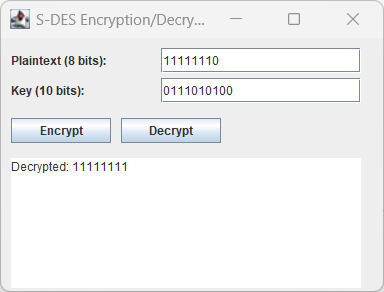
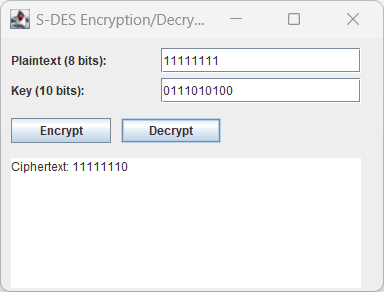
测试结果

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。

例如对以下数据对进行加解密基本测试，所得结果符合预期。

|  |  |  |
| --- | --- | --- |
| 明文 | 密钥 | 密文 |
| 11111111 | 0111010100 | 11111110 |



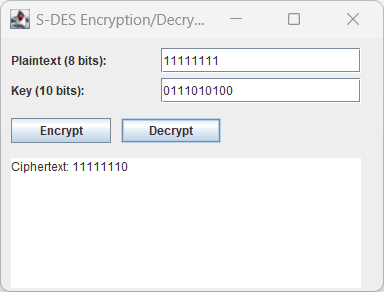
第2关：交叉测试

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。

选择如下明文和密钥进行交叉测试：

|  |  |
| --- | --- |
| 明文 | 密钥 |
| 11111111 | 0111010100 |

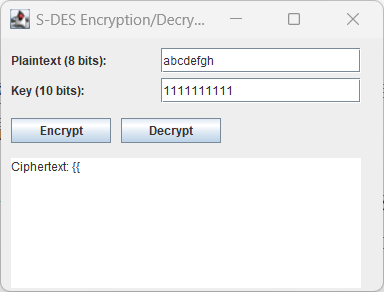


两组程序生成的密文相同，交叉测试通过。

第3关：扩展功能

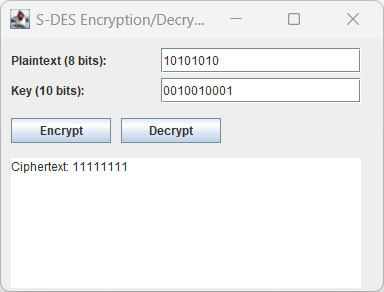
考虑到向实用性扩展，加密算法的数据输入可以是ASII编码字符串(分组为1 Byte)，对应地输出也可以是ACII字符串(很可能是乱码)。

通过编码实现，我们设计若输入数据为ASII编码字符串，输出的密文为随机大写字母、小写字母、数字和符号。例如输入abcdefgh，输出为 {{ 。



第4关：暴力破解

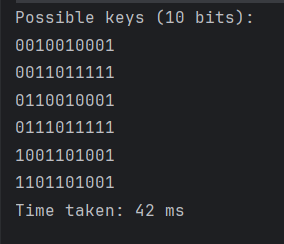
假设你找到了使用相同密钥的明、密文对(一个或多个)，请尝试使用暴力破解的方法找到正确的密钥Key。在编写程序时，你也可以考虑使用多线程的方式提升破解的效率。请设定时间戳，用视频或动图展示你在多长时间内完成了暴力破解。



以如下明密文和密钥为例，现在我们通过暴力破解程序来获得该明密文队的密钥。

|  |  |  |
| --- | --- | --- |
| 明文 | 密钥 | 密文 |
| 10101010 | 0010010001 | 11111111 |

破解结果为：

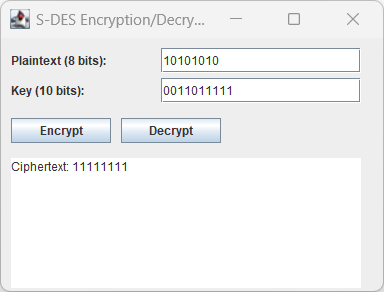
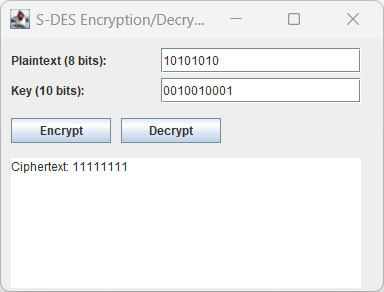


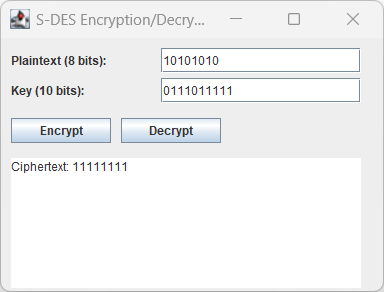
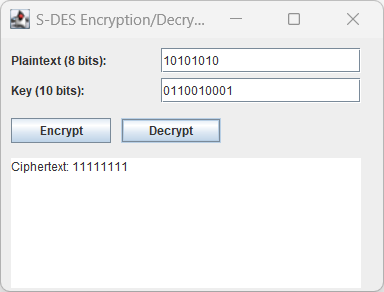
所花时间**42ms**

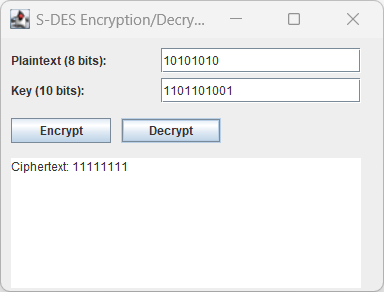
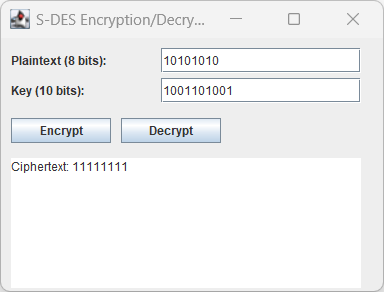
第5关：封闭测试

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，是不是有不止一个密钥Key？进一步扩展，对应明文空间任意给定的明文分组，是否会出现选择不同的密钥加密得到相同密文的情况？

在第四关的暴力破解中，我们遍历得到了明密文对10101010和11111111的6个可能密钥，我们对这6个密钥在加解密程序中进行验证，结果如下。







由此可以得到结论：对于一个已知的明密文对，可能存在多个密钥。