HW1

1. 凯撒密文

FBUQIUUDSHOFJOEKHDQCUMYJXJXUIQCUAUOQDTKFBEQTJEBUQHDYDWYDPZK

因为我们已知凯撒密文是用“偏移”来进行加密的，且密钥的数量最多为25个，我们可以使用枚举法，轻易的得到最后可被语义解释的结果。

我通过python来实现这一问题（因为大写字母往往会影响我对于词义的理解，因而我将其全部转化为小写，这并不会改变密钥的偏移量）：

str='FBUQIUUDSHOFJOEKHDQCUMYJXJXUIQCUAUOQDTKFBEQTJEBUQHDYDYDPZK'

str=str.lower()

str1=''

for j in range(1,25):

for i in range(0,len(str)):

ch=ord(str[i])

if ch-26+j>=97:

ch=ch-26+j

else:ch=ch+j

str1=str1+(chr(ch))

print(str1)

str1=''

#get the result ---------->

fbuqiuudshofjoekhdqcumyjxjxuiqcuauoqdtkfbeqtjebuqhdydydpzk

gcvrjvvetipgkpflierdvnzkykyvjrdvbvpreulgcfrukfcvriezezeqal

hdwskwwfujqhlqgmjfsewoalzlzwksewcwqsfvmhdgsvlgdwsjfafafrbm

iextlxxgvkrimrhnkgtfxpbmamaxltfxdxrtgwniehtwmhextkgbgbgscn

jfyumyyhwlsjnsiolhugyqcnbnbymugyeysuhxojfiuxnifyulhchchtdo

kgzvnzzixmtkotjpmivhzrdococznvhzfztviypkgjvyojgzvmididiuep

lhawoaajynulpukqnjwiasepdpdaowiagauwjzqlhkwzpkhawnjejejvfq

mibxpbbkzovmqvlrokxjbtfqeqebpxjbhbvxkarmilxaqlibxokfkfkwgr

njcyqcclapwnrwmsplykcugrfrfcqykcicwylbsnjmybrmjcyplglglxhs

okdzrddmbqxosxntqmzldvhsgsgdrzldjdxzmctoknzcsnkdzqmhmhmyit

**pleaseencryptyournamewiththesamekeyanduploadtolearnininzju**

qmfbtffodszquzpvsobnfxjuiuiftbnflfzboevqmpbeupmfbsojojoakv

rngcuggpetarvaqwtpcogykvjvjgucogmgacpfwrnqcfvqngctpkpkpblw

sohdvhhqfubswbrxuqdphzlwkwkhvdphnhbdqgxsordgwrohduqlqlqcmx

tpiewiirgvctxcsyvreqiamxlxliweqioicerhytpsehxspievrmrmrdny

uqjfxjjshwduydtzwsfrjbnymymjxfrjpjdfsizuqtfiytqjfwsnsnseoz

vrkgykktixevzeuaxtgskcoznznkygskqkegtjavrugjzurkgxtototfpa

wslhzllujyfwafvbyuhtldpaoaolzhtlrlfhukbwsvhkavslhyupupugqb

xtmiammvkzgxbgwczviumeqbpbpmaiumsmgivlcxtwilbwtmizvqvqvhrc

yunjbnnwlahychxdawjvnfrcqcqnbjvntnhjwmdyuxjmcxunjawrwrwisd

zvokcooxmbizdiyebxkwogsdrdrockwouoikxnezvykndyvokbxsxsxjte

awpldppyncjaejzfcylxphtesespdlxpvpjlyofawzloezwplcytytykuf

bxqmeqqzodkbfkagdzmyqiuftftqemyqwqkmzpgbxampfaxqmdzuzuzlvg

cyrnfrrapelcglbheanzrjvgugurfnzrxrlnaqhcybnqgbyrneavavamwh

dzsogssbqfmdhmcifboaskwhvhvsgoasysmobridzcorhczsofbwbwbnxi

我们发现在上述的输出结果中，高亮的语句具有正常的语义；

明文为：please encrypt your name with the same key and upload to learn in in zju

因而我们可以借此推出密钥的偏移量，从p---->f，而我姓名的拼音为mayuxiao，因此可以采用下述的python语句进行明文执行：

s='mayuxiao'

result=''

for i in range(0,len(s)):

ch=ord(s[i])

ch=ch+ord('p')-ord('f')

if ch - 26 >= 97:

ch = ch - 26

result=result+chr(ch)

print(result)

输出为 wkiehsky

1. Vignere密文

ktbueluegvitnthuexmonveggmrcgxptlyhhjaogchoemqchpdnetxupbqntietiabpsmaoncnwvoutiugtagmmqsxtvxaoniiogtagmbpsmtuvvihpstpdvcrxhokvhxotawswquunewcgxptlcrxtevtubvewcnwwsxfsnptswtagakvoyyak

首先将str转换为list，方便python进行操作

str=[]

for i in range(0,len(s)):

str.append(s[i])

#print(str)

copy=str

第一轮思考，感觉序列中有很多重复的字母，想以此为突破口，尝试程序：

for i in range(0,len(str)-1):

if i!=len(str):

if str[i]==str[i+1]:

str[i]=str[i].upper()

str[i+1]=str[i+1].upper()

print(str[i])

print(i)

得到重复结果序列：

# 11 50 11 10 5 36 25

突然发现没有明确的重复意义，因为重复字母并不相同，且偏移量的外推也无效，失败！

第二轮思考，寻找完全重复序列--->

因为每一个重复序列出现次数均较少，所以采用多个重复序列来弥补缺陷，序列长度分布从2-5不等。

for i in range(0,len(copy)-1):

if copy[i]=='u' and copy[i+1]=='e':

print(i)

得到序列间隔结果3 9

for i in range(0,len(copy)-1):

if copy[i]=='c' and copy[i+1]=='h':

print(i)

得到序列结果 6

for i in range(0,len(copy)-1):

if copy[i]=='g' and copy[i+1]=='m':

print(i)

得到序列结果 60 18

for i in range(0,len(copy)-1):

if copy[i]=='s' and copy[i+1]=='x':

print(i)

得到序列结果75

for i in range(0,len(copy)-1):

if copy[i]=='c' and copy[i+1]=='g' and copy[i+2]=='x'

and copy[i+3]=='p'and copy[i+4]=='t' :

print(i)

得到序列结果 114

for i in range(0,len(copy)-1):

if copy[i]=='b' and copy[i+1]=='p' and copy[i+2]=='s' and copy[i+3]=='m' :

print(i)

得到序列结果 39

for i in range(0,len(copy)-1):

if copy[i]=='g' and copy[i+1]=='t' and copy[i+2]=='a' and copy[i+3]=='g' :

print(i)

得到序列结果 18

乘法因子分析后，发现 3 极有可能为key的长度（甚至猜测有可能是zju)，

接下来进行频率分析尝试一下我的猜测：

str1=s[::3]

ch1=len(str1)

str2=s[1::3]

ch2=len(str2)

print(ch2)

str3=s[2::3]

ch3=len(str3)

dict={'a':0,'b':0,'c':0,'d':0,'e':0,'f':0,'g':0,'h':0,'i':0,'j':0,

'k':0,'l':0,'m':0,'n':0,'o':0,'p':0,'q':0,'r':0,'s':0,'t':0,

'u':0,'v':0,'w':0,'x':0,'y':0,'z':0}

for i in range(0,len(str1)):

for key in dict:

if str1[i]==key:

dict[key]=dict[key]+1

print(dict)

for key in dict:

dict[key]=dict[key]/ch1

print(dict)

分析：

在str1中，p出现最多，v、g第二，怀疑p对映于e；但结合了后续另两个切片的结果，发现输出结果无语义，失败！

思考后发现，不能仅关注于统计数量最多的字母，简单地对映于e，需要关照全字母表进行适当的匹配，因而g对映于e更为妥当--->C行。

在str2中，t最多，e第二，根据全表分析，e有可能对应于e--->A行.

在str3中，x最多，结合全表，x本身有可能对应于e--->T行

for i in range(0,len(str)):

ch = ord(str[i])

if (i+1)%3==1:

ch=ch+ord('e')-ord('g')

elif (i+1)%3==2:

ch = ch + ord('e') - ord('e')

else :

ch = ch + ord('e') - ord('x')

if ch<97:

ch=ch+26

str[i]=chr(ch)

print(''.join(str))

根据上述分析，密钥可能为cat，符合人之常情，用一个简单的单词qaq.

尝试分析，输出结果语义化后为一下语句；

明文：

it is essential to seek out enemy agents who have come to conduct espionage against you and to bribe them to serve you give them instructions and care for them thus doubled agents are recruited and used sun tzut heart of war.

重要的是要找出那些来对你进行间谍活动的敌方特工，并贿赂他们为你服务。你要给他们指示和照顾他们，这样就可以招募和利用双重间谍。

（成功，但是这怎么执行qaq）。

1. unknown 密文

详见图片，在解的过程还是出现了不少的问题，但是在解开密码后，好像一切又变得简单了，像忘记了自己是如何在of和on的差异中作出解码选择的......感觉这个解密还是很看由部分字母联想或者说猜到单词的能力的；以下图片是我的大致解题思路。

“最重要的知识是知道我们被神秘的事物包围！！！”

