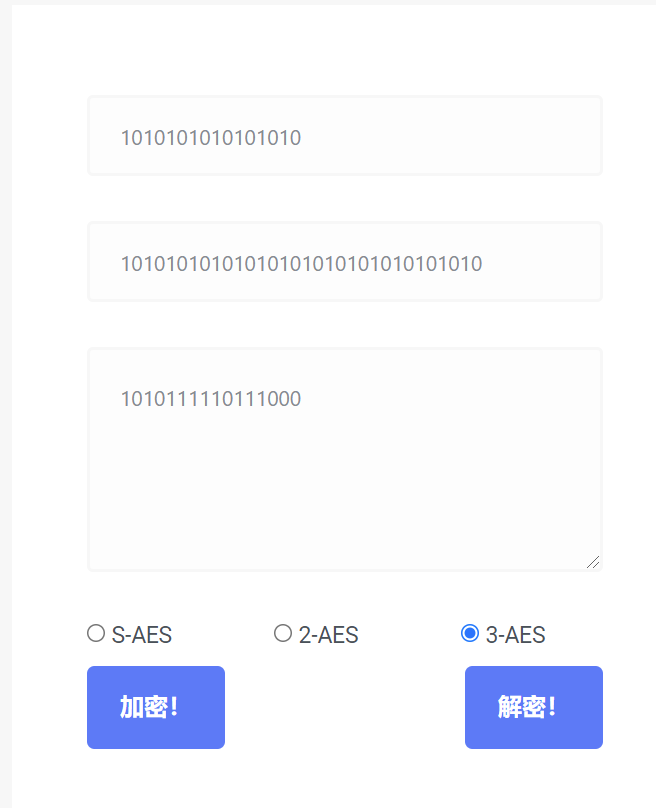
**第1关：基本测试**

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。

加密算法测试结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **明文** | **密钥** | **S-AES** | **2-AES** | **3-AES** |
| 1111111111111111 | 1111111111111111 | 0101001101000011 | 0111101010110101 | 0101001101000011 |
| 0000000000000000 | 1111111111111111 | 0000100011000001 | 1001100100101110 | 0000100011000001 |
| 1111111100000000 | 0000000000000000 | 1101000101000011 | 0001000101100011 | 1101000101000011 |
| 0000000011111111 | 0000000000000000 | 0101101011001000 | 0011000111110111 | 0101101011001000 |
| 1010101010101010 | 1010101010101010 | 1010111110111000 | 1001011001100110 | 1010111110111000 |

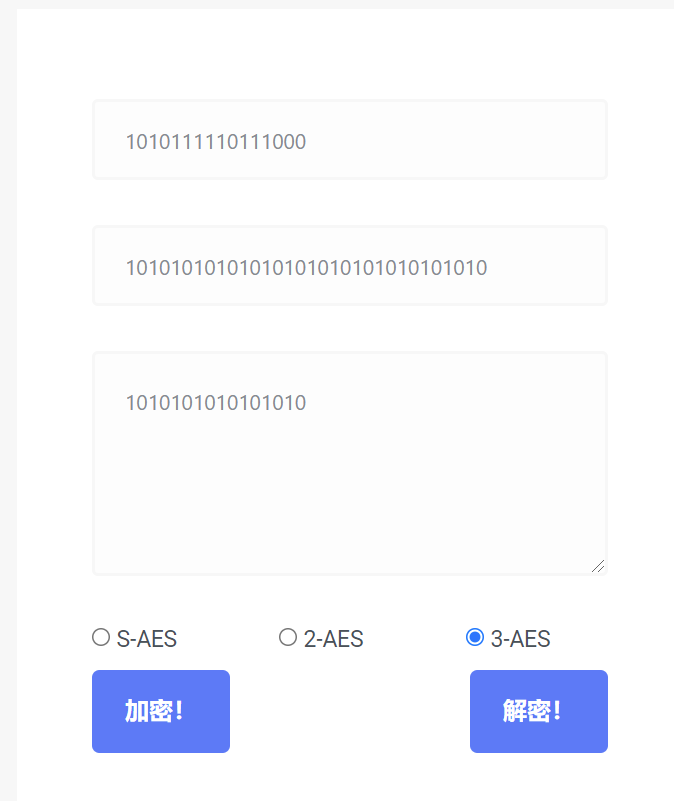
图形用户界面, 应用程序

描述已自动生成图形用户界面, 应用程序

描述已自动生成

解密算法的测试结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **密钥** | **S-AES** | **2-AES** | **3-AES** | **明文** |
| 1111111111111111 | 0101001101000011 | 0111101010110101 | 0101001101000011 | 1111111111111111 |
| 1111111111111111 | 0000100011000001 | 1001100100101110 | 0000100011000001 | 0000000000000000 |
| 0000000000000000 | 1101000101000011 | 0001000101100011 | 1101000101000011 | 1111111100000000 |
| 0000000000000000 | 0101101011001000 | 0011000111110111 | 0101101011001000 | 0000000011111111 |
| 1010101010101010 | 1010111110111000 | 1001011001100110 | 1010111110111000 | 1010101010101010 |

图形用户界面, 应用程序

描述已自动生成图形用户界面, 应用程序

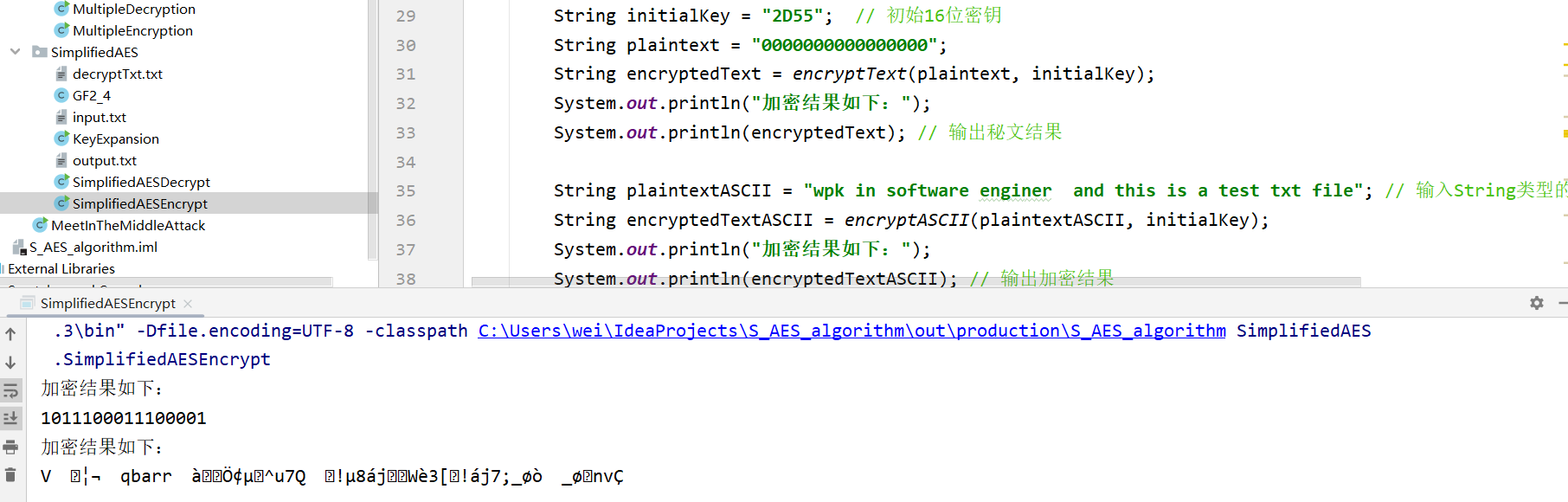
描述已自动生成

**第2关：交叉测试**

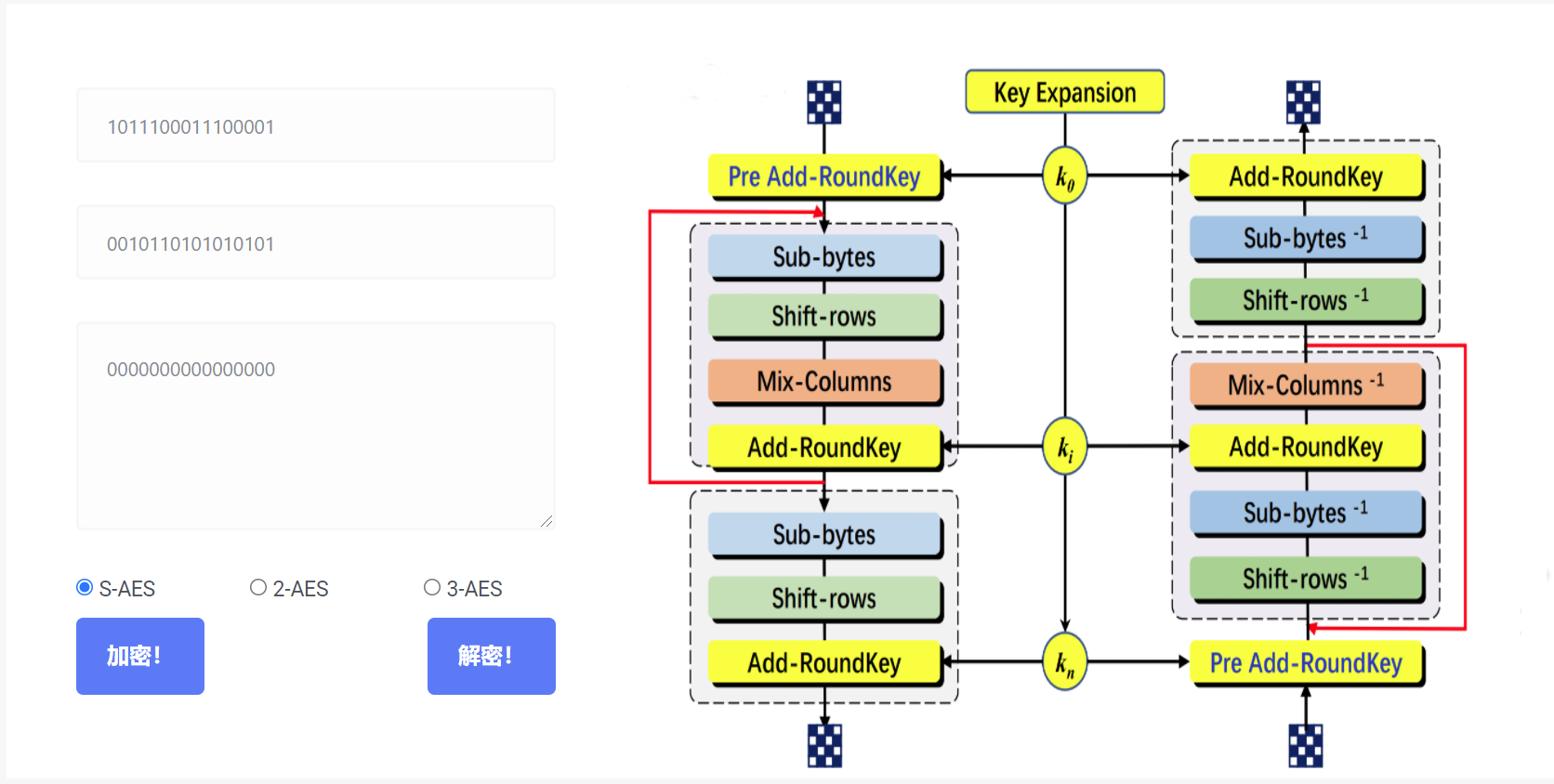
考虑到是"**算法标准"**，所有人在编写程序的时候需要使用相同算法流程和转换单元(替换盒、列混淆矩阵等)，以保证算法和程序在异构的系统或平台上都可以正常运行。

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。

A组同学结果如下



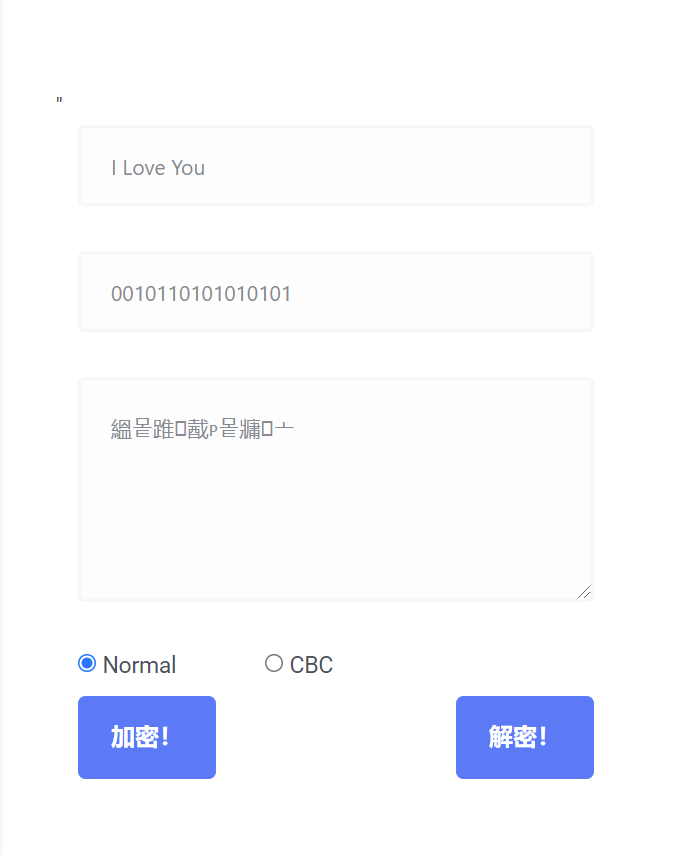
B组同学解密的结果如下



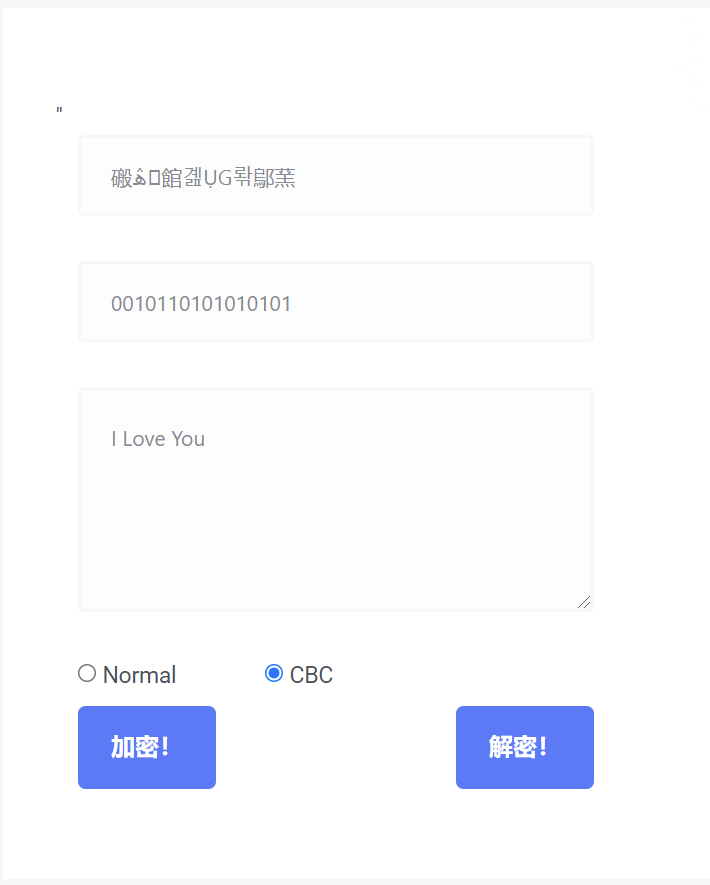
**第3关：扩展功能**

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1 Byte)，对应地输出也可以是ACII字符串(很可能是乱码)。

加密和解密字符串"I Love You"，结果如下：

图形用户界面, 应用程序

描述已自动生成图形用户界面, 应用程序, Teams

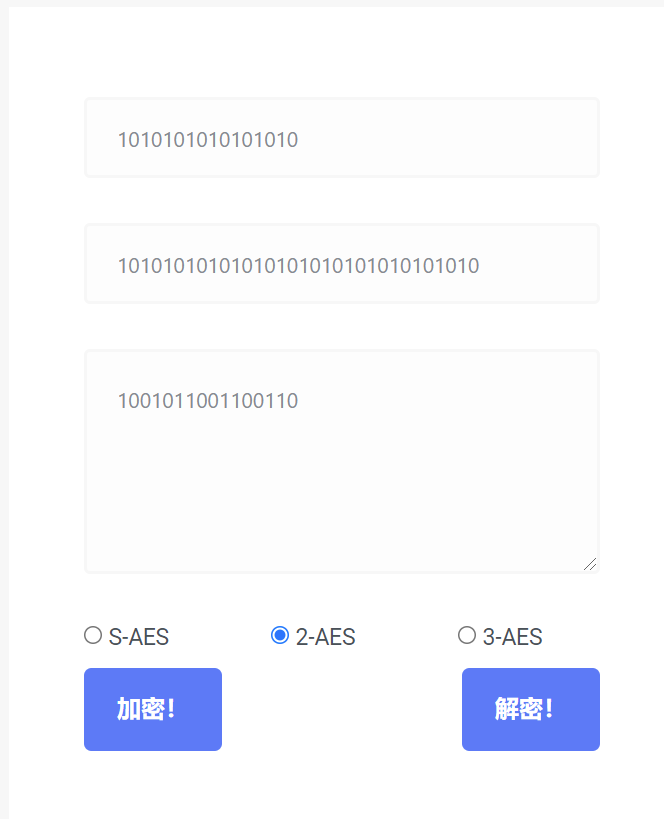
描述已自动生成

**第4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。

使用32bit的密钥10101010101010101010101010101010两次加密1010101010101010，得到的密文是1001011001100110再进行解密操作，可以得到原始明文。

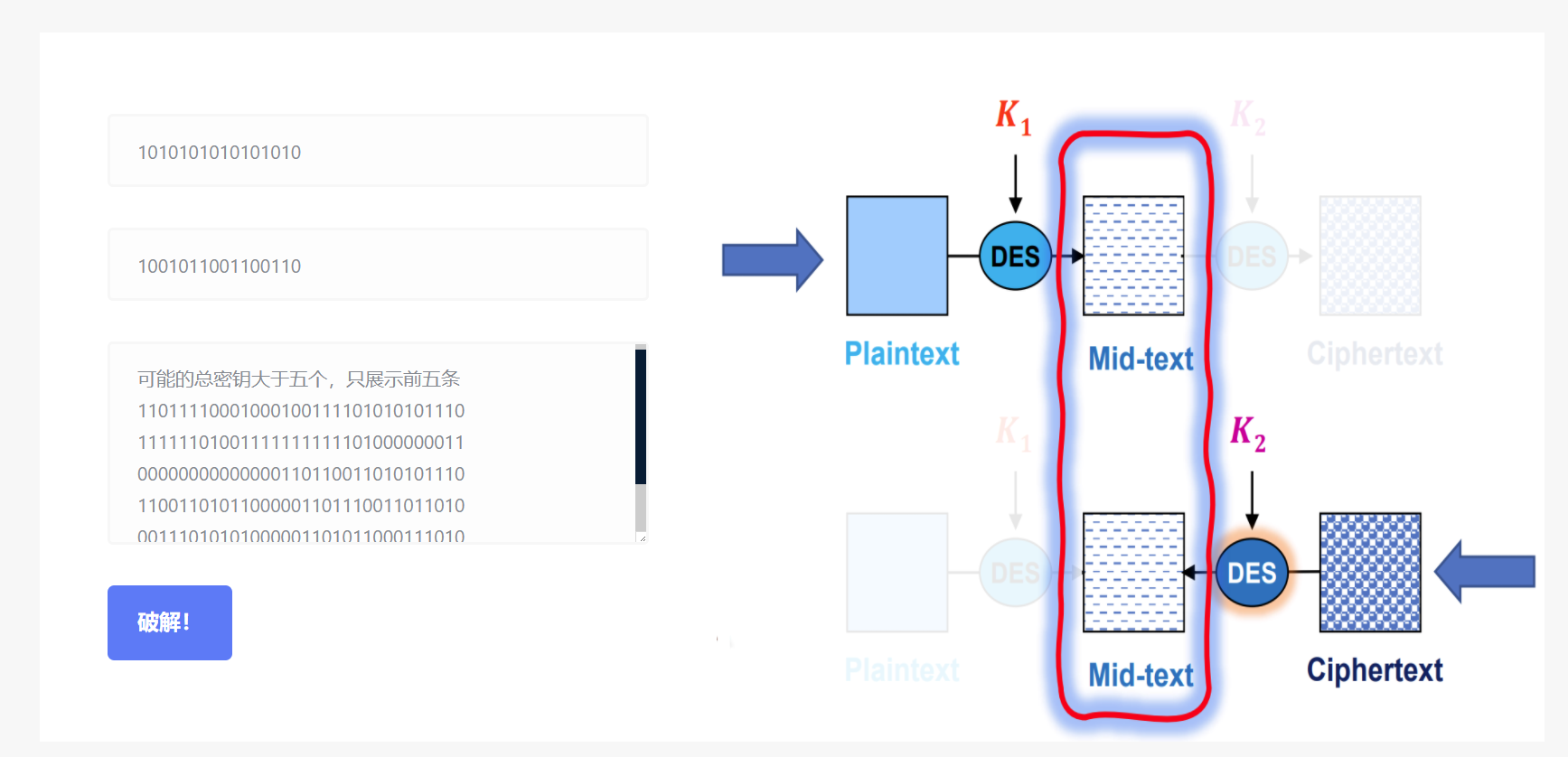
图形用户界面, 应用程序

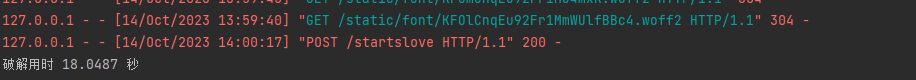
描述已自动生成

**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)。

转到相遇攻击页面破解刚刚的明文和密文，可以看到，可能的密钥总数大于5个，因此我们只显示了五个密钥。若需要找到精确的密钥，则需要更多的明密文对。



破解一次的总用时为18.0487s

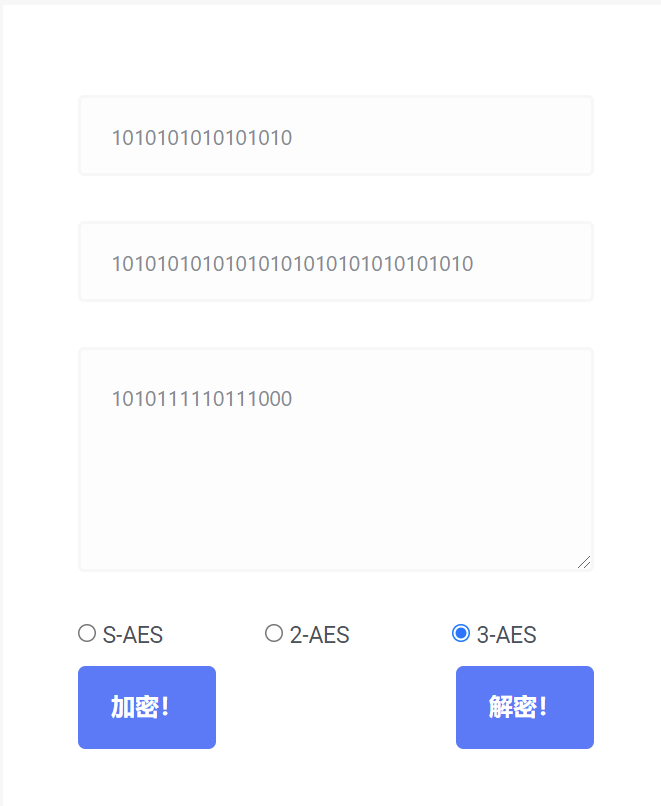
**3.三重加密**

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。

我们采用第(1)种方式进行了三重加密。使用32bit的密钥10101010101010101010101010101010三次加密1010101010101010，得到的密文是1010111110111000再进行解密操作，可以得到原始明文。

图形用户界面, 应用程序

描述已自动生成

**3.5 第5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。

将字符串视为长明文消息，用前一个字符迭代加密后一个字符，构造字符串CBC加密算法。加密I Love You得到结果，再进行反向解密可以得到明文。

篡改一个分组，可以发现，篡改之后的分组仍可以解密，但篡改之前的分组全部变为了乱码

图形用户界面, 应用程序, Teams

描述已自动生成