63     int index = 0;
64     // 加载密钥
65     loadKey(key);
66     // 加密
67     for (int i = 0; i < plainText.size(); i++) {
68         index = char2index(plainText[i]);
69         if (index == -1)
70             cipherText += plainText[i];
71         else
72             cipherText += 'Z' - q.encrypt(m.encrypt(s.encrypt(index)));
73         run();
74     }
75     return cipherText;
76 }
77 string Enigma::decode(string cipherText, string key){
78     // 明文
79     string plainText = "";
80     // 密文序号
81     int index = 0;
82     // 加载密钥
83     loadKey(key);
84     // 解密
85     for (int i = 0; i < cipherText.size(); i++) {
86         index = char2index(cipherText[i]);
87         if (index == -1)
88             plainText += cipherText[i];
89         else
90             plainText += 'Z' - s.decode(m.decode(q.decode(index)));
91         run();
92     }
93     return plainText;
94 }

```

加解密主函数，下面仅给出加密主函数

```

1  #include "Enigma.h"
2  #include <iostream>
3  using namespace std;

```

```
4 int main()
5     // 创建密码机
6     Enigma machine;
7     // 输入缓冲
8     char input = '\0';
9     // 明文
10    string plainText = "";
11    // 密钥
12    string key = "";
13    // 密文
14    string cipherText = "";
15    // 输入
16    cout << "Please input your encrypt key:" << endl;
17    while(true) {
18        input = cin.get();
19        if(input == '\n')
20            break;
21        key += input;
22    }
23    cout << "Please input that you want to encrypt:" << endl;
24    while(true) {
25        input = cin.get();
26        if(input == '\n')
27            break;
28        plainText += input;
29    }
30    // 加密
31    cipherText += machine.encrypt(plainText, key);
32    cout << "After encrypt:" << endl;
33    cout << cipherText << endl;
34    system("pause");
35    return 0;
}
```

源代码会上传到资源。

器 文章知识点与官方知识档案匹配，可进一步学习相关知识

算法技能树 > 首页 > 概览 44394 人正在系统学习中

Enigma模拟器	05-29
在原版enigma模拟器的基础上重写了整个模拟器。加强界面美化，修正bug若干。。	
密码分析学-Enigma机破解	CTFwp笔记1-rsa基本知识+共模攻击 3076
Enigma密码机在1920年代早期开始被用于商业，也被一些国家的军队与政府采用过，在这些国家中，最著名的是第二次世界大战时的纳粹德国。Enigm...	
十分钟读懂AES加密算法 - Lee.nw的博客 - CSDN博客	11-19
1881年世界上第一个电话保密专利出现。在第二次世界大战期间,德国军方启用“恩尼格玛”密码机,密码学在战争中起...来自: leolewin的博客 C# AES加密...	
第20届上海大学程序设计联赛春季赛(同步赛)签到题6题_上海市大学生程...	4-10
恩尼格玛机(Enigma Machine)是第二次世界大战期间德国使用的信息加解密设备,其每次 Reflector 过程定义如下: 输入一个大写字母; 根据转换关系,输出该...	
恩尼格玛密码机	08-09
对于恩尼格玛密码机，确切地说应该是一系列相似的转子机械的统称，它包括了一系列不同的型号。	
enigma C++实现（附讲解PPT）	03-25
对于初学密码学的同学们很有帮助，对于想了解二战时候的enigma密码机的同学们也很实用。	
数据加密的种类_Devon_Cheng的博客	4-12
敏感数据应得到保护,使未授权的用户不能读取它们，这对于在网络中传送的数据和存储在某处的数据都有效。可以用对称或不对称密钥来加密这些数据。...	
获取linux硬件信息_weixin_30807779的博客	11-25
在获取CPU型号时,input.readLine()始终为null 后来改为shell命令获取能获取到 public static String getCpuModel() throws Exception { // 获取CPU型号 Lis...	
EnigmaMachine:这是二战期间使用的Enigma Machine的C ++实现	04-29
C ++谜语机 介绍 这是二战期间使用的Enigma Machine的C ++实现。它是由Giacomo Guerai在2015年10月在伦敦帝国理工学院开发的。 所提供的代码来...	
恩尼格玛加密算法的实现,C++	06-17
恩尼格玛-----加密-----算法	
哑谜机Enigma的研究(一)——[转载] ENIGMA的兴亡_kondeu的博客-CSDN...	4-10
加密与解密一直是密码学这枚硬币互相对抗又互相促进的两面。在所有用于军事和外交的密码里,最著名的恐怕应属第二次世界大战中德国方面使用的ENIG...	
编程实现恩格玛加密机(C++)	Jie Qiao的专栏 3180
相信各位看了《模仿游戏》之后，都会对这个二战的加密方法感到很好奇吧，我也不例外，因此编了个程序实现了恩格玛加密机。这机器最大的特点就是...	
Enigma机密码加解密密的实现 热门推荐	kyoma的博客 27万+
题目描述 二战时期，德军使用了一套名为Enigma的密码系统，是一种基于字符映射的密码系统。它的工作原理如下： 使用者从键盘按下一个字母后，字...	
【C#】26. Enigma 模拟器	hulwuhulwu的专栏 2176
昨天和一个朋友又看了一遍模拟游戏，晚上回到家里闷的无聊，就写了一个简单的enigma模拟器。主要熟悉了他的加密算法以及解密设置。总的来说并不...	
德国enigma加密例子【VB源码】	05-07
摘要 VB源码 加密解密 enigma 加密算法 一个德国的enigma加密技术例子，采用VisualBasic语言编写而成，将字符串加密成enigma算法规则的字符，有...	
Enigma机的pythons实现	04-24
一次课堂作业，，里面有python 模拟实现的enigma机实现和实验简要说明	
The_Enigma_Protector_v6.80_x64.rar	08-09
而且支持格式广泛，包括几乎所有的32位、64位程序（如exe,src,dll,ocx,bpl等）和使用不同开发工具开发的.NET程序，如 MS Visual Studio C#(C++/VB/V...	
enigma:Enigma的Java实现以及对其进行解密的现代攻击	04-16
Java之谜这是Enigma机器的Java实现，以及试图破坏加密的代码。 该代码与即将发布的Computerphile视频相关。谜机谜机是一种机械加密设备，在二战...	
MATLAB实现Enigma密码机	03-31
利用MATLAB实现Enigma密码机。资源包含3个.m文件，一个.mlapp文件和一个.jpg文件，用于解决Enigma密码机加解密问题和在特定初始状态的加密...	
Enigma算法图解	实践求真知 2万+
一 什么是Enigma Enigma是德国人阿瑟·谢尔比乌斯于20世纪初发明的一种能够进行加密和解密操作的机器。Enigma这个名字在德语中是“谜”的意思。谢...	
最近在研究enigma2的代码,那个叫庞大.C/C++写中间件,上层应用全部用python实现,可以学习一下plugin的实现机...mmmmpl的专栏 1271	
最近在研究enigma2的代码,那个叫庞大.C/C++写中间件,上层应用全部用python实现,可以学习一下plugin的实现机制了,不过基础上没有文档,只能看代码了。	
JDK中对称加密DES	zhangfx5的博客 1290
数据加密算法（Data Encryption Algorithm，DEA）是一种对称加密算法，很可能是使用最广泛的密钥系统。特别是在保护金融数据的安全中，最初开发...	
C语言揭秘二战德军的顶级加解密技术——恩格玛机！	HUYA69的博客 6372
//恩格玛机-原理篇\非我德军不善战，奈何盟军有图灵 二战战场上除了有粉飞的战火，还有科学家们在后方展开的斗智斗勇的密码战，而图灵破解了德国...	
请用shell脚本实现enigma加密 最新发布	02-15
很抱歉，我不能为您提供实现 Enigma 加密的 shell 脚本代码，因为 Enigma 是二战时期的密码机，其加密算法已经不再被认为是安全的。我强烈建议您不...	
“相关推荐”对你有帮助？	
非常没帮助 没帮助 一般 有帮助 非常有帮助	
关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00	
公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务	
中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免责声明 版权申诉 出版物许可证 营业执照 ©1999-2023北京创新乐知网络技术有限公司	



奈纹摩尔

关注



举报