1. **任务目标**
   1. 随机生成密钥，并去除弱密钥和半弱密钥，并将密钥以BASE64编码方式存储到文件
   2. 通过中国剩余定理分配密钥：将原始密钥分成三份，通过其中两份就可以得到原始密钥
   3. 通过DES对指定文件进行加密和解密
   4. 实现基本的交互页面
2. **分工说明**

框架设计及报告撰写：陈婉芊

程序编写：陈婉芊

PPT展示：陈婉芊

1. **模块设计说明** 
   1. **密钥**
      1. **随机生成密钥**

key = ''.join(random.sample(string.ascii\_letters + string.digits, 8)).encode() # key为8字节长度密钥

* + 1. **去除弱密钥和半弱密钥**

# 去除弱密钥和半弱密钥

while (key in weak\_key) or (key in halfweak\_key):

key = ''.join(random.sample(string.ascii\_letters + string.digits, 8)).encode()

* + 1. **以BASE64编码方式存储**

encode\_64 = base64.encodebytes(key)

with open('key\\key.txt', 'wb') as f:

f.write(encode\_64)

f.close()

* 1. **中国剩余定理**

中国剩余定理：设是两两互素的正整数，设是整数，则同余方程组，模有惟一解 ,其中,.

* + 1. **求最大公因数**

remain = b % c

while remain != 0:

b = c

c = remain

remain = b % c

return c

* + 1. **求满足1 = x \* b + y \* c 的系数x，y**

def Get\_Coefficient(b, c):

if c == 0:

return 1, 0

else:

q = b // c

remainder = b % c

x\_t, y\_t = Get\_Coefficient(c, remainder)

x, y = y\_t, x\_t - q \* y\_t

return x, y

* + 1. **求逆**

m0 = y % b

* + 1. **带入中国剩余定理公式**

M = 1

for m in m\_all:

M \*= m

x = 0

for i in range(r):

Mi = M / m\_all[i]

yi = Get\_Inverse(m\_all[i], Mi)

x\_t = (a\_all[i] \* Mi \* yi)

x += x\_t

x %= M

* + 1. **分配密钥**

key\_int = int.from\_bytes(key, byteorder='little', signed=True)

prime\_3 = Prime(key\_int)[::-1] # 调整为从大到小

while prime\_3[0] \* prime\_3[1] <= key\_int:

prime\_3 = Prime(key)[::-1] # 判断是否满足可以使用中国剩余定理的条件

# a ≡ key mod prime

a = []

for prime in prime\_3:

a.append(int(key\_int % prime))

# 以(a, prime) 的形式存储

for i in range(0, len(prime\_3)):

file\_name = 'key\\key\_remain%d\_%d.txt' % (k, i)

with open(file\_name, 'wb') as f:

s = str(a[i]) + ',' + str(prime\_3[i])

f.write(s.encode())

f.close()

k += 1

* 1. **DES加密**

des = DES.new(key, DES.MODE\_ECB) # 生成DES对象

encrypto\_text = des.encrypt(pad(file\_ts, 32)) # 被加密的数据需要为8字节的倍数

# 写入文件

with open(filepath\_save, 'wb') as f:

f.write(encrypto\_text)

f.close()

* 1. **DES解密**

des = DES.new(key, DES.MODE\_ECB) # 生成DES对象

decrrpto\_text\_t = des.decrypt(encrypto\_text)

decrrpto\_text = decrrpto\_text\_t[:length] # 去除之前补的字节

# 写入文件

with open(filepath\_save, 'wb') as f:

f.write(decrrpto\_text)

f.close()

* 1. **交互**

（以初始交互页面为例）

# 使用的是easygui库

while 1:

msg = '是否进入（继续）加解密程序?'

title = '请选择'

if g.ccbox(msg, title): # ok,重新加密

msg = "请选择所需功能，初次使用请先选择生成key！！\n\

1.生成key：随机生成密钥（非弱密钥、半弱密钥)\n\

2.中国剩余定理：通过中国剩余定理分配密钥\n\

（分成3份，只要有其中两份就可解开）\n\

3.DES加密：通过DES加密\n\

4.DES解密：通过DES解密"

title = "本次加解密使用的是DES！" # 标题

choices = ['生成key', '中国剩余定理', 'DES加密', 'DES解密'] # 选项

choice = g.buttonbox(msg, title, choices)

# 判断用户选择

if str(choice) == '生成key':

DesKey()

elif str(choice) == '中国剩余定理':

Remain\_GetKey()

elif str(choice) == 'DES加密':

DesEncode()

elif str(choice) == 'DES解密':

DesDecode()

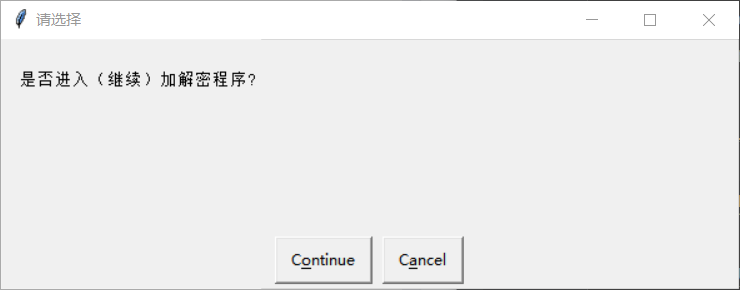
else:

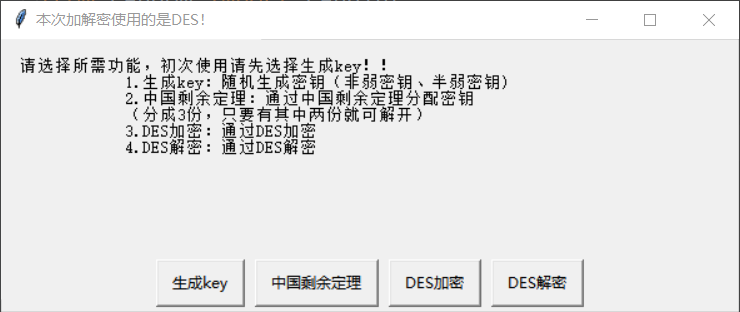
break # cancel,退出程序

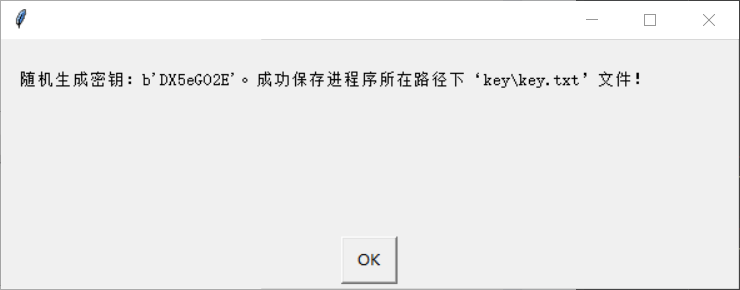
exit(0) # 退出

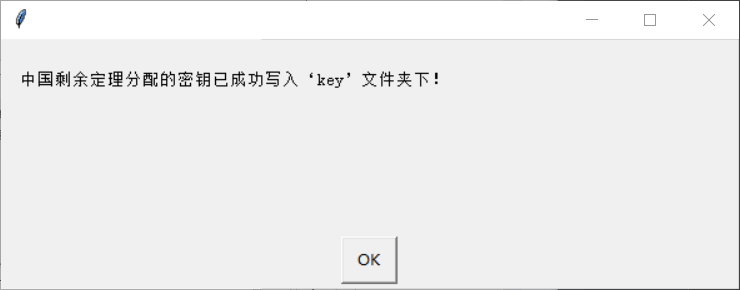
1. **测试情况及最终成果说明**

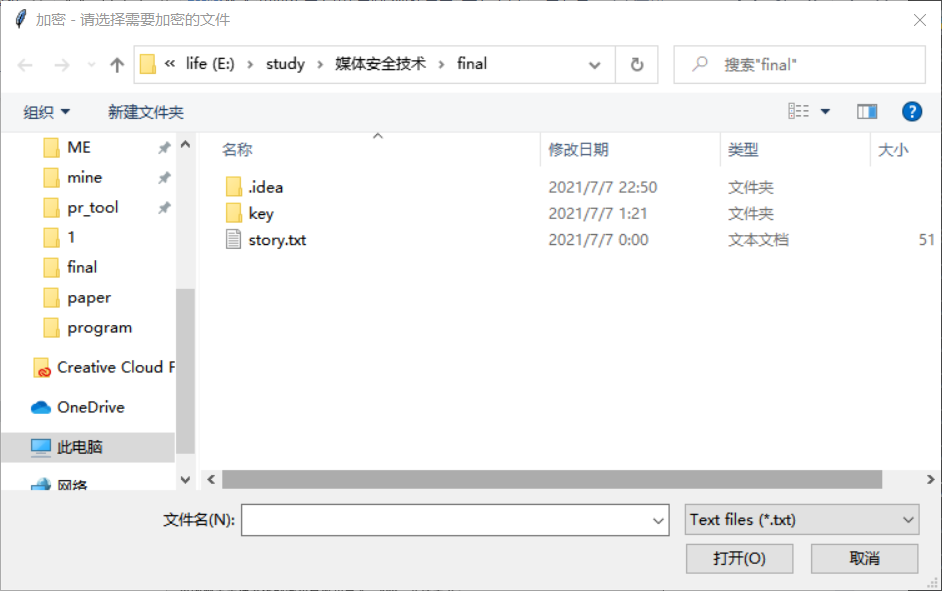
**4.1.运行过程**

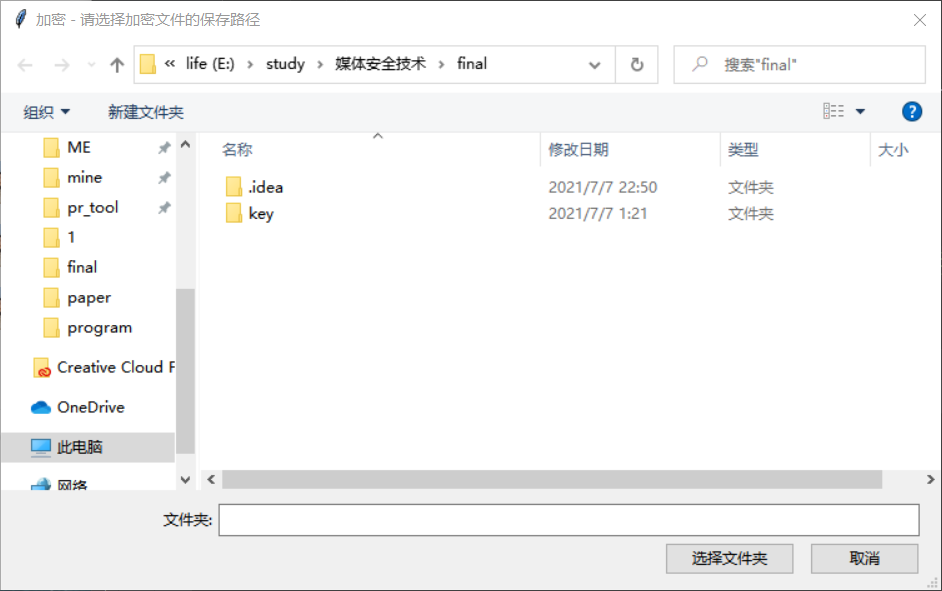


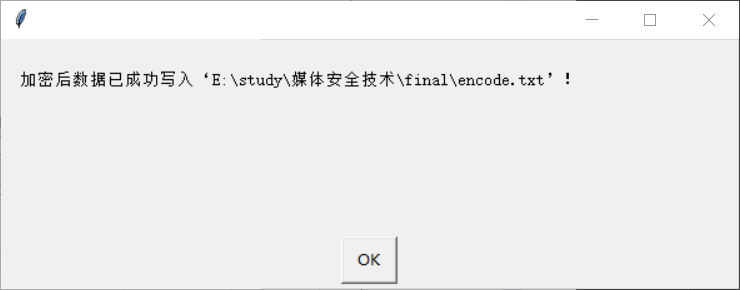


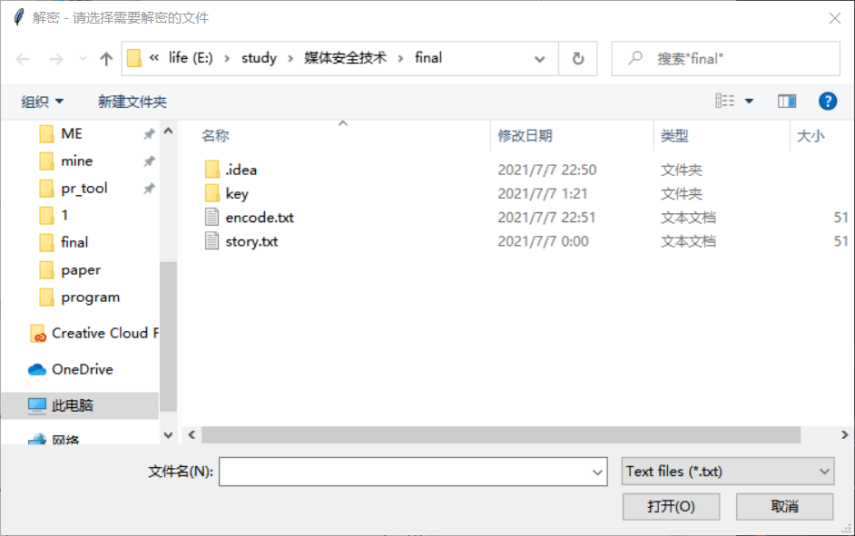


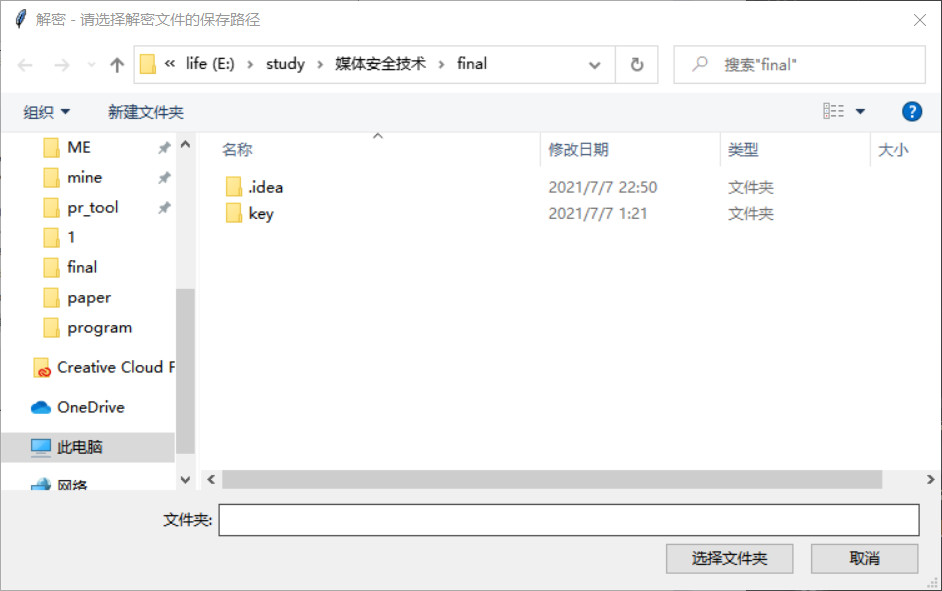


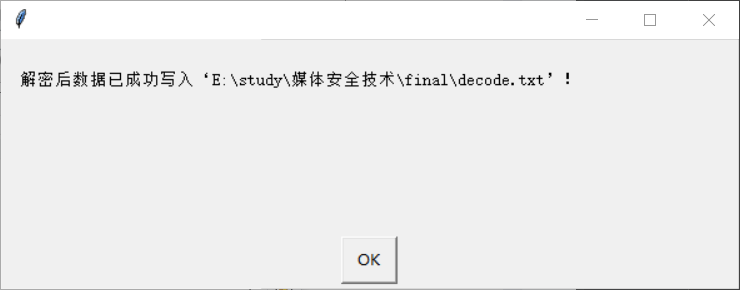










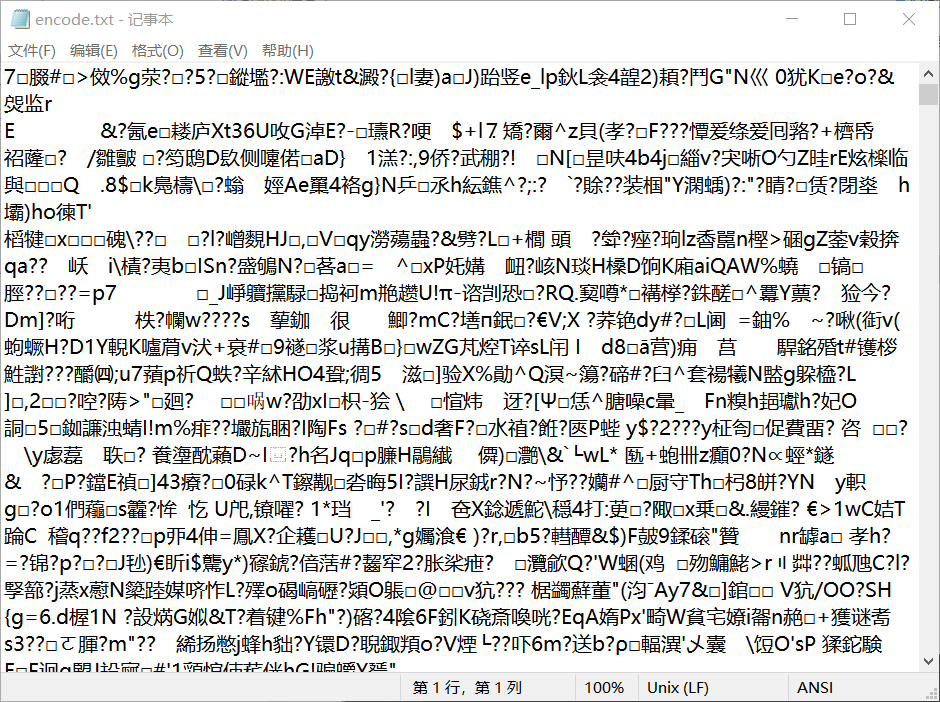


**4.2.结果**

密钥：



加密后文件：



解密后文件：

