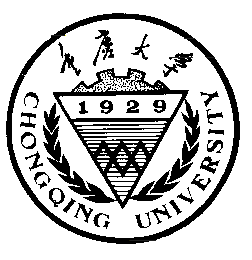
****



S-AES算法实现开发手册

|  |  |  |
| --- | --- | --- |
| 组员姓名 | 学号 | 班级 |
| 吴安婷 | 20221706 | 大数据1班 |
| 花蕊 | 20221702 | 大数据2班 |
| 殷宇欣 | 20221892 | 大数据1班 |

课程名称 信息安全导论

设计项目 S-AES算法实现

课程教师 向宏

开发起止日期 2024.10.18-2024.10.29

**大数据与软件学院**

目 录

[1、引言](#_Toc17696) 2

[1.1 背景 2](#_Toc2525)

[1.1.1项目背景 2](#_Toc11372)

[1.1.2成员介绍 2](#_Toc11372)

[1.1.3分工描述 2](#_Toc11372)

[1.2 术语及说明 2](#_Toc6584)

[2、设计概述 3](#_Toc17696)

[2.1 功能需求 3](#_Toc21108)

[2.2 设计方法和开发工具 4](#_Toc8644)

[2.2.1设计方法](#_Toc32640) 4

[2.1.2开发工具](#_Toc11828) 4

[3、系统详细需求分析 4](#_Toc9341)

[3.1系统运行环境和接口需求分析 4](#_Toc11970)

[3.2安全需求分析 5](#_Toc30942)

[4、总体方案确定](#_Toc3247) 5

[4.1 系统总体结构](#_Toc1202) 5

[4.2 系统详细界面](#_Toc7198) 5

[4.2.1基本测试界面 5](#_Toc32640)

[4.2.2交叉测试界面](#_Toc11828) 6

[4.2.3扩展功能界面](#_Toc32640) 8

[4.2.4多重加密界面](#_Toc11828) 9

[4.2.5 CBC模式界面](#_Toc32640) 11

**1 引言**

**1.1 背景**

**1.1.1项目背景**

在数字化时代，信息安全已成为全球关注的焦点。数据加密技术作为保护信息不被未授权访问的关键手段，其重要性不言而喻。AES算法因其高效性和广泛的应用场景，一直是信息安全领域的研究热点，其基本原理和结构是学习密码学不可或缺的一部分。

S-AES算法是高级加密标准（AES）的一个简化版本，它保留了AES的基本结构，但在关键参数上进行了缩减，使其更适合用于教学和理解AES算法的原理。S-AES可以作为开发和测试新加密技术的基础，因为它的结构与AES相同，但更简单，便于进行实验和分析。

S-AES通过简化的方式让我们更容易理解AES的工作原理，包括其加密和解密过程，以及如何处理数据块。通过对它的研究，可以更好地理解AES算法的安全性，以及如何抵御各种攻击。

**1.1.2成员介绍**

|  |  |  |
| --- | --- | --- |
| 成员 | 学号 | 性别 |
| 吴安婷 | 20221706 | 女 |
| 花蕊 | 20221702 | 女 |
| 殷宇欣 | 20221892 | 女 |

**1.1.3分工描述**

吴安婷：负责基本测试、扩展功能、三重加密、工作模式及GUI界面的美化和代码的最终完善处理。

花蕊：负责双重加密及中间相遇攻击的开发，撰写测试报告和开发手册的相关内容。

殷宇欣：负责中间相遇攻击、三重加密及GUI界面的开发，完善文档的最终内容。

**1.2 术语及说明**

**密钥长度**：密钥的位数，本项目中为16/32/48bits。

**密钥扩展算法**：是将原始密钥扩展成多个子密钥的过程，这些子密钥在加密过程中的不同轮次中使用。S-AES中，由于只需要加密两轮，因此只需要扩展出两个密钥。

**对称密钥算法**：也称为秘密密钥算法或单密钥算法，是一种加密方法，其中加密和解密过程使用相同的密钥。这意味着发送方和接收方都必须知道并使用相同的密钥来加密和解密数据。目前广泛使用的有AES算法等。

**S盒**：是对称密钥算法中用于执行置换计算的基本结构，它是算法中唯一的非线性结构，直接影响密码算法的安全性。

**字节替换**：是AES算法中的一个步骤，使用S盒对明文中的每个字节进行替换。在S-AES中，对16位二进制明文，8位一组，每组中包含两个16进制的二进制表达，也就是4位为一个单位进行字节替代。

**行变换**：是AES算法中的一个步骤，它将状态矩阵中的行进行循环位移。在S-AES中，第一行不变，第二行左移1位，第三行左移2位，第四行左移3位。

**列混淆**：是AES算法中的一个步骤，它对状态矩阵的每一列进行特定的数学运算，增加算法的扩散性。在S-AES中，这一步骤也有所简化，但基本原理相同。

**轮密钥加**：是AES算法中的一个步骤，将当前状态矩阵与一个轮密钥进行异或操作。在S-AES中，每一轮加密结束后都会进行这一步骤。

**ASCII编码**：一种字符编码标准，用于将字符转换为数字代码。

**GUI**：图形用户界面（Graphical User Interface），指通过图形方式显示的用户界面，用户可以通过图形元素进行交互操作。

**双重加密：**在双重加密中，S-AES算法使用32位密钥（K1+K2）。这意味着明文首先使用K1进行加密，然后再使用K2进行加密，从而生成密文。在解密时，需要以相反的顺序解密，首先使用K2解密，然后再使用K1解密，从而还原原始的明文。

1. 加密过程：明文 -> K1加密 -> 中间密文 -> K2加密 -> 最终密文
2. 解密过程：最终密文 -> K2解密 -> 中间密文 -> K1解密 -> 明文

**三重加密：**在三重加密中，S-AES算法可以采用两种模式之一：

1. 32位密钥Key（K1+K2）模式：这种模式使用32位密钥，首先使用K1加密，然后使用K2解密，最后再使用K1进行加密，从而生成密文。在解密时，需要以相反的顺序操作，即使用K1解密，然后使用K2加密，最后再使用K1解密以还原原始的明文。

加密过程：明文 -> K1加密 -> 中间密文1 -> K2解密 -> 中间密文2 -> K1加密 -> 最终密文

解密过程：最终密文 -> K1解密 -> 中间密文2 -> K2加密 -> 中间密文1 -> K1解密 -> 明文

1. 48位密钥（K1+K2+K3）模式：这种模式使用48位密钥，其中K1、K2和K3分别用于三个不同的加密/解密轮次。原理类似于32位密钥模式，但增加了一个额外的密钥。

加密过程：明文 -> K1加密 -> 中间密文1 -> K2解密 -> 中间密文2 -> K3加密 -> 最终密文

解密过程：最终密文 -> K3解密 -> 中间密文2 -> K2加密 -> 中间密文1 -> K1解密 -> 明文

**中间相遇攻击**：是一种密码分析方法，它利用加密过程中两个密文块在某个中间状态相遇的可能性。通过分析这些中间状态的数据，攻击者可以获取到关于明文的一些有用信息，从而进一步破解整个加密。

**CBC模式**：是分组密码的一种工作模式，其中每个明文块先与前一个密文块进行异或后，再进行加密，每个密文块都依赖于它前面的所有密文块。同时，为了保证每条消息的唯一性，在第一个块中需要使用初始化向量（IV）。这种模式可以确保相同的明文块生成不同的密文块，提高了安全性。

**2 设计概述**

**2.1 功能需求**

（1）算法实现：

密钥扩展：根据给定的16位密钥生成扩展密钥，S-AES的密钥扩展与AES类似，但更为简化。S-AES将4个nibble（4位）的密钥分组成两个8位的“词”，然后将这些词扩展成6个词的轮密钥。密钥扩展过程中，使用了轮常数和特定的代数运算。

初始和最终置换：使用初始置换盒（IP）和其逆置换盒（IP^{-1}）对数据进行置换。S-AES将16位的数据块分成2x2的nibble数组，即状态数组。初始置换和最终置换用于在加密和解密过程中重新排列数据块的nibbles。

轮函数F：实现轮函数F，包括S-Box和P-Box的逻辑。在S-AES中，每个nibble通过S-Box进行替换，然后与轮密钥进行XOR操作。S-Box用于将每个nibble映射到新的nibble，而P-Box则用于行变换后的列混淆过程。

行变换：在S-AES中，行变换仅对第二行进行，第一行保持不变。

列混淆：在行变换之后，执行列混淆。S-AES中的每一列都乘以一个特定的矩阵，这些操作在GF(16)中进行，需要对结果进行模的约简。

加密和解密流程：根据S-AES算法描述，实现加密和解密的流程。

（2）GUI设计：

输入界面：设计输入界面，允许用户输入明文和密钥。

输出界面：设计输出界面，显示加密后的密文或解密后的明文。

操作按钮：提供加密、解密、使用ASCII编码字符串的按钮，同时包括了基本测试、双重加密、中间相遇攻击、三重加密和CBC模式五种不同界面。

1. 编程实践：

变量命名：使用有意义的变量名，遵循驼峰命名法或下划线分隔命名法。

代码注释：为复杂的代码块添加清晰、有意义的注释。

函数式编程：将代码模块化，每个函数负责特定的功能实现。

1. 测试策略：

基本测试：测试程序是否能够正确加密和解密16-bit数据。

交叉测试：确保不同小组编写的程序能够相互加密和解密。

扩展功能测试：测试ASCII编码字符串的加密和解密功能。

多重加密测试：测试算法的双重加密和三重加密功能，其中三重加密使用了48-bit的模式，同时测试系统的中间相遇攻击功能。

CBC模式测试：测试CBC模式下对加长明文信息进行的加解密。

**2.2 设计方法和开发工具**

**2.2.1 设计方法**

（1）加密和解密功能：

实现S-AES算法的加密和解密过程，支持16-bit数据输入和16-bit密钥输入。

（2）用户交互：

通过GUI接收用户输入的明文和密钥，并显示加密后的密文或解密后的明文。

（3）ASCII编码支持：

扩展功能以支持ASCII编码字符串的加密和解密。

（4）多重加密：

实现双重加密、中间相遇攻击、三重加密的功能，并进行相关测试。

（5）CBC模式测试：

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。

**2.2.2 开发工具**

编程语言：JAVA

开发工具：Eclipse,IDEA

**3 系统详细需求分析**

**3.1 系统运行环境和接口需求分析**

（1）系统需要在以下环境中运行：

操作系统：Windows

JAVA：使用JAVA语言结合Swing库等来实现GUI

1. 系统接口需求：

GUI接口：使用库构建GUI界面，包括输入密钥等用户交互组件，以及按钮触发事件。

**3.2 安全需求分析**

（1）加密强度：

确保S-AES算法实现的加密强度足够，防止简单破解。

（2）多重加密安全：

在实现多重加密功能时，确保密钥的安全存储、传输和销毁

**4 总体方案确定**

**4.1 系统总体结构**

用户界面：负责与用户的交互，包括输入明文、密钥、密文和显示结果。

加密、解密：实现S-AES算法的加、解密过程。

ASCII模式：处理ASCII编码字符串的输入和输出。

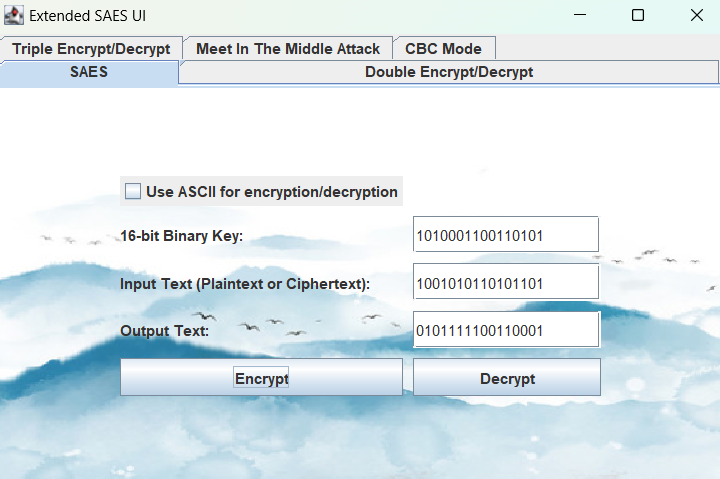
多重加密：使用双重加密和三重加密进行扩展，同时利用中间相遇攻击功能找到正确密钥。

CBC模式：引入初始向量，使得相同的明文块生成不同的密文块，增强数据的安全性。

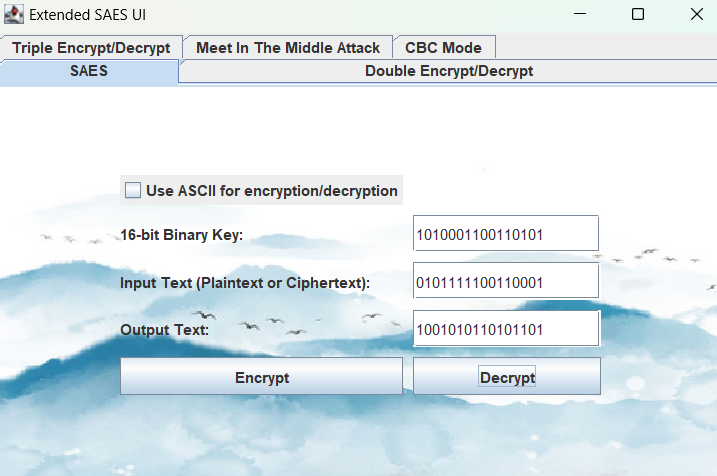
**4.2 系统详细界面**

**4.2.1 基本测试界面**

加密：输入16bit的密钥和16bit的明文，输出的是16bit的密文。



解密：输入16bit的密钥和16bit的密文，输出的是16bit的明文。

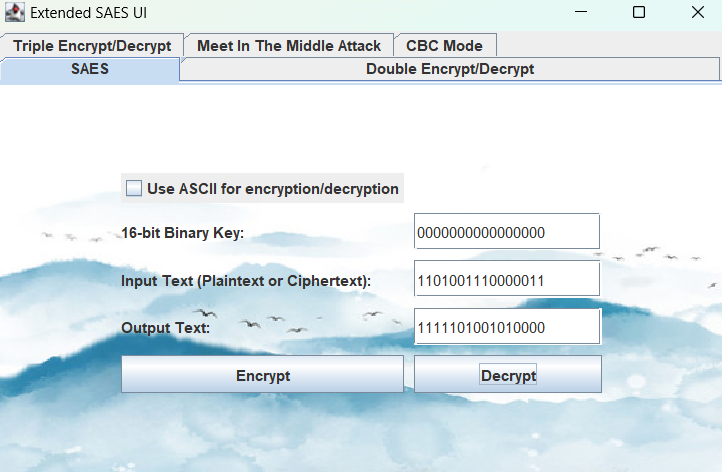


**4.2.2  交叉测试界面**

①我方解密结果：密钥：0000000000000000

密文：1101001110000011

求得明文：1111101001010000



对方加密结果：密钥：0000000000000000

明文：1111101001010000

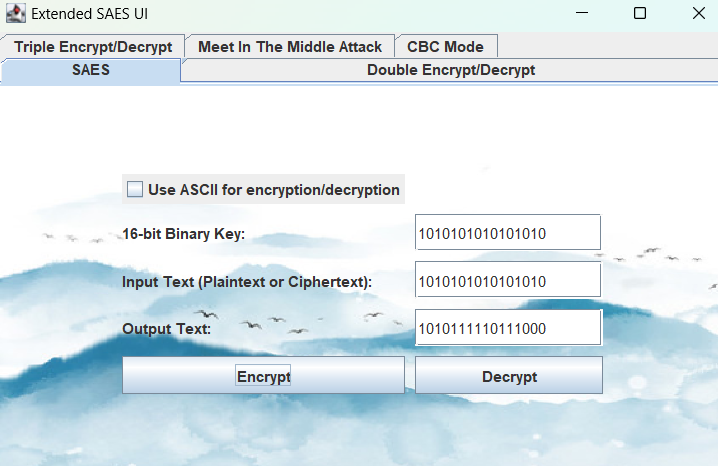
求得密文：1101001110000011



②我方加密结果：密钥：1010101010101010

明文：1010101010101010

求得密文：1010111110111000



对方解密结果：密钥：1010101010101010

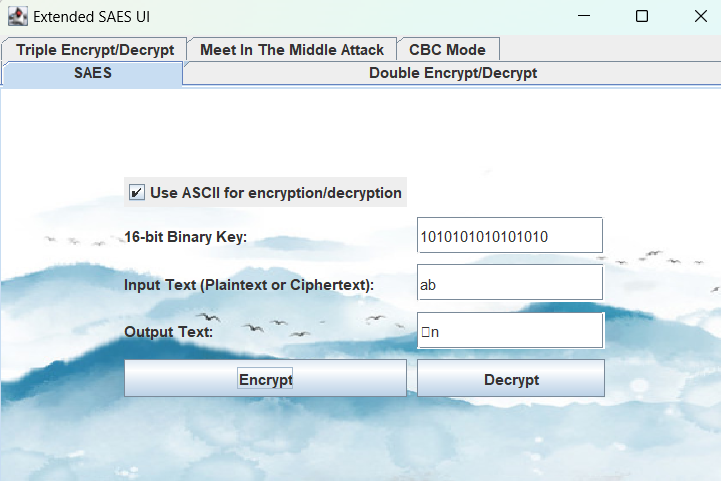
密文：1010111110111000

求得明文：1010101010101010

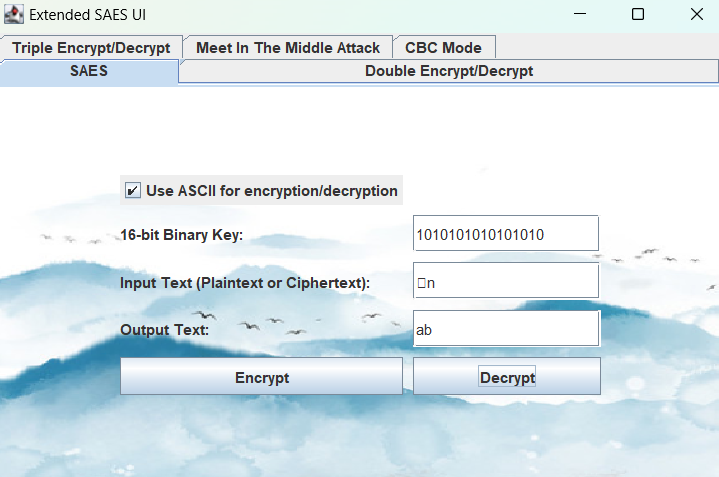


**4.2.3  扩展功能界面**

加密：输入的明文为字符串，对应的加密输出的可以是ACSII字符串或乱码。



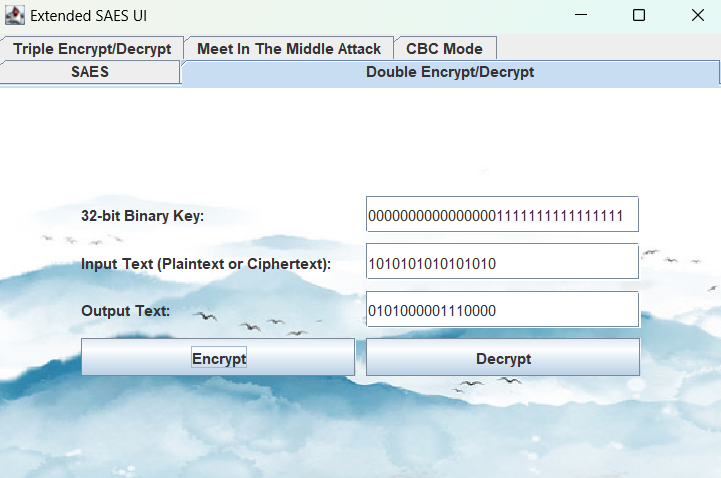
解密：输入的密文为字符串，对应的解密输出的可以是ACSII字符串或乱码。



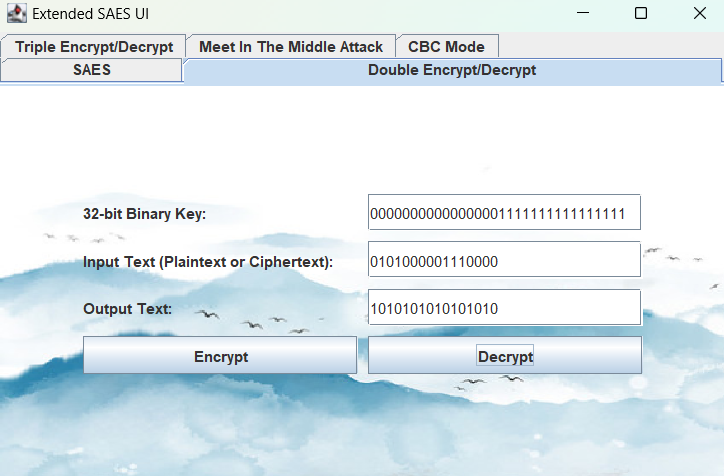
**4.2.4  多重加密测试界面**

1.二重加密：

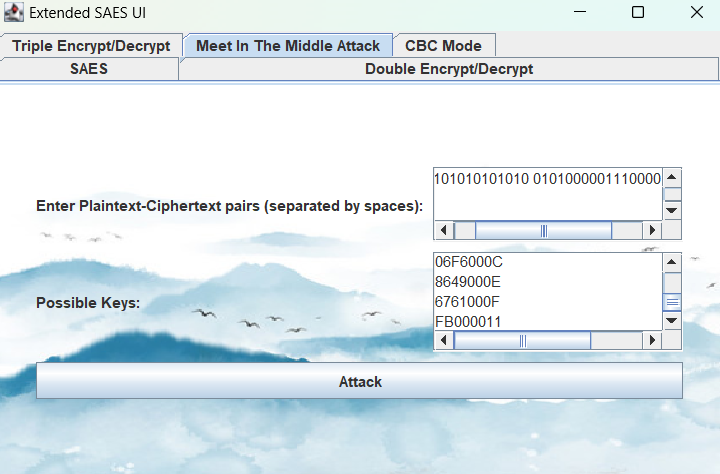
加密：



解密：

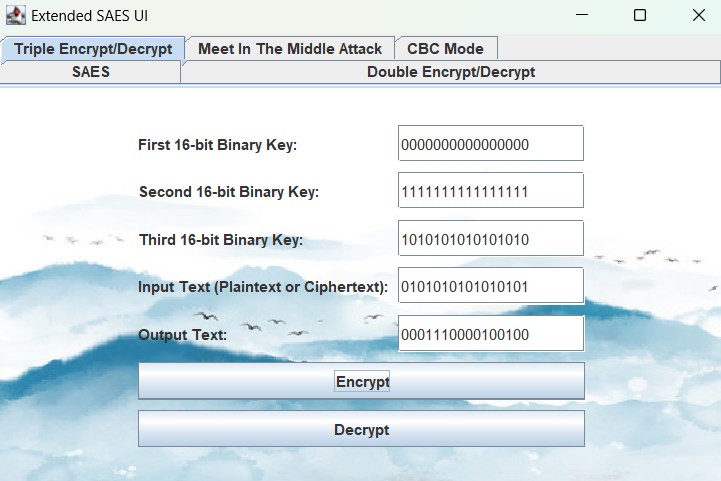


1. 中间相遇攻击：

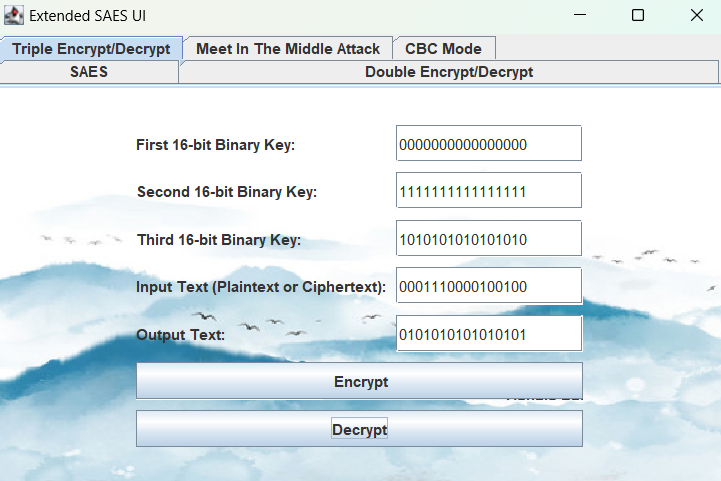


1. 三重加密：

加密：

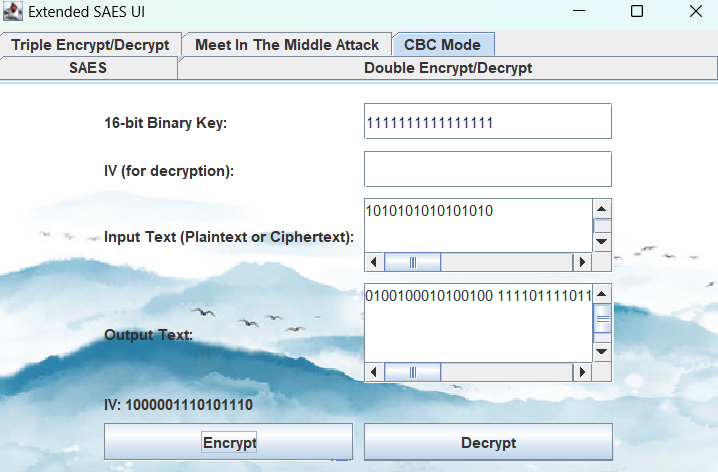


解密：

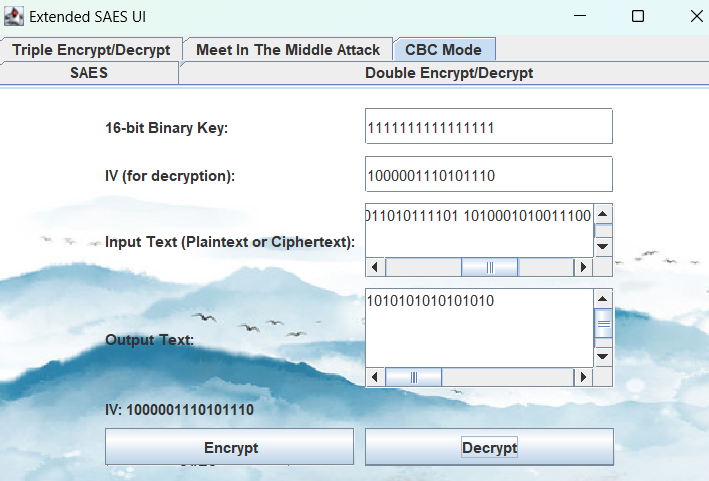


**4.2.5  CBC模式界面**

加密：



解密：



修改密文后，解密出的明文也有所改变：

