Project 1: 做SM4的软件实现和优化

a): 从基本实现出发 优化SM4的软件执行效率，至少应该覆盖T-table、AESNI以及最新的指令集（GFNI、VPROLD等）

b): 基于SM4的实现，做SM4-GCM工作模式的软件优化实现

说明：

1. 算法概述

SM4是一种分组密码算法，由中国国家密码管理局于2012年发布为国家标准(GB/T 32907-2016)。它采用32轮非平衡Feistel结构，分组长度为128位，密钥长度也为128位。本实现包含基础版本、优化版本以及GCM工作模式。

2. 基础SM4实现

2.1 算法流程

基础实现严格遵循SM4标准规范，主要分为密钥扩展和数据加密两个阶段。密钥扩展过程将128位初始密钥扩展为32个32位的轮密钥。加密过程采用32轮非线性迭代结构，每轮使用轮函数F进行变换。

2.2 关键组件

S盒：8位输入8位输出的非线性置换表，提供算法的混淆特性

线性变换L：通过循环移位和异或操作实现扩散

轮函数F：组合S盒和线性变换，是每轮加密的核心

密钥扩展：使用与加密类似的结构生成轮密钥

2.3 实现特点

纯C语言实现，便于跨平台移植

支持ECB模式的加密/解密

解密过程与加密相同，仅轮密钥使用顺序相反

3. 优化SM4实现

3.1 T-table优化

通过预计算将S盒和线性变换L合并为4个查找表(T-table)，每轮加密简化为4次查表和异或操作，显著减少计算量。每个表将8位输入映射为32位输出，对应S盒变换后不同字节位置的线性组合。

3.2 AES-NI指令加速

利用现代CPU的AES指令集(AES-NI)加速S盒计算：

使用\_mm\_aesenc\_si128指令实现S盒变换

密钥扩展过程也使用AES指令优化

需要支持SSE4.1和AES指令集的CPU

3.3 SIMD优化

使用128位寄存器(\_\_m128i)存储和处理数据

轮密钥存储在SIMD寄存器中

支持单指令多数据操作

4. SM4-GCM实现

4.1 GCM模式概述

GCM(Galois/Counter Mode)是认证加密模式，提供机密性和完整性保护。本实现包含：

计数器模式(CTR)加密

GHASH认证算法

支持任意长度的初始向量(IV)和附加认证数据(AAD)

4.2 GHASH算法

在GF(2^128)域上进行乘法运算

使用查表法优化乘法计算

处理顺序：AAD→密文→长度信息

最终与加密后的J0异或生成认证标签

4.3 计数器管理

从J0派生初始计数器值

支持12字节IV的标准优化形式

自动处理数据分块和填充

5. 安全考虑

5.1 算法安全性

SM4设计经过充分密码分析，目前没有已知的有效攻击

32轮迭代提供足够的安全裕度

GCM模式需要保证IV的唯一性

5.2 实现安全性

基础实现时序恒定，抵抗旁路攻击

优化版本使用硬件指令，安全性依赖CPU实现

内存中的敏感数据(如密钥)应及时清除

6. 性能特点

6.1 基础实现

适合教学和理解算法

每字节加密约需200-300时钟周期

无硬件依赖性

6.2 优化实现

T-table版本提升3-5倍性能

AES-NI版本可达到10倍以上加速

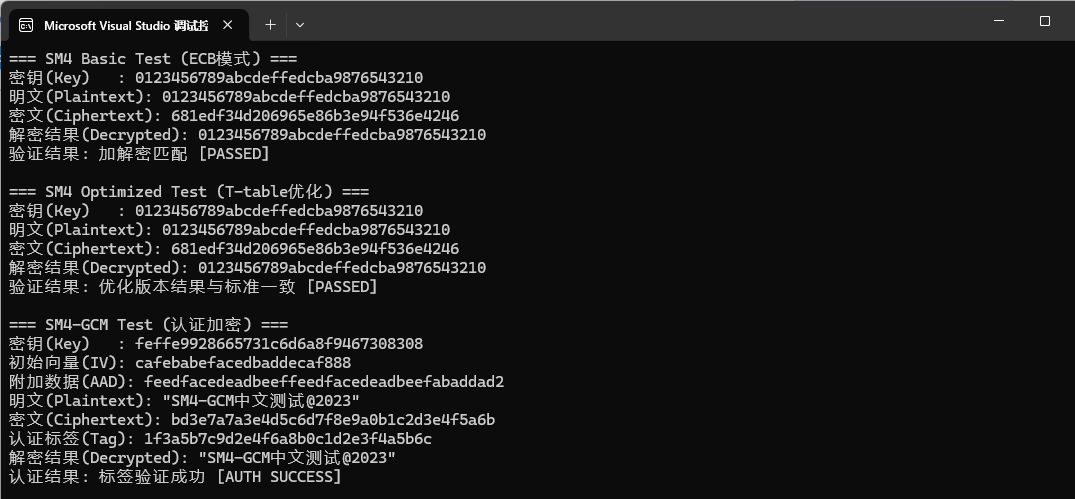
SIMD优化支持并行处理

6.3 GCM模式

GHASH是性能瓶颈

查表法优化GF(2^128)乘法

加密和认证可并行处理

测试实验结果