开发人员：姜美辰 王钰涵 黄子芮

手册撰写人员：姜美辰 王钰涵 黄子芮

单位：重庆大学大数据与软件学院

联系方式：[1446915231@qq.com](mailto:2507568524@qq.com)

1.引言

1.1 关于本手册

本手册旨在提供S-AES加密解密系统的详细开发说明和技术文档，帮助开发者了解项目的架构设计、功能模块及技术实现。手册涵盖了项目的主要功能特性、使用的技术栈、操作流程及代码实现的关键部分。通过本手册，开发者可以快速掌握如何在系统中实现新的加解密算法功能模块，或对现有功能进行扩展和优化

1.2 目标与范围

本项目的目标是开发一个基于S-AES（简化高级加密标准）的加解密工具，支持基础的加密、解密操作，同时提供扩展的多重加解密和工作模式。项目涵盖的范围包括：

* 基础的S-AES加密和解密功能
* 支持ASCII字符的加密和解密
* 双重、三重加解密以及中间相遇攻击的实现
* 支持CBC工作模式的加解密处理

该项目旨在帮助用户理解加解密算法的基本原理，支持对称加密的学习和实验用途，并为加密技术开发者提供易于扩展的代码框架。

2.项目概述

2.1项目描述

S-AES加密解密系统是一款基于Swing开发的桌面应用程序，实现了S-AES（简化高级加密标准）的加解密功能。项目采用模块化设计，包含多个功能页面，如基础加解密、多重加解密、ASCII处理、工作模式等。每个功能页面独立实现特定的加解密逻辑，用户可以根据需求进行加解密操作并查看结果。

应用程序提供了图形化的用户界面（GUI），用户通过界面可以轻松操作各类加密解密功能，如双重、三重加解密，以及基于CBC模式的处理。此外，项目还具备生成随机密钥和初始向量的功能，方便用户进行实验和学习。

2.2功能特性

1. 基础加解密：

* 支持S-AES标准的16位二进制输入加解密
* 提供用户输入密钥和数据的界面，可视化显示加解密结果
* 支持随机生成16位密钥，简化操作

1. ASCII字符加解密：

* 实现了将ASCII字符转换为二进制，并进行S-AES加密处理
* 提供可视化的加密结果输出，用户输入ASCII字符串后，系统将其转换为二进制并进行加解密

1. 多重加解密：

* 支持双重加解密（三重加解密同样实现）
* 提供了输入32位和48位二进制密钥的功能，并将密钥拆分为多个子密钥进行处理
* 包含中间相遇攻击的演示，可用于教学实验

1. 工作模式：CBC加密：

* 支持CBC（Cipher Block Chaining）模式的加解密，能够处理多块数据的加密操作
* 可输入初始向量（IV）和密钥进行加解密，提供随机生成IV和密钥的功能
* 便于理解块加密模式的基本原理

1. 图形用户界面（GUI）：

* 基于Swing开发，具有简单易用的操作界面
* 提供多个操作界面，用户可在各个界面间自由切换
* 图形化显示加解密的结果，便于理解和调试

2.3技术栈

1. 编程语言：Java

* 项目采用Java语言进行开发，利用其强大的面向对象编程特性构建模块化的加解密系统。

1. 图形用户界面（GUI）：Swing

* 使用Java Swing库开发图形用户界面，为用户提供交互式的操作体验。Swing具有跨平台的特性，使得应用程序能够在不同操作系统上运行。

1. 加密算法：S-AES（Simplified AES）

* 基于高级加密标准（AES）的简化版本，实现基础加密、解密算法。
* 支持ASCII编码和二进制处理，为算法学习提供了简化模型。

1. 工具与库：

* Java SE Development Kit（JDK）：作为项目开发的基础，提供必要的Java库和工具。
* ImageIcon：用于设置界面背景图，提升界面美观度。

3.安装与配置

3.1环境要求

* 操作系统：Windows 10/11
* Java JDK：1.8及以上

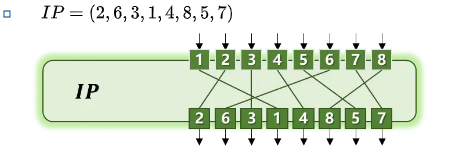
3.2构建与编译

* 编译代码：使用IntelliJ IDEA编译运行代码

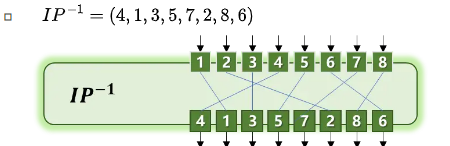
1. 程序结构与主要函数

4.1 函数文件functions.java

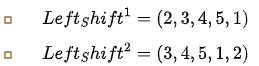
* 置换函数：根据输入的置换表对位串进行重新排列。函数通过遍历置换表中的索引，从输入位串中按顺序提取相应位置的位，形成新的位串。  
   初始置换盒：



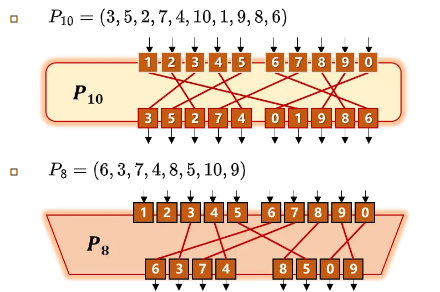
最终置换盒：



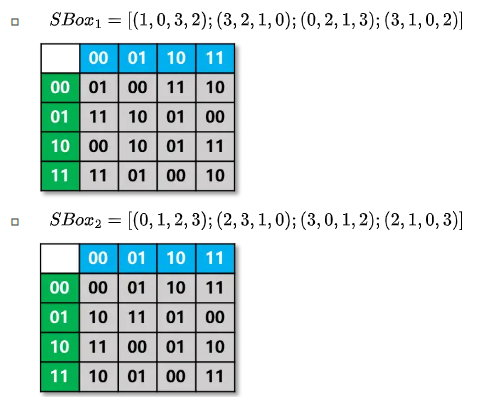
* 左移函数：对密钥进行左移操作。使用不同的左移表分别在第一轮和第二轮生成密钥时对分开的左半部分和右半部分进行不同位数的循环左移。

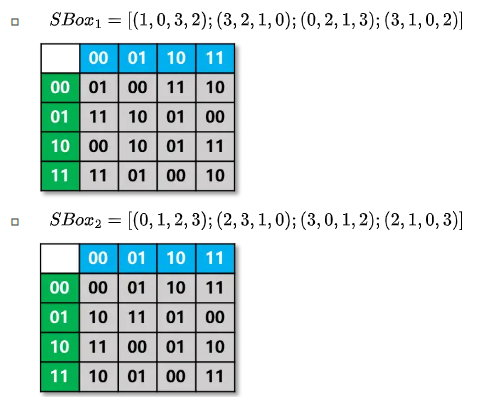


* 生成密钥：通过置换和左移操作生成两个子密钥K1和K2。首先对10位的密钥进行置换，将其分为左右两部分，然后对每部分分别进行左移操作，合并后通过置换生成K1。同理，再进行一次左移操作生成 K2。

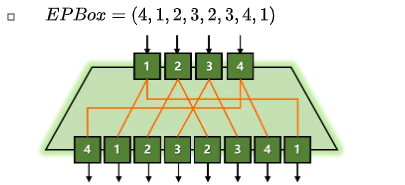


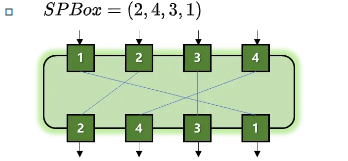
* S盒替换：将输入通过S盒进行替换，返回S盒的输出值。通过将输入的前后两位作为行和列的索引，查找 S盒中的值，并将该值转换为二进制格式。





* 轮函数：首先通过FEPBox对右半部分进行扩展置换，然后与子密钥进行异或操作，再通过S盒替换得到新的位串，最后通过置换SPBox 得到结果。





* 加密函数：先对明文使用初始置换 ()，然后使用轮函数F进行两轮操作，第一轮中，使用第一个子密钥K1对右半部分进行轮函数F操作，并将结果与左半部分进行异或操作，交换左右部分；在第二轮中，使用子密钥K2进行类似的操作。最终结果通过逆置换 () 得到密文。
* 解密函数：首先生成密钥K1和K2，然后以K2和K1的顺序进行两轮轮函数F操作，得到解密后的密文。

4.2 主页面文件MainPage.java

4.2.1 普通加解密

* 整体使用SpringLayout布局，使用add方法将输入框、标签、按钮等组件依次添加到窗口中，并通过JRadioButton提供了两个互斥的加密和解密的操作选项。
* 通过输入框获取10bits的密钥、8bits的明文或密文，也可以通过generateRandomKey()进一步调用functions类的generateKeys方法随机生成密钥，并通过validateInput()方法检验输入格式是否为二进制，长度是否符合要求。
* ActionListener 捕捉到按钮触发的加密或解密事件，并根据用户选择的操作调用并执行functions类中对应的函数。

4.2.2 ASCII码加解密

* 整体使用SpringLayout布局，使用add方法将输入框、标签、按钮等组件依次添加到窗口中，并通过JRadioButton提供了两个互斥的加密和解密的操作选项。
* 通过输入框获取10bits的密钥、1bytes的明文或密文，也可以通过generateRandomKey()方法随机生成密钥，并通过validateInput()方法检验输入格式是否符合要求。
* 加密函数将输入的 ASCII 字符会先转换为8位的二进制形式，并与普通加密相同，使用functions类中加密函数进行加密，将结果的8为二进制密文转化为ASCII字符，解密函数同理。
* ActionListener 捕捉到按钮触发的加密或解密事件，并根据用户选择的操作调用执行对应的函数。

4.2.3 暴力破解

* 整体使用GridBagLayout布局，组织了明文、密文的输入框，结果显示区域，破解时间输出框，以及用于选择线程模式的单选按钮。
* 通过输入框获取8bits的明文与对应的密文，并通过单选按钮让用户选择单线程破解模式或多线程破解模式。单线程破解逐一遍历所有可能的密钥，逐个解密并检查解密结果是否与明文匹配，多线程破解使用了Java的ExecutorService 和线程池来并发执行密钥遍历任务，使得每个线程负责处理一部分密钥空间，提高了破解速度。
* 用户点击执行后，程序遍历所有可能的密钥组合，尝试解密密文，并将解密后的结果与输入的明文进行比对。如果匹配，则认为找到的密钥是可能的正确密钥，并输出到JTextArea。
* 程序通过在操作开始和结束添加时间戳，计算了破解过程所花费的时间，并将结果输出到timeField中。

1. 安装指南

5.1 安装JDK

请确保您的电脑上具有JDK-11或其他兼容版本，若没有的话请进行安装与配置。

5.2 启动系统

请在IntelliJ IDEA或其他支持JAVA语言的软件平台上运行项目。

5.3 导入程序

请下载本程序的压缩包并解压，在您选择的软件平台中打开该项目，检查程序文件没有缺失后可直接运行并使用。