二、开发手册

1. 系统架构

系统采用面向对象设计，分为核心算法层和界面交互层：

核心算法层：实现 S-AES 及扩展功能的核心逻辑

界面交互层：通过 Tkinter 构建图形界面，关联算法功能与用户操作

2. 核心类说明

2.1 SAESEncryptor 类（基础加密实现）

核心属性：

s\_box/inv\_s\_box：S 盒与逆 S 盒表

round\_const：轮常量

debug\_mode：调试模式开关

核心方法：

encrypt(plaintext, key)：16 位明文加密

decrypt(ciphertext, key)：16 位密文解密

辅助方法：bin\_to\_matrix（二进制转矩阵）、matrix\_sub（矩阵半字节替换）、mix\_col（列混淆）等

2.2 扩展加密类

DoubleEncrypt：继承SAESEncryptor，实现double\_encrypt（双重加密）和double\_decrypt（双重解密）

TripleEncrypt：继承SAESEncryptor，实现 32 位 / 48 位密钥的三重加密解密

2.3 攻击与模式类

MiddleAttack：实现中间相遇攻击，通过预计算 K1 加密结果与 K2 解密结果的交集寻找密钥对

CBCMode：实现 CBC 模式的加密解密，处理分组链接逻辑

2.4 界面类 SAESInterface

构建 7 个功能选项卡，关联算法方法与界面控件

通过TextRedirect类将控制台输出重定向到界面文本框

3. 算法实现细节

3.1 S-AES 加密流程

初始轮密钥加：将明文与轮密钥 0 进行异或

第 1 轮：半字节替换 → 行移位 → 列混淆 → 轮密钥加（轮密钥 1）

第 2 轮：半字节替换 → 行移位 → 轮密钥加（轮密钥 2）

3.2 密钥扩展

16 位初始密钥生成 3 个 16 位轮密钥

通过g\_transform函数（循环左移 + S 盒替换 + 轮常量异或）扩展密钥

3.3 有限域运算

gf\_mul方法实现 GF (2⁴) 域上的乘法（模多项式 x⁴ + x + 1）

4. 扩展与维护建议

功能扩展：可新增 CTR、OFB 等工作模式，在CBCMode类基础上扩展

性能优化：中间相遇攻击可通过多线程加速密钥遍历（修改MiddleAttack.attack方法）

界面优化：增加输入验证（如二进制格式校验）和错误提示细化

测试维护：定期更新run\_std\_test中的标准测试用例，确保算法一致性

5. 常见问题处理

输入错误：确保二进制输入长度符合要求（明文 16 位、密钥 16/32/48 位）

解密失败：检查密钥是否正确，加密解密使用的密钥必须一致

攻击效率：测试密钥数设置过大会导致卡顿，建议先使用 1000 进行测试

依赖缺失：若提示numpy错误，执行pip install numpy安装依赖库