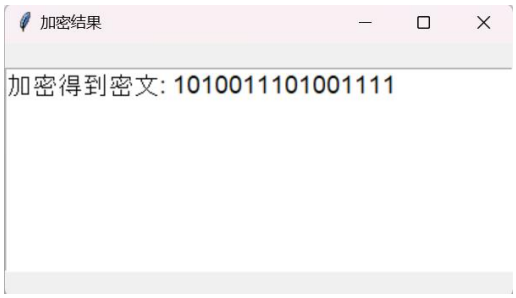
第一关：基本测试

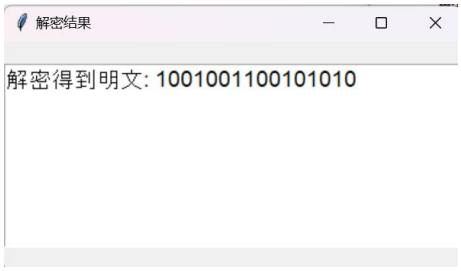
根据 S-AES 算法编写和调试程序，提供 GUI 解密支持用户交互。输入可以是 16bit 的数据和 16bit 的密钥，输出是 16bit 的密文。

加密功能：输入正确格式的明文和密钥，得到加密结果。



如果输入错误格式的明文或密钥，则会显示

解密功能：输入正确的密文和密钥，得到解密结果。



第二关：交叉测试

考虑到是"算法标准"，所有人在编写程序的时候需要使用相同算法流程和转换单元(替换盒、列混淆矩阵等)，以保证算法和程序在异构的系统或平台上都可以正常运行。
设有 A 和 B 两组位同学(选择相同的密钥 K)；则 A、B 组同学编写的程序对明文 P 进行加密得到相同的密文 C；或者 B 组同学接收到 A 组程序加密的密文 C，使用 B 组程序进行解密可得到与 A 相同的 P。

明文设置为 1001001001001111
密钥设置为 1111111100000000

第一组结果如下：

第二组结果如下:

结果表明加密结果一样，交叉测试成功。

第三关：扩展功能

考虑到向实用性扩展，加密算法的数据输入可以是 ASII 编码字符串(分组为 2 Bytes)，对应地输出也可以是 ACII 字符串(很可能是乱码)。

当我们输入 ASII 编码字符串时，通过“ascii 加密”和“ascii 解密”功能依然可以进行加密和解密。

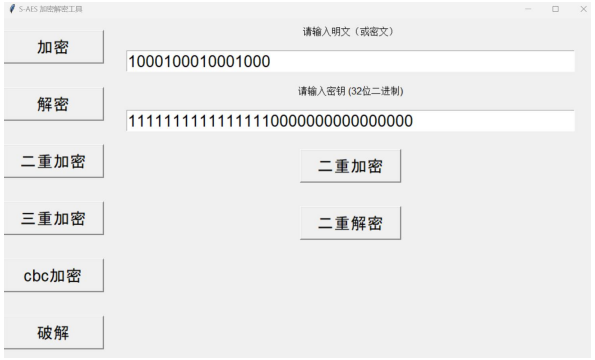
第四关：多重加密

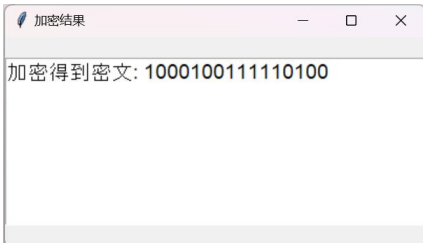
（1）双重加密

将 S-AES 算法通过双重加密进行扩展，分组长度仍然是 16 bits，但密钥长度为 32 bits。

通过二重加密功能实现

加密结果：





解密结果：

（2） 中间相遇攻击

假设你找到了使用相同密钥的明、密文对(一个或多个)，请尝试使用中间相遇攻击的方法找到正确的密钥 Key(K1+K2)。

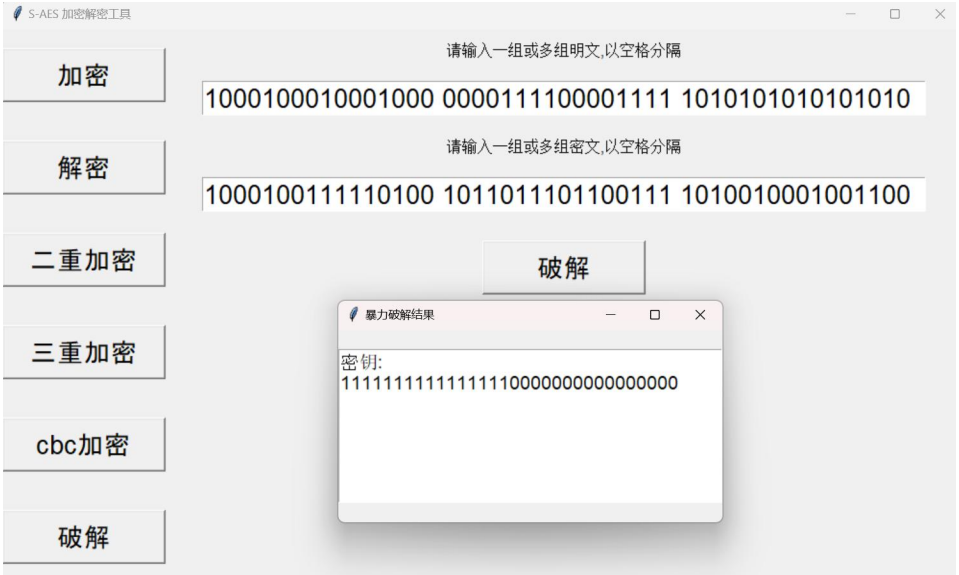
首先我们通过一个给定的 32 位密钥：11111111111111110000000000000000

获得多组明密文对：（1000100010001000，1000100111110100）

（0000111100001111，1011011101100111）

（1010101010101010，1010010001001100）

通过破解功能实现



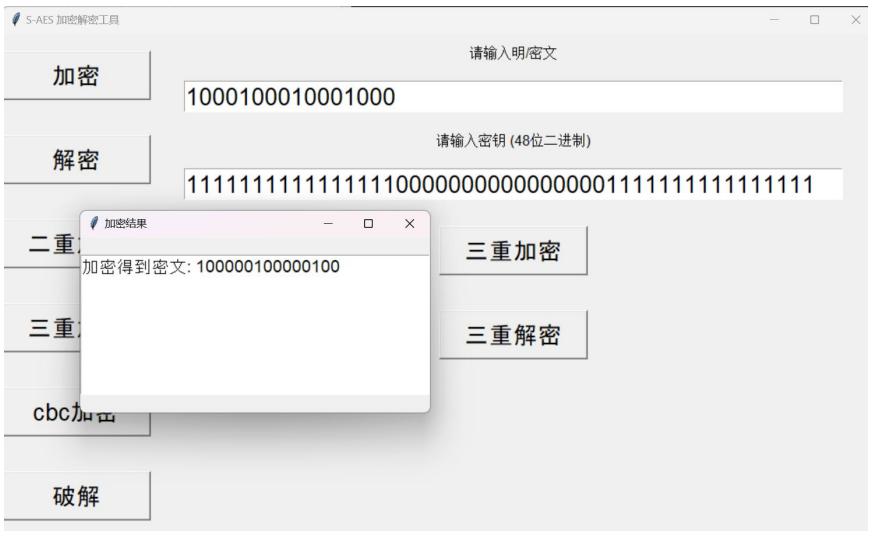
（3） 三重加密

将 S-AES 算法通过三重加密进行扩展，下面两种模式选择一种完成：

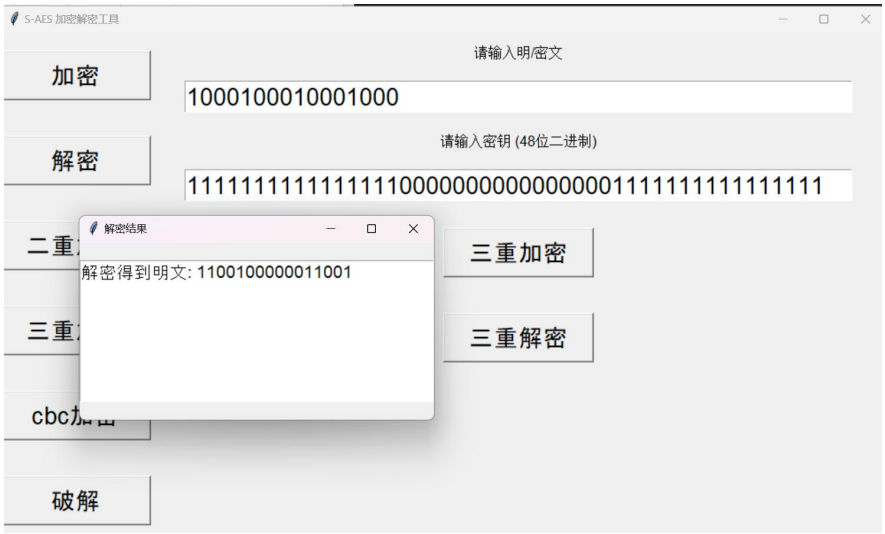
(1)按照 32 bits 密钥 Key(K1+K2)的模式进行三重加密解密，

(2)使用 48bits(K1+K2+K3)的模式进行三重加解密。

本程序采用第二种解决方式：
加密结果：



解密结果：

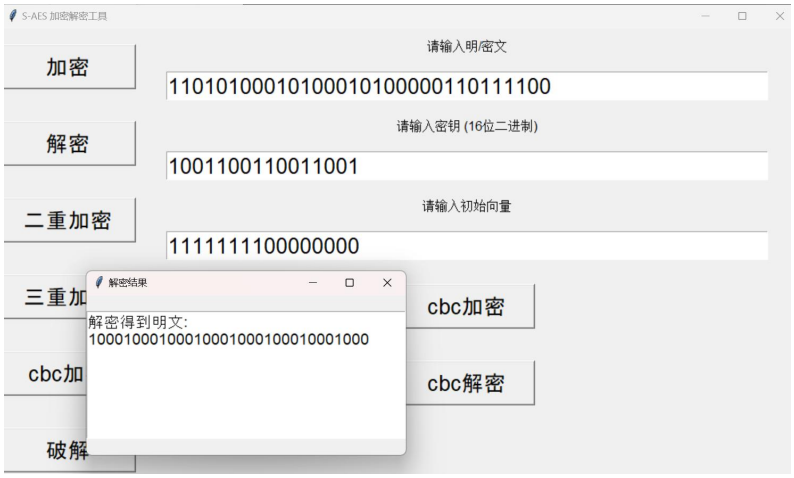


第五关：工作模式

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。注意初始向量(16 bits) 的生成，并需要加解密双方共享。
在 CBC 模式下进行加密，并尝试对密文分组进行替换或修改，然后进行解密，请对比篡改密文前后的解密结果。

加密结果：

对于密文分组进行替换或修改，然后进行解密，对比篡改密文前后的解密结果。
未修改的密文分组得到的解密结果（密文为 11010100010100010100000110111100）：



修改后的密文分组得到的解密结果（密文为：11010100010100010100000110111111）（第二组的密文进行了修改）：

由结果可得：
在 CBC 模式下，篡改密文会导致解密结果不同；
因此，修改密文的结果会对解密过程造成明显影响，充分体现了 CBC 模式在抵御密文篡改方面的特性。