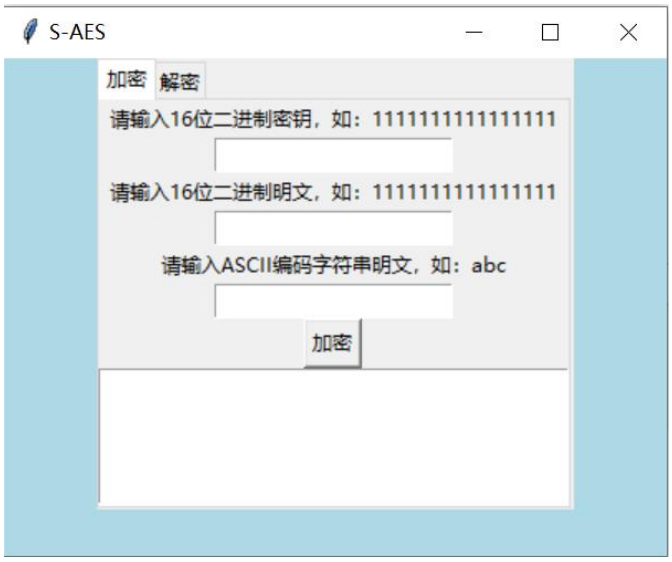
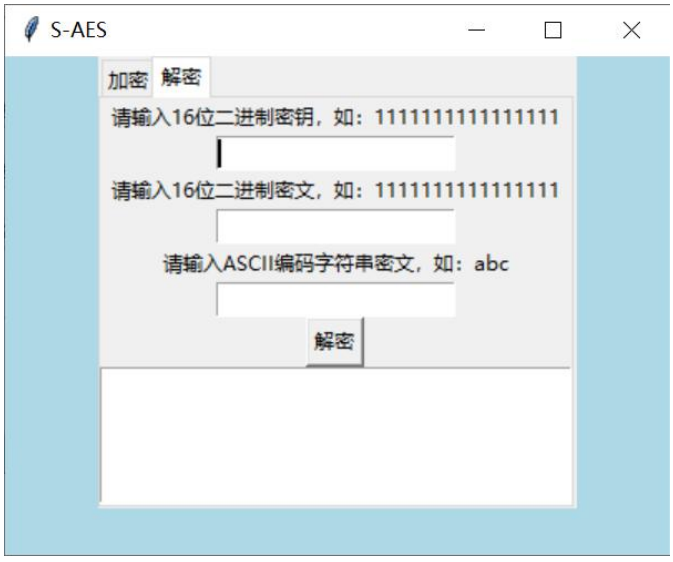
TiMi 小组 S-AES 1-5 关测试结果

成员：戴静、陈晓阳

第 1 关：基本测试

本小组 GUI 主界面如下：

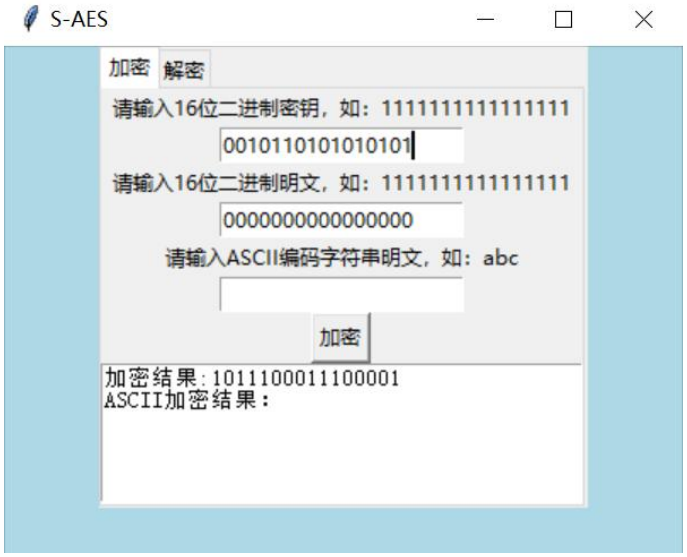


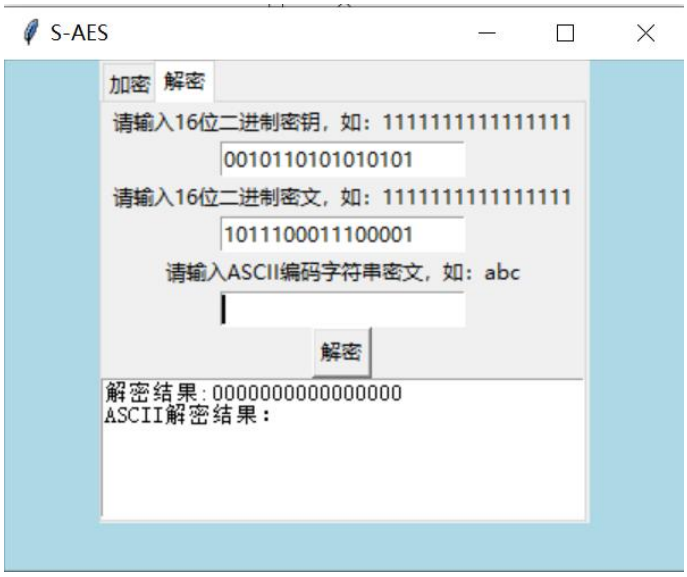


输入部分：加密选项卡输入 16-bit 的密钥、16-bit 的明文（ASCII 编码明文详见第 3 关）；

解密选项卡输入加密选项卡输入 16-bit 的密钥、16-bit 的密文（和 ASCII 编码密文）。

输出结果：加密选项卡输入密钥和明文后点击加密，文本框显示加密后的密文；解密选项卡输入密钥和密文后点击解密，文本框显示解密后的明文。





由上两图可见，加密前的明文和解密后的明文保持一致，说明加解密过程无误。第 1 关测试完成。

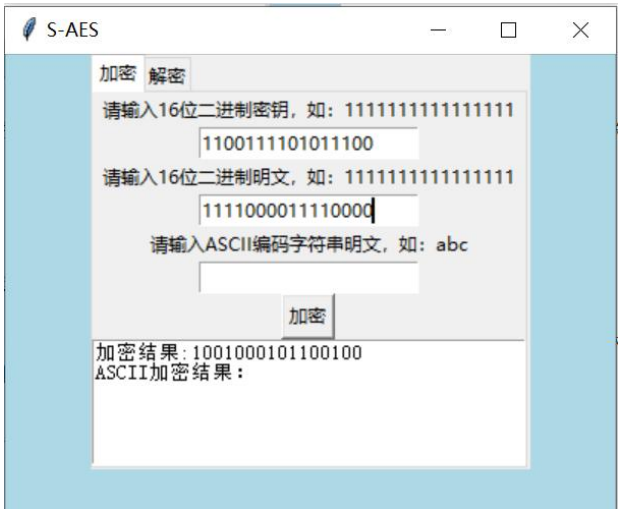
第 2 关：交叉测试

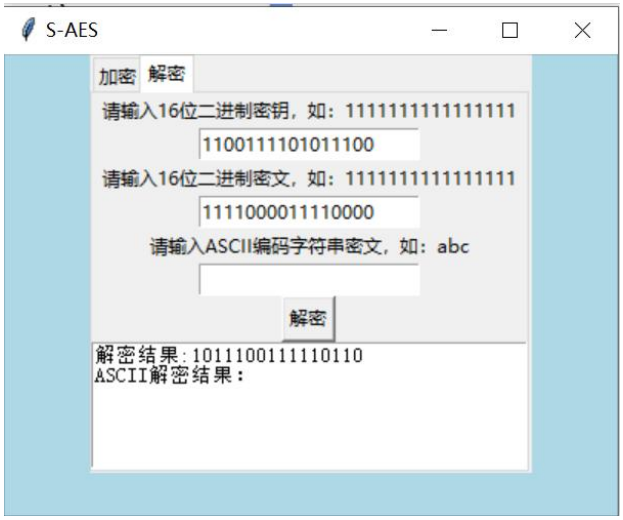
本小组与窦一冉组、鲁梦瑶组、唐豪组进行交叉测试。

测试明文：1111000011110000

测试密钥：1100111101011100

本组结果：

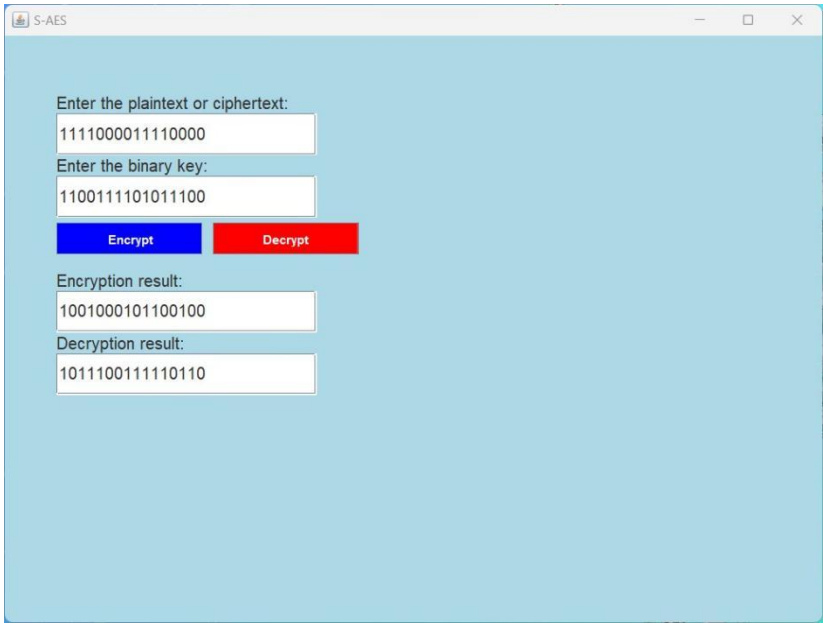




窦一冉组结果（加密）：

鲁梦瑶组结果（解密）：

唐豪组结果（加密）：



由上面四组加密结果截图可见，加密后密文均为 1001000101100100，解密后明文仍为 1111000011110000，符合交叉测试的通过要求。第 2 关测试完成。

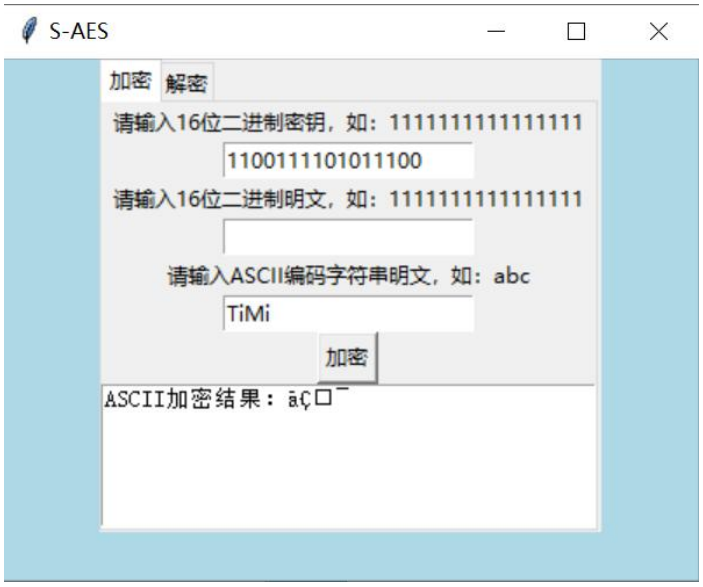
第 3 关：扩展功能

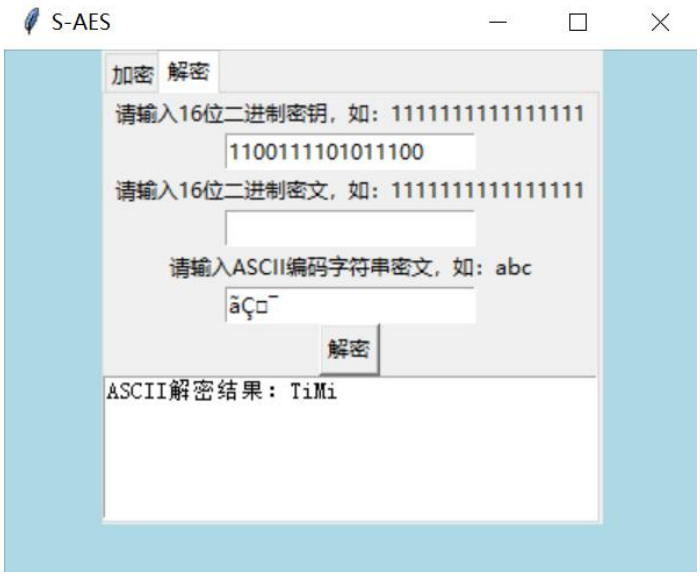
考虑到向实用性扩展，加密算法的数据输入可以是 ASII 编码字符串(分组为 2 Bytes)，对应地输出也可以是 ASCII 字符串(很可能是乱码)。本组成功实现了该扩展功能，具体方法如下：将 ASCII 字符串转化为二进制字符串，并以 2 Bytes 为一组对该二进制字符串进行循环加密，得到加密后的二进制字符串密文。随后将二进制字符串密文重新转化为 ASCII 字符串输出。解密同理。

输入部分：加密选项卡输入 16-bit 的密钥和 ASCII 编码明文；

解密选项卡输入加密选项卡输入 16-bit 的密钥和 ASCII 编码密文。

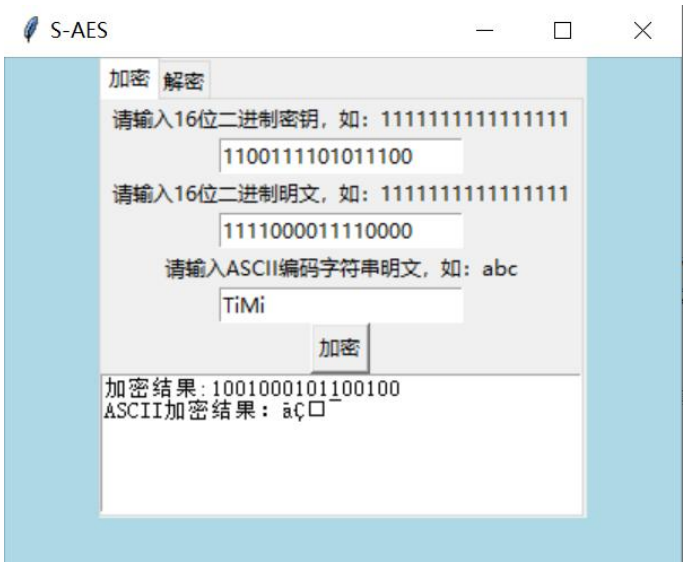
输出结果：加密选项卡输入密钥和 ASCII 明文后点击加密，文本框显示加密后的 ASCII 密文；解密选项卡输入密钥和 ASCII 密文后点击解密，文本框显示解密后的 ASCII 明文。

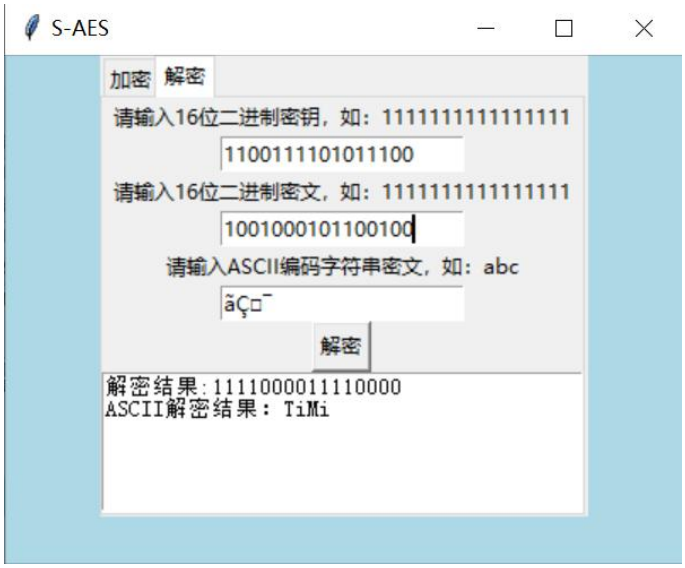




由上两图可见，加密前的明文和解密后的明文保持一致，说明加解密过程无误。第 3 关测试完成。

综合第 1 关和第 3 关，本组的 GUI 实现了普通 16-bit 二进制字符串和 ASCII 编码字符串的同时加\解密，并可以同时显示加\解密结果。效果如下：





第 4 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。

本组使用 Key(K1+K2)的 32-bit 密钥，使用两重 encryption 函数进行双重加密，效果如下：

二重加密：
请输入16位二进制明文或4位十六进制明文：*0000111100001111*
请输入32位二进制密钥或8位十六进制密钥：*00000000000000010010110101010101*
本次二重加密的密文为： 1001110100000000

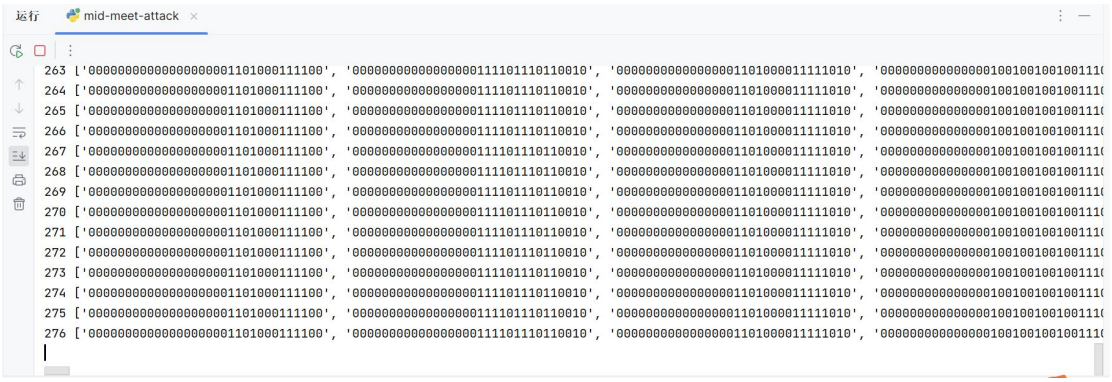
二重解密：
请输入16位二进制明文或4位十六进制密文：*1001110100000000*
请输入32位二进制密钥或8位十六进制密钥：*00000000000000010010110101010101*
本次二重解密的明文为： 0000111100001111

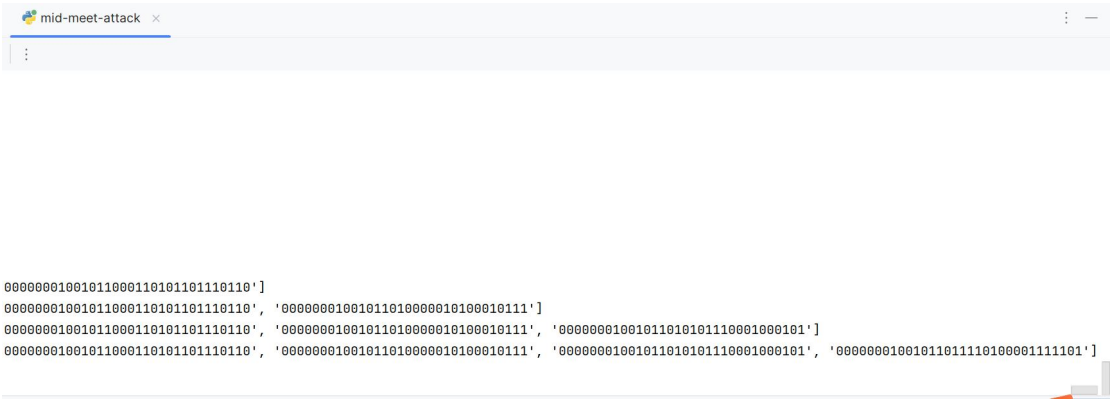
4.2 中间相遇攻击

我们使用已知的使用相同密钥的明、密文对(一个或多个)，使用中间相遇攻击的方法进行暴力破解，找到所有正确的密钥 Key(K1+K2)。以下是一个示例：

已知明文 known_plaintext = "0000111100001111"
已知密文 known_ciphertext = "1001110100000000"

由截图可见中间相遇攻击可以找到多个可能的密钥，截图为暴力破解运行部分截图：





由图可见，k1 外循环进行到一半左右，已生成 276 个可能的密钥。完整密钥过于冗余，此处不进行展示。

任选一个可能的密钥进行验证，加解密成功。

```
print(double_encryption("0000111100001111","00000000000000000111101110110010"))
print(double_decryption("1001110100000000","00000000000000000111101110110010"))
```

```
double_aes
C:\Users\achen\.conda\envs\ve\python.exe "D:\My Project\Cryptography\S-AES\double_aes.py"
1001110100000000
0000111100001111
```

4.3 三重加密

将 S-AES 算法通过三重加密进行扩展，本组选择“按照 32 bits 密钥 Key(K1+K2)的模式进行三重加密解密”的模式进行加解密，原理如下：

- $K = \{K_1, K_2\}$
- $|K| = 112\ bits$
- $C = E\left(K_1, D\left(K_2, E(K_1, P)\right)\right)$

加解密效果如下：

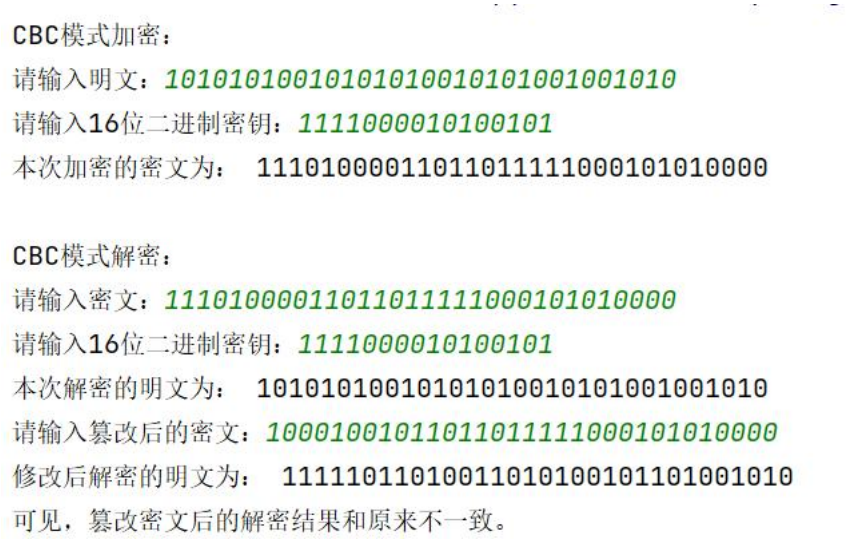
```
三重加密：
请输入16位二进制明文或4位十六进制明文：0000111100001111
请输入32位二进制密钥：00000000000000010010110101010101010
本次三重加密的密文为：  1001101001011000

三重解密：
请输入16位二进制明文或4位十六进制密文：1001110100000000
请输入32位二进制密钥：00000000000000010010110101010101010
本次三重解密的明文为：  1001001110010000
```

第 4 关测试完成。

第 5 关：工作模式

本组编写了在 CBC 模式下进行加解密的算法，并尝试对密文分组进行替换或修改，然后进行解密。

加解密效果如下：



对比篡改密文前后的解密结果可以发现并不相同。第 5 关测试完成。