编号：PB23051007

作品类别：☑软件设计 ☐硬件制作 ☐工程实践

2025年春季学期《密码学导论》作品设计报告

题目：单表代换辅助工具

作者： 付佳峻

学号： PB23051007

2025年 3 月 2 1 日

中国科学技术大学网络空间安全学院

基本信息表

编号：CACR20XXxxxxxx

作品题目：单表代换密文破译辅助工具

作品类别：☑软件设计 ☐硬件制作 ☐工程实践

作品内容摘要：
本作品围绕密码学中经典的古典密码学的单表代换加密解密的知识点展开，对唯密文攻击的本质做了详细的分析，主要实现功能包括单表代换加解密过程复现，以及结合密文特征进行频率分析、上下文分析根据动态建议逐步破译。
作品特色：
1）特色：密文（英文）的特征提取，并对多组特征进行频率分析
2）特色：可视化特征分布关系
3）特色：动态建议，根据密文与动态词典的匹配度给出较精确的建议

关键词：单表代换、唯密文攻击、频率分析、上下文分析

1 第一章 - 作品概述

1.1 引言

本作品旨在搭建一个古典密码学的单表代换辅助破译工具，密文示例仅适用于26个字母一一映射对应，故使用者在作品给出动态建议之外，还应结合自身的英语知识储备，在逐步破译过程中，对具有一定可读性的明文，自主破译单词映射对e.g.(chilxren->children)。

主要实现功能：

1.单表代换的加解密：

我们基于古典密码的凯撒密码原理，对需要明文P进行加密，C=E（K，P），E为加密函数，这里我们采用c[i]=(p[i]+k[i])mod26,所以我们要求密钥的长度一定要与明文长度相等，同时也能增加唯密文攻击的复杂性。解密即是加密的逆过程，p[i]=(c[i]-k[i])mod26.

2.初始化字母映射关系：

由Wikipedia搜索对英文中常见的字母和单词频率分析，我们可以构建特征：

1）如果出现单个字母时，则其一定映射到英文中的量词：a

2）基于对密文的高频字母频率分析，我们可以得出最高频字母必对应英文字母e

3）基于出现最多的三字母组合e.g.ozn->the,可以更新此映射关系

3.查看频率分析图表：

根据analyze_frequency提供的频率分析数据，将单字母、二字母介词、三字母组合出现最多的情况以可视化的形式展示出来

4.获取动态破译建议：

该模块先通过长度筛选获取可能相关的候选词（动态词典中的常见字母组合），逐个分析候选词中是否含有与已有映射相关的字母，且位置维持不变，基于此我们将计算匹配程度，进一步生成有效建议。

5.更新映射关系（逐步破译）：

基于功能4的前五次交互建议，进行逐步更新映射（在第五次交互之后，待完成破译的明文已具有较高的可读性），我们可以根据自身能力储备依次交互剩下的字母映射关系。

6.逐行打印明文和密文对：

该功能旨在对前面的功能实现做进一步的验证，让使用者能有更好的体验。

1.单表代换的加解密：

明文 P 进行加密过程中，C=E（K，P），E 为加密函数，这里我们采用 c[i]=(p[i]+k[i])mod26,解密即是加密的逆过程，在 python 语言中，我们即对字母的 ASCII 码做整数运算即可。encrypt_char=chr((ord(plain_text[i])-ord('a')+ord(key[i])-ord('a'))%26+ord('a'))，解密反之亦然

```
def encrypt(self,plain_text,key):
    encrypt_chars=[]
    for i in range(len(plain_text)):
        if plain_text[i].isalpha():
            encrypt_char=chr((ord(plain_text[i])-ord(
            encrypt_chars.append(encrypt_char)
        else:
            encrypt_chars.append(plain_text[i])
```

2.初始化字母映射关系：

由 Wikipedia 搜索对英文中常见的字母和单词频率分析，我们可以从密文中提取特征：

```
def analyze_frequency(ciphertext):#计算特征出现的频次
    words=words = re.findall(r'\b[a-zA-Z]+\b', ciphertext.lower())
    single_letter=Counter()
    binary_letter=Counter()
    triple_letter=Counter()
    unit_letter=Counter()
    prepositions=Counter()
    for word in words:
        if len(word)==1:
            unit_letter[word]+=1
        if len(word)==2:
            prepositions[word]+=1
        single_letter.update(word)
        for i in range(len(word)-1):
            binary_letter[word[i:i+2]]+=1
        for i in range(len(word)-2):
            triple_letter[word[i:i+3]]+=1
```

3.查看频率分析图表：

根据 analyze_frequency 提供的频率分析数据，将单字母、二字母介词、三字母组合出现最多的情况以可视化的形式展示出来

```python
#对不同特征进行可视化
import matplotlib.pyplot as plt
解释代码 | 注释代码 | 生成单测 | ×
def visualize_frequency(results):
    # 单字母频率可视化
    plt.figure(figsize=(15, 5))
    plt.subplot(1, 4, 1)
    single_letters, single_counts = zip(*results['single'])
    plt.bar(single_letters, single_counts)
    plt.title('Single Letter Frequency')
    plt.xlabel('Letters')
    plt.ylabel('Frequency')
```

4. 获取动态破译建议：

该模块先通过长度筛选获取可能相关的候选词（动态词典中的常见字母组合），

```python
DYNAMIC_DICT = {
    2: ["of", "in", "on", "at", "to", "is", "it", "he", "by", "as"],
    3: ["the", "and", "ing", "ion", "ent", "her", "tha", "for", "not", "you"],
    4: ["that", "this", "with", "have", "from", "they", "will", "your", "than", "them"],
    5: ["there", "which", "could", "other", "their", "about", "would", "after", "first", "these"],
}
```

逐个分析候选词中是否含有与已有映射相关的字母，且位置维持不变，基于此我们将计算匹配程度，进一步生成有效建议。

```python
# 逐字符检查映射兼容性
for i in range(word_len):
    cipher_char = word[i]   # 明文字符（来自破译中间结果）
    plain_char = cand[i]    # 候选词字符

    # 条件1：若候选字符已被映射，必须与当前密文字符一致
    if plain_char in reversed_mapping:
        if plain_char != cipher_char:
            conflict = True
            break
    # 条件2：记录需要的新映射
    else:
        required_mappings[cipher_char] = plain_char
```

5. 更新映射关系（逐步破译）：

```
cipher_char, plain_char = cmd.split()
mapping[cipher_char] = plain_char
print(f"已更新映射：{cipher_char} -> {plain_char}")
print("\n当前解密预览：")
print(decrypt(ciphertext, mapping))
print("无效格式！请按格式输入，例如：z h")
```

基于功能 4 的前五次交互建议，进行逐步更新映射（在第五次交互之后，待完成破译的明文已具有较高的可读性），我们可以根据自身能力储备依次交互剩下的字母映射关系。

3. 第三章-系统测试与结果

3.1 交互式界面：

```
****************** 密码分析交互界面 ******************
1.************单表代换加解密分析***************
2. ************显示当前单表映射关系************
3. ************查看频率分析图表************
4. ************获取破译建议************
5. ************输入替换规则（例：'z h'）
6. ************打印密文************
7. ************破译完成，退出界面************
请选择操作 (1-7):
```

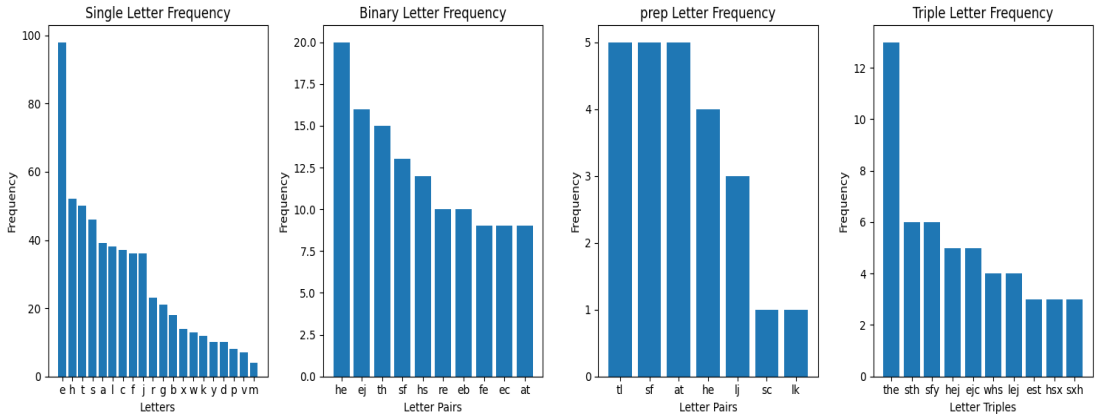
每次交互后，可以按照上述界面的顺序依次完成一组字母映射的更新，直至明文阅读没有错误。

3.2 单表代换加解密分析:

```
**请输入需要加密的明文和密钥,密钥长度请与明文长度相同**
明文: hello world
密钥: abcde fghij
密文: hfnos buytm

需要解密吗[y/n]:
y
明文：hello world
```

3.3 查看频率分析图表：



上述可视化分布图展示了第一次交互前密文初始化后单字母、双字母组合、介词组合、三字母组合的频率分布，从中我们可以验证：the 为英文中出现次数最多的三字母，e 为出现字母最多的字母。以上的分布图为我们接下来的提供建议建立了较为直观的支撑

3.4 获取破译动态建议：

```
动态破译建议（基于明文片段）：
'HADE' 匹配 'HAVE' (匹配度3/4) | 需映射: d→v
'LTHEJ' 匹配 'OTHER' (匹配度3/5) | 需映射: l→o, j→r
'WSTH' 匹配 'WITH' (匹配度2/4) | 需映射: w→w, s→i
'WHSXH' 匹配 'WHICH' (匹配度2/5) | 需映射: w→w, s→i, x→c
'FLT' 匹配 'NOT' (匹配度1/3) | 需映射: f→n, l→o
'AFB' 匹配 'AND' (匹配度1/3) | 需映射: f→n, b→d
```

3.5 更新映射关系：

3.5.1 第一次交互：

```
已更新映射：d -> v

当前解密预览：
hhsreac klyy wac flt mflwf tl have esthej wske lj xhsrbjef,
whsxh gav hahhef tl the glct hlfect helhre; esthej jeratsvec
lj feaj kjsefbc, whsxh sc xejtasfrv glje efecear. he rsveb
arlfe sf hsc hlece sf cavsrre jlw, whsthej flfe hefetjateb.
a csfyre blgectsx cekksxeb tl cejve hsg. he pjeamkacteb afb
bsfeb at the xrep, at hlejc gathegatsxarrv ksaeb, sf the cage
```

3.5.2 第五次交互：

在根据以上建议的提示后，我们依次对[d v],[l o],[j r],[w w],[s i],[x c],[f n],[b d]更新后，可以得出以下的待完成的明文：

```
hhireas koyy was not mnown to have either wike or chirdren,
which gav hahhen to the gost honest heohre; either reratives
or near kriends, which is certainrv gore enesear. he rived
arone in his hoese in savirre row, whither none henetrated.
a sinyre dogestic sekkiced to serve hig. he preamkasted and
dined at the crep, at hoers gathegaticarrv kiaed, in the sage
roog, at the sage tapre, never taminy his gears with other
gegpers, gech ress prinyiny a yeest with hig; and went hoge
at eaactrv gidniyht, onrv to retire at once to ped. he never
esed the cosv chagpers which the rekorg hrovides kor its
kavoered gegpers. he hassed ten hoers oet ok the twentv-koer
in savirre row. either in sreehiny or taminy his toiret
```

我们发现，现在的密文已经具有较高的可读性。

比如：[chirdren->children]，[wike->wife]，[iny->ing]，[reratives->relatives]，[kriends->friends]，[hoese->house]，[roog->room]，[mnown->known]，[hrovides->provides],然后更新[r l],[k f],[e u],[g m],[m k],[h p]……

进一步交互后，最终破译结果为

```
phileas fogg was not known to have either wife or children,
which may happen to the most honest people; either relatives
or near friends, which is certainly more unusual. he lived
alone in his house in saville row, whither none penetrated.
a single domestic sufficed to serve him. he breakfasted and
dined at the club, at hours mathematically fixed, in the same
room, at the same table, never taking his meals with other
members, much less bringing a guest with him; and went home
at exactly midnight, only to retire at once to bed. he never
used the cosy chambers which the reform provides for its
favoured members. he passed ten hours out of the twenty-four
in saville row. either in sleeping or taking his toilet
```

为了让使用者能对映射关系有进一步了解，我们将明文和密文对一行一行输出，见下图

```
密文（C）与明文（P）逐行对比：
第00行 C: hzsrnqc klyy wqc flo mflwf ol zqdn nsoznj wskn lj xzsrbjnf,
       P: phileas fogg was not known to have either wife or children,

第01行 C: wzsxz gqv zqhhnf ol ozn glco zlfnco hnlhrn; nsoznj jnrqosdnc
       P: which may happen to the most honest people; either relatives

第02行 C: lj fnqj kjsnfbc, wzsxz sc xnjoqsfrv gljn efeceqr. zn rsdnb
       P: or near friends, which is certainly more unusual. he lived

第03行 C: qrlfn sf zsc zlecn sf cqdsrrn jlw, wzsoznj flfn hnfnojqonb.
       P: alone in his house in saville row, whither none penetrated.
```

5 第五章 - 结论

本作品完成了单表代换的加解密过程，辅助破译了单表代换的密文，本次实验采用唯密文攻击的方式，根据动态建议逐步更新字母映射关系，完成了密文破译项目。

程序旨在帮助初学者进一步深入了解古典密码学中的单表代换破译的知识，故使用该程序者根据静态建议和动态建议不断更新明文密文对的字母映射关系，从宏观角度分析破译过程中明文的特征，反复迭代直至完成破译。

项目优化方向：对上下文分析机制做进一步的优化，采用模拟退火、n-gram模型对明文做可读性的度量，减少人工更新的劳动力，笔者初步设想从自注意力机制的角度对明文映射关系进一步的预测。

项目的局限：由于本次密文仅针对单表代换的破译机制，作品对多表代换以及其他非对称密码没有很强的泛化性。