实验 7 密钥加密实验

Task1：单字母替代密码的频率分析

单字母替代密码(也称为单字母密码)是不安全的，因为它可以进行频率分析。在这个实验室中，你会得到一个使用单字母密码加密的密码文本，也就是说，原始文本中的每个字母都被另一个字母替换，而替换并不变化(即在加密过程中，一个字母总是被同一个字母替换)，请使用频率分析找出原始文本。

步骤 1：创建 article.txt 和 test.txt 文件：

```
[09/27/20]seed@VM:~$ gedit article.txt
[09/27/20]seed@VM:~$ gedit test.text
[09/27/20]seed@VM:~$
```

步骤 2：将所有的大写转换为小写，然后删除所有的标点符号和数字：

```
[09/27/20]seed@VM:~$  tr [:upper:] [:lower:] < article.txt > lower
case.txt
[09/27/20]seed@VM:~$ tr -cd '[a-z][\n][:space:]' < lowercase.txt >
 plaintext.txt
```

步骤 3：生成加密密钥，即替换表，使用 Python 对字母从 a 到 z 进行排列，并使用排列后的字母作为键：

```
[09/27/20]seed@VM:~$ python
Python 2.7.12 (default, Nov 19 2016, 06:48:10)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more informat
ion.
>>> import random
>>> s = "abcdefghijklmnopqrstuvwxyz"
>>> list = random.sample(s, len(s))
>>> ''.join(list)
'kxqsdmozncuabwfigtlvjyehrp'
>>>
```

步骤 4：使用 tr 命令进行加密：

```
[09/27/20]seed@VM:~$ tr 'abcdefghijklmnopqrstuvwxyz' 'kxqsdmozncua
bwfigtlvjyehrp' < plaintext.txt > ciphertext.txt
```

步骤 5：分析密文中字母出现的频率和常见字母对：

```
Removed spaces

3931 chars

a : 116 ... 3.0 %
b : 83 ... 2.1 %
c : 104 ... 2.6 %
d : 59 ... 1.5 %
e : 76 ... 1.9 %
f : 49 ... 1.2 %
g : 83 ... 2.1 %
h : 235 ... 6.0 %
i : 166 ... 4.2 %
j : 5 ... 0.1 %
k : 5 ... 0.1 %
l : 90 ... 2.3 %
m : 264 ... 6.7 %
n : 488 ... 12.4 %
o : 4 ... 0.1 %
p : 156 ... 4.0 %
q : 276 ... 7.0 %
r : 82 ... 2.1 %
s : 19 ... 0.5 %
t : 183 ... 4.7 %
u : 280 ... 7.1 %
v : 348 ... 8.9 %
w : 1 ... 0.0 %
x : 291 ... 7.4 %
y : 373 ... 9.5 %
z : 95 ... 2.4 %
```

```
yt => 116
tn => 89
mu => 74
nh => 66
nq => 62
hn => 59
vu => 58
vh => 57
qy => 55
xu => 53
nv => 50
up => 47
yn => 47
np => 46
vy => 45
xh => 45
nu => 44
ym => 39
uy => 37
vi => 37
yx => 36
vq => 35
uv => 34
gn => 32
my => 32
av => 31
xz => 30
```

```
ytn => 79
vup => 30
nqy => 22
pyt => 20
mur => 20
ynh => 18
xzy => 16
nhn => 16
nuy => 14
ytv => 14
bxh => 14
gnq => 14
mxu => 14
vii => 13
vyn => 13
uvy => 12
lvq => 12
nvh => 12
tmq => 12
qyt => 12
muv => 11
upy => 11
xhy => 11
vym => 11
lmu => 11
ymu => 11
yxh => 11
tnv => 11
cmu => 11
```

步骤 6：使用频率分析，明文使用大写字母，密文使用小写字母，使用 tr 命令来完成这项工作：

```
[09/22/20]seed@VM:~$ tr 'nptuvy' 'EDHNAT' <in.txt> out_1.txt
[09/22/20]seed@VM:~$ cat out_1.txt
THE xqaAhq TzhN  xN qzNDAd lHmaH qEEcq AgxzT hmrHT AbTEh THmq ixNr qThANrE
AlAhDq Thme THE gArrEh bEEiq imsE A NxNArENAhmAN Txx
```

```
[09/22/20]seed@VM:~$ tr 'mq' 'IS' <out_1.txt> out_2.txt
[09/22/20]seed@VM:~$ cat out_2.txt
THE xSaAhS TzhN  xN SzNDAd lHIaH SEEcS AgxzT hIrHT AbTEh THIS ixNr SThANrE
AlAhDS ThIe THE gArrEh bEEiS iIsE A NxNArENAhIAN Txx
```

```
[09/22/20]seed@VM:~$ tr 'hekcfgx' 'RPXMVBO' <out_2.txt> out_3.txt
[09/22/20]seed@VM:~$ cat out_3.txt
THE OSaARS TzRN  ON SzNDAd lHIaH SEEMS ABOzT RIrHT AbTER THIS iONr STRANrE
AlARDS TRIP THE BArrER bEEiS iIsE A NONArENARIAN TOO
```

```
[09/22/20]seed@VM:~$ tr 'iza' 'LUC' <out_3.txt> out_4.txt
[09/22/20]seed@VM:~$ cat out_4.txt
THE OSCARS TURN  ON SUNDAd lHICH SEEMS ABOUT RIrHT AbTER THIS LONr STRANrE
AlARDS TRIP THE BArrER bEELS LIsE A NONArENARIAN TOO
```

```
[09/22/20]seed@VM:~$ tr 'rsdblj' 'GKYFWQ' <out_4.txt> out_5.txt
[09/22/20]seed@VM:~$ cat out_5.txt
THE OSCARS TURN  ON SUNDAY WHICH SEEMS ABOUT RIGHT AFTER THIS LONG STRANGE
AWARDS TRIP THE BAGGER FEELS LIKE A NONAGENARIAN TOO
```

```
[09/22/20]seed@VM:~$ cat out_6.txt
THE OSCARS TURN  ON SUNDAY WHICH SEEMS ABOUT RIGHT AFTER THIS LONG STRANGE
AWARDS TRIP THE BAGGER FEELS LIKE A NONAGENARIAN TOO
```

分析得到字母加密前后对应关系如下：

a	b	c	d	e	f	g	h	i	j	k	l	m
c	f	m	y	p	v	b	r	l	q	x	w	i

n	o	p	q	r	s	t	u	v	w	x	y	z
e	j	d	s	g	k	h	n	a	z	o	t	u

根据加密单表破译得到明文为：

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：破解维吉尼亚密码

步骤 1：编写分析密文的程序：

```
def findindexkey(subarr):
```

```
    visiable_chars = []
    for x in range(32, 126):
        visiable_chars.append(chr(x))
    # print(vi)
    test_keys = []
    ans_keys = []
    for x in range(0x00, 0xFF):
        test_keys.append(x)
        ans_keys.append(x)
    for i in test_keys:
        for s in subarr:
            if chr(s ^ i) not in visiable_chars:
                ans_keys.remove(i)
                break
    # print(ans_keys)
    return ans_keys

strmi                                                                                =
'F96DE8C227A259C87EE1DA2AED57C93FE5DA36ED4EC87EF2C63AAE5B9A7EFFD673BE4ACF7BE8923C\
AB1ECE7AF2DA3DA44FCF7AE29235A24C963FF0DF3CA3599A70E5DA36BF1ECE77F8DC34BE129A6CF4D126BF\
5B9A7CFEDF3EB850D37CF0C63AA2509A76FF9227A55B9A6FE3D720A850D97AB1DD35ED5FCE6BF0D138A84C\
C931B1F121B44ECE70F6C032BD56C33FF9D320ED5CDF7AFF9226BE5BDE3FF7DD21ED56CF71F5C036A94D96\
3FF8D473A351CE3FE5DA3CB84DDB71F5C17FED51DC3FE8D732BF4D963FF3C727ED4AC87EF5DB27A451D47E\
FD9230BF47CA6BFEC12ABE4ADF72E29224A84CDF3FF5D720A459D47AF59232A35A9A7AE7D33FB85FCE7AF5\
923AA31EDB3FF7D33ABF52C33FF0D673A551D93FFCD33DA35BC831B1F43CBF1EDF67F0DF23A15B963FE5DA\
36ED68D378F4DC36BF5B9A7AFFD121B44ECE76FEDC73BE5DD27AFCD773BA5FC93FE5DA3CB859D26BB1C63C\
ED5CDF3FE2D730B84CDF3FF7DD21ED5ADF7CF0D636BE1EDB79E5D721ED57CE3FE6D320ED57D469F4DC27A8\
5A963FF3C727ED49DF3FFFDD24ED55D470E69E73AC50DE3FE5DA3ABE1EDF67F4C030A44DDF3FF5D73EA250\
C96BE3D327A84D963FE5DA32B91ED36BB1D132A31ED87AB1D021A255DF71B1C436BF479A7AF0C13AA14794'
arr = []
for x in range(0, len(strmi), 2):
    arr.append(int(strmi[x:2 + x], 16))
```

```
for keylen in range(1, 14):
    for index in range(0, keylen):
        subarr = arr[index::keylen]
        ans_keys = findindexkey(subarr)
        print('keylen=', keylen, 'index=', index, 'keys=', ans_keys)
        if ans_keys:
            ch = []
            for x in ans_keys:
                ch.append(chr(x ^ subarr[0]))
            print(ch)
print('################')
import string


def findindexkey2(subarr):
    test_chars = string.ascii_letters + string.digits + ',' + '.' + ' '
    # print(test_chars)
    test_keys = []
    ans_keys = []
    for x in range(0x00, 0xFF):
        test_keys.append(x)
        ans_keys.append(x)
    for i in test_keys:
        for s in subarr:
            if chr(s ^ i) not in test_chars:
                ans_keys.remove(i)
                break
    # print(ans_keys)
    return ans_keys

vigenerekeys = []
for index in range(0, 7):
    subarr = arr[index::7]
    vigenerekeys.append(findindexkey2(subarr))
print(vigenerekeys)

print("##########")
ming = ''
for i in range(0, len(arr)):
    ming = ming + chr(arr[i] ^ vigenerekeys[i % 7][0])
print(ming)
```

步骤 2：运行该程序得到密钥和明文：

```
[[186], [31], [145], [178], [83], [205], [62]]
#########
Cryptography is the practice and study of techniques for, among other things, se
cure communication in the presence of attackers. Cryptography has been used for
hundreds, if not thousands, of years, but traditional cryptosystems were designe
d 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 kn
ow, and this exercise demonstrates, that it can be broken very easily.
```

Task3：破解重复的一次一密密码

步骤 1：编写分析密文的程序：

```
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()
```

步骤 2：运行该程序得到明文：

```
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.
```

实验感想

本次实验对破解单表代换密码、维吉尼亚密码、一次一密密码进行了实际操作，在先前对密码学了解的基础上，加深了对这些概念的理解，对 python 代码进行了研读学习，为今后的加密解密打下了一定的基础。