

第 1 关：基本测试

根据 S-DES 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 8bit 的数据和 10bit 的密钥，输出是 8bit 的密文。

二进制加解密



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(P-Box、S-Box 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。

使用某组同学相应明文与密钥加密：



11101000

1010101010

01011100

得到相同密文，通过交叉测试



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 ASCII 编码字符串(分组为 1 Byte)，对应地输出也可以是 ASCII 字符串(很可能是乱码)。

点击顶部按钮随机生成密钥(密钥直接复制到剪切板，如需使用，粘贴到密钥输入框中即可)



ASCII 加解密





中文 (UTF-16 字符) 加解密



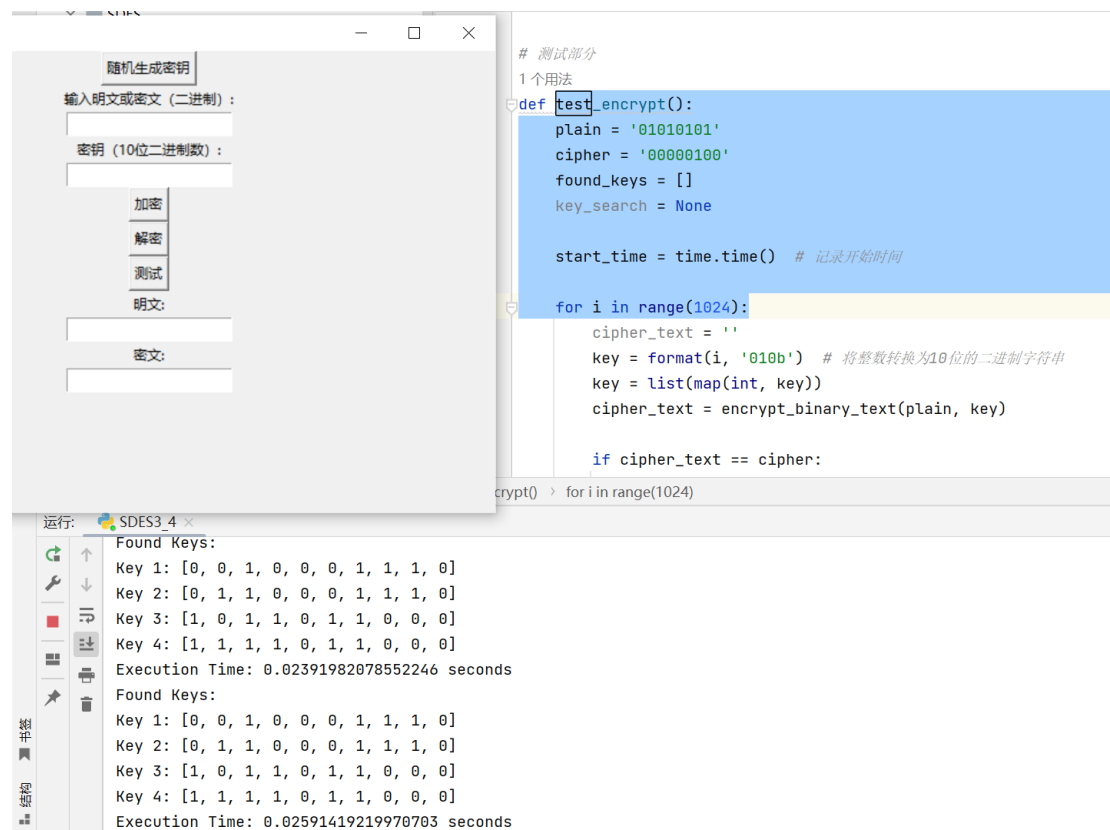
3.4 第4关：暴力破解

假设你找到了使用相同密钥的明、密文对(一个或多个)，请尝试使用暴力破解的方法找到正确的密钥 Key。在编写程序时，你也可以考虑使用多线程的方式提升破解的效率。请设定时间戳，用视频或动图展示你在多长时间内完成了暴力破解。

&

3.5 第5关：封闭测试

根据第4关的结果，进一步分析，对于你随机选择的一个明密文对，是不是有不只一个密钥 Key？进一步扩展，对应明文空间任意给定的明文分组 $P_{\{n\}}$ ，是否会出现选择不同的密钥 $K_{\{i\}}$ 加密得到相同密文 C_n 的情况？



The screenshot displays a Python application window titled "SDES3_4" with a GUI and a terminal window showing the execution results.

GUI Interface:

- Buttons: 随机生成密钥 (Randomly generate key), 加密 (Encrypt), 解密 (Decrypt), 测试 (Test).
- Input fields: 输入明文或密文 (二进制): (Enter plaintext or ciphertext (binary)), 密钥 (10位二进制数): (Key (10-bit binary number)).
- Output fields: 明文: (Plaintext), 密文: (Ciphertext).

Python Code (test_encrypt function):

```
# 测试部分
1个用法
def test_encrypt():
    plain = '01010101'
    cipher = '00000100'
    found_keys = []
    key_search = None

    start_time = time.time() # 记录开始时间

    for i in range(1024):
        cipher_text = ''
        key = format(i, '010b') # 将整数转换为10位的二进制字符串
        key = list(map(int, key))
        cipher_text = encrypt_binary_text(plain, key)

        if cipher_text == cipher:
```

Terminal Output:

```
运行: SDES3_4
Found Keys:
Key 1: [0, 0, 1, 0, 0, 0, 1, 1, 1, 0]
Key 2: [0, 1, 1, 0, 0, 0, 1, 1, 1, 0]
Key 3: [1, 0, 1, 1, 0, 1, 1, 0, 0, 0]
Key 4: [1, 1, 1, 1, 0, 1, 1, 0, 0, 0]
Execution Time: 0.02391982078552246 seconds
Found Keys:
Key 1: [0, 0, 1, 0, 0, 0, 1, 1, 1, 0]
Key 2: [0, 1, 1, 0, 0, 0, 1, 1, 1, 0]
Key 3: [1, 0, 1, 1, 0, 1, 1, 0, 0, 0]
Key 4: [1, 1, 1, 1, 0, 1, 1, 0, 0, 0]
Execution Time: 0.02591419219970703 seconds
```

点击测试按钮，可以找到“test_encrypt”方法中给定的明密文对的所有密钥，并显示验证密钥数量和所花费的时间，具体操作演示可见 github 项目内的“演示.mp4”文件