古典密码的分析——Playfair密码破译

课程名称：**信息安全**

课程代码：**SOFT130018.01**

任课教师：**李景涛**

实验课助教：杨鹏宇 24210240377@m.fudan.edu.cn

张安琪 24212010048@m.fudan.edu.cn

实验目的

- 了解古典密码中的加密和解密运算
- 了解古典密码体制
- 掌握古典密码的破译方法

实验原理

Playfair密码

提高单字母表密码安全性的思路之一

加密

以 FUDAN 为密钥，举例说明 playfair 密码的加密过程：

$$\begin{bmatrix} F & U & D & A & N \\ B & C & E & G & H \\ I/J & K & L & M & O \\ P & Q & R & S & T \\ V & W & X & Y & Z \end{bmatrix}$$

1. **构建加密矩阵**

   playfair 加密算法基于一个 5*5 的字母矩阵，该矩阵使用一个关键词（密钥）构造，方法是按约定的顺序（例如从左到右、从上到下），依次填入关键词的字母（去除重复字母）后，将字母表其余字母按原来的先后次序填入。

2. **整理明文**

   - 若明文出现相同字母在一组，则在重复的明文字母中插入一个填充字母(eg:x)进行分隔后重新分组(eg: balloon 被重新分组为 ba lx lo on)。
   - 若分组到最后一组时只有一个字母，则补充字母 x。

3. **编写明文**

   - 若明文字母在矩阵中同行，则循环取其右边下一个字母为密文(矩阵最右边的下一个是最左边的第一个)(eg: an 被加密为 NF)。
   - 若明文字母在矩阵中同列，则循环取其下边下一个字母为密文(矩阵最下边的下一个是最上边的第一个)(eg: cq 被加密为 KW)。
   - 若明文字母在矩阵中不同行不同列，则取其同行且与同组另一字母同列的字母为密文(eg: hs 被加密为 GT，fm 被加密为 AI 或 AJ)。

解密过程与加密过程相反

实验环境

C/C++

实验内容

1. 补全Playfair密码破译程序`playfaircrack.c`，包括`playfairDecipher`和`playfairCrack`函数，功能如下：

   `playfairDecipher`:解密函数，根据密钥解密密文，输入密文和密钥，输出明文(密文中的J用I代替)

   `playfairCrack`:模拟退火函数，输入密文和初始密钥，输出迭代过程中最好的结果

   具体参数功能参考`playfaircrack.c`中的注释说明

2. 编译运行`playfaircrack.c`文件，判断程序输出

   ```
   gcc -O3 -lm playfaircrack.c scoreText.c -o your_name
   ./your_name
   ```

3. 完成实验报告

文件结构

`playfaircrack.c`：

- `main`:主函数，重复调用退火函数，输出得分最高的明文
- `playfairDecipher`:解密函数
- `playfairCrack`:模拟退火函数
- `modifyKey`:修改密钥函数，被退火函数调用，不断寻找更好的密钥

`scoreText.c`

- `scoreTextQgram`:评估函数，输入解密得到的明文，评估其与真英文文本的相似度，得分越高，越相似

`verify.py`：

- 用于验证解密得出的明文，确保明文中包含分割重复字符添加的**'X'**

模拟退火算法

模拟退火是一种有效破译`Playfair`密码的方法，是**爬坡算法**的改进。

爬坡算法首先生成一个随机的密钥，解密得到伪明文，最后进行适应度评估（即判断明文与真正英文文本的相似度，值越大，越可能是真明文），而后修改密钥，重复此过程，直到适应度在一定次数内不再提升。爬坡算法的问题在于算法可能**陷入到极大值中**，无法得到真正最大值。

为了解决这个问题，模拟退火算法选择接受某些**不增加**适应度的密钥，以跳出极大值陷阱。退火算法使用**“温度变量”T**来计算接受不增加适应度的密钥的可能性，T会逐渐减小，这一概率也会降低，具体过程如下：

```
1. Generate a random key, called the 'parent', decipher the ciphertext
    using this key.
2. Rate the fitness of the deciphered text, store the result(call function
scoreTextQgram).
```

```
3. for(TEMP = 20;TEMP >= 0; TEMP = TEMP - STEP)
      for (count = 10,000; count>0; count--)
         Change the parent key slightly (e.g. swap two characters in the
           key at random,call function modifyKey) to get child key,
         Measure the fitness of the deciphered text using the child key
         set dF = (fitness of child - fitness of parent)
         If dF > 0 (fitness of child is higher than parent key),
             set parent = child
         If dF < 0 (fitness of child is worse than parent),
             Get a random value temp from 0 to 1
             if temp < e^(dF/T)
                set parent = child
```

tips：`main`函数会循环调用`playfairCrack`函数进行退火操作，除第一次使用初始密钥作为`playfairCrack`的输入密钥，其余`playfairCrack`的输入密钥是之前得到的最佳密钥。程序重复进行退火操作，需要人为判断输出的伪明文是否为真正的文本。

相似度

破解密码时，会尝试使用不同的密钥进行解密，然后查看得到的文本，如果文本与英语非常相似，我们认为这是一把好密钥。需要一种方法来确定文本是否与英语相似，在本次实验中使用**四字母组统计方法**，例如，文本`ATTACK`中的四字母组是:`ATTA`，`TTAC`和`TACK`。
使用这种方式，首先需要知道英语中四字母组的概率，这一概率已在`qgr.h`给出。
为了计算一段文本是英语的概率，首先提取所有的四字母组，然后乘以每个四字母组的概率。如：文本`ATTACK`，四边形是`ATTA`、`TTAC`和`TACK`，总概率为*p(ATTACK) = p(ATTA) x p(TTAC) x p(TACK)*。当将许多小概率相乘时，浮点数中可能出现数值下溢。因此，对每个概率取对数。因此最终的对数概率是*log (p(ATTACK)) = log (p(ATTA)) +log (p(TTAC))+log (p(TACK))*，这个对数概率被用作一段文本的**相似度**，更高的数值意味着它更有可能是英语。

实验提交

截止日期：**2025 年 3 月 23 日**

提交清单：

- 实验报告，文件名格式：学号-姓名-lab1-1；
- 项目源代码，`playfaircrack.c`；
- 可执行程序，编译后的程序；

提交方式：

将提交清单中所有文件打包成一个**压缩文件**（文件名：学号-姓名-lab1-1），在**elearning** 上进行提交。

拓展实验

仅使用暴力穷举的方法，不实用模拟退火，即使用双字母词频分析法破译Playfair密码，即分别统计明文和密文双字母出现的频率，而后相互对应，得到双字母替代表和密钥，破译密文。完成该部分实验需要较大数量的密文，具体方式请同学自行探索唯密文攻击场景下所需的密文数量。通过举例分析或程序实现，完成此部分内容，随基础实验一起提交相应文档。

此部分分数规则如下：

	5分	10分	15分
阐述思路正确、清晰	✔		

	5分	10分	15分
举例说明，用例完整		✔	
代码实现			✔

本实验分数规则如下【扩展实验为15分，基础实验为85分】：

源代码可编译运行	✔	✔	✔	✔	✔	✔	
源代码风格良好	✔	✔		✔			
程序运行结果正确	✔	✔	✔	✔	✔		
实验报告规范清晰	✔	✔	✔			✔	✔
扩展实验	✔						
最终得分	100	85	75-84	65-74	45-64	25-44	5-24