测试结果

1.实验概述

本次实验是据"信息安全导论"课程第 5 次课讲述的 S-DES 算法，以及所学知识基础上，使用 Python+QT 编程实现加密，解密算法，以及后续的测试闯关。

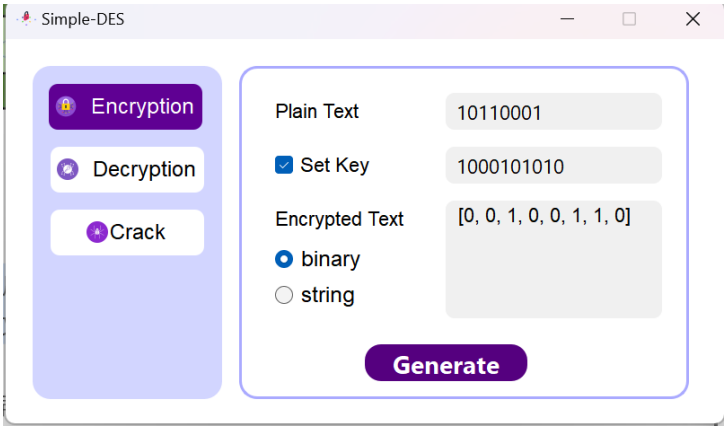
2.测试闯关

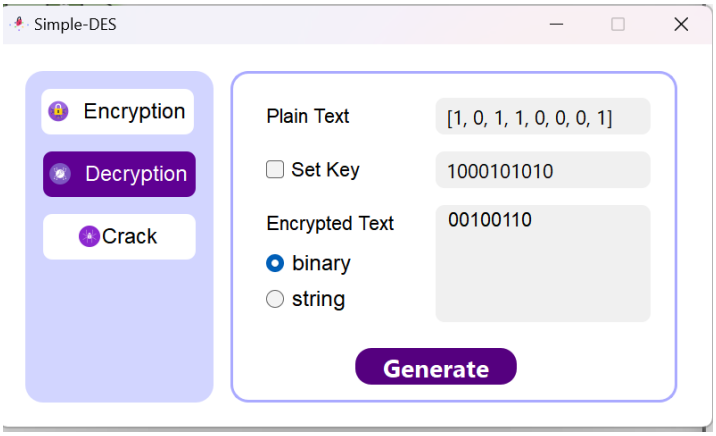
2.1 第 1 关：基本测试

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。

（1）用户交互界面：

（2）对同一对明密文对进行加/解密操作：





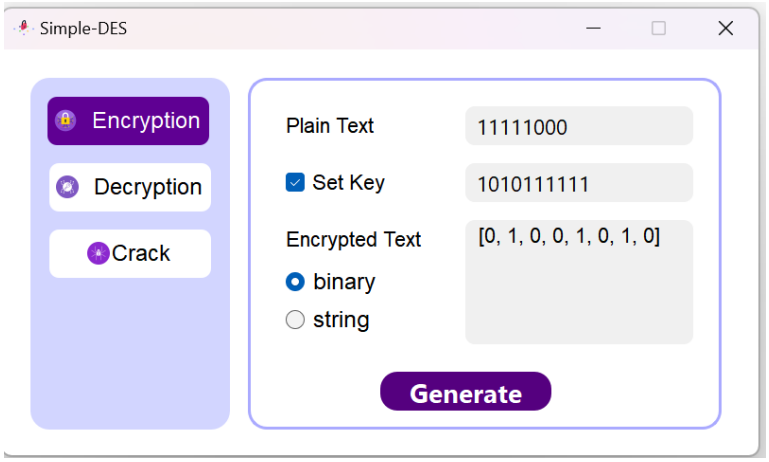
测试结果：加解密完成，测试通过。

2.2 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。我们小组与 L_Q 小组（使用 js 语言完成编程）进行交叉测试：

我们都使用明文：11111000，密钥：1010111111，进行密文生成的测试，如果结果相同则测试通过：

我们小组测试结果：

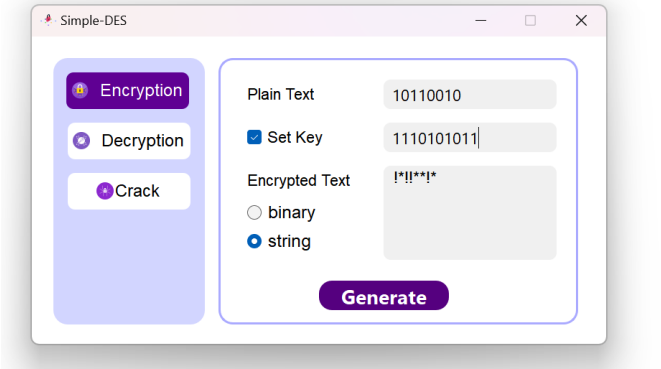


L_Q 小组测试结果：

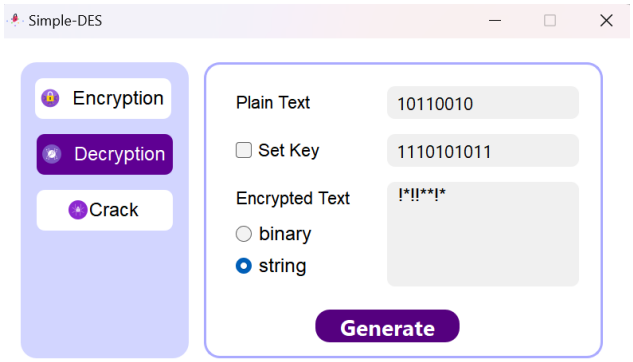
测试结果：双方的加密结果一致，测试通过。

2.3 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)。在我们的界面中设计了以符文输出的选项，在这里只需要调整选择就能输出相应的符文加密结果：



解密结果：

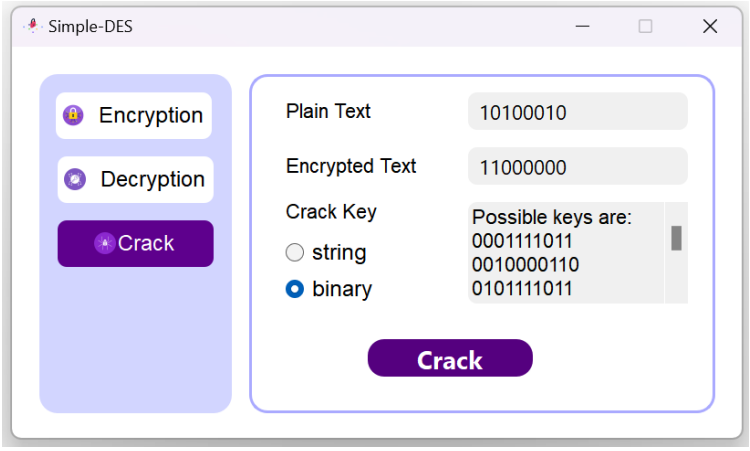


从测试结果可以看出测试通过。

2.4 第 4 关：暴力破解

假设我们找到了使用相同密钥的明、密文对(一个或多个)，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，我们也使用多线程的方式提升破解的效率。设定时间戳，用进度条展示完成了暴力破解的过程。

暴力破解：

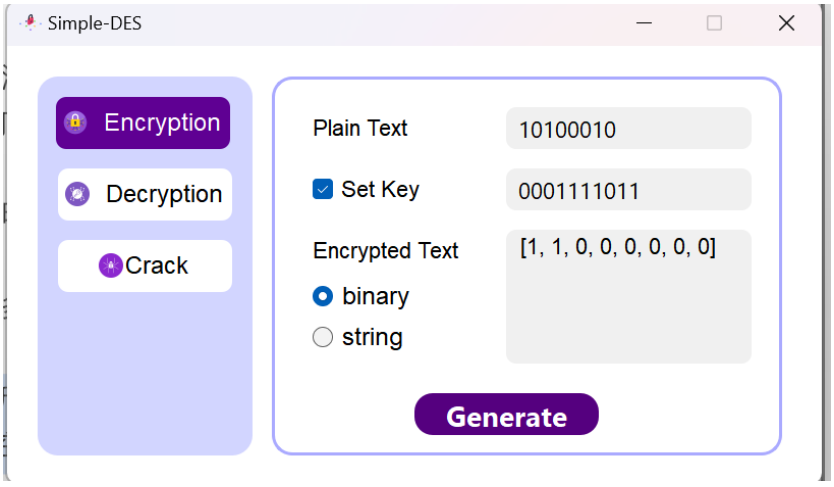


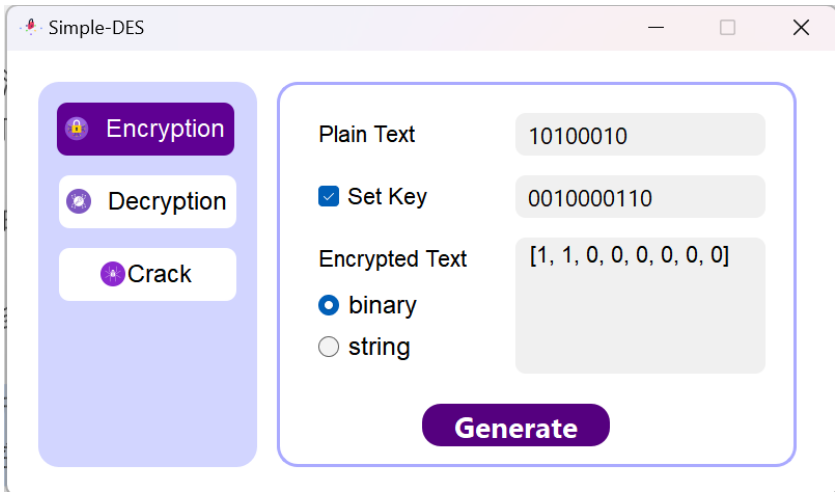
进度条与时长：

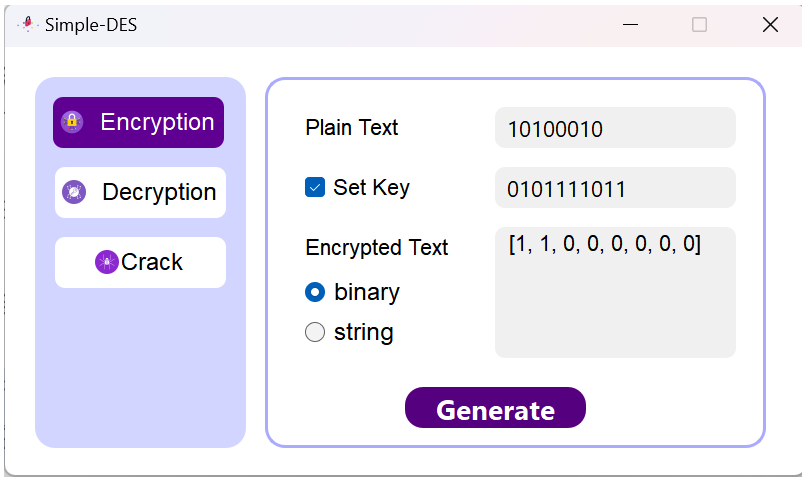
```
一共找到 4 个密钥：
0001111011
0010000110
0101111011
0110000110
进度：100%: ████████████████████████████████████████花了：0.01602959632873535 s
```

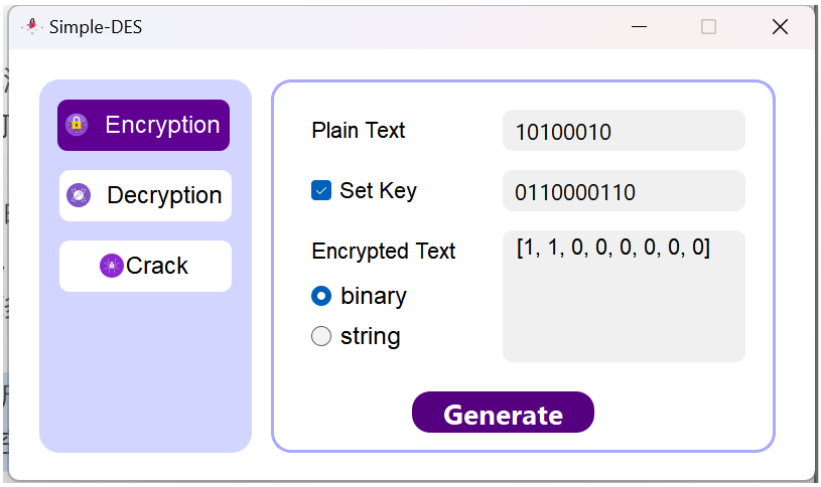
2.5 第 5 关：封闭测试

根据第四关得到的结果，随机选择得到的明密文对存在多个密钥 Key，故进一步扩展肯定会存在给定的明文加密得到相同的密文，在这里使用上一关得到的 4 个密钥加密上一关的明文，看是否得到相同的密文：









测试结果：四次得到的结果完全一致，说明存在上述的情况，完成封闭测试。