

# S-DES 算法实现报告

## 一、 开发者概括

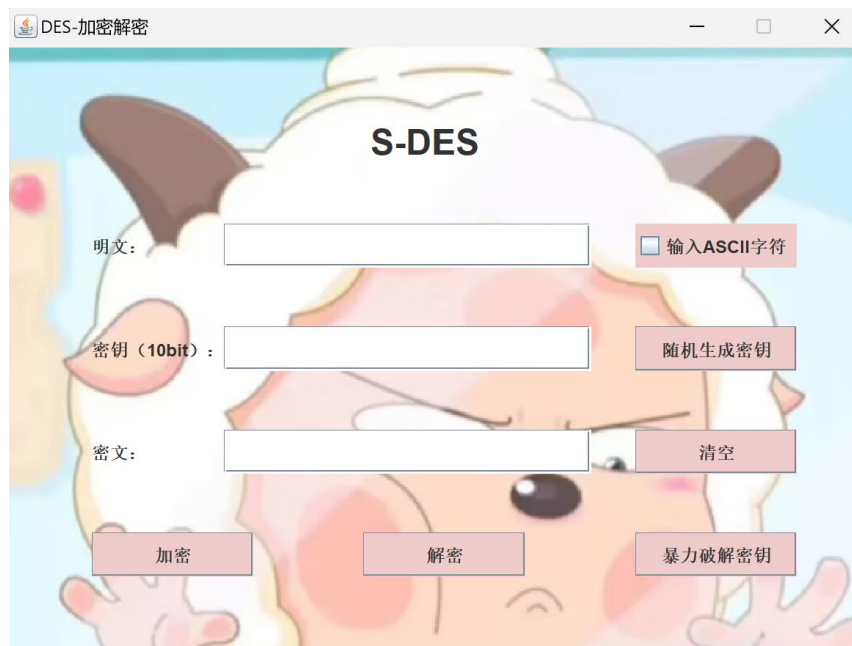
- 任课老师：向宏
- 小组代号：名字还没想好组
- 小组成员：杜瑞杰 20221231 王舟颖 20221459 邓湘 20221770

## 二、 测试报告

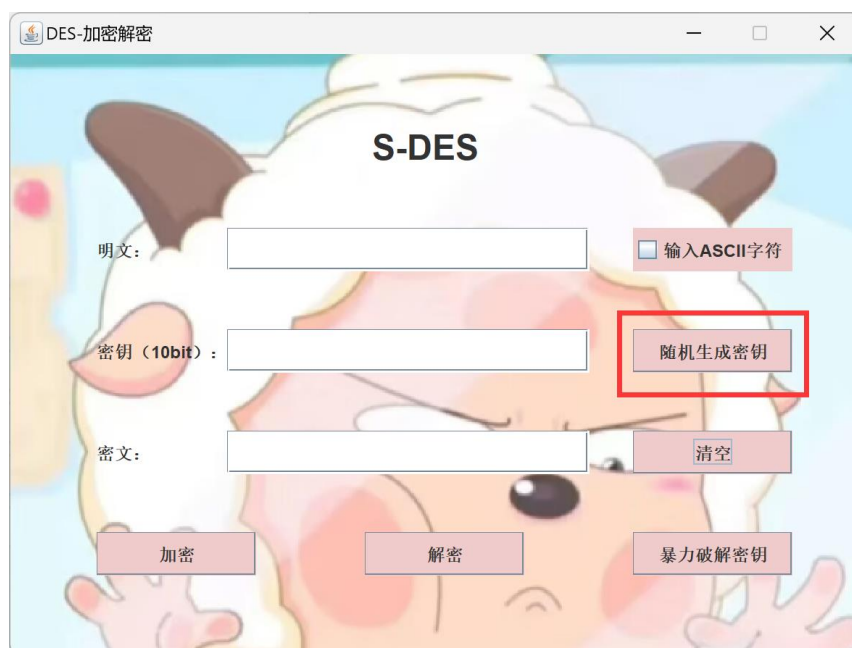
### 第一关：基本测试

根据 S-DES 算法编写和调试程序，提供 GUI 解密支持用户交互。输入可以是任意 bit 的数据（非 8 的整数倍则会在末尾补 0）和 10bit 的密钥，输出是 8 的整数倍 bit 的密文。

## 1.1 用户交互界面



1.2 点击“随机生成密钥”按钮，可以随机生成十位二进制密钥；



注：也可以手动输入密钥！

## 1.3 加密解密测试

该系统还设计了纯二进制加密解密功能及 ASCII 编码字符串加密解密功能。在用户交互界面左上角可以选择。不勾选表示使用纯二进制数字进行加密解密，勾选表示使用 ASCII 编

码字符串进行加密解密。

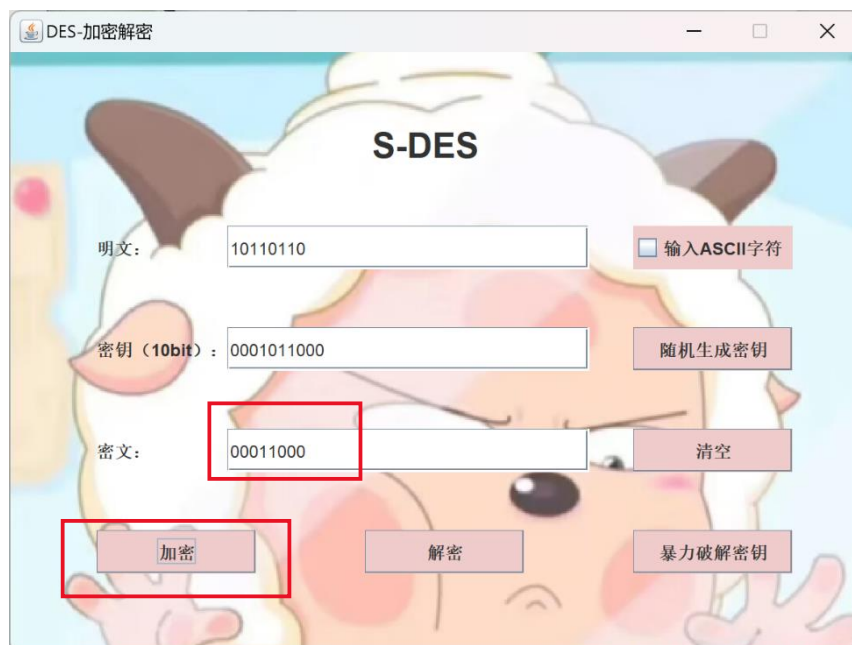
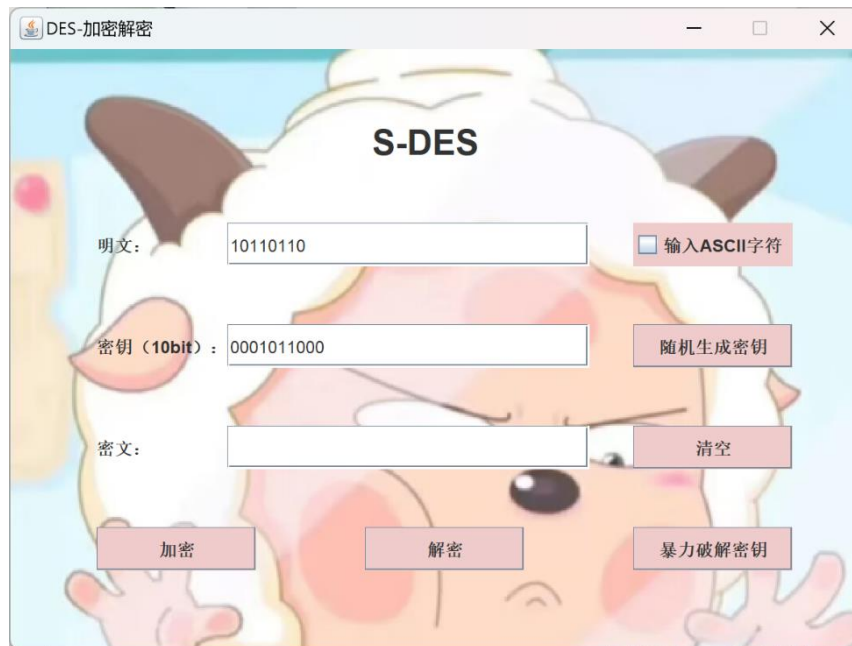
第一关为基础测试，展示使用纯二进制数字进行加密解密的功能；

### 1.3.1 加密测试

随机生成密钥：0001011000

明文：10110110

求得密文：00011000

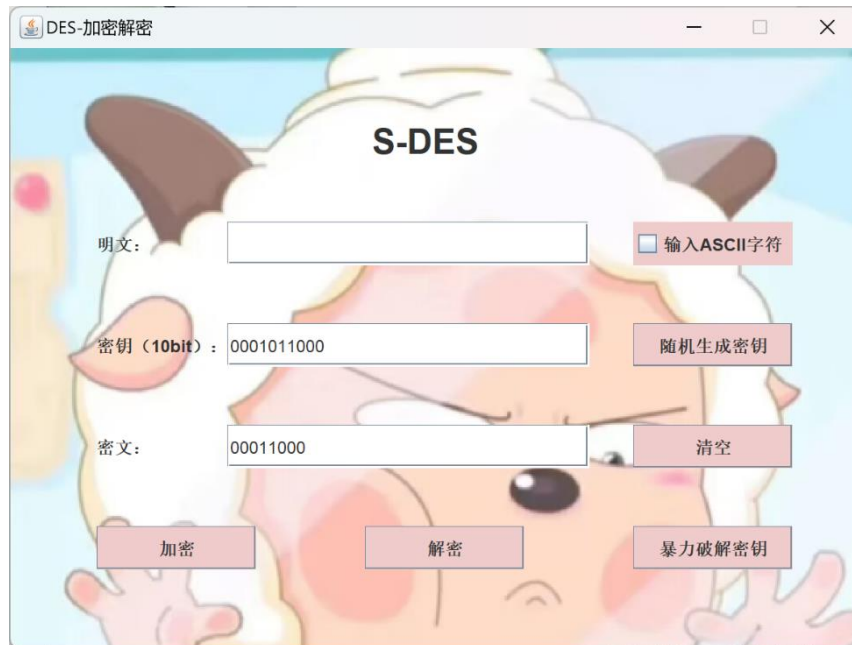


### 1.3.2 解密测试

沿用之前密钥：0001011000

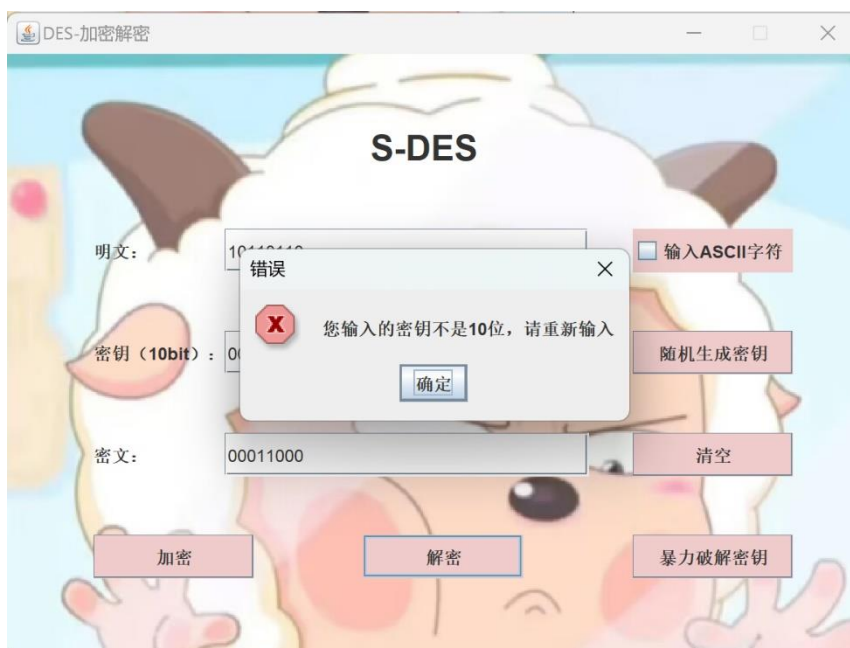
密文：00011000

解得原文：10110110



### 1.3.3 异常处理

当输入明文、密文或密钥不符合规范时；



## 1.4 总结

在此关卡中，我们小组主要实现了以下任务：

### 1. S-DES 算法的理解与实现。

成功掌握了 S-DES 算法的核心内容，如初始置换、轮函数、S 盒与 P 盒等概念，并开发了一个程序，可以根据输入的数据和 10 位密钥完成 S-DES 加密和解密过程。

### 2. GUI 设计与用户交互。

设计并实现了一个简洁友好的图形用户界面（GUI），包含输入数据与密钥的文本框和执行加密或解密、清除数据的按钮，提升了程序的可用性和用户体验。

### 3. 加密与解密功能。

全面实现了 S-DES 算法的加密和解密功能，并进行了系统的测试与调试，确保程序能够在多种情况下正常运行，保证了其正确性和稳定性。

### 4. 实现错误处理，提升用户友好性。

专注于提高界面的易用性，增加了错误提示和反馈机制，帮助用户解决输入或操作过程中遇到的问题，提升了程序的友好性。

## 第二关: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。

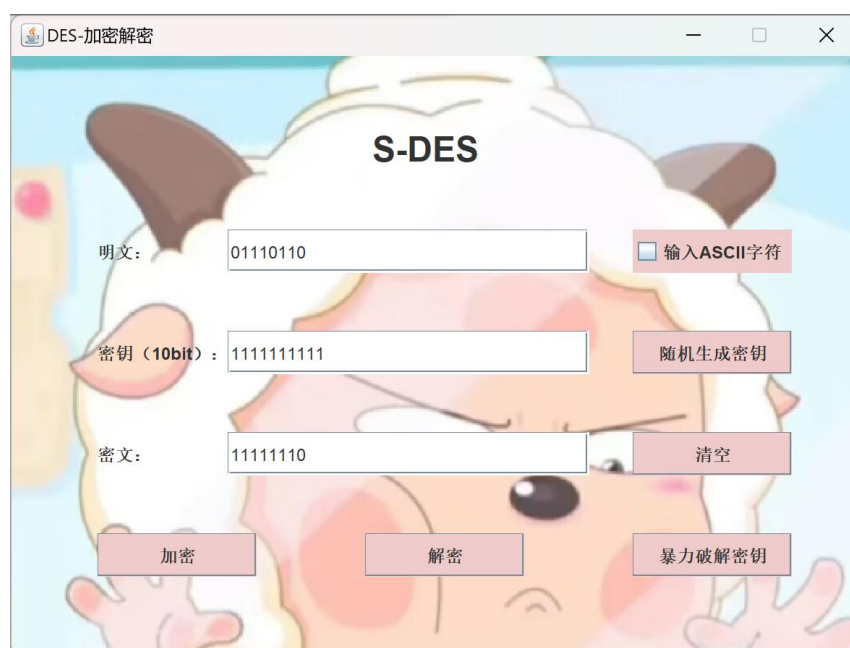
小组与班内默认安全组进行交叉测试，测试结果如下：

#### 1. 我组加密，对方解密

我组使用的明文为：01110110

对方使用的密钥为：1111111111

密文为：11111110



对方使用相同的密钥对密文 11111110 进行解密，解密结果为 01110110，与我组加密时使用的明文相同。

[介绍](#)[服务](#)[DES-加密解密](#)[统计分析](#)[暴力破解](#)[关于我们](#)

### S-DES加解密

#### 输入

请选择您的明/密文类型

☒ bit ☐ ASCII

请输入您的明/密文

11111110

加密解密

#### 密钥

请输入您的密钥

注意：请输入10bit!

1111111111

确认

#### 输出

以下为您的明/密文结果



0,1,1,1,0,1,1,0

## 2. 对方加密，我组解密

对方使用的明文为：10110001

对方使用的密钥为：1000111010

密文为：01100111

[介绍](#)[服务](#)[DES-加密解密](#)[统计分析](#)[暴力破解](#)[关于我们](#)

#### 输入

请选择您的明/密文类型

☒ bit ☐ ASCII

请输入您的明/密文

10110001

加密解密

#### 密钥

请输入您的密钥


注意：请输入10bit!

1000111010

确认

#### 输出

以下为您的明/密文结果



01100111

我组使用相同的密钥对密文 01100111 进行解密，解密结果为 10110001，与对方加密时使用的明文相同。





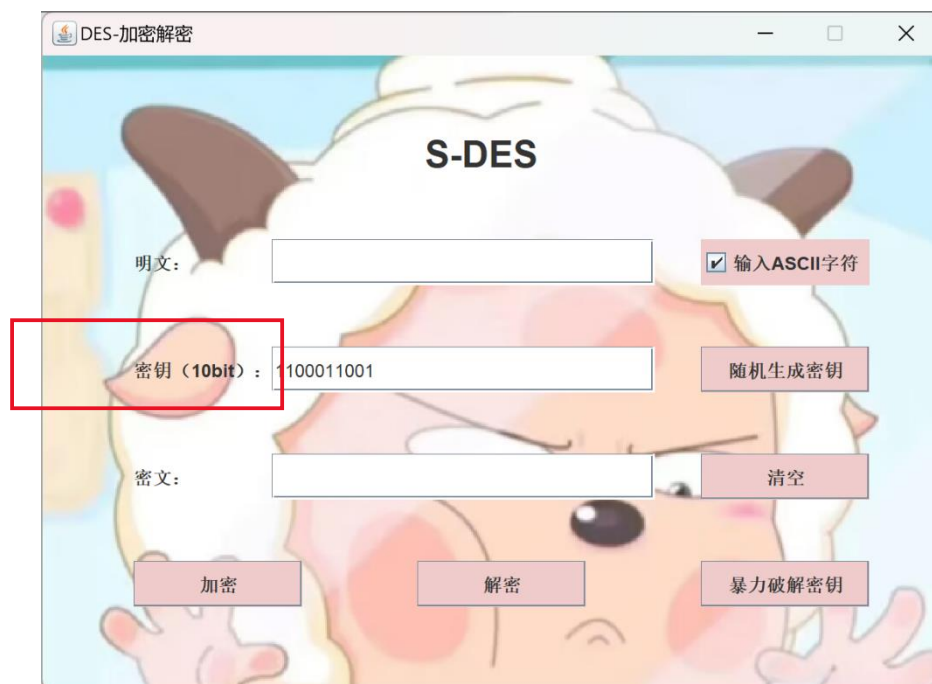
交叉测试成功，双方能够互相加密解密。

## 第三关:拓展功能

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)。

### 3.1 随机生成密钥

密钥: 1100011001





## 3.2 ASCII 字符加密解密测试

### 3.2.1 加密测试

密钥: 1100011001

原文: uysachi

求得密文如下图所示。

The screenshot shows a web application titled "S-DES" with a background of a cartoon sheep. The interface includes the following elements:

- 明文 (Plaintext):** A text input field containing "uysachi".
- 密钥 (10bit) (Key):** A text input field containing "1100011001".
- 密文 (Ciphertext):** A text input field containing "ÿ QKòX+", which is highlighted with a red rectangular box.
- Buttons:** "加密" (Encrypt), "解密" (Decrypt), "暴力破解密钥" (Brute force key), "随机生成密钥" (Randomly generate key), "清空" (Clear), and a checkbox "输入ASCII字符" (Input ASCII characters) which is checked.

### 3.2.2 解密测试

密钥: 1100011001

密文: ÿ QKòX÷

求得明文如下图所示。

The screenshot shows the same "S-DES" web application. The interface elements are:

- 明文 (Plaintext):** A text input field containing "uysachi", which is highlighted with a red rectangular box.
- 密钥 (10bit) (Key):** A text input field containing "1100011001".
- 密文 (Ciphertext):** A text input field containing "ÿ QKòX÷".
- Buttons:** "加密" (Encrypt), "解密" (Decrypt), "暴力破解密钥" (Brute force key), "随机生成密钥" (Randomly generate key), "清空" (Clear), and a checked checkbox "输入ASCII字符" (Input ASCII characters).

### 3.3 总结

在本阶段任务中，我们成功在原有的加密解密系统基础上，实现了对 ASCII 编码的扩展。通过修改加密算法，系统现在可以接受 ASCII 编码的字符串输入，其中每个字符代表一个字节。这一改进使用户能够输入文本数据，而不仅限于传统的二进制数据。此外，我们还将加密后的结果以 ASCII 字符串的形式输出，尽管输出可能会包含乱码字符，但我们成功地将加密结果以文本形式展示，便于用户理解和操作。

## 第四关:暴力破解

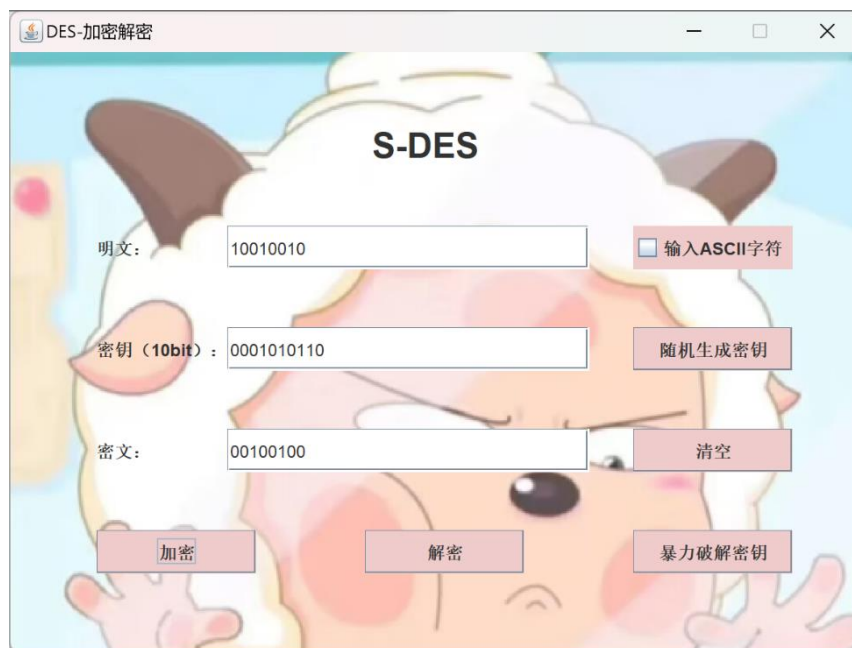
假设你找到了使用相同密钥的明、密文对(一个或多个)，请尝试使用暴力破解的方法找到正确的密钥 Key。在编写程序时，你也可以考虑使用多线程的方式提升破解的效率。请设定时间戳，用视频或动图展示你在多长时间内完成了暴力破解。

### 4.1 生成一组明、密文对

明文: 10010010

密钥: 0001010110

密文: 00100100



### 4.2 暴力破解测试

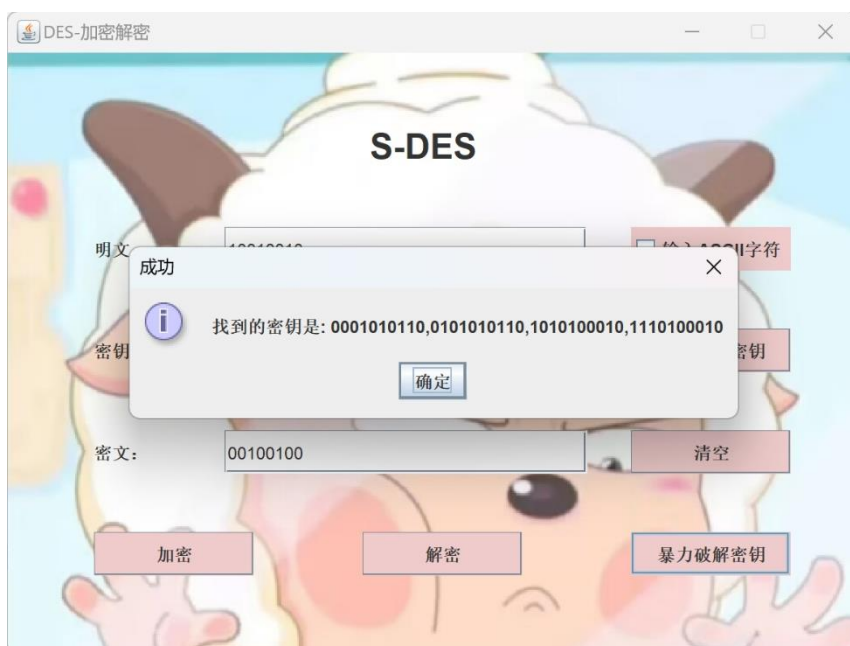
破解所得到的可能的密钥:

- ♦ 0001010110
- ♦ 0101010110
- ♦ 1010100010

- 1110100010

包含最初设置的密钥 0001010110。由时间戳可得出结论：暴力破解速度较快。

```
请求暴力破解的时间：2024-10-07 15:21:32
暴力破解完成的时间（找到第1个密钥）：2024-10-07 15:21:32
暴力破解完成的时间（找到第2个密钥）：2024-10-07 15:21:32
暴力破解完成的时间（找到第3个密钥）：2024-10-07 15:21:32
暴力破解完成的时间（找到第4个密钥）：2024-10-07 15:21:32
```



### 4.3 总结

暴力破解是一种基于穷举的攻击，尝试所有可能的密钥组合，以找到正确的密钥。在本关卡中，我们小组对随机生成的一组明、密文对进行了暴力破解，并得到了可能的密钥，可以发现不止一组密钥符合条件。同时，我们组使用了时间戳来记录整个破解过程所需要的时间，方便我们评估破解密钥的耗时情况以及不同安全性级别的密码系统的重要性。

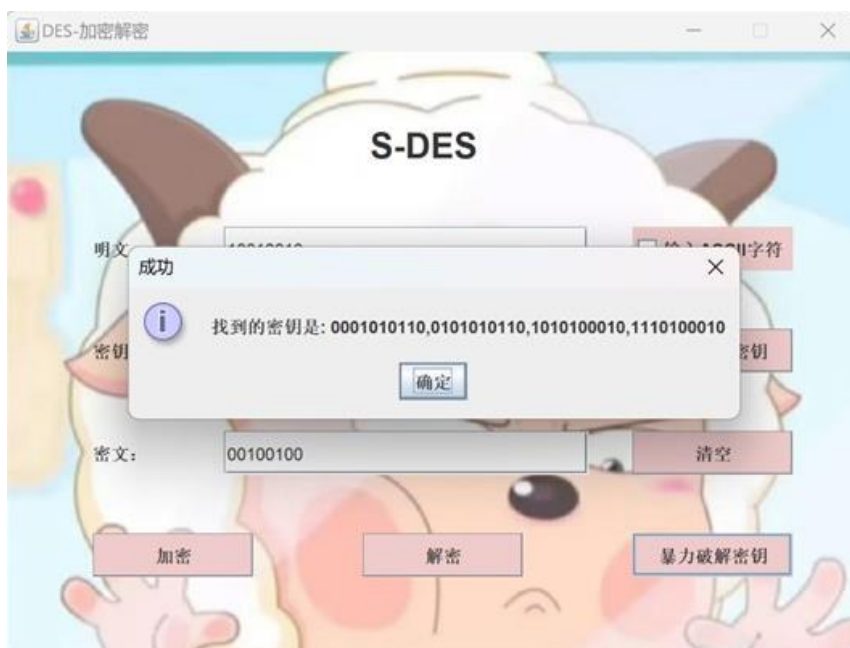
## 第五关:封闭测试

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 P，是否会出现选择不同的密钥 K 加密得到相同密文 C 的情况？

### 5.1 问题 1：对于一个随机选择的明密文对，是否存在不止一个密钥 Key？

对于关卡4中的结果，我们选取的明密文对为：10010010（明文）与 00100100（密文）；经过暴力破解得到可能的密钥为：

- ♦ 0001010110
- ♦ 0101010110
- ♦ 1010100010
- ♦ 1110100010

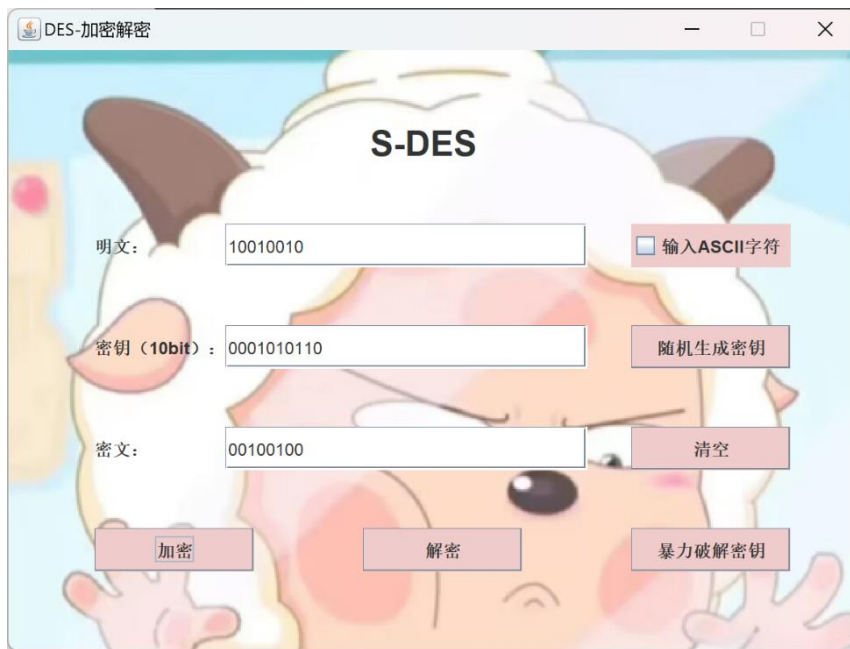


在 S-DES 算法中，由于密钥的长度只有 10 位，密钥空间相对较小，因此存在一定概率多个不同的密钥生成相同的加密结果。此时通过暴力破解密钥，可以发现多个密钥都能正确解密出相同的明文。

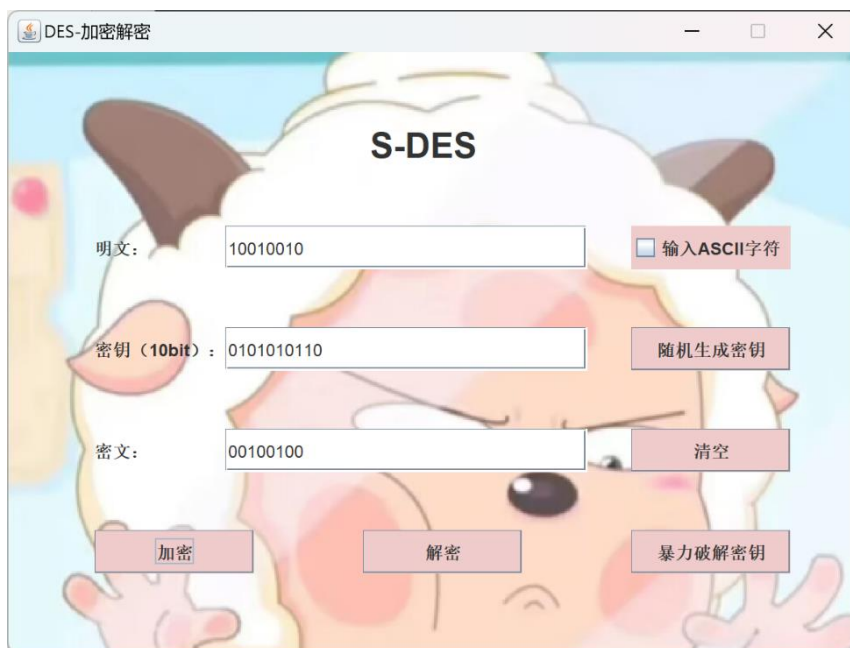
## 5.2 问题 2: 对应明文空间的任意给定明文分组 P，是否会出现选择不同的密钥 K 加密得到相同密文 C 的情况？

利用以上获得的密钥进行测试，得到如下结果：

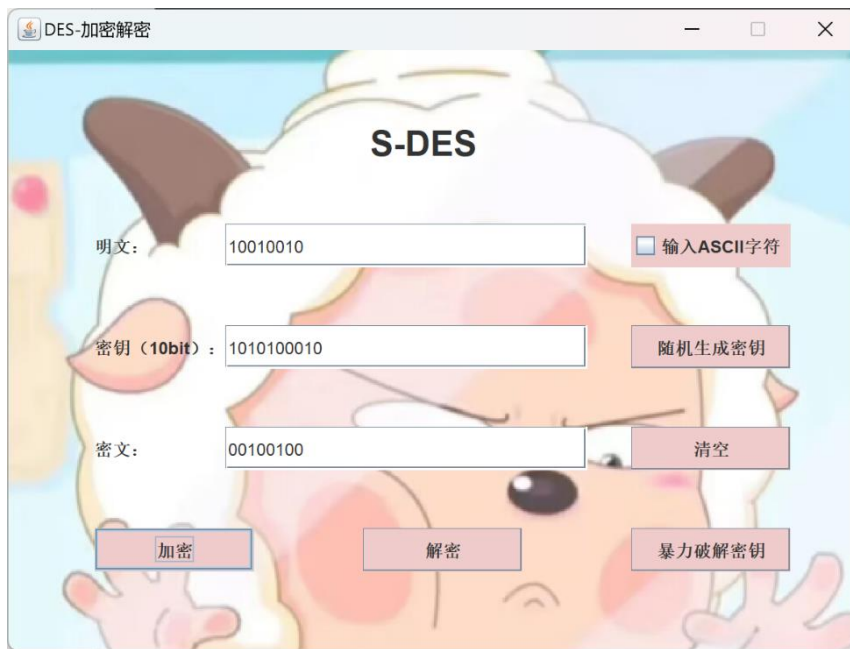
- ① 密钥：0001010110  
明文：10010010  
所得密文：00100100



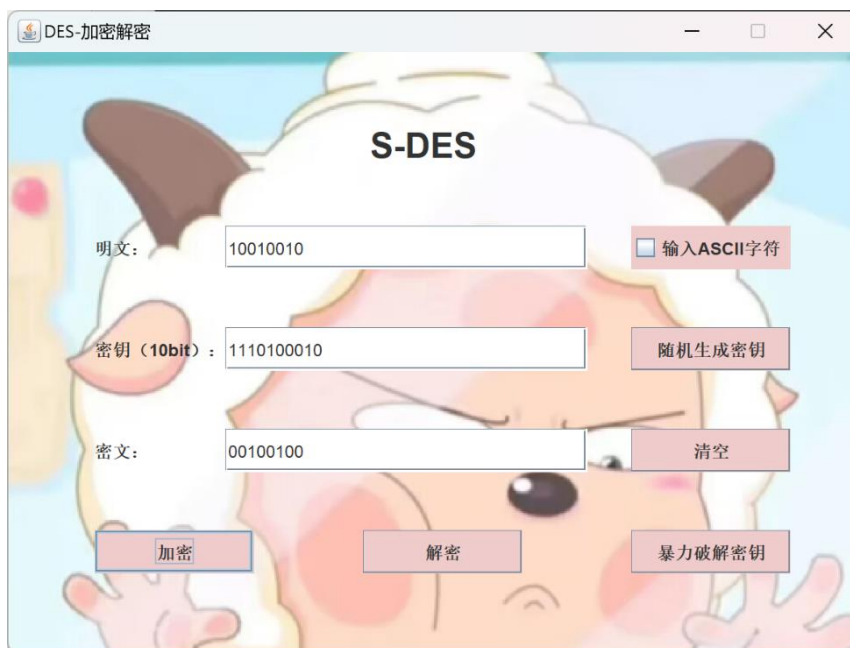
- ② 密钥: 0101010110  
明文: 10010010  
所得密文: 00100100



- ③ 密钥: 1010100010  
明文: 10010010  
所得密文: 00100100



- ④ 密钥: 1110100010  
明文: 10010010  
所得密文: 00100100



以上四个密钥经过测试均能得到最初给出的明文密文对。

S-DES 算法中的密钥长度为 10 位，因此一共有  $2^{10} = 1024$  个不同的密钥。S-DES 的明文分组为 8 位，因此明文空间为  $2^8 = 256$  个不同的明文分组。同样，由于密文也是 8 位，密文空间也只有  $2^8 = 256$  种不同的密文。

因此，在密钥空间（1024 个密钥）和密文空间（256 种密文）之间，存在一种**密钥碰撞现象**，即可能会有不同的密钥  $K_i \neq K_j$  对相同的明文  $P_n$  进行加密，得到相同的密文  $C_n$ 。



### 5.3 总结

对于任意给定的明文分组 $P_n$ ，我们验证了不同密钥可能会生成相同的密文。这种**密文碰撞**不仅削弱了加密算法的安全性，还为攻击者提供了更多破解路径。通过枚举密钥空间，攻击者可以找到多个可能的密钥，从而增加了逆向推导密钥的机会。这说明 S-DES 由于其有限的密钥和明文空间，在实际应用中的安全性存在不足。