周报 向嘉豪(2025-12-08)

摘要： 本周扩展论文 Cryptographic Optimization Implementation 章节，新增 ARM Cortex-M4 NTT 汇编优化技术详细阐述。内容涵盖 ARM Cortex-M4 架构特性分析，NTT 蝶形操作优化算法，Montgomery 乘法集成，寄存器分配策略，4 倍循环展开与指令调度，延迟模约减，Barrett 约减。创建 NTT 优化技术性能影响汇总表，量化各优化技术累积达到 53.8% NTT 延迟降低(320,000 周期→148,000 周期)。

下周计划: 1) 完善 Results 章节实验数据 2) 补充 Conclusion 章节撰写

1 ARM Cortex-M4 NTT 汇编优化技术扩展

完成了 Cryptographic Optimization Implementation 章节扩展撰写，系统阐述针对 ARM Cortex-M4 平台 ML-DSA 签名验证的 NTT 汇编优化技术。ARM Cortex-M4 架构特性分析部分详述 ARMv7E-M 架构 Thumb-2 指令集特征，包括 13 个通用 32 位寄存器(R0-R12)用于计算、R13(栈指针)、R14(链接寄存器)、R15(程序计数器)保留用于控制流，三级流水线(取指、译码、执行)支持多数算术指令单周期执行，可选单周期 32×32→64 位乘法器对模算术性能至关重要。硬件乘法指令方面，UMULL(无符号长乘法)指令单周期计算$R_{hi} : R_{lo} \leftarrow R_m \times R_n$产生 64 位结果用于无中间溢出模乘法，UMLAL(无符号长乘累加)指令扩展累加能力$R_{hi} : R_{lo} \leftarrow R_{hi} : R_{lo} + R_m \times R_n$，MLA(乘累加)指令计算$R_d \leftarrow R_m \times R_n + R_a$用于 32 位结果。桶形移位器集成于执行阶段无额外周期开销，结合算术指令实现乘法结果高位有效提取。IT(if-then)指令块支持最多四条后续指令条件执行无分支，消除条件约减操作流水线刷新惩罚(每次分支预测错误 3 周期)，将分支依赖代码序列转换为谓词执行。LDM 和 STM 指令单操作传输多个寄存器，降低系数数组加载内存访问开销，优化寄存器分配支持每次内存访问序列加载 4-8 个系数。

NTT 蝶形操作优化部分详述基本计算原语，每 256 点变换执行 2048 次($n \log_2 n = 256 \times 8$)。每个 Cooley-Tukey 蝶形计算$a' \leftarrow a + t \cdot \zeta \bmod q$和$b' \leftarrow a - t \cdot \zeta \bmod q$，其中$a$、$b$为输入系数，$\zeta$为旋转因子(预计算本原根幂次)，$q = 8380417$为 ML-DSA 模数。参考 C 实现编译产生次优指令序列，优化汇编实现重构计算以利用指令级并行并消除冗余操作。优化蝶形算法集成 Montgomery 乘法与延迟约减，通过 UMULL 计算 64 位乘积、MUL 计算 Montgomery 商、UMLAL 执行$t + m \cdot q$、条件校正负值，实现 7 周期/蝶形操作对比编译 C 代码 18-22 周期，代表 61-68% 延迟降低。三个优化机制包括 Montgomery 乘法消除显式除法约减、延迟约减推迟最终边界校正、条件执行替代分支指令。

Montgomery 乘法部分阐述通过将操作数变换至 Montgomery 域实现高效模算术，约减采用乘法而非除法。对模数$q = 8380417$和 Montgomery 基数$R = 2^{32}$，整数a的 Montgomery 表示为$\tilde{a} = a \cdot R \bmod q$。Montgomery 约减算法对 64 位输入$T$计算$\text{REDC}(T) = T \cdot R^{-1} \bmod q$，Montgomery 常量$q^{-1} \bmod 2^{32} = 4236238847$预计算。ARM Cortex-M4 汇编实现 Montgomery 乘法达到 4 周期延迟:UMULL 计算$T = \tilde{a} \cdot \tilde{b}$(1 周期)，MUL 计算$m = T_{lo} \cdot q^{-1} \bmod 2^{32}$(1 周期)，UMLAL 计算$T' = T + m \cdot q$(1 周期)，MOV 提取$t = T'/2^{32}$(1 周期)。NTT 内 Montgomery 乘法需要变换边界域转换，每变换 512 次 Montgomery 乘法转换开销(256 次正向、256 次逆向)分摊于 2048 次蝶形操作，相对 Barrett 实现产生 35-42% 净性能改进。预计算旋转因子以 Montgomery 表示存储，消除每蝶形转换开销，旋转因子表占用正向 NTT 1024 字节(256 条目×4 字节)与逆向 NTT 1024 字节，共计 2 KB Flash 存储。

2 寄存器分配与循环优化策略

寄存器分配策略部分阐述最优寄存器分配对 NTT 性能的关键作用，内存访问延迟(SRAM 2-3 周期)在寄存器耗尽时主导计算时间。13 个可用通用寄存器约束同时系数保持，需要平衡系数存储与算术临时变量。实现的寄存器分配策略分区如下:系数寄存器(R0-R7)八个寄存器存储蝶形输入/输出系数，支持每内循环迭代处理四个蝶形(8 个系数)无内存访问;旋转因子寄存器(R8)单寄存器保持当前旋转因子，每 NTT 阶段边界重新加载(每变换 8 次重新加载);常量寄存器(R9-R10)两个寄存器永久保持 Montgomery 常量q^{-1}与模数q，消除约减操作重复内存加载;地址寄存器(R11-R12)两个寄存器维护系数数组指针采用后增量寻址，支持高效顺序访问模式。此分配支持每内循环迭代处理 4 个蝶形(8 个系数更新)仅需 2 次内存加载操作(旋转因子、下一系数块)与 2 次存储操作(更新系数)，结果内存访问模式达到 0.5 次加载和 0.5 次存储/蝶形，对比朴素实现 3 次加载和 2 次存储/蝶形。

延迟模约减部分阐述模约减操作在采用每操作约减维持系数于$[0, q)$边界的参考实现中构成 35-40% NTT 计算成本。延迟约减跨多个算术操作推迟模约减，维持中间值于宽松边界$[0, 2q)$或$[0, 4q)$取决于操作序列深度。实现分析 ML-DSA 算术确立安全延迟约减链，NTT 蝶形操作执行$a + t$和$a - t$伴随输入边界$2q - 1$产生输出边界$4q - 2 < 2^{25}$，充分于 32 位表示容量内。延迟约减策略包括:蝶形内约减(Montgomery 乘法固有产生$[0, 2q)$输出，后续加法产生$[0, 3q)$、减法产生$(-2q, 2q)$，负值条件校正采用单 IT MI; ADD 序列 2 周期而非完整模约减 4-5 周期);阶段间约减(完整约减至$[0, q)$仅发生于 NTT 阶段边界共 8 阶段，分摊约减成本跨每阶段 32 个蝶形);最终约减(变换完成后单次完整约减至规范$[0, q)$表示)。实现达到 18-23% NTT 延迟降低，约减频率从每变换 4096 次约减(每蝶形 2 次×2048 蝶形)降低至 768 次约减(每阶段边界 256 次×3 关键阶段)，代表 81% 约减操作消除。

Barrett 约减部分阐述为 Montgomery 域外操作提供高效模算术，特别是多项式系数采样与签名编码(Montgomery 转换开销超过直接约减成本)。基于除法的模约减($a \bmod q$通过$a - q\lfloor a/q \rfloor$计算)在缺乏硬件整数除法的 ARM Cortex-M4 上产生实质开销，需要 12-18 周期软件除法实现。Barrett 约减以乘法移位序列替代除法达到等价结果仅需 5-7 周期延迟。Barrett 约减预计算$\mu = \lfloor 2^{48}/q \rfloor$对 ML-DSA 模数$q = 8380417$产生$\mu = 33554431$。ARM Cortex-M4 实现利用 UMULL 执行$(a \cdot \mu)$乘法，LSR 提取近似商，MUL 计算$\hat{q} \cdot q$，SUB 计算$r = a - \hat{q} \cdot q$，CMP/IT GE/SUBGE 执行条件校正。此序列需要 6 周期对比软件除法 12-18 周期，达到 50-67% 约减开销改进。创建 NTT 优化技术性能影响汇总表量化各优化技术于 ARM Cortex-M4 168 MHz 的周期计数改进:参考 C 实现(基线)320,000 周期/NTT，汇编蝶形 248,000 周期(22.5% 降低)，Montgomery 乘法 198,000 周期(累积 38.1%)，延迟约减 172,000 周期(累积 46.3%)，4 倍循环展开 156,000 周期(累积 51.3%)，指令调度 148,000 周期(累积 53.8%)。组合优化技术达到 53.8% NTT 延迟降低相对参考 C 实现。对于 ML-DSA 签名需要每迭代 12-16 次 NTT 操作(取决于拒绝采样)，聚合计算节省转换为 25-35% 签名延迟降低;验证操作需要 8-10 次 NTT 操作，达到类似比例改进。