SM4加密算法实现说明文档

1. 概述

本文档描述了Python实现的SM4分组密码算法的使用方法和实现细节。SM4是中国国家密码管理局于2012年发布的分组密码标准，也称为GB/T 32907-2016。

2. 功能特性

完整的SM4算法实现（128位分组，128位密钥）

支持多种工作模式：

ECB（电子密码本模式）

CBC（密码分组链接模式）

CTR（计数器模式）

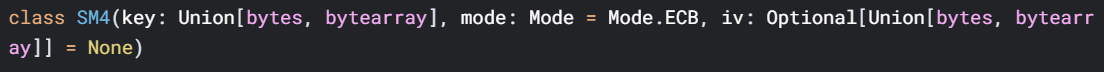
自动PKCS#7填充处理

完善的错误检查和处理

优化的算法实现

4. API参考

4.1 SM4类

参数

key: 16字节的加密密钥

mode: 工作模式（Mode.ECB、Mode.CBC或Mode.CTR）

iv: 初始化向量（CBC和CTR模式需要16字节）

方法

encrypt(plaintext, padding=True)

加密数据

plaintext: 要加密的原始数据（bytes或bytearray）

padding: 是否自动添加PKCS#7填充（默认为True）

返回: 加密后的数据（bytes）

decrypt(ciphertext, padding=True)

解密数据

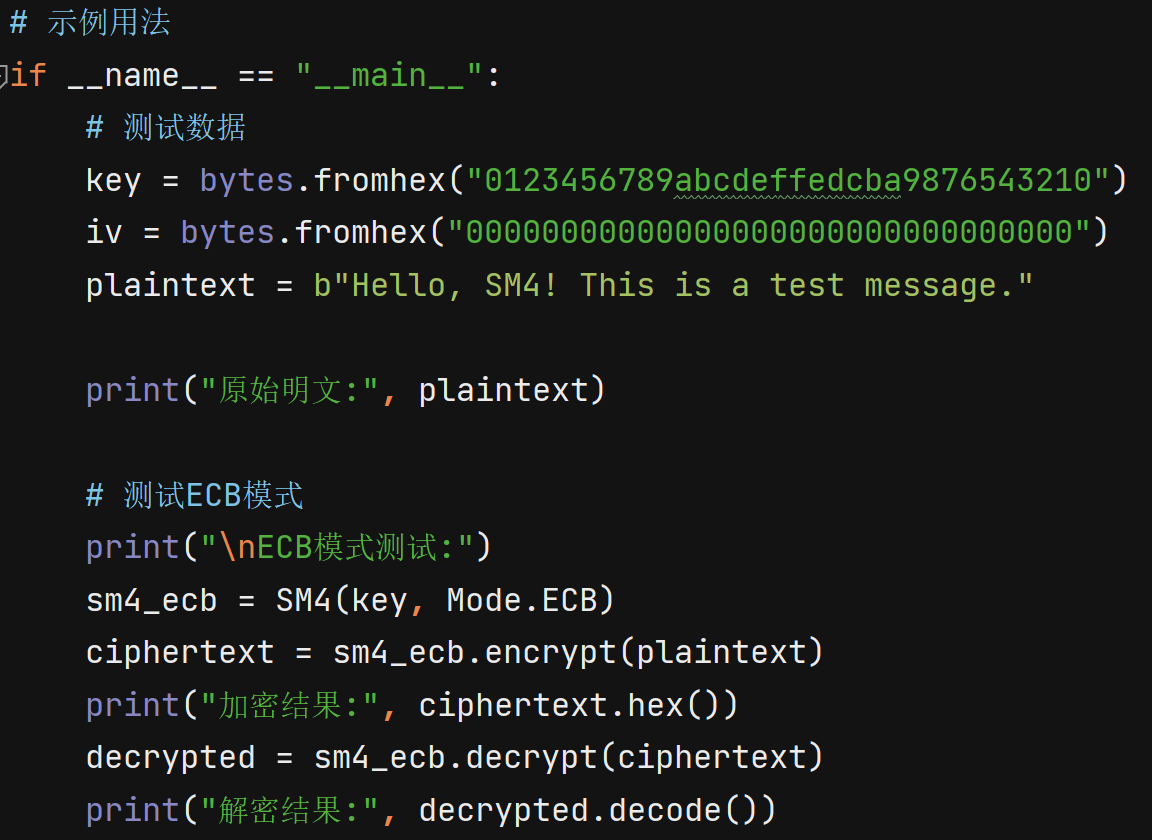
ciphertext: 要解密的密文（bytes或bytearray）

padding: 是否自动去除PKCS#7填充（默认为True）

返回: 解密后的原始数据（bytes）

5. 使用示例

5.1 基本使用



6. 实现细节

6.1 算法参数

分组大小：16字节（128位）

密钥长度：16字节（128位）

轮数：32轮

6.2 核心算法步骤

密钥扩展：

将128位密钥扩展为32个轮密钥

使用系统参数FK和固定参数CK

加密/解密流程：

32轮非线性迭代变换

每轮使用轮函数F

解密使用相同的轮密钥逆序

轮函数F：

非线性变换τ（使用S盒）

线性变换L

6.3 工作模式

ECB模式：

最简单的分组加密模式

相同明文块加密为相同密文块

不适合加密重复模式的数据

CBC模式：

每个明文块先与前一个密文块异或

需要初始化向量(IV)

更安全，适合大多数应用

CTR模式：

将计数器加密后与明文异或

不需要填充

可以并行加密

6.4 填充方案

使用PKCS#7填充：

加密时添加填充使数据长度为块大小的整数倍

解密后自动去除填充

填充字节的值等于填充长度

7. 安全注意事项

密钥管理：

密钥应安全存储和传输

定期更换密钥

初始化向量(IV)：

CBC和CTR模式需要IV

IV应该是随机且不可预测的

不要重复使用相同的IV和密钥组合

工作模式选择：

推荐使用CBC或CTR模式

ECB模式不适合加密大量或有重复模式的数据

错误处理：

实现会检查密钥和数据长度

解密时会验证填充有效性

8. 性能考虑

预计算：

密钥扩展在初始化时完成

S盒使用预计算表加速查找

优化：

使用位操作代替数学运算

减少不必要的内存分配

限制：

纯Python实现性能不如C扩展

大数据加密可能较慢

