\*\*1. 简明的介绍和概述：\*\*

欢迎使用S-AES加密软件用户指南。本软件是一个用于对文本进行加密和解密的工具，基于简化数据加密标准（S-AES）算法。它可以帮助您保护敏感信息，确保您的数据在传输和存储时得到安全保护。

\*\*2. 系统要求和安装：\*\*

\*\*系统要求：\*\*

- 操作系统：支持Windows、macOS和Linux。

- Python：需要安装Python解释器。

- Tkinter：需要安装Tkinter库。

- pyperclip：需要安装pyperclip库。

\*\*安装步骤：\*\*

1. 下载软件源代码到您的计算机。

2. 安装Python解释器（如果尚未安装）。

3. 安装Tkinter库（如果尚未安装）。

4. 安装pyperclip库（如果尚未安装）。

5. 运行源代码。

\*\*3. 用户界面和基本操作：\*\*

\*\*用户界面\*\*

用户界面简单易用，包括功能选择界面和操作界面。

功能选择包括二进制数加解密GUI按钮、ASCLL码加解密GUI按钮以及CBC防篡改测试以及中间相遇攻击破解双重加密以及三重加密等多种功能。

操作界面包括随机生成密钥按钮、文本输入框、密钥输入框、加密和解密按钮，以及结果显示区域。

\*\*基本操作\*\*

- 在功能选择界面选择二进制数加解密GUI、ASCLL码加解密GUI以进入操作界面。

- 点击随机生成密钥按钮生成随机密钥。（可选）

- 在文本输入框中输入要加密或解密的文本。

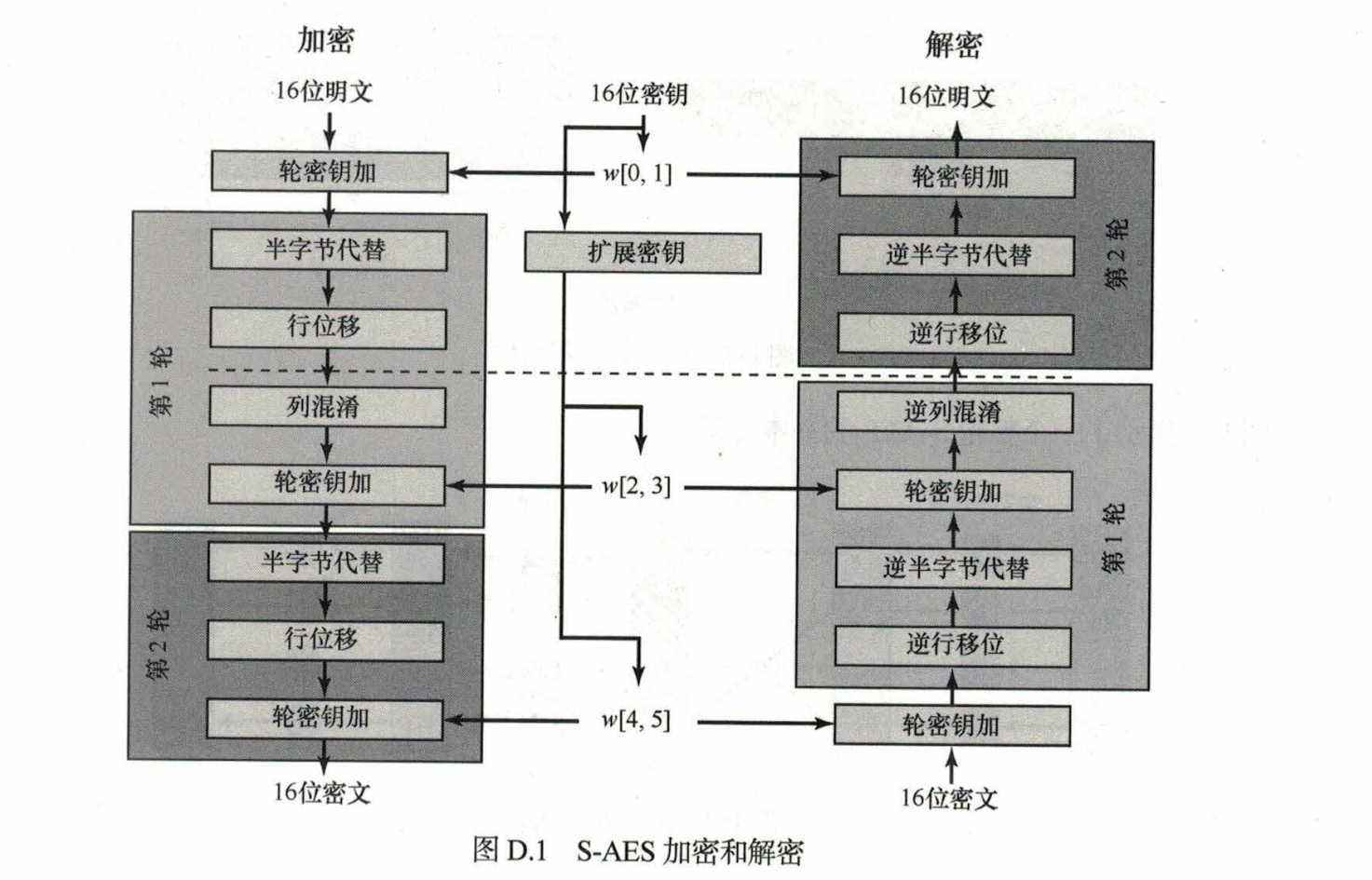
- 在密钥输入框中输入16位的二进制密钥。

- 点击加密按钮执行加密操作。

- 点击解密按钮执行解密操作。

- 结果将显示在结果区域。

\*\*4. 加密和解密流程：\*\*



本项目的核心加密部分代码实现如下：

# S-Box、逆S-Box和替换矩阵  
s\_box = [  
 [9, 4, 10, 11],  
 [13, 1, 8, 5],  
 [6, 2, 0, 3],  
 [12, 14, 15, 7]  
]  
  
inv\_s\_box = [  
 [10, 5, 9, 11],  
 [1, 7, 8, 15],  
 [6, 0, 2, 3],  
 [12, 4, 13, 14]  
]  
  
replacement\_matrix = [  
 [0, 0, 0, 0],  
 [0, 0, 0, 1],  
 [0, 0, 1, 0],  
 [0, 0, 1, 1],  
 [0, 1, 0, 0],  
 [0, 1, 0, 1],  
 [0, 1, 1, 0],  
 [0, 1, 1, 1],  
 [1, 0, 0, 0],  
 [1, 0, 0, 1],  
 [1, 0, 1, 0],  
 [1, 0, 1, 1],  
 [1, 1, 0, 0],  
 [1, 1, 0, 1],  
 [1, 1, 1, 0],  
 [1, 1, 1, 1]  
]  
  
# Rcon常数  
rcon1 = [1, 0, 0, 0, 0, 0, 0, 0]  
rcon2 = [0, 0, 1, 1, 0, 0, 0, 0]  
  
def multiply\_x(xfx, a):  
 # 注意要取模，既约多项式是x^4 + x + 1  
 if a[0] == 0:  
 xfx[0] = a[1]  
 xfx[1] = a[2]  
 xfx[2] = a[3]  
 xfx[3] = 0  
 else:  
 xfx[0] = a[1]  
 xfx[1] = a[2]  
 xfx[2] = a[3] ^ 1  
 xfx[3] = 1  
  
def polynomial\_multiply(a, b):  
 result = [0, 0, 0, 0]  
 xfx = [0, 0, 0, 0]  
 multiply\_x(xfx, a)  
 x2fx = [0, 0, 0, 0]  
 multiply\_x(x2fx, xfx)  
 x3fx = [0, 0, 0, 0]  
 multiply\_x(x3fx, x2fx)  
  
 if b[0] == 1:  
 for i in range(4):  
 result[i] ^= x3fx[i]  
 if b[1] == 1:  
 for i in range(4):  
 result[i] ^= x2fx[i]  
 if b[2] == 1:  
 for i in range(4):  
 result[i] ^= xfx[i]  
 if b[3] == 1:  
 for i in range(4):  
 result[i] ^= a[i]  
 return result  
  
def XOR8(a, b):  
 t = [0] \* 8  
 for i in range(8):  
 t[i] = a[i] ^ b[i]  
 return t  
  
def XOR4(a, b):  
 t = [0] \* 4  
 for i in range(4):  
 t[i] = a[i] ^ b[i]  
 return t  
  
def sbox\_substitution(temp):  
 t1 = 2 \* temp[0] + temp[1]  
 t2 = 2 \* temp[2] + temp[3]  
 t3 = 2 \* temp[4] + temp[5]  
 t4 = 2 \* temp[6] + temp[7]  
  
 tihuan1 = s\_box[t1][t2]  
 tihuan2 = s\_box[t3][t4]  
  
 for i in range(4):  
 temp[i] = replacement\_matrix[tihuan1][i]  
 for i in range(4):  
 temp[i + 4] = replacement\_matrix[tihuan2][i]  
  
def inverse\_sbox\_substitution(temp):  
 t1 = 2 \* temp[0] + temp[1]  
 t2 = 2 \* temp[2] + temp[3]  
 t3 = 2 \* temp[4] + temp[5]  
 t4 = 2 \* temp[6] + temp[7]  
 tihuan1 = inv\_s\_box[t1][t2]  
 tihuan2 = inv\_s\_box[t3][t4]  
  
 for i in range(4):  
 temp[i] = replacement\_matrix[tihuan1][i]  
 for i in range(4):  
 temp[i + 4] = replacement\_matrix[tihuan2][i]  
  
def left\_shift(temp):  
 #第一字节的右半部分和第二字节的右半部分进行替换  
 for i in range(4, 8):  
 t = temp[0][i]  
 temp[0][i] = temp[1][i]  
 temp[1][i] = t  
  
  
  
def g\_function(temp, rcon):  
 t = list(temp)  
 for i in range(4):  
 tt = t[i + 4]  
 t[i + 4] = t[i]  
 t[i] = tt  
  
 sbox\_substitution(t)  
 return XOR8(t, rcon)  
  
def mix\_columns(mingwen):  
 si\_de2jinzhi = [0, 1, 0, 0]  
 m00 = list(mingwen[0][:4])  
 m10 = list(mingwen[0][4:])  
 m01 = list(mingwen[1][:4])  
 m11 = list(mingwen[1][4:])  
 n00 = XOR4(m00, polynomial\_multiply(si\_de2jinzhi, m10))  
 n10 = XOR4(polynomial\_multiply(si\_de2jinzhi, m00), m10)  
 n01 = XOR4(m01, polynomial\_multiply(si\_de2jinzhi, m11))  
 n11 = XOR4(polynomial\_multiply(si\_de2jinzhi, m01), m11)  
 for i in range(4):  
 mingwen[0][i] = n00[i]  
 mingwen[0][i + 4] = n10[i]  
 mingwen[1][i] = n01[i]  
 mingwen[1][i + 4] = n11[i]  
  
def inverse\_mix\_columns(mingwen):  
 er\_de2jinzhi = [0, 0, 1, 0]  
 jiu\_de2jinzhi=[1,0,0,1]  
 m00 = list(mingwen[0][:4])  
 m10 = list(mingwen[0][4:])  
 m01 = list(mingwen[1][:4])  
 m11 = list(mingwen[1][4:])  
 n00 = XOR4(polynomial\_multiply(jiu\_de2jinzhi, m00), polynomial\_multiply(er\_de2jinzhi, m10))  
 n10 = XOR4(polynomial\_multiply(er\_de2jinzhi, m00), polynomial\_multiply(jiu\_de2jinzhi, m10))  
 n01 = XOR4(polynomial\_multiply(jiu\_de2jinzhi, m01), polynomial\_multiply(er\_de2jinzhi, m11))  
 n11 = XOR4(polynomial\_multiply(er\_de2jinzhi, m01), polynomial\_multiply(jiu\_de2jinzhi, m11))  
 for i in range(4):  
 mingwen[0][i] = n00[i]  
 mingwen[0][i + 4] = n10[i]  
 mingwen[1][i] = n01[i]  
 mingwen[1][i + 4] = n11[i]  
  
  
  
def round\_key\_addition(mingwen, key):  
 for i in range(2):  
 for j in range(8):  
 mingwen[i][j] ^= key[i][j]  
  
def output(a):  
 for i in range(2):  
 for j in range(8):  
 print(a[i][j], end=' ')  
 print()  
  
  
def aes\_encrypt(plaintext,key):  
  
 mingwen = [[int(plaintext[i]) for i in range(8)], [int(plaintext[i]) for i in range(8, 16)]]  
 key = [[int(key[i]) for i in range(8)], [int(key[i]) for i in range(8, 16)]]  
  
 w0=key[0]  
 w1=key[1]  
 w2=XOR8(w0, g\_function(w1, rcon1))  
 w3=XOR8(w2, w1)  
 w4=XOR8(w2, g\_function(w3, rcon2))  
 w5=XOR8(w4, w3)  
 key1=[w2,w3]  
 key2=[w4,w5]  
  
 round\_key\_addition(mingwen, key)  
  
 sbox\_substitution(mingwen[0])  
 sbox\_substitution(mingwen[1])  
 left\_shift(mingwen)  
 mix\_columns(mingwen)  
 round\_key\_addition(mingwen, key1)  
  
 sbox\_substitution(mingwen[0])  
 sbox\_substitution(mingwen[1])  
 left\_shift(mingwen)  
 round\_key\_addition(mingwen, key2)  
  
 return mingwen

\*\*5. 安全性和最佳实践：\*\*

数据安全至关重要。请遵循以下最佳实践：

- 不要分享密钥或密码。

- 使用强密码来保护您的密钥。

- 将密钥存储在安全的地方。

\*\*6. 高级功能和设置：\*\*

本软件提供高级功能和设置，支持生成随机密钥，支持对二进制数、ASCII码进行加密或解密。

\*\*7. 错误处理和故障排除：\*\*

常见错误消息和问题的解决方法将在本指南中列出，以帮助您在遇到问题时解决它们。

\*\*8. 法律和合规性：\*\*

请确保您在使用本软件时遵守适用的法律和合规性要求。我们不对用户的非法或不当使用负责。

\*\*9. 更新和支持：\*\*

我们会不定期地在Github上对该软件进行更新。如果您需要技术支持或有反馈，请联系我们的支持团队。

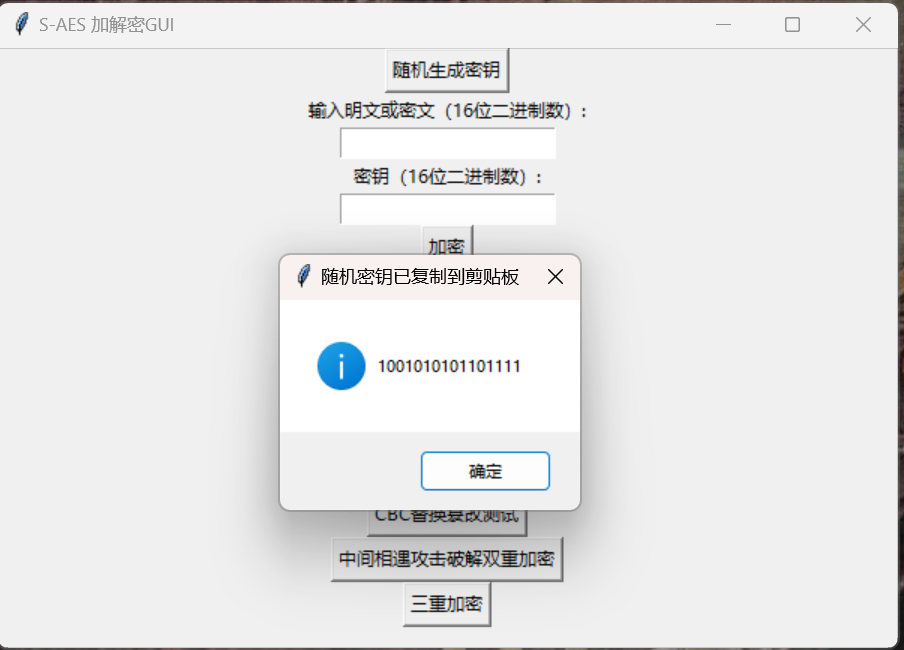
\*\*10. 示例和案例：\*\*

为了帮助您更好地理解软件的用途，我们提供了示例和案例，以展示不同情境下的使用方式。

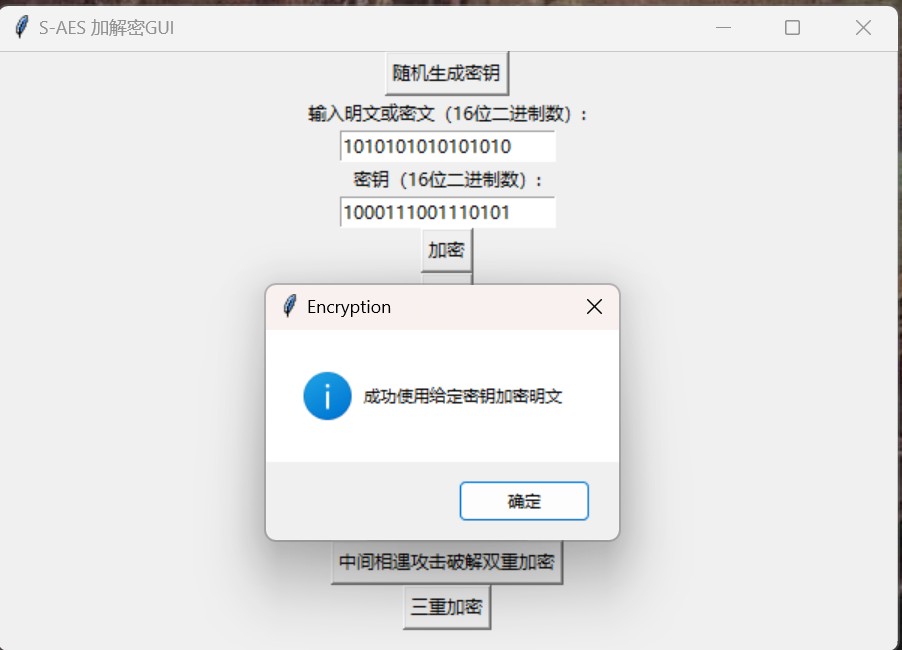
--运行程序，进入功能选择界面



--点击随机生成密钥（密钥已复制到剪切板，直接粘贴到密钥输入框中即可）



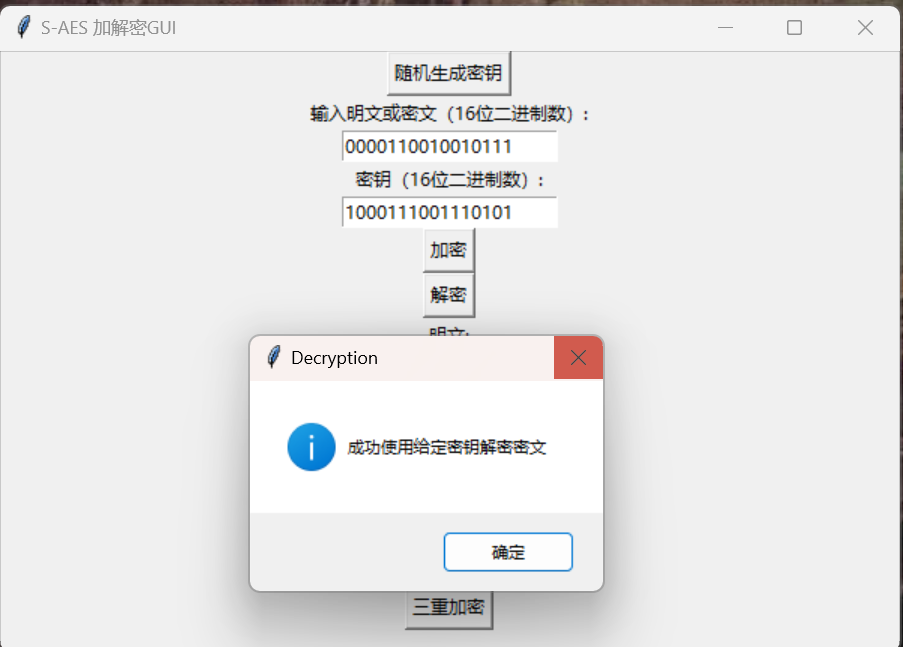
--将密钥粘贴到密钥输入框，在明文或密文输入框中输入将要加密的明文(16位二进制数)，点击加密按钮



--得到加密结果，显示在密文显示框中



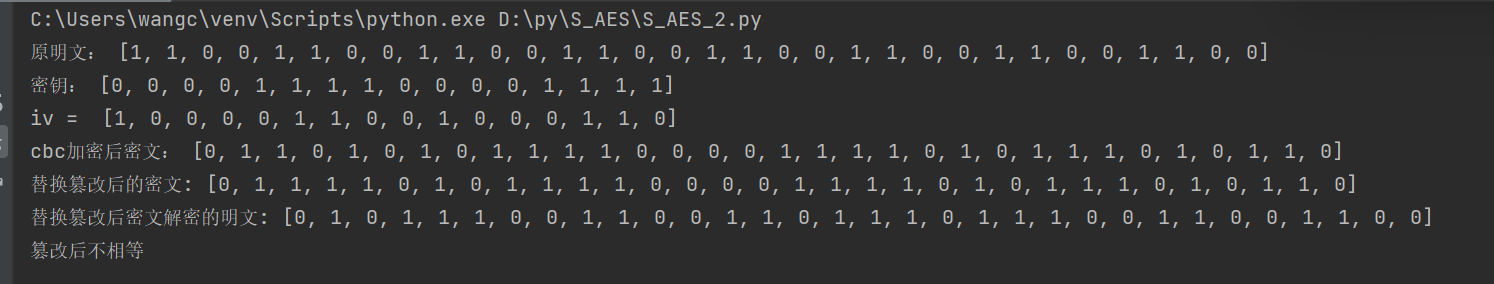
--对密文进行解密同样如此，在明文或密文输入框中输入将要解密的密文，点击解密按钮



--得到解密结果，显示在明文显示框中

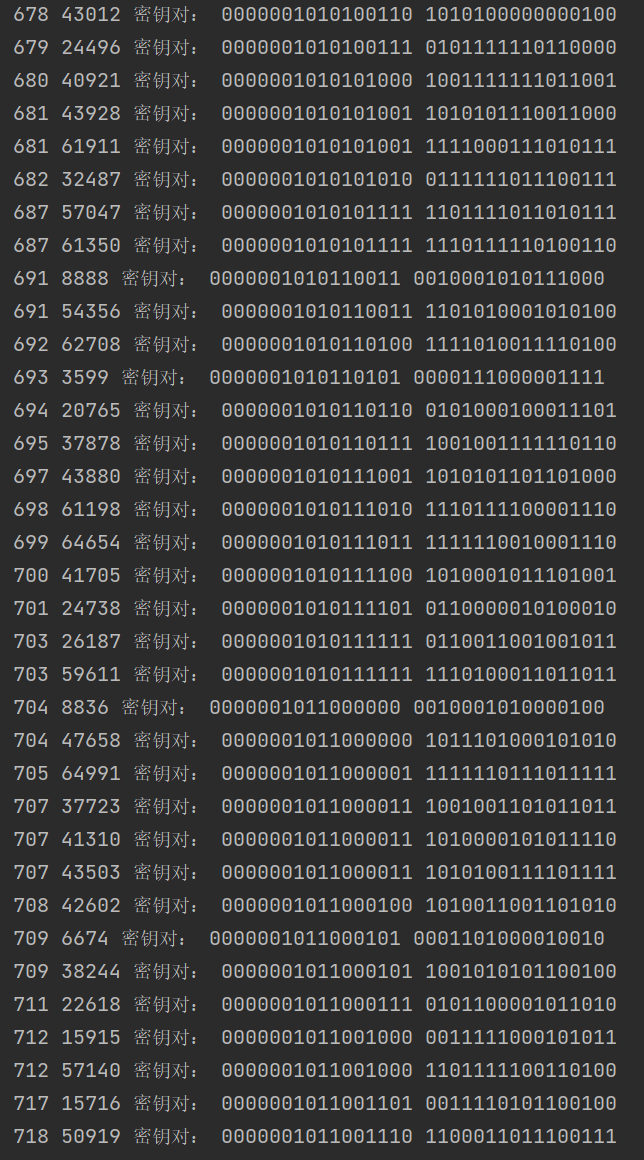


--点击CBC替换篡改测试按钮，可以找到原密文和密钥以及替换篡改前后的比较测试

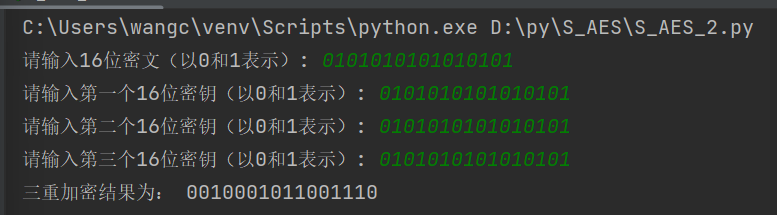


--同理，在功能选择界面点击中间相遇破解双重加密按钮，可以进行对应的测试

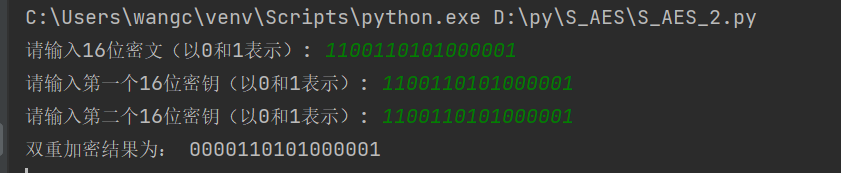
以下展示部分示例找到的密钥对：



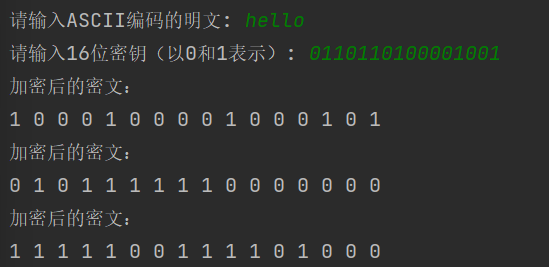
--在功能选择界面点击三重加密按钮，可以进行对应的测试



--在功能选择界面点击双重加密按钮，可以进行对应的测试



--在功能选择界面点击ASCII加密按钮，可以进行对应的测试



\*\*11. 安全注意事项：\*\*

强调用户不应分享密钥、密码或加密数据，以及在使用软件时应保持警惕。

\*\*12. 测试和校对：\*\*

我们经过充分的测试和校对，确保用户指南没有拼写或语法错误。我们的目标是提供清晰和准确的指南，以帮助用户轻松使用本软件。

希望这份用户指南能够帮助您了解并正确使用S-DES加密软件。如果您有任何问题或需要进一步的帮助，请随时联系我们的支持团队。