中国科学技术大学

本科生课程实践论文

论文题目：密码学导论课程实践

作者姓名：　梁兆懿

学科专业：　网络空间安全

导师姓名：　李卫海老师，胡红钢老师

完成时间：　2022 年 5 月 10 日

摘　　要

本次密码学导论课程实践的内容（上半学期）分为两部分，分别是单表代换破译工具的实现以及 AES 加密、解密工具的实现。

第一部分是实现单表代换破译工具的实现。笔者通过 python 的初步学习，尝试基于 python 语言实现程序的编写。通过反复的改进与优化，软件实现了读入文件、统计密文字母分布、根据概率分布给出破译建议、通过上下文分析字母代换、接入字典等功能，并且可以通过输入密钥实现解密。在程序的编写中，学习并使用了 python 中 GUI 的 tkinter 库实现简单的交互界面。在本篇论文中，对于该部分内容主要介绍程序实现的流程以及测试过程，以及遇到困难的解决方法和个人思考。

第二部分是通过编程实现 AES 加密、解密工具。AES 算法由快速查表方式加速运算。程序在事先计算好 4 张 256 字的 T 表和 1 张 256 字节的 S 表，从而最复杂的列混淆操作得以快速实现。程序基于 python 语言编写，使用 cv2 库读入文件，并且借助 tkinter 库实现简易的交互界面。在本篇论文中，对于该部分内容主要介绍程序使用的算法、测试过程，以及个人的思考和实验总结。

关键词：　单表代换　　AES 加解密　　python　　tkinter

ABSTRACT

The content of the introduction to cryptography course practice (the first half semester) is divided into two parts, respectively is the realization of single-table substitution decoding tool and the implementation of AES encryption and decryption tool.

The first part is the realization of single-table substitution decoding tool. I through the preliminary study of Python, trying to write programs based on Python language Through repeated improvement and optimization, software realization the read file Distribution statistics ciphertext letters According to the probability distribution of recommendations to decipher letters substitution by context access to dictionaries, and other functions, and can be realized through the input key decryption in the writing of the program, study and using the python GUI tkinter library implements a simple interface In this paper, for this part of the content mainly introduces the procedures to achieve the process and testing process, and encountered difficulties in the solution and personal thinking.

The second part is mainly about programming AES encryption and decryption tool AES algorithm by fast lookup table accelerated operation Compute program in advance four 256 - word T list and one 256 bytes of S list, confusion to the most complex column operation to quickly implement the program based on the python language, USES cv2 library reading files, and with the aid of tkinter library implementation simple interface In this paper, for this part mainly introduces the procedure of the algorithm used Testing process, as well as personal thinking and experiment summary.

Key Words: Single table substitution , AES encryption and decryption , python , tkinter

目 录

致　　谢

本学期的密码学导论课程学习中，在李卫海老师、胡红钢老师以及胡丞聪助教、王思成助教、周潮助教的带领下，我们在密码学导论课上收获颇丰，强化了专业知识的学习。密码学导论课程融合了各方面的知识，富有挑战性但又不失趣味性。李老师先由古典密码引入，在历史角度上引出密码学的重要性，激发了同学们的兴趣，在后续的学习中逐渐深入，由对称密码、非对称密码延伸到公钥密码、数字签名、消息认证等在生活中常常接触到的应用问题，让我们对信息安全问题有了更深入的认识。

除了理论的学习外，李老师也为我们布置了课后作业以及课程实践，结合到信息安全实践课程，让我们对基础知识有了更深的掌握，编程与实践能力也有了进一步的提高。

在此，向李卫海老师、胡红钢老师以及胡丞聪助教、王思成助教、周潮助教表示衷心的感谢。

2022 年 5 月 15 日

第 1 章 基于 python 实现单表破译辅助工具

1.1 程序实现过程

本次实验是基于 python 实现单表破译辅助工具，通过对 python 的学习[1]以及课上对单表代换密码的了解，实现了读入文件、统计密文字母分布、根据概率分布给出破译建议、通过上下文分析字母代换、接入字典等功能，并且可以通过输入密钥实现解密。如图 1-1.1 所示。

实验中使用类存储密文的各项信息，并且通过不同函数实现实验功能。在运行程序时，首先会弹出文件选择窗口，如图 1-1.2 所示。在选择好相应的密文文件后，将会自动读入密文并且启动分析。程序将统计密文中出现的字符（不统计空格和换行符）并且显示密文长度、字符类型总数、包含字符种类以及字符总数。接下来，程序将会统计各个字母出现的频率，并与字典中字母出现频率相比较，得出破译建议。同时，密文将会根据上下文对应统计三个字符的重复出现频率，并与 the 相匹配，给出破译建议。此外，程序可以外接字典—英汉词典以及有道词典来辅助破译。最后，程序可以通过输入对应关系实现破译的尝试。

在本次实验中，由于测试样例部分含有单词间隔而部分不含单词间隔，为了保持通用性，采取了统一按不含单词间隔的形式处理，部分功能如识别单词 I,a 等没有实现。当程序含有单词间隔时，通过明文、密文界面也很快可以观察出部分单词（如 a，on，to）的存在形式，推测出相应字母，从而可以快速通过上下文分析方便得出结果。含有空格的类型将在测试样例中分析。

本次实验使用 tkinter 交互界面，使用了 label ，text，entry，botton 等多种插件，程序界面较为简洁，但是设计有一定局限性且比较繁琐。

1.2 实验算法介绍

本次实验使用的算法主要是通过字符串的处理实现对密文的分析，主要运用的函数有如下：

Openfile 函数：通过 tkinter 库的 filedialog 函数快速实现文件的打开，从而读入密文。

Calculate：遍历密文，通过将未出现过的字符读入列表，实现字符个数、种类的统计，从而得到密文的大体信息。

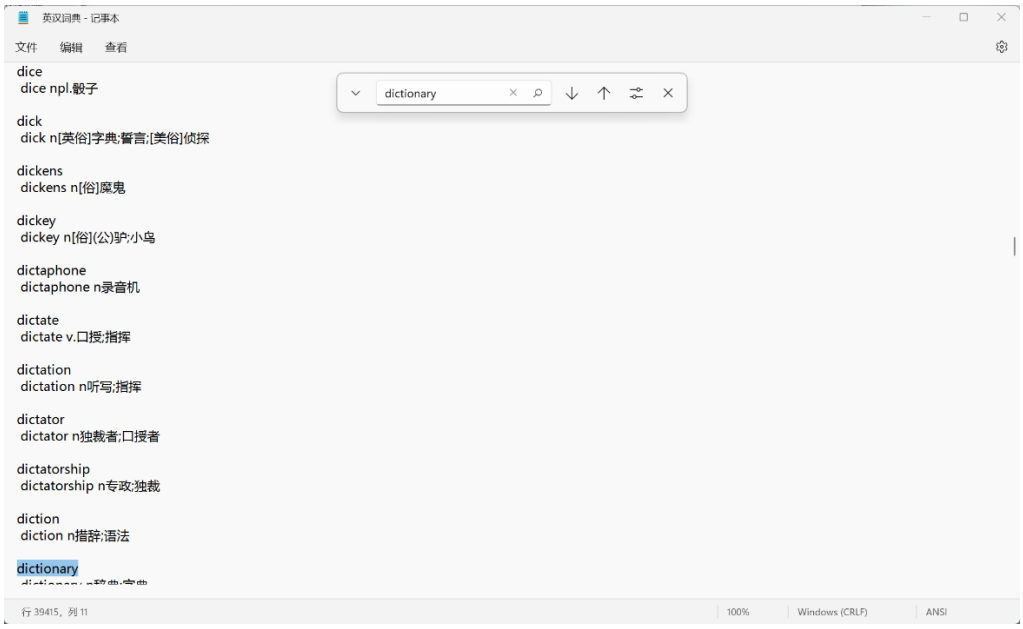
Sort：对各字符的数目进行统计，排序，列入 python 字典中并计算出现概率，从而辅助进行统计概率分析。

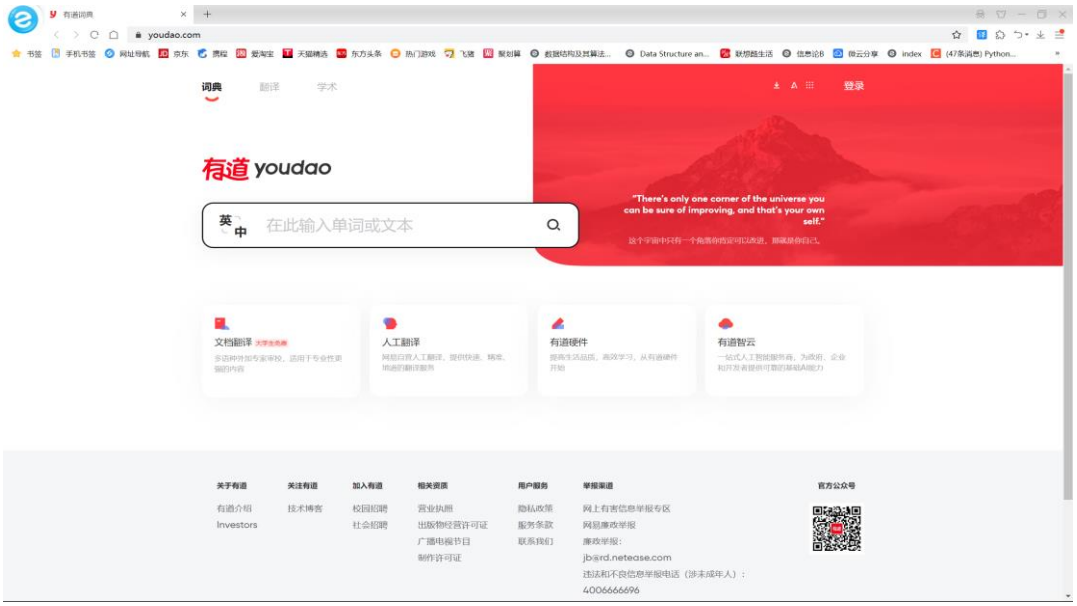
Findthe：顾名思义，通过该函数分析找到 the 对应的字符串。该函数运用多个列表的分析以及切片对重复出现的字符串进行统计，并且字符串最后一个字母为出现最多的字母，即 e，对该类字符串排序。从而获得出现最多的三个连续字符，从而得到 the 对应的字母。

Inputkey 和 decrypt：该函数实现密钥的读入以及解密，需要运用到到多组插件，难点在于交互界面的设计。

Search：通过库函数 os.system 以及 webbrowser.open 打开字典的 txt 文件以及网页，实现辅助破译。此外 Txt 文件查找功能可以快速方便查阅字典，如图所示：

基于 python 实现单表破译辅助工具

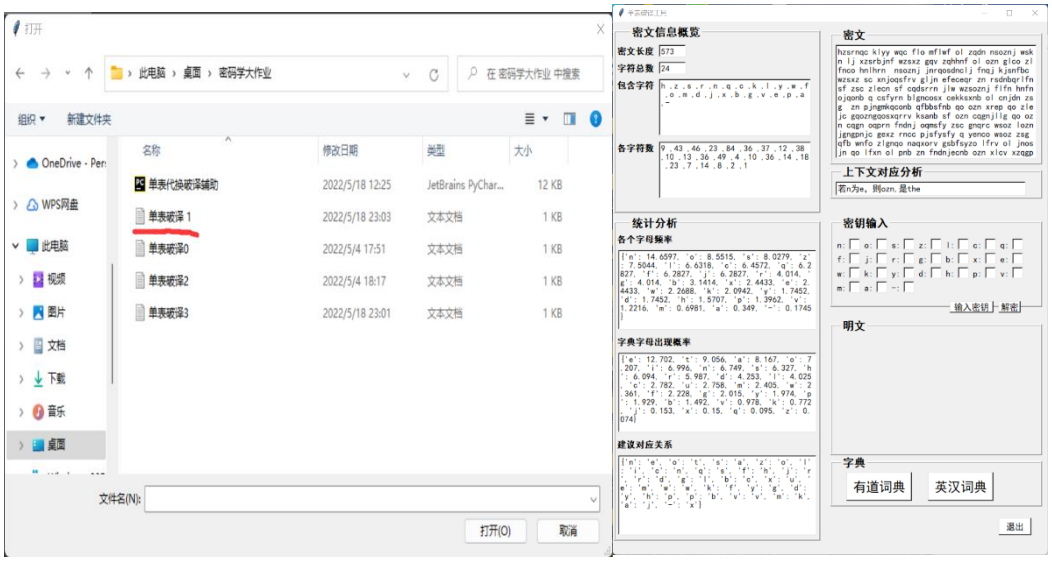




1.3 实验测试以及结果分析

1.3.1 实验测试

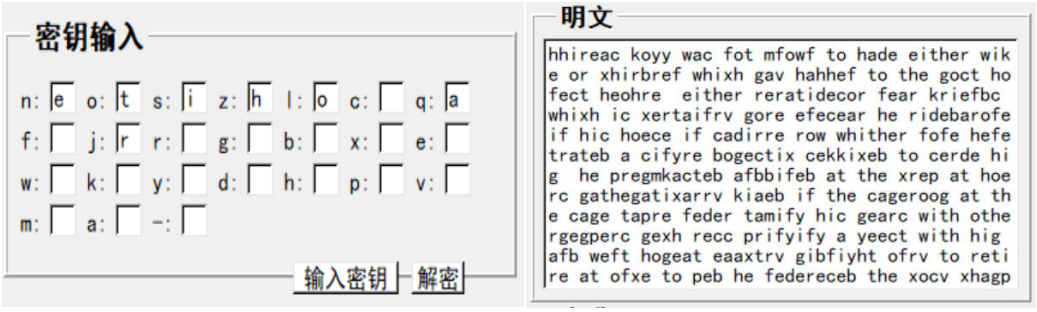
首先，我们运行程序来到打开文件界面，如图 1-3.1 所示：



打开后，程序立即完成分析，我们来到程序界面，如图所示，得知 ozn 对应 the，并且得道破译建议，我们输入 o->t,z->h,n->e。

同时注意到有单独的字母 q 出现，并且不是在句首，因此断定 q 为 a。

对二元词组分析，如 ol，lj，sf，zn，结合对应概率表，得出对应关系，带入，如图所示。



通过部分单词，如 hade，whixh 等，得到更多的对应关系 d->v，x->c。并且字母概率可以分类为：

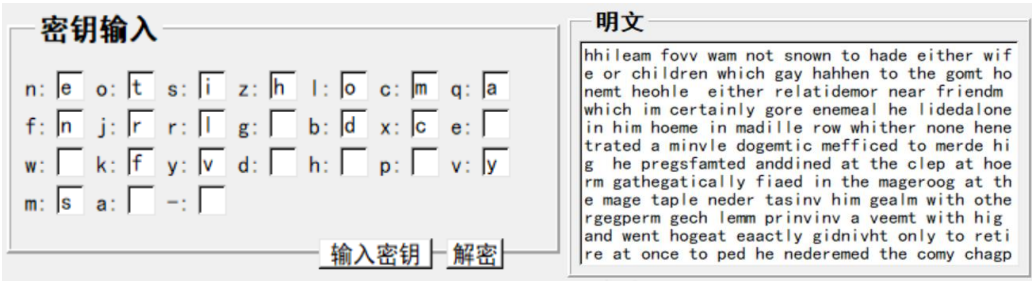
高频率组：e,t,a,o,n,I,r,s,h.(>=0.0528)

中频率组：d,l,u,c,m.

低频率组：p,f,y,w,g,b,v

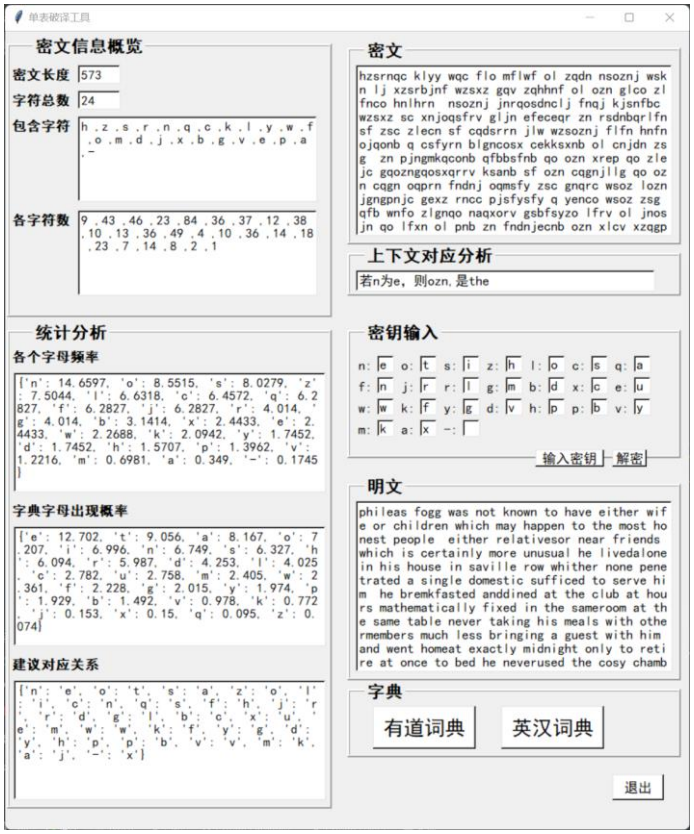
极低组：j,k,q,x,z

从而简化判断，增加判断依据。



之后我们可以通过查阅词典，得到许多单词形式，从而推出结果。（单词 whither->whether 过于类似，确实具有迷惑性）

最终结果：



明文为：

phileas fogg was not known to have either wife or children which may happen to the most honest people either relativesor near friends which is certainly more unusual he livedalone in his house in saville row whither none penetrated a single domestic sufficed to serve him he bremkfasted anddined at the club at hours mathematically fixed in

the sameroom at the same table never taking his meals with othermembers much less bringing a guest with him and went homeat exactly midnight only to retire at once to bed he neverused the cosy chambers which the reform provides for itsfavoured memb ers he passed ten hours out of the twenty-

fourin saville row either in sleeping or making his toilet

通过同样方法，我们可以实现作业一的测试样例（下文记为样例二）以及额外测试样例（样例三）（如图所示）的破译，如图所示：

样例二明文为：A good glass in the bishop's hostel in the devil's seat twenty-one degrees and thirteen minutes northeast and by north main branchseventh limb east side shoot from the left eye of the death's- head abee line from the tree through the shot fifty feet out.

样例三明文为：Success in dealing with unknown ciphers is measured by these four things in the order named,perseverance,careful methods of analysis,intuition,luck. The ability at least to read the the language of the original text is very desirable but not essential. Such is the opening sentence of Parker Hitt’s Manual for the solution of Military Ciphers.

值得注意的是，由于测试样例的多样性（作业一），有些样例以标点符号作为密文。因此，读入密文含标点符号时，标点符号也会进行频率分析，对概率统

计产生影响。因此使用该程序应格外注意含标点符号的频率统计，或者直接删去密文标点符号。

1.3.2 结果分析

在实验测试中，程序运行良好，可以正确给出破译建议。实验样例一分析较快，十分钟左右即可完成。而在样例二、三中，测试时间远远大于样例一，达到半个小时以上。归根结底，在有单词分隔时，上下文的分析较为容易，实现攻击比较简单。而没有单词分隔时，the 的寻找以及频率分布攻击成为首选，此时需要对不同密钥多次尝试得出结果。

此外，部分高频二元词组也应得到重视，尤其在有单词间隔的密文中。如 th,he,in,er,an,re,ed,on,es,st,en,at,to,nt,ha,nd,ou,a,ng,as,or,ti,is,et,it,ar 等词组。尽管二元词组出现概率的偶然性比较高，频率分析困难，主要通过上下文分析得出其对应关系。由于大多数二元词组含有元音字母，因此分析二元词组其对元音字母的分析至关重要（a、i、o），在元音字母对应关系得到后，密文单词的结构将会更加清晰。

1.4 实验总结

本次实验原本设想通过 c 语言实现破译工具，然而 c 语言的字符串处理十分繁琐以及实现代码过于冗长，加之交互界面不易设置，因此在历经波折后，我选择了学习 python 来实现软件的编写。期间一并学习了 tkinter 交互界面的使用，虽然花费了时间以及精力，结果可谓收获颇丰。

在编写代码时，有许多细小的 bug 需要花费大量时间修改。例如，在第一次写好程序时，解密过程通过 replace 函数修改密文，将密文中某一类字符直接替换。看似合理，然而替换必然有先后顺序，在解密 h->s，s->r 后，最后对应关系变成 h->r，显然不正确。需要在已经修改的字符上添加标记避免此类错

误。最后程序采取了 for 循环逐字修改密文，如图所示（#后为错误代码）

```
def decrypt(self):
    t11.place_forget()
    message1 = list(self.cip)
    for i in range(len(message1)):
        for j in self.s:
            if message1[i] == j:
                if(self.key[j]!=''):
                    message1[i] = self.key[j]
                    break
    #for i in range(len(self.key)):
        #if(self.key[self.s[i]]!=''):
            #message1 = message1.replace(self.s[i], self.key[self.s[i]])
```

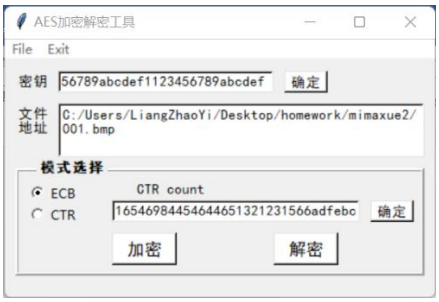
Python 实现代码时，字符串的操作相比 c 语言比较简介，在对密文分析时更加容易实现。

第 2 章 基于 python 实现 AES 加密解密工具

2.1 程序实现过程

本次实验是编写程序实现 AES 加密解密工具，通过上次实验积累下来的经验，笔者对 python 的使用有了一定的了解，因此在此实验中，将继续使用 python 编写程序。

AES 加密解密工具的实现主要由 AES 算法实现，通过 tkinter 实现交互界面，并且用 cv2 读入图片文件实现加密。AES 算法由多轮加密实现，包括字节代换、行移位、列混淆、轮密钥，加密则由相应的逆变换实现。密钥的输入、初始向量的的输入、文件的选择以及 ECB、CTR 模式的选择均可以在交互界面实现。如图所示。



实验使用 AES 类存储信息与函数，相较上一次实验，编写代码时有了更为丰富的经验，代码中交互界面的设计与 AES 算法函数分开实现，使程序代码的层次感更分明，更加简单易懂。

打开文件后，点击 file->open 选择文件，然后输入密钥，CTR 模式则需要输入 count，再进行加密解密。

当加密、解密成功时，分别会弹出窗口，表示加密、解密完成，如图所示。

2.2 实验算法介绍

本次实验主要的算法为 AES 加密解密算法构成。基本算法包括轮密钥的生成，字节代换，行移位，列混淆等以及其逆变换。实现方法分为 EBC 模式和 CTR 模式，主要运用到的相关函数如下(i+函数名为逆变换的函数)：

Changebytes:字节代换函数，借助 Sbox 实现字节代换操作，运用到移位和 bytes 类型与操作。

Moverow：行移位函数，通过 S 表查表快速实现。

Colchange：列混淆函数，可以借助查表快速实现向量移位，每列需要 4 次查表和 4 次异或操作。

Keysub:对 4 个字节实现轮密钥相加操作

Roundkey:轮密钥扩展函数，借助轮密钥相加生成 128 位密钥轮密钥。

e_EBC1: 借助上述加密函数使用 EBC 模式加密的函数，加密到 AESencry.bmp 文件中。

e_CTR1: 借助上述函数实现加密的并且需要读入计数器，从而实现 CTR 加密的函数，加密到 AESencry.bmp 文件中。

d_EBC1: EBC 解密函数，即用到 EBC 的逆变换，实现文件的解密，解密到 AESdecry.bmp 文件中。

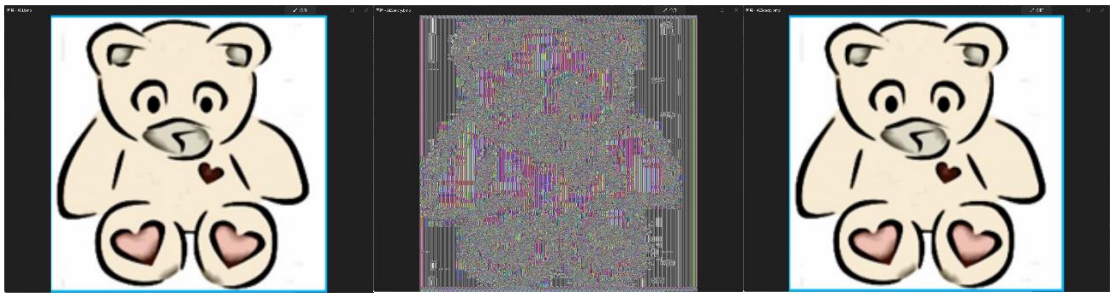
e_CTR1: CRT 解密函数，在读取计数器和密钥后，实现 CTR 解密到 AESdecry.bmp 文件中。

2.3 实验测试以及结果分析

2.3.1 实验测试

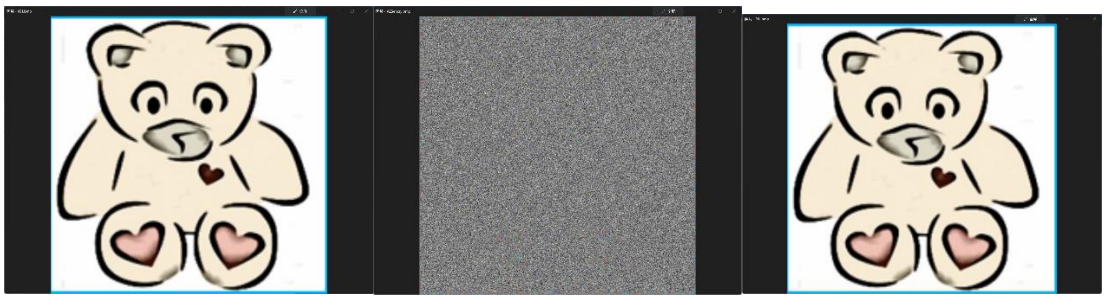
对实验需要图片进行测试，根据要求第一个测试样例为 2048×2048 分辨率的小熊图片，如图所示，选择 ECB 模式，输入 key，实现结果如下：

以下依次为原图，加密图片，解密图片：



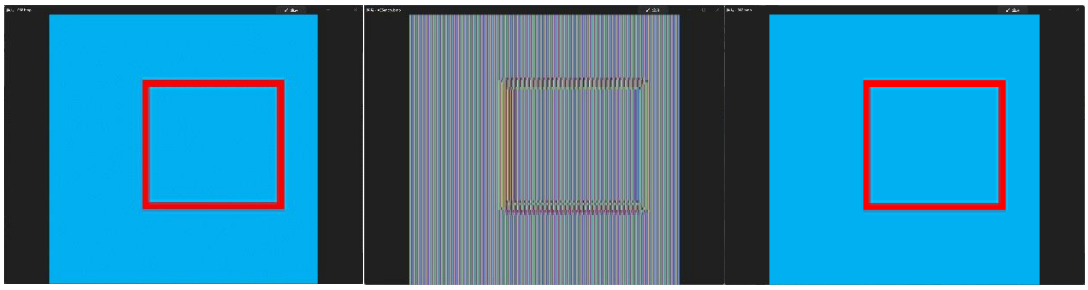
选择 CTR 模式，输入 key 以及计数器 count，实现结果如下：

以下依次为原图，加密图片，解密图片：

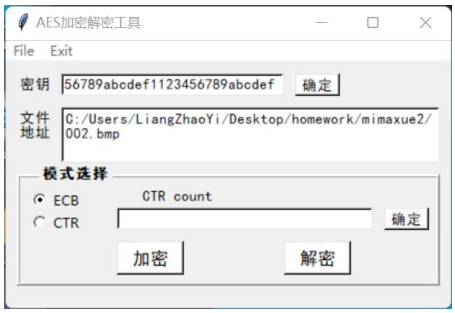


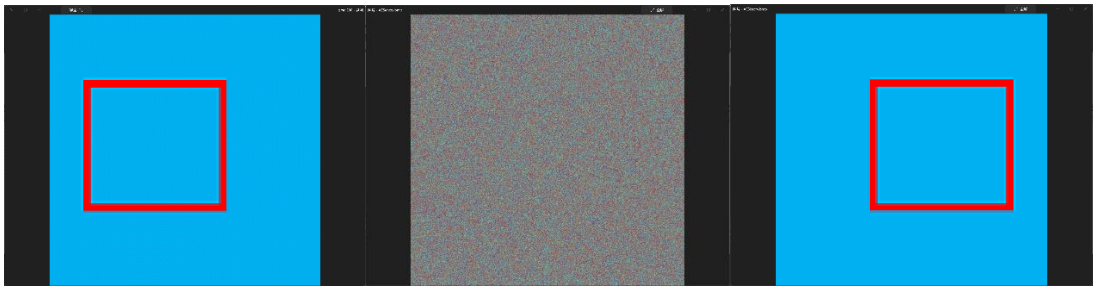
第二个测试样例为 2048×2048 分辨率的正方形与方框图片，如图所示，选择 ECB

模式，输入 key，实现结果如下：



选择 CTR 模式，输入 key 以及计数器 count，实现结果如下：





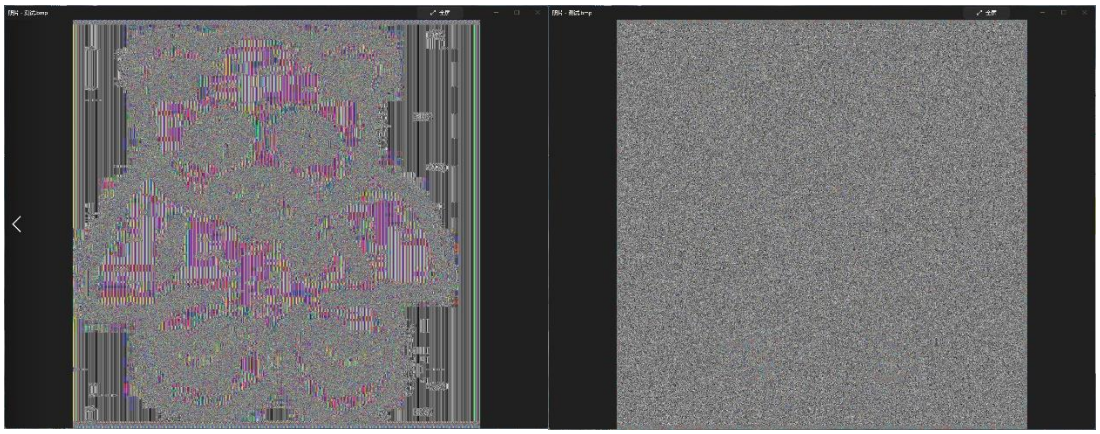
实验验证：

使用 python 的 Crypto 库进行验证，在网络上下载已知可以正确运行 AES 加密解密的代码[2]，对代码进行修改，设定密钥如图所示，并且添加 CTR 模式：

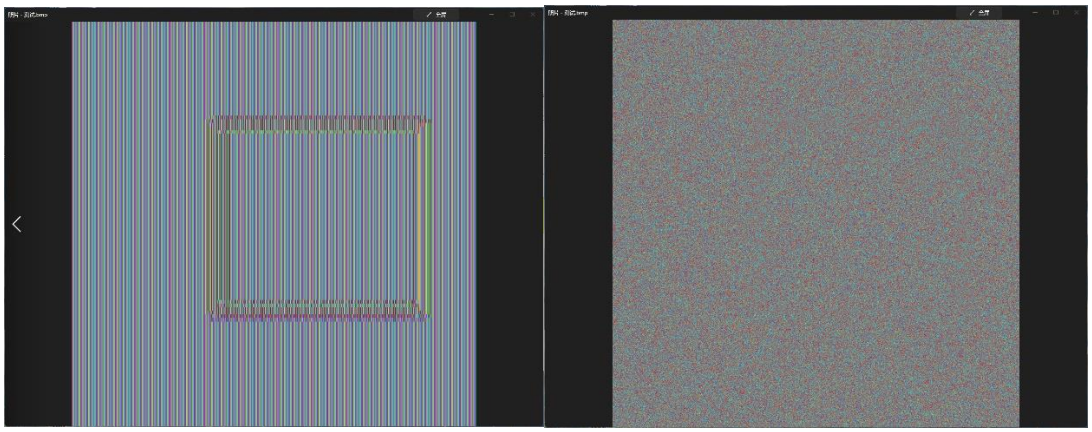
```
# 显示原始图像
cv2.imshow("Original image", imageOrig)
# 原始图像持续显示时间
cv2.waitKey()

# 将图像转化成字节
imageOrigBytes = imageOrig.tobytes()
print("imageOrigBytes:"+str(len(imageOrigBytes)))

# 加密
# 随机生成密钥key和初始向量IV
key0=0x1123456789abcdef1123456789abcdef
count=0x165469844546446513212315664adfebc
key = key0.to_bytes(16, byteorder='big')
count = count.to_bytes(16, byteorder='big')
#初始化AES加密器
cipher = AES.new(key, AES.MODE_CTR, count) if mode == AES.MODE_CTR else AES.new(key, AES.MODE_ECB)
# 将字节数据进行填充，得到填充后的数据
imageOrigBytesPadded = pad(imageOrigBytes, AES.block_size)
# 得到密文
ciphertext = cipher.encrypt(imageOrigBytesPadded)
```

对上述图片分别进行加密、加密，得出结果如下：

样例 1



样例 2

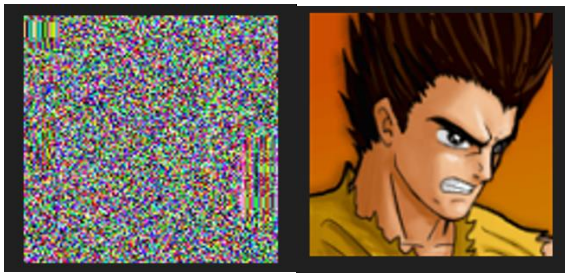


与实验文件有细微差别，结果可能由图片部分像素的填充方式不一样导致。

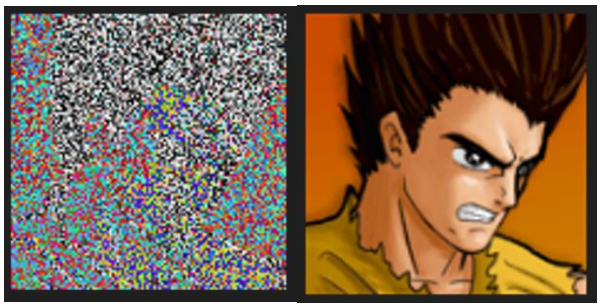
其他样例测试[3]：

测试图像：

ECB 模式：



CTR 模式：



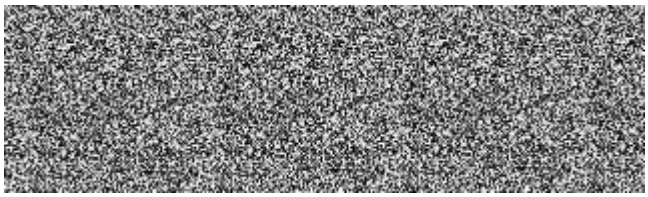
由此可见，文件的分辨率大小也与加密的结果有关。在前文实验中表现较好的 CTR 算法在加密较小文件中加密图片轮廓清晰，加密效果不尽如人意。

2.3.2 结果分析

在样例测试过程中，程序运行的时间较长，是由于加密图片较大，为 2048*2048 像素导致，加密时也需要占用较多内存。而运用 python 的 Crypto 库进行加密则加密时间大大减少。在加密后，2048*2048 像素使用 CTR 模式显示的图像比 ECB 模式更加隐蔽，说明 ECB 模式在加密时，由于是格式化加密，明文重复时，密文也会重复，因此加密的图像仍然可以看见部分轮廓。而 CTR 模式的加密则比较隐蔽，从图像上直接看则比较困难，查阅资料[4]也可得知，ECB 模式加密图片也有一定缺点，与 CTR 或者 CBC 模式相比还存在不足。如图所示：

EBC 加密结果：

CTR 加密结果：



CBC 加密结果：

2.4 实验总结

本次实验选择了 python 作为编程语言，由于在实验一对于 python 以及有了一定的使用经验，因此在实验二的主要困难在于算法的实验，在实验中对 bytes 类型的操作以及位运算需要一定的学习。AES 算法的实现也比较容易出错，需要多次的检查以及 debug 实现。此外，cv2 库的操作在文件路径上不能含有汉字，并且 os.system 库函数打开文件也不能含有空格，在编写程序没有注意则容易报错（这两个 bug 在编程时花费了大量时间解决）。

在这一次实验中，我对 AES 加密解密算法有了重温与进一步学习，也发现了之前对密码学学习的不足之处，在程序刚刚开始编写时尤其吃力。经过学习、编程与思考后，自己的自学能力得到加强，编程能力进一步提高，在课程实践中受益匪浅。

参考文献

[1] 论文中有关 python 的知识来源于《 PYTHON 编程，从入门到实践》, Eric Matthes 编著，袁国忠 译；

[2] 代码源于《基于 AES 的图像加密》，CSDN，所有恐龙都怕霸王龙 编著；

[3] 测试图像来源于网络；

[4]《A novel chaos-based bit-level permutation scheme for digital image encryption》Chong fu 编著