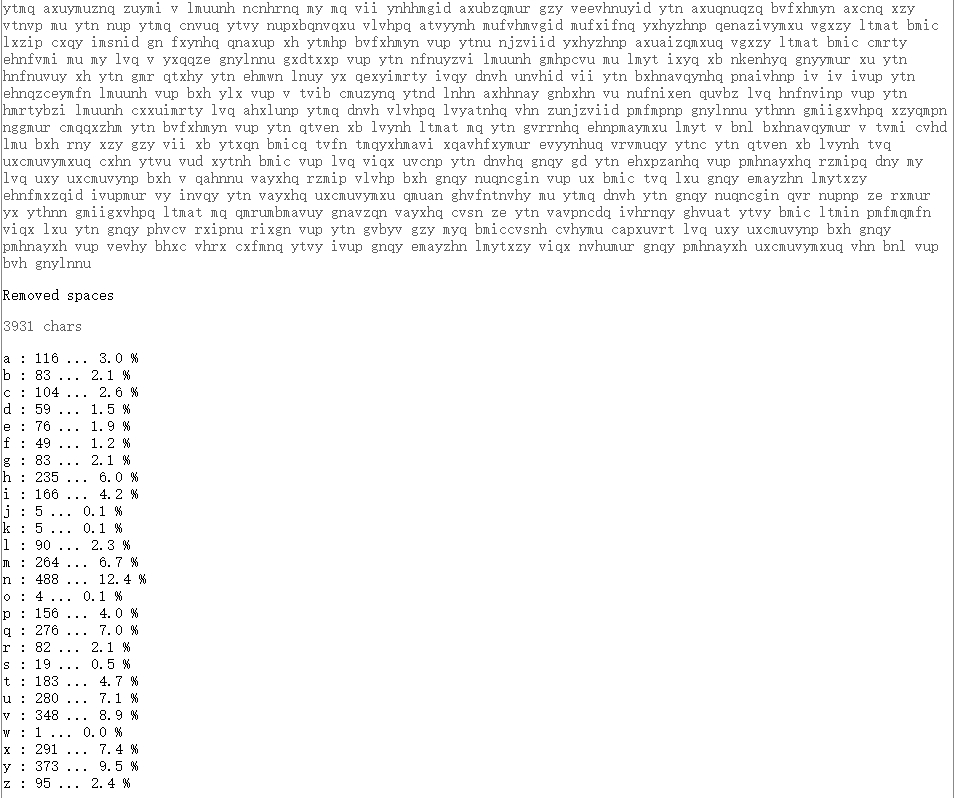
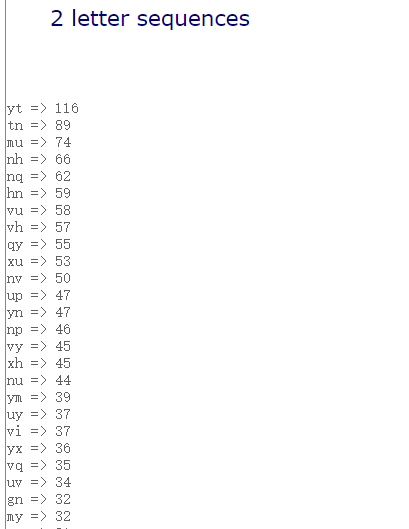
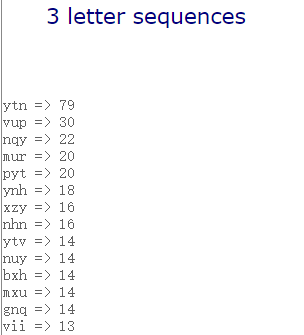
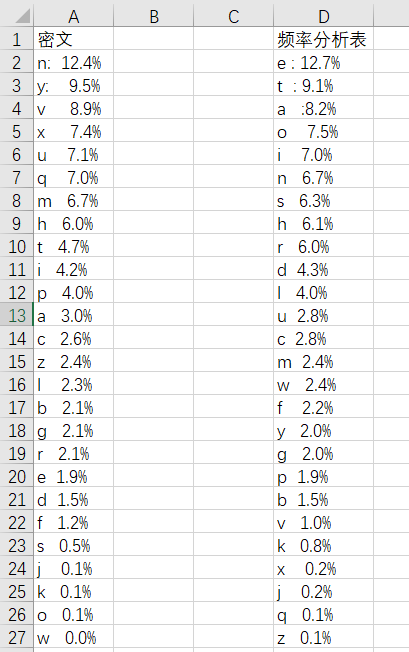
**Secret-Key Encryption Lab**

姓名：严佳豪，学号：57118136

**实验1 Task 1: Frequency Analysis Against Monoalphabetic Substitution Cipher**

从实验室网站上下载密文，传至所给网站中解析出各字母出现频率如下

和频率分析表对比如下：

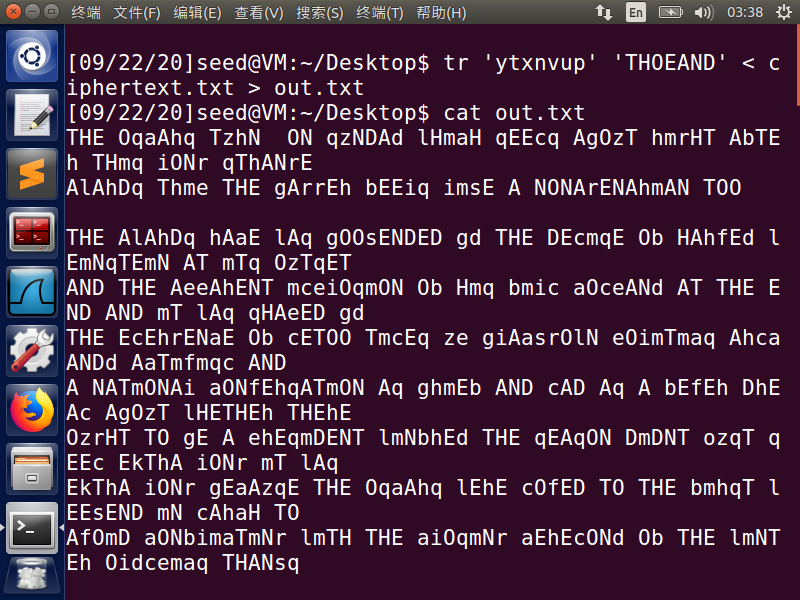
结合密文中v可单独出现，以及英文双字母的概率最大的30对字母按概率大小排列为：

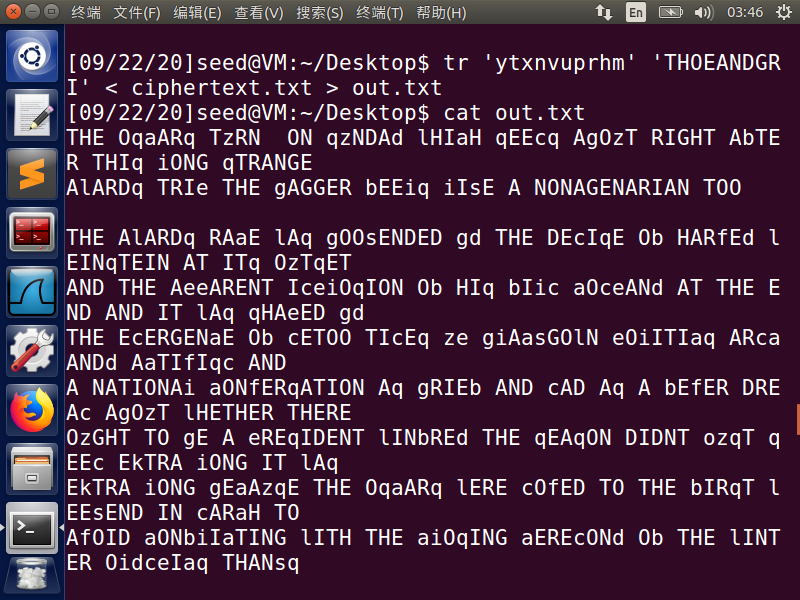
th he in er an re ed on es st en at to nt ha nd ou ea ng as or ti is et it ar te se hi of

概率最大的20组三字母按概率大小排列为：

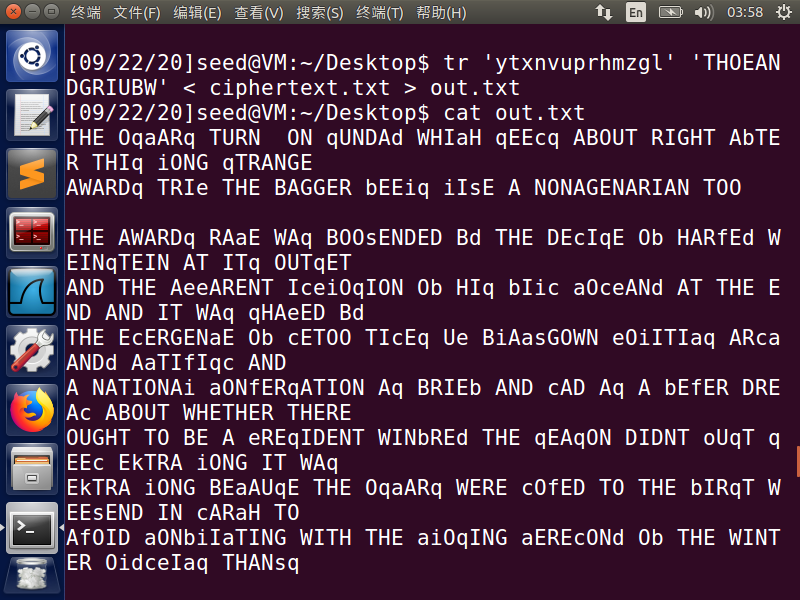
the ing and her ere ent tha nth was eth for dth hat she ion his sth ers ver

可以初步得出，y->T; t->H; n->E; v->A; u->N;p->D; x->O

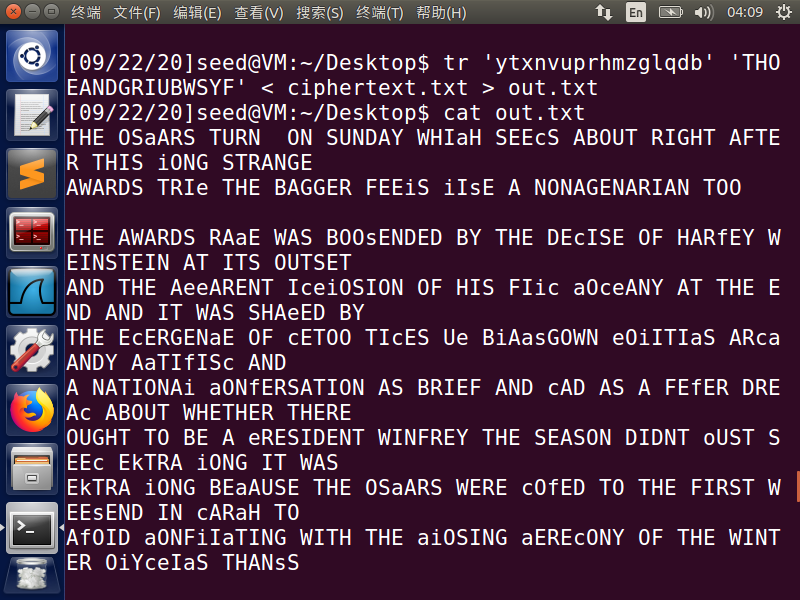
Tr 后输出文本如下

根据NONArENAhmAN 一词，查阅发现仅有NONAGENARIAN能对应，故 r->G; h->R; m->I

根据TzRN,gAGGER,lERE 几词，及频率表可推测出z->U；g->B; l->W



根据THIq,qTRANGE及频率表推出q->S;根据语法、频率表、单词Bd推出d->Y

根据剩余字母及频率表及Ob推出b->F;

根据NATIONAi FEEiS iIsE THANsS及剩余字母和频率表推出 i->L; s->K

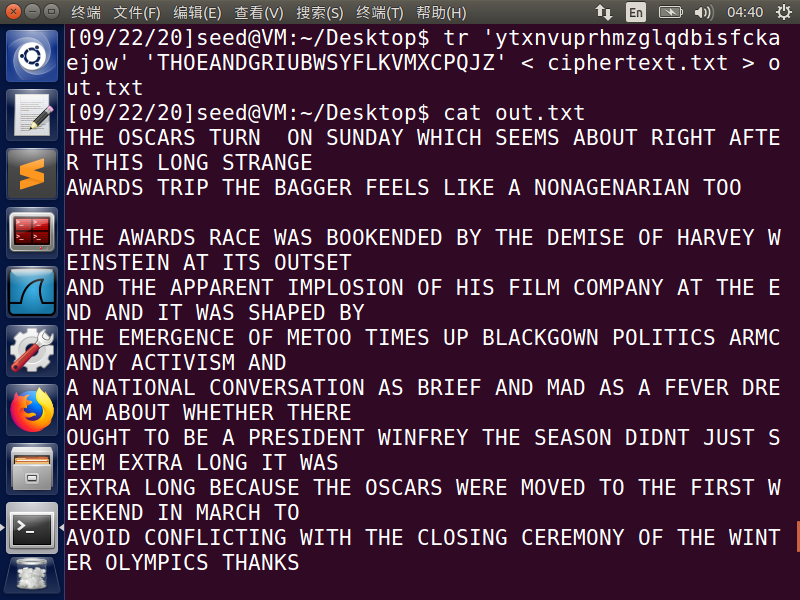
根据 FEfER DREAc 及剩余字母推出 f->V; c->M; 根据剩余字母及EkTRA推测出k->X;

根据WHIaH及剩余字母推出a->C;根据Ue及剩余字母推出e->P;

根据EjUALLY及剩余字母推出j->Q; 根据oUST及剩余字母推出o->J; 剩余最后一对w->Z

整理密钥对如下：

**|** a->C **|** b->F **|** c->M **|** d->Y **|** e->P **|** f->V **|** g->B **|** h->R **|** i->L **|** j->Q **|** k->X **|** l->W **|** m->L **|** n->E **|** o->J **|** p->D **|** q->S **|** r->G **|** s->K **|** t->H **|** u->N **|** v->A **|** w->Z **|** x->O **|** y->T **|** z->U



全部转换后明文如下

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

**实验二 1. 破解维吉尼亚密码**

观察密文，发现为字母数字组合，猜测原文中可能存在其他非字母字符，且可能需经过ASCⅡ码转换，重合指数不便计算，于是采用暴力破解的方法

所用python代码如下：

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))

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

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

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

sum=0

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

sum+=calc\_sum\_of\_Frequency\_squares(subarr)

sum = sum/keylen

print("{}: {:.10f}".format(str(keylen),sum))

# 计算各个子串平均频率平方和,发现keylen为7时最大，可能为密钥长

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

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

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)

得到明文如下

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.

**实验二：2. 以下密文使用了重复的一次一密密码，请通过异或运算和ASCII码的计算规律破解原始消息。**

根据其为密钥重复型的一次一密密码，采用MTP攻击方式对其进行破解，所用代码如下：

#!/usr/bin/env python3

from typing import List

import binascii

import argparse

SPACE = ord(' ')

def main():

parser = argparse.ArgumentParser(description='Many-time Pad Cracker')

parser.add\_argument(

'--filename',

type=str,

help='Name of the file containing the ciphertexts (default: ciphertexts.txt)',

default='ciphertexts.txt'

)

parser.add\_argument(

'-K', '--getkey',

action='store\_true',

help='Print cracked key instead of cracked cleartexts.'

)

parser.add\_argument(

'-k', '--key',

help='Encrypt messages with provided key.',

default=''

)

args = parser.parse\_args()

try:

with open(args.filename) as file:

ciphertexts = [binascii.unhexlify(line.rstrip()) for line in file]

except Exception as e:

print('Cannot crack {} --- {}'.format(args.filename, e))

raise SystemExit(-1)

cleartexts = [bytearray(b'?' \* len(line)) for line in ciphertexts]

if args.key:

decrypt(ciphertexts, cleartexts, args.key)

else:

crack(ciphertexts, cleartexts, args.getkey)

def decrypt(ciphertexts: List[bytes], cleartexts: List[bytearray], input\_key: str) -> None:

""" Decrypt ciphertexts using provided key and print cleartexts """

key = binascii.unhexlify(input\_key.rstrip())

for row in range(len(ciphertexts)):

for column in range(len(ciphertexts[row])):

cleartexts[row][column] = ciphertexts[row][column] ^ key[column % len(key)]

print(cleartexts[row].decode('ascii'))

def crack(ciphertexts: List[bytes], cleartexts: List[bytearray], getkey: bool) -> None:

""" Try to decrypt ciphertexts and print cleartexts or key """

max\_length = max(len(line) for line in ciphertexts)

key = bytearray(max\_length)

key\_mask = [False] \* max\_length

for column in range(max\_length): # go over characters from the beginning of lines

pending\_ciphers = [line for line in ciphertexts if len(line) > column]

for cipher in pending\_ciphers:

if is\_space(pending\_ciphers, cipher[column], column):

key[column] = cipher[column] ^ SPACE

key\_mask[column] = True

i = 0

for clear\_row in range(len(cleartexts)):

if len(cleartexts[clear\_row]) != 0 and column < len(cleartexts[clear\_row]):

result = cipher[column] ^ pending\_ciphers[i][column]

if result == 0:

cleartexts[clear\_row][column] = SPACE

elif chr(result).isupper(): # XOR with space return letter with swapped case

cleartexts[clear\_row][column] = ord(chr(result).lower())

elif chr(result).islower(): # XOR with space return letter with swapped case

cleartexts[clear\_row][column] = ord(chr(result).upper())

i += 1

break

if getkey:

for pos in range(max\_length):

if key\_mask[pos]:

print('{0:02x}'.format(key[pos]), end='')

else:

print('\_\_', end='')

print()

else:

print('\n'.join(line.decode('ascii') for line in cleartexts))

def is\_space(rows: List[bytes], current: int, column: int) -> bool:

"""

Return whether the current byte is encrypted space

If the current byte is space, XORing with other bytes should return alpha char or zero (when space)

"""

for row in rows:

result = row[column] ^ current

if not (chr(result).isalpha() or result == 0):

return False

return True

if \_\_name\_\_ == '\_\_main\_\_':

main()

解出来部分原文如下：

? am p?a?n?ng a s?cr?t missio??

?e is ?h? ?nly pe?so? to trus??

?he cu?r?n? plan ?s ?op secre??

?hen s?o?l? we me?t ?o do thi??

? thin? ?h?y shou?d ?ollow hi??

?his i? ?u?er tha? t?at one i??

?ot on? ?a?et is ?et?er than ??

根据密文结尾及英文语法单词猜测补全后如下：

I am planning a secret mission.

He is the only person to trust.

The current plan is top secret.

When should we meet to do this?

I think they should follow him.

This is ?u?er than that one i??

?ot on? ?a?et is better than ??

对第一句进行ASCII码转码

01001001 00100000 01100001 01101101 00100000 01110000 01101100 01100001 01101110 01101110 01101001 01101110 01100111 00100000 01100001 00100000 01110011 01100101 01100011 01110010 01100101 01110100 00100000 01101101 01101001 01110011 01110011 01101001 01101111 01101110 00101110

对第一句密文进行二进制转码

10111011 00111010 01100101 11110110 11110000 00000011 01001111 10101001 01010111 11110110 10100111 01100111 01101001 10011100 11100111 11111010 10111010 10000101 01011010 11111011 01001111 00101011 01010010 00001010 11101010 11010110 00010010 10010100 01001010 10000000 00011110

异或得到二进制密钥如下

11110010 00011010 00000100 10011011 11010000 01110011 00100011 11001000

00111001 10011000 11001110 00001001 00001110 10111100 10000110 11011010

11001001 11100000 00111001 10001001 00101010 01011111 01110010 01100111

10000011 10100101 01100001 11111101 00100101 11101110 00110000

对第二句进行ASCII码转码

01001000 01100101 00100000 01101001 01110011 00100000 01110100 01101000 01100101 00100000 01101111 01101110 01101100 01111001 00100000 01110000 01100101 01110010 01110011 01101111 01101110 00100000 01110100 01101111 00100000 01110100 01110010 01110101 01110011 01110100 00101110

对第二句密文进行二进制转码

10111010 01111111 00100100 11110010 10100011 01010011 01010111 10100000 01011100 10111000 10100001 01100111 01100010 11000101 10100110 10101010 10101100 10010010 01001010 11100110 01000100 01111111 00000110 00001000 10100011 11010001 00010011 10001000 01010110 10011010 00011110

解出密钥如下，与第一句解出的密钥相同，说明成功获取密钥

11110010 00011010 00000100 10011011 11010000 01110011 00100011 11001000

00111001 10011000 11001110 00001001 00001110 10111100 10000110 11011010

11001001 11100000 00111001 10001001 00101010 01011111 01110010 01100111

10000011 10100101 01100001 11111101 00100101 11101110 00110000

对密钥进行十六进制转码后如下

f2 1a 04 9b d0 73 23 c8 39 98 ce 09 0e bc 86 da c9 e0 39 89 2a 5f 72 67 83 a5 61 fd 25 ee 30

因为部分字符无法显示所以就放出十六进制的密钥

最终根据密钥解出明文如下：

I am planning a secret mission.

He is the only person to trust.

The current plan is top secret.

When should we meet to do this?

I think they should follow him.

This is purer than that one is.

Not one cadet is better than I.