**测试报告**

1. **测试意义**
   1. **验证算法的正确性**

测试可以验证S-AES算法是否按照设计规格正确实现。这包括验证加密和解密过程的正确性，以及确保算法的输出符合预期。通过测试，可以评估S-AES算法的性能，包括加密和解密的速度，以及在不同环境和条件下算法的表现。

**1.2 发现潜在的缺陷和漏洞**

测试有助于发现S-AES算法实现中的潜在缺陷和漏洞，从而进行修复，提高算法的安全性和可靠性。通过全面的测试，可以提高S-AES算法实现的可信度，增加用户和开发者对算法的信任。

**1.3 确保算法的兼容性**

测试还可以确保S-AES算法在不同的平台和环境中能够正确执行，包括不同的操作系统、硬件配置和编程语言实现。测试结果可以为算法的进一步改进和优化提供依据，帮助开发者识别算法的瓶颈和改进点。

通过这些测试，可以确保S-AES算法实现的质量和可靠性，为算法的广泛应用提供基础。

1. **测试目的**

**2.1 功能测试**

验证S-AES算法实现的所有功能是否按照设计文档和规格说明书正确执行。这包括加密和解密功能、密钥生成和处理等。

**2.2 性能测试**

评估S-AES算法在不同条件下的性能表现，包括处理速度、资源消耗和响应时间等。

**2.3 兼容性测试**

确保S-AES算法在不同的平台和环境中能够正确执行，包括不同的操作系统、硬件配置和编程语言实现。

**2.4 回归测试**

在算法实现发生变更后，进行回归测试以确保新的变更没有引入新的缺陷，并且原有功能仍然正常工作。

**2.5 边界条件测试**

测试S-AES算法在边界条件下的表现，包括处理非常短或非常长的输入数据、异常输入等情况。

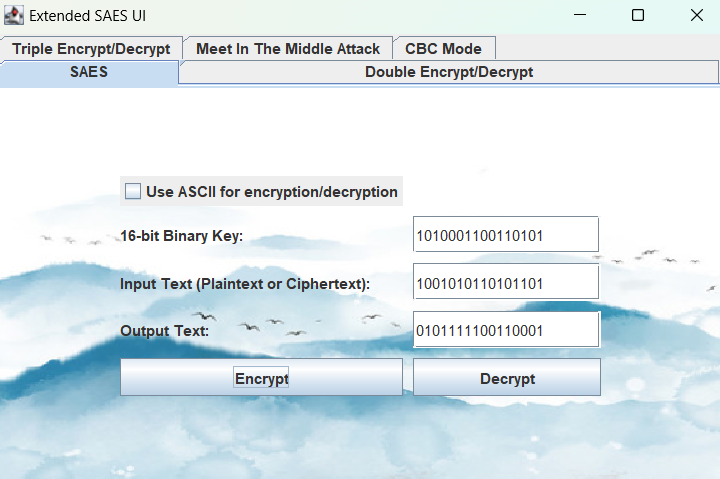
1. **测试步骤**
   1. **基本测试**
      1. 用户交互界面

实现了基本的GUI界面，可以在结果处展示“加密”和“解密”的结果。

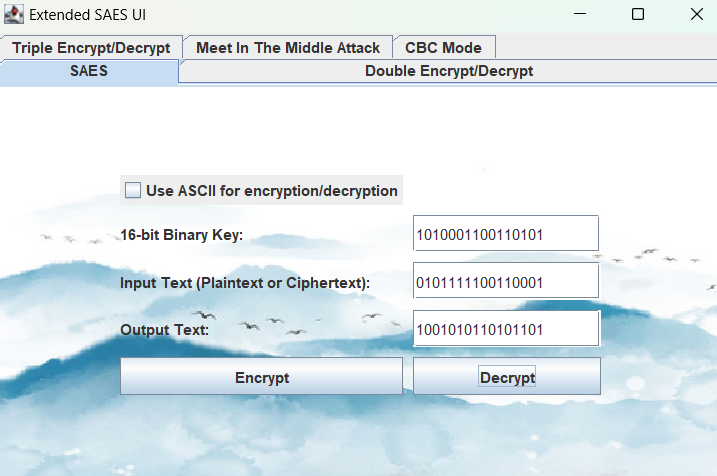
3.1.2 加密解密测试

本系统提供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，不勾选表示使用纯二进制数字进行加密解密。（此处选择不勾选）

①加密



②解密



3.1.3 测试总结

在本关卡中，我们小组成功完成了任务要求，主要做到了以下几点：

①通过实现GUI界面，我们学习了如何使用图形界面库（如Swing、awt）来创建用户交互界面，以及如何处理用户输入和展示结果。

②通过实现密钥生成功能，我们学习了随机数生成和密钥处理的基本原理，以及如何在程序中实现这些功能。

③通过实现加密解密功能，我们学习了加密解密的基本原理，以及如何在程序中实现这些功能。我们还学习了如何处理不同类型的输入数据，包括纯二进制数字和ASCII编码字符串（在以下的关卡中）。

* 1. **交叉测试**

项目要求我们实现S-AES算法，并确保不同组同学编写的程序在使用相同的密钥时，对同一明文进行加密得到的密文相同，以及对同一密文进行解密得到的明文相同。这要求我们严格遵循S-AES算法的标准流程和转换单元，以确保算法的正确性和一致性。

* + 1. A方主动加密，由B方进行协作测试；B方主动加密，由A方进行协作测试

①A方加密结果：密钥：0000000000000000

明文：1111101001010000

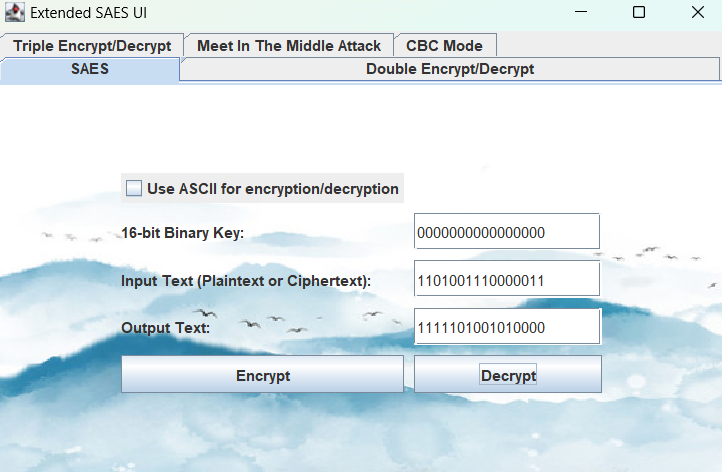
求得密文：1101001110000011



B方解密结果：密钥：0000000000000000

密文：1101001110000011

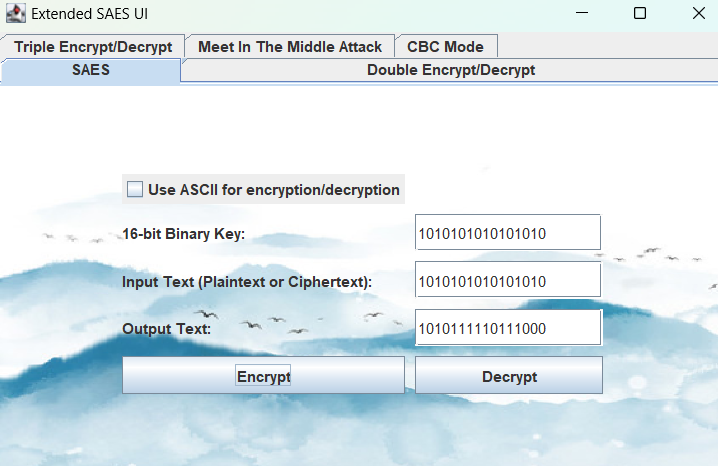
求得明文：1111101001010000



②B方加密结果：密钥：1010101010101010

明文：1010101010101010

求得密文：1010111110111000



A方解密结果：密钥：1010101010101010

密文：1010111110111000

求得明文：1010101010101010



* + 1. 总结

在本关卡中，我们小组成功完成了任务要求，主要做到了以下几点：

①算法一致性：我们学习了如何确保不同实现的S-AES算法在加密和解密过程中保持一致性，即使在不同的系统或平台上运行；同时我们学习了如何在团队中进行协作测试，确保不同成员编写的程序能够正确地相互操作。

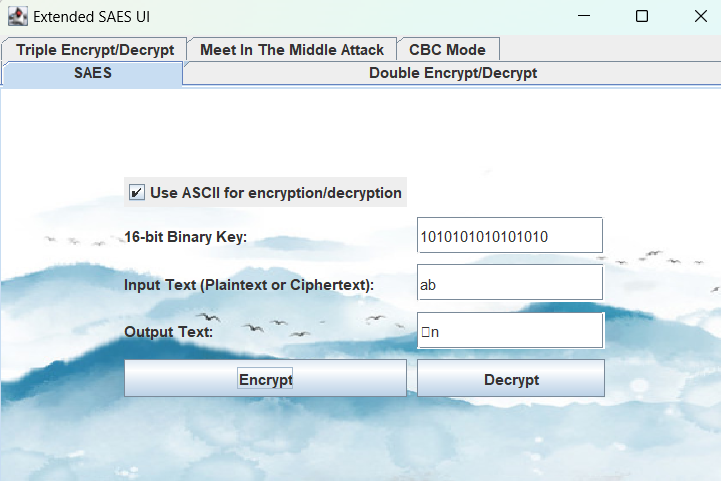
②加密解密流程及错误排查：我们深入理解了S-AES算法的加密和解密流程，包括转换原理和实现方法。在测试过程中，我们学习如何排查和解决不同实现之间的差异，确保算法的一致性和正确性。

* 1. **拓展功能**

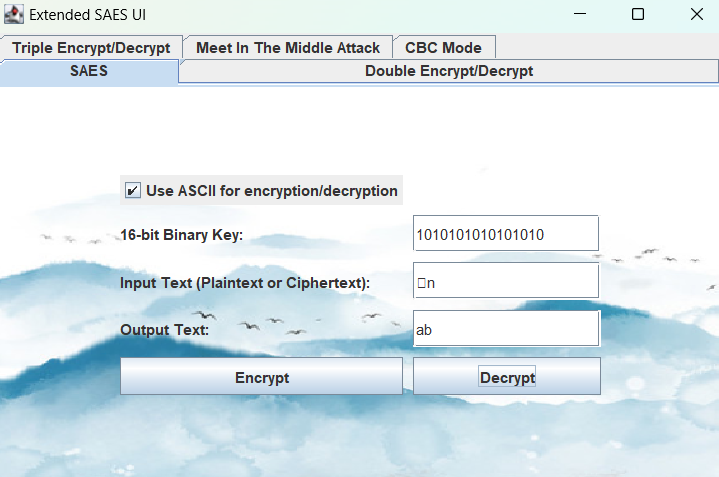
加密算法的数据输入可以是ASII编码字符串(分组为2 Bytes)，对应地输出也可以是ACII字符串(可能是乱码)。本项目提供纯二进制加密解密功能及ASCⅡ编码字符串加密解密功能，在用户交互界面左上角可以选择，在本关卡将展示使用ASCⅡ编码字符串进行加密解密的功能。

* + 1. 加密解密测试

①加密：



②解密：



* + 1. 总结

在本关卡中，我们小组成功完成了任务要求，主要做到了以下几点：

①安全性考虑：我们测试了如何生成随机密钥，并将其应用于ASCII编码字符串的加密解密中，了解密钥在加密过程中的重要性，以及如何确保密钥的安全性。

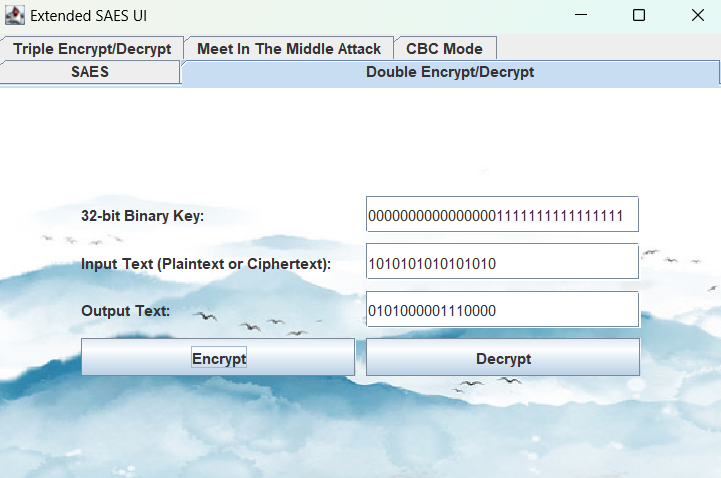
②数据处理：我们学习了如何将ASCII编码字符串转换为二进制数据，并使用S-AES算法进行加密解密。在测试过程中，我们学习了如何处理加密解密过程中的异常情况，确保程序的健壮性。

③通过这些测试，我们不仅提高了对S-AES算法处理不同类型数据的能力，还加深了对加密解密过程的理解，为我们的编程技能和加密解密知识打下了坚实的基础。

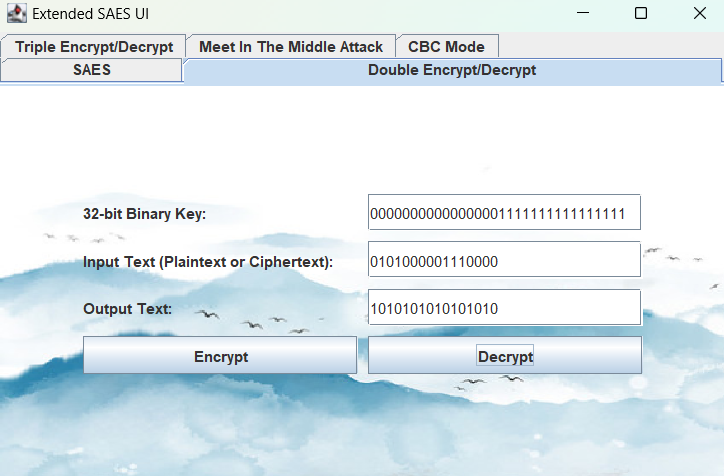
* 1. **多重加密**
     1. 双重加密

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

加密：

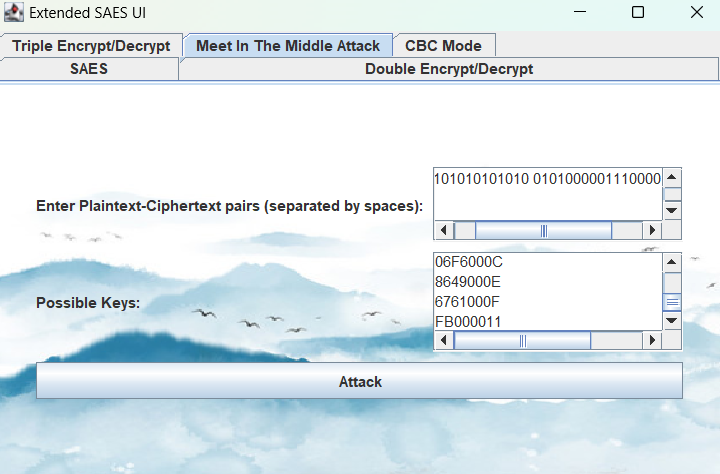


解密：



* + 1. 中间相遇攻击

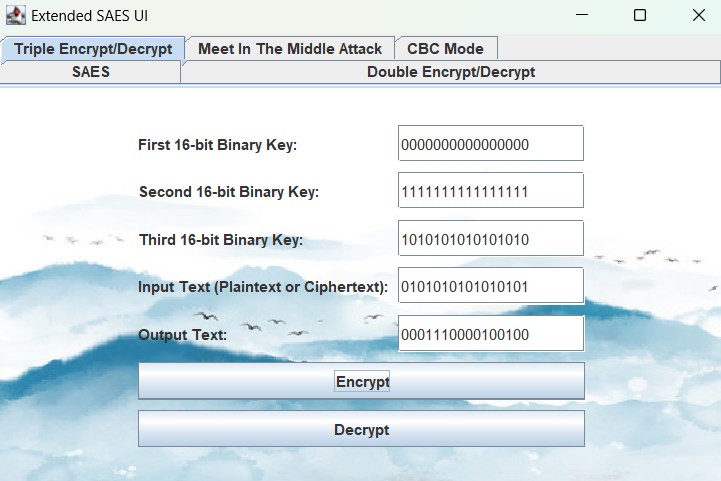
结社找到了使用相同密钥的明密文对，使用中间相遇攻击的方法找到正确的密钥（K1+K2）。



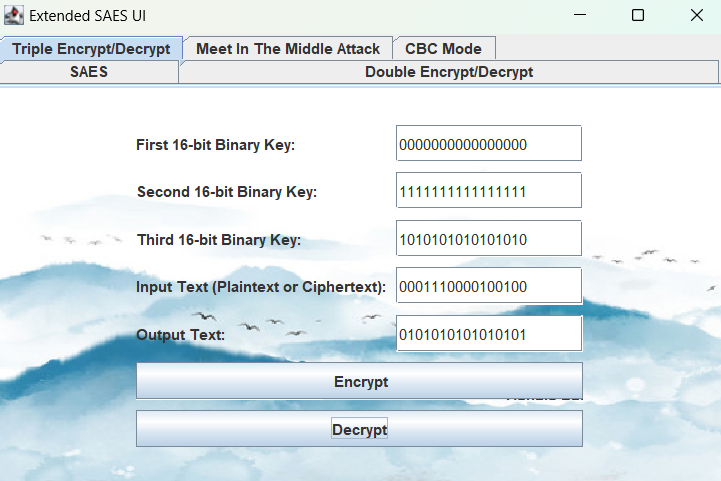
3.4.3三重加密

将S-AES算法通过三重加密进行扩展，分组长度仍然为16bits，但密钥长度为48bits。

加密：



解密：



3.4.4 总结

在本关卡中，我们小组成功完成了任务要求，主要做到了以下几点：

①多重加密理解：是一种增强数据安全性的技术，它通过使用相同的密钥对数据进行多次加密来提高加密强度。使用S-AES算法对密钥进行扩展，生成多个子密钥。接着，对明文数据进行多次加密。每次加密后，得到的密文作为下一轮加密的输入。这个过程重复进行，直到完成预定次数的加密，得到最终的密文。这个密文比单次加密的密文更难破解，因为攻击者需要破解多次加密层。解密时，需要使用与加密相同的密钥和子密钥，按照加密的逆序进行解密。每次解密后，得到的明文作为下一轮解密的输入，直到完全恢复原始明文。

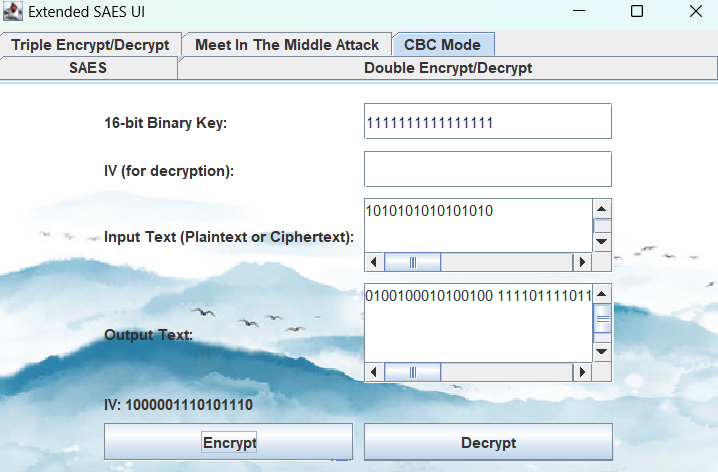
②安全性：多重加密可以显著提高加密数据的安全性，但同时也会增加计算负担和延迟。在实际应用中，需要根据数据的敏感性和安全需求，平衡性能和安全性。S-AES的多重加密特别适合于对安全性要求较高的场景，如军事通信、金融交易等。

③**轮函数应用**：在每次加密过程中，应用S-AES的轮函数，包括字节替换、行移位、列混淆和轮密钥加等步骤。

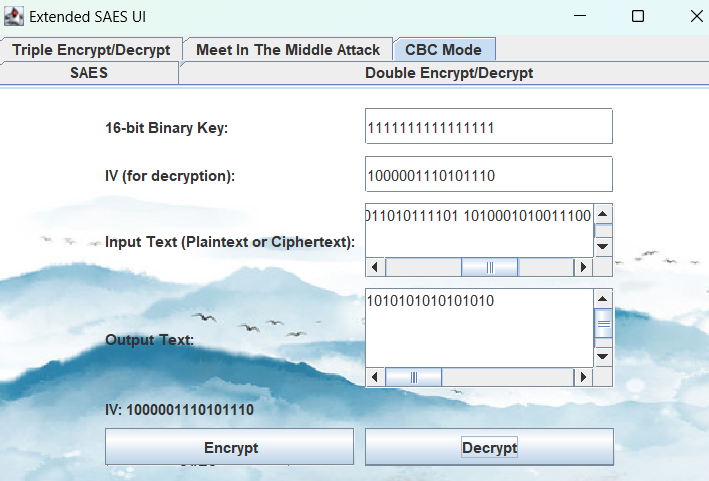
* 1. **CBC模式**

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，对比篡改密文前后的解密结果。

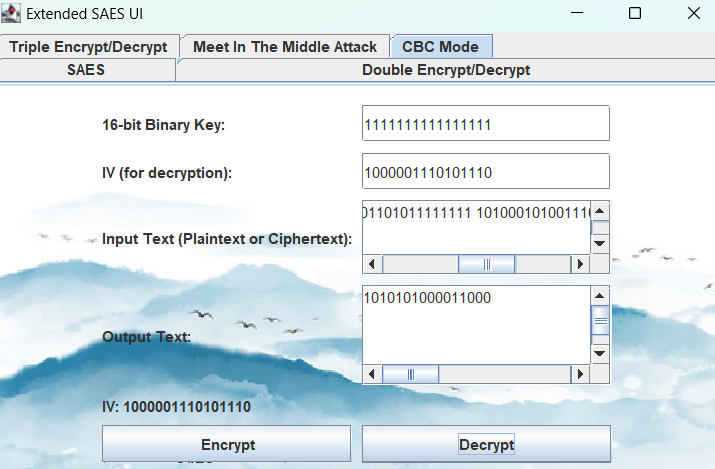
* + 1. 加密



3.5.2 解密



3.5.3 篡改密文



我们由此可以知道，在CBC模式下进行加解密，若其密文分组遭到篡改或替换，其最终得到的明文也有所改变。并且其初始化向量IV的强随机性增加了CBC工作模式的安全性。

3.5.4 总结

在本关卡中，我们小组成功完成了任务要求，主要做到了以下几点：

①密钥唯一性：我们测试了在给定明文-密文对的情况下，通常只存在一个密钥Key能够得到相应的密文C；了解了加密算法的目标是确保密文C与特定的密钥Key相关联，以保护数据的安全性。

② CBC模式设计原则：在CBC模式中，由于每个密文块都依赖于前一个密文块，因此明文中的任何模式都不会在密文中显现出来，这有助于防止攻击者通过分析密文来推断明文的内容。同时在CBC模式中，如果密文中的一个块损坏，那么只有该块及其直接后续的块会受到影响，而不会影响到后续所有的块。

③CBC模式的安全性评估：CBC模式通过使用初始化向量（IV）和前一个密文块与当前明文块进行异或操作，确保了即使相同的明文块在加密后也不会产生相同的密文块，这显著提高了加密的安全性。这种特性使得CBC模式不易受到重放攻击和某些类型的密码分析攻击。

1. **测试结果**

我们完成了基本的测试流程，以验证S-AES算法的实现是否正确，并确保其能够达到预定的目标，测试包括基本功能测试、性能测试、安全性测试等，以确保算法的正确性、效率和安全性。通过以上测试，我们证明了S-AES算法的实现是正确的，并且达到了预定的目标。算法在功能、性能和安全性方面都表现良好，可以用于实际的加密解密任务中。我们的测试结果表明，S-AES算法是一个可靠和高效的加密工具，可以用于保护数据的安全性。