S-DES 加解密系统用户手册

#### 1. 概述

S-DES加密解密工具是一个基于Python和Tkinter开发的图形用户界面(GUI)应用程序，用于执行S-DES算法的加密和解密操作。该工具支持二进制数据的加密和解密，以及ASCII文本的加密和解密。

#### 2. 系统要求

* 操作系统：Windows, macOS, Linux
* Python版本：Python 3.x
* 依赖库：Tkinter

#### 3. 安装和启动

1. 确保Python已安装在您的系统上。
2. 下载sdes.py文件。
3. 打开命令行或终端，导航到包含sdes.py的文件夹。
4. 运行命令python sdes.py启动应用程序。

#### 4. 使用指南

##### 4.1 二进制数据加密/解密

1. 在“请输入8位数据”框中输入8位二进制数据。
2. 在“请输入10位密钥”框中输入10位二进制密钥。
3. 点击“加密”按钮进行加密操作，结果将显示在下方的“加密数据”标签中。
4. 点击“解密”按钮进行解密操作，结果将显示在下方的“解密数据”标签中。

##### 4.2 ASCII数据加密/解密

1. 在“请输入ASCII数据”框中输入需要加密的ASCII文本。
2. 在“请输入10位密钥”框中输入10位二进制密钥。
3. 点击“加密ASCII”按钮进行加密操作，加密后的ASCII数据将显示在下方的“加密的ASCII数据”标签中。
4. 点击“解密ASCII”按钮进行解密操作，解密后的ASCII数据将显示在下方的“解密的ASCII数据”标签中。

##### 4.3 暴力破解

1. 在“密文(二进制)”框中输入8位二进制密文。
2. 在“已知明文(二进制)”框中输入已知的8位二进制明文。
3. 点击“暴力破解”按钮开始破解过程。可能的密钥将显示在“尝试密钥”标签中，进度将显示在“进度”标签中。

5. 注意事项

* 确保输入的二进制数据和密钥格式正确。
* 对于ASCII数据加密/解密，确保输入的文本不包含非ASCII字符。

S-DES 加解密系统开发手册

### **1. 概述**

S-DES加密解密工具使用Python编写，利用Tkinter库构建GUI。该工具实现了S-DES算法的加密和解密功能。

### **2. 开发环境**

* 编程语言：Python
* GUI库：Tkinter
* 版本控制：Git

### **3. 系统架构**

* **主界面**：包含多个部分，用于二进制数据加密/解密、ASCII数据加密/解密和暴力破解。
* **加密/解密功能**：使用S-DES算法对输入的二进制数据或ASCII文本进行加密或解密。
* **暴力破解功能**：尝试所有可能的密钥，以找到能够将给定的密文解密成已知明文的密钥。

### **4. 核心组件**

#### **4.1 S-DES算法实现**

* **置换表和S盒**：定义了S-DES算法中使用的置换表和S盒。
  + P10, P8, IP, invIP, EP, P4, S0, S1
* **置换函数**：实现数据的置换操作。
  + permute(input\_bits, table)
* **左移函数**：实现数据的左移操作。
  + left\_shift(bits, n)
* **密钥生成**：根据输入的10位密钥生成两轮加密所需的子密钥。
  + keygen(key)
* **异或操作**：实现两个二进制字符串的异或操作。
  + xor(bits1, bits2)
* **S盒替换**：根据输入的二进制字符串和S盒进行替换操作。
  + sbox(input\_bits, sbox)
* **f函数**：实现S-DES算法中的f函数。
  + f(right, subkey)
* **单轮加密函数**：实现S-DES算法的一轮加密过程。
  + fk(bits, subkey)
* **加密解密函数**：实现整个S-DES算法的加密和解密过程。
  + encrypt\_decrypt(bits, key, encrypt)

#### **4.2 GUI组件**

* **输入框**：用于输入数据和密钥。
* **按钮**：触发加密、解密、ASCII加密、ASCII解密和暴力破解操作。
* **标签**：显示操作结果和进度信息。

### **5. 开发步骤**

1. **环境搭建**：安装Python和Tkinter库。
2. **算法实现**：编写S-DES算法的核心逻辑。
3. **GUI设计**：使用Tkinter设计用户界面。
4. **功能实现**：将算法逻辑与GUI组件绑定，实现加密、解密、ASCII加密、ASCII解密和暴力破解功能。
5. **测试**：对每个功能进行测试，确保正确性和稳定性。
6. **部署**：打包应用程序，供用户下载和使用。

### **6. 版本更新**

* 使用Git进行版本控制，记录每次更新的详细信息。

### **7. 维护**

* 定期检查并修复发现的问题。
* 根据用户反馈进行功能改进和优化。