S-DES加解密系统文档

用户指南

1. 系统简介

S-DES (Simplified DES) 是一个简化的数据加密标准算法，用于实验作业目的。本系统提供完整的S-DES加解密功能，支持图形化操作。

2. 系统要求

操作系统: Windows 10/11, macOS 10.14+, Ubuntu 18.04+

Python版本: Python 3.7+

依赖库: PyQt5, numpy, matplotlib, pillow

3. 安装步骤

3.1 安装Python依赖

bash

pip install PyQt5 numpy matplotlib pillow

3.2 下载程序文件

确保以下文件在同一目录：

main.py # 主程序入口

SDES.py # 核心算法实现

SDESDialog.py # GUI界面

sdes\_gif\_demo.py # 暴力破解演示

style.qss # 界面样式

3.3 运行程序

python main.py

4. 使用说明

4.1 主界面介绍



4.2 基本加解密操作

步骤1: 输入密钥

在密钥输入框中输入10位密钥

选择密钥格式：二进制、十进制、十六进制、ASCII

步骤2: 输入明文/密文

在左侧明文区域输入待加密文本

或在右侧密文区域输入待解密文本

步骤3: 选择文本编码

在下拉框选择文本编码格式

步骤4: 执行操作

点击"加密"按钮进行加密

点击"解密"按钮进行解密

点击"⇌"按钮交换明文和密文

点击"清空"按钮清除所有输入

4.3 支持的编码格式

密钥编码:

二进制: 10位0/1字符串，如 1010001110

十进制: 0-1023的数字，如 523

十六进制: 0x000-0x3FF，如 1A3

ASCII: 1个ASCII字符，如 K

文本编码:

二进制: 任意长度0/1字符串

十进制: 数字字符串

十六进制: 0-9A-F字符串

ASCII: 可读文本字符串

4.4 暴力破解演示

python sdes\_gif\_demo.py

该功能会生成一个展示暴力破解过程的GIF动画。

5. 常见问题解答

Q1: 密钥验证失败

A: 确保密钥格式正确：

二进制：10位0/1

十进制：0-1023

十六进制：0-3FF

ASCII：1个字符

Q2: 加解密结果不正确

A: 检查文本编码格式是否一致

Q3: 暴力破解动画生成失败

A: 确保已安装matplotlib和pillow

开发手册

1. 系统架构

text

应用层: main.py → SDESDialog.py

↓

业务层: SDES.py (核心算法)

↓

工具层: sdes\_gif\_demo.py

2. 核心组件接口文档

2.1 SDES类 (SDES.py)

类定义:

python

class SDES:

"""S-DES算法核心实现类"""

主要方法:

encrypt(plaintext: str, key: str) -> str

功能: 加密8位二进制数据

参数:

plaintext: 8位二进制明文字符串

key: 10位二进制密钥字符串

返回: 8位二进制密文字符串

异常:

ValueError: 输入长度不符合要求

decrypt(ciphertext: str, key: str) -> str

功能: 解密8位二进制数据

参数:

ciphertext: 8位二进制密文字符串

key: 10位二进制密钥字符串

返回: 8位二进制明文字符串

encrypt\_ascii(text: str, key: str) -> str

功能: 加密ASCII字符串

参数:

text: 普通文本字符串

key: 密钥(支持多种格式)

返回: 加密后的字符串(可能包含转义序列)

decrypt\_ascii(ciphertext: str, key: str) -> str

功能: 解密ASCII字符串

参数:

ciphertext: 加密后的字符串

key: 密钥

返回: 解密后的明文字符串

generate\_subkey(key\_array: np.ndarray) -> tuple

功能: 生成两个子密钥

参数: 10位密钥数组

返回: (subkey1, subkey2) 元组

F(R: np.ndarray, subkey: np.ndarray) -> np.ndarray

功能: F轮函数

参数:

R: 右半部分4位数据

subkey: 8位子密钥

返回: 4位输出

2.2 Ui\_SDESDialog类 (SDESDialog.py)

主要方法:

key\_verify() -> bool

功能: 验证密钥输入合法性

返回: 验证是否通过

key\_trans() -> str

功能: 将各种格式的密钥转换为二进制

返回: 10位二进制密钥字符串

encrypt()

功能: 执行加密操作并更新界面

decrypt()

功能: 执行解密操作并更新界面

2.3 SDESGIFDemo类 (sdes\_gif\_demo.py)

主要方法:

create\_progress\_animation(test\_case, output\_path) -> str

功能: 创建暴力破解进度动画

参数:

test\_case: 测试用例字典

output\_path: 输出GIF路径

返回: 找到的密钥

3. 数据格式规范

3.1 密钥格式

python

# 有效的密钥示例

binary\_key = "1010001110" # 10位二进制

decimal\_key = "654" # 0-1023

hex\_key = "28F" # 0x000-0x3FF

ascii\_key = "A" # 单个ASCII字符

3.2 文本编码转换

系统自动处理以下转换：

ASCII → 二进制

十进制 → 二进制

十六进制 → 二进制

二进制 → 目标编码

4. API使用示例

4.1 基本加解密

python

from SDES import SDES

sdes = SDES()

# 二进制加解密

cipher = sdes.encrypt("10101010", "1010000010")

plain = sdes.decrypt(cipher, "1010000010")

# ASCII加解密

encrypted\_text = sdes.encrypt\_ascii("Hello", "secret")

decrypted\_text = sdes.decrypt\_ascii(encrypted\_text, "secret")

4.2 暴力破解演示

python

from sdes\_gif\_demo import SDESGIFDemo

demo = SDESGIFDemo()

test\_case = {

'plaintext': '10101010',

'ciphertext': '11001100',

'true\_key': '1010000010'

}

found\_key = demo.create\_progress\_animation(test\_case, "demo.gif")

4.3 自定义算法测试

python

# 测试特定密钥的加密效果

sdes = SDES()

for i in range(1024):

key = format(i, '010b')

cipher = sdes.encrypt("00000000", key)

print(f"Key: {key} -> Cipher: {cipher}")

1. 测试和调试

见测试报告

6. 部署说明

6.1 打包为可执行文件

bash

pip install pyinstaller

pyinstaller --onefile --windowed main.py

6.2 依赖管理

PyQt5==5.15.9

numpy==1.21.0

matplotlib==3.5.0

pillow==9.0.0