**S-AES加解密系统用户手册**

**一、系统概述**

S-AES加解密系统是一个基于简化AES算法的图形化加密工具，支持多种加密模式和功能。系统采用PyQt5构建，界面友好，操作简单。

**二、系统要求**

1.Python 3.6+

2.PyQt5库

3.操作系统：Windows/Linux/macOS

**三、安装说明**

1. 确保已安装Python 3.6或更高版本

2. 安装所需依赖：pip install PyQt5

3. 运行程序：python 2.py

**四、功能模块说明**

1. 基本测试

（1）功能：基础的S-AES加密和解密操作

（2）操作步骤：

a. 在"明文"输入框输入4位十六进制数（如：6F6B）

b. 在"密钥"输入框输入4位十六进制数（如：A73B）

c. 点击"加密"按钮进行加密

d. 点击"解密"按钮进行解密

e. 结果将显示在输出区域

（3）注意事项：

a.输入必须为4位十六进制字符（0-9, A-F）

b.加密和解密使用相同的密钥

2. ASCII加解密

（1）功能：对ASCII文本进行加密和解密

（2）操作步骤：

a. 在"ASCII文本"输入框输入要加密的文本

b. 在"密钥"输入框输入4位十六进制数

c. 点击"ASCII加密"按钮进行加密

d. 点击"ASCII解密"按钮进行解密

e. 结果将显示在输出区域

（3）注意事项：

a.文本将按2字节分组进行加密

b.解密时需确保使用正确的密钥

3. 多重加密

（1）功能：支持双重和三重S-AES加密

（2）加密类型：

a.双重加密（8位十六进制密钥）

b.三重加密-8位十六进制密钥

c.三重加密-12位十六进制密钥

（3）操作步骤：

a. 选择加密类型

b. 在"明文"输入框输入4位十六进制数

c. 在"密钥"输入框输入对应长度的十六进制数

d. 点击"多重加密"或"多重解密"按钮

e. 结果将显示在输出区域

（4）注意事项：

a.密钥长度必须与所选加密类型匹配

b.双重加密使用32位密钥（8位十六进制）

c.三重加密使用32位或48位密钥（8位或12位十六进制）

4. CBC模式

（1）功能：密码分组链接模式加密

（2）操作步骤：

a. 在"明文"输入框输入多个4位十六进制块，用空格分隔

b. 在"密钥"输入框输入4位十六进制数

c. 在"初始向量"输入框输入4位十六进制数

d. 点击"CBC加密"或"CBC解密"按钮

e. 点击"篡改测试"可进行错误传播测试

（3）注意事项：

a.每个数据块必须为4位十六进制数

b.初始向量(IV)应为随机值，确保每次加密不同

c.篡改测试展示CBC模式的错误传播特性

5. 密码分析

（1）功能：中间相遇攻击分析

（2）操作步骤：

a. 在"明文"输入框输入4位十六进制明文

b. 在"密文"输入框输入4位十六进制密文

c. 点击"添加对"按钮添加明密文对

d. 可添加多个明密文对提高攻击准确性

e. 点击"开始攻击"按钮执行中间相遇攻击

f. 查看攻击结果，包括可能的密钥对

（3）注意事项：

a. 需要至少一个明密文对

b. 多个明密文对可以提高攻击准确性

c. 为加快演示速度，默认只搜索部分密钥空间

1. **输入输出格式**

1.输入格式：

（1）十六进制数：使用0-9和A-F字符，不区分大小写

（2）ASCII文本：标准ASCII字符

（3）数据块：多个十六进制数用空格分隔

2.输出格式：

（1）十六进制结果：4位十六进制数

（2）二进制结果：16位二进制数

（3）文本结果：加密后的文本可能包含不可见字符

（4）攻击结果：可能的密钥对和验证信息

**六、常见问题**

Q: 加密后为什么出现乱码？

A: 加密后的数据可能包含非打印字符，这是正常现象。解密时使用正确密钥即可恢复原文本。

Q: 为什么CBC模式需要初始向量？

A: 初始向量确保相同的明文加密后产生不同的密文，增强安全性。

Q: 篡改测试有什么作用？

A: 篡改测试展示CBC模式下，一个密文块的错误会影响两个明文块，体现了加密链的完整性保护特性。

Q: 中间相遇攻击有什么用途？

A: 中间相遇攻击用于分析双重加密的安全性，通过已知的明密文对来寻找可能的加密密钥。

Q: 为什么中间相遇攻击有时找不到密钥？

A: 为了演示速度，默认只搜索部分密钥空间。要搜索完整密钥空间，需要修改代码中的搜索范围。

**七、技术支持**

如遇程序问题，请检查：

1. 输入格式是否正确

2. 密钥长度是否匹配

3. Python和PyQt5是否安装正确

4. 明密文对是否使用相同的加密密钥