基于MLWE的低膨胀率加密算法

柯程松[1,2] 吴文渊[2] 冯 勇[2]
(重庆邮电大学计算机科学与技术学院 重庆 400065)[1]
(中国科学院重庆绿色智能技术研究院自动推理与认知重庆市重点实验室 重庆 400714)[2]

摘 要 基于模容错学习问题(MLWE)的格密码算法Kyber具有抗量子攻击、加密效率高的优势,但密文膨胀率较大约为1:25,仅适用于密钥封装等少数场景。为了构建一种能够应用于一般的公钥加密场景的加密算法,提出了一种改进的基于MLWE的低膨胀率公钥加密算法。文中在Kyber加密算法中引入新的加密参数dp,扩大了明文空间,通过严格的理论推导与实验分析了dp对加密算法正确性的影响,并优化了加密参数,降低了密文膨胀率。改进后的算法会扩大有限域(即计算空间),导致直接使用原算法中的有限域上的多项式乘法运算,需调用额外的大整数计算库,从而降低了加密效率。通过使用基于浮点运算的复数域上的快速傅里叶变换进行多项式乘法,避免了在增大后的有限域上进行大整数多项式乘法。最后,对浮点运算产生的误差进行了分析,同时使用C++实现了改进算法,并将其与Kyber的实验数据进行对比。实验表明,所提算法在保证了计算效率的同时使密文膨胀率由1:25左右降低到了1:4.25左右。

关键词 模容错学习问题,公钥加密,密文膨胀率,复数域,浮点运算
中图法分类号 TP309 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.04.023

Low Expansion Rate Encryption Algorithm Based on MLWE

KE Cheng-song[1,2] WU Wen-yuan[2] FENG Yong[2]
(College of Computer Science & Technology, Chongqing University of Posts & Telecommunications, Chongqing 400065, China)[1]
(Chongqing Key Laboratory of Automated Reasoning & Cognition, Chongqing Institute of Green & Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China)[2]

Abstract The lattice encryption algorithm Kyber based on MLWE has the advantage of anti-quantum attack and high encryption efficiency. However, the expansion rate of ciphertext is as high as about 1:25, which is just suitable for a few scenes such as key encapsulation. In order to give an encryption algorithm that can be applied to common public key encryption scenes, an improved MLWE based low expansion rate public key encryption algorithm was proposed in this paper. A new encryption parameter dp was introduced into this algorithm to expand the plaintext space. By strict theoretical deduction and experiment, the influence of dp on the correctness of the encryption algorithm was analyzed, and the encryption parameter was optimized to reduce the expansion rate of ciphertext. The improved algorithm will work in a large finite field, which leads to low encryption efficiency. To avoid this situation, this paper proposed a polynomial multiplication method based on FFT in complex field by using floating point arithmetic. And the improved algorithm was implemented in C++. Experimental results show that the new method is more efficient and the floating error is controllable. Compared with Kyber, the ciphertext expansion rate is reduced from 1:25 to about 1:4.25.

Keywords Module learning with errors (MLWE), Public key encryption, Ciphertext expansion rate, Complex field, Floating point arithmetic

1 引言

量子计算机发展迅速,量子Shor算法[1]、量子Grover算法[2]等基于量子计算机的算法对传统密码学造成极大冲击。传统密码学大多基于数论上的困难问题,如当前应用最为广泛的传统密码有基于大数分解问题的RSA密码体制、基于离散对数问题的椭圆曲线密码等。如果量子计算机位数足够多,量子Shor算法能在多项式时间内破解整数分解、离散对数等问题,所有基于这两类问题的密码体制将不安全。

为了应对量子计算发展对传统密码产生的威胁,2015年

到稿日期:2018-03-09 返修日期:2018-07-03 本文受重庆市科委项目(cstc2017zdcy-yszxX0011),国家自然科学基金资助项目(11471307,11671377),中科院前沿重点项目(QYZDB-SSW-SYS026)资助。
柯程松(1994—),男,硕士生,主要研究方向为信息安全、同态加密;吴文渊(1976—),男,博士,研究员,主要研究方向为自动推理、混合计算、格密码,E-mail:wuwenyuan@cigit.ac.cn(通信作者);冯 勇(1965—),男,博士,研究员,主要研究方向为自动推理、符号计算、格理论。

8 月，美国国家安全局（NSA）发布将政府使用的密码套件 B 升级的通知[3]，作为抗量子密码体制实现之前的过渡。2016 年 4 月，美国国家标准局（NIST）提出抗量子密码路线图[4]，定义了公钥加密、数字签名和密钥交换的新的标准。由于现在已知的量子算法并不能在多项式时间内破解格上困难问题（如最短向量问题），所以有望通过格密码构建抗量子的密码体系。

1996 年，Ajtai[5] 开创性地提出一种随机格，求解这个格中的一个短向量与求解最短线性无关向量问题（SIVP）同样困难。这个问题被称作最小整数解问题（SIS），这个短向量可以与这个随机格同时生成，并作为公钥密码体制中的密钥。基于 SIS 问题的密码体制在 2008 年由 Gentry 等[6] 做了具体实现。Ajtai 等[7] 于 1997 年提出了第一个基于格的密码体制，即基于 Ajtai 对格中困难问题最坏情况到平均情况的规约。2005 年，Regev[8] 提出了容错学习问题（LWE），其成为了之后大多数基于格的密码体制的基础。由于基于 LWE 问题构造的密码体制效率较低，密文膨胀率较大，Lyubashevsky 等[9] 于 2010 年提出了环上容错学习问题（RLWE），提升了效率，降低了密文膨胀率。由于 RLWE 问题只能规约到理想格上的困难问题上，因此为了安全性，环的维数要取比较大的整数，从而导致计算效率不高。关于格密码的研究概况参见文献[10]。基于格密码有望构建抗量子攻击的密码体系，并且能够构造全同态加密方案，相关现状可参见文献[11]。

2015 年，Langlois 等[12] 提出了模容错学习问题（MLWE），并将 MLWE 问题归约到模格上的困难问题，能够在保证同等安全性的情况下，取更小的加密参数，进一步提高了加密效率，降低了密文膨胀率。2017 年，Bos 等[13] 基于 MLWE 构建了 Kyber 密钥交换算法，Kyber 算法已经成为后量子密码备选方案之一（刘亚敏等[14] 对基于格的后量子密码研究进行了概述）。由于该算法主要应用于密钥封装，因此只实现了明文空间是 $\{0,1\}^{256}$ 的加密，密文膨胀率较大约为 1∶25。密文膨胀率大意味着如果将该算法应用于一般的公钥加密场景，将极大地占用网络带宽以及用户或者云服务器的存储空间。

上述问题使该加密算法的使用领域受到局限。为了将加密算法用于如云服务等一般的公钥加密场景下，本文首先在 Kyber 原算法[13] 的基础上引入新的参数 dp，并通过严格的推导与实验分析了各个参数如何选取能够在保证正确性的情况下获得更低的密文膨胀率，最终将明文空间扩大到了 $\{0,1,\cdots,2^{16}-1\}^{256}$，密文膨胀率降低为 1∶4.25。原算法只局限于密钥封装，而改进后的算法能够用于更普遍的公钥加密。其次，由于本文改进算法将增大有限域（即计算空间），需要进行大整数计算，如果使用原算法中的方法（有限域上的多项式乘法）进行计算，需要调用专门的大整数计算软件包，将会降低算法的效率。因此，本文通过使用基于浮点运算的复数域上的快速傅里叶变换进行多项式乘法，避免了在增大后的有限域上进行大整数多项式乘法，实验结果表明本文方法的计算效率更高。本文对浮点运算产生的误差进行了分析，说明了误差不会对算法的正确性产生影响。

本文第 2 节介绍了本文算法中出现的一些符号以及预备知识；第 3 节首先回顾了 Kyber 原加密算法[13]，详述了本文对该算法的改进，然后分析了加密参数与算法正确性的关系并给出了优化参数，最后证明了本文算法是 IND-CPA 安全的；第 4 节说明了本文使用的基于浮点运算的复数域上的多项式乘法及其稳定性；第 5 节给出实验结果，并对比了原算法、本文算法，以及几个同样基于容错问题构造的抗量子攻击加密算法的性能；最后总结全文，并对未来值得关注的研究方向进行初步探讨。

2　预备知识

本节给出了本文算法中出现的符号以及一些相关概念，并解释了为何选择 MLWE 问题作为加密算法的困难问题。

2.1　基础符号

令 $R=\mathbb{Z}[X]/f(X)$ 表示多项式环，$R_q=\mathbb{Z}_q[X]/f(X)$ 表示模 q 多项式环。其中如果未作特殊说明，$f(X)=X^n+1$，n 表示 f 的次数。

如果 q 为大于 2 的素数，令 $\bmod q$ 表示模 q，$\bmod q$ 的取值范围是 $[-(q-1)/2,(q-1)/2]$，$\bmod^+ q$ 的取值范围是 $[0,q-1]$，$\bmod^- q$ 的范围是 $[-(q-1),0]$。

对于 $x\in\mathbb{R}$，定义 $round(x)$ 表示四舍五入，$\lceil x\rceil$ 表示向上取整，$\lfloor x\rfloor$ 表示向下取整。

对于 R_q 上的多项式 $u(x)=u_0+u_1x+\cdots+u_{n-1}x^{n-1}$，用 u 表示 $u(x)$，u 的无穷范数为 $\|u\|_\infty=\max\{|u_j|\}$，$u$ 的第二范数为 $\|u\|_2=\sqrt{\sum_{j=0}^{n-1}(u_j)^2}$，由多个多项式 $(u_1,u_2,\cdots,u_n)\in R_q^n$ 组成的向量 $\boldsymbol{u}$ 的无穷范数为 $\|\boldsymbol{u}\|_\infty=\max\|u_j\|$。如未作特殊说明，通常使用 $\|\cdot\|$ 简略表示无穷范数。

令 $a\leftarrow\boldsymbol{A}$ 表示从集合 $\boldsymbol{A}$ 均匀选取元素 a，或从分布 $\boldsymbol{A}$ 中均匀取样元素 a。本文的 log 均表示以 2 为底的对数。令 $\Pr[\cdot]$ 表示概率，$negl$ 表示概率忽略不计。令 $\mathbb{C}$ 表示复数域。令 $\boldsymbol{\Lambda}$ 表示对角矩阵，$diag(\lambda_1,\cdots,\lambda_n)$ 表示对角线元素为 λ_j 的对角矩阵。

2.2　中心二项分布

本文采用与 Kyber[13] 原算法中同样的噪声分布 B_η，其定义如下。

均匀随机取 $(a_1,\cdots,a_\eta,b_1,\cdots,b_\eta)\leftarrow\{0,1\}^{2\eta}$，输出 $\sum_{j=1}^{\eta}(a_j-b_j)$。如果从 B_η 取样 k 个系数满足分布的多项式 $\boldsymbol{v}$，则称 $\boldsymbol{v}\leftarrow\beta_\eta^k$，其中 $\boldsymbol{v}$ 是 k 个系数满足 B_η 分布的多项式组成的向量。

根据 Bai 等[15] 的工作，β_η 是一个期望为 0、方差为 $\eta/2$ 的近似高斯分布，能够用作容错学习问题中的噪声分布。

2.3　压缩技术（Compress）与解压缩技术（Decompress）

本文使用 Kyber 原算法中的压缩技术（Compress）与解压缩技术（Decompress）。

定义函数（$Compress_q(x,d)$）：输入 $x\in\mathbb{Z}_q$，$d<\lceil\log q\rceil$，输出 $y=round((2^d/q)\cdot x)\bmod^+ 2^d$。

定义函数 $Decompressq(y,d)$：输入 $y=Compress_q(x,d)$，输出 $x'=round((q/2^d)\cdot y)$。

对整数 $x\in Z_q$ 先压缩然后解压缩会产生误差：$|x-x'|\bmod q\leqslant round(q/2^{d+1})$。

引入压缩技术(Compress)与解压缩技术(Decompress)的目的是在一定程度上压缩公钥与密文的大小。但是其会引入误差,因此为了保证算法的正确性,必须选择合适的参数(将在第 3 节中详细分析)。

2.4 MLWE 问题

MLWE 问题是容错学习问题的新进展,本节简要回顾了容错学习问题的发展进程。

2005 年,Regev[8] 提出了 LWE 问题。设安全参数为 λ,$n=n(\lambda)$为向量的维数,$q=q(\lambda)\geqslant 2$ 表示模数,$\chi=\chi(\lambda)$是 $\mathbb{Z}_q$ 上的一个分布。$LWE_{n,q,\chi}$ 问题即区分以下两个分布:

1)分布$(\boldsymbol{a}_j,b_j)$,其中均匀随机选取向量 $\boldsymbol{a}_j\leftarrow\mathbb{Z}_q$,均匀随机取整数 $b_j\leftarrow\mathbb{Z}_q$。

2)分布$(\boldsymbol{a}_j,b_j)$,其中均匀随机选取向量 $a_j\leftarrow\mathbb{Z}_q$,均匀选取向量 $s\leftarrow\mathbb{Z}_q$,从噪声分布中取样整数 $e_j\leftarrow\chi$,$b=\langle\boldsymbol{a}_j,\boldsymbol{s}\rangle+e_j$。

$LWE_{n,q,\chi}$ 问题的困难程度归约到了格上最短线性无关向量问题(SIVP)。由于基于 LWE 问题来构造加密算法时,需要进行大量的矩阵运算,存在计算效率过低、密钥较大、密文膨胀率较大的缺点,Lyubaskevsky 等[9] 于 2008 年提出了环上容错学习问题(RLWE),作为 LWE 的改进。

设安全参数为 λ,$n=n(\lambda)$为 2 的幂,$f(x)=x^n+1$,模数 $q=q(\lambda)\geqslant 2$。多项式环 $R=\mathbb{Z}[x]/f(x)$,模 q 多项式环 $R_q=\mathbb{Z}_q[x]/f(x)$,分布 $\chi=\chi(\lambda)$是 R_q 上的一个分布。$RLWE_{n,q,\chi}$ 问题即区分以下两个分布:

1)分布(a_j,b_j),其中均匀随机选取多项式 $a_j\leftarrow R_q$,均匀随机取多项式 $b_j\leftarrow R_q$。这里 $1\leqslant j\leqslant N$,$N=\log(q\cdot\lambda^{\omega(1)})$。

2)分布(a_j,b_j),其中均匀随机选取多项式 $a_j\leftarrow R_q$,均匀选取多项式 $s\leftarrow R_q$,从噪声分布中取样多项式 $e_j\leftarrow\chi$,$b_j=a_j\cdot s+e_j$。

$RLWE_{n,q,\chi}$ 问题归约到了理想格上的最短向量问题(SVP)。由于 RLWE 问题只能规约到理想格上的困难问题,因此通过 RLWE 问题构建加密算法时,多项式的个数要取 $N=\log(q\cdot\lambda^{\omega(1)})$个,$N$ 会影响加密效率与密文膨胀率。2015 年,Langlois 等[12] 提出了模容错学习问题(MLWE),并将 MLWE 问题归约到模格上的困难问题。模格是一般格与理想格的推广,可保证在同等安全性的前提下选取更小的加密参数,以此提升加密效率,降低密文膨胀率。

设安全参数为 λ,$d=d(\lambda)$为 2 的幂,$f(x)=x^d+1$,模数 $q=q(\lambda)\geqslant 2$。多项式环 $R=\mathbb{Z}[x]/f(x)$,模 q 多项式环 $R_q=\mathbb{Z}_q[x]/f(x)$。分布 $\chi=\chi(\lambda)$是 R_q 上的一个分布。令 k 为模的维数,一般来说 $k=2$ 时就能够抵御敌手 2^{100} 次攻击,如果需要获得更高的安全性,可以通过本文思想获得 $k>2$ 的情形。本文以 $k=2$ 为例,则 $MLWE_{n,q,\chi,k}$ 问题即区分以下两个分布:

1)分布:$\left(\begin{bmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{bmatrix},\begin{bmatrix}b_1\\b_2\end{bmatrix}\right)$。其中均匀随机选取多项式 $a_{jr}\leftarrow R_q(1\leqslant j\leqslant k,1\leqslant r\leqslant k)$,$b_j\leftarrow R_q(1\leqslant j\leqslant k)$。

2)分布:$(\begin{bmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{bmatrix},\begin{bmatrix}b_1\\b_2\end{bmatrix}=\begin{bmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{bmatrix}\cdot\begin{bmatrix}s_1\\s_2\end{bmatrix}+\begin{bmatrix}e_1\\e_2\end{bmatrix})$,简记为$(\boldsymbol{A},\boldsymbol{b}=\boldsymbol{A}\cdot\boldsymbol{s}+\boldsymbol{e})$。其中均匀随机选取多项式 $a_{jr}\leftarrow R_q(1\leqslant j\leqslant k,1\leqslant r\leqslant k)$,$s_j\leftarrow R_q(1\leqslant j\leqslant k)$,$e_j\leftarrow\chi(1\leqslant j\leqslant k)$。此处向量 s 即加密算法中的私钥,A 是加密算法的公钥。

当 $k=1$ 时,$MLWE_{n,q,\chi,k}$ 就转化为 $N=2$ 的 RLWE 问题,当将 R_q 变为$\mathbb{Z}_q$时 $MLWE_{n,q,\chi,k}$ 就转化为 LWE 问题。MLWE 问题是将 LWE 问题与 RLWE 问题的优缺点做了平衡,既保证 LWE 在一般格上的安全性,又保持了 RLWE 的加密效率高、密钥小、密文膨胀率低的优势。Langlois 等[12] 将 MLWE 问题归约到模格上的最短线性无关向量问题(SIVP),其在保证同等安全性的前提下使加密参数取得更小值。

3 Kyber 算法的改进

随着量子计算机的发展,人们对抗量子密码算法的需求越来越迫切。Kyber 算法[13] 基于 MLWE 问题构建,有望入选抗量子密码体制,但是 Kyber 算法明文空间较小,仅适用于密钥封装。基于格构建的密码都有可能推广为同态加密方案,从这方面考虑,人们对明文空间的需求必然不只是$\{0,1\}^{256}$;并且 Kyber 算法的密文膨胀率为1∶25,也就是说加密 100 MB 明文,会生成约 2.5 GB 密文,如果将该加密算法用于云计算或云存储,将会消耗极大的网络带宽以及云存储空间。为了解决上述问题,本文对 Kyber 算法进行了改进。

3.1 Kyber 密钥封装算法

本节首先回顾了 Kyber 密钥封装算法。

Kyber 密钥封装算法总共分为 3 步:1)密钥生成,$Kyber.keygen(parameters)$;2)加密,$Kyber.Enc(pk,m)$;3)解密 $Kyber.Dec(sk,c)$。

1)$Kyber.keygen(parameters)$

设安全参数为 λ,根据 $MLWE_{n,q,\chi,k}$ 中的安全性规约,输入加密参数 $n=n(\lambda)$(n 必须是 2 的幂),$k=k(\lambda)$,模数 $q=q(\lambda)=1\bmod 2n$,环 $R_q=\mathbb{Z}_q[x]/x^n+1$,噪声分布 β_η,公钥压缩参数 dt。均匀随机取样 $\boldsymbol{A}\leftarrow R_q^{k\times k}$。均匀取样私钥与噪声$(\boldsymbol{s},\boldsymbol{e})\leftarrow\beta_\eta^k\times\beta_\eta^k$。计算 $\boldsymbol{t}:=Compress_q(\boldsymbol{A}\boldsymbol{s}+\boldsymbol{e},dt)$。输出公钥 $pk:=(\boldsymbol{t},\boldsymbol{A})$,私钥 $sk:=\boldsymbol{s}$。

2)$Kyber.Enc(pk,m)$

输入公钥 $pk:=(\boldsymbol{t},\boldsymbol{A})$,明文 $m\in\{0,1\}^n$,公钥压缩参数 dt,密文压缩参数 du 和 dv。对公钥进行解压缩操作 $\boldsymbol{t}':=Decompress_q(\boldsymbol{t},dt)$。均匀随机取样随机向量及噪声$(\boldsymbol{r},\boldsymbol{e}_1,e_2)\leftarrow\beta_\eta^k\times\beta_\eta^k\times\beta_\eta$。加密得到密文 $\boldsymbol{u}:=Compress_q(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}+\boldsymbol{e}_1,du)$。

$$v:=Compress_q(\boldsymbol{t}'^{\mathrm{T}}\boldsymbol{r}+e_2+round(\frac{q}{2})\cdot m,dv) \tag{1}$$

输出密文 $c:=(\boldsymbol{u},v)$。

3)$Kyber.Dec(sk,c)$

输入私钥 $sk:=\boldsymbol{s}$,密文 $c:=(\boldsymbol{u},v)$,密文解压缩参数 du 和 dv。解压缩得到 $\boldsymbol{u}'=Decompress_q(\boldsymbol{u},du)$与 $v'=Decompress_q(v,dv)$。输出明文:

$$m':=Compress_q(v'-\boldsymbol{s}^{\mathrm{T}}\boldsymbol{u}',1) \tag{2}$$

3.2 算法的改进

为了保证计算空间在多项式环 R_q 内,通过式(1)可知,Kyber 算法的明文空间只能是$\{0,1\}^{256}$,为了扩大明文空间必

须引入一个新的参数 dp。

具体地，将式(1)改为：$v:=Compress_q(\boldsymbol{t}^{\mathrm{T}}\boldsymbol{r}+e_2+round(\frac{q}{2^{dp}})\cdot m,dv)$，此处将原算法中的$\frac{q}{2}$改进为$\frac{q}{2^{dp}}$，即可以输入范围为 $0\leqslant m_j<2^{dp}$ 的明文，为了正确解密，式(2)必须改为 $m':=Compress_q(v-\boldsymbol{s}^{\mathrm{T}}\boldsymbol{u},dp)$，其他步骤均与原方案相同。但是改进之后，模数 q 必然需要增大，加密算法中的噪声也会增大，将对解密正确性产生影响，以下分析模数 q 与新参数 dp 的关系。

设 *Compress* 和 *Decompress* 产生的误差为：$\boldsymbol{t}'-\boldsymbol{t}=\boldsymbol{c}_t$，$\boldsymbol{u}'-\boldsymbol{u}=\boldsymbol{c}_u$，$v'-v=c_v$。

定理 1　1)解密时的噪声为 $\varepsilon=\boldsymbol{e}^{\mathrm{T}}\boldsymbol{r}+e_2+c_v+\boldsymbol{c}_t^{\mathrm{T}}\boldsymbol{r}-\boldsymbol{s}^{\mathrm{T}}\boldsymbol{e}_1+\boldsymbol{s}^{\mathrm{T}}\boldsymbol{c}_u$；2)当噪声的范数 $\|\varepsilon\|$ 与本文算法中的模数 q 和压缩参数 dp 满足以下关系时，算法能解密得到正确的明文。

$$\Pr(\|\varepsilon\|<round(\frac{q}{2^{dp+1}}))>1-negl \tag{3}$$

证明：将误差代入算法中进行计算：

$\boldsymbol{u}':=Decompress(\boldsymbol{u},du)=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}+\boldsymbol{e}_1+\boldsymbol{c}_u$

$v':=Decompress(v,dv)=(\boldsymbol{A}\boldsymbol{s}+\boldsymbol{e}+\boldsymbol{c}_t)^{\mathrm{T}}\boldsymbol{r}+e_2+round(\frac{q}{2^{dp}})\cdot m+c_v=(\boldsymbol{A}\boldsymbol{s}+\boldsymbol{e})^{\mathrm{T}}\boldsymbol{r}+e_2+round(\frac{q}{2^{dp}})\cdot m+c_v+\boldsymbol{c}_t^{\mathrm{T}}\boldsymbol{r}$

因此在解密时，有：

$$v'-\boldsymbol{s}^{\mathrm{T}}\boldsymbol{u}'=\boldsymbol{e}^{\mathrm{T}}\boldsymbol{r}+e_2+round(\frac{q}{2^{dp}})\cdot m+c_v+\boldsymbol{c}_t^{\mathrm{T}}\boldsymbol{r}-\boldsymbol{s}^{\mathrm{T}}\boldsymbol{e}_1+\boldsymbol{s}^{\mathrm{T}}\boldsymbol{c}_u=\varepsilon+round(\frac{q}{2^{dp}})\cdot m$$

定理 1 1)得证。

又有 $m':=Compress_q(v'-\boldsymbol{s}^{\mathrm{T}}\boldsymbol{u}',dp)$，如果要使得 $m=m':=Compress_q(v'-\boldsymbol{s}^{\mathrm{T}}\boldsymbol{u}',dp)$，必须满足条件：$m-m'=\varepsilon\cdot\frac{2^{dp}}{q}<\frac{1}{2}$，即满足式(3)，定理 1 2)得证。证毕。

以下给出了密文膨胀率与压缩参数 dt,du,dv 和新的压缩参数 dp 的定量关系。

定理 2　设密文膨胀率为 ρ，压缩参数为 du,dv,dp(同第 3.2 节中描述的压缩参数)，它们的关系为：

$$\rho=\frac{2\cdot du+dv}{dp} \tag{4}$$

证明：明文 m 为 n 个小于 2^{dp} 的整数，由于安全参数 $k=2$，因此密文 $\boldsymbol{u}:=Compress_q(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}+\boldsymbol{e}_1,du)$ 是 $2n$ 个小于 2^{du} 的整数，密文 $v:=Compress_q(\boldsymbol{t}^{\mathrm{T}}\boldsymbol{r}+e_2+round(q/2^{dp})\cdot m,dv)$ 是 n 个小于 2^{dv} 的整数，由此式(4)得证。证毕。

Kyber 原算法的压缩参数 $du=11,dv=3,dt=11$，明文空间为{0,1}，即 $dp=1$，根据式(4)，原算法的密文膨胀率为 1:25。

3.3　优化参数

根据 Kyber 原算法[13]的分析，当安全参数 $\lambda=102$ 时，$k=k(\lambda)=2,n=n(\lambda)=256$，噪声分布的参数 $\eta=5$。为了保证安全性，本文相应参数与上述参数相同。模数 q 决定计算空间，提高模数 q 不会对安全性产生影响，由于使用 64 位操作系统，模数 q 必须小于 2^{32}，因此本文使用了基于浮点运算的复数域上的多项式乘法，模数 q 不必满足 $q\equiv1 \bmod 2n$ 的条件，取 $q=4294967291$。如果要取更大的模数 q，则超过了 64 位操作系统的限制。根据定理 1，解密的正确性还与噪声范数 $\|\varepsilon\|$ 有关，本节分析了各个参数与噪声范数 $\|\varepsilon\|$ 的关系，并给出了优化的参数(在保证正确性的前提下密文膨胀率尽可能小)。

本文算法将模数 q 增大为 4294967291，加密产生的噪声范数 $\|\varepsilon\|$ 会随之增大，因此需要分析各个参数与 $\|\varepsilon\|$ 的关系，然后选择优化的新参数。

根据定理 1 中的证明，噪声的形式为：

$\varepsilon=\boldsymbol{e}^{\mathrm{T}}\boldsymbol{r}+e_2+c_v+\boldsymbol{c}_t^{\mathrm{T}}\boldsymbol{r}-\boldsymbol{s}^{\mathrm{T}}\boldsymbol{e}_1+\boldsymbol{s}^{\mathrm{T}}\boldsymbol{c}_u$

ε 是由系数满足两种不同类型的分布的多项式计算得到。第一种类型为：

$(\boldsymbol{s},\boldsymbol{e},\boldsymbol{e}_1,e_2,\boldsymbol{r})\leftarrow\beta_\eta^k\times\beta_\eta^k\times\beta_\eta^k\times\beta_\eta\times\beta_\eta^k$

根据 Bai 等[15]的工作，β_η 近似于 $N(0,\eta/2)$ 分布，随机变量的取值是整数且在$[-\eta,\eta]$之间。

第二种类型为：

$(\boldsymbol{c}_t,\boldsymbol{c}_u,c_v)\leftarrow\Psi_{dt}^k\times\Psi_{du}^k\times\Psi_{dv}$

根据 Kyber 算法，对 x 进行压缩和解压缩得到误差 y：

$y=(x'-x) \bmod q=(Decompress_q(Compress_q(x,d),d)-x) \bmod q$

其中，$|y|\leqslant B_q=round(\frac{q}{2^{d+1}})$且满足分布 Ψ_d。Ψ_d 是期望为 0、方差为$\frac{B_q^2}{3}$的近似均匀分布，近似指的是当$-B_q<y<B_q$ 时 Ψ_d 是均匀分布的，在 $y=\pm B_q$ 的概率密度稍小，可以近似看作均匀分布。

首先对噪声 ε 的表达式进行化简，得到：

$$\varepsilon=(\boldsymbol{e}^{\mathrm{T}}+\boldsymbol{c}_t^{\mathrm{T}})\boldsymbol{r}+\boldsymbol{s}^{\mathrm{T}}(\boldsymbol{c}_u-\boldsymbol{e}_1)+e_2+c_v \tag{5}$$

其中，$\boldsymbol{e}^{\mathrm{T}}+\boldsymbol{c}_t^{\mathrm{T}}$ 与 $\boldsymbol{c}_u-\boldsymbol{e}_1$ 计算结果的全部系数满足期望为 0、方差为$\frac{(B_q+\eta)^2}{3}$的近似均匀分布。为了均衡公钥与密文之间的压缩参数，本文采取原算法中相同的策略，即使 $du=dt$，所以这里 $B_q=\frac{q}{2^{du+1}}=\frac{q}{2^{dt+1}}$。令系数满足该分布的多项式表示为 $W=w_0+w_1x+\cdots+w_{n-1}x^{n-1}$。

式(5)中 $\boldsymbol{r}$ 与 $\boldsymbol{s}^{\mathrm{T}}$ 均由系数满足 $N(0,\eta/2)$的分布组成，设这类多项式为 $U=u_0+u_1x+\cdots+u_{n-1}x^{n-1}$。

令多项式 $Y=W\cdot U=y_0+y_1x+\cdots+y_{n-1}x^{n-1}$。该乘法是 R_q 上的多项式乘法，容易推出：

$$(y_0,\cdots,y_{n-1})=(w_0,\cdots,w_{n-1})\cdot\begin{bmatrix}u_0 & u_1 & u_2 & \cdots & u_{n-1}\\ -u_{n-1} & u_0 & u_1 & \cdots & u_{n-2}\\ -u_{n-2} & -u_{n-1} & u_0 & \cdots & u_{n-3}\\ \vdots & \vdots & \vdots & \cdots & \vdots\\ -u_1 & -u_2 & -u_3 & \cdots & u_0\end{bmatrix}$$

为了描述方便，不妨使用 y_0 进行分析，所有 y_j 具有与 y_0 相同的性质。Y 的系数 $y_0=w_0u_0-w_1u_{n-1}-\cdots-w_{n-1}u_1$，是 n 个独立同分布的 w_gu_h 之和，可以通过中心极限定理得到 y_0 满足的分布。因此下面需分析 w_gu_h 满足的分布的期望与方差。

因为 w_g 与 u_h 是两个独立分布，设 w_gu_h 满足期望为 $E(w_gu_h)$、方差为 $D(w_gu_h)$的分布，所以有：

$$E(w_g u_h)=E(w_g)\cdot E(u_h)=0$$

$$D(w_g u_h)=\sum_{g=0}^{n-1}\sum_{h=0}^{n-1}(w_g\cdot u_h)^2\cdot \Pr[w_g]\cdot \Pr[u_h]-E^2(w_g u_h)$$
$$=D(w_g)\cdot D(u_h)$$

得到：

$$D(w_g u_h)=\frac{\eta\cdot(B_q+\eta)^2}{6}=\frac{\eta}{6}\cdot\left(\frac{q}{2^{du+1}}+\eta\right)^2$$

式(5)中，$(\boldsymbol{e}^{\mathrm{T}}+\boldsymbol{c}_t^{\mathrm{T}})\boldsymbol{r}$ 与 $\boldsymbol{s}^{\mathrm{T}}(\boldsymbol{c}_u-\boldsymbol{e}_1)$ 均是 $Y=W\cdot U$ 型多项式乘法。又由于本文取 $k=2$，所以 $(\boldsymbol{e}^{\mathrm{T}}+\boldsymbol{c}_t^{\mathrm{T}})\boldsymbol{r}+\boldsymbol{s}^{\mathrm{T}}(\boldsymbol{c}_u-\boldsymbol{e}_1)$ 的计算结果的每个系数均是 1024 个独立同分布的 $w_g u_h$ 之和。根据中心极限定理，该系数满足的分布近似于 $N(0,\frac{1024\cdot\eta}{6}\cdot(\frac{q}{2^{du+1}}+\eta)^2)$ 分布。

根据定理 1，噪声 ε 的范数必须满足 $\|\varepsilon\|=\|(\boldsymbol{e}^{\mathrm{T}}\boldsymbol{r}-\boldsymbol{s}^{\mathrm{T}}\boldsymbol{e})+(\boldsymbol{c}_t^{\mathrm{T}}\boldsymbol{r}+\boldsymbol{s}^{\mathrm{T}}\boldsymbol{c}_u)+e_2+c_v\|<round(\frac{q}{2^{dp+1}})$ 才能保证算法的正确性，其中 e_2 可以忽略不计，$\|c_v\|\leqslant(q/2^{dv+1})$。标准正态分布中，随机变量大于 12 倍标准差的概率为 $2^{-107.8}$，可以忽略不计。由此得到：

$$\Pr(\|\varepsilon\|>12\cdot\sqrt{\frac{2\cdot 512\cdot\eta}{6}\cdot\left(\frac{q}{2^{du+1}}+\eta\right)^2}+\frac{q}{2^{dv+1}})=negl$$

即加密参数满足：

$$384\cdot\sqrt{\frac{\eta}{6}\cdot\left(\frac{q}{2^{du+1}}+\eta\right)^2}+\frac{q}{2^{dv+1}}>\frac{q}{2^{dp+1}}$$

又根据定理 2 密文膨胀率 $\rho=\frac{2\cdot du+dv}{dp}$，可以得到，当参数 $du=dt=25$，$dv=18$，$dp=16$ 时，膨胀率 $\rho=4.25$ 是较为满意的结果。此时算法出错的概率约为 $2^{-107.8}$ 可以忽略不计。通过 C++编写程序验证了算法出错的可能性是可以忽略不计的，但如果进一步减小压缩参数，算法出错的概率较高。通过上述加密参数，本文的明文空间为 $\{0,1,\cdots,2^{16}-1\}^{256}$，密文膨胀率降低为 1∶4.25。

3.4 安全性分析

定理 3 在 $MLWE_{n,q,\chi,k}$ 问题假设的前提下，本文的公钥加密方案是 IND-CPA 安全的。

证明：使用基于游戏的证明方法，用 $Adv_{\mathrm{Game}}[\mathcal{A}]$ 表示敌手 $\mathcal{A}$ 在下列游戏中的优势。

Game 0：Game 0 即标准的 IND-CPA 游戏，挑战者 C 初始化加密方案，按照方案生成公钥 $pk:=(\boldsymbol{t},\boldsymbol{A})$，私钥 $sk:=\boldsymbol{s}$。敌手 $\mathcal{A}$ 获得方案中的公钥，并从明文空间中选择两个挑战明文 m_0 和 m_1 发送给挑战者 C。挑战者 C 随机选择 $b\in\{0,1\}$，将 m_b 用公钥加密，并将密文发送给敌手。敌手 $\mathcal{A}$ 猜测密文所对应的明文输出 b'，如果 $b'=b$ 则敌手 $\mathcal{A}$ 攻击成功，敌手 $\mathcal{A}$ 的优势记为 $Adv_{CPA}[\mathcal{A}]=|\Pr[b=b'\text{ in Game 0}]-1/2|$。

Game 1：Game 1 改变 Game 0 中公钥 $\boldsymbol{t}:=\boldsymbol{A}\boldsymbol{s}+\boldsymbol{e}$ 的生成过程，均匀随机取 $\boldsymbol{t}'\leftarrow R_q^k$，敌手 $\mathcal{A}$ 区分 $\boldsymbol{t}'$ 是否等于 $\boldsymbol{A}\boldsymbol{s}+\boldsymbol{e}$ 与 $MLWE_{n,q,\chi,k}$ 问题一样难，所以 Game 1 与 Game 0 中敌手 $\mathcal{A}$ 的优势差为：$|Adv_{Game}[\mathcal{A}]-Adv_{CPA}[\mathcal{A}]|\leqslant Adv_{MLWE}[\mathcal{A}]$。

Game 2：Game 2 改变 Game 1 中挑战密文的生成方式，不使用公钥加密得到挑战密文，而是均匀随机取 $(\boldsymbol{u}',v')\leftarrow R_q^k\times R_q$，敌手 $\mathcal{A}$ 区分 $(\boldsymbol{u}',v')$ 是否等于 $(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{r}+\boldsymbol{e}_1,\boldsymbol{t}^{\mathrm{T}}\boldsymbol{r}+e_2+round(\frac{q}{2^{dp}})\cdot m)$ 与 $MLWE_{n,q,\chi,k}$ 问题一样难，所以 Game 2 与 Game 1 中敌手 $\mathcal{A}$ 的优势差为 $|Adv_{Game2}[\mathcal{A}]-Adv_{Game1}[\mathcal{A}]|\leqslant Adv_{MLWE}[\mathcal{A}]$。

因此，总的来说：

$Adv_{CPA}[\mathcal{A}]\leqslant 2\cdot Adv_{MLWE}[\mathcal{A}]$

在 $MLWE_{n,q,\chi,k}$ 问题是困难的假设成立的情况下，$Adv_{MLWE}[\mathcal{A}]$ 可以忽略不计，本文的公钥加密方案可证明是 IND-CPA 安全的。

4 基于浮点运算的复数域多项式乘法

算法中多项式乘法的效率决定了整个计算效率，根据第 3.3 节的分析，本文加密算法的模数 $q=4\,294\,967\,291$，极大地扩大了计算的有限域，那么如果采用原算法中的方法(有限域上的多项式乘法)，则须调用 NTL 大整数计算库，导致加密效率下降。因此本文通过用基于浮点运算的复数域上的快速傅里叶变换进行多项式乘法，然后对浮点运算的结果取整得到整数域$\mathbb{Z}$上的数，最后将$\mathbb{Z}$上的数模参数 q 得到有限域$\mathbb{Z}_q$上的最终结果。本节先简单介绍本文的多项式乘法的实现，然后分析浮点运算产生的误差对结果的影响。

4.1 多项式乘法的实现

一般情况下，需要计算两个多项式的乘积 $r(x)=f(x)\cdot g(x)$，其中 $f(x)$ 与 $g(x)$ 的系数分别为 $\boldsymbol{a}=\begin{bmatrix}a_0\\ \vdots\\ a_{n-1}\end{bmatrix}$ 与 $\boldsymbol{g}=\begin{bmatrix}z_0\\ \vdots\\ z_{n-1}\end{bmatrix}$。那么 $r(x)$ 的系数 $\boldsymbol{r}=\begin{bmatrix}r_0\\ \cdots\\ r_{n-1}\end{bmatrix}=\boldsymbol{F}_n^{-1}\cdot(\boldsymbol{F}_n\cdot\boldsymbol{a}\cdot diag(\overline{\boldsymbol{F}_n}\cdot\boldsymbol{g}))^{\mathrm{T}}$，其中 $\boldsymbol{F}_n=(f_{jr})\in\mathbb{C}^{n\times n}$ 是一个循环矩阵，$f_{jr}=\omega_n^{(j-1)(r-1)}$，$n$ 次单位复数根 $\omega_n=\mathrm{e}^{-\frac{2\pi i}{n}}=\cos(\frac{2\pi}{n})-i\cdot\sin(\frac{2\pi}{n})$(具体证明参考文献[16])。在计算 $\boldsymbol{r}$ 的过程中，$\boldsymbol{F}_n\cdot\boldsymbol{a}$ 与 $\overline{\boldsymbol{F}_n}\cdot\boldsymbol{g}$ 是两次复数域上的快速傅里叶变换，$\boldsymbol{F}_n^{-1}\cdot$ 后面的向量是一次复数域上的快速傅里叶变换的逆变换，因此计算复杂度为 $O(3n\log n+n)$。最后计算 $\boldsymbol{r}\bmod q$ 使 $r(x)\in R_q$。原算法使用有限域上的快速傅里叶变换，q 必须满足 $q\equiv 1\bmod 2n$ 的条件，即$\mathbb{Z}_q$上有 $2n$ 个本原单位根，复数域上的快速傅里叶变换对 q 是没有限制条件的。

经过实验，在取同等大小的模数 q 的情况下，本文使用基于浮点运算的复数域上的快速傅里叶变换进行多项式乘法，比调用大整数计算包 NTL 直接在有限域上计算快 6 倍左右。

4.2 算法的计算复杂度

本节讨论了本文算法各个阶段的计算复杂度。以下所有的 n,k,dp 均对应第 3 节中给出的本文加密算法中的参数。

调用本文算法加密一次时，是将 n 个 dp 比特的整数作为明文多项式 $m\in R_q$ 的系数进行加密，所以本文算法一次能够加密 $n*dp$ 比特明文。如果加密的目标明文大小为 x 比特，则需调用本文算法加密$\lceil x/(n*dp)\rceil$次。向上取整指的是如果最后一次加密时的明文不足 $n*dp$ 比特，则将剩下的

位置用 0 补全。下面分析用本文算法加密一次（即加密 $n*dp$ 比特明文）的计算复杂度。

在生成公私钥对时由于要调用中心二项分布函数生成满足算法要求分布的噪声，时间复杂度为 $O(n)$，空间复杂度为公钥大小 $O(k^2n)$。整个算法中无论加密多少明文，生成公私钥对的操作只需进行一次。

在进行加密操作时，主要消耗的计算量是多项式乘法，其他计算（如压缩与解压缩操作）的计算复杂度可以忽略不计。加密时要进行 k^2+k 次多项式乘法，根据 4.1 节对多项式乘法计算复杂度的分析，加密时的时间复杂度是 $O((k^2+k)\cdot(3n\log n+n))$；根据文献[16]，空间复杂度为快速傅里叶变换所消耗的空间 $O(n)$。

同加密操作中的分析，解密操作要进行 k 次多项式乘法，解密的时间复杂度为 $O(k\cdot(3n\log n+n))$，空间复杂度为 $O(n)$。

使用本文算法加密 $n*dp$ 比特明文的各阶段的计算复杂度如表 1 所列。

表 1　算法计算复杂度

Table 1　Algorithmic computational complexity

算法阶段	时间复杂度	空间复杂度
密钥生成	$O(n)$	$O(k^2n)$
加密操作	$O((k^2+k)\cdot(3n\log n+n))$	$O(n)$
解密操作	$O(k\cdot(3n\log n+n))$	$O(n)$

4.3　浮点计算的误差

本节讨论了基于浮点运算的复数域上的快速傅里叶变换进行多项式乘法产生的误差，并给出实验结果，说明了本文的加密方案中多项式乘法的稳定性。

引理 1[17]　令 $C\in\mathbb{C}^{n\times n}$ 是一个循环矩阵，$a\in\mathbb{C}^n$ 是一个向量。用复数域上的快速傅里叶变换求向量 $\boldsymbol{b}=\boldsymbol{Ca}$，产生的误差为：

$$\frac{\|\Delta b\|_2}{\|b\|_2}\leqslant\eta\log n+6u+O(u^2)$$

其中，$\eta=\mu+\gamma_n(\sqrt{2}+\mu)$，$\gamma_n=\dfrac{nu}{1-nu}$，$\mu=cu$，$u$ 表示数据精度（一般 double 形的精度为 10^{-16}），$|c|\leqslant1$ 是一个常数。

根据引理 1，误差与多项式的维数 n 以及 $\boldsymbol{b}$ 的范数有关。随着维数 n 以及 $\boldsymbol{b}$ 的范数的增大，误差随之变大。这里使用不同的维数 n 与 $\|\boldsymbol{b}\|_\infty$ 进行误差实验，通过实验给出近似的最坏误差。

本文方案中使用 $\|\boldsymbol{b}\|_\infty\leqslant10^{11}$ 并使用 $\|\boldsymbol{b}\|_\infty=10^{11}$ 作为固定参数测试维数 n 对误差的影响。表 2 列出不同维数 n 下的近似最坏误差。

表 2　不同维数的近似最坏误差

Table 2　Approximate worst-case error for different dimensions

维数 n	$\|\boldsymbol{b}\|_\infty$	测试次数	近似最坏误差
64	10^{11}	2000	8×10^{-4}
128	10^{11}	2000	1.3×10^{-3}
256	10^{11}	2000	8×10^{-3}
512	10^{11}	2000	8×10^{-2}
1024	10^{11}	2000	1.5×10^{-1}

本文的维数 $n=256$，实验证明 $\|b\|_\infty\leqslant10^{11}$ 且 $n=256$ 时误差很小，四舍五入后对结果不产生影响，复数域上的快速傅里叶变换求多项式乘法的稳定性很好。

扩大维数 n 与范数 $\|b\|_\infty$ 时产生误差如表 3 所列。

表 3　扩大维数与范数的近似最坏误差

Table 3　Approximate worst error of expanded dimension and norm

维数 n	$\|\boldsymbol{b}\|_\infty$	测试次数	近似最坏误差
256	10^{12}	2000	1×10^{-1}
512	10^{12}	2000	5×10^{-1}
256	10^{13}	2000	9×10^{-1}
512	10^{13}	2000	5×10^{0}

从表 3 可以看出，当 $\|\boldsymbol{b}\|_\infty$ 扩大为 10^{12}，维数 $n=256$ 时，产生的误差会大于 0.5，导致四舍五入的结果出错，这时如果要降低误差必须提升数据的精度。

5　实验结果

本文的实验环境：CPU 为 Intel Core i5-3470，频率为 3.20 GHz，内存为 8 GB 的计算机，操作系统为 CentOS7 64 位，算法使用 C++编程实现。

本节从基于的困难问题、加密效率、密文膨胀率 3 个方面对比了同为抗量子攻击的加密算法的 Frodo 算法[18]、NewHope 算法[19]、Kyber 算法[13]和本文加密算法。本文选取大量不同大小以及类型的文件，通过 C++将这些随机从网上选取的文件转化为二进制串，并将这些二进制串作为各加密算法的明文进行实验。表 4 给出的加密效率是各个算法平均加密每 256 比特数据所消耗的时间（单位为 ms）。

表 4　方案性能对比

Table 4　Comparison of scheme performance

算法	困难问题	加密效率/ms	密文膨胀率
Frodo 算法[18]	LWE	1.2～2.3	大于1∶100
NewHope 算法[19]	RLWE	0.11	1∶28 左右
Kyber 算法[13]	MLWE	0.07	1∶25 左右
本文算法	MLWE	0.10	1∶4.25 左右

从表 4 可以看出，本文算法的加效率与密文膨胀率均优于基于 LWE 问题构造的 Frodo 算法[18]，并且在加密效率与 NewHope 算法[19]和 Kyber 算法[13]差距不大的情况下，密文膨胀率得到了极大的提升。

本算法将模数 q 由原算法的 7 681 扩大到 4 294 967 291，如果直接使用原算法中的有限域上的多项式乘法实现本文算法（使用 NTL 大整数计算库），加密时间约为 1 ms。使用基于浮点运算的复数域上的快速傅里叶变换进行多项式乘法有效避免了 q 增大后导致的加密效率大幅度降低的问题。本文算法还可以采用 Kyber[13]原算法所提到的 AVX2 并行计算方法来进一步提高效率。

结束语　随着量子计算机的发展，后量子密码体制的需求与日俱增。本文对基于格困难问题 MLWE 的密钥封装算法 Kyber 进行改进，扩大了 Kyber 方案的明文空间，降低了密文膨胀率。通过误差可控的浮点复数域多项式乘法保证了加密的高效性，构建出能够用于一般公钥加密的新方案，并将其与已有的基于格的抗量子攻击加密算法进行了对比，结果表

明性能有一定提升。本文给出了噪声与密文膨胀率的关系式,其可以作为基于 MLWE 的加密算法的其他研究(如同态加密)的参考。

参 考 文 献

[1] SHOR P W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer[J]. SIAM Review,1999,41(2):303-332.

[2] GROVER L K. A fast quantum mechanical algorithm for database search[C]// Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing. ACM,1996:212-219.

[3] NSA Suite B cryptography:Cryptography today[OL]. https://www.nsa.gov/ia/programs/suiteb_cryptography/. 2015.

[4] CHEN L,CHEN L,JORDAN S,et al. Report on post-quantum cryptography[M]. US Department of Commerce,National Institute of Standards and Technology,2016.

[5] AJTAI M. Generating hard instances of lattice problems[C]// Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing. ACM,1996:99-108.

[6] GENTRY C,PEIKERT C,VAIKUNTANATHAN V. How to Use a Short Basis:Trapdoors for Hard Lattices and New Cryptographic Constructions[OL]. http://www.cs.toronto.edu/~vinodv/GentryPeikertVSTOC2008.pdf. 2008.

[7] AJTAI M,DWORK C. A public-key cryptosystem with worst-case/average-case equivalence[C]// Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing. ACM,1997:284-293.

[8] REGEV O. On lattices,learning with errors,random linear codes,and cryptography[J]. Journal of the ACM,2009,56(6):34.

[9] LYUBASHEVSKY V,PEIKERT C,REGEV O. On ideal lattices and learning with errors over rings[C]// Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer,Berlin,Heidelberg,2010:1-23.

[10] WANG X Y,LIU M J. Survey of lattice-based cryptography[J]. Journal of Cryptologic Research,2014,1(1):13-27. (in Chinese)
王小云,刘明洁. 格密码学研究[J]. 密码学报,2014,1(1):13-27.

[11] LI Z P,MA C G,ZHOU H S. Overview on Fully Homomorphic Encryption[J]. Journal of Cryptologic Research,2017,4(6):561-578. (in Chinese)
李增鹏,马春光,周红生. 全同态加密研究[J]. 密码学报,2017,4(6):561-578.

[12] LANGLOIS A,STEHLÉ D. Worst-case to average-case reductions for module lattices[J]. Designs,Codes and Cryptography,2015,75(3):565-599.

[13] BOS J,DUCAS L,KILTZ E,et al. CRYSTALS-Kyber:a CCA-secure module-lattice-based KEM[C]// 2018 IEEE European Symposium on Security and Privacy. 2018.

[14] LIU YM,LI XX,LIU H L. Post-quantum key exchange from lattice[J]. Journal of Cryptologic Research,2017,4(5):485-497. (in Chinese)
刘亚敏,李祥学,刘晗林. 基于格的后量子密钥交换研究[J]. 密码学报,2017,4(5):485-497.

[15] BAI S,LANGLOIS A,LEPOINT T,et al. Improved security proofs in lattice-based cryptography:using the Rényi divergence rather than the statistical distance[C]// International Conference on the Theory and Application of Cryptology and Information Security. Springer,Berlin,Heidelberg,2015:3-24.

[16] GOLUB G H,VAN LOAN C F. Matrix computations[M]. JHU Press,2012:219-222.

[17] HIGHAM N J. Accuracy and stability of numerical algorithms[M]. Society for Industrial and Applied Mathematics,2002:451-458.

[18] BOS J,COSTELLO C,DUCAS L,et al. Frodo:Take off the ring! practical,quantum-secure key exchange from LWE[C]// Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM,2016:1006-1018.

[19] ERDEM A,LÉO D,THOMAS P,et al. Post-quantum Key Exchange-A New Hope[C]// USENIX Security Symposium. 2016.