S-AES 算法实现报告

一、 开发者概括

任课老师：向宏
小组代号：名字还没想好组
小组成员：杜瑞杰 20221231 王舟颖 20221459 邓湘 20221770

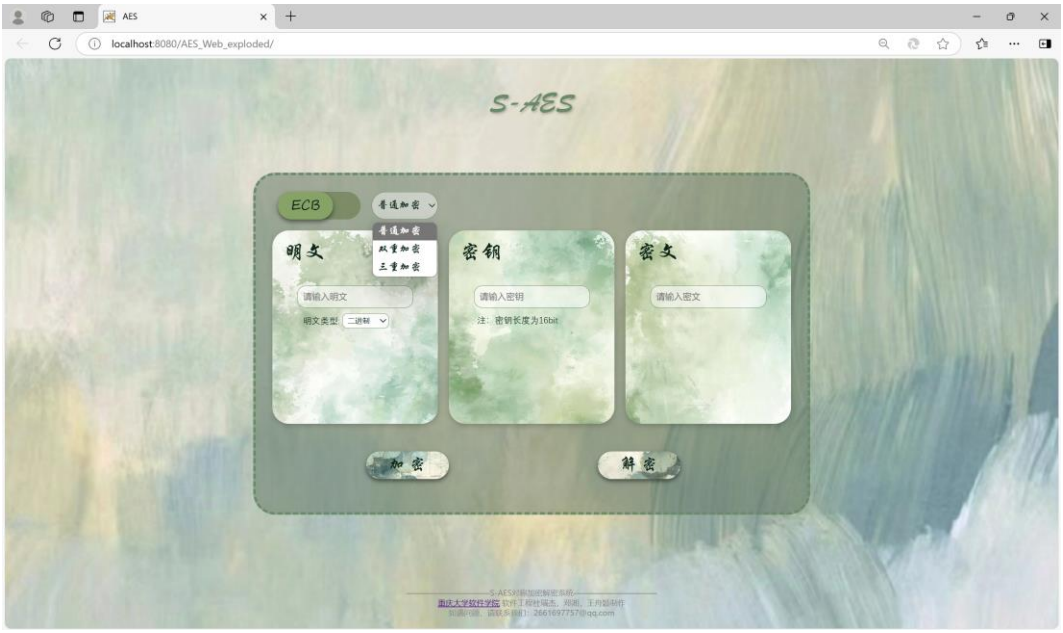
二、 测试报告

第一关：基本测试

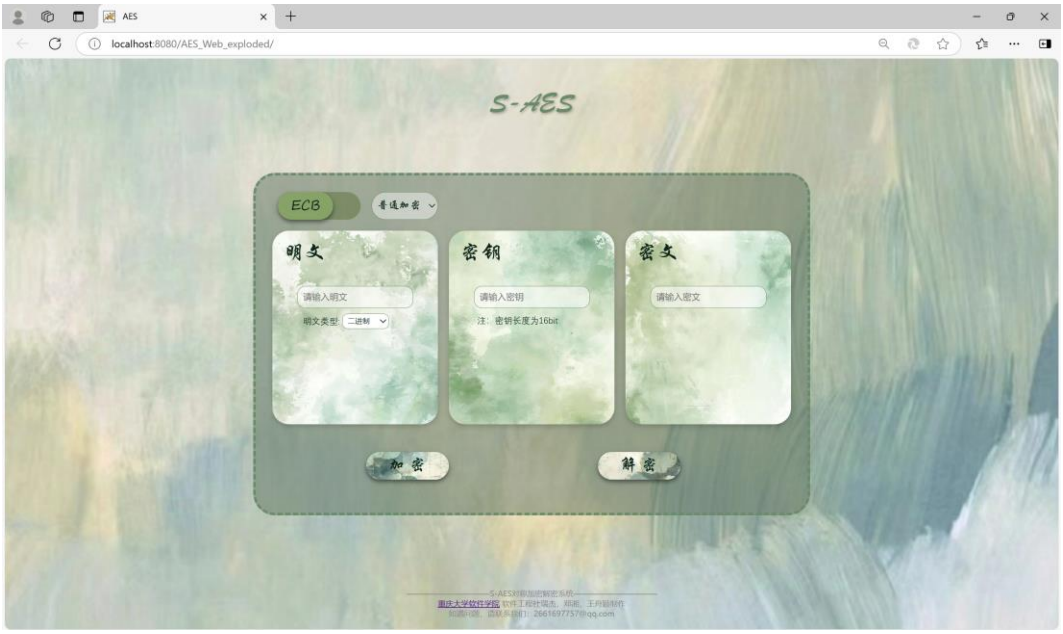
根据 S-AES 算法编写和调试程序，提供 GUI 解密支持用户交互。输入可以是 16bit 的数据和 16bit 的密钥，输出是 16bit 的密文。

1.1 用户交互界面

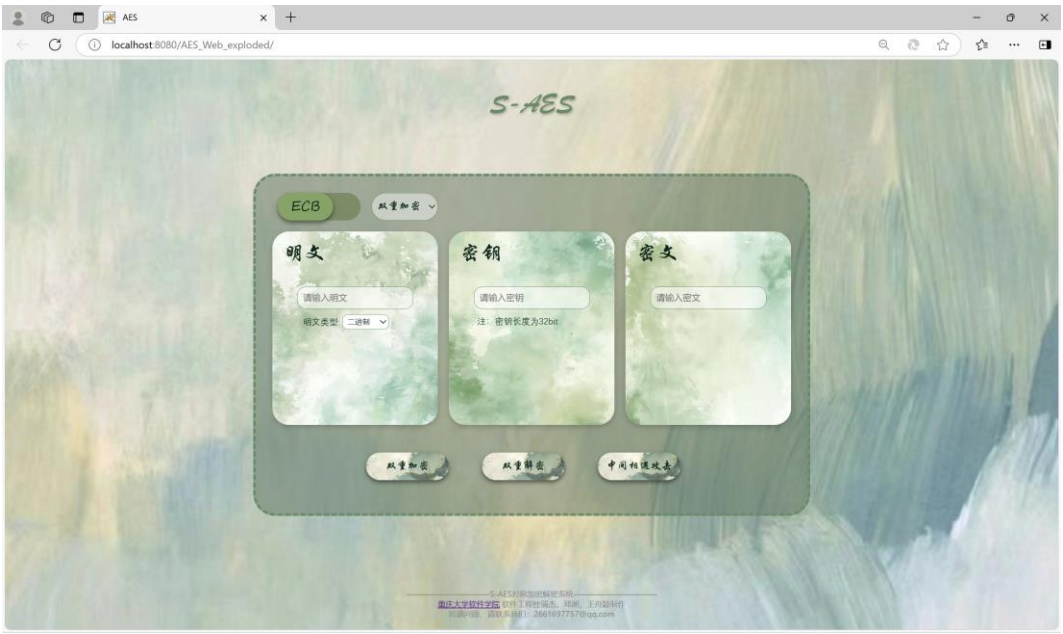
运行该程序，可得到如下界面，在该界面用户可以选择“ECB”模式或者“CBC”模式，手动添加明文与密钥，点击加密按钮进行加密。



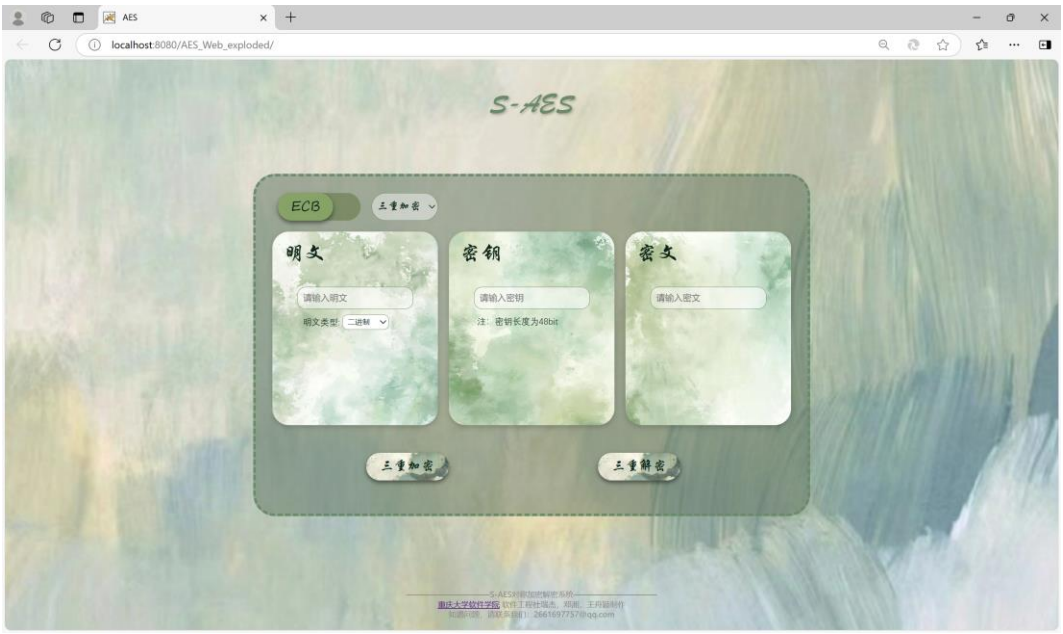
1.1.1 ECB-普通加密



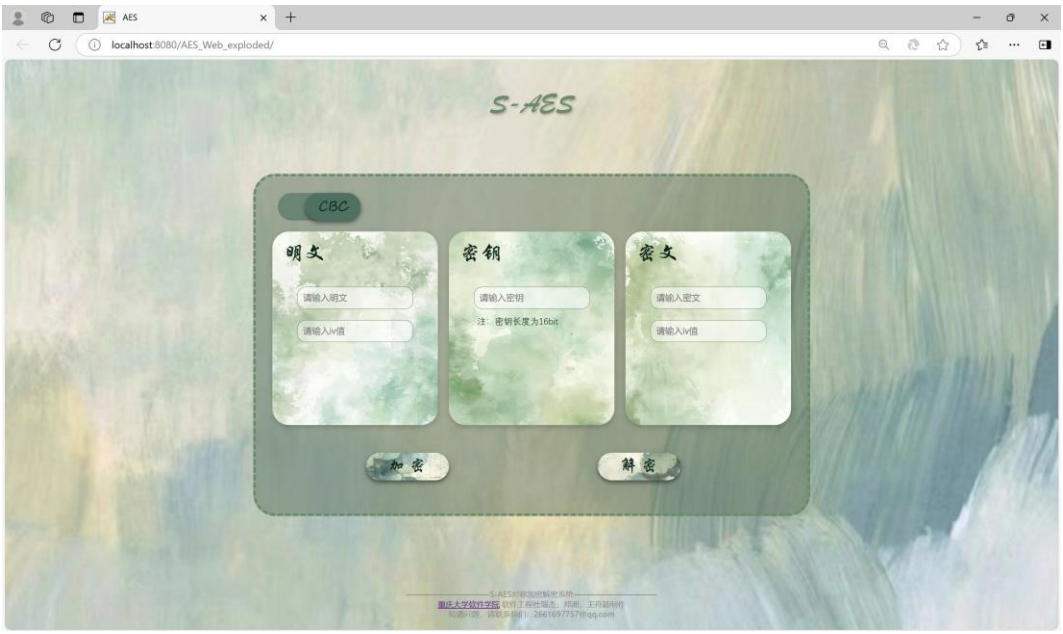
1.1.2 ECB-双重加密



1.1.3 ECB-三重加密



1.1.4 CBC 加密



1.2 加密解密测试

该系统还设计了纯二进制加密解密功能及 ASCⅡ编码字符串加密解密功能。在用户交互界面左上角可以选择。

第一关为基础测试，展示使用纯二进制数字进行加密解密的功能；

1.2.1 加密测试

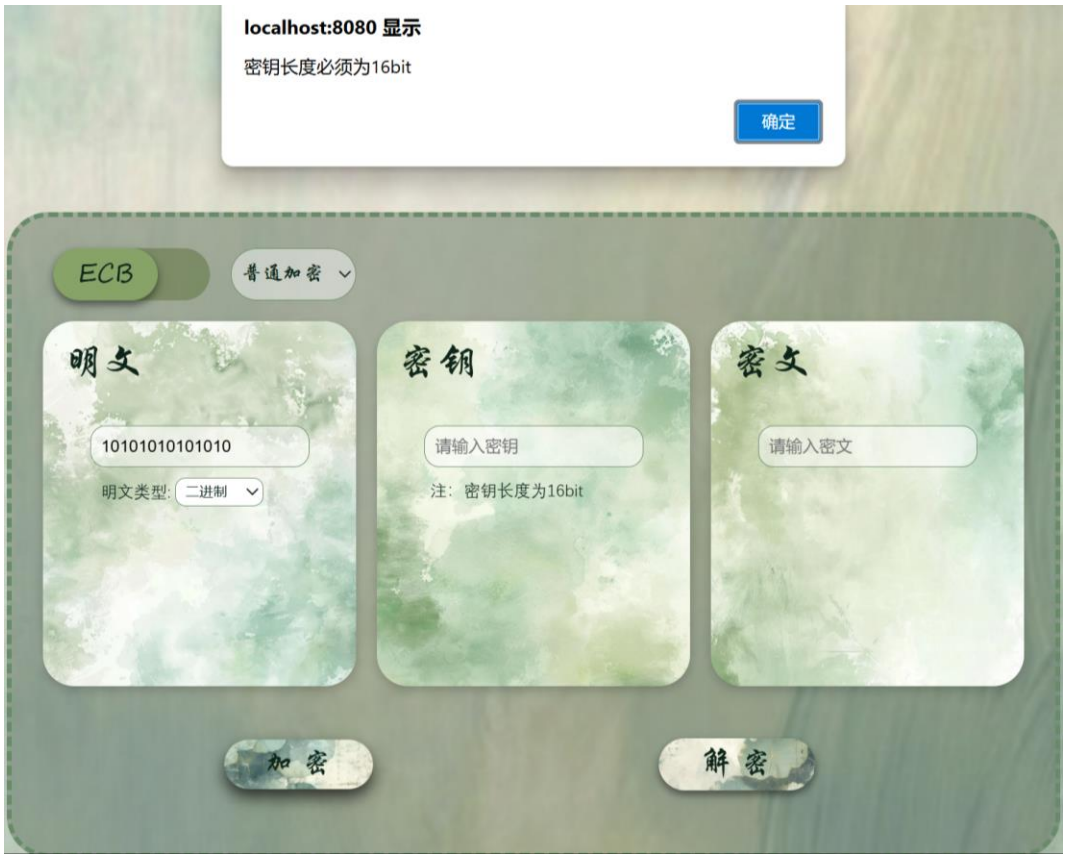
密钥：1010101010101010
明文：1010101010101010
求得密文：1010111110111000

1.2.2 解密测试

沿用**之前**密钥：1010101010101010
密文：1010111110111000
解得原文：1010101010101010

1.2.3 异常处理

当输入明文、密文或密钥不符合规范时：



1.3 总结

在本关卡中，小组成功实现了 S-AES 算法的加解密功能，支持用户输入 16 位明文和密钥并完成加解密操作。我们设计了一个直观的 GUI 界面，便于用户输入数据和查看加解密结果，并通过测试和调试确保了算法的稳定性和正确性。界面还加入了错误提示功能，帮助用户识别并纠正输入问题，进一步提升了程序的易用性和用户体验。

第二关：交叉测试

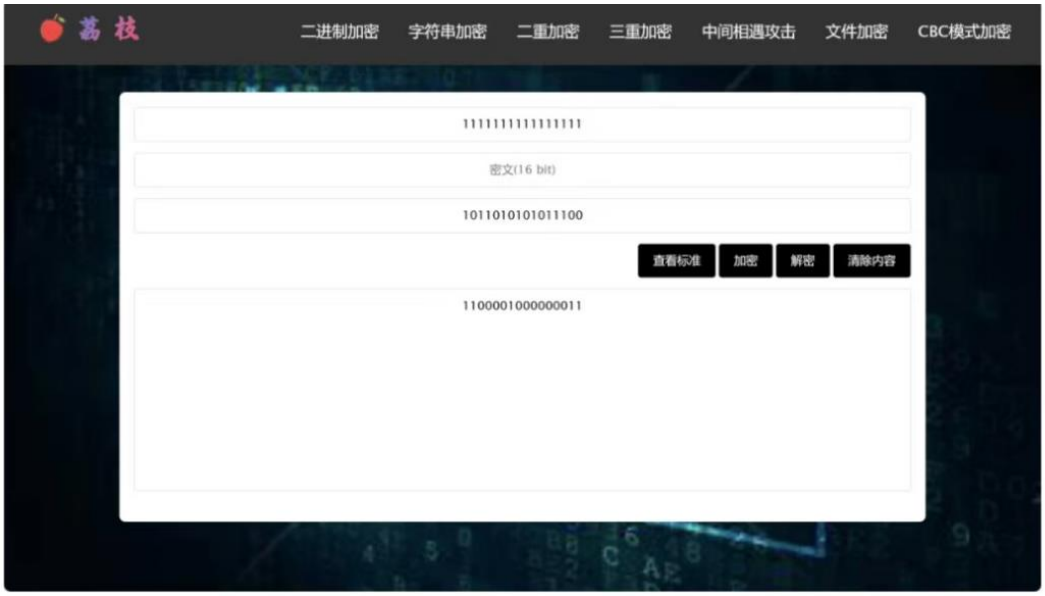
考虑到是**算法标准**，所有人在编写程序的时候需要使用相同算法流程和转换单元(P-Box、S-Box 等)，以保证算法和程序在异构的系统或平台上都可以正常运行。设有 A 和 B 两组位同学(选择相同的密钥 K)；则 A、B 组同学编写的程序对明文 P 进行加密得到相同的密文 C；或者 B 组同学接收到 A 组程序加密的密文 C，使用 B 组程序进行解密可得到与 A 相同的 P。

小组与班内小组进行交叉测试，测试结果如下：

1. **我组加密，对方解密**

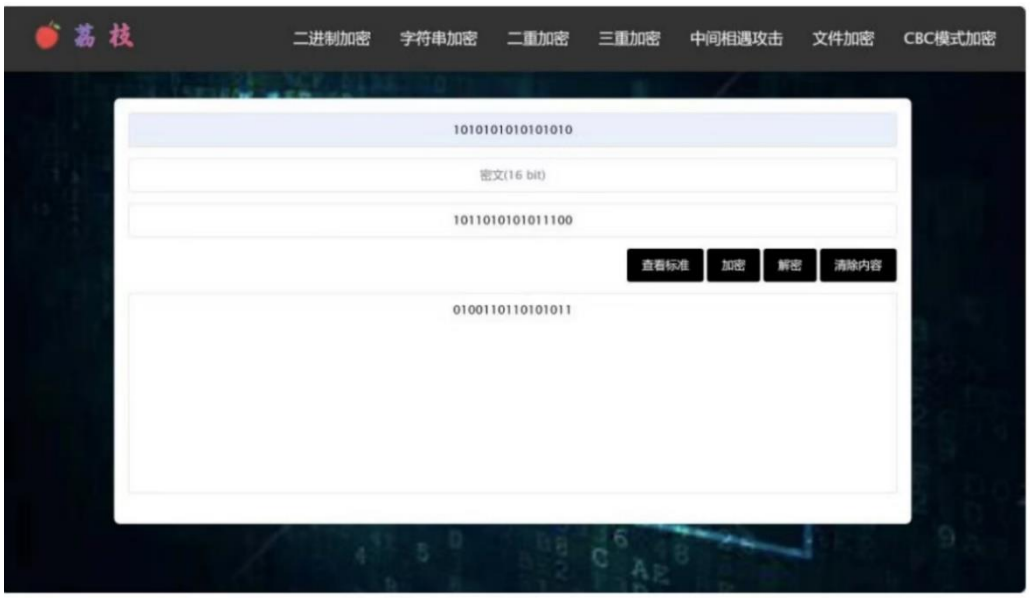
我组使用的明文为：1111111111111111
对方使用的密钥为：1011010101011100
密文为：1100001000000011

对方使用相同的密钥对密文 1100001000000011 进行解密，解密结果为1111111111111111，与我组加密时使用的明文相同。



2. 对方加密，我组解密

对方使用的明文为：1010101010101010
对方使用的密钥为：1011010101011100
密文为：0100110110101011



我组使用相同的密钥对密文 0100110110101011 进行解密，解密结果为 1010101010101010，与对方加密时使用的明文相同。

交叉测试成功，双方能够互相加密解密。

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 1Byte)，对应地输出也可以是 ACII 字符串(很可能是乱码)。

3.1 ASCII 字符加密解密测试

3.1.1 加密测试

密钥：1010101010101010
明文：27
求得密文如下图所示。

3.1.2 解密测试

密钥：1010101010101010
密文：c@
求得明文如下图所示。

3.2 总结

在本关卡中，我们扩展了 S-AES 算法的功能，实现了 ASCII 码加密和解密的功能。此功能允许用户输入 2 个字符的 ASCII 明文和 16 位密钥，程序将通过 S-AES 算法进行加密，将字符转换为密文，并在解密时还原为原始 ASCII 字符。该扩展功能为用户提供了更多选择，支持多种输入格式，使程序能够适应更广泛的应用场景。我们在 GUI 界面中加入了 ASCII 模式选择项，用户可以轻松切换输入格式，查看加解密结果。同时，程序经过充分测试，确保了 ASCII 模式下的正确性和稳定性，为用户带来更加灵活和实用的加密体验。

第四关：多重加密

4.1 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。

4.1.1 加密测试

二进制模式：

密钥：10101010101010101010101010101010

明文：1010101010101010

求得密文如下图所示。

ASCII 码模式：

密钥：10101010101010101010101010101010

明文：de

求得密文如下图所示。

4.1.2 解密测试

二进制模式：

密钥：10101010101010101010101010101010

密文：1001011001100110

求得明文如下图所示。

ASCII 码模式：

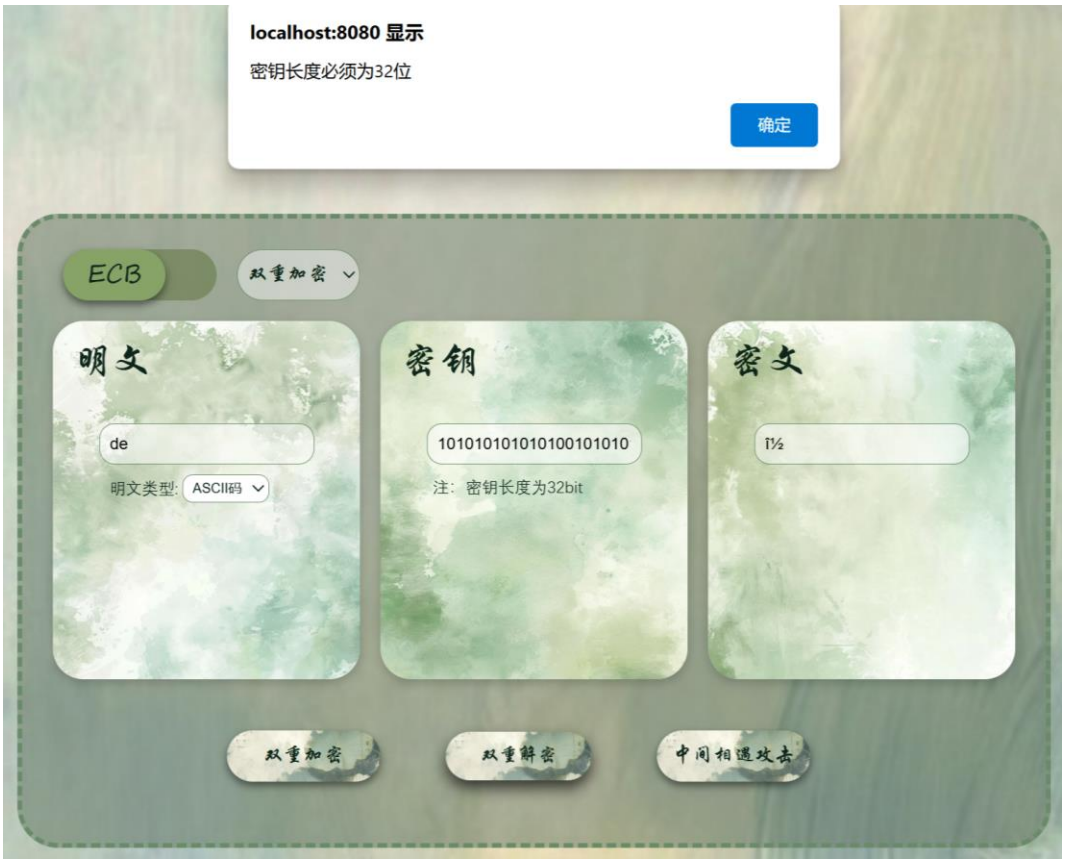
密钥：10101010101010101010101010101010

密文：î½

求得明文如下图所示。

4.1.3 异常处理

当输入明文、密文或密钥不符合规范时：



4.2 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)。

测试如下：

以二进制模式为例，输入数据如下：

明文：1010101010101010，

密钥：00000000000000011110001101000011O，

加密所得密文为：1111111111111111

输入明文 1010101010101010，密文 1111111111111111，点击中间相遇攻击按钮，找到 20 个可能的密钥，生成的密钥如下：

11011110001000100101110111001011， 11111101001111111111010111011111，
00000000000000011110001101000011O， 10101100111000011010001001101100，
11110001001011011111110011001101， 00000000000010001101100100101101，
00000000000010011001100100000110， 11001101011010011000101000000101，
00000000000011100010001011001101， 00010101000001101100111011001011，
10101110100011100010011001111011， 00000000000100111110001110001100，
00000000000101001010011001110111， 00100000010110111000001001100001，
10100101101011101110100011100001， 11110101110000110000010101010111，
00000000000110101011111001110010， 00000000000011011100110011001000O，
00000000000111011111100011001100， 10110111001110001101000110001110

其中第 3 个密钥为加密时所使用的密钥。（注：为保证运行速度，设置系统在找到 20 个密钥后停止寻找。如果需要更多的可能密钥，可对代码进行修改。）

运行截图如下：

4.3 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。

选择 48bits(K1+K2+K3)的模式。

4.3.1 加密测试

二进制模式：
密钥：10101010101010101010101010101010101010101010101010101010
明文：1010101010101010
求得密文如下图所示。

ASCII 码模式：
密钥：10101010101010101010101010101010
明文：a?
求得密文如下图所示。

4.3.2 解密测试

二进制模式：

密钥：1010101010101010101010101010101010101010101010101010

密文：0100111001000000

求得明文如下图所示。

ASCII 码模式：

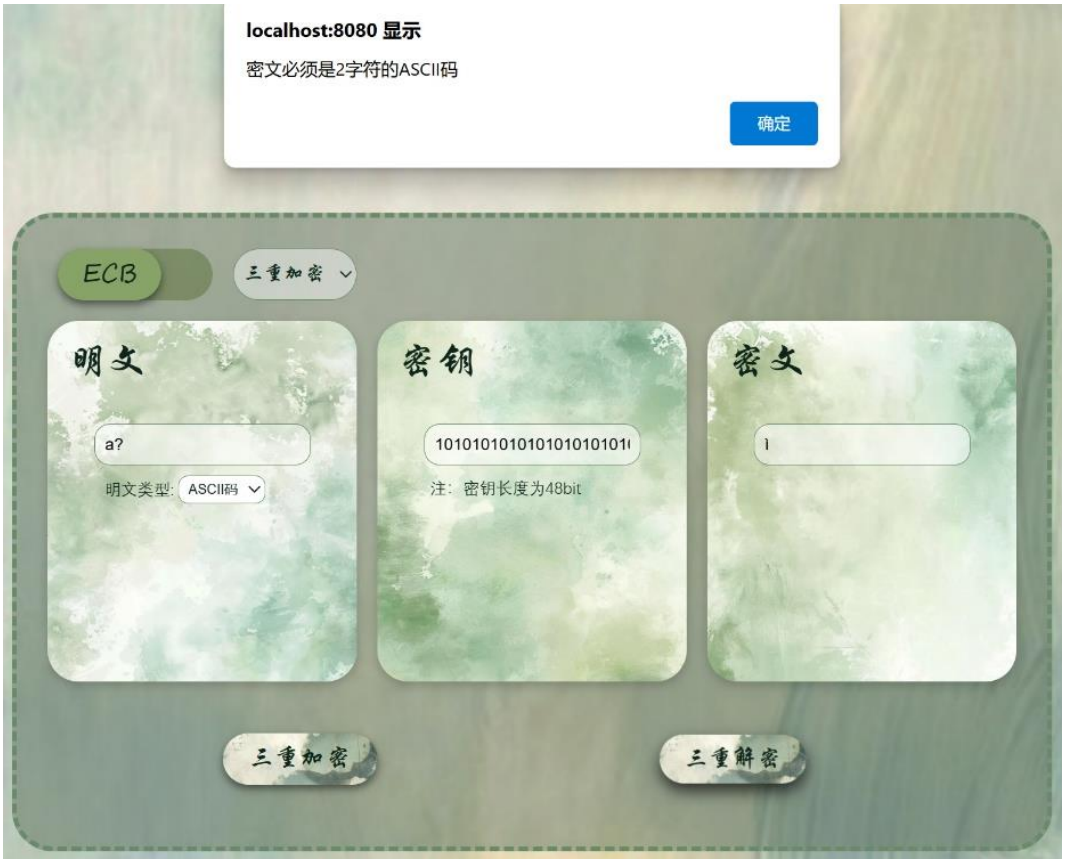
密钥：1010101010101010101010101010101010101010101010101010

密文：iN

求得明文如下图所示。

4.3.3 异常处理

当输入明文、密文或密钥不符合规范时：



4.4 总结

在本关卡中，我们扩展了 S-AES 算法的功能，实现了双重加密、中间相遇攻击和三重解密的功能。允许用户输入 32bits 或 48bits 的密钥，对明文进行双重、三重加密或对密文进行双重、三重解密；支持用户在获得明文和其对应密文后，通过中间相遇攻击找到密钥。以上功能对于二进制和 ASCII 码均适用。

扩展功能为用户提供了更多选择，加强了加密算法的安全性，使 S-AES 算法能够适应更广泛的应用场景。我们在 GUI 界面中按钮根据加解密模式进行切换的效果，使网页更便于使用。同时，程序经过充分测试，确保了多重加密模式的正确性和稳定性，为用户带来更加灵活和实用的加密体验。

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。

5.1 加密测试

密钥：1010101010101010
明文：10101010101010101111111111111111
iv 值：0100000000000010
求得密文为 11001111101110000110111001010000，运行情况如下图所示。

5.2 解密测试

密钥：1010101010101010
密文：11001111101110000110111001010000
iv 值：0100000000000010
求得明文为 10101010101010101111111111111111，运行情况如下图所示。

5.3 篡改密文测试

现有以下数据：

明文：10101010101010101111111111111111
密钥：1010101010101010
密文：11001111101110000110111001010000
iv 值：0100000000000010

将密文第一位修改为 0，其它位不变，用相同的密钥和 iv 值对修改后的密文进行解密，解密所得的明文为 01111010101000110111111111111111，与原明文有较大差异。

5.4 总结

在本关卡中，我们基于 S-AES 算法，实现了使用密码分组链(CBC)模式进行加解密的功能，支持用户输入自定义的 iv 值，提升了加密算法的安全性。

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。篡改密文前后的解密结果差异较大，说明 CBC 模式的 S-AES 算法的安全性较高。