**项目名称**

**介绍：**

S-AES（Simplified Advanced Encryption Standard）是一种轻量级对称密钥加密算法，旨在提供基本数据保护功能。它采用固定大小的数据块，执行基本的替代和置换步骤，支持不同的运算模式，但由于其轻量级性质，不适合处理高度敏感的机密数据。通常用于教育和研究，以帮助理解对称密钥加密的基本概念和算法。如果需要更高级的数据安全性，通常会选择更复杂的加密标准，如 AES。**项目的目的和范围**

本项目旨在通过s-aes算法在学习s-aes的基础上学习和加强理解AES，加强编程技能，增加对加密算法的安全性理解和理解加密标准。

**环境配置**

系统：Windows

环境：pycharm

库：sys， math， pyqt5

**开发规范**

**1** 变量命名规范使用有意义的名字来为变量、函数和类命名，描述其用途。使用驼峰命名法（CamelCase）或下划线分隔命名法（snake\_case）。使用描述性的名字，避免使用单个字符的变量名，除非是临时变量。

**2** 代码注释使用各编程语言对应的符号进行单行或多行注释，在注释中需使用清晰、有意义的语言，用于解释单行代码或代码块的功能、算法或特定说明（如代码的创新）。同时避免过多的注释，只需复制相对复杂或不明显的部分。

**3** 函数式编程使用函数式编程，将代码模块化，每个小模块负责特定的功能实现。对于重复的代码，可以使用函数或类来实现代码的复用，推荐使用标准库和第三方库辅助算法的实现。

**架构设计**

系统分为程序和界面两部分。

**1程序主体实现了s-aes的细节：**

1.1 密钥扩展

S-AES使用密钥扩展算法，将用户提供的密钥扩展为多个轮次所需的子密钥。密钥扩展是整个算法的关键，确保了数据的安全性。

1.2 轮函数

S-AES包括多个轮次，每个轮次都包含轮函数，用于执行替代和置换操作。这些轮函数采用子密钥对数据进行转换。

1.3 数据块处理

S-AES将数据分为固定大小的数据块，每个数据块都经过轮函数的处理。这些数据块之间可以是无关的，从而支持加密和解密操作。

1.4加密流程

S-AES的加密流程如下：

密钥扩展：使用用户提供的密钥生成多轮次所需的子密钥。

数据分块：将待加密的数据分成固定大小的数据块。

轮次处理：对每个数据块执行多轮次的替代和置换操作，每轮使用不同的子密钥。

输出：加密后的数据块输出。

1.5解密流程

S-AES的解密流程如下：

密钥扩展：使用用户提供的密钥生成多轮次所需的子密钥，与加密时使用的子密钥相反。

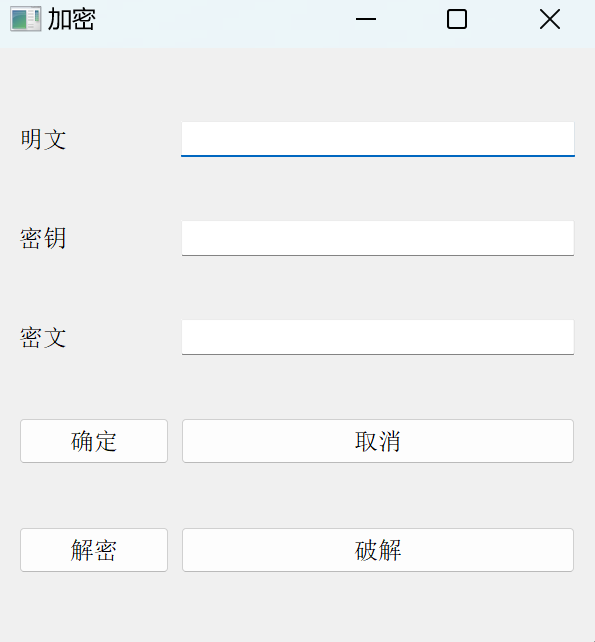
数据分块：将待解密的数据分成固定大小的数据块。

轮次处理：对每个数据块执行多轮次的替代和置换操作，每轮使用不同的子密钥，与加密时相反。

输出：解密后的数据块输出。

**2界面**

在**应用界面**部分实现了给程序使用者进行交互的界面如图1，2。使用者可以在明文栏输入想加密的明文，在密钥行输入密钥，在加密状态下可以得到相关密文。可以点击解密/加密按钮切换加密解密模式，输入明文密钥或者密文和密钥后点击确定即可进行计算。点击破解可以通过一组明密文对得到对应的密钥，切换到破解界面如图3所示：

** **

**图1 图2**



**图3**