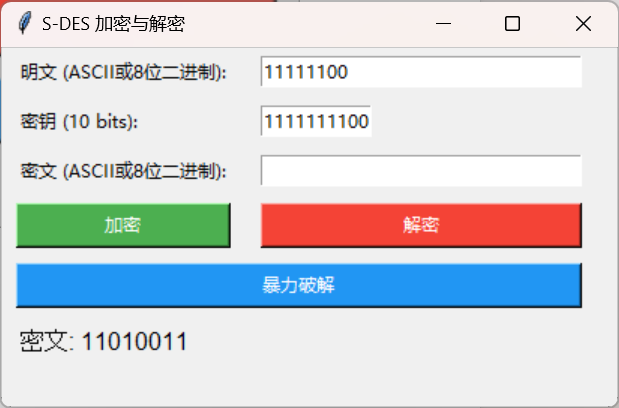
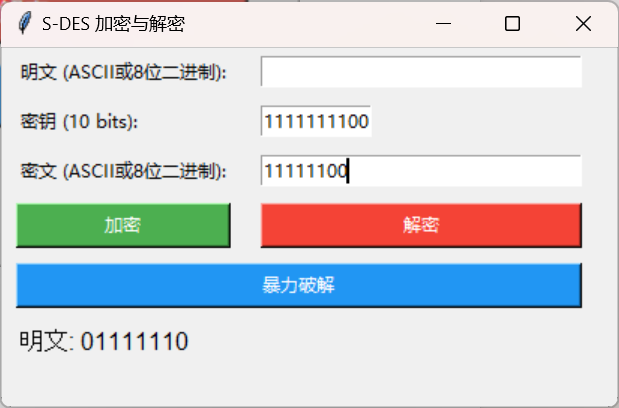
# s-des1

第一关

加密



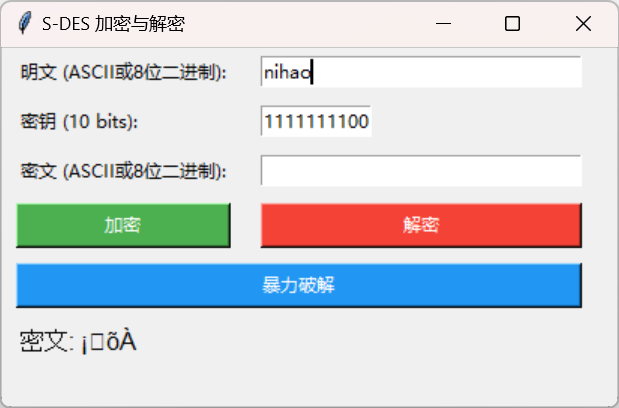
解密



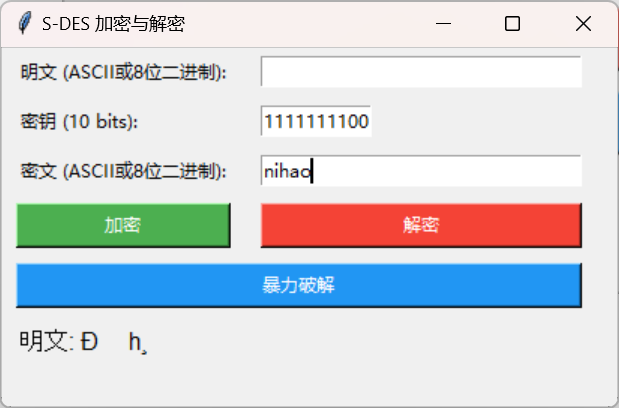
第二关

第三关

加密



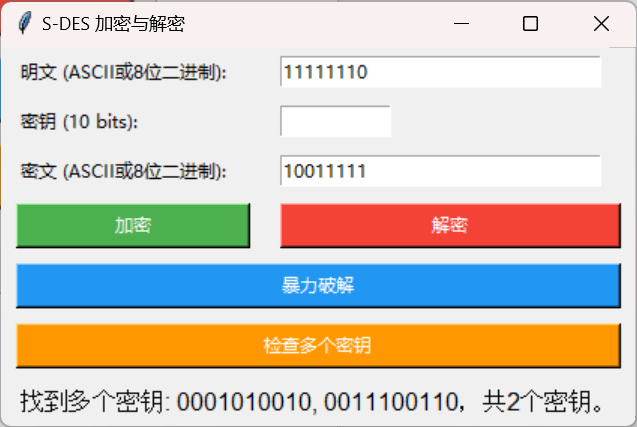
解密



第四关



第五关



S-DES 加密与解密系统用户手册

1. 系统概述

本系统实现了一个简化的数据加密标准（Simplified Data Encryption Standard, S-DES），并提供了一个图形用户界面（GUI）来执行加密、解密、暴力破解和检查多个密钥的功能。S-DES是一种用于教学目的的简化版DES算法，其密钥长度为10位，数据块长度为8位。

2. 系统结构

前端：使用Python的tkinter库构建GUI界面，允许用户输入明文、密钥和密文，并显示结果。

后端：实现了S-DES算法的加密、解密、密钥生成以及多线程暴力破解等功能。

3. 功能描述

加密：将用户输入的明文和密钥进行S-DES加密，输出密文。

解密：将用户输入的密文和密钥进行S-DES解密，输出明文。

暴力破解：通过尝试所有可能的密钥来找到能够将给定明文加密成给定密文的密钥。

检查多个密钥：检查是否存在多个不同的密钥可以将给定的明文加密成相同的密文。

4. 代码解析

4.1 常量定义

python

深色版本

IP = [2, 6, 3, 1, 4, 8, 5, 7]IP\_1 = [4, 1, 3, 5, 7, 2, 8, 6]P10 = [3, 5, 2, 7, 4, 9, 1, 10, 8, 6]P8 = [6, 3, 7, 4, 8, 5, 10, 9]EP = [4, 1, 2, 3, 2, 3, 4, 1]P4 = [2, 4, 3, 1]S0 = [ [1, 0, 3, 2], [3, 2, 1, 0], [0, 2, 1, 3], [3, 1, 0, 2]]

S1 = [ [0, 1, 2, 3], [2, 3, 1, 0], [3, 0, 1, 2], [2, 1, 0, 3]]

这些常量定义了S-DES中使用的置换表和S盒。

4.2 主要函数

置换函数 permute(block, table)：根据给定的置换表对块进行置换。

左移函数 left\_shift(key, n)：将密钥左移n位。

密钥生成 generate\_keys(key)：生成两个子密钥K1和K2。

轮函数F f\_k(block, subkey)：实现S-DES中的轮函数。

S-DES加密 sdes\_encrypt(plaintext, key)：使用S-DES算法对明文进行加密。

S-DES解密 sdes\_decrypt(ciphertext, key)：使用S-DES算法对密文进行解密。

ASCII字符串到二进制字符串的转换 ascii\_to\_binary(ascii\_str)：将ASCII字符串转换为二进制字符串。

二进制字符串到ASCII字符串的转换 binary\_to\_ascii(binary\_str)：将二进制字符串转换为ASCII字符串。

暴力破解单个块 brute\_force\_single\_block(plaintext, ciphertext, start, end)：在指定范围内暴力破解单个块。

多线程暴力破解 brute\_force\_attack(plaintext, ciphertext, num\_threads=4)：使用多线程进行暴力破解。

4.3 GUI类 SDESApp

初始化：设置GUI布局，包括标签、输入框、按钮和结果标签。

加密方法 encrypt：处理加密操作。

解密方法 decrypt：处理解密操作。

暴力破解方法 brute\_force：处理暴力破解操作。

检查多个密钥方法 check\_multiple\_keys：检查是否存在多个密钥可以加密同一明文得到相同的密文。

5. 接口文档

5.1 加密接口

功能：对明文进行S-DES加密。

输入：

plaintext (str)：明文，可以是ASCII字符串或8位二进制字符串。

key (str)：10位二进制密钥。

输出：

ciphertext (str)：密文，格式与输入相同（ASCII或二进制）。

示例：

python

深色版本

plaintext = "10101010"key = "1010101010"ciphertext = sdes\_encrypt(plaintext, key)print(f"密文: {ciphertext}")

5.2 解密接口

功能：对密文进行S-DES解密。

输入：

ciphertext (str)：密文，可以是ASCII字符串或8位二进制字符串。

key (str)：10位二进制密钥。

输出：

plaintext (str)：明文，格式与输入相同（ASCII或二进制）。

示例：

python

深色版本

ciphertext = "01010101"key = "1010101010"plaintext = sdes\_decrypt(ciphertext, key)print(f"明文: {plaintext}")

5.3 暴力破解接口

功能：通过尝试所有可能的密钥来找到能够将给定明文加密成给定密文的密钥。

输入：

plaintext (str)：明文，可以是ASCII字符串或8位二进制字符串。

ciphertext (str)：密文，可以是ASCII字符串或8位二进制字符串。

输出：

key (str)：找到的10位二进制密钥。

示例：

python

深色版本

plaintext = "10101010"ciphertext = "01010101"found\_key = brute\_force\_attack(plaintext, ciphertext)if found\_key: print(f"找到密钥: {found\_key}")else: print("没有找到匹配的密钥。")

5.4 检查多个密钥接口

功能：检查是否存在多个不同的密钥可以将给定的明文加密成相同的密文。

输入：

plaintext (str)：明文，可以是ASCII字符串或8位二进制字符串。

ciphertext (str)：密文，可以是ASCII字符串或8位二进制字符串。

输出：

keys (list of str)：找到的所有10位二进制密钥。

示例：

python

深色版本

plaintext = "10101010"ciphertext = "01010101"keys = check\_multiple\_keys(plaintext, ciphertext)if keys: print(f"找到多个密钥: {', '.join(keys)}，共{len(keys)}个密钥。")else: print("没有找到匹配的密钥。")

6. 使用说明

启动程序：运行提供的Python脚本以启动GUI。

输入明文/密文：在相应的输入框中输入明文或密文。

输入密钥：在密钥输入框中输入10位二进制密钥。

加密/解密：点击“加密”或“解密”按钮来执行相应的操作。

暴力破解：点击“暴力破解”按钮来尝试找到正确的密钥。

检查多个密钥：点击“检查多个密钥”按钮来查看是否存在多个密钥可以加密同一明文得到相同的密文。

7. 注意事项

密钥必须是10位二进制数。

明文和密文必须是8位的倍数，可以是ASCII字符串或二进制字符串。

暴力破解可能会花费较长时间，具体取决于计算资源。