**S-AES算法实现文档**

**1. AES算法简介**

AES（Advanced Encryption Standard）是一种广泛使用的对称密钥加密标准，由美国国家标准与技术研究院（NIST）于2001年发布，用以取代旧的数据加密标准（DES）。AES加密算法以其高安全性和快速的加解密速度而闻名，在软件和硬件上都能高效运行，且实现相对简单，需要的存储空间较少。

**2. AES算法加密步骤**

**（一）密钥扩展（Key Expansion）**

将原始密钥扩展成多个轮密钥，每个轮密钥用于加密过程中的一轮。

**（二）初始轮（AddRoundKey）**

将明文与初始密钥进行按位异或操作。

**（三）多轮迭代**

每轮迭代包括以下四个步骤：

1. **SubBytes**：字节替换。
2. **ShiftRows**：行移位。
3. **MixColumns**：列混合。
4. **AddRoundKey**：添加轮密钥。

**（四）最后一轮**

不包括MixColumns操作，仅进行SubBytes、ShiftRows和AddRoundKey操作。

**3. S-AES过关测试**

**第1关：基本测试**

根据S-AES算法编写和调试程序，提供命令行界面支持用户交互。输入可以是16位的数据和16位的密钥，输出是16位的密文。

* **加密**：
  + 输入明文和密钥，执行加密操作，输出密文。
* **解密**：
  + 输入密文和密钥，执行解密操作，输出明文。

如下如所示：





**第2关：交叉测试**

所有人编写程序时需使用相同算法流程和转换单元，确保在异构系统或平台上能正常运行。

* **测试结果**:

（1）以明文为ABCD，密钥为0x2233为例，在我方输入后得到密文：



将密文送给对方解密后，成功得到明文ABCD：



(2):而若以“good morning”为输入，0x2233为密钥作为对方输入之后，得到密文äR÷näR¥éÉ



输入我方进行解密：



成功得到“good morning”

综上，该算法能够保证在异构系统或平台进行运行并保持正确性

**第3关：扩展功能**

支持ASCII编码字符串(任意长度)输入，输入分组为2 Bytes，输出为ASCII字符串（乱码）。

* **加密**：
  + 输入ASCII明文，通过S-AES加密，生成密文。
* **解密**：
  + 输入密文和密钥，获取明文。

如下图所示: 

**第4关：多重加密**

* **双重加密**：
  + 使用两把密钥对明文进行加密。



* **中间相遇攻击**：
  + 尝试破解出双重加密的两把密钥。

(例如，对于明文0x1234，密文¢¬=ìE则没有对应密钥)



* **三重加密**：
  + 使用三把密钥对明文进行加密。



**第5关：工作模式**

基于S-AES算法，使用CBC模式对较长的明文消息进行加密，注意初始向量的生成并需共享。

* **篡改测试**：
  + 尝试对密文进行替换或修改，然后解密并对比结果。

例如，对于加密：



进行CBC加密后的密文为²Ï9=~É-)BC\*£，可正常运行并解密。

**4. 用户手册**

**一、引言**

1. **文档目的**：帮助用户了解和使用基于Python开发的S-AES加密算法应用程序。
2. **适用范围**：适用于需要对数据进行加密和解密的用户，以及对密码学和网络安全感兴趣的技术爱好者。

**二、产品概述**

1. **功能简介**：
   * **二进制加密**：用户输入明文并获取随机密钥，加密后生成密文。
   * **ASCII加密**：支持ASCII明文的加密和解密。
   * **双重加密**：使用两把密钥进行加解密。
   * **三重加密**：使用三把密钥进行加解密。
   * **中间相遇攻击**：输入明文和密文，破解出双重加密的密钥。
   * **工作模式**：使用CBC模式对较长消息进行加密和解密。
2. **系统要求**：
   * Python 3.x。
3. **安装与配置**：
   * 无需安装，用户只需通过命令行运行Python程序即可使用。

**三、使用说明**

1. **界面介绍**：
   * 通过命令行输入明文、密钥，显示加密和解密结果。
2. **操作流程**：
   * **二进制加密**：
     + 输入明文和密钥，点击加密等按钮，输出密文。
     + 输入密文和密钥，点击解密等按钮，输出明文。

**5. 开发手册**

**一、引言**

1. **文档目的**：提供基于Python开发的S-AES加密算法应用程序的接口说明。
2. **适用范围**：适用于需要对S-AES算法进行集成或二次开发的开发人员。

**二、系统概述**

1. **架构说明**：
   * 主要由一系列函数组成，实现了S-AES加密算法的各个步骤，包括字节替换、行移位、列混合、密钥调度、加密、解密以及一些辅助函数。
2. **技术选择**：
   * 使用Python的内置函数和列表操作实现各种置换和转换操作。

**三、接口列表**

以下是S-AES算法实现中的所有接口及其详细描述（以加密为例，解密部分同加密）：

1. **Key\_plus(plaintext\_bits,key\_bits)**
   * **功能**：进行密钥加操作，对密钥和明文进行异或。
   * **返回值**：一个4位16进制字符串，表示结果。
2. **Half\_byte\_sub(value)**
   * **功能**：输入明文，进行半字节替换。
   * **返回值**：一个由ASCII字符组成的字符串，表示结果。
3. **Swap\_nibbles(value)**
   * **功能**：对输入数值进行行位移。
   * **返回值**：返回行位移后的结果。
4. **Mix\_columns(text)**
   * **功能**：进行行混淆操作。
   * **返回值**：行位移后的4位16进制字符串。
5. **Key\_expansion(key)**
   * **功能**：对给定的密钥进行扩展。
   * **返回值**：返回三把4位16进制密钥。
6. **encry(plaintext, key1)**
   * **功能**：最基础的加密函数，对给定的明文（16bit）进行加密。
   * **参数**：
     + plaintext：明文字符串（4位16进制）。
     + key：密钥（4位16进制）。
   * **返回值**：加密后的密文字符串（16bit）。
7. **Encry\_string(plaintext, key1)**
   * **功能**：对给定的明文字符串（任意长度）进行加密。
   * **参数**：
     + plaintext：明文字符串。
     + key1：密钥（4位16进制）。
   * **返回值**：加密后的密文字符串。
8. **Double\_encrypt(plaintext, key)**
   * **功能**：对给定的明文进行双重加密。
   * **参数**：
     + plaintext：明文字符（16bit）。
     + key：双重加密密钥（32bit）。
   * **返回值**：加密后的密文字符（16bit）。
9. **tertiary\_encrypt(plaintext, key)**
   * **功能**：对给定的明文进行三重加密。
   * **参数**：
     + plaintext：明文字符（16bit）。
     + key：三重加密密钥（48bit）。
   * **返回值**：加密后的密文字符（16bit）。
10. **CBC(plaintext, key，vector)**
    * **功能**：对给定的明文字符串（任意长度）进行CBC加密。
    * **参数**：
      + plaintext：明文字符串。
      + key：CBC加密密钥（16bit）
      + vector：CBC加密所使用的初始向量。
    * **返回值**：加密后的密文字符串。
11. **Mid\_attack(plaintext, ciphertext)**
    * **功能**：根据输入明文密文，找出对应的密钥。
    * **参数**：
      + plaintext：明文字符串
      + ciphertext：密文字符串
    * **返回值**：找到的全部密钥组合。

**6. 附录**

**1. 术语表**

* **S-AES**：简化的高级加密标准（Simplified Advanced Encryption Standard），是一种教学用的对称密钥加密算法，简化了AES的结构，以便于学习和理解。
* **AES**：高级加密标准（Advanced Encryption Standard），是一种对称密钥加密标准，广泛用于数据加密，采用分组加密方式。
* **明文（Plaintext）**：未经加密的原始数据或文本，在加密过程中会被转换为密文。
* **密文（Ciphertext）**：经过加密处理的文本，不能被直接理解，需通过解密才能还原为明文。
* **密钥（Key）**：用于加密和解密的秘密信息，只有授权用户应知晓。密钥的安全性直接影响加密系统的安全性。
* **初始向量（Initialization Vector, IV）**：在某些加密模式（如CBC模式）中使用的随机值，用于确保相同的明文在每次加密时产生不同的密文。
* **轮密钥（Round Key）**：在加密过程中从主密钥扩展生成的多个密钥，每个轮密钥用于不同的加密轮次。
* **字节替换（SubBytes）**：AES和S-AES中的非线性变换步骤，通过查找表替换数据中的每个字节。
* **行移位（ShiftRows）**：在AES和S-AES中对数据进行行级别的循环移位，增强数据的扩散性。
* **列混合（MixColumns）**：对数据的每一列进行线性变换，进一步增强数据的扩散性。
* **轮密钥加（AddRoundKey）**：将明文或中间状态与当前轮密钥进行按位异或操作的步骤。
* **中间相遇攻击（Meet-in-the-Middle Attack）**：一种密码攻击方式，通过在加密和解密过程中寻找中间状态，减少搜索空间，从而破解密钥。
* **双重加密（Double Encryption）**：使用两把密钥对数据进行加密的过程，通常提供比单次加密更高的安全性。
* **三重加密（Triple Encryption）**：使用三把密钥对数据进行加密的过程，进一步增强数据的安全性。

**2. 参考资料**

* **《密码编码学与网络安全—原理与实践（第8版）》**：该书全面介绍了密码学的基础知识，包括AES算法的原理及其实现细节。
* **Python官方文档**：提供Python语言的全面参考，包括内置函数、标准库的使用和示例，适合开发人员查阅。
* **《信息安全导论》**：一本关于信息安全基础知识的教材，适合初学者了解密码学和安全协议。
* **AES算法标准文档**：由美国国家标准与技术研究院（NIST）发布的AES算法标准，提供了AES算法的详细定义和实施指南。
* **Cryptography and Network Security by William Stallings**：一本关于密码学和网络安全的经典教材，深入讨论了对称加密、非对称加密和各种安全协议。