实 验 报 告

课程名称 信息安全

学　院 软件学院

专　业 软件工程

姓　名 于康

学　号 **20302010040**

开 课 时 间 **2022** 至 **2023** 学年第 二 学期

实验项目名　　称	Cracking Vigenère Cipher	成绩	

一、实验目的

1. 理解 Vigenère Cipher，进行加密和解密
2. 理解并进行 Kasiski Test
3. 理解并进行 Friedman's Index of Coincidence Test
4. 通过频率分析破解 Caesar Cipher

二、实验内容

1. Vigenère Cipher

维吉尼亚密码是一种多字母表密码，是使用一系列凯撒密码组成密码字母表的加密算法。根据实验文档，将字母与数字相映射，用公式表示出维吉尼亚密码的加密与解密过程。

Suppose we have an -character alphabet $A = (a_1, a_2, \dots, a_n)$, an m-character key, $K = (k_1, k_2, \dots, k_m)$ and a t-characterplaintext, $M = (m_1, m_2, \dots, m_t)$.

The alphabet are encoded by integer number, i.g. the letters A – Z are taken to be the numbers 0–25 .

Then, we define a **Vigenère Cipher** :

$$E_k(a_i) = m_i + k_i \ (mod\ n)$$

and

$$D_k(c_i) = c_i - k_i \ (mod\ n)$$

2. Kasiski Test

该测试用于破解出 Vigenère Cipher 中 key 的长度，在密文中相同的片段对（一般长度大于等于 3），很可能它们对应于相同的明文片段，认为它们之间的距离是 key 长度的整数倍，计算每对相同密文片段对之间的距离，并认为所有距离的最大公因数是 key 的长度。

3. Friedman's Index of Coincidence Test

Index of Coincidence（重合指数）是在字符串中随机选择两个字符相同的概率。

$$IC = \frac{\sum_{i=1}^{c} n_i(n_i - 1)}{N(N-1)}$$

密文中的重合可能是由底层明文中的重合引起的，Friedman test 利用重合指数破解 Vigenère Cipher 中 key 的长度。随机英文文本的 IC 总是大约为 0.0385，而一段有意义的英文文本的 IC 总是大

约为 0.0655。

Key 的长度可由以下公式得出：

$$L \approx \frac{k_p - k_r}{\mathrm{IC} - k_r} = \frac{0.027\mathrm{N}}{\mathrm{IC} * (\mathrm{N}-1) - 0.0385\mathrm{N} + 0.0655}$$

三、实验步骤

1. Computing the chance for the false positive rate of Kasiski test

两处相同的长度为 3 的片段对不是相同明文片段的可能性有多大？其中 n=26。

对于该问题的探讨实际上就是对 Kasiski test 可靠性的验证。

根据《*Kasiski's Test: Couldn't the Repetitions be by Accident?*》，通过简化问题，可以将这种片段对的意外重复与生日悖论归为同一类问题，那么对于长度为 r 的字符串，出现长度为 3 的意外重复的概率为 $K(26^3, r)$。继续根据文中边界的计算公式，可得：

$$1 - e^{-\frac{r(r-1)}{2\times 26^3}} \le K(26^3, r) \le \frac{r(r-1)}{2 \times 26^3}$$

举例来说，当 r=100 时，可以由文中表格得到出现长度为 3 的意外重复的概率为 0.246。

2. Implementing Vigenère Cipher

维吉尼亚密码加密解密函数的实现依据文档介绍部分，遍历字符串，分别得到此次循环要处理的字符 c 的 index 以及此时相对应的 key 的字符的 index，加密相加，解密相减，关键代码如下：

```
encrypted_index = (c_index + key_c_index) % len(alphabet)
decrypted_index = (c_index - key_c_index) % len(alphabet)
```

3. Determining the Key Size

补全函数 **kasiski_test** 时，遍历密文字符串，每次选择长度为 3 的片段，若它未在片段数组中则加入，若已在，则将它与该片段第一次出现的距离的距离记录到距离数组。代码如下图所示：

```
for i in range(len(ciphertext) - 2):
    trigraph = ciphertext[i : i+3]
    if trigraph not in trigraphs:
        trigraphs.append(trigraph)
    else:
        first_appear = ciphertext.index(trigraph)
        distance = i - first_appear
        distances.append(distance)
```

补全函数 **index_of_coincidence** 时，遍历字母表，依据 IC 的计算公式补全代码：

```
n = len(ciphertext)
for letter in alpha:
    freq = common[letter]
    ioc += (freq * (freq - 1))
ioc /= (n * (n - 1))
```

4. Determining the Keyword

通过上一步得到了 key 的长度，接下来需要破解 key 中每一位对应的字母，这里依据的是正常文本中不同字母的频率分布差异。在 **crack** 函数中，通过 **make_cosets** 函数将密文字符串按照 key 长度进行分组，然后分别解析每一组所对应的偏移字母。

在需要补充的函数 **find_likely_letters** 中，首先统计该组中不同字母的频率分布，得到 coset_freq 数组，如下所示：

```
    coset_count = Counter(coset)
    for letter in alpha:
        freq = coset_count[letter] / len(coset)
        coset_freq.append(freq)
```

由于维吉尼亚密码是由多个凯撒密码组成，将 coset_freq 与给定的频率分布 eng_freq 进行比较，并利用 **rotate_list** 函数偏移 coset_freq，循环尝试所有可能，计算与 eng_freq 的差异值，如下所示：

```
    for i in range(len(alpha)):
        difference = find_total_difference(coset_freq, eng_freq)
        differences.append(difference)
        coset_freq = rotate_list(coset_freq)
```

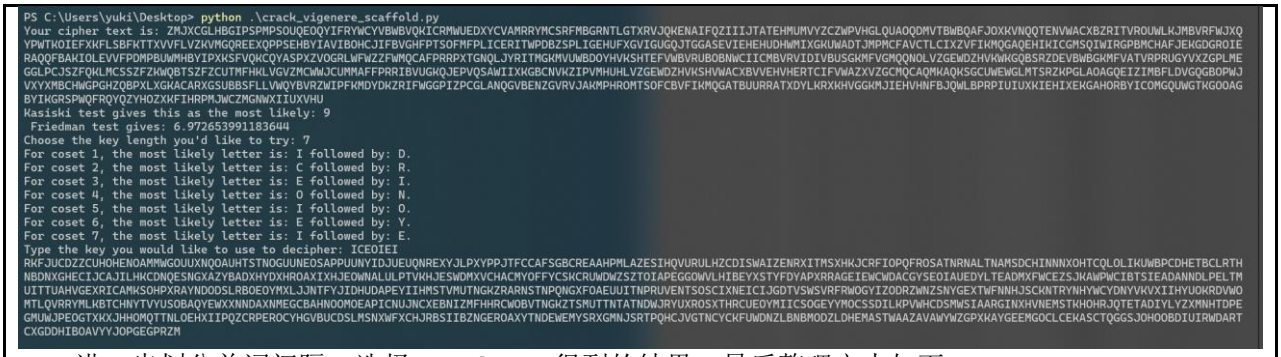
由于给定代码打印了两个可能字母，所以在代码中计算出差异值最小的两种偏移所对应的字母：

```
    firstletter = differences.index(min(differences))
    differences[firstletter] = float('inf')
    secondletter = differences.index(min(differences))

    letter1 = alpha[firstletter]
    letter2 = alpha[secondletter]
    return "the most likely letter is: " + letter1 + " followed by: " + letter2
```

四、实验结果及分析

完成上述实验步骤后，由于学号末位偶数，解除对应文本的注释，之后运行代码，结果如下：

```
PS C:\Users\yuki\Desktop> python .\crack_vigenere_scaffold.py
Your cipher text is: ZMJXCGLHBGIPSPMPSOUQEOQYIFRYWCVYBWBVQKICRMWUEDXYCVAMRRYMCSRFMBGRNTLGTXRVJQKENAIFQZIIIJTATEHMUMVYZCZWPVHGLQUAQQDMVTBWBQAFJOXKVNQQTENVWACXBZRITVROUWLKJMBVRFWJXQ
YPWTWOIEFXKFLSBFKTTXVVFLVZKVMGQREEXQPPSEHBYIAVIBOHCJIFBVGHFPTSOFMFPLICERITWPDBZSPLIGEHUFXGVIGUGQJTGGASEVIEHEHUDHWMIXGKUWADTJMPMCFAVCTLCIXZVFIKMQGAQEHIKICGMSQIWIRGPBMCHAFJEKGDGROIE
RAQQFBAKIOLEVVFPDMPBUWMHBYIPXKSFVQKCQYASPXZVOGRLWFWZZFWMQCAFPRRPXTGNQLJYRITMGKMVUWBDOYHVKSHTEFWBVRUBOBNWCIICMBVRVIDIVBUSGKMFVGMQQNOLVZGEWDZHVKWKGQBSRZDEVBWBGKMFVATVRPRUGYVXZGPLME
GGLPCJSZFQKLMCSSZFZHWQBTSZFZCUTMFHKLVGVZMCWWJCUMMAFFPRRIBVUGHQJEPVQSAWIIXHGBCNVKZIPVMHUHLVZGEWDZHVKSHVWACXBVVEHVHERTCIFVWAZXVZGCMQCAQMWAQKSGCUWEWGLMTSRZKPGLAOAGQEIZIMBFLDVGQGBOPWJ
VXYXMBCHWGPGHZQBPXLXGKACARXGSUBBSFLLVWQYBVRZWIPFKMDYDKZRIFWGGPIZPCGLANQGVBENZGVRVJAKMPHROMTSOFCBVFIKMQGATBUURRATXDYLKRXKHVGGKMJIEHVHNFBJQWLBPRPIUIUXKIEHIXEKGAHORBYICOMGQUWGTKGOOAG
BYIKGRSPWQFRQYQZYHOZXKFIHRPMJWCZMGNWXIIUXVHU
Kasiski test gives this as the most likely: 9
 Friedman test gives: 6.972653991183644
Choose the key length you'd like to try: 9
For coset 1, the most likely letter is: D followed by: S.
For coset 2, the most likely letter is: I followed by: X.
For coset 3, the most likely letter is: R followed by: V.
For coset 4, the most likely letter is: E followed by: R.
For coset 5, the most likely letter is: C followed by: N.
For coset 6, the most likely letter is: T followed by: X.
For coset 7, the most likely letter is: I followed by: M.
For coset 8, the most likely letter is: O followed by: Z.
For coset 9, the most likely letter is: N followed by: Y.
Type the key you would like to use to decipher: DIRECTION
WESTANDTODAYONTHEBRINKOFAREVOLUTIONINCRYPTOGRAPHYTHEDEVELOPMENTOFCHEAPDIGITALHARDWAREHASFREEDITFROMTHEDESIGNLIMITATIONSOFMECHANICALCOMPUTINGANDBROUGHTTHECOSTOFHIGHGRADECRYPTOGRAPH
ICDEVICESDOWNTOWHERETHEYCANBEUSEDINSUCHCOMMERCIALAPPLICATIONSASREMOTECASHDISPENSERSANDCOMPUTERTERMINALSINTURNSUCHAPPLICATIONSCREATEANEEDFORNEWTYPESOFCRYPTOGRAPHICSYSTEMSWHICHMINIM
IZETHENECESSITYOFSECUREKEYDISTRIBUTIONCHANNELSANDSUPPLYTHEEQUIVALENTOFAWRITTENSIGNATUREATTHESAMETIMETHEORETICALDEVELOPMENTSININFORMATIONTHEORYANDCOMPUTERSCIENCESHOWPROMISEOFPROVID
INGPROVABLYSECURECRYPTOSYSTEMSCHANGINGTHISANCIENTARTINTOASCIENCETHEDEVELOPMENTOFCOMPUTERCONTROLLEDCOMMUNICATIONNETWORKSPRONSESESEFFORTLESSANDINEXPENSIVECONTACTBETWEENPEOPLEORCOMPUTE
RSONOPPOSITESIDESOFTHEWORLDREPLACINGMOSTMAILANDMANYEXCURSIONSWITHTELECOMMUNICATIONSFORMANYAPPLICATIONSTHESECONTACTSMUSTBEMADESECUREAGAINSTBOTHEAVESDROPPINGANDTHEINJECTIONOFILLEGIT
IMATEMESSAGESATPRESENTH

PS C:\Users\yuki\Desktop>
```



进一步划分单词间隔，选择 Kasiski test 得到的结果，最后整理文本如下：

we stand today on the brink of a revolution in cryptography.

the development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals.

in turn such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature.

at the same time theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems changing this ancient art into a science.

the development of computer controlled communication networks pronses effortless and inexpensive contact between people or computers on opposite sides of the world replacing most mail and many excursions with telecommunications.

for many applications these contacts must be made secure against both eavesdropping and the injection of illegitimate messages.

at present h

在网络上可以查找到论文原文，其标题为 ***New Directions in Cryptography***。

五、实验总结

该实验进行了对于维吉尼亚密码的破解。

首先通过了解加密解密过程，知道了它是一种多字母表密码。

破解过程分为两步，确定 key 的长度和确定 key 中的每个字母。

确定 key 的长度可以使用两种途径，Kasiski test 和 Friedman test。并且对 Kasiski test 的可行性进行了计算验证，对重合指数 IC 的计算也有了一定了解。

确定 key 中每个字母，利用了正常英文文本中不同字母的频率分布不同，将密文进行分组，对每一组分别进行分析，确定该组的偏移字母，最终得到完整的 key。

最后，通过 key 可以将密文解密成明文。