**Lab7实验报告**

57118111 王雯

**实验目的**：熟悉秘密密钥加密的概念，获得有关加密算法，加密模式，填充和初始向量（IV）的第一手经验。 此外，学生将能够使用工具和编写程序来加密/解密消息。

涉及以下主题：

•密钥加密

•替代密码和频率分析

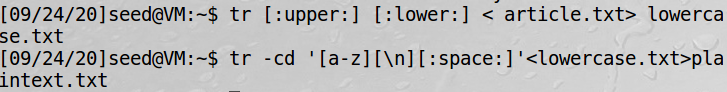
•加密模式和填充

•使用加密库进行编程

Task1: Frequency Analysis Against Monoalphabetic Substitution Cipher

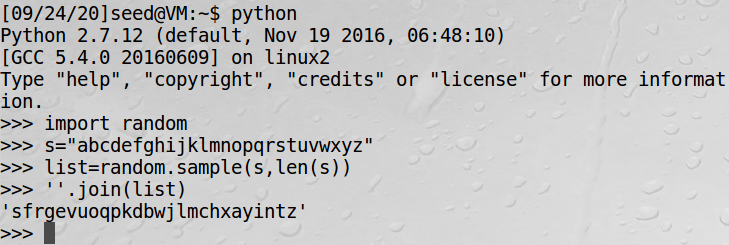
步骤一：

步骤1:对原文做一些简化。我们把所有的大写字母转换成小写字母，然后去掉所有的标点符号和数字

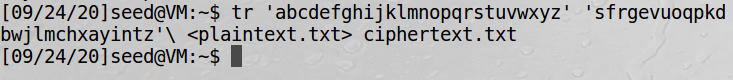


步骤二：

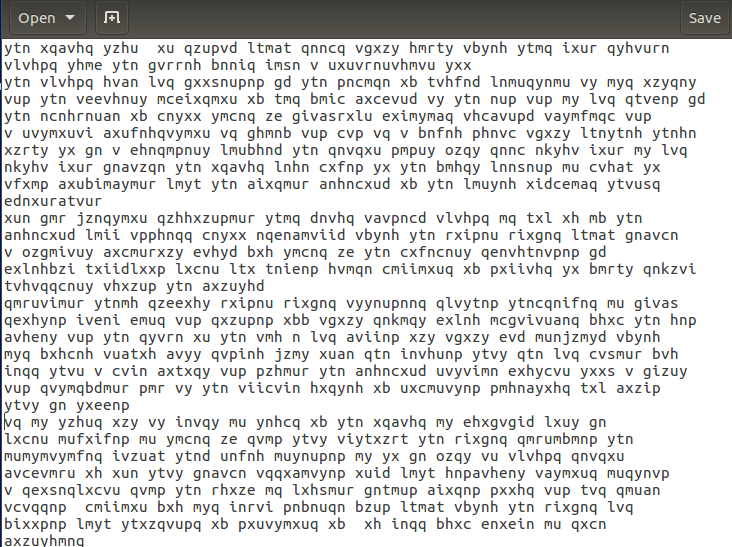
生成加密密钥，即替换表。我们将改变字母表，使用Python从a到z，并使用组合字母作为键



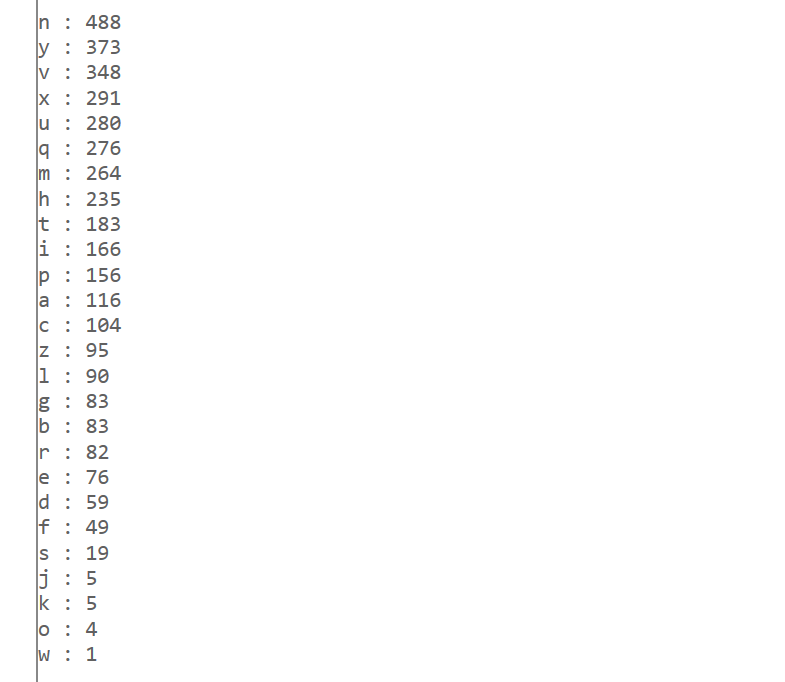
步骤三：

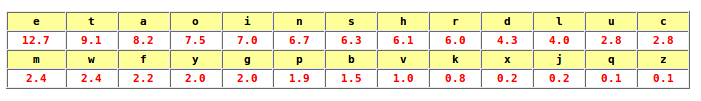


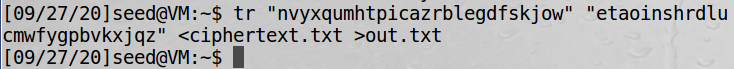
首先下载原始密文到ciphertext.txt



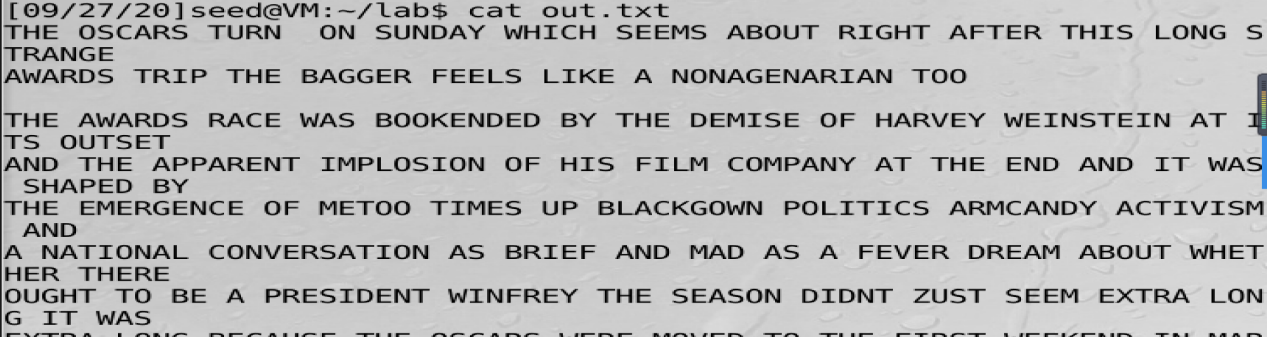
登陆到<http://www.richkni.co.uk/php/crypta/freq.php>网站进行分析密文







解密后的密文为：



得到具体明文

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 ZUST 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 ZUBILANT 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 ZUST 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 ZUDD 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

PRIwE 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

实验源码如下：

def findindexkey(subarr):

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

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]].

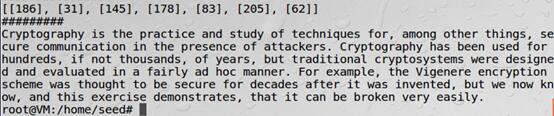
print("#########")

ming = ''

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

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

print(ming)



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.

Task3

实验源码如下：

import binascii  
import argparse  
  
SPACE = ord(' ')  
  
  
def countalphas(char, position, ciphertexts):  
 count = 0  
 for ciphertext in ciphertexts:  
 if len(ciphertext) > position:  
 if chr(ciphertext[position] ^ char).isalpha(): count += 1  
 return count  
  
  
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")  
 args = parser.parse\_args()  
 try:  
 with open(args.filename) as f:  
 ciphertexts = [binascii.unhexlify(line.rstrip()) for line in f]  
 # Cyphertexts puliti (tolgo i vuoti), anche se non è necessario  
 # ciphertexts = [c for c in ciphertexts if c]  
 cleartexts = [bytearray(b'?' \* len(c)) for c in ciphertexts]  
 except Exception as e:  
 print("Cannot crack {} --- {}".format(args.filename, e))  
 raise SystemExit(-1)  
  
 # 'a'.isalpha() => true  
 # '!'.isalpha() => false  
 # ord('z') => 122  
  
  
 for col in range(max([len(x) for x in ciphertexts])):  
 for c1 in ciphertexts:  
 for c2 in ciphertexts:  
 if (len(c1) > col) and (len(c2) > col):  
 if chr(c1[col] ^ c2[col]).isalpha():  
 for k, c in enumerate(ciphertexts):  
 if len(c) > col:  
 if countalphas(c1[col], col, ciphertexts) >= countalphas(c2[col], col, ciphertexts):  
 cleartexts[k][col] = c1[col] ^ 0b100000 ^ c[col]  
 else:  
 cleartexts[k][col] = c2[col] ^ 0b100000 ^ c[col]  
  
 break  
  
 for line in cleartexts:  
 print(line)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

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