## task1

下载所需要的文件

<https://seedsecuritylabs.org/Labs_16.04/Crypto/Crypto_Encryption/>

加密对文档做操作

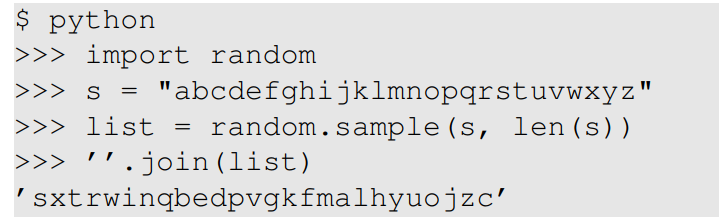
首先将大写字母转换为小写字母

tr [:upper:] [:lower:] < article.txt > lowercase.txt

而后删除标点符号等，不属于下面这三类字符集的字符。

tr -cd ’[a-z][\n][:space:]’ < lowercase.txt > plaintext.txt

而后生成一个打乱顺序的table



最后映射替换即可

tr ’abcdefghijklmnopqrstuvwxyz’ ’sxtrwinqbedpvgkfmalhyuojzc’ \ < plaintext.txt > ciphertext.txt

词频分析网站

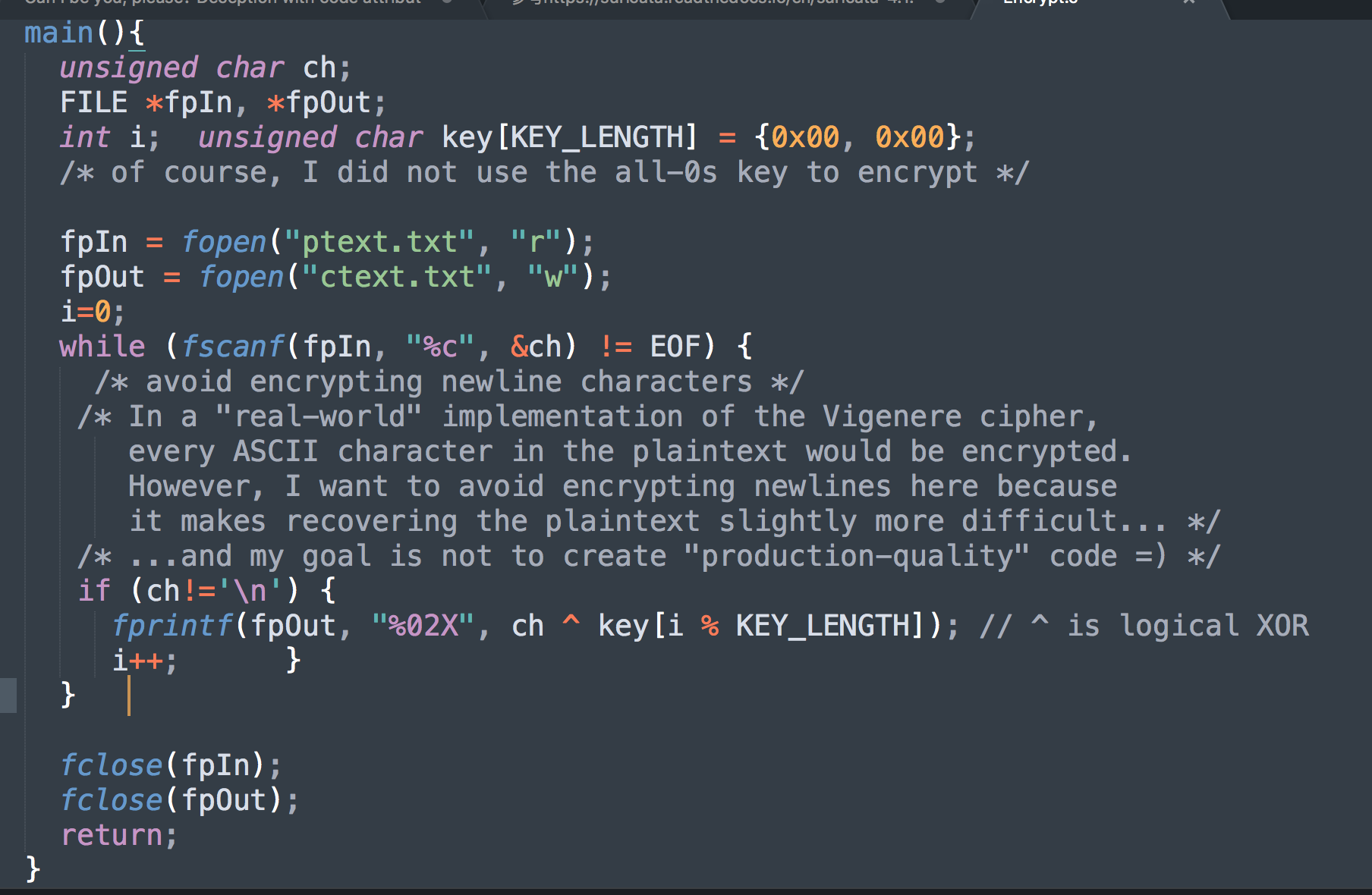
<https://quipqiup.com/>

### 结果

the oscars turn on sunday which seems about right after this long strange awards trip the bagger feels like a nonagenarian too the awards race was bookended by the demise of harvey weinstein at its outset and the apparent implosion of his film company at the end and it was shaped by the emergence of metoo times up blackgown politics armcandy activism and a national conversation as brief and mad as a fever dream about whether there ought to be a president winfrey the season didnt just seem extra long it was extra long because the oscars were moved to the first weekend in march to avoid conflicting with the closing ceremony of the winter olympics thanks pyeongchang one big question surrounding this years academy awards is how or if the ceremony will address metoo especially after the golden globes which became a jubilant comingout party for times up the movement spearheaded by powerful hollywood women who helped raise millions of dollars to fight sexual harassment around the country signaling their support golden globes attendees swathed themselves in black sported lapel pins and sounded off about sexist power imbalances from the red carpet and the stage on the air e was called out about pay inequity after its former anchor catt sadler quit once she learned that she was making far less than a male cohost and during the ceremony natalie portman took a blunt and satisfying dig at the allmale roster of nominated directors how could that be topped as it turns out at least in terms of the oscars it probably wont be women involved in times up said that although the globes signified the initiatives launch they never intended it to be just an awards season campaign or one that became associated only with redcarpet actions instead a spokeswoman said the group is working behind closed doors and has since amassed million for its legal defense fund which after the globes was flooded with thousands of donations of or less from people in some countries no call to wear black gowns went out in advance of the oscars though the movement will almost certainly be referenced before and during the ceremony especially since vocal metoo supporters like ashley judd laura dern and nicole kidman are scheduled presenters another feature of this season no one really knows who is going to win best picture arguably this happens a lot of the time inarguably the nailbiter narrative only serves the awards hype machine but often the people forecasting the race socalled oscarologists can make only educated guesses the way the academy tabulates the big winner doesnt help in every other category the nominee with the most votes wins but in the best picture category voters are asked to list their top movies in preferential order if a movie gets more than percent of the firstplace votes it wins when no movie manages that the one with the fewest firstplace votes is eliminated and its votes are redistributed to the movies that garnered the eliminated ballots secondplace votes and this continues until a winner emerges it is all terribly confusing but apparently the consensus favorite comes out ahead in the end this means that endofseason awards chatter invariably involves tortured speculation about which film would most likely be voters second or third favorite and then equally tortured conclusions about which film might prevail in it was a tossup between boyhood and the eventual winner birdman in with lots of experts betting on the revenant or the big short the prize went to spotlight last year nearly all the forecasters declared la la land the presumptive winner and for two and a half minutes they were correct before an envelope snafu was revealed and the rightful winner moonlight was crowned this year awards watchers are unequally divided between three billboards outside ebbing missouri the favorite and the shape of water which is the baggers prediction with a few forecasting a hail mary win for get out but all of those films have historical oscarvoting patterns against them the shape of water has nominations more than any other film and was also named the years best by the producers and directors guilds yet it was not nominated for a screen actors guild award for best ensemble and no film has won best picture without previously landing at least the actors nomination since braveheart in this year the best ensemble sag ended up going to three billboards which is significant because actors make up the academys largest branch that film while divisive also won the best drama golden globe and the bafta but its filmmaker martin mcdonagh was not nominated for best director and apart from argo movies that land best picture without also earning best director nominations are few and far between

## task2

https://blog.csdn.net/jakekong/article/details/79884365



使用重合因子过于麻烦了，参考网络上给的wp，这是南邮的一道ctf赛题。

<https://blog.csdn.net/jakekong/article/details/79884365>

利用了解密出来的应该是可见字符，来进行爆破。可见字符范围是32到126。

一个比较重要的点，是关于python中的字符编码以及异或，和c语言中的联系。可以看到在python中，直接int类型异或，而后chr()函数转char即可。不需要考虑类型差异，也不需要考虑反码等因素，c语言用了unsigned了。以及c语言中可能存在的截取。

所以最难的部分解决后，程序还是比较好写的。

### 代码

def findindexkey(subarr):#该函数可以找出将密文subarr解密成可见字符的所有可能值

visiable\_chars=[]#可见字符

for x in range(32,126):

visiable\_chars.append(chr(x))

#print(vi)

test\_keys=[]#用于测试密钥

ans\_keys=[]#用于结果的返回

for x in range(0x00,0xFF):# 枚举密钥里所有的值

test\_keys.append(x)

ans\_keys.append(x)

for i in test\_keys:#对于0x00~0xFF里的每一个数i和subarr里的每个值s异或

for s in subarr:

if chr(s^i) not in visiable\_chars:#用i解密s，如果解密后明文不是可见字符，说明i不是密钥

ans\_keys.remove(i)#去掉ans\_keys里测试失败的密钥

break

#print(ans\_keys)

return ans\_keys

strmi='F96DE8C227A259C87EE1DA2AED57C93FE5DA36ED4EC87EF2C63AAE5B9A7EFFD673BE4ACF7BE8923C\

AB1ECE7AF2DA3DA44FCF7AE29235A24C963FF0DF3CA3599A70E5DA36BF1ECE77F8DC34BE129A6CF4D126BF\

5B9A7CFEDF3EB850D37CF0C63AA2509A76FF9227A55B9A6FE3D720A850D97AB1DD35ED5FCE6BF0D138A84C\

C931B1F121B44ECE70F6C032BD56C33FF9D320ED5CDF7AFF9226BE5BDE3FF7DD21ED56CF71F5C036A94D96\

3FF8D473A351CE3FE5DA3CB84DDB71F5C17FED51DC3FE8D732BF4D963FF3C727ED4AC87EF5DB27A451D47E\

FD9230BF47CA6BFEC12ABE4ADF72E29224A84CDF3FF5D720A459D47AF59232A35A9A7AE7D33FB85FCE7AF5\

923AA31EDB3FF7D33ABF52C33FF0D673A551D93FFCD33DA35BC831B1F43CBF1EDF67F0DF23A15B963FE5DA\

36ED68D378F4DC36BF5B9A7AFFD121B44ECE76FEDC73BE5DD27AFCD773BA5FC93FE5DA3CB859D26BB1C63C\

ED5CDF3FE2D730B84CDF3FF7DD21ED5ADF7CF0D636BE1EDB79E5D721ED57CE3FE6D320ED57D469F4DC27A8\

5A963FF3C727ED49DF3FFFDD24ED55D470E69E73AC50DE3FE5DA3ABE1EDF67F4C030A44DDF3FF5D73EA250\

C96BE3D327A84D963FE5DA32B91ED36BB1D132A31ED87AB1D021A255DF71B1C436BF479A7AF0C13AA14794'

arr=[]#密文，每个元素为字符的ascii码，所以是对以上密文进行两字符的分割

for x in range(0,len(strmi),2):

arr.append(int(strmi[x:2+x],16))

for keylen in range(1,14):#枚举密钥的长度1~14

for index in range(0,keylen):#对密钥里的第index个进行测试

subarr=arr[index::keylen]#每隔keylen长度提取密文的内容，提取出来的内容都被密文的第index个加密

ans\_keys=findindexkey(subarr)#找出密钥中第index个的可能的值

print('keylen=',keylen,'index=',index,'keys=',ans\_keys)

if ans\_keys:#如果密钥第index个有可能存在，尝试用密钥的index个去解密文

ch=[]

for x in ans\_keys:

ch.append(chr(x^subarr[0]))

print(ch)

#运行到这里，观察输出可以发现，密钥长度为7时有解

print('###############')

import string

def findindexkey2(subarr):#再造一个函数筛选密钥

test\_chars=string.ascii\_letters+string.digits+','+'.'+' '#将检查的字符改为英文+数字+逗号+句号+空格

#print(test\_chars)

test\_keys=[]#用于测试密钥

ans\_keys=[]#用于结果的返回

for x in range(0x00,0xFF):# 枚举密钥里所有的值

test\_keys.append(x)

ans\_keys.append(x)

for i in test\_keys:#对于0x00~0xFF里的每一个数i和substr里的每个值s异或

for s in subarr:

if chr(s^i) not in test\_chars:#用i解密s，如果解密后不是英文、数字、逗号、句号、空格，说明i不是密钥

ans\_keys.remove(i)#去掉ans\_keys里测试失败的密钥

break

#print(ans\_keys)

return ans\_keys

vigenerekeys=[]#维基尼尔密码的密钥

for index in range(0,7):#已经知道密钥长度是7

subarr=arr[index::7]

vigenerekeys.append(findindexkey2(subarr))

print(vigenerekeys)#输出的是[[186], [31], [145], [178], [83], [205], [62]].

print("#########")

ming=''

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

ming=ming+chr(arr[i]^vigenerekeys[i%7][0])

print(ming)

### 结果

Cryptography is the practice and study of techniques for, among other things, secure communication in the presence of attackers. Cryptography has been used for hundreds, if not thousands, of years, but traditional cryptosystems were designed and evaluated in a fairly ad hoc manner. For example, the Vigenere encryption scheme was thought to be secure for decades after it was invented, but we now know, and this exercise demonstrates, that it can be broken very easily.

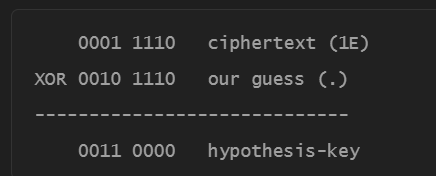
## task3

参考

http://andrewhe.me/2016/perfect-secrecy/

都以1E结尾，应该是一个比较特殊的字符，在英文句子中，应当为句号。

获得密钥后和特殊字符0f异或，得到?，证明猜测正确。



接下来就要猜测，猜测消息中有明文the，

### 代码（python2）

# Verifier function. Returns false if any of the characters in the possible solution are not English language characters.

def verify\_results(results):

for i in results:

if i < 32 or i > 122:

return False

else:

if i < 48:

if i != 32 and i != 33 and i != 44 and i != 45 and i != 46: return False

elif i > 57 and i < 65:

if i != 63 and i != 58 and i != 59: return False

elif i > 90 and i < 97:

return False

return True

# List of ciphertexts encrypted with the same key

c = ["BB3A65F6F0034FA957F6A767699CE7FABA855AFB4F2B520AEAD612944A801E",

"BA7F24F2A35357A05CB8A16762C5A6AAAC924AE6447F0608A3D11388569A1E",

"A67261BBB30651BA5CF6BA297ED0E7B4E9894AA95E300247F0C0028F409A1E",

"A57261F5F0004BA74CF4AA2979D9A6B7AC854DA95E305203EC8515954C9D0F",

"BB3A70F3B91D48E84DF0AB702ECFEEB5BC8C5DA94C301E0BECD241954C831E",

"A6726DE8F01A50E849EDBC6C7C9CF2B2A88E19FD423E0647ECCB04DD4C9D1E",

"BC7570BBBF1D46E85AF9AA6C7A9CEFA9E9825CFD5E3A0047F7CD009305A71E"]

# Put ciphertexts into byte arrays

for i,e in enumerate(c):

c[i] = bytearray.fromhex(e)

# ASCII representation of the word, "the"

the\_ascii = [116, 104, 101]

# Try for each offset

for i in range(0, len(c)-3):

# Try for each ciphertext

for n in range(0, 7):

crib\_word = [the\_ascii[0]^c[n][i], the\_ascii[1]^c[n][i+1], the\_ascii[2]^c[n][i+2]]

ciphers = []

legit = True

for j in range(0, 7):

results = [crib\_word[0]^c[j][i], crib\_word[1]^c[j][i+1], crib\_word[2]^c[j][i+2]]

if verify\_results(results):

ciphers.append(results)

else:

# If the verification fails, end

legit = False

break

if legit:

print "Results for offset", i, "on ciphertext", n+1, ":"

for x in ciphers:

print chr(x[0]), chr(x[1]), chr(x[2])

### 输出

Results for offset 6 on ciphertext 2 :

l a n

t h e

r r e

h o u

k t

s p

e c

Results for offset 8 on ciphertext 5 :

n n i

e o

e n t

u l d

t h e

p u r

c a d

Results for offset 11 on ciphertext 4 :

: x

: s y

t o l

t h e

- ? s

1 m

1 k

Results for offset 18 on ciphertext 1 :

t h e

d u n

d : t

c : t

s : f

7 n h

r n t

Results for offset 19 on ciphertext 7 :

r y t

o r

h o

h o

z o

t t a

t h e

Results for offset 23 on ciphertext 3 :

9 r s

; ; t

t h e

0 t

8 t w

t t n

t o h

Results for offset 26 on ciphertext 3 :

d s o

e o s

t h e

c r i

7 r i

r : i

v t

### 分析

选取其中有可能的结果，进行手动分析。