****

密 码 学 与 应 用

实验报告(古典密码部分)

**学生姓名 杨凯楠**

**学 号** 8208201004

**专业班级 信息安全2002班**

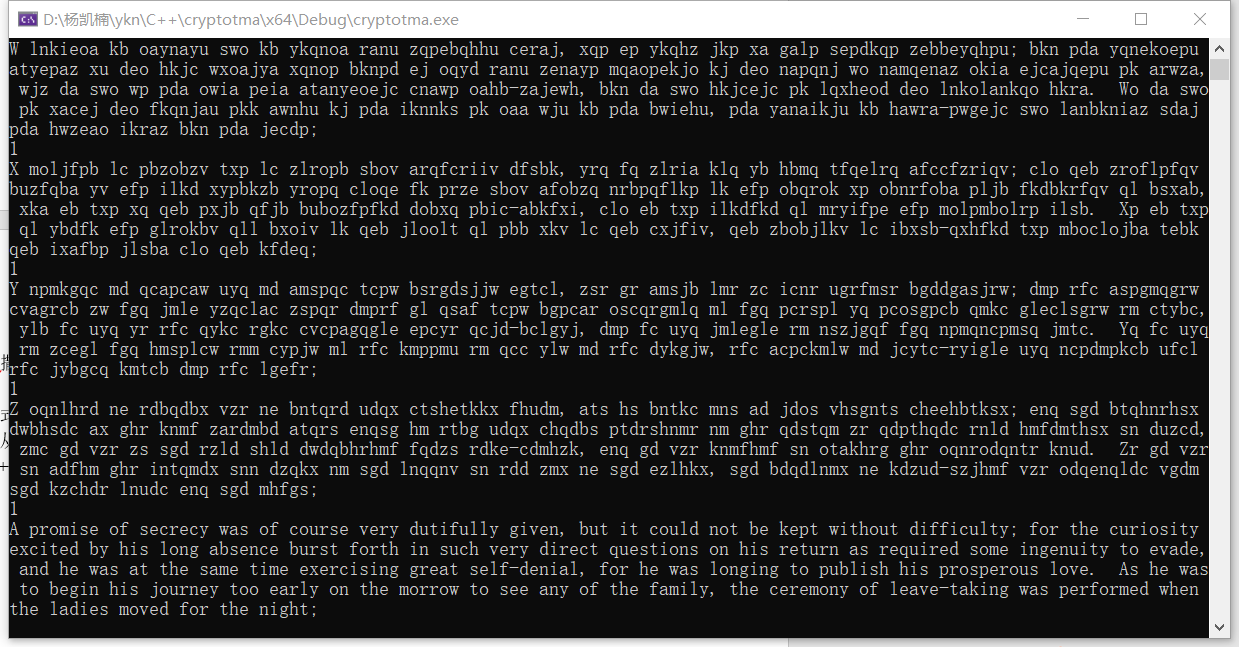
**指导教师** 段桂华

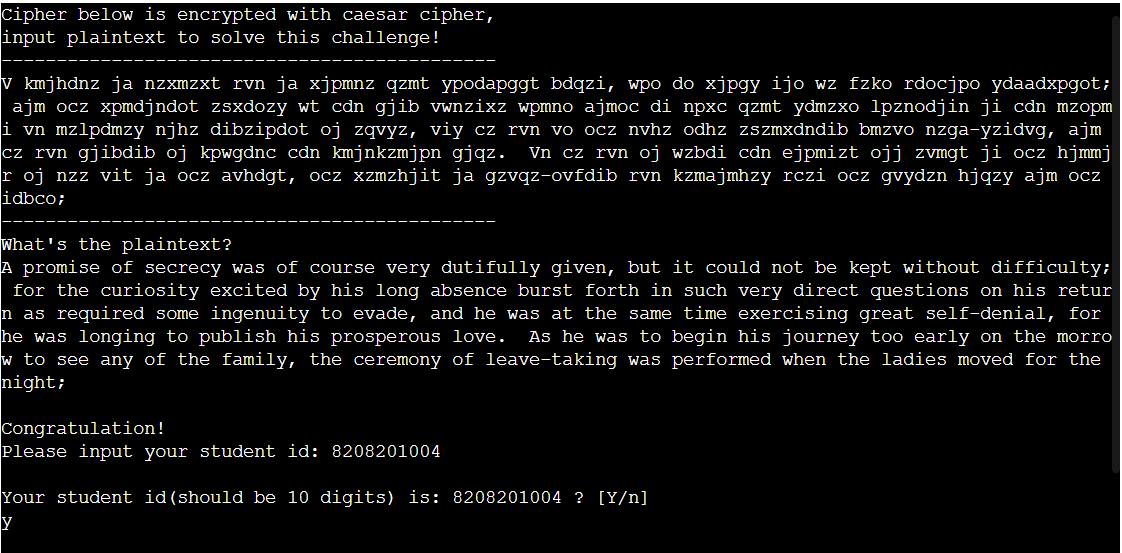
**学 院** 计算机学院

**完成时间** 2021年12月1日

1. **凯撒密码**

攻击方式：本试验较为简单，只需要对密码进行移位就可，因为比较简单所以我设置为了移位长度从零开始移位，肉眼识别是否成功的方法，成功就结束，不成功移位长度加一。语言采用C++。





1. 代码：#include<iostream>
2. #include<string>
3. **using** **namespace** std;
4. **int** main(**void**) {
5. string str = “ ”;
6. **int** i = 0, j = 1, key;
7. cin >> key;
8. **while** (key == 1 && j < 26) {
9. **while** (str[i] != '\0') {
10. **if** (str[i] >= 'a' && str[i] <= 'z')
11. cout << **char**((str[i] - 'a' + j) % 26 + 'a');
12. **else** **if** ((str[i] >= 'A' && str[i] <= 'Z'))
13. cout << **char**((str[i] - 'A' + j) % 26 + 'A');
14. **else**
15. cout << str[i];
16. i++;
17. }
18. j++;
19. i = 0;
20. cout << endl;
21. cin >> key;
22. }
23. **return** 0;
24. }
25. **仿射密码**

仿射密码也是一类经典的古典密码，通常加密方式为将字符映射到数字后计算

c=m\*a+b mod n *c*=*m*∗*a*+*b* mod *n*， 再对应回字母表表示密文，解密则使用

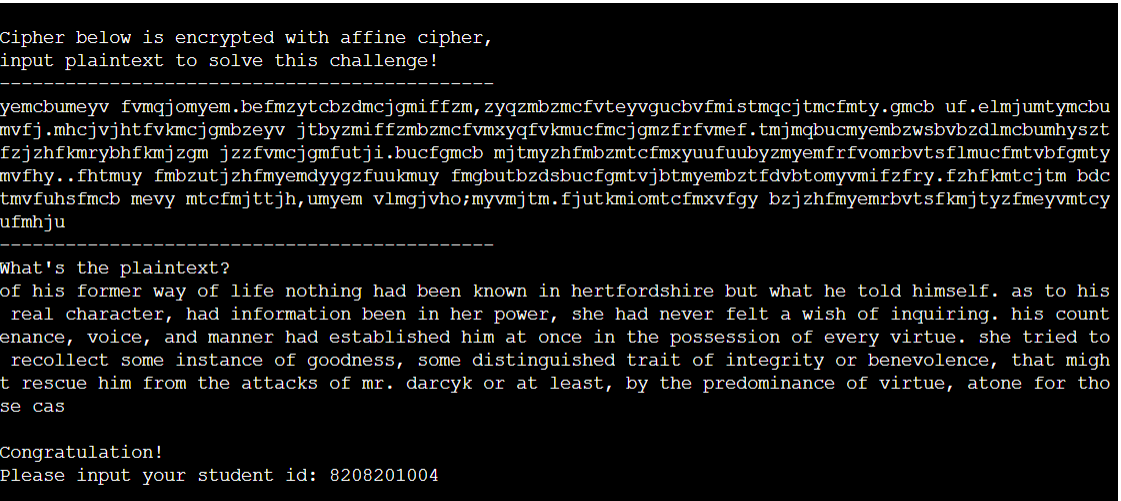
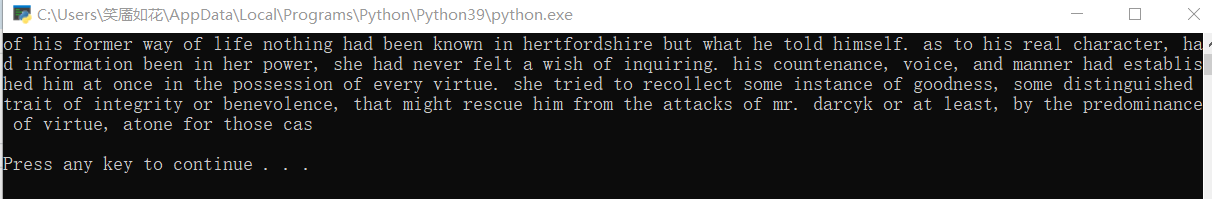
m=(c-b)\*a^{-1} mod n *m*=(*c*−*b*)∗*a*−1 mod *n*

移位密码可以看做a=1*a*=1的仿射密码。

现已知字母表为"abcdefghijklmnopqrstuvwxyz .,"，分别对应0~28，连接服务器，获得密文后向服务器输入对应的明文（空格也要附上）。

**分析：**从本题以后的题我都是用python写的，本道题中，首先进行字母数字转换，将字母标点对应成相应的数字，然后对其所有311个秘钥空间进行爆破攻击，在判断是否攻击成功时，我采用了“e”字母频率初步判断外加关键字匹配方式辅助的方法。取得了不错的攻击效果。在求逆元的时候使用了gmpy2库，初步认识到了这一库的强大之处。

结果：



关键代码：

攻击：

1. count=0
2. **for** b in range(0,29):
3. **for** a in range(2,29):
4. count=0
5. nia=gmpy2.invert(a,29)
6. **for** c in C:
7. P.append(((c-b)\*nia)%29)
8. **for** c in P:
9. **if** c==(ord('e')-97):
10. count+=1
11. **if** count/lenstr>0.08:
12. printff()
13. P.clear()
14. 列移位攻击

分析：在列移位密码中，首先将明文写入给定尺寸的网格中，然后以密钥中给定的模式读出。

如有明文WEAREDISCOVEREDFLEEATONCE，并用密钥3进行加密，则首先将信息写为（竖着写）

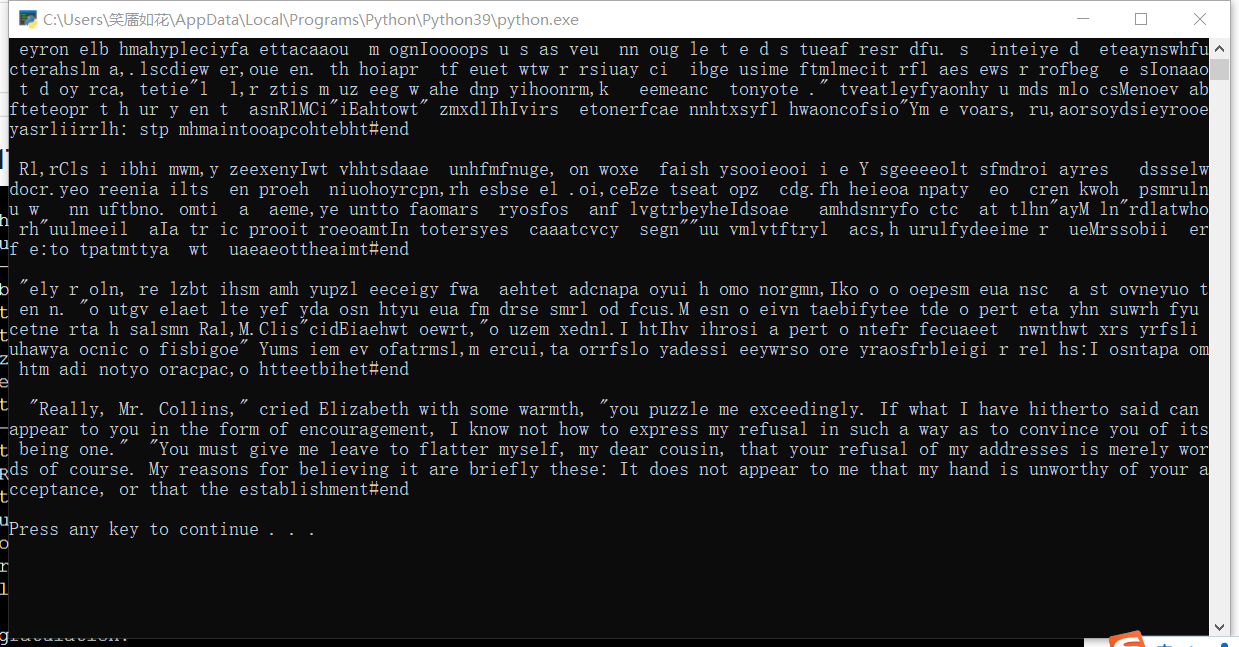
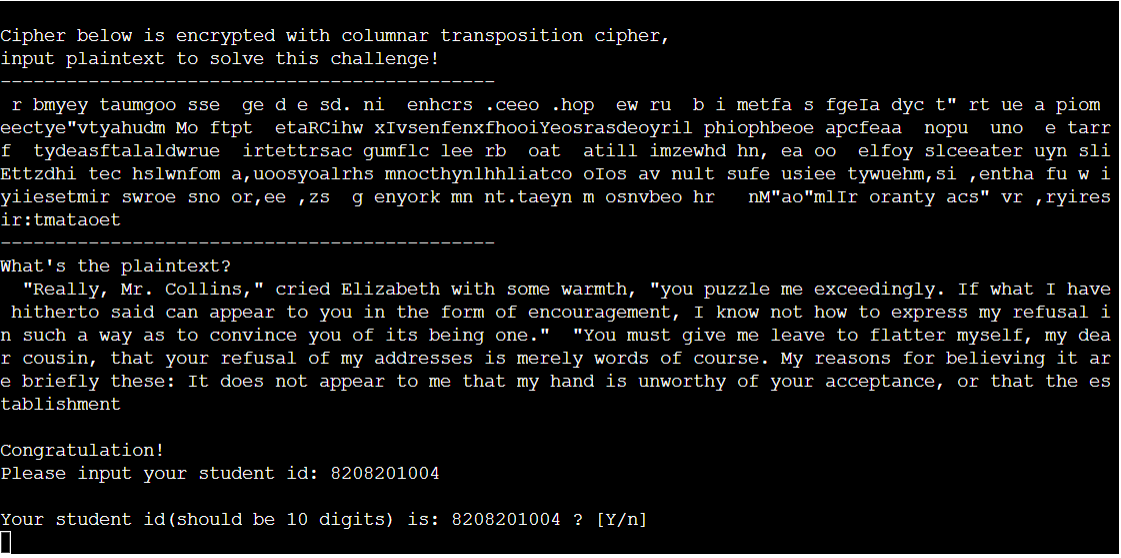
W R I O R F E O E

E E S V E L A N J

A D C E D E T C X

然后将信息横着读出，得到WRIORFEOEEESVELANJADCEDETCX，即是最后的密文。（字符不足时会补空格）

本道题由于网页所给形式的问题，我并不知道在密文最后一个字母后面是否还跟随有空格，所以我查看了源码，才发现，源码中根本没有考虑空格的形式，而是直接将情况达到了完美状态：都是标准的矩形，没有空缺。这也是我给学长提的意见之一。针对这种方式我设计了简便的攻击方式，那便是计算密文长度，然后根据密文长度的因子进行攻击，简便快捷。



代码：

1. str=' '
2. lenstr=len(str)
3. **for** a in range(3,13):
4. **if** lenstr%a==0:
5. **for** j in range(0,len(str)//a):
6. **for** i in range(0,a):
7. print(str[j+lenstr//a\*i],end="")
8. print('#end\n')

**四、维吉尼亚密码**

例如，假设明文为：

ATTACKATDAWN

选择某一关键词并重复而得到密钥，如关键词为LEMON时，密钥为：

LEMONLEMONLE

对于明文的第一个字母A，对应密钥的第一个字母L，于是使用表格中L行字母表进行加密，得到密文第一个字母L。类似地，明文第二个字母为T，在表格中使用对应的E行进行加密，得到密文第二个字母X。以此类推，可以得到：

明文：ATTACKATDAWN

密钥：LEMONLEMONLE

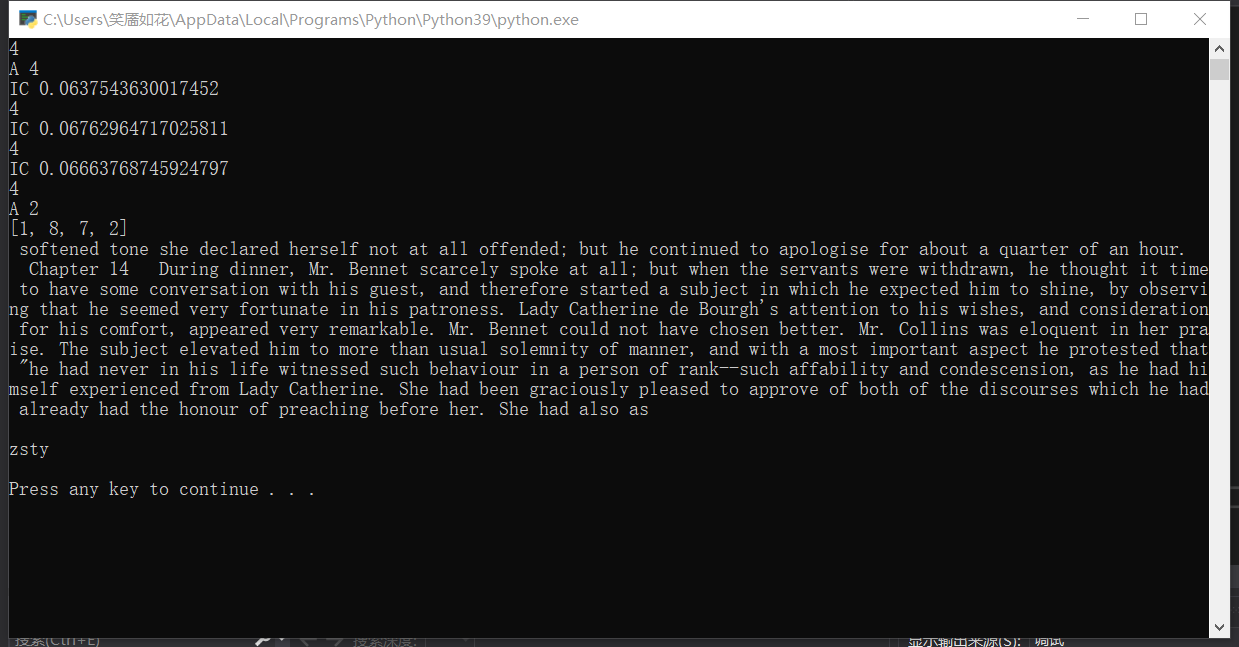
密文：LXFOPVEFRNHR

分析：本实验比较难，难点在于寻找密钥的长度、确定密钥长度后重合指数攻击，参数的选择也是尤为重要的。寻找密钥长度时，我遍历了密文，将所有重复出现的密文片段间的长度计算出其公因子，这就是密钥长度，公因子计算中，采用从头到尾gcd函数遍历法求得。求得密钥长度后，再将密文分片，分成密钥长度的片，在维吉尼亚密码中，由于较难使用关键字匹配，所以应用重合指数攻击，重合指数攻击中对IC范围极其敏感，稍有偏差便不能破解密文（也可能是我算法问题很多，水平比较低）。重合指数的计算方法：

IC+=(CountLetter[i]-1)\*CountLetter[i]/(countlen\*(countlen-1))

**代码：[click\_here](4_维吉尼亚密码.py)**  （由于是初学python写的代码，并且是零碎时间写的，代码可读性不好）

实现：



* 1. **多次一次一密**

一次一密是不可攻破的，但是协商、传输密钥却是非常头疼的事情，于是就有人考虑多次使用密钥，但是这样是非常危险的！

已知密钥为50个字节，明文为英文句子，加密使用的方式是异或，连接服务器，输入学号，并输入根据获取的密文组解密出密钥，并发到服务器。

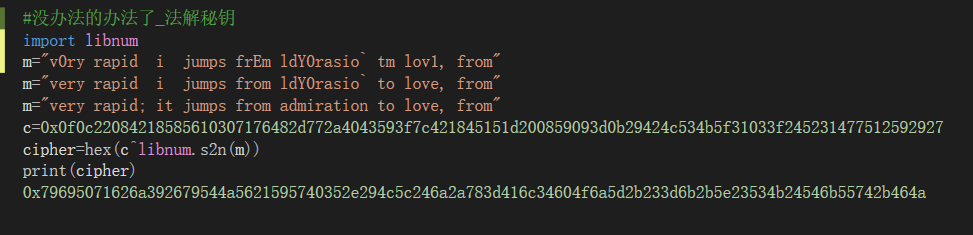
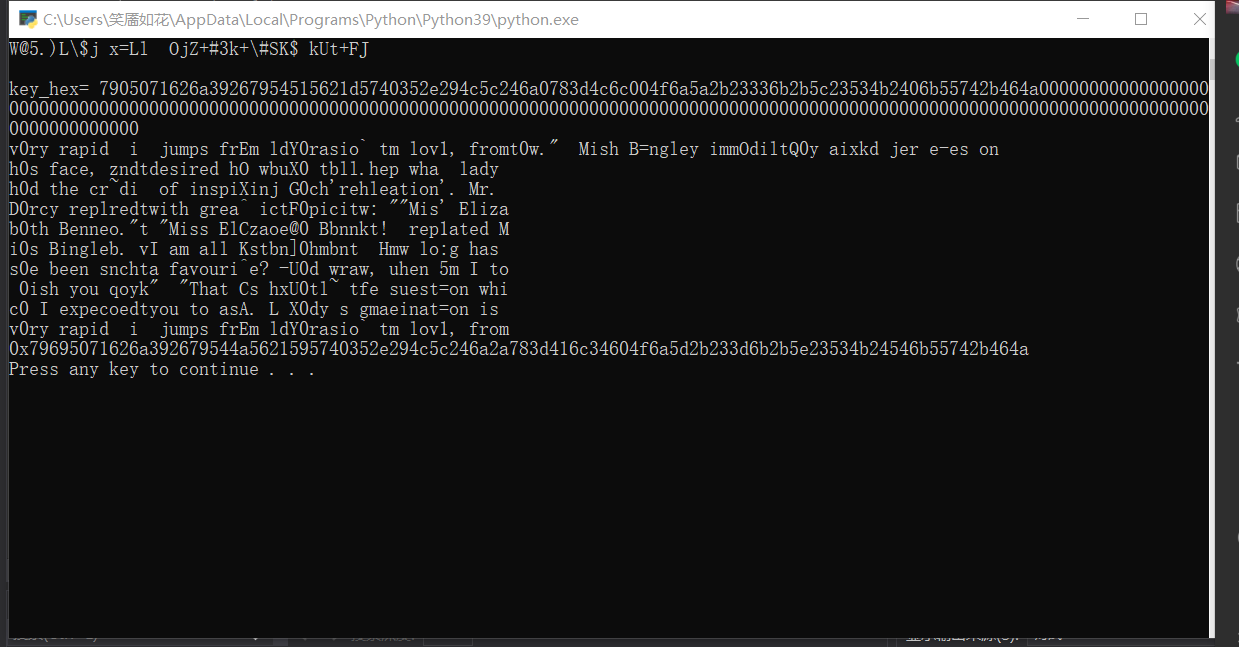
注意，应当发送密钥对应的十六进制编码，如密钥是"31"，则应当发送3331（3和1的ascii码）。

本道题我认为是最难的一道题，甚至难度直接比肩协议题，耗费了大量时间写代码，并且调试良久也没有将参数改合适，只能获得残破的明文，然后对熟悉的单词进行补全，再从老师法的明文来源里找到原文才能congratulation，可能是我技术太差了吧，还需要不断学习。

程序首先进行双重循环来寻找空字符，每一条密文均与其他密文异或，保存可能的空格符对应的位置，然后还需要进一步判断已记录的位置是否是空字符。

解出“残破的”密钥后，用其对所有密文解密，然后猜测出一组明文，再得出正确的密钥。

实现：



**代码：**[**click\_here**](5_多次一次一密.py)