

# TiMi 小组关于 S-AES 加解密项目开发手册

## 一 . 概述

本项目可通过 GUI 界面实现对二进制、ASCII 编码的数据进行加/解密，此外还可以实现双重加密、三重加密以及 CBC 加/解密，中间相遇攻击（即暴力破解）。

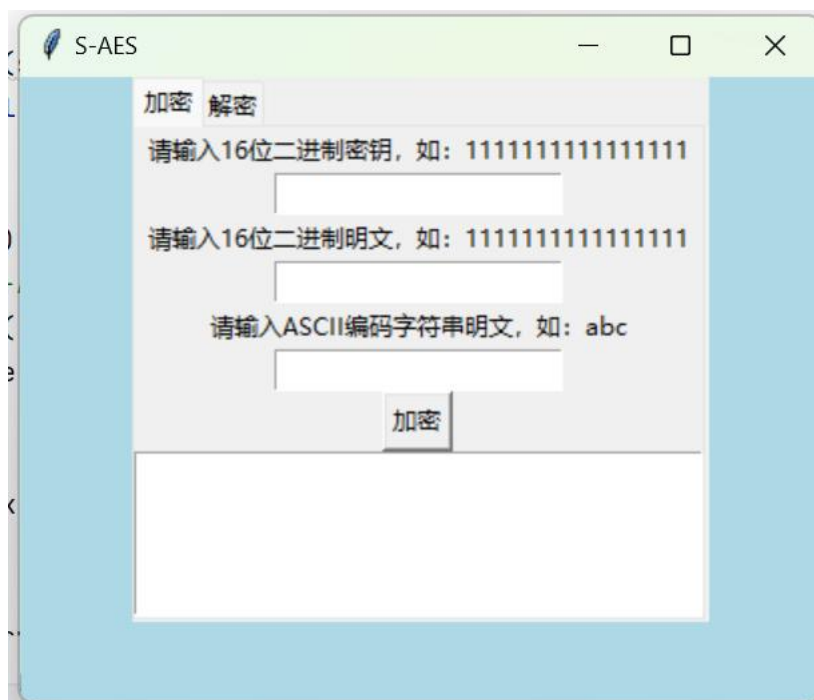
## 二 . GUI 界面

### 2.1 相关代码

GUI 界面相关代码可参考代码项目中 GUI.py 相关文件。

### 2.2 具体界面及操作解释

用户可以通过运行 GUI.py 文件可得：



用户输入 16 位二进制的明/密文以及 ASCII 码值，可进行加/解密：

S-AES

加密 解密

请输入16位二进制密钥，如：1111111111111111

1111000010100101

请输入16位二进制明文，如：1111111111111111

1111111111111111

请输入ASCII编码字符串明文，如：abc

hellx

加密

加密结果：0011010000100011  
ASCII加密结果：Ã½Q

S-AES

加密 解密

请输入16位二进制密钥，如：1111111111111111

1111000010100101

请输入16位二进制密文，如：1111111111111111

0011010000100011

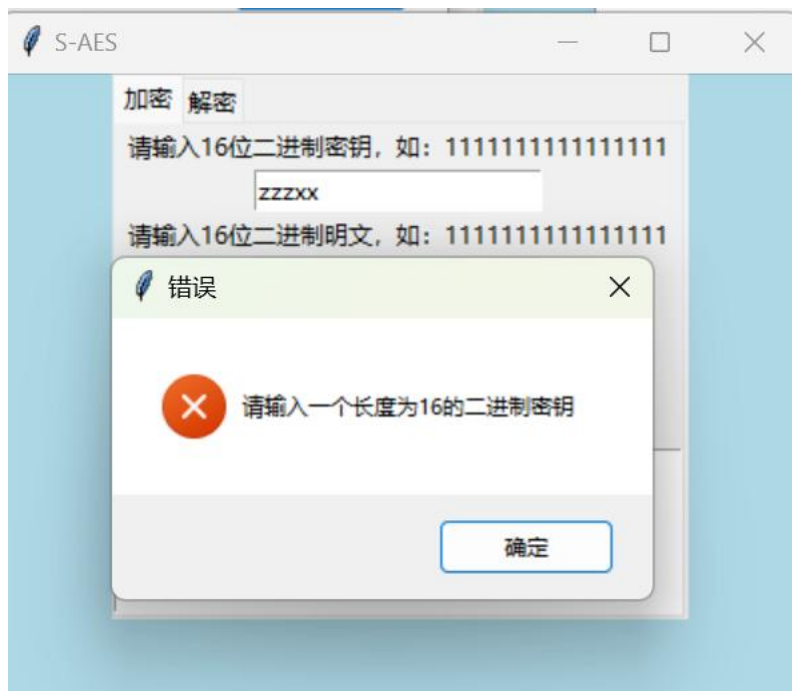
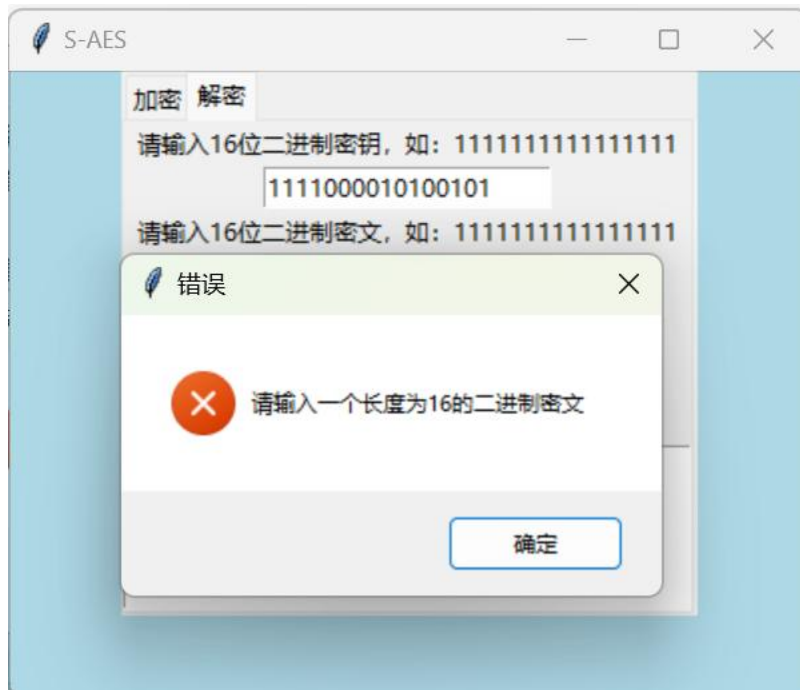
请输入ASCII编码字符串密文，如：abc

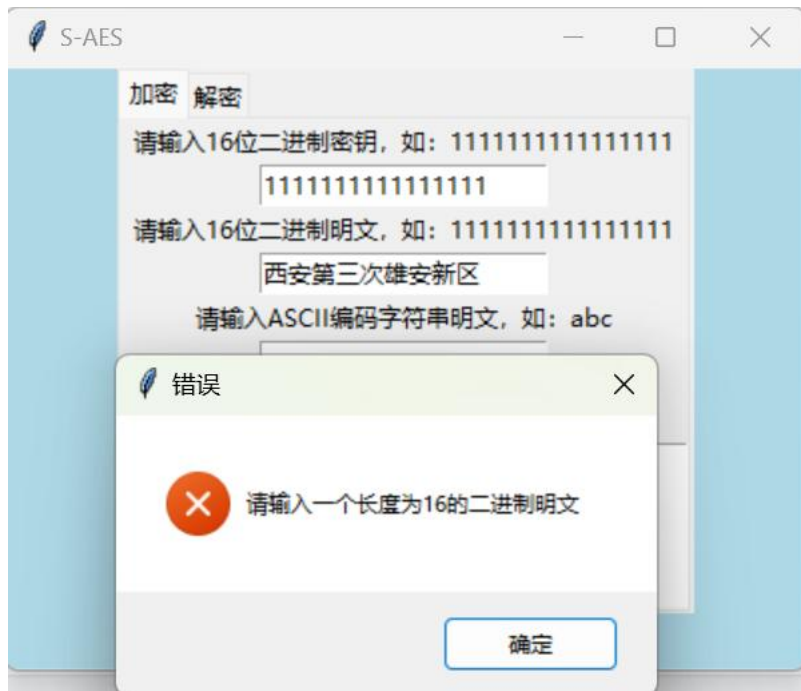
Ã½Q

解密

解密结果：1111111111111111  
ASCII解密结果：hellx

若输入密钥或者明文的长度、格式不对（比如密钥长度不为 16，二进制明文长度不为 16，或者格式不为二进制），会有相关提醒：





双重加密/三重加密：

双重加密采用 ppt 上第一种加/解密方式：

二重加密：

请输入16位二进制明文或4位十六进制明文：0000111100001111

请输入32位二进制密钥或8位十六进制密钥：00000000000000001001011010101010

本次二重加密的密文为：1001110100000000

二重解密：

请输入16位二进制明文或4位十六进制密文：1001110100000000

请输入32位二进制密钥或8位十六进制密钥：00000000000000001001011010101010

本次二重解密的明文为：0000111100001111

三重加密采用{k1, k2}模式：

三重加密：

请输入16位二进制明文或4位十六进制明文：0000111100001111

请输入32位二进制密钥：00000000000000001001011010101010

本次三重加密的密文为：1001101001011000

三重解密：

请输入16位二进制明文或4位十六进制密文：1001110100000000

请输入32位二进制密钥：00000000000000001001011010101010

本次三重解密的明文为：1001001110010000

CBC 加/解密：其中可发现小小的篡改密文后引来变化也是很大

CBC模式加密:

请输入明文: 10101010010101010010101001001010

请输入16位二进制密钥: 1111000010100101

本次加密的密文为: 11101000011011011111000101010000

CBC模式解密:

请输入密文: 11101000011011011111000101010000

请输入16位二进制密钥: 1111000010100101

本次解密的明文为: 10101010010101010010101001001010

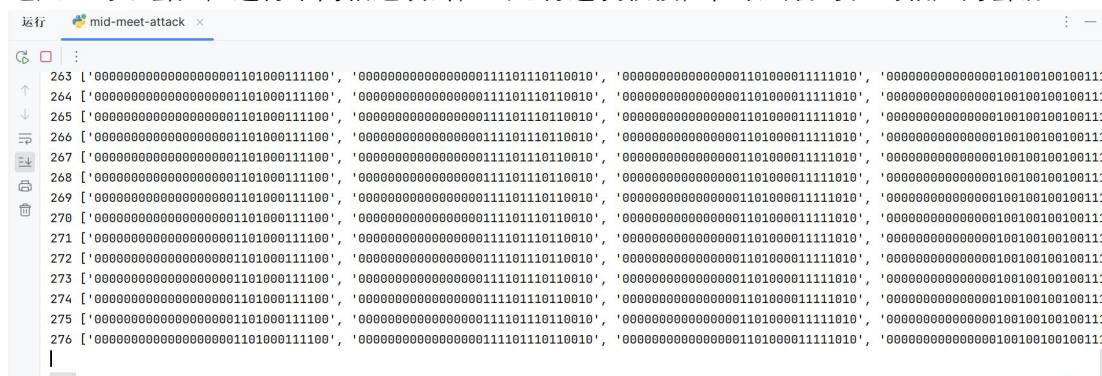
请输入篡改后的密文: 10001001011011011111000101010000

修改后解密的明文为: 11111011010011010100101101001010

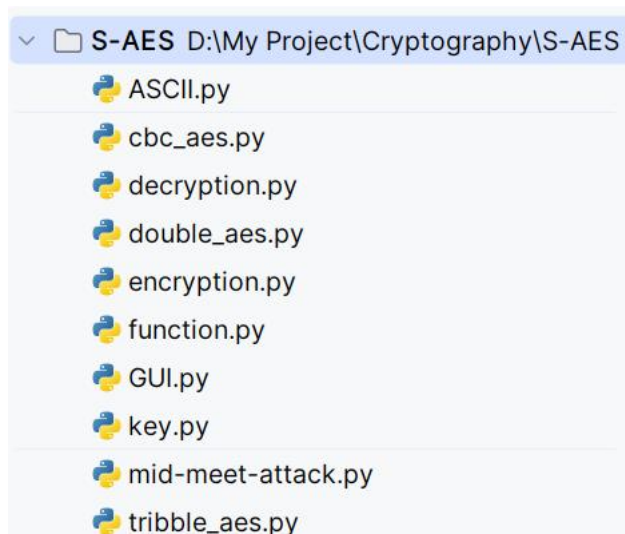
可见, 篡改密文后的解密结果和原来不一致。

中间相遇攻击:

选定一对明密文, 进行中间相遇攻击, 虽运行速度较慢, 但最终仍可找到相应的密钥



### 三. 项目代码部分相关介绍



其中 GUI.py 主要关于界面与函数接口等的融合; function.py 主要设计轮密钥加、行位移、列混淆、S-box 等的设计; key.py 主要设计了密钥扩展; encryption.py 主要完成加密过程;

decryption.py 主要完成解密过程；ASCII.py 完成了对于 ASCII 编码的加/解密；mid-meet-attack.py 完成了中间相遇攻击；tribble\_aes.py, double\_aes.py, cbc\_aes.py 分别是关于三重加/解密、双重加/解密、CBC 对长明/密文加/解密。  
可运行文件为：GUI.py；tribble\_aes.py, double\_aes.py, cbc\_aes.py, mid-meet-attack.py 文件

## 四 . 项目背景介绍

S-AES 算法加/解密原理流程图如下：

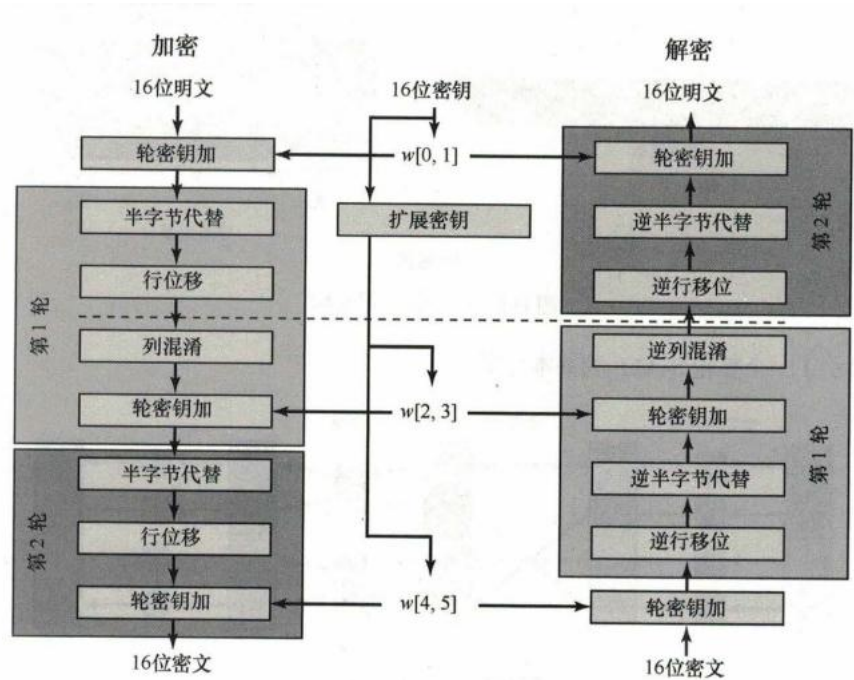
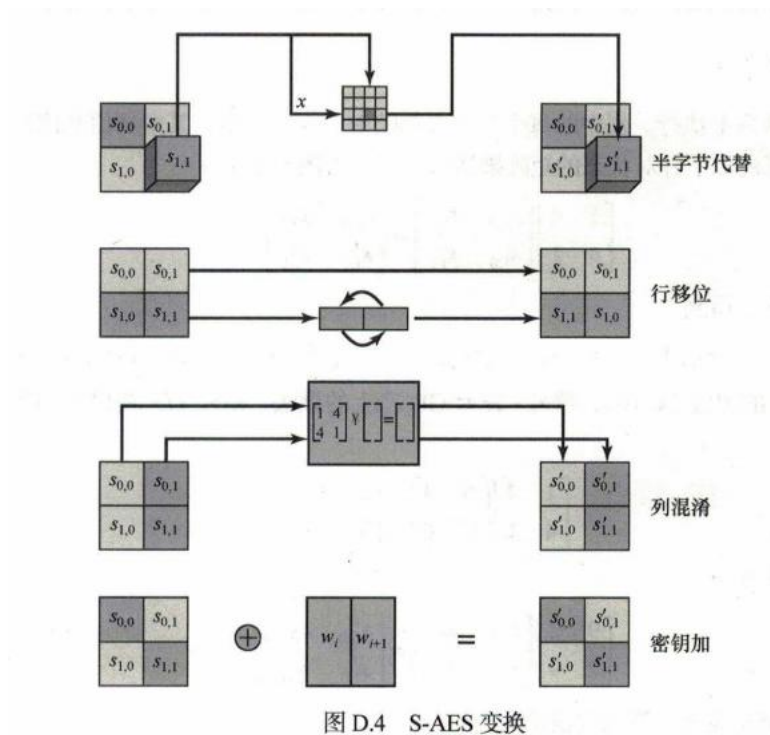


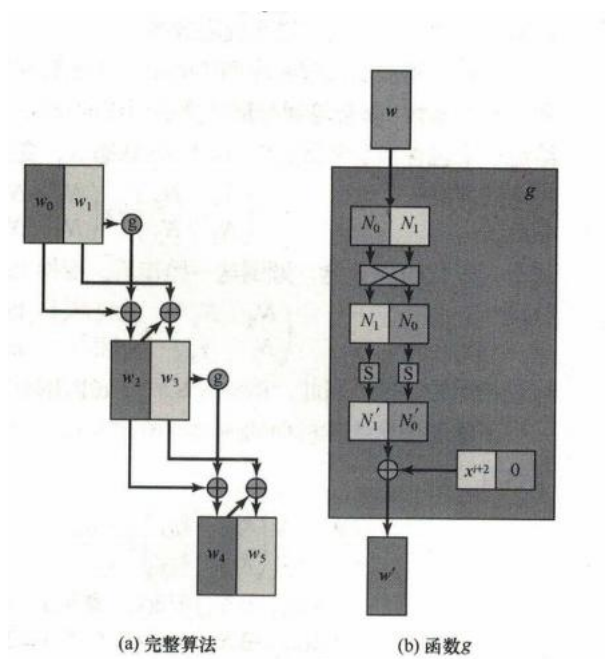
图 D.1 S-AES 加密和解密

S-AES 流程如下：





密钥扩展：



## 五．使用步骤

- 运行 GUI.py 文件
- 可选择“加密”或者“解密”选项
- 输入相应的密钥，二进制明/密文（可选），ASCII 编码的明/密文（可选），选择加/解

密

- 若进行双重/三重/CBC 加/解密，可运行 tribble\_aes.py, double\_aes.py, cbc\_aes.py, mid-meet-attack.py 文件
- 运行 mid-meet-attack.py 文件即可进行中间相遇攻击（但注意暴力破解由于是十六进制，运行时间较长，请谨慎使用）

## 六．其他帮助

TiMi 小组是一个优秀的团队，且热情负责。若您在使用过程中出现任何困惑不解，[可发送邮件至 891073279@qq.com](mailto:891073279@qq.com) 或者 3416924346@qq.com。