LWE 问题实际安全性分析综述

毕 蕾[1,2], 李帅钢[1,2], 刘亚敏[1,2], 张 江[3], 范淑琴[3]

[1] 中国科学院信息工程研究所, 北京 中国 100049
[2] 中国科学院大学网络空间安全学院, 北京 中国 100049
[3] 密码科学技术国家重点实验室, 北京 中国 100878

摘要 LWE 问题被广泛用于设计安全的格上密码方案。为了评估基于 LWE 的格密码方案在给定具体参数下的安全强度, 我们需要研究目前求解 LWE 问题算法的复杂度。本文以 Albrecht 等人[33]2015 年的研究工作为基础, 概述了求解 LWE 问题的主流算法及其复杂度, 并给出了针对具体 LWE 实例的评估结果。

关键词 格; 带错误的学习问题; 安全性分析
中图法分类号 TP309.7DOI 号 10.19363/J.cnki.cn10-1380/tn.2019.03.01

A Survey on the Analysis of the Concrete Hardness of LWE

BI Lei[1,2], LI Shuaigang[1,2], LIU Yamin[1,2], ZHANG Jiang[3], FAN Shuqin[3]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100049, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
[3]State Key Laboratory of Cryptology, Beijing 100878, China

Abstract The Learning with errors (LWE) problem has been widely used in designing secure lattice-based cryptosystems. In order to assess the concrete security of LWE-based schemeswhen given the parameters, we need to investigate the current algorithms which can be used to solve LWE problem and their actual complexity. In this paper, we give a brief survey on the main LWE solving algorithms and their complexity models, based on the survey of Martin R. Albrecht et al. in the year 2015[33]. We also give some estimation results on concrete LWE instances.

Key words Lattice; learning with errors (LWE) problem; security analysis

1 引言

格理论是几何中的经典研究课题, 它的起源可以追溯到十七世纪对于球堆积问题的研究, 在计算数论、组合优化、信息编码等领域中都有广泛的应用。格理论最初作为一种分析工具被引入密码学中, 曾被用于分析背包密码体制、RSA 密码体制[1-2]等。

1996 年, Ajtai 开创性地给出了格中唯一最短向量问题(unique shortest vector problem, uSVP)最坏情况(worst-case)下到小整数解(Small Integer Solutions, SIS)问题在平均情况(average-case)下的归约证明[3], 即这些困难问题在最坏情况下的困难性可以归约到一类随机格中问题的困难性, 因此基于格的密码体制可以提供最坏情况下的可证明安全性。这一证明是理论密码学领域的里程碑, 它说明了基于平均情况困难问题构造的密码方案的安全性可以由最坏情况困难问题的困难性来保障。1997 年, Ajtai 和 Dwork 首次将其作为一种设计工具, 构造了第一个基于格的公钥密码体制 Ajtai-Dwork[4], 该方案的安全性依赖于格上最近向量问题(Closest Vector Problem, CVP)的困难性。此后, 基于格的密码体制相继出现, 例如 NTRU 密码体制[5]、GGH 密码体制[6]等。

初期的格密码方案由于存在密钥尺寸过大或者缺乏严格的安全性证明等缺陷, 无法满足实际应用的需求。直到 2005 年, 格密码理论取得突破性进展——Regev 提出了基于带错误的学习(Learning with

通讯作者: 毕蕾, 硕士, Email:bilei@iie.ac.cn。
本课题得到国家自然科学基金(No. 61772515)资助。
收稿日期: 2018-12-10; 修改日期: 2019-02-15; 定稿日期: 2019-02-26

errors, LWE)问题的公钥加密算法[7], 大幅度缩小了密文和密钥尺寸, 同时又将加密算法的安全性归约到了格上最坏情况困难问题的难解性——在量子归约下, 它至少与 worst-case 下的近似因子为$Q(\frac{n}{_})$的 SVP 的变体一样困难。LWE 问题的这些优势, 吸引了众多密码学研究者对其进行深入研究, LWE 问题在公钥密码方案设计中, 如密钥协商、签名、加密, 均得到了广泛的应用。例如基于 LWE 问题构造的基于身份的加密(Identity-Based Encryption, IBE)[8-9]、密钥相关消息(Key-Dependent Message, KDM)安全的加密[10], 以及全同态加密(Full Homomorphic Encryption, FHE)[11]等。

基于 LWE 设计的密码方案范围广泛, 设计方法和参数选择也相差很大。若方案的安全性由 LWE 问题的困难性来保障, 则攻破 LWE 即可攻破方案。因此, 研究基于 LWE 的方案在当前参数设置下的安全性级别和在未来设计新方案选定参数时, 均需考虑 LWE 问题本身的困难性, 只有这样, 才能使得这些方案在保证一定的安全性强度的情况下, 效率尽可能地高。这使得对 LWE 的困难性的分析成为一个重要问题。

通常, 对于困难问题的分析方式主要分为两种, 第一种是渐进式的分析方法, 这种分析方法将对数或者常数因子隐藏在表达式之内, 这样有助于宏观地理解和比较不同类型算法的表现; 第二种是利用当前技术条件下解决这些问题最快的算法来实际评估这些问题的安全性强度, 在设计实际方案时, 一般需采用这种分析方法来确定方案在目标安全强度下的参数设置。目前大多数对于 LWE 问题的研究与分析, 都采用了渐进式的分析方法。因此, 分析现有算法在求解 LWE 问题的表现和性能, 以此来评估和分析基于 LWE 的方案构造的实际安全性强度具有十分重要的理论意义和应用价值。

2015 年, Albrecht、Player 和 Scott 已经对这一问题进行了较为详尽的研究和综述[33], 本文以该综述为蓝本, 同时又对后续一系列研究成果进行整理和总结, 对研究 LWE 问题时涉及的格上困难问题、格基约化算法、解决 LWE 问题的不同策略和方法进行了概述, 并给出了基于现有求解算法的两种不同评估程序[33,41]对于 LWE 困难性的评估结果。

2 预备知识

2.1 格

定义 1(格). 已知$\mathbb{R}^m$中的n个线性无关的行向量$\boldsymbol{b}_1,\cdots,\boldsymbol{b}_n(m\geq n)$, 这组向量的所有整系数线性组合的集合称为格_, 记为

$$\Lambda=L(\boldsymbol{b}_1,\cdots,\boldsymbol{b}_n)=\left\{\sum_{i=1}^{n}x_i\boldsymbol{b}_i,x_i\in\mathbb{Z}\right\},$$

线性无关的向量组$\boldsymbol{b}_1,\cdots,\boldsymbol{b}_n$称为格_的一组基, n为格的秩, m为格的维数。

定义 2(q-ary 格).给定矩阵$\boldsymbol{A}_\mathbb{Z}_q^{m_n}$, 定义$m$维的$q$-ary 格如下:

$$\Lambda_q(\boldsymbol{A}^T)=\{\boldsymbol{x}\in\mathbb{Z}^m\,|\exists\boldsymbol{y}\in\mathbb{Z}^n\ s.t.\ \boldsymbol{x}=\boldsymbol{y}\boldsymbol{A}^T\ mod\ q\},$$

$$\Lambda_q^{\perp}(\boldsymbol{A})=\{\boldsymbol{x}\in\mathbb{Z}^m\,|\boldsymbol{x}\boldsymbol{A}=0\ mod\ q\}。$$

2.2 格上困难问题

定义 3(最短向量问题, Shortest Vector Problem, SVP).给定$\mathbb{Z}^n$中秩为m的格_的一组基$\boldsymbol{B}$, 寻找一个非零向量$\boldsymbol{u}_$使得它满足$||\mathrm{u}||=\min_{\mathrm{v}_\backslash\{0\}}||\mathrm{v}||$。

类似地, 近似最短向量问题(γ-Shortest Vector Problem, γ-SVP)可以表述为给定格_的一组基$\boldsymbol{B}$, 寻找一个非零向量$\boldsymbol{u}_$使得它满足$\|u\|\leqslant\gamma\lambda_1(\Lambda)$, 其中$\gamma$称为近似因子, $\lambda_1(_)$是指格_中最短向量的长度。SVP 是格理论中最重要的困难问题之一。1998 年 Ajtai 证明了在随机归约下 2-范数下的 SVP 问题是 NP-hard 的[12]。对 1-范数和无穷范数的 SVP 问题目前最好的结果分别是在近似因子小于$2-\varepsilon$和$n^{\frac{c}{\log\log n}}$(c是任意正常数)的情况下是 NP-hard 的[14,15]。对γ-SVP 问题, Khot 在 2005 年证明了在随机归约下, p-范数($1<p<\_$)的常数近似因子的γ-SVP 问题是 NP-hard 的[13]。

同样可以定义θ-唯一最短向量问题(θ-unique Shortest Vector Problem, θ-uSVP): 当格_满足$\lambda_2(_)>\theta\lambda_1(_)$时, 寻找一个非零向量$\boldsymbol{u}_$使得它满足$||\boldsymbol{u}||=\min_{\boldsymbol{v}_\backslash\{0\}}||\boldsymbol{v}||$。

定义 4(α-有界距离解码问题, α-Bounded Distance Decoding, α-BDD). 给定$\mathbb{Z}^n$中秩为n的格_的一组基$\boldsymbol{B}$, 一个距离参数$\alpha>0$和一个目标向量$\boldsymbol{x}_pan(_)$, 其中$dist(\boldsymbol{x},_)<\alpha\lambda_1(_)$, 寻找一个向量$\boldsymbol{u}_$使得它满足$||\boldsymbol{u}-\boldsymbol{x}||=dist(\boldsymbol{x},_)$。

当γ满足$\gamma(n)=n^{O(1)}$时, 在多项式时间内, $\frac{1}{2\gamma}$-BDD 问题可以归约到γ-uSVP 问题, 反之, γ-uSVP 问题也可以归约到$\frac{1}{\gamma}$-BDD 问题[16]。

定义 5(LWE 问题[7]). 给定正整数n和$q,\boldsymbol{s}$是$\mathbb{Z}_q^n$中的向量。从$\mathbb{Z}_q^n$中均匀随机地抽取向量$\boldsymbol{a}$, 从标准差为$_=\frac{\alpha q}{\sqrt{2_}}$($0<\alpha<1$表示错误率)的高斯分布$_=D_{\mathbb{Z},_^2}$中随机抽取整数$e$, 计算$(\boldsymbol{a},c)=(\boldsymbol{a},<\boldsymbol{a},\boldsymbol{s}>$

$+e)\_\mathbb{Z}_q^n\_\mathbb{Z}_q$, 并用$L_{s,_}$表示$(\boldsymbol{a},c)$在$\mathbb{Z}_q^n_\mathbb{Z}_q$上的概率分布。

判定版本的 LWE 问题可描述为: 判断一组随机实例$(\boldsymbol{a},c)_\mathbb{Z}_q^n_\mathbb{Z}_q$是来自分布$L_{s,_}$还是来自于$\mathbb{Z}_q^n_\mathbb{Z}_q$上均匀分布。

计算版本的 LWE 问题可描述为: 根据来自分布$L_{s,\chi}$的一组实例$(\boldsymbol{a},c)_\mathbb{Z}_q^n_\mathbb{Z}_q$, 计算$\boldsymbol{s}$。

一般将一组 LWE 的实例写作矩阵的形式$(\mathrm{A},\mathrm{c})=(\mathrm{A},\mathrm{As}+\mathrm{e})_\mathrm{Z_q^{m_n}}_\mathrm{Z_q^m}$, 其中的每一行是一个服从分布$L_{s,_}$选取的 LWE 实例$(\boldsymbol{a},c)=(\boldsymbol{a},<\boldsymbol{a},\boldsymbol{s}>+e)_\mathbb{Z}_q^n_\mathbb{Z}_q$, m是实例的个数。

LWE 问题是 2005 年 Regev 提出的[7], 在提出的同时, Regev 也证明了在量子归约下, 它至少与 worst-case 下的近似因子为$O(\frac{n}{_})$的 SVP 的变体一样困难。为了得到这个归约结果, Regev 首先将 BDD 问题量子归约到了 LWE 问题, 而实际上, LWE 问题等价于$_q(\boldsymbol{A^T})$上的 BDD 问题, 只是二者在错误的抽样方法上稍有不同, LWE 问题的错误抽样使用的是离散高斯分布。2009 年, Peikert[17]利用 LWE 问题和 BDD 问题的等价性得到了 GapSVP 到 LWE 问题的经典归约, 但这个归约还存在一些问题, 例如需要模数q是维数n的指数级。2013 年, Brakerski 等人[18]将模数q缩小到了n的多项式级别, 证明了解决这一模数下的 LWE 问题的困难性不低于解决近似因子为$\sqrt{n}$的 SVP 的困难性。

定义 6(小整数解问题, Small Integer Solutions, SIS[3])给定模数q_1, 实常数v和矩阵$\boldsymbol{A}_\mathbb{Z}_q^{n_m}$, 其中$m_n>0$。寻找一个非零向量$\boldsymbol{u}_\mathbb{Z}^m$使得$\mathrm{uA}=0 \bmod \mathrm{q}$, 并且$||\boldsymbol{u}||_v$。

Ajtai 首先给出了 SIS 问题的雏形, 随后 Micciancio 和 Regev 第一次明确提出 SIS 问题[19], 并对 SIVP 和 GapSVP 到 SIS 问题的归约进行了讨论。根据定义, SIS 问题可以平凡地归约到 SVP, 这是因为格$_\bar{q}(\boldsymbol{A})$上 SVP 的解是由矩阵$\boldsymbol{A}$生成的格上 SIS 问题的解。

从 LWE 问题到 SIS 问题的归约最早是由 Micciancio 和 Regev[20]在 2006 年提出的。从定义可以看出, LWE 对偶格中的短向量问题即为 SIS 问题, 因此 LWE 对偶格中的短向量可以用来解决判定版本的 LWE 问题。具体来说, 当对偶格中短向量长度不超过 3/2 时, LWE 问题可以归约到 SIS 问题, 相对应的 SIS 问题到 LWE 问题的归约结果由 Stehlé 等人[21]在 2009 年得到。

2.3 格基与高斯启发式

对于一组给定的格基$\boldsymbol{B}=(\boldsymbol{b_1},_,\boldsymbol{b_n})$, 它的 Gram-Schmidt 正交化基表示为$\boldsymbol{B}^*=(\boldsymbol{b_1^*},_,\boldsymbol{b_n^*})$, 其中

$$\boldsymbol{b}_i^*=\boldsymbol{b}_i-\sum_{j=1}^{i-1}\mu_{ij}\boldsymbol{b}_j^*(1\leqslant j<i\leqslant n),\mu ij\frac{\left\langle \boldsymbol{b}_i,\boldsymbol{b}_j^*\right\rangle}{\left\|\boldsymbol{b}_j^*\right\|^2}。$$

为了衡量格基约化算法输出基的质量, Gama 和 Nguyen 引入了 Hermite 因子的概念[25]: 对于一个m维的格_, 利用格基约化算法对其进行约化, 设输出基为$\{\boldsymbol{b_1},_,\boldsymbol{b_m}\}$, 其中$\boldsymbol{b_1}$为这组基中最短向量, 那么_的 Hermite 因子定义为

$$\delta_0^m=\frac{||\boldsymbol{b}_1||}{vol(\Lambda)^{\frac{1}{m}}},$$

其中$vol(_)$表示格_的体积。

高斯启发式(Gaussian Heuristic)意为对于一个连续的集合$S_\mathbb{R}^m, S__$中的格点数约为$\frac{vol(S)}{vol(L)}$, 其中$vol(L)$为格L的体积。据此, 可以得到对于格中最短向量长度$\lambda_1(L)$的估计:

$$GH(L)=\frac{vol(L)^{\frac{1}{m}}}{V_m(1)^{\frac{1}{m}}}\approx\sqrt{\frac{m}{2\pi e}}\,vol(L)^{\frac{1}{m}},$$

其中$V_m(1)$表示一个半径为1的m维球的体积。

3 格基约化算法

非正式地说, 格基约化算法的主要目的是将任意给定的一组格基转化为一组正交性更好的基, 并使得这个基中的各个向量尽可能地短。大多数基于格的密码体制的安全性, 都依赖格基约化算法解决格上困难问题的困难性。

1982 年, Lenstra、Lenstra 和 Lovasz[22]提出了著名的 LLL 算法, 为解决格理论中近似最短向量问题带来了十分重大的突破。1994 年, Schnorr 和 Euchner[23]提出了 BKZ 格基约化算法, 2011 年 Chen 和 Nguyen[24]将其优化为 2.0 版本, 是目前最实用、应用最广泛的格基约化算法。

3.1 LLL 算法

LLL 算法可以看作是二维格上高斯约化算法在更高维数格上的扩展。LLL 算法可被用于求解一些 NP 问题的近似解, 因此在理论计算机科学中有着广泛的应用。在密码分析领域, LLL 算法对于公钥密码系统安全性的评估起了十分重要的作用, 例如用于对于 RSA 密码算法的分析[55]等。

LLL 算法是一个多项式时间算法[22], 理论上, LLL 算法的 Hermite 因子可以达到$\delta_0=\left(\frac{4}{3}\right)^{\frac{n-1}{4n}}_1.075$, 但 LLL 算法实际执行效果比理论结果好很多, 文献[27]中给出 LLL 算法在实际应用中可以达到 Hermite

因子$\delta_0 = 1.0219$。

3.2 BKZ 算法

BKZ 算法[23]需调用 LLL 算法和一个求解低维格上 SVP 问题的算法作为子程序, 这个 SVP 求解算法在实际应用中通常选用枚举算法或者筛法来实现(下文将其称为一个 SVP oracle)。

在 BKZ 算法中, 有一个十分重要的参数——分组长度β, 它的选择与算法效率息息相关。直观地, 选择的分组长度β越大, 算法输出约化基质量越好, 但同时算法执行时间也会越长。

BKZ 算法执行过程如下: 输入一组 LLL 约化基$\{\boldsymbol{b_1}, _, \boldsymbol{b_m}\}$, 在$\{\boldsymbol{b_1}, _, \boldsymbol{b_\beta}\}$张成的子格中调用 SVP oracle 得到这个子格中的一个短向量$\boldsymbol{b_1'}$, 用$\{\boldsymbol{b_1}, _, \boldsymbol{b_\beta}, \boldsymbol{b_1'}\}$作为输入调用 LLL 算法得到一组新的$\beta$个线性无关向量用于替代格基中的前$\beta$个向量; 接着, 用$\{\boldsymbol{b_2}, _, \boldsymbol{b_{\beta+1}}\}$在$< \boldsymbol{b_1} >$-(由$\boldsymbol{b_1}$张成空间的垂直空间)中的投影调用 SVP oracle 得到短向量$\boldsymbol{b_2'}$, 并执行和上一步相同的 LLL 和替换工作; 重复这一过程直到这组基的最后一个向量。其中前 $m-\beta+1$ 组在执行中分组长度为β, 在此之后的分组长度每次比之前减一。将如上操作执行一轮输出的更新之后的基当作输入, 继续执行同样的下一轮操作, 直到得到的基不再发生变化, 则算法执行结束, 并将此时得到的基称为一组 BKZ 约化基输出。

对于一组 BKZ 约化基, 有如下假设成立: 一组 BKZ 约化基 Gram-Schmidt 正交化之后的向量的范数满足$\left|\left|\boldsymbol{b_i^*}\right|\right|_\alpha^{i-1}||\boldsymbol{b_1}||(0<\alpha<1)$。这一假设称为 GS 假设(Geometric Series Assumption, GSA)[28]。由 Hermite 因子的定义可知$\left|\left|\boldsymbol{b_1}\right|\right| = \delta_0^m_vol(_)^{\frac{1}{m}}$, 另外显然有$vol(_) = _{i=1}^{m}||\boldsymbol{b_i^*}||$, 据此可计算 GSA 中的参数$\alpha = \delta_0^{-\frac{2m}{m-1}}_\delta_0^{-2}$。

一般来说, 可粗略地认为对于一组 BKZ 约化基, GSA 都是成立的, 但更精细地, 文献[29]中将 GSA 用β表示为$\left|\left|\boldsymbol{b_i^*}\right|\right|_\beta^{-\frac{i}{\beta}}||\boldsymbol{b_1}||$, 证明了该关系式仅对前$m-\beta$个 Gram-Schmidt 正交化向量成立, 而对于后面的β个 Gram-Schmidt 正交化向量, 则满足关系式$\left|\left|\boldsymbol{b_i^*}\right|\right|_e^{-\frac{1}{4}\log(m-i)^2}$, 可见, 随着下标的增长$\left|\left|\boldsymbol{b_i^*}\right|\right|$减小的速度比 GSA 中更快[24]。

2011 年, Chen 和 Nguyen 对其进行了四点改进, 得到 BKZ2.0 算法[24]:

(1) 提前终止算法: 初始 BKZ 算法执行直至得到的基不能再优化, BKZ2.0 算法执行有限轮就提前终止。实验证明, 执行一定轮数后就提前终止算法得到的约化基的质量与继续执行下去相差不大。

(2)~(4)三个改进均旨在降低枚举子程序的时间复杂度。

(2) 合理的修剪: 枚举算法的整个过程可以看作一个在树状结构上进行搜索, “修剪”即意味着通过某些策略选择剪去树的某些分支, 以此来提高搜索效率。

(3) 优化输入基的质量: 初始 BKZ 算法在每次调用 SVP oracle 前对部分基进行 LLL 约化, BKZ2.0 算法中利用比原算法中更小分组长度的 BKZ 算法对部分基进行约化, 提高了 SVP oracle 输入基的质量。

(4) 减小枚举算法的搜索半径: 初始 BKZ 算法在每一次调用 SVP oracle 时的搜索半径不变, 但其实随着算法的进行可搜索的范围在逐渐减小, BKZ2.0 算法对此进行了改进, 在每一步调用 SVP oracle 前会对搜索半径进行重新选择优化。

对于一个 BKZ 算法, 可通过如下两个参数分别对它的输出基的质量和算法运行时间进行评估:

(1) Hermite 因子: Hermite 因子可以刻画输出基的质量。Chen 在文献[31]中给出了分组长度为β的 BKZ 约化基的δ_0的下界: 在高斯启发式和 GSA 成立的条件下, 执行分组长度为β的 BKZ 约化算法得到的约化基的 Hermite 因子的下界为$\delta_0 = (\frac{(_\beta)^{\frac{1}{\beta}}\beta}{2_e})^{\frac{1}{2(\beta-1)}}$, 下文中对于$\delta_0$和$\beta$关系的转换均适用该等式。对于$\delta_0$有一个经验估计式$\delta_0 = \beta^{\frac{1}{2\beta}}$, 对此, Stehlé 又给出了一个更精简的近似表达式$\delta_0 = 2^{\frac{1}{\beta}}$[32]。三者之间的关系如下图 1 所示[33]。

(2) 分组长度β: BKZ 算法在调用子程序 SVP oracle 时的分组长度为β。在经过多项式轮的 SVP oracle 调用后, BKZ 算法输出基中的最短向量(一般为$\boldsymbol{b_1}$)满足$\left|\left|\boldsymbol{b_1}\right|\right|_(\beta)^{\frac{m}{2\beta}}_vol(_)^{\frac{1}{m}}$。因此 BKZ 的运行时间可以表示为$T_{BKZ} = poly(m)_T_{SVP}(\beta)$, 其中$T_{SVP}(\beta)$是在$\beta$维格上调用一次 SVP oracle 所需的时间[34]。目前 BKZ 算法的运行时间关于分组长度β是指数级别的, 从最初文献[23]中的$2^{O(\beta^2)}$降到现在的$2^{O(\beta)}$[34]。

Chen 和 Nguyen 在提出 BKZ2.0 算法的同时, 还给出了一个模拟程序, 用来估计 BKZ2.0 算法的实际执行结果。2016 年, Aono、Wang 等人对 BKZ 算法中分组长度β的取值方式进行了更细致的分析, 提出了渐进式 BKZ(progressive-BKZ)算法[35], 这类 BKZ 算法中, 分组长度β的值不再从始至终保持不变, 而是呈现出从小到大的变化形式, 选择这一列β的原则是

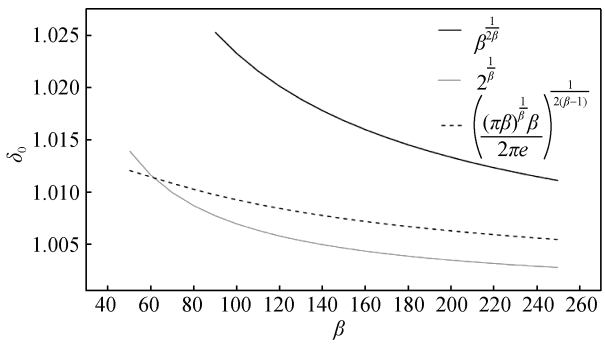

图 1　分组长度为β的 BKZ 算法的δ_0的估计表达式

Figure 1　the Estimates for δ_0 for BKZ-β

使得算法整体时间复杂度最低。与 BKZ2.0 相比, 渐进式 BKZ 算法在解决最高 160 维的 SVP 挑战(SVP challenge)[36]时比 BKZ2.0 最多可以快约 50 倍[35]。同时, Aono、Wang 等人观察到 BKZ 约化基 Gram-Schmidt 范数曲线相对于 GSA 曲线的“头部凹陷”现象并提出了分析和利用它提升 BKZ 效率的方法并出了一个新的模拟程序。2017 年, Yu 和 Ducas[56]进行了大量实验对 BKZ 的实际行为进行了评估, 对于BKZ约化基Gram-Schmidt正交化后范数的分布进行了更为细致地研究, 并且通过观察结果更准确地量化了“头部凹陷”现象。2018 年, Bai、Stehle 和 Wen[57]通过考虑随机格中短向量的分布又提出了一个更为精确的 BKZ 模拟程序。

除了 LLL 和 BKZ(以及 BKZ2.0)之外, 还有一类特殊的格基约化算法: 滑动约化(slide reduction)[25]和对偶 BKZ(dual-BKZ)[37], 这类算法在理论分析上比 BKZ 算法更优: BKZ 理论上可以输出的最短向量满足$\|\boldsymbol{b}_1\| \leqslant 2\gamma_k^{\frac{n-1}{2(k-1)}+\frac{3}{2}} vol(L)^{\frac{1}{n}}$, 而滑动约化和对偶 BKZ 算法的上界可以减小到$\gamma_k^{\frac{n-1}{2(k-1)}} vol(L)^{\frac{1}{n}}$, 其中$\gamma_k$称为 Hermite 常量, 用来表示一个$k$维格$L$的$\left(\frac{\lambda_1(L)}{vol(L)^{\frac{1}{k}}}\right)^2$的上界。可以看出后者的理论值略小于 BKZ, 但在实际应用中, 对于相同的分组长度β, BKZ 算法的实际输出基质量要优于这类格基约化算法。

在现有文献中一般使用 BKZ 算法进行安全性评估时, 使用筛法作为 SVP oracle, 在β维格中调用一次 SVP oracle 的时间估计为$2^{c\beta+o(\beta)}$, 其中经典情形下$c=0.292$[38], 在加入 Grover 量子搜索算法后这一参数降为$c=0.265$[39]。根据文献[40]中的实验结果, 一些研究者将时间成本估计式中的$o(\beta)$省略或者替换为常数16.4。另外, 在文献[40]中还对空间和时间综合考虑, 在空间复杂度最低的情况下, 成本模型中的参数$c=0.3366, o(\beta)$替换为常数12.31。在进行安全性评估时, 根据考虑的 SVP oracle 调用次数, 可将其分为三种模型: Core-SVP 模型[41]、β-SVP 模型(例如文献[42])和$8m$-SVP 模型(例如文献[43])。

表 1　成本模型

Table 1　Cost Model

模型		成本
Core-SVP	Core-SVP	$2^{0.292\beta+16.4}$
	Core-SVP(min space)	$2^{0.3366\beta+12.31}$
	Q-Core-SVP	$2^{0.265\beta+16.4}$
β-SVP	β-SVP	$\beta\_2^{0.292\beta+16.4}$
	Q-β-SVP	$\beta\_2^{0.265\beta+16.4}$
$8m$-SVP	$8m$-SVP	$8m\_2^{0.292\beta+16.4}$
	Q-$8m$-SVP	$8m\_2^{0.265\beta+16.4}$

(注: 每种评估模型中的第一个子类, 例如 Core-SVP 表示经典情形下的模型; 第二个子类, 例如 Q-Core-SVP 表示加入 Grove 量子搜索后的情形)

4　分析策略及方法

目前, 对于 LWE 问题的分析方法主要分为以下三种策略: SIS 策略、BDD 策略和直接求解(direct)策略(见图 2)。

其中 SIS 策略的基本思想是通过找到对偶格上的短向量(即求解对偶格上的 SIS 问题), 再利用它来解决原格上的 decision-LWE 问题。具体地, 对于给定服从分布$L_{s,\chi}$选取的m个 LWE 实例$(\boldsymbol{A}, \boldsymbol{c})_\mathbb{Z}_q^{m_n}_\mathbb{Z}_q^m$, 判断它是服从分布$L_{s,\chi}$还是均匀分布。为此, 可以在由$\boldsymbol{A}$生成的对偶格$\Lambda_q^{\perp}(\boldsymbol{A}) = \{\boldsymbol{x}_\mathbb{Z}^m | \boldsymbol{xA} = \boldsymbol{0}\ mod\ q\}$上找到一个短向量$\boldsymbol{v}$, 此时得到的$\boldsymbol{v}$满足条件$\boldsymbol{vA} = \boldsymbol{0}$。考虑$< v, c >$的分布: 如果$\boldsymbol{c} = \boldsymbol{As} + \boldsymbol{e}$那么有$< v, c >= \boldsymbol{vAs} + < \boldsymbol{v}, \boldsymbol{e} >=< v, e >$, 由于$\boldsymbol{e}$是服从高斯分布的, 所以$< \boldsymbol{v}, \boldsymbol{c} >$也服从$\mathbb{Z}_q$上的高斯分布(通常, $\boldsymbol{v}$是一个短向量, 而$\boldsymbol{e}$是一个小错误, 所以$< \boldsymbol{v}, \boldsymbol{c} >$也是一个比较小的数字); 如果$\boldsymbol{c}$是均匀随机选择的, 那么$< \boldsymbol{v}, \boldsymbol{c} >$在$\mathbb{Z}_q$上也是均匀随机的。所以利用求解 SIS 问题得到的向量$\boldsymbol{v}$可以区分这两种情况, 这也就意味着成功解决了 decision-LWE 问题。

BDD 策略的基本思想是将解决 search-LWE 问题转化为解决对应q-ary 格上的 BDD 问题。直观地, 给定服从分布$L_{s,\chi}$选取的m个 LWE 实例$(\boldsymbol{A}, \boldsymbol{c} = \boldsymbol{As} + \boldsymbol{e})_\mathbb{Z}_q^{m_n}\ _\mathbb{Z}_q^m$, 由于错误$\boldsymbol{e}$是服从高斯分布的, 所以几乎所有错误都落在三倍标准差(即$\frac{3\alpha q}{\sqrt{2\pi}}$)范围内, 因而向量$\boldsymbol{c}$是格点$\boldsymbol{As}$附近的一个点, 所以可将这样的 LWE 实例视作一个 BDD 实例。

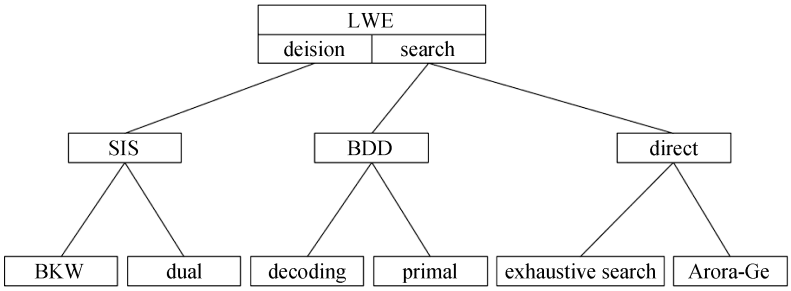

图 2　LWE 分析策略及方法

Figure 2　The Analysis Strategies and Methods of LWE

对于不同策略下求解 LWE 问题的具体方法又可分为四个类别: 组合方法、代数方法、基于格的方法和穷搜(exhaustive search)方法。其中, 组合方法是指 BKW 算法[44], 该算法可看作一种扩展的高斯消元法, 一次处理多个元素; 代数方法是指 Arora-Ge 算法[45], 该方法是一种线性化技术, 算法关于维数是指数的, 对于错误向量较小的情况可以降到亚指数级别。由于 BKW 算法要求实例数量为指数级别、Arora-Ge 算法、穷搜方法的时间复杂度过高, 三者均不适宜直接在实际应用中使用, 故本章主要叙述基于格的分析方法。

基于格的方法共有三种: dual 方法、decoding 方法和 primal 方法, 其中 dual 方法基于 SIS 策略对 decision-LWE 问题进行求解, 即先解决对偶格上的 SIS 问题, 再用得到的解来求解原格上的 LWE 问题; decoding 方法和 primal 方法基于 BDD 策略, 将 LWE 问题实例视作 BDD 问题实例, 再用不同方法对其进行求解。

下面对这三种方法进行详细叙述。

4.1　Dual 方法

Dual 方法通过 SIS 策略对 decision-LWE 问题进行求解, 该方法的基本思想是考虑由$\boldsymbol{A}$生成的对偶格$\Lambda_q^{\perp}(\boldsymbol{A})=\{\boldsymbol{x}\in\mathbb{Z}^m|\boldsymbol{xA}=\boldsymbol{0}\ mod\ q\}$, 使用格基约化算法在$\Lambda_q^{\perp}(\boldsymbol{A})$中寻找短向量, 此时得到的$\boldsymbol{v}$满足$\boldsymbol{vA}=\boldsymbol{0}$, 然后再继续利用它来对$<\boldsymbol{v},\boldsymbol{c}>$的分布进行判断, 从而解决 decision-LWE 问题。

如 SIS 策略所述, 向量$\boldsymbol{v}$的范数$||\boldsymbol{v}||$要足够小, 如果$||\boldsymbol{v}||$太大, 则$<\boldsymbol{v},\boldsymbol{e}>$的高斯分布与均匀分布无法区分:

定理 1[46]. 给定参数为n,q,α的 LWE 实例和一个对偶格$\Lambda_q^{\perp}(\boldsymbol{A})=\{\boldsymbol{x}\in\mathbb{Z}^m|\boldsymbol{xA}=\boldsymbol{0}\ mod\ q\}$中的向量$\boldsymbol{v}$, 区分$<\boldsymbol{v},\boldsymbol{e}>$和一个$\mathbb{Z}_q$中的随机元素的优势约为$e^{-\pi(||v||)^2}$。

显然, 向量$\boldsymbol{v}$的范数越小, 区分$<\boldsymbol{v},\boldsymbol{e}>$和均匀分布的优势就越大。换言之, 若想以成功率$\varepsilon$解决 decision-LWE 问题, 在对偶格上找到短向量$\boldsymbol{v}$的范数需满足$||\boldsymbol{v}||\leq\frac{1}{\alpha}\sqrt{\frac{\ln\frac{1}{\varepsilon}}{\pi}}$。

由于在 dual 方法中得到短向量$\boldsymbol{v}$一般采用格基约化算法, 因此可以建立起格基约化算法质量和解决 decision-LWE 问题成功率之间的关系:

定理 2[33]. 给定参数为n,q,α的 LWE 实例, 任意格基约化算法的 Hermite 因子满足$\log\delta_0=\frac{\log^2\left(\alpha\sqrt{\frac{\pi}{\ln\frac{1}{\varepsilon}}}\right)}{4n\log q}$时, 可以以概率$\varepsilon$区分出分布$L_{\boldsymbol{s},\chi}$。

可以看出, 若要成功率ε越大则需要$||\boldsymbol{v}||$越小, 而$||\boldsymbol{v}||$越小也就意味着用于得到$\boldsymbol{v}$的 BKZ 算法所需的分组长度β越大、执行时间越长。为了协调这个矛盾, 降低算法整体时间复杂度, 一般会先选择一个比较小的成功率ε, 然后将算法执行$\frac{1}{\varepsilon^2}$次, 由切诺夫界可知, 此时可以接近 1 的概率解决 decision-LWE 问题。

对于一种特殊形式的 LWE 问题——$\boldsymbol{s}$和$\boldsymbol{e}$均服从高斯分布, 文献[41]又考虑了另一种不同的格$\Lambda(\boldsymbol{A})=\{(\boldsymbol{x},\boldsymbol{y})\in\mathbb{Z}^m\times\mathbb{Z}^n|\boldsymbol{xA}=\boldsymbol{y}\ mod\ q\}$。这种形式的 LWE 问题称为 normal form LWE 问题, 普通形式的 LWE 问题可以以损失n个实例为代价, 转换为 normal form LWE 问题的形式[47]。

格$\Lambda(\boldsymbol{A})$的维度为$m+n$, 且有$vol(\Lambda(\boldsymbol{A}))=q^n$。在$\Lambda(\boldsymbol{A})$中利用格基约化算法找到一短向量$(\boldsymbol{v},\boldsymbol{w})$, 如果$(\boldsymbol{A},\boldsymbol{c})\in\mathbb{Z}_q^{m\times n}\times\mathbb{Z}_q^m$服从分布$L_{\boldsymbol{s},\chi}$, 那么有

$$\begin{aligned}<\boldsymbol{v},\boldsymbol{c}>&=v\in(\boldsymbol{As}+\boldsymbol{e})\\&=\boldsymbol{vAs}+<v,e>\\&=<w,s>+<v,e>\end{aligned}$$

文献[41]中有结论: 区分$<\boldsymbol{v},\boldsymbol{c}>$和一个$\mathbb{Z}_q$中的随机元素的优势约为$\varepsilon=4e^{-2\pi(||(\boldsymbol{v},\boldsymbol{w})||)^2}$。利用格基约化算法在$\Lambda(\boldsymbol{A})$上找到的最短向量长度为$\delta_0^{m+n}q^{\frac{n}{m+n}}$, 因

此可以通过下式建立起格基约化算法质量和解决 decision-LWE 问题成功率之间的关系:

$$\varepsilon = 4e^{-2\pi^2\cdot(\delta_0^{m+n}\frac{n}{qm+n\cdot\alpha})^2}。$$

4.2 Decoding 方法

Decoding 方法的基本思想是将解决 search-LWE 问题转化为解决q-ary 格上的 BDD 问题, 然后直接利用最近平面算法求解。具体地, 给定服从分布$L_{s,_}$选取的m个 LWE 实例$(\boldsymbol{A}, \boldsymbol{c}=\boldsymbol{As}+\boldsymbol{e})_\mathbb{Z}_q^{m_n}_\mathbb{Z}_q^m$, 其中错误$\boldsymbol{e}$是服从高斯分布的, 用矩阵$\boldsymbol{A}$构造$q$-ary 格$_q(\boldsymbol{A^T})$并利用格基约化算法得到一组新的约化基, 将$\boldsymbol{c}$视作 BDD 问题的目标向量, 用它调用最近平面算法即可以以一定的概率得到$\boldsymbol{As}$。

因此 decoding 方法可以视为一个两阶段的算法, 第一阶段调用格基约化算法对输入格基进行预处理, 第二阶段调用最近平面算法找到目标向量。显然, 第一阶段得到的约化基质量越好, 第二阶段最近平面算法的时间复杂度就越低、成功率就越高, 但为了得到更好的格基, 第一阶段的时间复杂度就越高。因此, 需要综合考虑如何平衡两阶段之间效率和质量问题。

4.2.1 Babai 最近平面算法

Babai 最近平面算法[48]的基本思想是: 给定格的一组基$\boldsymbol{B}=\{\boldsymbol{b_1}, _, \boldsymbol{b_m}\}$和目标向量$\boldsymbol{t}$, 算法输出一个向量$\boldsymbol{v}$, 使得$\boldsymbol{v}$是满足$\boldsymbol{v}_\boldsymbol{t}+P(\boldsymbol{B}^*)$的唯一的格向量($P(\boldsymbol{B}^*)$是正交基$\boldsymbol{B}^*$张成的格的基本区域)。

给定服从分布$L_{s,_}$选取的的m个 LWE 实例$(\boldsymbol{A}, \boldsymbol{c})_\mathbb{Z}_q^{m_n}_\mathbb{Z}_q^m$, 构造格$L(\boldsymbol{A^T})$, 利用格基约化算法得到它的一组基$\boldsymbol{B}=\{\boldsymbol{b_1}, _, \boldsymbol{b_m}\}$。以$\boldsymbol{c}$为目标向量调用 Babai 最近平面算法, 若算法的输出为$\boldsymbol{As}$, 则可利用它恢复出$\boldsymbol{s}$, 那么也就解决了 search-LWE 问题。根据 Babai 最近平面算法的性质可知, 算法输出$\boldsymbol{As}$当且仅当$\boldsymbol{e_s}+P(\boldsymbol{B}^*)$, 这一事件发生的概率为

$$\Pr[\boldsymbol{e\_s}+P(\boldsymbol{B}^*)] = \_{i=1}^{m}\Pr[|<\boldsymbol{e}, \boldsymbol{b_i^*}>| < \frac{\left|\left|\boldsymbol{b_i^*}\right|\right|^2}{2}]$$
$$\_\_{i=0}^{m-1}\mathrm{erf}\_(\frac{\left|\left|\boldsymbol{b_i^*}\right|\right|\sqrt{\_}}{2\alpha q}),$$

其中$\mathrm{erf}(x)=\frac{2}{\sqrt{_}}\int_0^x e^{-t^2}\mathrm{d}t$。

可以看出, Babai 最近平面算法的成功率依赖于输入基的质量, 输入基的正交性越好, 找到的靠近目标向量的格向量越短。但通常情况下, 输入基的形状比较“细长”, 并且随着维度的增加错误规模也会增加, 在这样的情况下, $\boldsymbol{e_s}+P(\boldsymbol{B}^*)$的概率下降, 算法的成功率下降。

4.2.2 LP 最近平面算法

2011 年, Lindner 和 Peikert[46]将 Babai 最近平面算法进行推广, 得到的算法称为 LP 最近平面算法。该算法在每一维上选择多个超平面$(d_i_\mathbb{Z}_+)$, 将搜索范围在每个$\boldsymbol{b_i^*}$方向上扩大为原来的d_i倍, 增大了$\boldsymbol{e}$落在其中的概率。此时, 算法可以得到$d_m___d_1$个备选结果, 分别计算它们到目标向量$\boldsymbol{c}$的距离, 距离最小的即为 LP 最近平面算法的最终输出。

利用这种思想, 算法成功率增大为

$$\Pr[\mathrm{e\_s}+\mathrm{P}(\mathrm{B}^*\_)] = \_{i=1}^{m}\Pr[|<e, b_i^*>| < \frac{d_i\left|\left|b_i^*\right|\right|^2}{2}]$$
$$\_\_{i=0}^{m-1}\mathrm{erf}\_(\frac{d_i\left|\left|b_i^*\right|\right|\sqrt{\_}}{2\alpha q}),$$

其中$\boldsymbol{D}$是对角矩阵, 对角线上的每个对应元素为d_i。

4.3 Primal 方法

同 decoding 方法类似, primal 方法也通过 BDD 策略解决 search-LWE 问题。与 decoding 方法不同的是, primal 方法将 search-LWE 问题转化为q-ary 格上的 BDD 问题后, 再用嵌入技术将其归约到 uSVP 问题上, 最后通过格基约化算法解决 uSVP 问题解决 search-LWE 问题。

嵌入技术是 1987 年 Kannan[49]在 CVP 到 SVP 问题的归约中首次提出的, 现已成为处理 CVP 问题的通用方法。Primal 方法中主要涉及两种嵌入技术, 一种是 Kannan 嵌入[49], 另一种称为 dual 嵌入[50]。

4.3.1 Kannan 嵌入

Kannan 嵌入的基本思想是将 LWE 问题归约到 BDD 问题, 再归约到 uSVP 问题[49]。

具体地, 给定服从分布$L_{s,_}$选取的m个 LWE 实例$(\boldsymbol{A}, \boldsymbol{c})_\mathbb{Z}_q^{m_n}_\mathbb{Z}_q^m$, 可构造格$L(\boldsymbol{A})=\{\boldsymbol{x}_\mathbb{Z}^m|_\boldsymbol{u}_\mathbb{Z}_q^n\ s.t.\ \boldsymbol{x}=\boldsymbol{Au}\}$。对$\boldsymbol{A^T}_\mathbb{Z}_q^{n_m}$作初等列变换得到$[\boldsymbol{I_n}|\boldsymbol{A'}]$, 将其嵌入一个$m+1$维的$q$-ary 格$L(\boldsymbol{B})$中, 其中$\boldsymbol{B}$是一组基, 它的构造为$\boldsymbol{B}=\begin{pmatrix}\boldsymbol{I_n} & \boldsymbol{A'} & 0\\ \boldsymbol{0} & q\boldsymbol{I_{m-n}} & 0\\ \boldsymbol{c^T} & & t\end{pmatrix}$, 其中$t$为嵌入因子。设置嵌入因子$t=\left|\left|\boldsymbol{e}\right|\right|=dist(\boldsymbol{c}, L(\boldsymbol{A}))$时, $L(\boldsymbol{A})$中包含一个最短向量$\boldsymbol{c'}=(\boldsymbol{e}, -t)$[16], 提取$\boldsymbol{c'}$的前$m$位就可以可恢复出$\boldsymbol{e}$。在实验中可看出, 选取$t<\left|\left|\boldsymbol{e}\right|\right|$可以使得利用格基约化算法解决这样的 uSVP 实例时更加高效, 故在实际应用中一般设置$t=1$。

对于这样的 uSVP 实例, 在使用格基约化算法进行求解时, 对于格基约化算法分组长度β的取值有两种不同的分析方式: 其中第一种方式以文献[27]为代表, 利用 uSVP 中第一和第二短的向量长度(表示为λ_1和λ_2)之间的比值和δ_0之间的关系作为纽带计算β的取值, 即$\frac{\lambda_2}{\lambda_1}\geqslant c\cdot\delta_0^{m+1}$, 其中$c_(0,1)$是一个与

uSVP 实例和所用格基约化算法相关的常数, 在文献[33]中对于β的选取采用的就是这种方式; 第二种方式以文献[41]为代表, 文献[41]中给出了利用 BKZ 算法通过 Kannan 嵌入的方法(设置嵌入因子$t=1$), 在 GSA 成立的条件下, 对所需β的估计式:

$$\sqrt{\frac{\beta}{m+1}}\|(\boldsymbol{e}|1)\| \approx \sqrt{\beta\sigma} \leqslant \delta_0^{2\beta-(m+1)} vol(L(\boldsymbol{A}))^{\frac{1}{m+1}}$$

后来, 在2017年Albrecht等人[59]以Kannan嵌入为例, 对 primal 方法中β的选取方式进行了研究, 用理论分析和实验数据论证了第二种β的选取方式更为合理和契合实际。这一结论对于下文的其他嵌入方式也同样适用。

4.3.2 Dual 嵌入

Dual 嵌入由 Bai 和 Galbraith[50]在 2014 年提出, 它的基本思想是将 LWE 问题归约到 ISIS 问题, 再归约到 CVP, 最终归约到 uSVP。

给定服从分布$L_{\boldsymbol{s},\chi}$选取的m个 LWE 实例$(\boldsymbol{A},\boldsymbol{c})\in\mathbb{Z}_q^{m\times n}\times\mathbb{Z}_q^m$, 可以将其嵌入一个$n+m+1$维的格$L(\boldsymbol{M})=\{\boldsymbol{x}\in\mathbb{Z}^n\times\mathbb{Z}^{m+1}|\boldsymbol{x}(\boldsymbol{A}|\boldsymbol{I_m}|-\boldsymbol{c})^T\equiv 0\ mod\ q\}$中, 这个格的一组基为$\boldsymbol{M}=\begin{pmatrix}\boldsymbol{I_n} & -\boldsymbol{A^T} & 0\\ \boldsymbol{0} & q\boldsymbol{I_m} & 0\\ \boldsymbol{0} & \boldsymbol{c} & 1\end{pmatrix}$。显然, 对于适当选定的向量$*$, 有$(\boldsymbol{s}|*|1)\cdot\boldsymbol{M}=(\boldsymbol{s}|\boldsymbol{e}|1)$。

文献[51]中有结论: 对于普通形式的 LWE(除 normal form LWE 以外), Kannan 嵌入和 dual 嵌入的效果基本无差别。

另外, 在实际应用中, 还有一类 LWE 问题实例中的$\boldsymbol{s}$的每个分量都取自$\{-1,0,1\}$, 此类 LWE 问题称为二进制 LWE(binary-LWE)。针对这种情况, 引入模转换和 rescale 两种技术, 可以有效地降低算法时间复杂度。

4.3.3 模转换技术

模转换技术最初提出时是用于改善全同态加密体制的效率, 它的思想与固定位数的低精度浮点数运算十分类似。在对于 LWE 问题的分析中, 模数的大小直接影响着算法总体时间复杂度, 一般来说, 在其他参数保持不变的情况下, 模数越小, 时间复杂度越低。因此可以利用该技术对 LWE 的分析进行优化。

具体地, 一个参数为n,q,α的 LWE 实例$(\boldsymbol{a},c)\in\mathbb{Z}_q^n\times\mathbb{Z}_q$, 利用模转换技术可以被转化为一个参数为$n,p,\sqrt{\frac{2n}{12}}\cdot\frac{s}{\alpha},\sqrt{2}\alpha(\cdot_s$是$\boldsymbol{s}$中元素的标准差)的形如$(\frac{p}{q}\cdot\cdot,\frac{p}{q}\cdot\cdot)\in\mathbb{Z}_p^n\times\mathbb{Z}_p$的新的 LWE 实例, 其中$\frac{p}{q}\cdot\cdot$计算过程如下:

$$\begin{aligned}\frac{p}{q}\cdot\cdot &= \frac{p}{q}\cdot(<\boldsymbol{a},\boldsymbol{s}>+q\cdot+e)\cdot\\ &= -<\frac{p}{q}\cdot\boldsymbol{a},\boldsymbol{s}>_p+\frac{p}{q}\cdot\cdot\\ &= -<\frac{p}{q}\cdot\boldsymbol{a}\cdot,\boldsymbol{s}>_p+<\frac{p}{q}\cdot\boldsymbol{a}-\frac{p}{q}\cdot\boldsymbol{a}\cdot,\boldsymbol{s}>_p+\frac{p}{q}\cdot\cdot\\ &= <\frac{p}{q}\cdot\boldsymbol{a}\cdot,\boldsymbol{s}>_p+<\frac{p}{q}\cdot\boldsymbol{a}-\frac{p}{q}\cdot\boldsymbol{a}\cdot,\boldsymbol{s}>_p+\frac{p}{q}\cdot\cdot+e'\\ &= <\frac{p}{q}\cdot\boldsymbol{a}\cdot,\boldsymbol{s}>_p+e''+\frac{p}{q}\cdot\cdot+e'\end{aligned}$$

其中$u\in\mathbb{Z},e'\in[-0.5,0.5],\langle x,y\rangle_p$表示在模$p$的意义下对向量$\boldsymbol{x}$和$\boldsymbol{y}$求内积。

如上文所述, 当模数p减小时, 算法的时间复杂度会降低, 但是当p取得过小时, 又会造成噪音规模的增加, 导致解决这个实例的困难度增加, 因此p的取值需要平衡这两者之间的矛盾, 最优的p值的选择可以使得$|<\frac{p}{q}\cdot\boldsymbol{a}-\frac{p}{q}\cdot\boldsymbol{a}\cdot,\boldsymbol{s}>|\cdot\frac{p}{q}\cdot\cdot e|$, 即在模转换后的错误规模大致放缩为原来的$\frac{p}{q}$, 据此, 可以计算出$p\cdot\sqrt{\frac{2n}{12}}\cdot\frac{s}{\alpha}$。

模转换技术可以应用于 Kannan 嵌入和 dual 嵌入中。

4.3.4 Rescale 技术

Rescale技术由Bai和Galbraith[52]在2014年提出, 这种技术可以被应用在dual嵌入中, 并将其称为BG嵌入。在 BG 嵌入中, 嵌入格可重写作$L(\boldsymbol{M})=\{\boldsymbol{x}\cdot(\nu\mathbb{Z})^n\times\mathbb{Z}^{m+1}|\boldsymbol{x}\cdot\left(\frac{1}{\nu}\boldsymbol{A}|\boldsymbol{I_m}|-\boldsymbol{c}\right)^T\cdot 0\ mod\ q\}(\nu\cdot 0)$的形式, 基$\boldsymbol{M}=\begin{pmatrix}\nu\boldsymbol{I_n} & -\boldsymbol{A^T} & 0\\ \boldsymbol{0} & q\boldsymbol{I_m} & 0\\ \boldsymbol{0} & \boldsymbol{c} & 1\end{pmatrix}$, 且有$(\boldsymbol{s}|*|1)\cdot\boldsymbol{M}=(\nu\boldsymbol{s}|\boldsymbol{e}|1)$。在普通的 dual 嵌入情况下时, $\nu=1$, 但可以看出当$\boldsymbol{s}$的每个分量很小或很稀疏时, 若选择$\nu=1$, 向量$(\boldsymbol{s}|\boldsymbol{e}|1)$是不平衡的, 也就是说, 有$\frac{||\boldsymbol{s}||}{\sqrt{n}}\cdot\frac{||\boldsymbol{e}||}{\sqrt{m}}=\cdot$。此时, 将其转化为平衡的, 即重新选择合适的$\nu$值使得$||(\nu\boldsymbol{s}|\boldsymbol{e}|1)||\approx\sqrt{n+m}$, 可以使得调整后的向量$(\nu\boldsymbol{s}|\boldsymbol{e}|1)$就是格$L(\boldsymbol{M})$中的一个最短向量, 且利用格基约化算法找到它变得更加高效。以$\boldsymbol{s}\overset{\$}{\cdot}\{-1,0,1\}^n$为例, 有$||\nu\boldsymbol{s}||^2\cdot\frac{2}{3}\nu^2 n$, 因此令$\nu=\sqrt{\frac{3}{2}}\cdot$, 可以算出$||\nu\boldsymbol{s}||\approx\sqrt{n}$, 那么有$||(\boldsymbol{s}|\boldsymbol{e}|1)||\approx\sqrt{n+m}$。

5 实际安全性分析结果

在渐进性分析方法中, 通常将对数或者常数因子隐藏在表达式中, 这样可以从宏观上理解和比较不同算法。但在设计实际方案时, 需要选择具体的参

数使得方案在保证一定的安全性强度的情况下效率尽可能地高。为此, 必须研究对于求解 LWE 问题的实际算法的性能, 并以此作为参考来确定方案的安全级别和参数设置。

本章对于 LWE 问题的具体安全性分析结果, 主要基于两类不同的评估程序: 文献[33]中的评估程序和文献[41]中的评估程序。

LWE 的主要参数有维数n、模数q、错误率α(或错误分布的标准差$_ = \frac{\alpha q}{\sqrt{2\pi}}$), 以及用到的实例数量$m$。需要注意的是关于实例数量, 文献[33]中给出的评估结果所用的实例数量, 是使得分析结果达到最优时的实例数量。而在文献[41]中, 作者考虑到在实际情况中, 攻击者能够拿到的实例数量是有限的, 因此将实例数量m限制在了$(0,2n)$范围内。对于文献[33]中的评估程序在使用有限的实例数量时的安全性的评估结果如何, 文献[58]对此进行了实验并给出了相应结果。

下文安全性评估结果的表格中参数含义与此处相同。在进行实际安全性评估时, 在上述参数选定的条件下, 可以分别得到利用不同方法解决 LWE 问题时的时间成本——此处时间成本统一用计算机时钟周期(clock cycle)的数目进行衡量。

这些评估程序分别利用 dual 方法、decoding 方法和 primal 方法对几种典型的基于 LWE 的加密方案设计(Regev 方案[7]、LP 方案[46]、BCNS 方案[53]、NTRU 方案[54])进行了安全性评估并给出了评估结果。

文献[33]中的评估程序利用 dual 方法、decoding 方法、primal 方法(Kannan 嵌入和 BG 嵌入)对 Regev[7]和 LP[46]两种加密方案的安全性评估结果分别如表 2、表 4、表 5 所示, 其中参数λ表示某方案在对应参数情况下的安全性级别, “Regev with $s_i \leftarrow \$ \{0,1\}$”和“LP with $s_i \leftarrow \$ \{0,1\}$”两列是这两种方案在$\boldsymbol{s}$的每个分量均取自$\{0,1\}$上的均匀分布时的情况。

文献[41]中的评估程序利用 dual 方法对 BCNS[53]和 NTRU[54]加密方案的安全性评估结果如表 3 所示。

表 2 文献[33]中 dual 方法的安全性评估结果

Table 2 The Estimation Results of Dual Approach in [APS15]

	Regev	Regev with $s_i \leftarrow \$ \{0,1\}$	LP	LP with $s_i \leftarrow \$ \{0,1\}$
q, α	$n^2, \frac{1}{\sqrt{2\pi n}\log_2^2 n}$		use [AFC+13] to select parameters	
n	λ			
64	50	49	49	49
128	86	76	81	85
256	171	146	156	185
512	339	368	308	449
1024	732	862	636	946

(注: 评估所用成本模型为 Core-SVP (min space))

表 3 文献[41]中 dual 方法的安全性评估结果

Table 3 The Estimation Results of Dual Approach in [ADPS16]

	BCNS			NTRU		
n, q, α	$1024, 2^{32}-1, 3.192$			$743, 2^{12}, \sqrt{\frac{2}{3}}$		
cost	β	m	λ	β	m	λ
classical	296	1055	86	600	635	175
quantum			78			159

(注: 评估所用成本模型为 Core-SVP(classical 行)和 Q-Core-SVP(quantum 行))

表 4 文献[33]中 decoding 方法的安全性评估结果

Table 4 The Estimation Results of Decoding Approach in [APS15]

	Regev	Regev with $s_i \leftarrow \$ \{0,1\}$	LP	LP with $s_i \leftarrow \$ \{0,1\}$
q, α	$n^2, \frac{1}{\sqrt{2\pi n}\log_2^2 n}$		use [AFC+13] to select parameters	
n	λ			
64	34	34	34	34
128	65	54	60	60
256	168	135	146	157
512	453	345	376	413
1024	1203	838	954	913

表 5 文献[33]中 primal 方法的安全性评估结果

Table 5 The Estimation Results of Primal Approach in [APS15]

		Regev	Regev with $s_i \leftarrow \$ \{0,1\}$	LP	LP with $s_i \leftarrow \$ \{0,1\}$
q, α		$n^2, \frac{1}{\sqrt{2\pi n}\log_2^2 n}$		use [AFC+13] to select parameters	
Cost	n	λ			
Kannan	64	50	50	50	50
	128	73	70	69	73
	256	153	146	140	184
	512	335	384	303	441
	1024	747	870	652	942
BG	64	-	49	-	49
	128	-	52	-	52
	256	-	83	-	91
	512	-	168	-	233
	1024	-	519	-	687

(注: 评估所用成本模型为 Core-SVP (min space))

6 总结

本文概述了在研究 LWE 问题时涉及的格上困难问题、格基约化算法、解决 LWE 问题的不同策略和方法, 并给出了基于现有求解算法对于 LWE 困难性的评估结果。

对于 LWE 问题, 现在主要存在两种不同的评估方法, 分别基于文献[33]和文献[41], 两种方法对于同样参数的 LWE 问题评估结果略有差别, 关于它们哪一个更契合实际情况, 目前尚无定论。这一问题是我们未来研究的重点之一。另外, 现有的分析方法, 尚有改善的余地, 这也是我们未来的研究内容之一。

参考文献

[1] J.C. Lagariasand A.M. Odlyzko, “Solving low-density subset sum problem”, *Journal of ACM*, vol. 32, no.1, pp.229-246, 1985.

[2] R.L. Rivest, A. Shamir and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems”, *Communications of the ACM*, vol.21, no.2, pp.120-126, 1978.

[3] M. Ajtai, “Generating hard instances of lattice problems (extended abstract)”, *In Proceedings of STOC'9*, pp.99-108, 1996.

[4] M. Ajtai and C. Dwork, “A public-key cryptosystem with worst-case/average-case equivalence”, *In Proceedings of STOC'97*, pp. 284-293, 1997.

[5] J. Hoffstein, J. Pipher and J.H. Silverman, “NTRU: a ring based public key Cryptosystem”, *Algorithmic Number Theory Symposium(ANTS)*, pp.267-288, 1998.

[6] O. Goldreich, S. Goldwasser and S. Halevi, “Collision-free hash from lattice problems”, https://eprint.iacr.org/1996/009, 1996.

[7] O. Regev, “On lattices, learning with errors, random linear codes, and cryptography”, *ACM Symposium on Theory of Computing* (*STOC'05*), pp.84-93, 2005.

[8] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, “Bonsai Trees, or How to Delegate a Lattice Basis”,*EUROCRYPT'10*, pp.523-552,2010.

[9] S. Agrawal, D. Boneh, X. Boyen. “Efficient lattice (H)IBE in the standard model”,*EUROCRYPT'10*, pp.553-572, 2010.

[10] B. Applebaum, D. Cash, C. Peikert and A. Sahai, “Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems”,*CRYPTO'09,* pp.595-618,2009.

[11] C. Gentry, “A fully homomorphic encryption scheme”, PhD thesis, Stanford University,2009.

[12] M. Ajtai, “The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract)”, *ACM Symposium on Theory of Computing(STOC'98)*, pp.10-19, 1998

[13] S. Khot, “Hardness of Approximating the Shortest Vector Problem in Lattices”, *Foundations of Computer Science(FOCS'04)*, pp.126-135, 2004.

[14] D, Micciancio, “The shortest vector in a lattice is hard to approximate to within some constant”, *Foundations of Computer Scienc(FOCS'98)*, pp.92-98, 1998.

[15] I. Dinur, “Approximating SVP ∞, to within Almost-Polynomial Factors Is NP-Hard”, *Italian Conference on Algorithms and Complexity(CIAC'00*, pp.263-276, 2000.

[16] V. Lyubashevsky and D. Micciancio, “On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem”, *EUROCRYPT'09*, pp.577-594, 2009.

[17] C. Peikert, “Public-key cryptosystems from the worst-case shortest vector problem: extended abstract”, *ACM Symposium on Theory of Computing(STOC'09)*, pp.333-342, 2009.

[18] Z. Brakerski, A. Langlois, C. Peikert, O. Regev and D. Stehlé, “Classical hardness of learning with errors”, *ACM Symposium on Theory of Computing(STOC'13)*, pp.575-584, 2013.

[19] D. Micciancio and O. Regev, “Worst-case to average-case reductions based on Gaussian measures”, *Foundations of Computer Science(FOCS'04)*, pp.372-381, 2004.

[20] D. Micciancio and O. Regev, “Lattice-Based Cryptography”, *CRYPTO '06*, pp.131-141,2006.

[21] D. Stehlé, R. Steinfeld, K. Tanaka and K. Xagawa, “Efficient Public Key Encryption Based on Ideal Lattices”, *ASIACRYPT'09*, pp.617-635, 2009.

[22] A.K. Lenstra, H.W. Lenstra and L. Lovász, “Factoring polynomials with rational coefficients”, *MathematischeAnnalen*, vol. 261, no.4, pp.515-534, 1982.

[23] C.P. Schnorr and M. Euchner, “Lattice basis reduction: Improved practical algorithms and solving subset sum problems”, *Mathematical Programming*, vol.66, pp.181-199, 1994.

[24] Y. Chen and P.Q. Nguyen, “BKZ 2.0: Better Lattice Security Estimates”, *ASIACRYPT'11*, pp.1-20, 2011.

[25] N. Gama and P.Q. Nguyen, “Finding short lattice vectors within mordell's inequality”, *ACM Symposium on Theory of Computing*(*STOC'08*), pp.207-216, 2008.

[26] P.Q. Nguyen and D. Stehlé, “Floating-Point LLL Revisited”, *EUROCRYPT'05*, pp.215-233, 2005.

[27] N. Gama and P.Q. Nguyen, “Predicting lattice reduction”, *EUROCRYPT'08*, pp.31-51, 2008.

[28] C.P. Schnorr, “Lattice Reduction by Random Sampling and Birthday Methods”,*Symposium on Theoretical Aspects of ComputerScience (STACS '03),* pp.145-156, 2003.

[29] G. Hanrot and D. Stehlé. “Improved Analysis of Kannan’s Shortest Lattice Vector Algorithm”, *Lecture Notes in Computer Science*, pp.170-186, 2007.

[30] G. Hanrot and D. Stehlé, “Improved Analysis of Kannan’s Shortest Lattice Vector Algorithm”, *CRYTPO'07*, pp.170-186, 2007.

[31] Y. Chen, “Reduction de reseau et securite concrete du chiffrementcompletementhomomorphe”, Paris, 2013.

[32] D.Stehl´e, “An overview of lattice reduction algorithms”, Invited talk at*International Conference on Information Security and Cryptology(ICISC'13)*, 2013.

[33] M.R. Albrecht, R. Player and S. Scott, “On the concrete hardness of Learning withErrors”, *Journal of Mathematical Cryptology*, vol.9, no.3, pp.169-203, 2015.

[34] D. Micciancio and P. Voulgaris, “Faster exponential time algorithms for the shortest vector problem”, *ACM-SIAM symposium on Discrete algorithms (SODA'10)*, pp.1468-1480, 2010.

[35] Y. Aono, Y. Wang, T. Hayashi and T. Takagi, “Improved progressive BKZ algorithms and their precise cost estimation by sharp simulator”,*EUROCRYPT'16, pp.65-102, 2016.*

[36] TU Darmstadt Lattice challenge,http://www.latticechallenge.org/svp-challenge/

[37] D. Micciancio and M. Walter, “Practical, Predictable Lattice Basis Reduction”,*EUROCRYPT'16*, pp.820-849, 2016.

[38] A. Becker, L.Ducas, N. Gama and T.Laarhoven, “New directions in nearest neighbor searching with applications to lattice sieving”,*ACM-SIAM symposium on Discrete algorithms (SODA'16)*, pp.10-24, 2016.

[39] T.Laarhoven, “Search problems in cryptography: From fingerprinting to lattice sieving”, PhDthesis, Eindhoven University of Technology, 2015.

[40] T.Laarhoven, “Sieving for shortest vectors in lattices using angular locality-sensitive hashing”,*CRYPTO'15, pp.*3-22, 2015.

[41] E. Alkim, L. Ducas, T.Pöppelmann and P. Schwabe. “Postquantum key exchange - A new hope”, *Usenix Security Symposium*（*USENIX Security'16)*, pp.327-343, 2016.

[42] J. Bos, C. Costello and L. Ducas, et al, “Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE”, *ACM Sigsac Conference on Computer and Communications Security*(*CCS'16*), pp.1006-1018,2016.

[43] M.R. Albrecht, “On Dual Lattice Attacks Against Small-Secret LWE and Parameter Choices in HElib and SEAL”, *EUROCRYPT'17*, pp.103-129, 2017.

[44] A. Blum, A.Kalai and H. Wasserman, “Noise-tolerant learning, the parity problem,and the statistical query model”, *Journal of the ACM(JACM'03)*, vol. 50, no. 4, pp.506-519, 2003.

[45] S. Arora and R. Ge, “New Algorithms for Learning in Presence of Errors”, *International Colloquim Conference on Automata, Languages and Prgramming*(*ICALP '11*), pp.403-415, 2011.

[46] R. Lindner and C. Peikert. “Better Key Sizes (and Attacks) for LWE-Based Encryption”, *Cryptographers' Track at the RSA Conference* (*CT-RSA'11)*, pp.319--339, 2011.

[47] B.Applebaum, D. Cash, C.Peikert, and A.Sahai, “Fast cryptographic primitivesand circular-secure encryption based on hard learning problems”,*CRYPTO'09*, pp.595-618, 2009.

[48] L.Babai, “On Lov´asz’ lattice reduction and the nearest lattice point problem (shortenedversion)”, *Symposium of Theoretical Aspects of Computer Science*(*STACS'86)*, pp.13-20, 1985.

[49] R. Kannan, “Minkowski's convex body theorem and integer programming”, *Mathematics of Operations Research*, vol.12, no.3, pp.415-440, 1987.

[50] S. Bai and S.D. Galbraith, “An Improved Compression Technique for Signatures Based on Learning with Errors”, *RSA Conference, Cryptographers' Track (CT-RSA'14)*, pp.28-47, 2014.

[51] M.R. Albrecht, R. Fitzpatrick and F. Göpfert, “On the Efficacy of Solving LWE by Reduction to Unique-SVP”, *International Conference on Information Security and Cryptology(ICISC'13)*, pp.293-310, 2013.

[52] S. Bai and S.D. Galbraith, “Lattice Decoding Attacks on Binary LWE”, *Australasian Conference on Information Security and Privacy(ACISP'14)*, pp.322-337, 2014.

[53] J.W. Bos, C. Costello, M. Naehrig and D. Stebila, “Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem”, *Security and Privacy(SP'15)*, pp.553-570, 2015.

[54] J. Hoffstein, J. Pipher and J.M. Schanck, et al, “Choosing Parameters for NTRUEncrypt”, *Cryptographers' Track at the RSA Conference(CT-RSA'17)*, pp.3-18, 2017.

[55] D. Coppersmith, “Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities”, *Journal of Cryptology*, vol.10(4), pp.233-260, 1997.

[56] Y. Yu and L. Ducas, “Second order statistical behavior of LLL and BKZ”, in *SAC*, pp.3-22, 2017.

[57] S Bai, D.Stehlé and W.Q. Wen, “Measuring, simulating and exploiting the head concavity phenomenon in BKZ”, *IACR Cryptology ePrint Archive 2018*: 856, 2018.

[58] M. Schmidt and N. Bindel, “Estimation of the hardness of the learning with errors problem with a restricted number of samples”, *IACR Cryptology ePrint Archive 2017*: 140, 2017.

[59] M.R. Albrecht, F. Göpfert, F. Virdia and T. Wunderer, “Revisiting the Expected Cost of Solving uSVP and Applications to LWE”, *ASIACRYPT 2017*, pp.297-322, 2017.

毕蕾 于 2017 年在中国科学院大学计算机技术专业获得硕士学位。现在中国科学院大学网络空间安全专业攻读博士学位。研究领域为格密码。研究兴趣包括: 格上困难问题分析等。Email: bilei@iie.ac.cn

李帅钢 于 2016 年在郑州大学信息与计算科学专业获得学士学位, 现在就读于中国科学院大学信息安全专业攻读博士学位。研究领域为格密码, 研究兴趣包括: 格攻击等。Email: lishuaigang@iie.ac.cn

刘亚敏 于 2011 年在中国科学院大学信息安全专业获得工学博士学位。现任中国科学院信息工程研究所助理研究员。研究领域为公钥密码学。研究兴趣包括: 可证明安全理论、基于格的公钥密码算法设计。Email: ymliu@is.ac.cn

张江 于 2015 年在中国科学院大学信息安全专业获得博士学位, 现任密码科学技术国家重点实验室副研究员。研究领域为现代密码学。研究兴趣包括: 公钥密码可证明安全理论、抗量子密码、多方安全计算协议设计与分析研究。Email: jiangzhang09@gmail.com

范淑琴 于 2003 年在信息工程大学密码学专业获得博士学位, 现任密码科学技术国家重点实验室研究员。研究领域为公钥密码学。研究兴趣包括: 格公钥密码等。
Email: fansq@sklc.org