S-DES加解密程序开发手册

# 简介

本开发手册提供了一个基于S-DES（Simplified Data Encryption Standard）算法的加解密程序的详细接口文档，用于对数据进行简单小规模加密和解密操作。通过S-DES算法对信息加密，实现保证信息安全，保障信息传输过程中的可靠性的功能。同时确保其在传输过程中不被未经授权的人员所访问。

在次开发手册中，将包含以下内容：

1. 组件及接口概述：介绍了加密解密程序的各个组件和对外提供的接口。
2. 使用示例：提供了一些示例代码，展示了如何使用加密解密程序完成加密和解密操作。
3. 注意事项：列出了在使用加密解密程序时需要注意的事项和建议。

通过阅读此开发手册，您将了解如何正确使用S-DES加密解密程序，以及如何保护您的数据安全。请确保遵守安全性原则，并妥善保管生成的密钥，一面数据被未授权的人员获取。

# 组件及接口概述

本程序包含以下组件及其对应的接口：

## generation\_key （密钥生成）

参数：

P\_10：密钥置换盒

P\_8：密钥压缩装置

Key：加密时需要使用的密钥

## encrypt & decrypt（加解密）

key1：第一轮加密时使用的密钥

key2：第二轮加密时使用的密钥

IP：初始置换盒

SBox1：第一次置换盒

SBox2：第二次置换盒

SPBox：最终置换盒

## GUI界面展示：

调用python中的tkinter库，设计并实现程序的UI界面。

# 使用示例（基于python）

## 3.1密钥生成

self.P\_10 = [3, 5, 2, 7, 4, 10, 1, 9, 8, 6]

self.P\_8 = [6, 3, 7, 4, 8, 5, 10, 9]

self.key\_start = key\_start

self.key = self.P\_10\_Box()

## 3.2 加解密示例

self.IP = [2, 6, 3, 1, 4, 8, 5, 7]

self.IP\_1 = [4, 1, 3, 5, 7, 2, 8, 6]

self.EPBox = [4, 1, 2, 3, 2, 3, 4, 1]

self.SBox1 = np.array([[1, 0, 3, 2], [3, 2, 1, 0], [0, 2, 1, 3], [3, 1, 0, 2]])

self.SBox2 = np.array([[0, 1, 2, 3], [2, 3, 1, 0], [3, 0, 1, 2], [2, 1, 0, 3]])

self.SPBox = [2, 4, 3, 1]

input = "abcd123"

data = spl(input)

cip = ""

for i in range(data.shape[0]):

    a = Enc.gen\_cip(data[i, :])

    cip = cip + chr(sum\_bin(a))

print(cip)

data1 = spl(cip)

pla = ""

for i in range(data1.shape[0]):

    b = Enc.sol\_cip(data1[i, :])

    pla = pla + chr(sum\_bin(b))

print(pla)

# 注意事项

在GUI界面中输入密钥的时候直接输入密钥数字，不需要按特定格式输入。但是需要保证输入的密钥位数为十。

请每一次通讯时妥善保管密钥，不要将密钥泄露给他人。

本程序仅适用于小规模的数据加密需求，不建议用于对大量或敏感数据进行加密。