**用户指南：S-AES算法实现程序**

1. **简介**

S-AES是一种简化的高级加密标准（Advanced Encryption Standard，AES），用于数据的加密和解密。本程序旨在为用户提供S-AES算法的加密和解密功能。此外，本程序还实现了CBC（Cipher Block Chaining）模式加密和解密以及使用中间相遇攻击算法破解密钥。

1. **安装和运行**

2.1 系统要求

* 操作系统：程序支持运行在Windows、Mac和Linux操作系统上。
* Python版本：程序要求安装Python 3.7或更高版本。

2.2 安装步骤

1. GitHub链接：<https://github.com/StianTang/S-AES.git>，下载程序的代码文件并解压缩到您选择的目录中。
2. 打开命令行页面。
3. 使用cd命令导航到程序代码所在的目录。
4. 安装所需的依赖项，可以使用以下命令：pip install -r reqWebrements.txt

2.3 运行程序

1. 打开命令行页面。
2. 使用cd命令导航到程序代码所在的目录。
3. 运行Web页面的主程序，使用以下命令：python app.py
4. 程序的Web页面将在浏览器打开，您现在可以开始使用S-AES算法的加密和解密功能。
5. **加密和解密**

3.1 选择输入类型并输入文本和密钥

* 在Web页面的"Input\_type"下拉菜单中选择要输入的文本类型，可以选择"ASCII"或"Binary"类型。
* 在"Input"文本框中输入要加密或解密的文本。
* 在"Key"文本框中输入用于加密或解密的密钥。

注：程序将根据密钥长度16位、32位、48位分别选择单次、双重、三重加密解密。

3.2 选择加密或解密模式

* 在Web页面的"Mode"下拉菜单中选择要执行的操作，可以选择"Encrypt"进行加密或"Decrypt"进行解密。

3.3 执行加密或解密操作

* 单击Web页面的"Encrypt/Decrypt"按钮以执行加密或解密操作。

3.4 显示结果

* 加密或解密操作完成后，在Web页面的"Result"标签下将显示加密或解密后的结果。

1. **中间相遇攻击**

5.1 输入已知明文密文组

* 在导航栏切换到"Meet in the middle attack"页面。
* 在"明文"输入框中输入要执行中间相遇攻击算法的明文序列。
* 在"密文"输入框中输入要执行中间相遇攻击算法的密文序列。

5.2 执行中间相遇攻击算法

* 单击"Run"按钮以执行中间相遇攻击算法。

5.3 显示结果

* 中间相遇攻击算法完成后，在Web页面的"Result"标签下将显示可能密钥集结果。

1. **密码分组链CBC模式**

5.1 选择输入类型并输入文本、密钥、初始向量

* 在导航栏切换到" Cipher Block Chaining"页面。
* 在Web页面的"Input\_type"下拉菜单中选择要输入的文本类型，可以选择"ASCII"或"Binary"类型。
* 在"Input"输入框中输入要加密或解密的文本。
* 在"iv"输入框输入用于加密或解密的初始向量
* 在"key"输入框输入用于加密或解密的密钥

5.2 选择加密或解密模式

* 在Web页面的"Mode"下拉菜单中选择要执行的操作，可以选择"Encrypt"进行加密或"Decrypt"进行解密。

5.3 执行加密或解密操作

* 单击Web页面的"Encrypt/Decrypt"按钮以执行加密或解密操作。

5.4 显示结果

* 加密或解密操作完成后，在Web页面的"Result"标签下将显示加密或解密后的结果。

1. **注意事项**

6.1 输入格式要求

* 输入文本应为二进制或ASCII格式。
* 密钥应为16/32/48位二进制字符串。

6.2 程序限制和注意事项

* 由于S-AES算法是一种简化的算法，只提供了数据的保密性，并不能保证数据的完整性。
* 使用中间相遇攻击需要大量的计算和时间。
* 程序仅供学习和演示目的使用，不适用于真实的安全应用。

1. **安全功能**

* 本系统的目的是为了对信息进行加密，从而确保信息安全。其中，CBC模式是一种对称加密模式，通过使用前一个加密块与当前明文块进行异或操作，增加了加密的安全性。它还使用初始化向量（IV）来增加加密的随机性。