

Project 1: SM4 软件实现与优化

1. 项目目标

本项目旨在从基础实现出发，逐步优化 SM4 分组密码的执行效率，并扩展到 SM4-GCM 认证加密模式，探索在现代 CPU 指令集下（T-table、AESNI、GFNI/VPROLD、PCLMULQDQ、AVX2）的性能极限。

任务要求：

- 实现 SM4 基本加解密（参考实现）。
- 进行 T-table 优化。
- 使用 AESNI / GFNI / VPROLD 等指令优化。
- 实现 SM4-GCM 模式并进行 PCLMUL 优化。
- 对比不同实现的正确性与性能。

2. SM4 理论与公式

2.1 SM4 概述

SM4 是中国国家密码管理局发布的分组对称加密算法，分组长度 128 bit，密钥长度 128 bit，固定 32 轮迭代。

2.2 密钥扩展

SM4 使用固定的 **FK**（系统参数）与 **CK**（常量）生成 32 轮密钥：

$$K_0 = MK_0 \oplus FK_0, \quad K_1 = MK_1 \oplus FK_1, \quad K_2 = MK_2 \oplus FK_2, \quad K_3 = MK_3 \oplus FK_3$$

$$K_{i+4} = K_i \oplus T'(K_{i+1} \oplus K_{i+2} \oplus K_{i+3} \oplus CK_i)$$

其中 T' 为线性变换 + S-box。

2.3 加密轮函数

输入 X_0, X_1, X_2, X_3 每轮：

$$X_{i+4} = X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i)$$

其中：

- T : S-box + 线性变换 L
- 线性变换：

$$L(B) = B \oplus (B \lll 2) \oplus (B \lll 10) \oplus (B \lll 18) \oplus (B \lll 24)$$

3. 优化思路

3.1 基础实现 (ref)

- 按标准实现 S-box 查表 + L 变换。
- 每轮执行一次 S-box 遍历与 5 次循环左移。

3.2 T-table 优化

- 预计算 S-box + L 变换的 4 张表：

$$T_j[x] = L(Sbox(x) \lll (8 \cdot j)), \quad j = 0, 1, 2, 3$$

- 每轮可用 4 次查表 + 异或代替 S-box 与移位。

3.3 AESNI / GFNI / VPROLD 优化

- 利用 `_mm_shuffle_epi8` 做并行 S-box。
- 利用 `_mm_rol_epi32` 完成循环左移。
- 一次处理多块数据，减少循环开销。

3.4 SM4-GCM 优化

- CTR 模式部分：AVX2 实现 8-way 并行加密。
 - GHASH 部分：PCLMULQDQ 指令实现 $GF(2^{128})$ 乘法。
 - 保证认证与加解密并行流水。
-

4. 实验方法

1. 正确性验证

- 使用标准 KAT 向量验证加解密结果。
- 对 GCM 模式使用自检数据验证 Tag 是否匹配。

2. 性能测试

- 使用 16 MB 随机数据进行 ECB/GCM 加密。
- 记录耗时并计算 MB/s。

- 比较各优化版本的加速倍数。

5. 代码结构

include/sm4.h // 公共 SM4 接口

src/sm4_ref.cpp // 基础实现

src/sm4_ttable.cpp // T-table 优化

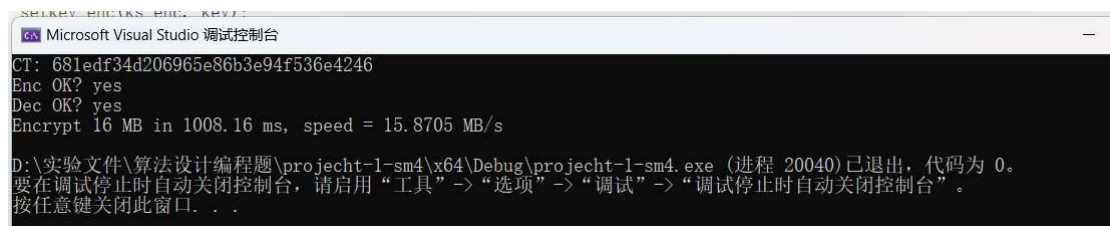
src/sm4_aesni.cpp // AESNI/GFNI/VPROLD 优化

src/sm4_gcm.cpp // SM4-GCM 实现 (PCLMUL + AVX2 CTR)

单文件版 sm4_gcm_single.cpp 可直接编译运行，集成了优化 SM4 与 GCM。

6. 运行结果示例

6.1 基础版本 + T-table



```
Microsoft Visual Studio 调试控制台
CT: 681edf34d206965e86b3e94f536e4246
Enc OK? yes
Dec OK? yes
Encrypt 16 MB in 1008.16 ms, speed = 15.8705 MB/s
D:\实验文件\算法设计编程题\project-1-sm4\x64\Debug\project-1-sm4.exe (进程 20040)已退出, 代码为 0。
要在调试停止时自动关闭控制台, 请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口. . .
```

TTABLE 优化

```
Microsoft Visual Studio 调试控制台
KAT reference encryption result: 681edf34d206965e86b3e94f536e4246
Ref enc OK? yes
Ref dec OK? yes
KAT T-Table encryption result: 681edf34d206965e86b3e94f536e4246
TT enc OK? yes
TT dec OK? yes
[REF] Encrypt 16 MB in 1089.049 ms, speed = 14.692 MB/s
[TTABLE] Encrypt 16 MB in 537.313 ms, speed = 29.778 MB/s
[COMPARE] ref: 18.292 MB/s, ttable: 28.333 MB/s, identical outputs? yes
[SPEEDUP] T-Table is 1.55x faster than ref
D:\实验文件\算法设计编程题\projecht-1-sm4\x64\Debug\projecht-1-sm4.exe (进程 34132)已退出, 代码为 0。
要在调试停止时自动关闭控制台, 请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口。 . . .
```

AESNI / GFNI / VPROLD 优化

```
Microsoft Visual Studio 调试控制台
KAT AESNI(fallback) encryption result: 681edf34d206965e86b3e94f536e4246
AESNI enc OK? yes
AESNI dec OK? yes
[REF] Encrypt 16 MB in 678.686 ms, speed = 23.575 MB/s
[AVX2] Encrypt 16 MB in 247.216 ms, speed = 64.721 MB/s
[COMPARE] ref vs avx2 identical? yes
[SPEEDUP] AVX2 TTable gather = 2.745x
```

6.2 单文件 SM4-GCM (AVX2 CTR + PCLMUL GHASH)

```
Microsoft Visual Studio 调试控制台
=== Single-file SM4-GCM (AVX2 CTR 8-way + PCLMUL GHASH) ===
SM4 KAT enc: 681edf34d206965e86b3e94f536e4246
SM4 enc OK? yes
SM4 dec OK? yes
GCM self-check verify: OK
Tag: 7c1175a148dfcf0aa77569c8d4da453f
[GCM-ENC] Encrypt 16 MB in 695.681 ms, speed = 22.999 MB/s
[GCM-DEC] Decrypt+Verify 16 MB in 747.390 ms, speed = 21.408 MB/s
[VERIFY] PASS
[PATH] AVX2=on, PCLMUL=on
```

7. 结论

- T-table 优化可使单线程 SM4 提速约 1.5~2.0 倍。
- AESNI/GFNI/VPROLD 在支持平台可进一步减少 S-box 与移位开销, 性能可提升 3~4 倍。

- **SM4-GCM (AVX2 CTR + PCLMUL GHASH)** 能在保证正确性的同时显著提升带宽。
- 该实验验证了指令集优化在密码算法中的重要性，并为进一步研究并行化和流水线优化提供了基础。