**开发手册**

**1.概述**

本项目是基于Python 3和其内置Tkinter库实现的S-DES算法图形化工具。项目的设计遵循关注点分离(Separation of Concerns)原则，将用户界面(UI)与核心加密逻辑完全解耦，使得代码易于理解、维护和扩展。

**2.架构设计**

项目采用分层架构，主要分为以下三个层次：

**（1）表现层**

文件:gui.py,main.py

职责:构建和管理所有UI组件，处理用户输入事件（如点击按钮、输入文本），对输入数据进行初步的格式验证，并调用逻辑层来执行核心任务。对于耗时操作（暴力破解、算法分析），通过threading模块创建新线程来执行，避免UI阻塞，并通过回调函数更新进度。

**（2）逻辑层(Logic Layer)**

文件:sdes\_core.py,brute\_force.py,closed\_test.py

职责:实现项目的所有核心算法。sdes\_core.py包含无状态的、纯粹的S-DES加密、解密和密钥生成函数。brute\_force.py和closed\_test.py则封装了特定的业务逻辑，它们依赖于sdes\_core.py来完成原子操作。该层完全独立于UI。

**（3）适配/工具层(Adapter/Utility Layer)**

文件:ascii\_processor.py

职责:充当逻辑层和特定数据格式之间的桥梁。它将ASCII字符串适配成逻辑层能够处理的二进制分组数据，反之亦然。

**3.模块详解**

**main.py**

功能:程序的唯一入口。职责是初始化Tkinter主窗口和SDesGUI类实例，并启动事件循环。

**gui.py**

核心:SDesGUI类。

功能:

-初始化UI界面，包括窗口、标签页、输入框、按钮和文本显示区。

-为所有交互组件绑定事件监听器（例如command=self.basic\_encrypt）。

-从输入框获取用户输入，并进行格式校验（如检查位长、是否为二进制）。

-在接收到用户指令后，调用其他模块的函数执行实际操作。

-通过启动线程(threading.Thread)来处理耗时任务，并将结果和进度安全地更新回UI。

**sdes\_core.py**

功能:S-DES算法的纯粹数学实现。

关键常量:P10,P8,IP,IP\_inv,EP,SP,Sbox1,Sbox2等所有置换和替换表。

关键函数:

-permute(bits,table):通用的位置换函数。

-generate\_keys(key):从10位主密钥生成两个8位的子密钥k1和k2。

-f\_function(right\_half,subkey):实现S-DES的核心轮函数F。

-encrypt(plaintext,key)/decrypt(ciphertext,key):分别实现单个8位数据块的加密和解密流程。

**ascii\_processor.py**

功能:在字符串和二进制表示之间进行转换。

关键函数:

-encrypt\_ascii(text,key):遍历输入字符串，将每个字符转换为8位二进制，再调用sdes\_core.encrypt进行加密，最后将结果转回字符。

-decrypt\_ascii(text,key):执行与加密相反的解密过程。

**brute\_force.py**

功能:实现密钥暴力破解。

关键函数:brute\_force\_crack(...)

-使用itertools.product('01',repeat=10)高效生成所有1024个可能的密钥。

-根据传入的mode参数，选择调用sdes\_core.encrypt或ascii\_processor.encrypt\_ascii进行验证。

-通过progress\_callback回调函数向调用者（GUI）报告进度，实现解耦。

**closed\_test.py**

功能:实现算法封闭性测试。

关键函数:closed\_test(...)

-遍历所有1024个密钥，对同一个明文进行加密。

-使用一个字典(ciphertext\_to\_keys)来存储加密结果，其中key为密文，value为生成该密文的密钥列表。

-遍历结束后，筛选出字典中值列表长度大于1的项，即为密钥碰撞。

**4.扩展与维护**

**修改UI:**所有UI相关的代码都集中在gui.py中。修改组件、布局或样式不会影响后端的加密逻辑。

**替换算法:**若要将S-DES替换为其他分组密码算法，只需重写sdes\_core.py中的函数，只要保持函数签名（输入输出格式）不变，上层逻辑几乎无需改动。

**增加新功能:**可以遵循现有模式，例如：

（1）在逻辑层创建一个新模块（如new\_feature.py）。

（2）在gui.py中创建一个新的标签页。

（3）在新标签页的UI逻辑中调用new\_feature.py中的函数来完成功能。