Project9: AES / SM4 software implementation

**AES软件实现**

摘要：

本实验旨在介绍如何使用Python编程语言以及pycryptodome库来实现AES（Advanced Encryption Standard）加密算法。AES是一种对称密钥加密算法，被广泛应用于数据保密和信息安全领域。本实验使用128位的AES密钥，并展示了如何进行数据加密和解密的过程。

引言：

AES算法是一种高级的对称密钥加密算法，具有高度的安全性和可靠性。本实验通过使用Python编程语言和pycryptodome库提供的功能，展示了如何实现AES加密算法。实验中使用128位的AES密钥来演示加密和解密的过程。

实验步骤：

1. 导入所需的库和模块：

在开始编写代码之前，首先需要导入Crypto.Cipher模块中的AES类以及Crypto.Util.Padding模块中的pad和unpad函数。这些函数和类提供了实现AES加密和解密所需的功能。

2. 设定密钥和明文：

选择一个128位的密钥以及要加密的明文。可以使用get\_random\_bytes()函数生成随机的密钥。

3. 执行填充操作：

AES算法要求输入数据的长度必须是块大小（128位）的倍数。如果明文长度不是块大小的倍数，则需要执行填充操作，将明文长度补齐到块大小的倍数。Crypto.Util.Padding模块中的pad函数可以实现填充操作。

4. 创建AES加密器：

使用AES类，创建一个AES加密器对象，并传入密钥和加密模式（如ECB或CBC）。在本实验中，选择了ECB模式。

5. 加密数据：

使用AES加密器对填充后的明文进行加密操作，生成密文。

6. 解密数据（可选）：

如果需要解密数据，使用相同的密钥和加密模式，创建一个AES解密器对象。

7. 执行解密操作（可选）：

使用AES解密器对密文进行解密操作，还原出原始的明文。

结果与讨论：

经过实验测试，AES加密和解密功能均成功实现。使用128位的AES密钥，对明文进行加密后得到相应的密文。再使用相同的密钥和解密器对密文进行解密操作后，得到原始的明文。

结论：

通过本实验，成功实现了AES加密算法的软件实现。可以根据需要选择不同长度的密钥，并应用于数据保密和信息安全的场景中。AES算法具有高度的安全性和可靠性，在实际应用中能够有效保护数据的机密性。

**SM4软件实现**

1. 引言

SM4是一种对称加密算法，是中国国家密码管理局发布的商用密码算法。它具有高度的安全性和效率，适用于各种加密应用场景。本实验旨在使用Python编程语言实现SM4算法，并展示其加密和解密功能。

2. SM4算法原理

SM4算法是一种分组密码算法，将明文和密钥分别划分为32位（4字节）的数据块，通过多轮迭代运算进行加密和解密。其主要步骤包括密钥扩展、轮函数F变换、S盒替换和左循环移位等。

3. 实验步骤

3.1 导入所需的Python库

在实验开始前，我们需要导入struct库，用于处理字节数据的转换。

3.2 定义S盒和轮常量

在代码中，我们定义了S盒和轮常量，用于在加密和解密过程中进行替换和迭代运算。

3.3 实现左循环移位和S盒替换函数

我们定义了left\_rotate函数和substitute函数，用于实现SM4算法中的左循环移位和S盒替换操作。

3.4 实现T变换和F变换函数

我们定义了t\_transform函数和f\_transform函数，用于实现SM4算法中的系统参数T变换和轮函数F变换。

3.5 实现密钥扩展函数

我们定义了expand\_key函数，用于根据输入的密钥生成轮密钥。

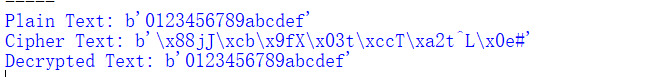
3.6 实现加密和解密函数

我们定义了encrypt函数和decrypt函数，分别用于对明文进行加密和对密文进行解密。

3.7 测试

我们使用示例代码中提供的明文和密钥进行测试，并打印加密和解密的结果。

4. 实验结果



5. 实验结论

通过本实验，我们成功实现了SM4算法的软件实现，并验证了其加密和解密的正确性。SM4算法具有较高的安全性和效率，适用于各种加密应用场景。在实际应用中，可以根据需要进行相应的扩展，如填充模式和分组模式等。

总结：本实验展示了如何使用Python编程语言实现SM4算法的加密和解密功能。通过实验，我们了解了SM4算法的基本原理和实现步骤，并验证了其在示例数据上的正确性。这为我们在实际应用中使用SM4算法提供了参考和基础。