S-AES 算法加密解密程序用户指南

1. 简介

S-AES（简化高级加密标准）是一种简化版的高级加密标准，旨在为教学和学习目的提供一种易于理解的加密算法。S-AES保留了AES算法的核心思想，但通过简化密钥长度和块数据大小，产品更适合小规模的数据加密需求。本程序是用Java+swing完成的一个简单的关于S-AES加解密、双重加密、三重加密和CBC的可视化程序。

2. 功能概述

本程序提供以下四个功能模块：

1.普通程序（s_aes）

加密：使用 S-AES 算法对输入的明文进行加密
解密：使用 S-AES 算法对输入的明文进行解密

2.双重加密（2s_aes）

加密：使用 S-AES 算法对输入的明文进行加密
解密：使用 S-AES 算法对输入的明文进行解密
中间相遇攻击

3.三重加密（3s_aes）

加密：使用 S-AES 算法对输入的明文进行加密
解密：使用 S-AES 算法对输入的明文进行解密

4.CBC模式

加密：使用 S-AES 算法对输入的明文进行解密
解密：使用 S-AES 算法对输入的明文进行解密

3. 使用方法

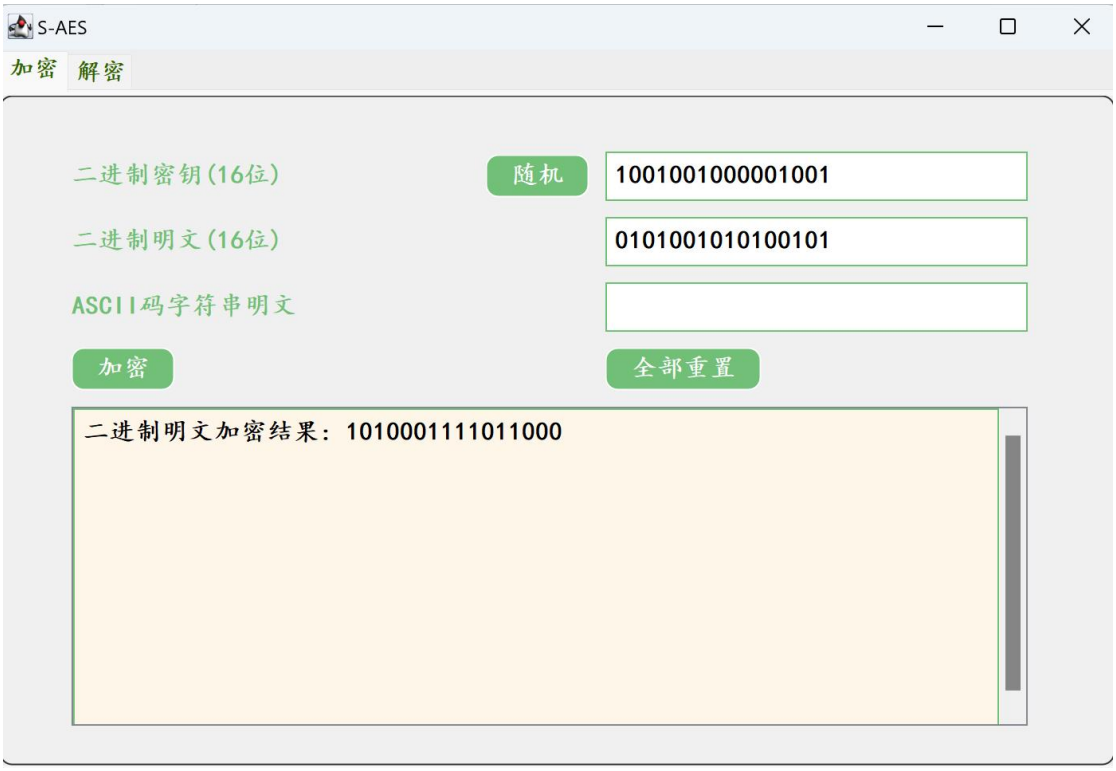
运行 UI 包下的 Main 类的main 函数，即可展示 UI 界面，进行功能展现。

3.1. 普通程序

3.1.1加密

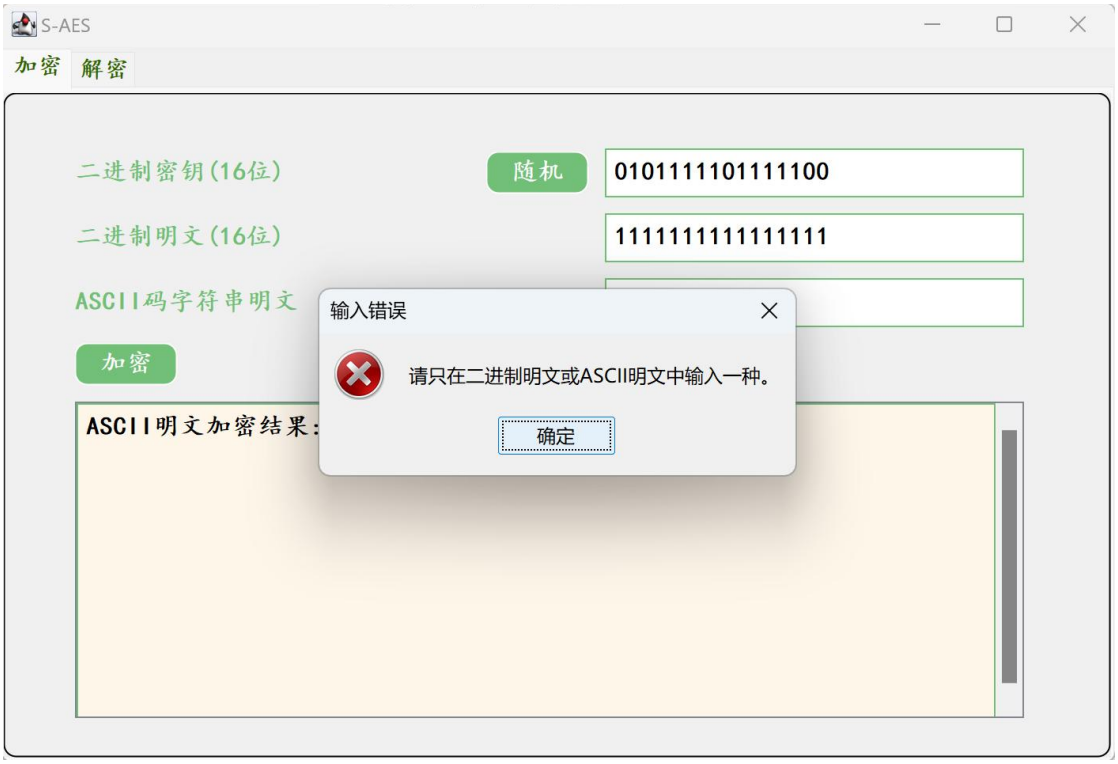
在这里可以实现16bit二进制明文和任意2bit以上的ASCII字符的加密， 但是同时只支持一种形式的加密，否则会报错；对于十六位二进制的密钥，可以选择随机生成或者手动输入，但是需要满足16位。具体展示如下：

- 对16bit二进制明文加密



- 对ASCII字符加密

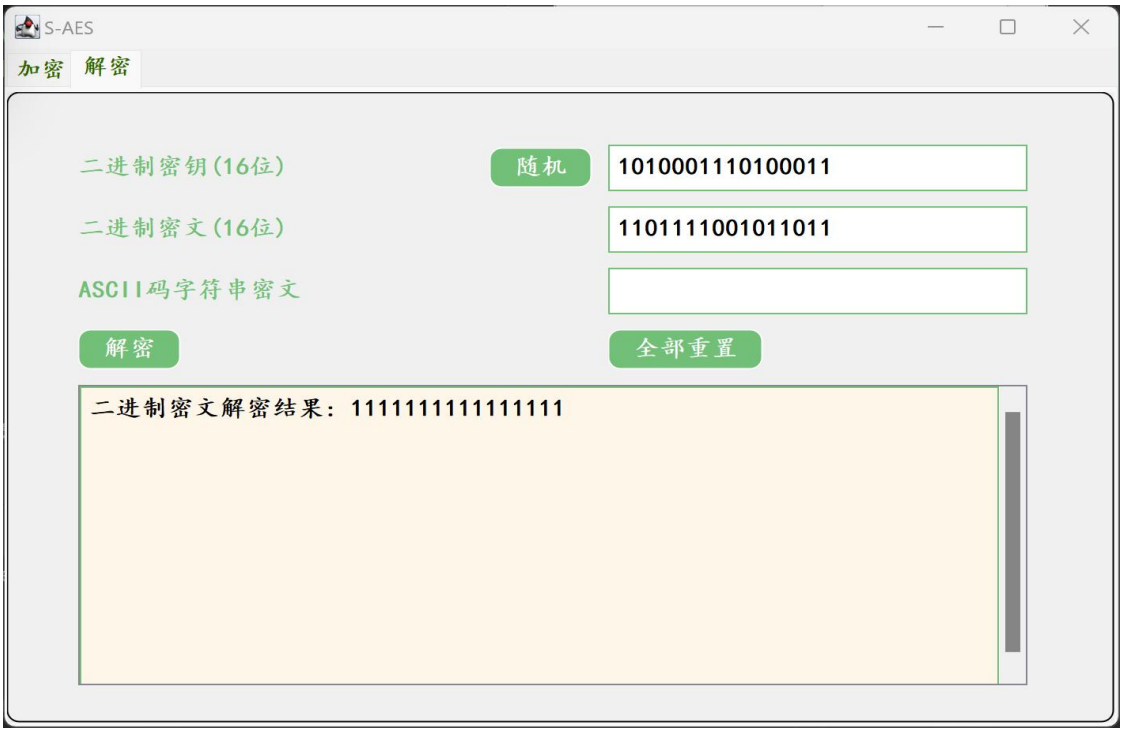
- 同时对16bit二进制明文和ASCII字符加密，会报错，提示只支持一种形式的加密



3.1.2解密

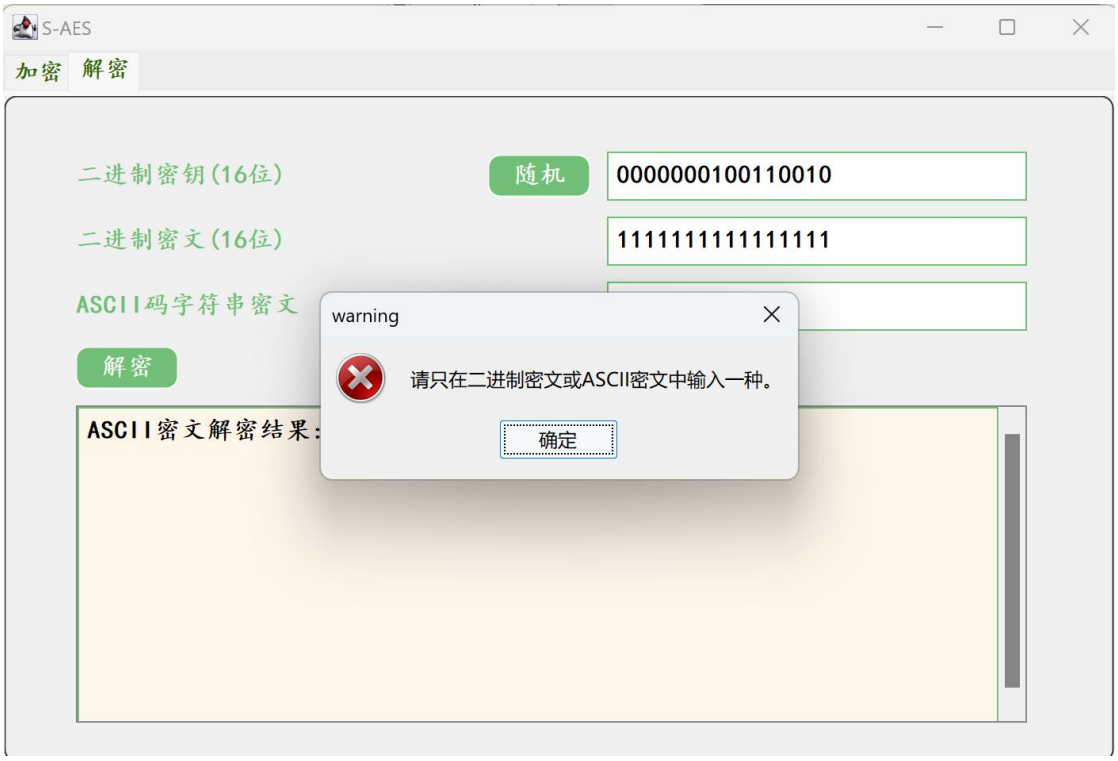
在这里可以实现16bit二进制密文和任意bit的ASCII字符的解密， 但是同时只支持一种形式的加密，否则会报错；对于十六位二进制的密钥，可以选择随机生成或者手动输入，但是需要满足16位。具体展示如下：

- 对16bit二进制密文解密，可以发现输入在加密界面输入的密钥和加密出的密文，解密出了正确的明文1111111111111111



- 对ASCII字符解密，可以发现输入在加密界面输入的密钥和加密出的密文，解密出了正确的明文abc

- 同时对16bit二进制密文和ASCII字符解密，会报错，提示只支持一种形式的解密



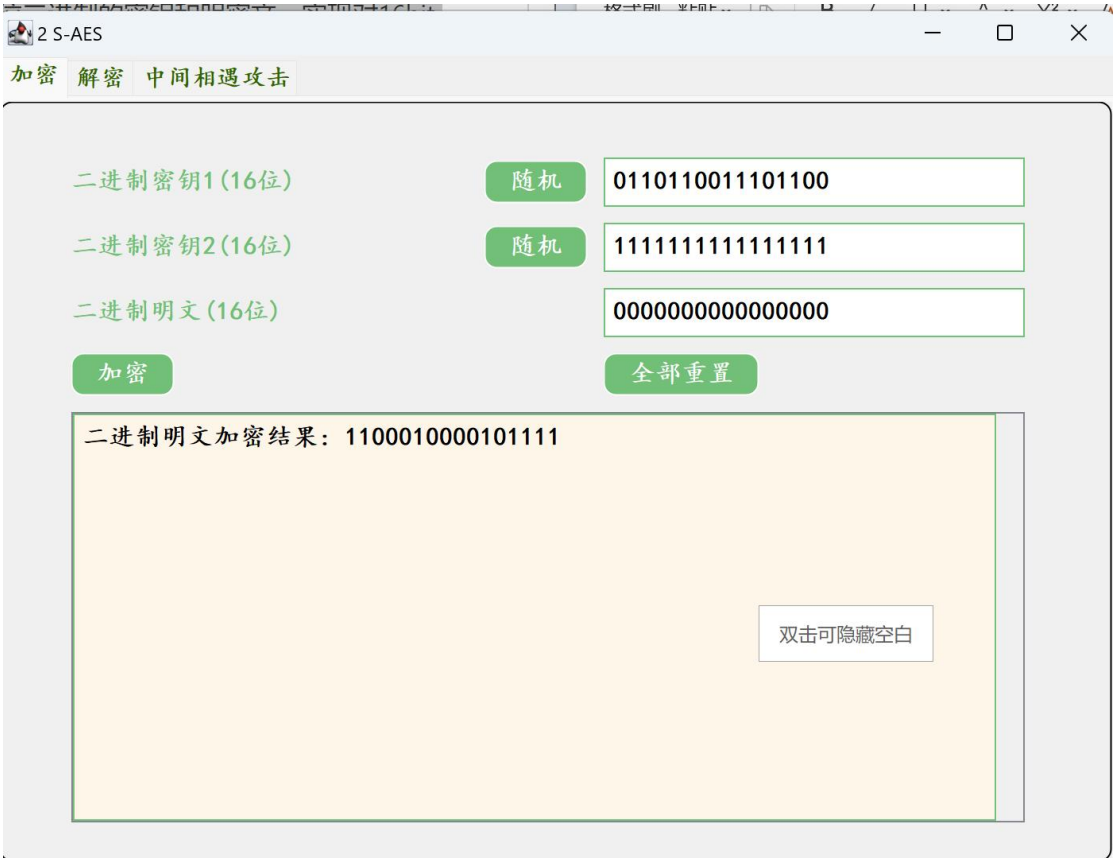
3.2 多重程序

3.2.1 2—AES

在这里可以实现输入两个十六位二进制的密钥和明密文，实现对16bit二进制明密文的加解密

3.2.1.1加密、解密

- 加密

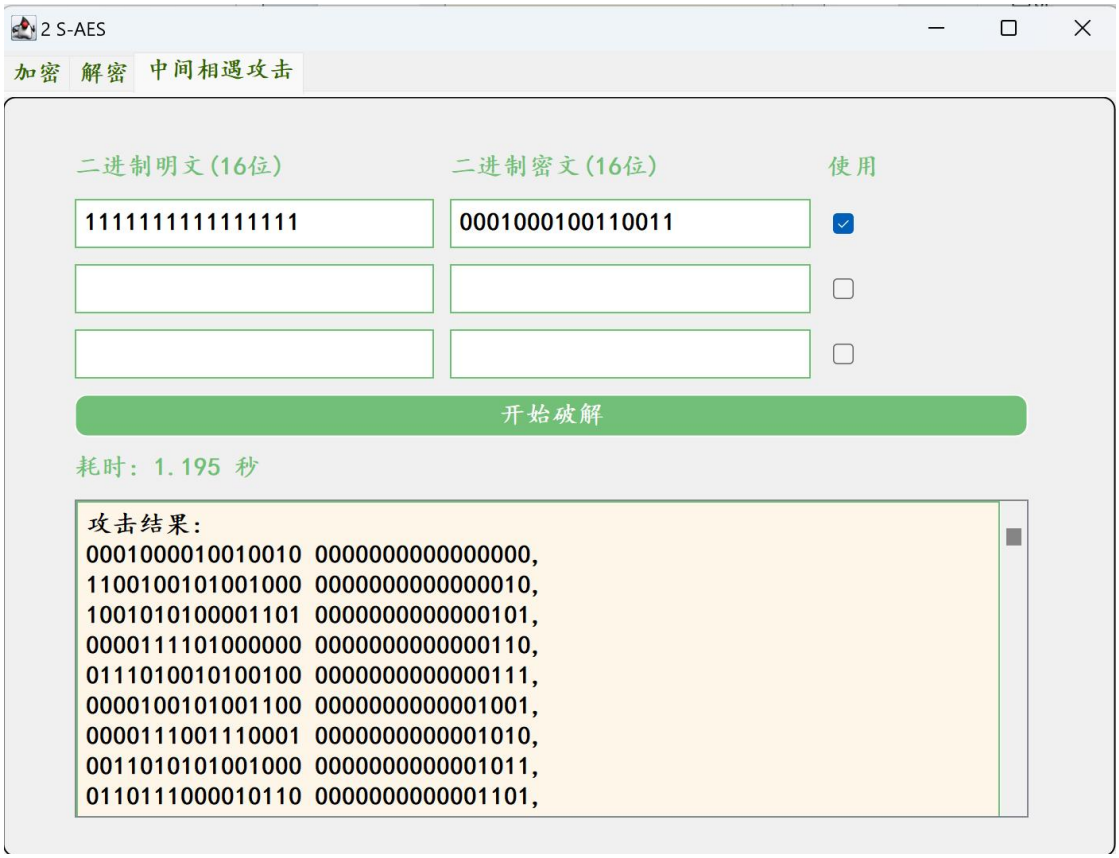


- 解密

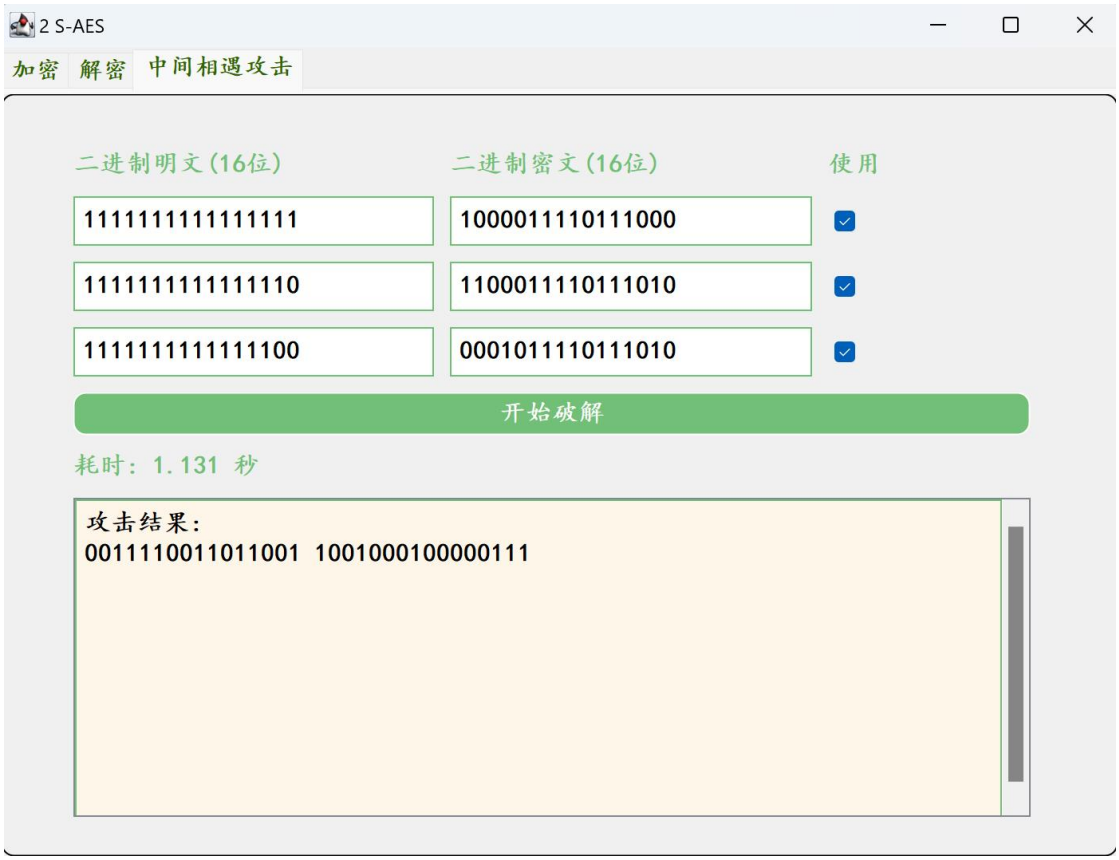
3.2.1.2中间相遇攻击

在这里可以输入一对或多对使用相同密钥的明、密文对，使用中间相遇攻击的方法可以找到正确的密钥Key(K1+K2)。

- 输入一对明密文对会出现多对可行的密钥



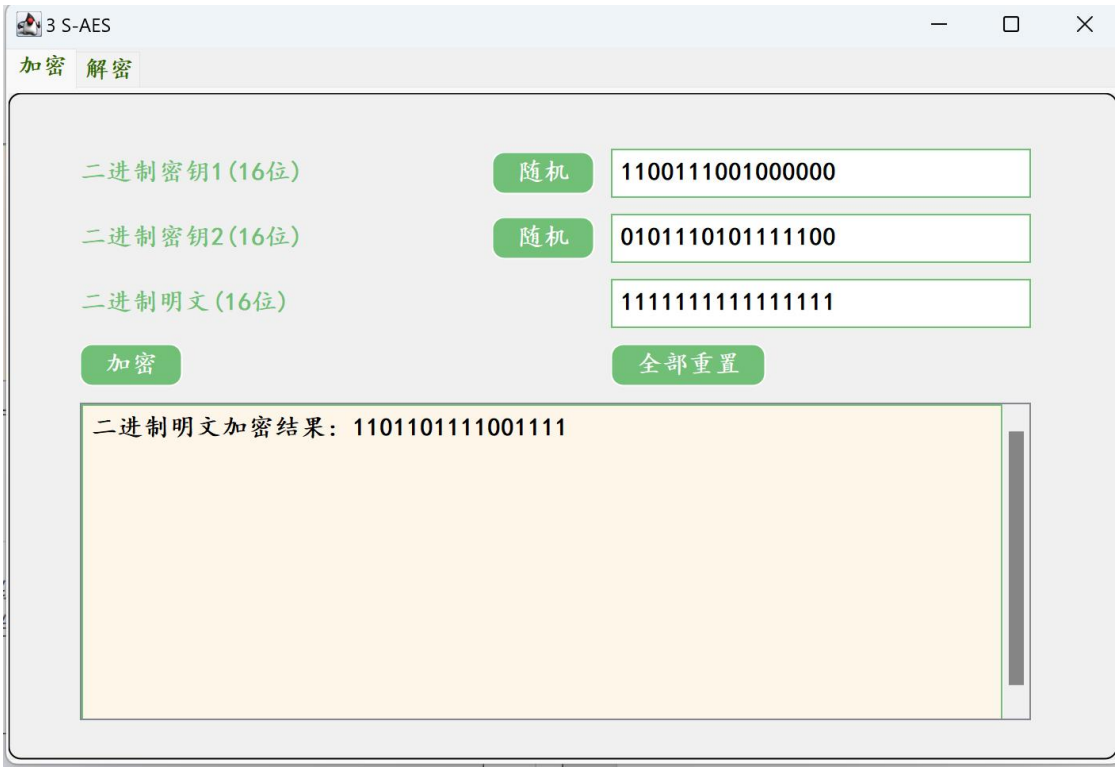
- 输入多对明密文对出现正确的一对密钥



3.2.2 3-AES

这里按照32 bits密钥Key(K1+K2)的模式进行三重加密解密，C=E(K1,D(K2,E(K1,P)))，输入两个16 bit的密钥和16bit的明密文就能对明密文加解密。

- 加密



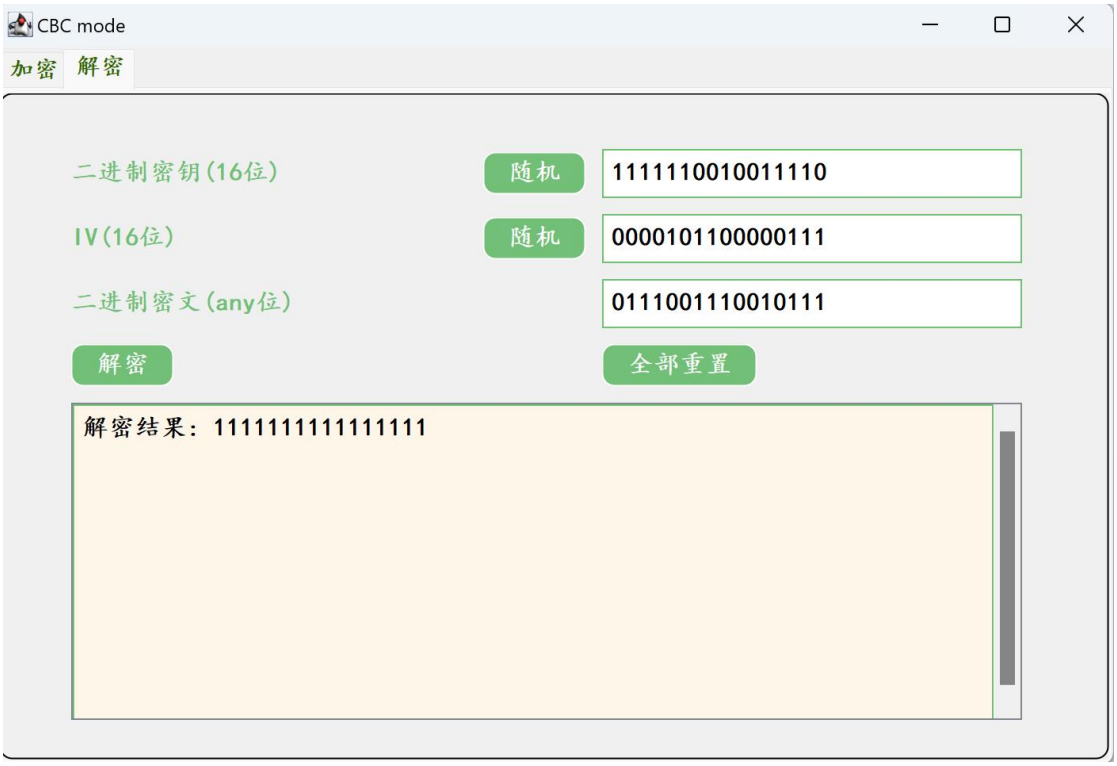
- 解密

3.3 工作模式

基于S-AES算法，使用密码分组链(CBC)模式对较长的明文消息进行加密。 输入16位的明密文和初始向量(16 bits) 和密钥对明密文加解密。

- 加密

- 解密



这里在CBC模式下进行加密，并尝试对密文分组进行修改，然后进行解密。 可以看到修改密文之前解密得到的结果是Hello,S-AES! 在对密文分组进行篡改最后一个字符之后进行解密得到的结果是Hello,S-AEC(。
发现篡改密文后会引起最后一个块的更改，用篡改后的密文进行解密会影响解密结果的正确性

```
原始明文: Hello,S-AES!
原始密文: 0010101100000001100011110010010011000101011010000000110110101001101111101011111011110111010100
在加密完成后对密文分组进行篡改最后一个字符的密文: 0010101100000001100011110010010011000101011010000000110110101001101111101011111011110111010101
对上述篡改密文进行解密的明文: Hello,S-AEC(
```

4.加密解密参数

S-AES 加解密参数

- 密钥：** S-AES算法使用一个16位的密钥。确保输入正确的密钥以保证加密解密的一致性。密钥的正确性直接影响到加密和解密的结果，因此请务必妥善保管密钥，避免泄露。
- 数据块：** S-AES操作的数据块大小为16位（2字节）。在进行加解密操作时，确保输入的数据块大小符合要求。如果数据长度不是16位的倍数，可能需要进行填充或拆分。
- 加密轮数：** S-AES通常执行两轮加密。每轮加密都会对数据块进行多次变换，增强安全性。解密过程则是加密过程的逆操作，同样需要执行两轮。
- 初始向量（IV）：** 在某些模式下（如CBC模式），S-AES可能需要使用一个16位的初始向量（IV）。IV用于在加密过程中引入额外的随机性，确保相同的明文块加密后的密文不同。确保在相同的加密会话中使用相同的IV。
- 填充模式：** 如果输入的数据长度不是16位的倍数，需要选择合适的填充模式（如PKCS7填充）。填充模式决定了如何处理数据块的最后一个块，确保其长度符合要求。
- 加密模式：** S-AES支持多种加密模式（如ECB、CBC、CFB等），不同的模式有不同的安全性和适用场景。选择合适的加密模式以满足特定的应用需求。
- 密文输出：** 加密后的数据将以密文形式输出，通常表示为十六进制字符串或二进制数据。确保密文的存储和传输安全，避免未授权访问。
- 解密输入：** 在进行解密操作时，确保输入的密文格式正确，并使用与加密时相同的密钥和参数。任何不匹配的参数都可能导致解密失败或产生无效结果。

以上参数在S-AES加解密过程中至关重要，确保正确配置这些参数以获得预期的加密和解密效果。

5. 常见问题

若点击生成明密文按钮时程序无反应，请检查输入是否正确

6. 注意事项

在使用S-AES进行数据加密时，请注意以下事项：
(1)、确保密钥的安全性，不要将密钥泄露给他人。密钥是加密和解密过程中至关重要的部分，任何未授权的访问都可能导致数据泄露。
(2)、在使用S-AES解密功能之前，请确认输入的密文确实是由本程序生成的。使用错误的密钥或尝试解密非本程序生成的密文可能会导致解密结果无效，甚至损坏数据。
(3)、S-AES设计用于小规模的数据加密需求，不建议用于大量或敏感数据的加密。对于大规模或高度敏感的数据，建议使用更强大的加密算法或结合其他安全措施。
(4)、定期更新密钥和算法版本，以应对可能出现的安全威胁。保持加密工具的更新是确保数据安全的重要步骤。
(5)、在加密过程中，确保数据的完整性和一致性。加密不应改变数据的实际内容，只是保护其免受未授权访问。
(6)、备份原始数据和加密后的数据，以防在加密或解密过程中发生意外情况。这有助于在出现问题时恢复数据。