**开发文档**

1. 简介

该组件包含S-DES加密算法加解密组件、暴力破解组件及用户界面组件。其中加解密组件结构类似，都是通过实例化加解密构造器并传入参数，之后调用公有函数返回所需的内。而暴力破解组件则是基于加密组件，通过蛮力法枚举所有可能的结果并检验是否符合要求。用户界面组件用于实现用户操作并显示结果。

1. 安装和配置

首先在命名空间中导入窗口类、EncryptCreator类、DecryptCreator类与ForcelyAttack类，其次在主函数将Window类实例化并调用窗口类的构造函数，接着在窗口的事件监听器中实例化完成对应操作所需要的类，并分别调用EncryptCreator类中的encrypt函数、DecryptCreator类中的decrypt函数、ForcelyAttack类中的run和find函数。在运行程序后，向文本框输入符合要求的内容并点击对应的执行按钮，即可在下方的主文本框中显示结果。

1. 接口列表

EncryptCreator类：参数为明文与密钥，无返回值，调用其encrypt函数后返回密文

DecryptCreator类：参数为密文和密钥，无返回值，调用其decrypt函数后返回明文

ForcelyAttack类： 无参数，无返回值，调用run函数后返回该明密文对被暴力破解后可能的密钥，调用find函数后返回通过程序运行以来所有输入的明密文对解得的结果。

Window类：无参数无返回值

1. 实例代码
2. EncryptCreator encryptCreator = new EncryptCreator(plaintext,key);

cipertext = encryptCreator.encrypt();

1. DecryptCreator drcryptCreator = new DecryptCreator(ciphertext,key);

Plaintext = decryptCreator.decrypt();

3 ForcelyAttack forcelyAttack = new ForcelyAttack();

possibleKeys = forcelyAttack.run(plaintext,ciphertext);

actuallyResults = forcelyAttack.find();

4. Window window = new Window();

win.setTitle(title);

win.setBounds(x,y,width,height);

1. 类说明

EncryptCreator类的构造函数中会调用Init函数，完成对各个置换盒参数的初始化，而encrypt函数中会按照加密步骤依次调用执行该操作所对应的函数，最终完成加密。

DecryptCreator类的构造函数中会调用Init函数，完成对各个置换盒参数的初始化，而decrypt函数中会按照解密步骤依次调用执行该操作所对应的函数，最终完成解密。

调用Window类的setTitle函数并将窗口名称作为参数传入，再调用setBounds函数并将窗口位置及大小作为参数传入