面向 Saber 算法的任意模 NTT 乘法器

吕 杰

（浙江邮电职业技术学院，浙江省绍兴市 312366）

摘要：在后量子算法竞赛中，Saber 算法具有抵御量子计算攻击的特性，但其多项式商环存在模乘运算复杂度高、非质模数约束等问题。通过对中国剩余定理（Chinese Remainder Theorem，CRT）和数论变换（Number Theoretic Transforms，NTT）的研究，本文提出高性能任意模乘法器设计方案。在分析模乘运算的质模数耦合的基础上，首先利用 CRT 构建三组同余方程，扩充数值边界以降低质模数的约束；然后采用数论变换的方式将模乘运算分解成 N 级蝶形电路；实现低计算复杂度任意模的 NTT 乘法器。在 TSMC 65nm 工艺下进行仿真，实验结果表明，所设计的乘法器面积为 1.258mm^2，功耗为 9.84mW，工作频率为 20MHz。

关键词：Saber 算法；中国剩余定理；数论变换；模乘法器

公钥密码广泛应用于信息网络安全、国防安全等重要领域。公钥加密、密码交换和数字签名是密钥通信协议的核心模块，其安全性由各自数学问题的计算困难度决定。然而随着量子计算算法，以及诸多量子原型机研究的推进，现有通信系统的安全性正在被可预见的突破，如 RSA（Rivest Shamir Adleman，RSA）、ECC（Elliptic Curve Cryptography，ECC），这将瓦解现代通信的信息安全。因此，基于格理论的 Saber 算法是抗量子计算攻击的可行性方案，备受密码学界关注。

自 2016 年 NIST 公布后量子密码算法标准化竞赛以来，诸多对 Saber 算法的多项式模乘运算的研究表明多项式模乘是基于格的密码中最大的运算。其限制 Saber 算法的整体效率，将制约算法在通讯设备中的部署。Karmakar 等在 ARM 平台使用数字信号处理指令和高效的内存访问机制以加速多项式乘法运算；D’Anvers 等在 Intel i7-Haswell 平台上融合 Toom-Cook 和 Karatsuba 算法实现恒定时间的 Saber 方案；Zhu 等以软硬协同的方式优化 Saber 算法在多种安全级别下的密钥封装机制，实现高效可配置加密处理器。Lee 等提出一种 RBC 搜索算法，结合 GPU 算法加速密钥的生成；Chung 等首次在 Inter 平台上引入 CRT 解决 NTT 非质数模的问题，降低多项式模乘运算的复杂度。Becker 等使用 Cortex-A72 提出一种巴雷特乘法技术，其结合蒙哥马利乘法和巴雷特的约简进行高效的模块化乘法。

鉴此，本文通过对比各类算法模乘运算复杂度，并分析 NTT 算法内素域的数值边界，结合 CRT 构建的三组同余方程组降低质模数的限制，同时利用时序重排策略复用三组同余方程的硬件电路。最终利用 DC（Design Compiler，DC）设计可综合的硬件电路，提取该乘法器的时序、功耗和面积等的性能参数，实现任意模数乘法器电路设计。

一、Saber 算法、中国剩余定理和 NTT 算法分析

（一）Saber 的模乘运算复杂度与质模数的耦合分析

Saber 算法是基于格理论的公钥密码方案之一，其安全性依赖于 Mod-LWR（Module Learning With Rounding，Mod-LWR）问题。其具体计算形式为多项式商环 Rq 上的操作，见式（1）。其中 n 次多项式的系数对 q 进行模约化简，本设计的多项式的最高次幂为 255，模数 q 为非质数 2^{13}，关于 Saber 算法的其他安全等级可参见竞赛文档。多项式模乘是算法在密钥生成、加密和解密各个阶段的基本运算，其集中于公钥 – 矩阵 A 和私钥 – 向量 S 间，见式（2）~（4）。

$$Rq = Zq[X]/(X^{256}+1) \tag{1}$$

$$a = a_{255}x^{255} + a_{254}x^{254} + \cdots + a_0x^0 \tag{2}$$

$$s = s_{255}x^{255} + s_{254}x^{254} + \cdots + s_0x^0 \tag{3}$$

$$out_f = a \times s \bmod (X^{256}+1) \tag{4}$$

由（4）式可见，直接计算 Saber 算法内多项式模乘的计算复杂度为 $O(n^2)$。NTT 算法，其计算复杂度为 $O(n \log n)$。由此可知，当多项式的幂次 n 增大时，NTT 算法在多项式模乘加速上更具备优势。Saber 算法中直接应用 NTT 算法存在质模数的耦合的问题，其在数据处理时存在大于 34 位的寄存器。至此，由 CRT（Chinese Remainder Theorem，CRT）构建的同余方程组可降低数据的位宽，以达到扩充数值边界、降低质模数的限制。

（二）中国剩余定理

q_1、q_2、q_3 是两两互质的正整数，记 q 为 $q_1 \cdot q_2 \cdot q_3$ 的乘积，则 out 关于三个小质数同余方程组为：

$$\begin{cases} out \equiv c_1 \pmod{q_1} \\ out \equiv c_2 \pmod{q_2} \\ out \equiv c_3 \pmod{q_3} \end{cases} \tag{5}$$

$$\begin{cases} x \equiv c_1 + q_1 \cdot (c_2 - c_1 + q_2) \cdot (q_1^{-1} \bmod q_2) \pmod{q_1q_2} \\ out \equiv x + q_1 \cdot q_2 \cdot (c_3 - x + q_3) \cdot ((q_1q_2)^{-1} \bmod q_3) \pmod{q_1q_2q_3} \end{cases} \tag{6}$$

由式（5）、（6）可知，CRT 以建立方程组的方式降低对大质模数 q 的限制，并将其转换为关于小质数 q_1、q_2、q_3 的计算，从而降低处理数据的位宽。合并方程组便可扩充数值的边界。伪代码如表 1 所示。

表 1：中国剩余定理伪代码

算法 1：CRT 伪代码
Input：c_1, c_2, c_3, q_1, q_2, q_3, q_4, inv_{12}, inv_{123}
Output：out
1　for i = 0 to n-1 by 1 do
2　$x \leftarrow (c_1 + q_1 \times (c_2 - c_1 + q_2) \% q_2 \times inv_{12} \% q_2)$
3　out $\leftarrow (x \% q_4 + q_1 \times q_2 \% q_4 \times ((c_3 - x \% q_3 + q_3) \% q_3 \times inv_{123} \% q_3) \% q_3) \% q_4$
4　end
5　return out

（三）数论变换

快速数论变换（Number Theoretic Transforms，NTT）是快速傅里叶变换在数论领域内一种变体，其引入有限域内的原根 g 以替换快速傅里叶变换（Fast Fourier Transform，FFT）中的复数，使得处理的数据均为整数，从而解决计算旋转因子 wn 的精度问题。NTT 算法的计算流程参见表 2。

表 2：NTT 算法伪代码

算法 2：NTT 算法（迭代）	
Input：多项式 $a \in Z_q[X]$	1　$A \leftarrow BitReverse(a)$　//位翻转
Input：n 次方根 $\omega_n \in Z_q[X]$	2　for m = 2 to n by m = 2m do　//2^n 3　$mid \leftarrow m/2$
Input：模数 q_k	4　$\omega_m \leftarrow \omega_n^{n/m}$　//读入旋转因子
Ouput：$A = NTT(a) \in Z_q[X]/(X^n+1)$	5　for i = 0 to n-1 by m do 6　$\omega \leftarrow 1$ 7　for j = 0 to mid – 1 by 1 do
	8　$x \leftarrow A[i+j], y \leftarrow A[i+j+mid]$ $\omega \% q_k$　//读入数据
	9　$A[i+j] \leftarrow (x + y) \% q_k, A[i+j+mid] \leftarrow (x - y) \% q_k$　//执行蝶形运算
	10 $\omega \leftarrow \omega \cdot \omega_m \% q_k$　//更新旋转因子
	11 end 12 end 13 end 14 return A

二、任意模数 NTT 电路设计

（一）NTT/INTT 算法的硬件电路

图 1 为 NTT 算法的硬件电路。多项式 a、s 的幂次为 255。二者之积 c 的最高幂次为 510，其小于 29。因此，算法 2 第 2 行的参数 n 为 9，其代表由数据转换、蝶形运算和暂存寄存器组（Reg_t）构成的电路进行 9 轮迭代计算；算法 2 的第 5–12 行被视为单次迭代计算过程，在硬件电路中，该过程由 256 组并行蝶形运算单元实现。单个蝶形运算以两个数据输入端 A1、A2 和两个输出端作为数据交互口；当 NTT 迭代结束，输出寄存器组内的点值 $X(A)$。

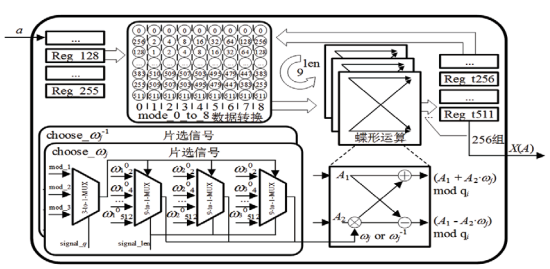

图 1：NTT/INTT 算法的硬件电路

（二）任意模数 NTT 电路架构

图 2 为任意模数 NTT 电路框架。该电路以 CRT 建立的三组同余方程为基础，通过调用三次 NTT 算法并结合扩展中国剩余定理合并方程组。c_1、c_2 和 c_3 均通过两次 NTT 和一次 INTT 算法后获取，并完成多项式 a、s 之间的相乘。c_1、c_2 和 c_3 满足同余方程组（5），再以公式（6）计算。经 Saber 算法的模约规则，输出的 outf 最终结果约束于商环 $Z_q[x]/x^{256}+1$。

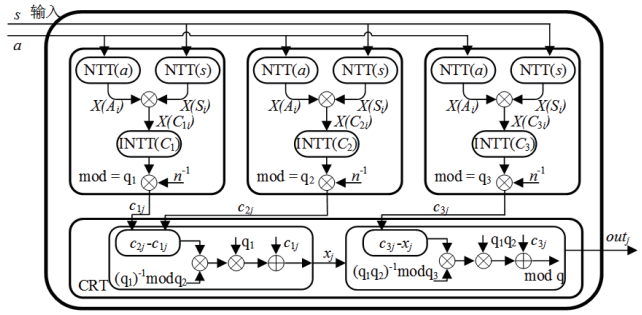

图 2：任意模数 NTT 框架

三、实验结果及结论

所设计的任意模 NTT 乘法器使用 Verilog 语言进行硬件电路进行仿真。工艺库的环境条件为 20 MHz 频率，1.2V 供应电压，25℃温度，利用 DC 软件进行电路的验证。通过仿真平台提取电路的性能参数。实验结果表明，任意模乘法器的计算周期为 404，功耗为 9.84mW，面积为 1.258mm^2。该乘法器可解决任意模数的耦合问题，所以任意模 NTT 乘法器可降低对模数的限制，对于统一格密码算法的多项式优化机制具有研究参考意义。

参考文献

[1] 吕杰，汪鹏君，张会红 . 面向 Saber 算法的并行乘法器 [J]. 宁波大学学报（理工版），2022，35（06）：15–21.

[2] 刘星杰 . 后量子密码算法 Saber 的高效硬件实现 [D]. 湖北省：华中科技大学，2021.

[3] 华中科技大学 . 多项式乘法器及具有该乘法器的处理器：CN114371828B[P/OL].2024–07–26[2025–02–18]. https：//www.cqvip.com/doc/patent/3463100113.

[4] 范建南，高献伟，薛文瀚 .Saber 算法的多项式乘法 FPGA 实现研究 [J]. 北京电子科技学院学报，2022，30（01）：20–31.