#### SAES加解密系统开发手册

1. 系统概述

SAES（Simplified AES）加解密系统是一个基于简化版AES算法的加密工具，提供基本的加密解密功能以及多种扩展功能。

本程序实现了SAES（简化版AES）加密算法及其变种，包括：

基本SAES加密/解密、双重SAES加密/解密、三重SAES加密/解密（支持32位和48位密钥）、CBC工作模式、ASCII文本加解密

程序使用PyQt5构建了友好的图形用户界面，提供了四个主要功能标签页：基本测试、ASCII加解密、多重加密和CBC模式。

系统采用模块化设计，使用Python和PyQt5实现图形用户界面。系统架构如下：

SAES系统

├── 核心算法层 (SAES, DoubleSAES, TripleSAES, CBCMode,

MeetInTheMiddleAttack)

├── 业务逻辑层 (SAESGUI)

└── 表示层 (PyQt5 GUI)

2. 核心算法组件接口文档

2.1 SAES类

功能描述：实现SAES加密算法的核心类，提供基本的加密和解密功能。

公共成员

类变量：

S\_BOX：S盒替换表，用于加密过程中的半字节替换

INV\_S\_BOX：逆S盒替换表，用于解密过程中的半字节替换

MIX\_MATRIX：列混淆矩阵，用于加密过程中的列混淆操作

INV\_MIX\_MATRIX：逆列混淆矩阵，用于解密过程中的列混淆操作

RCON：轮常量，用于密钥扩展

公共方法

构造函数

**\_\_init\_\_(self)**

参数：无，返回值：无

说明：初始化SAES实例，无需特别的初始化操作

**gf\_mult(self, a, b)**

参数：a：整数，015范围内，表示GF(2^4)中的一个元素

b：整数，015范围内，表示GF(2^4)中的一个元素

返回值：整数，a和b在GF(2^4)上的乘法结果

说明：使用查找表实现GF(2^4)有限域上的乘法运算

**key\_expansion(self, key)**

参数：key：整数，16位密钥（0x00000xFFFF）

返回值：列表，包含3个轮密钥的列表 [k0, k1, k2]

说明：将16位主密钥扩展为3个轮密钥，用于加密和解密过程

**sub\_nib(self, byte, s\_box)**

参数：byte：整数，8位字节值

s\_box：二维列表，S盒或逆S盒替换表

返回值：整数，经过半字节替换后的8位字节值

说明：对输入字节的高4位和低4位分别进行S盒查找替换

**shift\_row(self, state)**

参数：state：整数，16位状态值

返回值：整数，经过行移位后的16位状态值

说明：对状态矩阵的第二行进行循环左移一个半字节的操作

**mix\_columns(self, state, mix\_matrix)**

参数：state：整数，16位状态值

mix\_matrix：二维列表，列混淆矩阵或逆列混淆矩阵

返回值：整数，经过列混淆后的16位状态值

说明：使用指定的混淆矩阵对状态进行列混淆操作

**add\_round\_key(self, state, round\_key)**

参数：state：整数，16位状态值

round\_key：整数，16位轮密钥

返回值：整数，轮密钥加操作后的16位状态值

说明：将状态值与轮密钥进行逐位异或操作

**encrypt(self, plaintext, key)**

参数：plaintext：整数，16位明文值（0x00000xFFFF）

key：整数，16位密钥值（0x00000xFFFF）

返回值：整数，16位密文值（0x00000xFFFF）

说明：执行SAES加密过程，包括密钥扩展、初始轮密钥加、一轮完整的加密轮和一轮简化的加密轮

**decrypt(self, ciphertext, key)**

参数：ciphertext：整数，16位密文值（0x00000xFFFF）

key：整数，16位密钥值（0x00000xFFFF）

返回值：整数，16位明文值（0x00000xFFFF）

说明：执行SAES解密过程，包括密钥扩展、初始轮密钥加、一轮完整的解密轮和一轮简化的解密轮

2.2 DoubleSAES类

功能描述：实现双重SAES加密，通过将两个SAES加密级联，提高安全性。

公共成员

实例变量：

saes：SAES类的实例，用于执行基本的SAES加密和解密

公共方法

构造函数：

**\_\_init\_\_(self)**

参数：无

返回值：无

说明：初始化DoubleSAES实例，创建SAES对象

**encrypt(self, plaintext, key)**

参数：plaintext：整数，16位明文值（0x00000xFFFF）

key：整数，32位密钥值（0x000000000xFFFFFFFF），高16位为key1， 低16位为key2

返回值：整数，16位密文值（0x00000xFFFF）

说明：执行双重SAES加密，先用key1加密，再用key2加密

**decrypt(self, ciphertext, key)**

参数：ciphertext：整数，16位密文值（0x00000xFFFF）

key：整数，32位密钥值（0x000000000xFFFFFFFF），高16位为key1， 低16位为key2

返回值：整数，16位明文值（0x00000xFFFF）

说明：执行双重SAES解密，先用key2解密，再用key1解密

2.3 TripleSAES类

功能描述：实现三重SAES加密，支持两种密钥长度：32位和48位。

公共成员

实例变量：saes：SAES类的实例，用于执行基本的SAES加密和解密

公共方法

构造函数

**\_\_init\_\_(self)**

参数：无

返回值：无

说明：初始化TripleSAES实例，创建SAES对象

**encrypt\_32bit(self, plaintext, key)**

参数：plaintext：整数，16位明文值（0x00000xFFFF）

key：整数，32位密钥值（0x000000000xFFFFFFFF），高16位为key， 低16位为key2

返回值：整数，16位密文值（0x00000xFFFF）

说明：执行32位密钥的三重SAES加密，使用加密解密加密（EDE）模式

**decrypt\_32bit(self, ciphertext, key)**

参数：ciphertext：整数，16位密文值（0x00000xFFFF）

key：整数，32位密钥值（0x000000000xFFFFFFFF），高16位为key1， 低16位为key2

返回值：整数，16位明文值（0x00000xFFFF）

说明：执行32位密钥的三重SAES解密，使用解密加密解密（DED）模式

**encrypt\_48bit(self, plaintext, key)**

参数：plaintext：整数，16位明文值（0x00000xFFFF）

key：整数，48位密钥值，高16位为key1，中间16位为key2，低 16位为key3

返回值：整数，16位密文值（0x00000xFFFF）

说明：执行48位密钥的三重SAES加密，使用加密加密加密（EEE）模式

**decrypt\_48bit(self, ciphertext, key)**

参数：ciphertext：整数，16位密文值（0x00000xFFFF）

key：整数，48位密钥值，高16位为key1，中间16位为key2，低 16位为key3

返回值：整数，16位明文值（0x00000xFFFF）

说明：执行48位密钥的三重SAES解密，使用解密解密解密（DDD）模式

2.4 CBCMode类

功能描述：实现CBC工作模式，提供分组密码的链接加密方式。

公共成员

实例变量：saes：SAES类的实例，用于执行基本的SAES加密和解密

公共方法

构造函数

**\_\_init\_\_(self)**

参数：无

返回值：无

说明：初始化CBCMode实例，创建SAES对象

**encrypt(self, plaintext\_blocks, key, iv)**

参数：plaintext\_blocks：列表，包含多个16位整数的明文块

key：整数，16位密钥值（0x00000xFFFF）

iv：整数，16位初始向量（0x00000xFFFF）

返回值：列表，包含多个16位整数的密文块

说明：执行CBC模式加密，每个明文块先与前一个密文块异或，再进行加密

**decrypt(self, ciphertext\_blocks, key, iv)**

参数：ciphertext\_blocks：列表，包含多个16位整数的密文块

key：整数，16位密钥值（0x00000xFFFF）

iv：整数，16位初始向量（0x00000xFFFF）

返回值：列表，包含多个16位整数的明文块

说明：执行CBC模式解密，先解密每个密文块，再与前一个密文块异或

2.5 MeetInTheMiddleAttack类

功能描述：实现中间相遇攻击，用于分析双重加密的安全性。

公共成员

实例变量：saes：SAES类的实例，用于执行基本的SAES加密和解密

公共方法

构造函数

**\_\_init\_\_(self)**

参数：无

返回值：无

说明：初始化MeetInTheMiddleAttack实例，创建SAES对象

**generate\_encryption\_table(self, plaintext)**

参数：plaintext：整数，16位明文值

返回值：字典，中间值到密钥K1的映射表

说明：生成加密表，记录所有可能的K1加密明文得到的中间值

**generate\_decryption\_table(self, ciphertext)**

参数：ciphertext：整数，16位密文值

返回值：字典，中间值到密钥K2的映射表

说明：生成解密表，记录所有可能的K2解密密文得到的中间值

**attack\_single\_pair(self, plaintext, ciphertext)**

参数：plaintext：整数，16位明文值

ciphertext：整数，16位密文值

返回值：列表，包含可能的密钥对(K1, K2, 完整密钥)的列表

说明：对单个明密文对执行中间相遇攻击

**attack\_multiple\_pairs(self, pairs)**

参数：pairs：列表，包含多个(明文, 密文)元组的列表

返回值：列表，经过验证的可能的密钥对列表

说明：对多个明密文对执行中间相遇攻击，提高攻击准确性

2.6 SAESGUI类

功能描述：实现SAES加解密系统的图形用户界面，提供用户交互功能。

公共成员

实例变量：

saes：SAES类的实例

double\_saes：DoubleSAES类的实例

triple\_saes：TripleSAES类的实例

cbc\_mode：CBCMode类的实例

mitm\_attack：MeetInTheMiddleAttack类的实例

各种UI控件（如QLineEdit、QTextEdit等）

公共方法

构造函数

**\_\_init\_\_(self)**

参数：无

返回值：无

说明：初始化SAESGUI实例，创建各个加密类的实例，并初始化用户界面

**init\_ui(self)**

参数：无

返回值：无

说明：设置主窗口属性，创建标签页控件，并添加四个功能标签页

**create\_basic\_tab(self)**

参数：无

返回值：QWidget对象

说明：创建基本测试标签页，包含明文输入、密钥输入和加密/解密按钮

**create\_ascii\_tab(self)**

参数：无

返回值：QWidget对象

说明：创建ASCII加解密标签页，支持ASCII文本的加解密

**create\_multi\_tab(self)**

参数：无

返回值：QWidget对象

说明：创建多重加密标签页，支持双重和三重SAES加密

**create\_cbc\_tab(self)**

参数：无

返回值：QWidget对象

说明：创建CBC模式标签页，支持CBC模式的加解密和篡改测试

**create\_attack\_tab(self)**

参数：无

返回值：QWidget对象

说明：创建密码分析标签页，支持中间相遇攻击

**validate\_hex(self, text, length)**

参数：text：字符串，要验证的十六进制字符串

length：整数，期望的字符串长度

返回值：布尔值，验证是否通过

说明：验证输入是否为指定长度的合法十六进制字符串

**hex\_to\_int(self, hex\_str)**

参数：hex\_str：字符串，十六进制字符串

返回值：整数，转换后的整数值

说明：将十六进制字符串转换为整数

**int\_to\_binary(self, number, length)**

参数：number：整数，要转换的整数值

length：整数，期望的二进制字符串长度

返回值：字符串，格式化的二进制字符串

说明：将整数转换为指定长度的二进制字符串

**int\_to\_hex(self, number, length)**

参数：number：整数，要转换的整数值

length：整数，期望的十六进制字符串长度

返回值：字符串，格式化的十六进制字符串

说明：将整数转换为指定长度的十六进制字符串

**basic\_encrypt(self)**

参数：无

返回值：无

说明：执行基本的SAES加密，处理UI事件

**basic\_decrypt(self)**

参数：无

返回值：无

说明：执行基本的SAES解密，处理UI事件

**ascii\_encrypt(self)**

参数：无

返回值：无

说明：执行ASCII文本的SAES加密，处理UI事件

**ascii\_decrypt(self)**

参数：无

返回值：无

说明：执行ASCII文本的SAES解密，处理UI事件

**multi\_encrypt(self)**

参数：无

返回值：无

说明：执行多重SAES加密，处理UI事件

**multi\_decrypt(self)**

参数：无

返回值：无

说明：执行多重SAES解密，处理UI事件

**cbc\_encrypt(self)**

参数：无

返回值：无

说明：执行CBC模式加密，处理UI事件

**cbc\_decrypt(self)**

参数：无

返回值：无

说明：执行CBC模式解密，处理UI事件

**tamper\_test(self)**

参数：无

返回值：无

说明：执行CBC模式的篡改测试，展示篡改一个密文块对解密结果的影响

**add\_attack\_pair(self)**

参数：无

返回值：无

说明：添加明密文对到攻击列表

**perform\_attack(self)**

参数：无

返回值：无

说明：执行中间相遇攻击并显示结果

**clear\_basic(self)**

参数：无

返回值：无

说明：清空基本测试标签页的输入和输出

**clear\_ascii(self)**

参数：无

返回值：无

说明：清空ASCII标签页的输入和输出

**clear\_multi(self)**

参数：无

返回值：无

说明：清空多重加密标签页的输入和输出

**clear\_cbc(self)**

参数：无

返回值：无

说明：清空CBC模式标签页的输入和输出

**clear\_attack(self)**

参数：无

返回值：无

说明：清空密码分析标签页的输入和输出

3. 程序使用方法

3.1 基本测试

1. 输入4位十六进制明文（如：6F6B）

2. 输入4位十六进制密钥（如：A73B）

3. 点击「加密」按钮进行加密，或点击「解密」按钮进行解密

4. 查看输出区域的结果

3.2 ASCII加解密

1. 输入ASCII文本（如：Hello）

2. 输入4位十六进制密钥（如：A73B）

3. 点击「ASCII加密」按钮进行加密，或点击「ASCII解密」按钮进行解密

4. 查看输出区域的结果，包括加密/解密文本和对应的块数据

3.3 多重加密

1. 选择加密类型：双重加密或三重加密

2. 输入4位十六进制明文

3. 根据选择的加密类型，输入8位或12位十六进制密钥

4. 点击「多重加密」按钮进行加密，或点击「多重解密」按钮进行解密

5. 查看输出区域的结果

3.4 CBC模式

1. 输入多个4位十六进制块，用空格分隔（如：6F6B 7365）

2. 输入4位十六进制密钥

3. 输入4位十六进制初始向量

4. 点击「CBC加密」按钮进行加密，或点击「CBC解密」按钮进行解密

5. 可以点击「篡改测试」按钮查看CBC模式下篡改密文的影响

3.5 密码分析

1. 在明文和密文输入框中分别输入4位十六进制值

2. 点击「添加对」按钮添加明密文对

3. 可以添加多个明密文对提高攻击准确性

4. 点击「开始攻击」按钮执行中间相遇攻击

5. 查看攻击结果，包括可能的密钥对和验证信息

4. 注意事项

1. 输入格式验证：程序对输入的十六进制字符串进行严格验证，请确保输入符合要求的格式和长度

2. ASCII文本限制：ASCII加解密功能仅支持ASCII编码的字符（0127）

3. 密钥管理：请妥善保管密钥，密钥丢失将导致无法解密数据

4. 篡改测试：CBC模式下，篡改一个密文块会影响当前块和下一个块的解密结果

5. 中间相遇攻击：为了演示速度，默认只搜索前256个密钥。要搜索完整密钥空间，需要修改代码中的循环范围

6. 错误处理：程序包含基本的错误处理机制，如输入验证失败会显示警告对话框

5. 代码结构与扩展建议

5.1 代码结构

程序采用面向对象设计，各个类的职责明确：

SAES：实现核心加密算法

DoubleSAES 和 TripleSAES：基于SAES实现多重加密

CBCMode：实现工作模式

MeetInTheMiddleAttack：实现密码分析攻击

SAESGUI：提供用户界面

5.2 扩展建议

1. 增加更多工作模式：可以添加ECB、CFB、OFB等工作模式

2. 支持更多编码格式：扩展对UTF8等编码的支持

3. 文件加密功能：添加文件加密和解密功能

4. 密钥生成：增加随机密钥生成功能

5. 性能优化：对大型数据的加解密进行性能优化

6. 更多攻击方法：添加差分分析、线性分析等密码分析方法

6. 依赖要求

Python 3.x

PyQt5库

7. 运行程序

直接运行Python文件即可启动程序：

bash

python 2.py

程序将显示一个图形用户界面，包含四个功能标签页，用户可以根据需要选择相应的功能进行加解密操作。