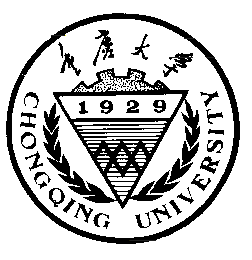
****



S-DES算法实现开发手册

|  |  |  |
| --- | --- | --- |
| 组员姓名 | 学号 | 班级 |
| 吴安婷 | 20221706 | 大数据1班 |
| 花蕊 | 20221702 | 大数据2班 |
| 殷宇欣 | 20221892 | 大数据1班 |

课程名称 信息安全导论

设计项目 S-DES算法实现

课程教师 向宏

开发起止日期 2024.9.24-2024.10.7

**大数据与软件学院**

目 录

[1、引言](#_Toc17696) 2

[1.1 背景 2](#_Toc2525)

[1.1.1项目背景 2](#_Toc11372)

[1.1.2成员介绍 2](#_Toc11372)

[1.1.3分工描述 2](#_Toc11372)

[1.2 术语及说明 2](#_Toc6584)

[2、设计概述 3](#_Toc17696)

[2.1 功能需求 3](#_Toc21108)

[2.2 设计方法和开发工具 3](#_Toc8644)

[2.2.1设计方法](#_Toc32640) 3

[2.1.2开发工具](#_Toc11828) 3

[3、系统详细需求分析 4](#_Toc9341)

[3.1系统运行环境和接口需求分析 4](#_Toc11970)

[3.2安全需求分析 4](#_Toc30942)

[4、总体方案确定](#_Toc3247) 4

[4.1 系统总体结构](#_Toc1202) 4

[4.2 系统详细界面](#_Toc7198) 4

[4.2.1基本测试界面 4](#_Toc32640)

[4.2.2交叉测试界面](#_Toc11828) 5

[4.2.3扩展功能界面](#_Toc32640) 7

[4.2.4暴力破解界面](#_Toc11828) 8

[4.2.5封闭测试界面](#_Toc32640) 9

**1 引言**

**1.1 背景**

**1.1.1项目背景**

在数字化时代，信息安全已成为全球关注的焦点。数据加密技术作为保护信息不被未授权访问的关键手段，其重要性不言而喻。对称加密算法因其高效性和广泛的应用场景，一直是信息安全领域的研究热点。DES（数据加密标准）作为一种广泛使用的对称加密算法，虽然在实际应用中已逐渐被更安全的算法所取代，但其基本原理和结构仍然是学习密码学不可或缺的一部分。

S-DES（简化数据加密标准）算法是DES的一个简化版本，它保留了DES的核心思想，但在设计上更为简洁。在这个背景下，我们将设计一个安全可靠的S-DES算法的加解密程序，以满足用户需求，增强服务体验。

**1.1.2成员介绍**

|  |  |  |
| --- | --- | --- |
| 成员 | 学号 | 性别 |
| 吴安婷 | 20221706 | 女 |
| 花蕊 | 20221702 | 女 |
| 殷宇欣 | 20221892 | 女 |

**1.1.3分工描述**

吴安婷：负责基本测试、扩展功能、暴力破解、封闭测试及GUI界面的开发美化和代码的最终完善处理。

花蕊：负责GUI界面及基本测试的开发，撰写用户指南和测试报告的相关内容。

殷宇欣：负责交叉测试、暴力破解及GUI界面的开发，撰写开发手册的相关内容。

**1.2 术语及说明**

对称加密算法：加密和解密使用相同密钥的加密算法。

密钥长度：密钥的位数，本项目中为10-bit。

分组长度：数据被分割成固定长度的块，本项目中为8-bit。

初始置换（IP）：加密前对数据块进行的第一次置换。

最终置换（IP^{-1}）：解密后对数据块进行的最后一次置换。

密钥扩展：将原始密钥扩展为多个子密钥的过程。

轮函数（F）：加密过程中对数据块进行的一系列变换。

S-Box：一种非线性替换函数，用于将输入的比特转换为输出的比特。

P-Box：置换盒，用于重新排列数据比特的顺序。

SW：交换函数，用于交换数据块的某些部分。

左移位：对数据或密钥进行的循环左移操作。

ASCII编码：一种字符编码标准，用于将字符转换为数字代码。

暴力破解：尝试所有可能的密钥组合以找到正确的密钥。

多线程：同时执行多个线程以提高程序的执行效率。

GUI：图形用户界面（Graphical User Interface），指通过图形方式显示的用户界面，用户可以通过图形元素进行交互操作。

**2 设计概述**

**2.1 功能需求**

（1）算法实现：

密钥扩展：根据给定的密钥生成两个子密钥。

初始和最终置换：使用IP和IP^{-1}置换盒对数据进行置换。

轮函数F：实现轮函数F，包括S-Box和P-Box的逻辑。

加密和解密流程：根据S-DES算法描述，实现加密和解密的流程。

（2）GUI设计：

输入界面：设计输入界面，允许用户输入明文和密钥。

输出界面：设计输出界面，显示加密后的密文或解密后的明文。

操作按钮：提供加密、解密、随机生成密钥和暴力破解按钮。

1. 编程实践：

变量命名：使用有意义的变量名，遵循驼峰命名法或下划线分隔命名法。

代码注释：为复杂的代码块添加清晰、有意义的注释。

函数式编程：将代码模块化，每个函数负责特定的功能实现。

1. 测试策略：

基本测试：测试程序是否能够正确加密和解密8-bit数据。

交叉测试：确保不同小组编写的程序能够相互加密和解密。

扩展功能测试：测试ASCII编码字符串的加密和解密功能。

暴力破解测试：测试暴力破解功能，记录破解时间和效率。

封闭测试：分析密钥空间，探讨密钥的唯一性。

**2.2 设计方法和开发工具**

**2.2.1 设计方法**

（1）加密和解密功能：

实现S-DES算法的加密和解密过程，支持8-bit数据输入和10-bit密钥输入。

（2）用户交互：

通过GUI接收用户输入的明文和密钥，并显示加密后的密文或解密后的明文。

（3）ASCII编码支持：

扩展功能以支持ASCII编码字符串的加密和解密。

（4）暴力破解：

实现暴力破解功能，尝试所有可能的密钥组合以找到正确的密钥。同时支持多线程以便在暴力破解中使用多线程来提高破解效率。

（5）封闭测试：

分析密钥空间，探讨是否存在多密钥加密到同一密文的情况。

**2.2.2 开发工具**

编程语言：python

开发工具：Pycharm，Anaconda

**3 系统详细需求分析**

**3.1 系统运行环境和接口需求分析**

（1）系统需要在以下环境中运行：

操作系统：Windows

Python：使用Python语言结合Tkinter库来实现GUI

1. 系统接口需求：

GUI接口：使用库构建GUI界面，包括输入密钥等用户交互组件，以及按钮触发事件。

**3.2 安全需求分析**

（1）加密强度：

确保S-DES算法实现的加密强度足够，防止简单破解。

（2）多线程安全：

在实现暴力破解功能时，确保多线程操作不会引入竞态条件或死锁

**4 总体方案确定**

**4.1 系统总体结构**

用户界面：负责与用户的交互，包括输入明文、密钥、密文和显示结果。

加密、解密：实现S-DES算法的加、解密过程。

ASCII模式：处理ASCII编码字符串的输入和输出。

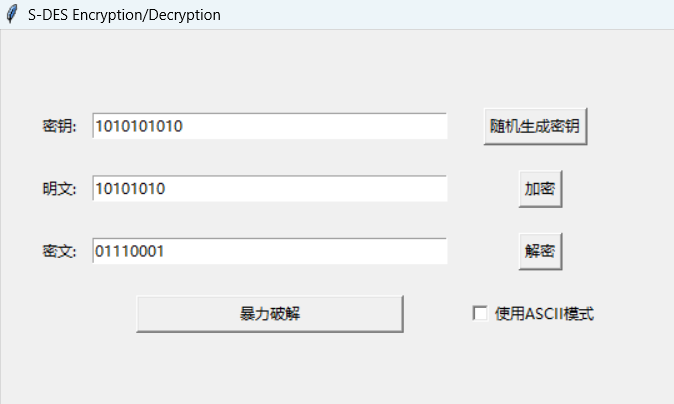
暴力破解：实现暴力破解功能，尝试所有可能的密钥组合。

随机生成密钥：

**4.2 系统详细界面**

**4.2.1 基本测试界面**

加密：输入10bit的密钥和8bit的明文，输出的是8bit的密文。



解密：输入10bit的密钥和8bit的密文，输出的是8bit的明文。



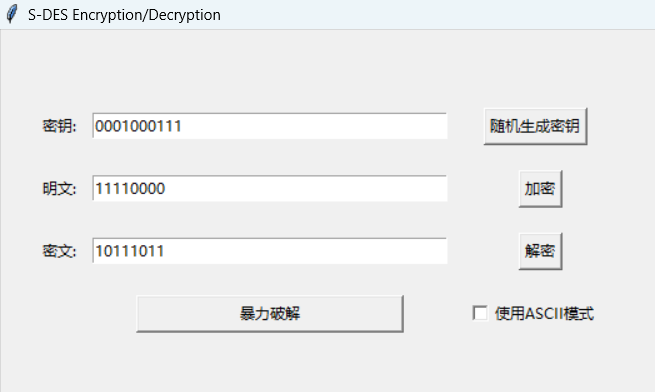
**4.2.2  交叉测试界面**

**二进制：**

我方解密结果：密钥：0001000111

密文：10111011

求得明文：11110000



对方加密结果：密钥：0001000111

明文：11110000

求得密文：10111011



**ASCII字符串：**

我方解密结果：密钥：1100110100

密文：ùÂg

求得明文：CQU



对方加密结果：密钥：1100110100

明文：CQU

求得密文：ùÂg



**4.2.3  扩展功能界面**

加密：输入的明文为字符串，对应的加密输出的可以是ACSII字符串或乱码。

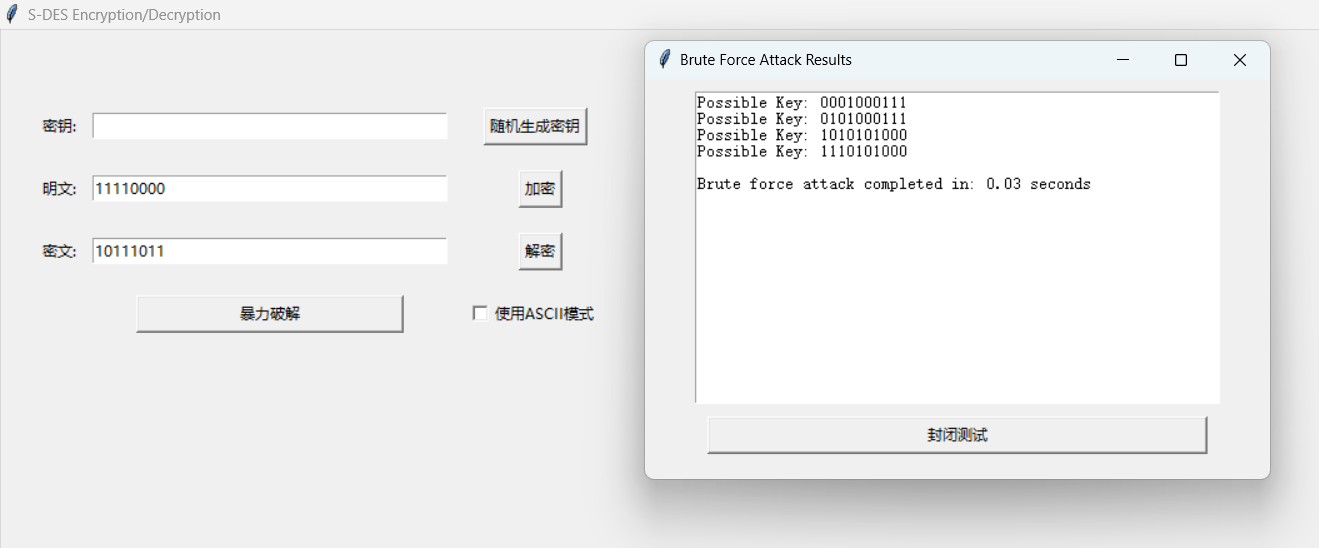


解密：输入的密文为字符串，对应的解密输出的可以是ACSII字符串或乱码。

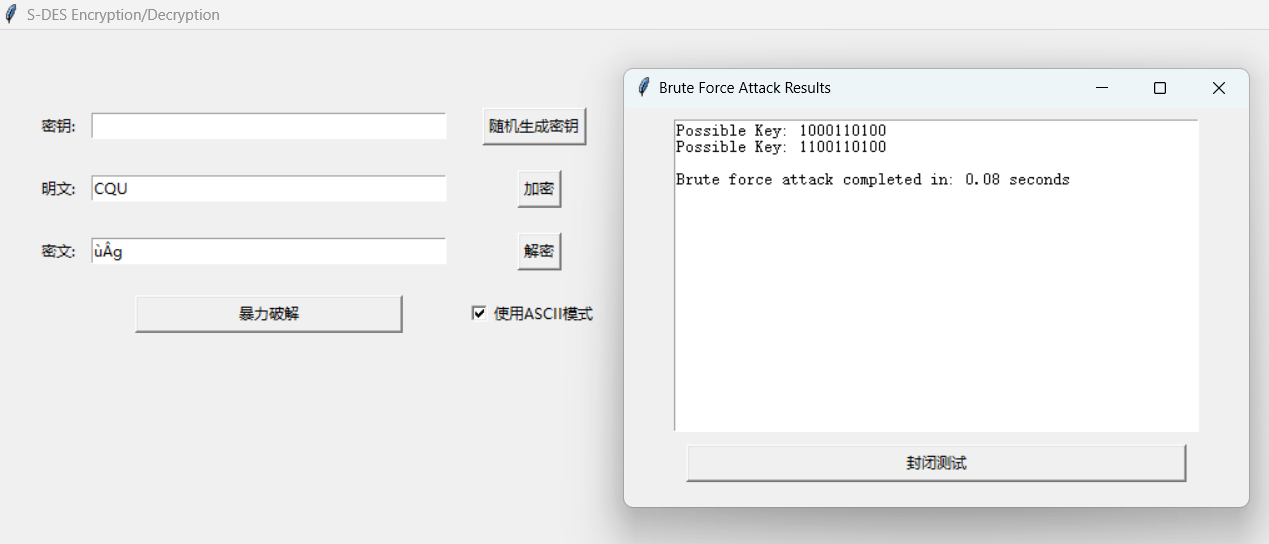


**4.2.4  暴力破解界面**

二进制：

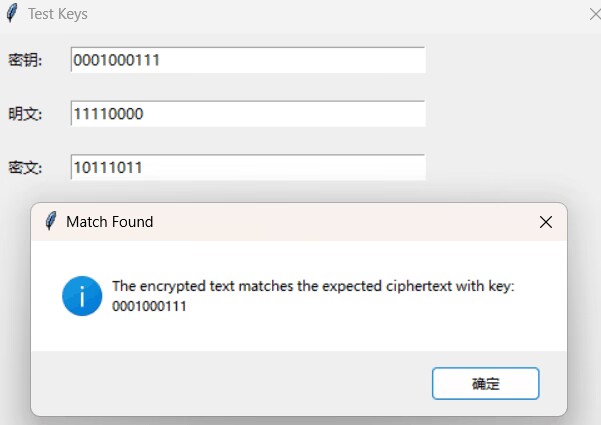


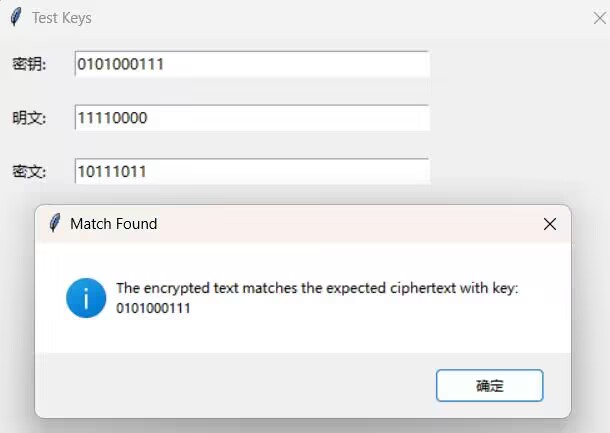
ACSII字符串：

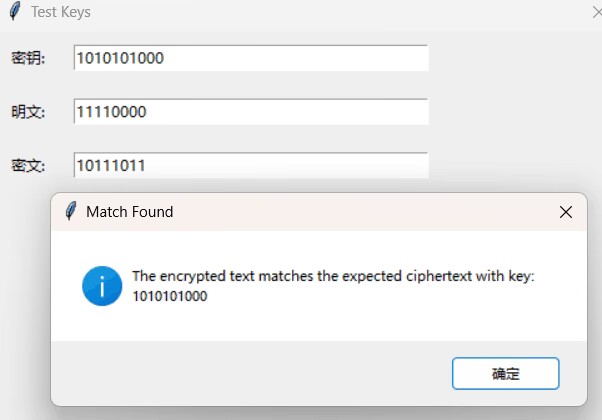


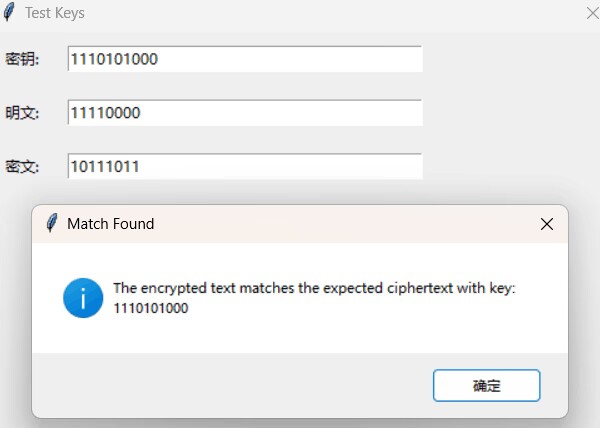
**4.2.5  封闭测试界面**

二进制：









ASCII字符串：

