● ISSN 0254-4164
● CODEN JIXUDT

计算机学报

CHINESE JOURNAL OF COMPUTERS

第45卷 Vol.45 第6期 No.6

2022.6

● 中国计算机学会 中国科学院计算技术研究所 主办
● 科学出版社 出版

计 算 机 学 报

(JISUANJI XUEBAO)

第 45 卷 第 6 期 2022 年 6 月

目 次

人工智能与信息安全

CHINESE JOURNAL OF COMPUTERS

Vol.45 No.6 June 2022

CONTENTS

环上舍入学习和模上舍入学习的通用实现算法与参数选取方法

姜子铭[1),2)] 周永彬[1),2),3)] 张 锐[1),2)]

[1)](中国科学院信息工程研究所 北京 100093)

[2)](中国科学院大学网络空间安全学院 北京 100049)

[3)](南京理工大学网络空间安全学院 南京 210094)

摘 要 格密码领域中的环上舍入学习(RLWR)和模上舍入学习(MLWR)问题是构造后量子密码原语的一类重要数学工具,已广泛应用于伪随机函数、陷门函数等基础密码构造.RLWR和MLWR实现通常包括三项基础操作:多项式乘法、模约化和舍入计算,三项基础操作均有多种适用于不同参数的实现方案,相同运行平台、相同安全等级下RLWR和MLWR软件实现效率受参数影响显著.然而,现有RLWR和MLWR方案实现及效率优化工作大多仅针对某些特定参数,无法处理任意参数;此外,现有RLWR和MLWR方案参数选取大多只考虑了部分基础操作的效率,缺乏系统的快速实现参数选取方法.为解决上述问题,本文提出了通用高效的RLWR实现算法和MLWR实现算法,以及RLWR和MLWR快速实现参数选取方法.本文首先给出了NTT和NTT负折叠卷积的使用条件与方案参数以及CPU字长之间的量化关系,扩展了可快速实现的基于多项式环的密码方案参数空间;其次,提出了一种适用于RLWR和MLWR实现的新舍入算法,与通用的传统舍入实现相比,新舍入算法的效率在64位Intel i7平台下提高了11%左右;最后,提出了通用RLWR实现算法和MLWR实现算法,通用实现算法根据方案参数和CPU字长灵活选取高效的基础操作实现方案,将其应用于Saber方案实现,在64位Intel i7平台下未使用编译优化指令的Saber密钥封装效率提升了52%左右,此外,对比分析了不同参数下RLWR和MLWR的效率,提出了RLWR和MLWR快速实现参数选取方法,为RLWR和MLWR方案设计与实现中的参数选取提供了指导.

关键词 格密码;舍入学习;多项式乘法;数论变换;舍入计算

中图法分类号 TP309 **DOI号** 10.11897/SP.J.1016.2022.01326

The General Implementation Algorithms and the Parameters Selection Methods of Ring Learning with Rounding and Module Learning with Rounding

JIANG Zi-Ming[1),2)] ZHOU Yong-Bin[1),2),3)] ZHANG Rui[1),2)]

[1)](*Institute of Information Engineering (IIE), Chinese Academy of Sciences (CAS), Beijing* 100093)

[2)](*School of Cyber Security, University of Chinese Academy of Sciences (UCAS), Beijing* 100049)

[3)](*School of Cyber Security, Nanjing University of Science and Technology (NJUST), Nanjing* 210094)

Abstract The learning with rounding (LWR) problem in lattice cryptography is one of the fundamental tools for constructing post-quantum cryptographic primitives. As a deterministic variant of learning with errors (LWE), LWR replaces random errors in LWE with deterministic rounding. The removal of random error sampling makes LWR-based cryptosystems more efficient than LWE. Since it was proposed, LWR problem has been extensively applied to the construction of basic cryptosystems, such as low-depth pseudorandom functions, lossy trapdoor functions and public-key encryption schemes. In order to reduce the computational complexity and bandwidth,

收稿日期:2021-06-03;在线发布日期:2021-11-18.本课题得到国家自然科学基金(61632020,U1936209,62002353)、北京市自然科学基金(4192067)资助.**姜子铭**,博士研究生,主要研究方向为格密码.E-mail:jiangziming@iie.ac.cn.**周永彬**(通信作者),博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为网络与信息安全理论及技术.E-mail:zhouyongbin@iie.ac.cn.**张 锐**,博士,研究员,博士生导师,中国计算机学会(CCF)会员,主要研究领域为信息安全、密码学、密码工程学.

the ring and module version of LWR, that is, ring LWR (RLWR) and module LWR (MLWR), are mostly used in practice. To make the post-quantum cryptosystems more practical, it is of great significance to improve the efficiency of RLWR and MLWR. The ring in lattice usually takes the polynomial ring, so RLWR/MLWR includes three basic operations: polynomial multiplication, modular reduction, and rounding. For each of the three operations, there are a variety of implementation methods with different efficiencies which apply to different parameters. The efficiency of RLWR/MLWR with different parameters may vary greatly at the same security level. However, most of the existing implementations of RLWR and MLWR are only applicable to some certain schemes with specific set of parameters. In addition, most of the existing RLWR and MLWR schemes have not given consideration to the implementation efficiency of all basic operations. There is a lack of RLWR/MLWR parameter selection method for the purpose of achieving high efficiency. In order to solve the above problems, we propose the general and efficient implementation algorithms of RLWR and MLWR, as well as the parameters selection methods of RLWR and MLWR to achieve high efficiency. In this work, we first discuss the conditions of existing implementation methods of the three basic operations on CPU platform and focus on the special parameters commonly used in lattice-based cryptosystems. In particular, we clarify the conditions of NTT-based multiplication and negative wrapped convolution, so we can also use NTT to accelerate polynomial multiplication when the modulus of RLWR/MLWR does not meet the requirements of NTT modulus but meets the conditions in this work. As a result, the parameter space of high-efficient cryptosystems based on polynomial ring structures would be expanded. Secondly, we present a new rounding algorithm that uses shift, addition and subtraction operations based on precomputation instead of the division operation. The efficiency of the new rounding algorithm is improved by about 11% compared with the universal traditional rounding algorithm which precomputes the reciprocal of the RLWR/MLWR modulus. Although the efficiency of the new rounding algorithm is about 2.5% lower than the most efficient rounding method based on addition and shift operations, the new rounding algorithm may be applicable when the addition-then-shift operation cannot be used, especially when the modulus of RLWR/MLWR is NTT modulus. Finally, we propose general implementation algorithms of RLWR and MLWR. Their core idea is to select efficient implementation schemes of the three basic operations according to the parameters and the CPU word length. When it is applied to Saber scheme, the efficiency of Saber key encapsulation is improved about 52% on 64-bit Intel i7 platform without compiler optimization instruction. We then compare the efficiency of RLWR/MLWR with different parameters and propose parameters selection methods of RLWR and MLWR. It aims to provide readers with reference for parameter selection in the design and implementation of RLWR/MLWR-based cryptosystems.

Keywords　lattice-based cryptography; learning with rounding; polynomial multiplication; number theory transformation; rounding

1　引　言

一旦通用量子计算机问世，现有广泛使用的基于大数分解和离散对数等问题的传统公钥密码体制将被攻破[1]，发展能抵抗量子攻击的后量子密码体制是当前的研究热点，也是迫切的现实需求. 美国国家标准与技术研究院(National Institute of Standards and Technology，NIST)和中国密码学会先后举办了后量子密码算法征集活动. 目前公开的后量子公钥密码主要包括：基于格的密码体制、基于编码的密码体制、基于多变量的密码体制和基于哈希的密码

体制，其中，基于格的密码体制凭借其抵抗已知量子攻击、可并行实现以及存在“最坏情形”到“平均情形”的底层困难问题安全性归约等优势，被认为是最有前景的后量子密码候选之一.

舍入学习(Learning with Rounding，LWR)作为格密码中应用最广泛的确定误差底层函数，是后量子密码原语的重要构造工具，可直接用于构造伪随机函数和陷门函数等[2-3]. 为构造基于格的伪随机函数，Banerjee 等人[2]于 2012 年提出 LWR 问题，并给出超多项式级别模数下带错学习(Learning with Errors，LWE)到 LWR 的归约. 随后，Alwen 等人[3]借助有损采样完成了多项式级别模数下的归约，但模数 p 和 q 仍需满足 $Q\geq 2nmEp$ 且 Q^2 不整除 q，其中 Q 是 q 的最大素因子，n 为向量维数，m 为采样个数，E 为 LWE 的误差边界. Bogdanov 等人[4]进一步改进了归约方法，解除了多项式级别下对模数的其它限制条件，模数只需满足 $q\geq 2mEp$.

LWR 可视为去随机化的 LWE，由于不再需要随机误差采样，LWR 的实现效率更高、抗侧信道攻击能力更强，已广泛应用于低深度伪随机函数[5]、有损陷门函数[3]、确定性公钥加密[6-9]以及全同态加密[10]等基础密码构造. 实际应用中需考虑计算复杂度和带宽等效率指标，现有实用的格密码方案大多采用环结构或模结构，LWR 的环版本和模版本分别称为环上舍入学习(Ring LWR，RLWR)和模上舍入学习(Module LWR，MLWR). NIST 后量子密码标准算法征集活动中的 Lizard[7]提案和 Round5[8]提案提供了基于 RLWR 的公钥加密方案，Saber[9]方案是基于 MLWR 的公钥加密方案，特别地，Saber 方案入围了公钥加密的决赛. 考虑到后量子密码体制实用化的现实需求，提升 RLWR 和 MLWR 实现效率具有重要意义.

RLWR 分布 $(a,b=\lfloor a\cdot s\rceil_p)$ 由参数 N,f,p,q 和环元素 s 决定，其中 N,p,q 是正整数，f 是单变元 N 次首一多项式，$a,b,s\in R_q=\mathbb{Z}_q[x]/(f)$. 考虑到 s 通常取自 0-1 分布等特殊分布，方案实现中 s 系数的上界可能远小于 q，本文假设多项式 s 系数绝对值的上界为 $\frac{1}{2}S$. RLWR 分布中 $b(x)$ 的计算分为三步：

① 整数域上的多项式乘法，$b_1(x)=a(x)\cdot s(x)$；

② 模约化，$b_2(x)=((b_1(x) \bmod f(x)) \bmod q)$；

③ 舍入计算，$b(x)=\left\lfloor \frac{p}{q}\cdot b_2(x)\right\rceil$.

MLWR 分布 $(\boldsymbol{a},\boldsymbol{b}=\lfloor(\boldsymbol{a}^{\mathrm{T}}\boldsymbol{s})\rceil_p)$ 由参数 N,f,p,q,l 和向量 $\boldsymbol{s}$ 决定，其中 l 是正整数，其他参数与 RLWR 相同，$\boldsymbol{a}\in R_q^{l\times l}$，$\boldsymbol{b},\boldsymbol{s}\in R_q^l$，本文假设方案实现中向量 $\boldsymbol{s}$ 所有元素的系数绝对值的上界为 $\frac{1}{2}S$. MLWR 分布中向量 $\boldsymbol{b}$ 的计算同样分为三步，与 RLWR 的区别在于，步骤①中计算的是矩阵乘法，等价于 $l\times l$ 次整数域上的多项式乘法和 $l\times(l-1)$ 次整数域上的多项式加法. 现有 RLWR 和 MLWR 实现通常包括三项基础操作：多项式乘法、模约化和舍入计算，三者均有多种实现方案，RLWR 以及 MLWR 的实现即为三项基础操作实现方案的组合.

多项式乘法是格密码领域最常用的基础运算之一，在基于 NTRU、RLWE 和 RLWR 等假设的密码体制中，计算两个多项式的乘积是一个基本操作，往往也是耗时最高的操作. 常用多项式乘法快速实现算法包括 Karatsuba[11]、Toom-Cook[12-13]、快速傅里叶变换[14](Fast Fourier Transform，FFT)、数论变换[15](Number Theoretic Transform，NTT)以及稀疏乘法算法，格密码领域多项式乘法的实现算法往往取决于方案参数[16]：模数 q 是 NTT 模数时通常采用 NTT 乘法算法，例如 Kyber[17]、Dilithium[18]和 qTESLA[19]方案；对于模数不是 NTT 模数的方案，次数界 N 较小时通常采用 Karatsuba 或 Toom-Cook 算法，例如 Saber 方案，N 较大时通常采用 FFT 乘法算法，例如 LIMA[20]方案；此外，s 的系数取自集合 $\{-1,0,1\}$ 时通常采用稀疏乘法算法，例如 Round5 和 Lizard 方案. 上述多项式乘法实现算法效率差异显著，其中 NTT 乘法算法的理论复杂度最低且实际效率相对较高. 虽然很多格密码方案实现采用了 NTT 乘法算法，但这些高效的 NTT 实现大多针对特定参数，无法处理任意参数.

此外，现有格密码方案大多认为，方案模数 q 是 NTT 模数时才可使用 NTT 加速多项式乘法[9]. 然而，NTT 乘法算法理论上没有限制条件，实用中仅对 NTT 模数的大小有要求，这是因为计算机能直接处理的整数存在上下界，NTT 模数过大时，引入大整数运算反而会降低效率. 事实上，即使 q 不满足 NTT 模数的一般性要求，若整数域上乘积多项式的系数较小，即参数 N,q,S 较小，使得存在 NTT 模数 M 满足 $qSN<2M$，也可以采用 NTT 乘法算法. 例如采用 NTT 乘法的 Saber 实现[21]，其密钥封装效率在 Cortex-M4 平台与原始实现相比提高了 22%左右. 目前，格密码领域对 NTT 实际使用条件的研究工作较少，现有计算平台下 NTT 的限制条

件与格密码参数之间的关系尚待研究.

格密码领域通常涉及两种模约化操作:模多项式和模整数.格密码方案通常取$R_q=\mathbb{Z}_q[x]/(x^N+1)$,模多项式采用减法实现.模整数运算主要的计算开销是除法,常用通用模整数快速实现算法包括Montgomery算法[22]和Barrett算法[23],二者都是在预计算的基础上,利用加减、乘法和移位代替除法.现有格密码方案模数q通常取特殊值,采用特殊模整数实现方法:q为2的方幂时采用按位与操作,例如Round5、Saber和Lizard方案;q是NTT模数时,采用模数为q的NTT乘法算法,可省略步骤②模约化中的模q运算,NTT算法内部模约化通常采用Montgomery算法,例如Kyber、Dilithium和qTESLA方案.现有计算平台下,Barrett算法和Montgomery算法是否是格密码领域效率最高的通用模整数算法尚待研究.

舍入计算同样是格密码领域的基础运算,除计算LWR分布外,还应用于压缩与解压缩、编码与解码等模块.与模整数类似,舍入计算主要的计算开销是除法.目前,RLWR/MLWR参数q和p通常取2的方幂,显然,若模数比值q/p是2的方幂,则可使用先加法再移位操作计算舍入值,例如Round5、Saber和Lizard方案.模数比值q/p非2的方幂的方案较少,通用舍入计算优化方法尚待研究.

综上所述,RLWR以及MLWR的实现即为三项基础操作实现方案的组合,三者均有多种实现方案,效率较高的实现方案都只适用于特殊参数:NTT乘法实现大多要求模数q为NTT模数(通常取奇素数),采用按位与实现模整数、先加法再移位实现舍入计算均要求模数q的因子包含2的方幂.故参数q不能同时满足上述限制条件.为提升模约化、舍入计算和随机数采样的效率,RLWR和MLWR的参数q和p通常取2的方幂,而现有NTT实现大多无法处理这种模数.目前,RLWR以及MLWR实现主要存在以下两个问题:

(1)现有RLWR和MLWR实现难以兼顾通用性与高效性.虽然已有许多RLWR和MLWR实现及优化工作[7-9],但这些实现方法大多针对特定密码方案,只适用于某些特殊参数,无法处理任意参数.

(2)缺乏系统的RLWR和MLWR快速实现参数选取方法.多项式乘法是三项基础操作中耗时最高的,但现有RLWR和MLWR方案选取参数时大多未考虑多项式乘法的效率,缺乏系统性研究分析.

针对以上问题,本文提出通用的RLWR实现算法和MLWR实现算法,以及RLWR和MLWR参数选取方法,具体贡献如下:

(1)扩展现有可采用NTT类乘法算法的格密码方案参数空间.本文给出通用CPU平台下,NTT和NTT负折叠卷积的限制条件与格密码方案参数N,f,q,S和CPU字长B之间的量化关系.当参数满足本文给出的限制条件时,即使模数q不满足现有NTT类乘法实现的一般性要求,也可以采用NTT类乘法算法.除RLWR/MLWR实现外,该成果可广泛应用其它于基于多项式环的密码方案实现,例如,基于Ring-LWE和NTRU等假设的密码方案.

(2)提出一种适用于RLWR和MLWR实现的新舍入算法.当q是奇素数且q可作为NTT模数时,舍入计算无法采用最高效的先加法再移位操作.针对这类q,本文提出一种基于预计算的新舍入算法,在64位通用CPU平台,其效率与传统通用舍入实现相比提高11%左右,虽然该算法的效率略低于先加法再移位,但二者的效率相差不超过2.5%.新舍入算法适合搭配NTT乘法算法使用,此外还可以应用于压缩与解压缩、编码与解码等其他密码学常用模块的实现.

(3)提出通用RLWR实现算法和MLWR实现算法.本文根据方案参数和CPU字长,灵活选择三项基础操作实现方案的优化组合方式,在保证通用性的情况下达到较高的软件实现效率.将该成果应用于NIST后量子密码算法征集中的Saber方案,与Saber第三轮原始实现相比,未使用编译优化指令时Saber密钥封装的效率可提升52%左右.

(4)进行实验对比分析,提出RLWR和MLWR快速实现参数选取方法.本文完成了相关算法的C语言实现,在64位Intel(R) Core(TM) i7-6700 CPU@3.40 GHz, ubuntu16.04操作系统上使用GCC编译器测试了各算法的实际运行时间.根据实际效率,本文提出RLWR和MLWR快速实现参数选取方法.在参数选择合理的情况下,512次多项式a和s系数均取自均匀分布的RLWR的运行时间最快为45 μs左右,基本满足实用预期.

2　预备知识

2.1　符号说明

$\mathbb{Z}$、$\mathbb{N}$:整数集合、自然数集合;

$\mathbb{Z}_+$:正整数集合;

$\mathbb{Z}_q$：对 $\forall q\in\mathbb{Z}_+$，$\mathbb{Z}_q$表示$\mathbb{Z}$模q的剩余类，本文默认取绝对最小剩余系$\left\{-\left\lfloor\frac{q-1}{2}\right\rfloor,\cdots,0,\cdots,\left\lfloor\frac{q}{2}\right\rfloor\right\}$；

$\mathbb{Z}[x]$：记x是一个变元，$\mathbb{Z}[x]$表示所有整系数多项式构成的集合；

$\mathbb{Z}_q[x]$：记x是一个变元，对$\forall q\in\mathbb{Z}_+$，$\mathbb{Z}_q[x]$表示所有系数属于$\mathbb{Z}_q$的多项式构成的集合；

$R=\mathbb{Z}[x]/(f)$：对$\forall f\in\mathbb{Z}[x]$，$(f)$表示由$f$生成的理想，$\mathbb{Z}[x]/(f)$表示$\mathbb{Z}[x]$模$(f)$剩余类环，记作$R$；

$R_q=\mathbb{Z}_q[x]/(f)$：对$\forall q\in\mathbb{Z}_+$，$\mathbb{Z}_q[x]/(f)$表示$\mathbb{Z}_q[x]$模(f)剩余类环，记作R_q；

a，A：白体字母表示集合中的元素；

$\boldsymbol{a}$，$\boldsymbol{A}$：黑体字母表示向量或矩阵；

$\boldsymbol{a}^{\mathrm{T}}$：向量或矩阵的转置；

log：如不做特殊说明，本文中对数默认以2为底；

$\lfloor x\rfloor$、$\lceil x\rceil$、$\lfloor x\rceil$：向下取整、向上取整、舍入取整.

$\gcd(a,b)$：a和b最大公因子.

2.2 LWR、RLWR和MLWR

对任意整数$q\geq p\geq 2$，舍入函数$\lfloor\cdot\rceil_p:\mathbb{Z}_q\to\mathbb{Z}_p$定义为$\lfloor x\rceil_p=\left\lfloor\left(\frac{p}{q}\right)\cdot x\right\rceil \bmod p$.

定义1. LWR分布. 假设$N,p,q,m\in\mathbb{Z}_+$，其中，$q\geq p\geq 2$. 给定向量$\boldsymbol{s}\in\mathbb{Z}_q^{N\times 1}$，LWR分布$L_{\mathrm{LWR}}$定义为$\mathbb{Z}_q^{N\times m}\times\mathbb{Z}_p^m$上的分布$(\boldsymbol{a},\boldsymbol{b})$，

$$L_{\mathrm{LWR}}=\{(\boldsymbol{a},\boldsymbol{b})\mid\boldsymbol{a}\leftarrow\mathbb{Z}_q^{N\times m},\boldsymbol{b}=\lfloor(\boldsymbol{a}^{\mathrm{T}}\boldsymbol{s})\rceil_p\in\mathbb{Z}_p^m\},$$

其中$\boldsymbol{a}\leftarrow\mathbb{Z}_q^{N\times m}$，表示矩阵$\boldsymbol{a}$的每一个元素都取自$\mathbb{Z}_q$上的均匀随机分布.

定义2. RLWR分布. 假设$N,p,q\in\mathbb{Z}_+$，其中，$q\geq p\geq 2$；$f(x)=\sum_{i=0}^{N}f_ix^i$是$N$次多项式；环$R_q=\mathbb{Z}_q[x]/(f)$. 给定环元素$s\in R_q$，RLWR分布$L_{\mathrm{RLWR}}$定义为$R_q\times R_p$上的分布$(a,b)$，

$$L_{\mathrm{RLWR}}=\{(a,b)\mid a\leftarrow R_q,b=\lfloor a\cdot s\rceil_p\in R_p\},$$

其中$a\leftarrow R_q$，表示环元素a取自环R_q上的均匀随机分布，多项式运算在环R_q中进行.

定义3. MLWR分布. 假设$N,p,q,l,m\in\mathbb{Z}_+$，其中，$q\geq p\geq 2$；$f(x)=\sum_{i=0}^{N}f_ix^i$是$N$次多项式；环$R_q=\mathbb{Z}_q[x]/(f)$. 给定环元素$s\in R_q^{l\times 1}$，MLWR分布$L_{\mathrm{MLWR}}$定义为$R_q^{l\times m}\times R_p^m$上的分布$(\boldsymbol{a},\boldsymbol{b})$，

$$L_{\mathrm{MLWR}}=\{(\boldsymbol{a},\boldsymbol{b})\mid\boldsymbol{a}\leftarrow R_q^{l\times m},\boldsymbol{b}=\lfloor(\boldsymbol{a}^{\mathrm{T}}\boldsymbol{s})\rceil_p\in R_p^m\},$$

其中$\boldsymbol{a}\leftarrow R_q^{l\times m}$表示矩阵$\boldsymbol{a}$的每一个元素都取自环$R_q$上的均匀随机分布，多项式运算在环$R_q$中进行.

RLWR分布由参数N,f,p,q和环元素s决定. 格密码中多项式环通常取$R_q=\mathbb{Z}_q[x]/(x^N+1)$. a取自R_q上的均匀随机分布，记$a=\sum_{i=0}^{N-1}a_ix^i$，其系数$|a_i|\leq\frac{1}{2}q$. s除均匀分布外，还可能服从0-1分布、高斯分布等特殊分布，s系数实际的上界可能远小于q，本文假设方案实现中s系数绝对值的上界为$\frac{1}{2}\mathrm{S}$，记$s=\sum_{i=0}^{N-1}s_ix^i$，其系数$|s_i|\leq\frac{1}{2}\mathrm{S}$，其中正整数$\mathrm{S}\leq q$；此外，若$s$系数等于0的概率较大，则可采用某些特殊多项式乘法算法，本文用参数$\mathrm{P}[s_i\neq 0]$表示s的系数不为0的概率. 给定s的分布后即可确定参数S和$\mathrm{P}[s_i\neq 0]$的值. RLWR的实现方案取决于参数N,f,p,q,S和$\mathrm{P}[s_i\neq 0]$.

MLWR方案大多取$m=l$，此时MLWR分布由参数N,f,p,q,l和向量$\boldsymbol{s}$决定. MLWR参数$l=m=1$时即为RLWR，参数N,f,p,q含义与RLWR相同. 向量$\boldsymbol{s}$中的元素均取自环R_q上的某一分布，给定该分布后即可确定参数S和$\mathrm{P}[s_i\neq 0]$的值. MLWR的实现方案取决于参数N,f,p,q,l,S和$\mathrm{P}[s_i\neq 0]$.

2.3 FFT和NTT

多项式有两种表示方法：系数表达和点值表达. 假设$a(x)$的次数是k，任意一个大于k的整数都是该多项式的次数界. 对于次数界为N的多项式，系数表达即$a(x)=\sum_{j=0}^{N-1}a_jx^j$，记$\boldsymbol{a}=(a_0,\cdots,a_{N-1})$为系数向量；点值表达是由$N$个点值对组成的集合$\{(x_i,y_i=a(x_i))\}_{0<i<N}$. 若$c(x)=a(x)\cdot b(x)$，那么$c(x)$对应的点值表达就是$\{(x_i,a(x_i)\cdot b(x_i))\}_{0<i<2N}$. 系数表达的多项式乘法经典算法复杂度为$O(N^2)$，而点值表达的多项式乘法复杂度为$O(N)$. 通过FFT或NTT对多项式的表示方法进行转换，可将系数表达的多项式乘法复杂度降低为$\Omega(N\log N)$，见图1[24]，注意，乘积的次数界为$2N$，因此需加倍次数界.

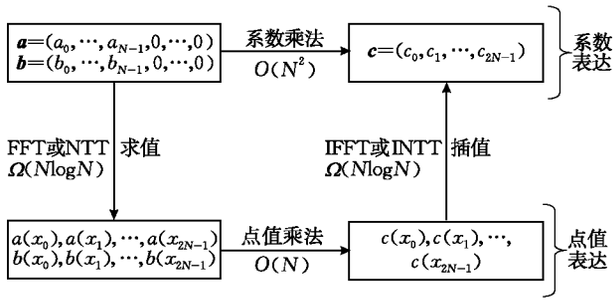

图1 多项式的系数乘法与点值乘法

计 算 机 学 报

（月刊，1978 年创刊）

第 45 卷　第 6 期　总第 474 期　2022 年 6 月

主　　管　中国科学院

主　　办　中国计算机学会
中国科学院计算技术研究所

主　　编　孙凝晖

编　　辑　《计算机学报》编辑委员会
中国科学院计算技术研究所
邮政编码 100190，北京 2704 信箱
E-mail：cjc@ict.ac.cn
http://cjc.ict.ac.cn

编辑部主任　李　刚

出　　版　科 学 出 版 社

印刷装订　北京科信印刷有限公司

总发行处　科 学 出 版 社
北京东黄城根北街 16 号
邮政编码 100717

国外总发行　中国国际图书贸易总公司
（中国国际书店）
北京 399 信箱

Chinese Journal of Computers

(Monthly, Started in 1978)

Vol. 45　No. 6　Series No. 474　June　2022

Supervised by Chinese Academy of Sciences

Sponsored by China Computer Federation, Institute of Computing Technology, CAS

Editor-in-Chief: SUN Ning-Hui

Edited by Editorial Board of Chinese Journal of Computers
P. O. Box 2704, Beijing 100190, China

Director: LI Gang

Published by Science Press

Printed by Beijing Kexin Printing Co., Ltd.

Distributed by Science Press
16 Donghuangchenggen North Street, Beijing 100717, China

Foreign: Guoji Shudian
P. O. Box 399, Beijing 100044, China

国内统一连续出版物号：CN 11-1826/TP

订 购 处：全国各邮电局

定　　价：73.00 元

国内邮发代号：2-833

国外发行代号：M 206

国内外公开发行

ISSN 0254-4164