arXiv:2303.03978v1 [cs.CR] 7 Mar 2023

量子算法中分圆域单位群的特殊情形

Razvan Barbulescu[1⋆] 和 Adrien Poulalion[2]

[1] 法国波尔多大学、法国国家科学研究中心、波尔多国立高等综合理工学院、数学基础研究所、UMR 5251，F-33400，塔朗斯，法国 razvan.barbulescu@u-bordeaux.fr[2] Alice & Bob，矿业学院，法国 adrien.poulalion@mines.org

摘要。 单位群计算是一种密码学原语，针对该原语已有快速量子算法，但所需的量子比特数量为 $\tilde{O}(m^5)$。在本文中，我们针对分圆域提出了该算法的一种修改版本，使量子比特数量降为 $\tilde{O}(m^2)$。此外，根据关于$\mathbb{Q}$ $\zeta_m + \zeta_m^{-1}$类群大小的一个近期猜想，该量子算法会简单得多，因为它是一个隐藏子群问题（HSP）算法，而不是其误差估计对应算法：连续隐藏子群问题（CHSP）。我们还讨论了利用伽罗瓦自同构所获得的（小幅）加速，这得益于在 $\mathcal{O}_K$-格上使用 Buchmann-Pohst 算法。

1 引言

计算类群及其基数（即类群）的难度在密码学中发挥着重要作用。值得注意的是，它们是时间承诺协议 [49], 、同态加密方案 [15] 以及可验证延迟函数 [50] 的基础。需要注意的是，特定情形也很重要：前两个例子使用二次域，而对于后一个例子，分圆情形可能是最快的。

从理论计算机科学的角度来看，这些方案已被攻破，因为存在一个多项式时间复杂度的量子算法 [10]。然而，从密码学角度来看，还需要更精确地估算所需的量子比特数量和量子门数量，正如针对公钥密码学的其他原语所做的那样[7,4,40,6,44,51,29,45,40]。

让我们回顾一下研究这一问题的工作历程。值得注意的是，在经典算法和量子算法中，类群与单位群非常相似，有时会同时计算它们以获得终止条件，例如 Buchmann和McCurley的经典算法。1994年，Shor设计了一种量子算法，用于分解整数，并求解循环群中的离散对数问题 [48]。Kitaev [33] 重新表述了这些算法，将这两个问题都归结为寻找函数的周期集合；这些函数

⋆第一作者获得了法国“法国2030”计划混合量子倡议（HQI）的资助。

定义在 $\mathbb{Z}^r$ 上，具有有限的 r ，且取值于一个有限集合；这就是隐藏子群问题（HSP）。值得注意的是，参数 r 仅以多项式方式影响时间和空间复杂度。

2002年，Hallgren [28][3] 将实二次域基本单位的计算归结为寻找定义在 $\mathbb{R}$ 上的函数的周期集合。更一般地，如果定义在 $\mathbb{R}^m$ 上的函数具有一个周期格（离散子群），那么寻找周期集合就是连续隐藏子群问题（CHSP）。与因式分解和离散对数一样，单位群在经典计算机上可在亚指数时间内计算，而在量子计算机上则可在多项式时间内计算。

当 $m = 1$ 时，隐藏子群问题与 CHSP 之间的主要区别在于：给定两个整数 k 和 ℓ，以及 $k\alpha$ 和 $\ell\alpha$ 的近似值，求得 $\alpha \in \mathbb{R}$ ；这可以通过连分数来解决。要计算一族次数为常数的数域的基本单位，就必须求解其中 $m \geq 1$ 有界于某个常数的 CHSP。在近期的研究中，Hallgren [27] 和 Schmidt [46] 用 Buchmann 和 Pohst [13] 提出的基于 LLL算法的算法取代了连分数。

到目前为止，CHSP和隐藏子群问题具有相同的量子部分，但在经典后处理方面有所不同。当数域的次数在判别式趋于无穷大的一族数域中可以无界增长时，就必须针对无界参数 m求解CHSP，这些参数可能大到判别式的对数映射。在这种情况下，先前的算法 [27,46] 需要关于 m（数域的次数）呈指数级的时间，正如Biasse和Song在 [11, 命题B.2][4]中所证明的那样。

2014年，Eisenträger等人 [21][5] 成功地将该算法推广为适用于任意次数数域的多项式时间算法。为此，他们将周期函数乘以一个高斯函数。在这项工作的完整版本 [22] 公开之前，第二个团队de Boer等人 [19], 对该算法进行了更为深入的分析，并于2019年确立了CHSP的精确复杂度（空间和时间）。

我们的贡献与密码学建议。本文旨在按照 NIST 规范 [43] 的要求，精确比较一方面计算单位群和类群，另一方面破解 AES 等对称密码系统所需的资源。量子计算机技术领域的专家正在研究 NISQ 场景：经过纠错、拥有 100 到 1000 个量子比特的量子计算机将成为现实，而拥有 10^{20} 个量子比特的量子计算机仍难以实现，或成本过高而不值得考虑。在此背景下，仅将量子比特数估算为关于 [22] 的“多项式”并不足够。我们在 CHSP [19] 的一般框架下具体分析单位群和类群的情形，得出所需量子比特数为 $O(m^5)$；例如，当 $m = 10000$ 时，量子攻击需要 10^{20} 个经过纠错的量子比特。

[3] 2002年版本有6页，而2007年发表的版本有19页。[4] 该命题在2015年版本中编号为2。
[5] 2014年的10页长原始版本不包含证明，这些证明直到2019年的47页长版本中才公开。

接下来，我们将研究可能存在的弱数域 K：如果 K 存在自同构，这会构成安全问题吗？如果 K 是分圆的，这会是一种弱点吗？

在第 3.3 节中，我们说明自同构使得能够在$\mathcal{O}_K$ 上实现LLL算法，并获得加速效果；加速幅度至多为自同构数量的多项式级。

对于分圆域，加速效果并非自动实现：如果在这些域上运行通用算法，所需量子比特数为 $O(m^5$ 对数映射 $m)$。然而，我们有如下观察：**假设 L 是一个未知格，$M \subset L$ 是已知的且具有短基，并且有一种算法能够生成接近 L^*的向量。那么，可以利用 M 的短基，将任意接近 L^* 的向量映射到 M^* 中与其最近的向量；这些向量也会自动属于 L^***。

密码学建议。对计算类群和单位群所需资源的精确估算表明，只要避免弱密钥，它们就比对称密码系统更具抗攻击能力。如果要使用基于类群（CGP）的密码学，那么：

– 必须避免使用固定次数域，因为它们需要 $O(m)$ 个量子比特；– 必须避免使用分圆域，因为它们需要 $O(m^2 \log m)$ 个量子比特；此外，还存在一种基于隐藏子群问题的算法；– 可以考虑使用具有自同构的域来加速计算，因为它们需要 $\Omega(m^3)$ 个量子比特，而加速效果仅涉及 Buchmann-Pohst 算法，且与自同构的数量呈多项式关系。

路线图。 本文组织如下。在第 2 节中，我们将改进作为一个一般的格问题来阐述，不涉及单位群和量子计算的背景。接着，在第 3 节中，我们详细介绍用于单位群的量子算法，并结合相关结果，给出任意数域上这些算法的复杂度。我们在第 3.3 小节中利用自同构。在第 4 节中，我们将前述复杂度具体应用于分圆域和阿贝尔域的情形，并与第 2 节中的改进进行比较。最后，在第 4.2 节中，我们尝试将单位群计算归约为隐藏子群问题，而非 CHSP；这大幅减少了所需的量子比特数，尤其是对于指数为 p^k 且 p 为小素数的分圆域。

2 将我们的改进视为一个格问题

令 $\langle x, y\rangle$ 表示两个向量 $x, y \in \mathbb{C}^n$的点积。给定一个格$L \subset \mathbb{R}^m$，我们称 L 的对偶格为

$$L^* = \{y \in \mathbb{R}^m \mid \forall x \in L, \langle x, y\rangle \in \mathbb{Z}^n\}.$$

我们回顾一下 L^* 的一系列性质（参见 [3] ）。

引理 21 *1.* 若 L 由矩阵 B 的各行生成，则 L^* 由 $(B^t)^{-1}$的各行生成；特别地，行列式 $L^* = 1/$ 行列式L；*2.* 若 M是 L的子格，则 $L^* \subset M^*$和 $[L\colon M] = [M^*\colon L^*]$。

2.1 非正式介绍

用于计算单位群的 CHSP算法称 L 为单位格，并采用以下策略:

1. 使用量子过程生成 $\mathbb{C}^n$ 中的向量，这些向量分别位于 L^*的一组生成元中每个向量附近的小球内。2. 应用 Buchmann-Pohst 算法（该算法是经典的多项式时间算法），并利用先前找到的向量，求出 L^*的一组基。3. 对先前求得的基的矩阵求逆，以得到 L 的一组基（这是经典的多项式时间操作）。

Buchmann-Pohst 算法的一个特点是精度会大幅下降[6], ，也就是说，为了使 L^*的基所需的精度位数大幅减少，步骤 1 中计算出的向量必须具有较高的精度。给定精度要求 τ，即 [19] 中有 log τ 位精度，可以计算得出，Buch mann-Pohst 的输入精度必须取决于 k。在这项工作中，我们减少了步骤 1 所需的量子比特数。

对于分圆域，分圆单位格有一个基 $M \subset L$。这使我们能够计算出 M^* 的一个基，其中包含未知格 L^*。我们的想法如下：给定一个足够接近 L^* 的向量，针对 M^* 求解最近向量问题，它便会自动落在 L^* 中。更具体地，设 m_1、...、...、m_n 是 L 的一个基。我们以低精度执行步骤 1，得到 $\widetilde{y}$，然后将其修正为 y，使得对于所有 $i = 1$=2，.....、n，都有 $\langle y$、$m_i\rangle \in \mathbb{Z}^n$。此时，我们便得到了 L^* 的一个高精度生成集。为简化说明，我们仍可应用 Buchmann-Pohst 算法。实际实现时，也可以使用 Hermite标准形算法。

2.2 精确陈述

步骤 1 生成的向量 $\widetilde{y}$ 接近 L^*，且满足如下性质： $y :=$ 最近向量问题$(\widetilde{y}, L^*)$ 在 $\mathbb{R}^m$ 上服从一个连续分布，该分布接近于 L^*上的离散高斯分布 c。下面的定义精确描述其输出，并着重说明后续各节将使用的 c 的性质。

定义 22（对偶格采样器）设 c： $L^* \to \mathbb{C}$ 是一个映射，满足$\sum_{\ell^* \in L^*} |c_{\ell^*}|^2 = 1$。设 $1/4 > \epsilon > 0$、 $1/2 > \delta > 0$ 和 $r > 0$ 为三个参数。若某算法输出一个向量 $x \in \mathbb{R}^m$，且对任意有限集合 $S \subset L^*$，均有

$$\mathrm{Prob}\left(y \in \bigcup_{\ell^* \in S} B(\ell^*, \delta\lambda_1^*)\right) \geq \sum_{\ell^* \in S} |c_{\ell^*}|^2 - \eta.$$

对于满足以下条件的映射 c：

[6] 精确估计请参见 [19] 中的定理 10。

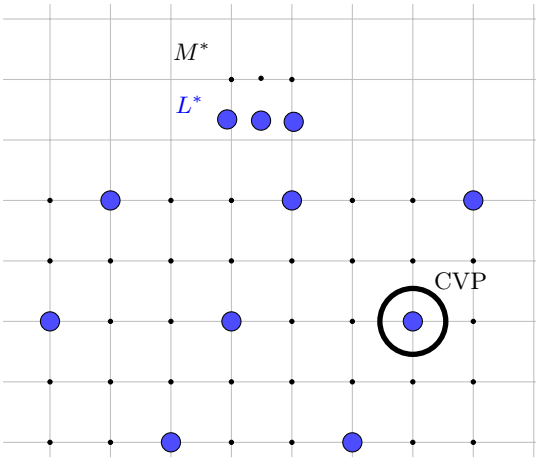

图 1：算法 2 的示意图。

1. **均匀性**：存在 $\varepsilon \leq 1/4$，使得对于每个严格子格 $N \subsetneq L^*$：

$$\sum_{\ell^* \in N} |c_{\ell^*}|^2 < \frac{1}{2} + \varepsilon.$$

2. **集中性**：存在 $r = r(m)$ 以及 $0 < p < \frac{1}{2} - \varepsilon - \eta$，使得：

$$\sum_{|\ell^*| > r} |c_{\ell^*}|^2 < p.$$

现在我们可以定义要解决的问题。

问题 *1.* 设 $M \subset L \subset \mathbb{R}^m$ 是两个格。设 ϵ、δ、$r > 0$ 和 $k \in \mathbb{N}$ 为参数。给定

– M 的一组基；– $[L$ 的一个上界：$M]$；– 一个针对我们所选参数 δ 和 η 的 L 对偶格求解器。

计算由 $[L$ 的值给出的基 $(z_1, \ldots, z_m) \in L :\ M]$ 以及 $[L^{1}$ 的一个基中 z_i 的坐标：$: {}_{M]}M$。

令 B_M 为矩阵，其各行为 M 的基 $(z_1, \ldots, z_m)$。我们提出以下解决该问题的方法，稍后将明确 δ 的取值，使得时间复杂度为多项式。

算法 1 使用对偶格采样器完整计算 L。

输入： – 格 L 的行列式 L 的上界，以及 λ_1^* 的下界
– ϵ、δ、$r > 0$，以及这些参数的 L 对偶采样器；
– 子格 $M \subset L$ 的一组基
输出： L的基
1: $k \leftarrow m$ 对数映射$_2(\sqrt{m}\,\mathrm{Lip}(f)) + n$对数映射$_2(\det L))$ ▷ 来自 [19, Th 3]的值
2: **对于** $i = 1, 2, \ldots, k$ **执行** ▷ 步骤 1 - 量子
3: $\widetilde{y_i} \leftarrow$输出（对偶格采样器，$\boxed{\delta\lambda_1^*(L) = 1/2\,\|B_M\|}$）
4: $\boxed{y_i \leftarrow \text{有界距离解码}(\widetilde{y_i}, M^*, 1/2\,\|B_M\|)}$
5: **结束于**
6: ~~对（$y_1, \ldots, y_t$）使用 Buchmann-Pohst 算法。~~ 计算 Hermite标准形以从（$y_1, \ldots, y_k$）中找出 L^* 的一个基（$y'_1, \ldots, y'_m$）。这里的向量具有精确整数坐标，这些坐标基于 M^* 的一个基。 ▷ 步骤 2 - 经典
7: 计算（$y'_1, \ldots, y'_m$）的 Smith标准形（SNF），以获得精确值 of $[L : M]$。输出（B^{-1}）$^t \in \frac{1}{[L:M]}\,\mathrm{Mat}_m(\mathbb{Z})$，其中 $B \in \mathrm{Mat}_m(\mathbb{Z})$ 是矩阵 a 基 L^*，用 M^* 的一个基表示。 ▷ 第 3 步 - 经典

记号 23 对于任意矩阵 $B \in \mathrm{GL}_n(\mathbb{C})$ 以及任意 α，$\beta \in \mathbb{R}_+ \bigcup \{\infty\}$ 算子范数为

$$\|B\|_{\alpha,\beta} = \max\{\|Mv\|_\alpha \mid \|v\|_\beta \leq 1\}.$$

当未指定 α 和 β 时[7]，$\|B\|$ 表示$\|B\|_{\infty,\,1}$：

$$\|B\| = \|B\|_{\infty,1} = \max_j \sum_i |B_{i,j}|.$$

当 $\alpha = \beta$ 时，我们简单地写作 $\|B\|_\alpha$。注意，

$$\|B\|_{\infty,1} \geq \|B\|_2 = \max_i \sqrt{\sum_j |B_{i,j}|^2} \geq \|B\|_{\infty,1}/\sqrt{m}.$$

在本文中，算法的复杂度取决于 O(对数映射$\|B\|_{\infty,\,1} =$)（对数映射$O\|B\|_2$+对数映射^m）。由于 对数映射 $\|B\|^m = \Omega()$，这证明了我们可以省略算子范数的下标。

定理 24 *a)* 如果 $\delta\lambda_1^*(M) < 1/(2\,\|B_M\|)$，算法 *1* 可以解决问题 *1*。量子比特数为 $O(Qm)$，其中 Q 见方程 (1)。

我们用两个引理来展开证明。

引理 25（引理 5 见 [19]） 注意到 $k = \alpha$（$m + m$ 对数映射$_2 R +$ 对数映射$_2$（行列式 L））， 其中 $\alpha > 1$ 为绝对常数。

[7] Buchmann-Pohst 算法 [19, 附录 B] 的分析中使用的范数算子为 $\|\cdot\|_{\infty,\,1}$。

设 $\widetilde{y_1}$，$\widetilde{y_2}$，.....，$\widetilde{y_k}$ 为对偶基采样器输出的前 k 个向量。对任意$i=1$，2，.....，k，令 $y_i=$最近向量问题($\widetilde{y_i}$，L)。则对于绝对常数$\alpha>2$ 的任意取值，我们有

$$\mathrm{Prob}(y_1,\ldots,y_k \ generate \ L)\geq 1-c^m,$$

其中 $c<1$ 是一个可显式计算的常数。

从现在起，我们固定这样的一个 k，并在算法 1 中使用它。

算法 2 Babai 的有界距离解码求解器。

输入： $\widetilde{y}\in\mathbb{R}^m$，使得 $d(\widetilde{y},M^*)<1/(2\left\|B_M\right\|)$**输出：** CVP($\widetilde{y},M^*$)，以向量形式及其在 M^* 的一组基下的坐标形式给出；计算 $\widetilde{z}:=B_M^t\widetilde{y}$；将 $z=(z_1,\ldots,z_n)$ 舍入为：$=(\lfloor\widetilde{z_1}\rceil,\ldots,\lfloor\widetilde{z_n}\rceil)$ **返回** $y:=B_{M^*}z\in M^*$ 以及 $z\in\mathbb{Z}^m$。

令 δ，使得 $d(\widetilde{y},M^*)=\delta\lambda_1^*(M)$。以下结果可视为对 Babai 的一个结果的重新表述。

引理 26（[2]式（4.3）） 当 $\delta\lambda_1^*(M)<1/(2\left\|B_M\right\|_2)$ 时，有界距离解码($\widetilde{y},M^*$, $\delta\lambda_1^*(M)$) 可在经典多项式时间内求解，其中 B_M 是 M 的基矩阵。

证明。我们断言 $\lfloor B_M^t\widetilde{y}\rceil=B_M^ty'$，其中 $y'=$最近向量问题($\widetilde{y},M^*$)。事实上

$$\left\|B_M^t\widetilde{y}-B_M^ty'\right\|_\infty\leq\left\|B_M^t\widetilde{y}-B_M^ty'\right\|_2\leq\left\|B_M\right\|_2\cdot\left\|\widetilde{y}-y'\right\|_2=\left\|B_M\right\|_2\delta\lambda_1^*(M)<1/2.$$

由于 $\lfloor B_M^t\widetilde{y}\rceil=B_M^ty'$，根据引理 21，$y=B_{M^*}=((B_M)^t)^{-1}(B_M^ty')=y'$，因此该算法是正确的。

证明（定理 *24.a* 的证明）。我们将证明分为几个步骤。

步骤 1。我们将引理 25 应用于格 L^*。然后，以概率 $1-c^m$，向量 $y_1,y_2,\ldots,y_k$ 生成 L^*。

步骤 2。固定 $i\in\{1,2,\ldots,k\}$。根据对偶采样器的定义，以大于 $1-\eta$ 的概率，dist($\widetilde{y_i},L^*$) $<\delta\lambda_1^*(M^*)$。因此引理 26 适用，并且第 4 行可在经典多项式时间内执行。

由于，$L^*\subset M^*$和 y_i'：=CVP($\widetilde{y_i},L^*$) 属于 M^*。由于 $d(\widetilde{y_i},L^*)<\lambda_1^*(M)/2$，$y_i'$是 M^*中最接近 $\widetilde{y_i}$的向量，或等价地 $y_i'=y_i$。我们将其概括为 $y_i=$CVP($\widetilde{y},L$)。

步骤 3。算法 2 除了输出 z之外，还可以输出 y。这意味着，向量$y_i\in L^*$ 由它们相对于 M^*的一组基的整数坐标给出。

在第 6 行中，通过仅操作整数系数来计算厄米标准形 [16, 第 2.4 节。2]这一过程可在经典多项式时间内完成，且不会损失精度。

第 4 步。由于 $B_M\in$Mat$_k(\mathbb{Z})$，$(B_M^{-1})\in$ ~~$\frac{1}{\text{行列式}}$~~ B_MMat$_k(\mathbb{Z})$。最后，B_M及其 Smith标准形 [16, 第 2.4 节。4] 具有相同的行列式，其值为 $[L:\ M]$。因此，最后一步可在多项式时间内计算 $[L:\ M]$。

3 关于量子单位群计算的既有结果

3.1 CHSP 概述

我们首先定义连续隐藏子群问题，计算单位群的问题可以归约到该问题。

定义 31（连续隐藏子群问题 - CHSP [22]）设f： $\mathbb{R}^m \to \mathcal{S}$，其中 $\mathcal{S} = \oplus_{i\in\{0,1\}^n}\mathbb{C}|i\rangle$ 是由 n 个量子比特构成的状态空间。函数 f当且仅当满足以下技术条件时，才是一个（a， r， ε）-预言机，其隐藏满秩格 L：

1. L是 f的周期，即 $\forall x \forall \ell \in L$， f（$x+\ell$）$= f$（x）。（周期性）*2.* 函数 f 是 a-利普希茨的。（利普希茨条件）*3.* $\forall x$， $y \in \mathbb{R}^m$，使得对于 $dist$（$x-y$， L）$\geq r$，有 $|\langle f$（x）$\mid f$（y）$\rangle| \leq \varepsilon$。（强周期性）

给定一个能够高效计算 f 的量子算法，计算周期为 L 的隐藏格。

现在我们回顾一下求解 CHSP 的算法。我们模仿求解隐藏子群问题（HSP）的技术。对于后者，设 g： $G \to \mathcal{S}$，其中 G 是一个有限循环群。求解过程如下：

1. 计算叠加态 $\sum_{x\in G}|x\rangle$，然后应用 g，得到 g 的取值的叠加态： $|\psi(g)\rangle = \sum_{x\in G} g(x)|\ x\ \rangle$；2. 应用量子傅里叶变换（见 [48]），得到$\sum_{y\in G}\hat{g}(y)|\ y\ \rangle$；该表达式等于 $\sum_{y\in H^\perp}\hat{g}(y)|\ y\ \rangle$；3. 测量该状态，得到一个元素 $g \in H^\perp$。

经过几轮算法迭代，即可得到 $H^\perp$ 的一组生成元，从而计算 h。

回到连续情形，如果我们能够实现傅里叶变换，就能够从 L^* 中随机抽取向量。如果已知一组生成元，就可以提取一个基，从而完整描述 L^*。然后根据引理 21 即可得到 L。这里的障碍在于 $\mathbb{R}^m$ 是无限的，而且傅里叶变换（QFT）现在是积分，而非有限求和，因此我们无法精确计算 QFT。作为一种解决方法，我们通过黎曼和近似计算 QFT，这相当于将定义域限制在一个线段上并对其离散化。设 $\rho_\sigma(x) = \exp(-\frac{\pi^2||x||^2}{\sigma^2})$。

引理 32（ [19]中的定理 2）存在一种多项式时间的量子对偶格采样算法，使用 $Qm+n$ 个量子比特，其中

$$Q = O\left(m\log\left(m\log\frac{1}{\eta}\right)\right) + O\left(\log\left(\frac{\mathrm{Lip}(f)}{\eta\delta\lambda_1^*}\right)\right). \tag{1}$$

算法 3 CHSP 求解器

输入： 函数 f，其可在多项式时间内计算，如定义 31 所述
具有隐藏周期格 L。我们要求 $\det L$ 达到一位精度，并且
$\mathrm{Lip}(f)$。所需精度的参数 τ。

输出： $(\tilde{x_1}, ..., \tilde{x_m})$，使得 $(x_1, \ldots, x_m)$ 是 $x_i = \mathrm{CVP}(\tilde{x_i})$ 的 L 基，并且 $\|x_i - \tilde{x_i}\| \leq \tau$

1: $k \mapsto$ 对数映射$_2(\sqrt{m}\,\mathrm{Lip}(f)) + $对数映射$_2$(行列式 L)) ▷ 来自第 [19, th 3] 个值

2: **对于** $i = 1, 2, \ldots, k$ **执行** ▷ 步骤 1 - 量子

3: $\tilde{y_i} \leftarrow$输出（对偶格采样器：$\delta\lambda_1^* = \frac{(\lambda_1^*)^3(\text{行列式 } L)^{-1}}{2^{O(mk)}\|B_{L^*}\|_\infty^m}$） ▷ 来自 [19, 的值附录 B.3 Th 4]

4: pass ▷ 与算法 1 比较

5: </style}结束循环

6: 对 $(\tilde{y_1}, ..., \tilde{y_k})$ 应用 Buchmann-Pohst 算法，得到 $(\tilde{y_1'}, ..., \tilde{y_m'})$ so 该 $\mathrm{CVP}(\tilde{y_i}, L^*)_{i=1,\ m}$ 是 L^* 的一组基 ▷ 步骤 2 - 经典

7: 输出 $(B^{-1})^t$ 的各行，其中 B 有 $\tilde{u_i'}$ 行。 ▷ 步骤 3 - 经典

定理 33（ [19]中的 CHSP 求解器的空间复杂度，定理 1）我们设定[8]$\eta = 1/k^2$。令 N_f 为编码 f所需的量子比特数。我们将 τ 设为基向量上的预期误差。然后，算法 *3* 是正确的，且所需的量子比特数为：

$$\begin{aligned} N_{qubits} = O(m^3 \cdot \log(m)) + O(m^3 \cdot \log(\mathrm{Lip}(f))) + O(m^2 \cdot \log(\det(L))) \\ + O\left(m \cdot \log \frac{\mathrm{Lip}(f)}{\lambda_1^*}\right) + O\left(m \cdot \log \frac{1}{\lambda_1^* \cdot \tau}\right) + O(N_f). \end{aligned} \tag{2}$$

算法的复杂度取决于 λ_1^*，其优点在于它是格的不变量。然而， λ_1^* 难以给出下界，而且在某些应用中，我们对 B_L 的系数有界，其中 B_L 是 L 的某个基的矩阵。

引理 34 $2^{-3m}\,\|B_L^t\| \leq \frac{1}{\lambda_1^*} \leq \|B_L\|$。

证明。 第一个不等式是附录B中的 [19, 推论 61]。

第二个不等式显然成立。设 $v \in L^*$ 使得 $\|v\|_2 = \lambda_1^*$。根据记号 23，我们有 $\|B_L v\|_1 \leq \|B_L\|\,\|v\|_\infty$。由于 $v \in L^*, v \neq 0, B_L v \in \mathbb{Z}^n$，因此$\|B_L v\|_1 \geq 1$。此外，$\|v\|_2 \geq \|v\|_\infty$，所以

$$1 \leq \|B_L\|\,\lambda_1^*,$$

这证明了第二个不等式。

[8] CHSP 求解器的复杂度取决于 $\log 1/\eta$，而不是 η 本身。因此， η 的贡献隐藏在 $\tilde{O}$ 记号中。

3.2 将单位群的计算归约为 CHSP

在本小节中，我们介绍 [22] 将单位群的计算归约为 CHSP 的方法。为求完整起见，我们重述 [10, Sec] 中的描述。设 $G := \mathbb{R}^{n_1+n_2} \times (\mathbb{Z}/2\mathbb{Z})^{n_1} \times (\mathbb{R}/\mathbb{Z})^{n_2}$，映射为

$$\begin{aligned}\varphi: \quad & (u_1, \ldots, u_{n_1+n_2}, \mu_1, \ldots, \mu_n, \theta_1, \ldots, \theta_{n_2}) \\ \mapsto & ((-1)^{\mu_1} e^{u_1}, \ldots, (-1)^{\mu_{n_1}} e^{u_{n_1}}, e^{2i\pi\theta_1} e^{u_{n_1+1}}, \ldots, e^{2i\pi\theta_{n_2}} e^{u_{n_1+n_2}})\end{aligned}$$

设 σ_1, σ_{n_1} 和 $\sigma_{n_1+1}, \ldots, \sigma_{n_1+n_2}$ 分别是 K 的实嵌入和复嵌入。我们考虑笛卡尔积环

$$E = \sigma_1(\mathcal{O}_K) \times \cdots \times \sigma_{n_1+n_2}(\mathcal{O}_K).$$

一个 E-理想是 E 的一个子-$\mathbb{Z}$-格 Λ，满足 $\forall x \in E, x\Lambda \subset \Lambda$。我们定义

$$\begin{aligned} g : G &\to \{E\text{-ideals}\} \\ x &\mapsto g(x)E. \end{aligned}$$

定义 35（ [22]中的例 5.3）令 $\mathcal{H} = \otimes_{i\in\mathbb{N}}\mathbb{C}|i\rangle$, $\mathcal{H}_Q = \otimes_{i=0}^{Q}\mathbb{C}|i\rangle$ 和 π_Q：$\mathcal{H} \to \mathcal{H}_Q$ 投影到前 Q 个量子比特。我们定义参数为 ν 的跨越编码，str_ν：$\mathbb{R} \to \mathcal{H}$, $\mathrm{str}_{m,\nu}$： $\mathbb{R}^m \to \mathcal{H}^m$ 以及 f： { 的格如下：

- $|\mathrm{str}_\nu(x)\rangle = \cos(\frac{\pi}{2}t)|k\rangle + \sin(\frac{\pi}{2}t)|k+1\rangle$, *where* $k = \lfloor x/\nu \rfloor$, $t = x/\nu - k$;
- $|\mathrm{str}_{m,\nu}(x_1, \ldots, x_m)\rangle = \otimes_{i=1}^{m}|\mathrm{str}_\nu(x_i)\rangle$;
- $|\mathrm{str}_{lattice,m,\nu}(L)\rangle = \gamma^{-1/2}\sum_{x\in L} e^{-\pi\|x\|^2/s}|\mathrm{str}_{m,\nu}(x)\rangle$
  with $\gamma = \sum_{x\in L} e^{-2\pi\|x\|^2/s^2}$.

注意，计算 $\mathrm{str}_{lattice,m,\nu}$ 且误差不超过 τ，需要 $O(m\nu$ 对数映射 $\tau)$ 个量子比特。在附录 *A* 中，我们提出了一个替代函数 f，可用于全实 K，且不使用跨越编码。

最后，定义 f 如下：

$$f : G \xrightarrow{g} \{E\text{-ideals}\} \xrightarrow{|\mathrm{str}\rangle} \{\text{quantum states}\} = \mathcal{H}_Q^m. \tag{3}$$

以下结果是 [22]中若干结果的合取。

定理 36（ [22]中的定理 5.5、D.4 和 B.3）设 K 是一个次数为 n、判别式为 D、单位秩为 m 且调节子为 R的数域。设 L ： = 对数映射 $\mathcal{O}_K^*$，它是方程 (3)中函数 f 的隐藏周期。

令 $s = 3 \cdot 2^{2n}\sqrt{nD}$和 $\nu = 1/(4n(s\sqrt{n})^{2n})$。那么 $f_5 = \otimes^5 f$是一个带有 $\varepsilon = 243/1024$、$\mathrm{Lip}(f) = a = \frac{\sqrt{\pi n}s}{4\nu} + 1$和 $r = s\sqrt{n}^{n-1}2\nu\sqrt{m}$的$(r, a, \varepsilon)$-预言机，其中$c$是一个可显式计算的常数。特别地， $\varepsilon < 1/4$, $r < \lambda_1(L)/6$ 和

$$\log_2 \mathrm{Lip}(f) = O\left(m^2 + m\log D\right).$$

证明。首先注意，若 $K=\mathbb{Q}$ 是实数域，或是虚二次域，则 $m=0$，无需进行任何计算。还要注意，$n/2-1 \le m \le n-1$，因此我们可以在渐近复杂度中互换 $O(m)$和 $O(n)$。

根据 [22, Th 5.5] f是一个 (r,∞,ε)-预言机，隐藏着格 $L=\log \mathcal{O}_K^*$，其中$r=(s\sqrt{n})^{n-1}2\nu\sqrt{n}$ 且 $\varepsilon'=3/4$。

根据 [22, Th D.4] f是 a-利普希茨的，其中 $a=\frac{\sqrt{\pi n}s}{4\nu}+1$。

根据 [22, Lem E.1] $f_5:=f\otimes f\otimes f\otimes f\otimes f$是一个 (5 Lip($f$) r, $(\varepsilon')^5$)-预言机，隐藏着相同的格。

特别地，$(\varepsilon')^5=243/1024<1/4$ 满足 CHSP 求解器的要求（参见 Def 22）。

条件 $r<\lambda_1(L)/6$ 成立，因为 $\lambda_1(L)\ge 1/2$，根据 [22, Lem B.3] 以及

$$r=O\left(\frac{s\sqrt{n}^{n-1}2}{4n(s\sqrt{n})^{2n}}\right)\le\frac{1}{2s^{n-1}}\le\frac{1}{6\cdot 2^n\sqrt{D}}<\frac{1}{12}.$$

最后，我们得到

$$\log_2 \mathrm{Lip}(f)=O(ms)=O(m^2+m\log D).$$

推论 37 设 K 是判别式为 D、单位秩为 m 的数域。对于任意误差界 $\tau>0$，存在一个量子算法，其运行时间为 $\mathrm{poly}(m,$对数映射 $D,$对数映射 $\tau)$，所用量子比特数为

$$N_{qubits}=O(m^5+m^4\log D)+O(m\log\tau^{-1})$$

对于一组单位 μ，ε_1，.....，ε_m，其中 μ 是单位根，其余单位均为无限阶

$$\mathcal{O}_K^*\simeq\mu^{\mathbb{Z}/\omega}\times\varepsilon_1^{\mathbb{Z}}\times\cdots\times\varepsilon_m^{\mathbb{Z}},$$

该算法输出 $\sigma_i(\log(\sigma_i(\varepsilon_j))_{1\le i\le m,\ 1\le j\le n}+\tau_{i,\ j}$，其中 $\tau_{i,\ j}\in\mathbb{R}$，满足$|\tau_{i,\ j}|\le\tau$。

证明。根据定理 36，存在一个函数 f，它隐藏了对数映射 $\mathcal{O}_K^*$，并且是一个 (a,r,ε)-预言机，满足 $r\le\lambda_1(L)/6$ 和 $\varepsilon<1/4$。根据定理 33，存在一个多项式时间算法，其使用的量子比特数如方程（2）所示。

在 CHSP 求解器中，将 f 的值存储在 Qm 个量子比特上，因此 $O(N_f)=O(Qm)$。接下来需要估计 Q。为此，关键部分是引理 21：

$$O(\log(1/\lambda_1^*))=\log_2(\|B_L\|)=O(m+\frac{1}{m}\log D).$$

我们利用了 L 存在一个 LLL约化基这一事实，这足以给出 $1/\lambda_1^*$ 的上界，但由于直到 CHSP 算法结束时我们才有 L 的基，因此无法计算它。将其代入方程（2），即可得到所述的值。

给定对数映射 $\mathcal{O}_K^*$ 后，我们只需计算 $\mu,\varepsilon_1,..,\varepsilon_m$。令 ω 为 K 的单位根个数，并回忆 ω 整除 D。将运行时间乘以 $\log D$ 后，我们可以枚举 D 的所有因数，因此可以假设我们已知 ω。

根据狄利克雷定理，一个满足 Vol(Log(ε_j)) $= R$ 且包含一个阶为 ω 的单位根的 m 元组，构成单位群的一组基。[9]

注 38 [10] 解析类数公式指出， $Rh = \sqrt{D}^{1+o(1)}$。关于 h 的分布，我们所知甚少，但没有任何猜想（例如 *Cohen-Lenstra* 启发式）与以下事实矛盾：对于一定比例的数域， $h = 1$ 和 $R = \sqrt{D}^{1+o(1)}(1)$。因此，$\max_i(\|$对数映射 $\varepsilon_i\|) \sim R^{1/m}$，且只有当 对数映射 $\tau = \Omega(\sqrt{D})$ 时，单位群才被完全确定。在这种情况下，*CHSP* 无法在多项式时间内计算完整的单位群。

不过请注意，对于经典算法，目前没有算法能够在不完全计算调节子的情况下，计算出调节子的部分信息（例如参见 *[12]*）。因此，本文研究的算法表明，在单位群问题上存在量子优势。

例*1.* 1. 如果 $K = \mathbb{Q}(\zeta_n)$，且素数 n，那么 $m = (n-1)/2$，并且 $\log_2 D = (m-2)\log_2 m$。如果应用算法 1，但未注意到这是一个非常特殊的情形，该算法会使用 $O(m^5 \log m)$ 个量子比特。2. 如果 K 是一个库默尔扩张，即对于某个不含幂因子 D 的整数，有 $K = \mathbb{Q}(\sqrt[n]{D})$，那么 $O(\log \operatorname{disc} K) = O(\log D)$，量子比特数为 $O(n^5 + n^4 \log D)$。n 和 D 的值彼此独立。当 n 固定时，也就是实二次域的情形，量子比特数为 $O(\operatorname{disc} K)$。当 $n \sim \log D$ 时，量子比特数为 $(\log D)^5/(\log\log D)^4$。

与 Shor算法形成对比的是，本算法使用了大量量子比特；Shor算法在因数分解和离散对数问题中需要 $2m + O(1)$ 个量子比特，在椭圆曲线离散对数问题中则需要 $7m + O(1)$ 个。文末“结论与研究方向”一节 [19, Sec 1] 指出，即使不考虑对数映射中的复杂度，并在量子傅里叶变换中采用近似技术，该算法仍需要 $\Omega(m^3)$ 个量子比特。关于量子算法情形下可提出的安全级别，我们在附录B中作非正式讨论。

3.3 利用自同构

本节讨论不使用分圆单位的分圆域情形。我们改为利用 K 具有伽罗瓦自同构这一事实。

本节的主要影响是降低 Buchmann-Pohst 步骤（该步骤并非量子步骤）的实际成本。事实上，这一步是对格 L 应用 LLL算法；该格具有更多代数结构：它是一个整数 k 的 $\mathbb{Z}[\zeta_k]$-模。从渐近角度来看，Fieker 和 Stehlé [24] 证明，若要找到一个新的 L 的 $\mathbb{Z}[\zeta_k]$-基，使其按照稍后将说明的意义更短，那么遵循以下步骤可获得最佳渐近复杂度（相差一个多项式因子）：

[9] 这里我们使用了不同于 [22] 中的证明；后者将定义在任意阿贝尔群上的 CHSP 情形归约为定义在 $\mathbb{R}^m$上的 CHSP 情形。[10] 我们感谢 Bill Allombert 指出了这一问题。

1. 忽略 $\mathbb{Z}[\zeta_k]$-模结构；2. 约化底层 $\mathbb{Z}$-格的基；
3. 重新计算 $\mathbb{Z}[\zeta_k]$-模结构。

不过，从实际角度来看，我们将说明，直接使用 $\mathbb{Z}[\zeta_k]$-模结构更快。

结构 $\mathbb{Z}[G]$-格。下文中，具有某个环 R 上的 R-模结构的格称为 R-格。当 $G :=\mathrm{Aut}(K/\mathbb{Q})$（即伽罗瓦自同构群）是阿贝尔群时（分圆域正是如此），$\mathcal{O}_K^*/(\mathcal{O}_K^*)_{\mathrm{tors}}$ 具有 $\mathbb{Z}[G]$-格结构。事实上，对于 $\sigma \in G$ 和 $u \in \mathcal{O}_K^*$，我们定义 $\sigma \cdot u = \sigma(u)$；对于 $\mathbb{Z}[G]$的一组环生成元 $\sigma_1, \ldots, \sigma_k \in G$，以及任意的 $\lambda = \sum_{e_1,\ldots,e_k} c_{e_1,\ldots,e_k}\sigma_1^{e_1} \cdots \sigma_k^{e_k}$，我们定义

$$u^{\lambda} = \prod_{e_,\ldots,e_k} (\sigma_1^{e_1}(u) \cdots \sigma_k^{e_k}(u))^{c_{e_1,\ldots,e_k}}.$$

引理 39 设 $G :=$ 自同构群$(F/\mathbb{Q})$ 且 $\sigma \in G$ 的阶为 k。

1. 那么 $\mathcal{O}_F^*$ 是一个 $\mathbb{Z}[\zeta_k]$-格。
2. 如果 k 不具有如下形式

$$k = 2^{\varepsilon}\prod p_i^{e_i} \ \ with\ \varepsilon \in \{0,1\}\ and\ p_i \equiv 1 \pmod 4$$

那么，对于 k的某个因子 d， $\mathcal{O}_F^*$ 是一个 $\mathcal{O}_{\mathbb{Q}(i\sqrt{d})}$-格。

证明。1. 若 $\sigma \in G$，则每个 $\mathbb{Z}[G]$-格都是一个 $\mathbb{Z}[\sigma]$-格。为了证明 Φ_k是与 σ相关联的自同态的最小多项式，设 $u \in \mathcal{O}_F^*$ 并令 $v = u^{\Phi_k(\sigma)}$。于是 $v^{\sigma-1} = u^{\sigma^k-1} = u/u = 1$，进而 $\sigma(v) = v$。但是 $\sigma \in$自同构群$(F/\mathbb{Q})$，所以 $v \in \mathcal{O}_F^* \bigcap \mathbb{Q} = \pm 1 \in (\mathcal{O}_F^*)_{\mathrm{tors}}$。等价地，$\Phi_k(\sigma)(u) \in (\mathcal{O}_F^*)_{\mathrm{tors}}$，且 Φ_k 是 σ的湮灭多项式。由于它不可约，因此是 σ的最小多项式。但 ζ_k 具有相同的最小多项式，所以 $\mathbb{Z}[\sigma] \simeq \mathbb{Z}[\zeta_k]$。

2. 一个经典结果是，若对素数 p 令 $p^* = (-1)^{\frac{p-1}{2}}p$，则

$$\mathbb{Q}(\sqrt{p^*}) \subset \mathbb{Q}(\zeta_p).$$

当 $k' \mid k$ 时，有 $\mathbb{Q}(\zeta_{k'}) \subset \mathbb{Q}(\zeta_k)$。因此，若 k 能被 4 整除，或是一个满足 $p \equiv 3 \pmod 4$ 的素数，则在引理的陈述中可以取 $d = 4$ 或 $d = p$。

范数欧几里得环 $\mathcal{O}_K$。回顾一下， $\mathbb{Z}[i] = \mathbb{Z}[\zeta_4]$ 和$\mathcal{O}_{\mathbb{Q}(i\sqrt{3})} = \mathbb{Z}[\zeta_3]$ 是范数欧几里得环：令 $k = 3$ 或 4，为了用 $b \in \mathbb{Z}[\zeta_k]$ 除 $a \in \mathbb{Z}[\zeta_k]$，将 $a/b \in \mathbb{Q}[\zeta_k]$ 的每个坐标四舍五入，并将所得结果记为 q，然后令 $r = a - bq$；如果对于所有 a、b， $N_{K/\mathbb{Q}}(r) < N_{K/\mathbb{Q}}(b)$，我们就称 $\mathbb{Z}[\zeta_K]$ 是范数欧几里得的。Napias [42] 证明了，LLL算法可以自然地扩展到范数欧几里得环 $\mathbb{Z}[i]$ 和 $\mathbb{Z}[\zeta_3]$。

定义 310（范数欧几里得环， [14], 的命题 1.2.5， [32]的定义 2）数域 K 的欧几里得最小值为

$$\mathfrak{M}_K := \max_{x\in K} \min_{y\in\mathcal{O}_K} N_{K/\mathbb{Q}}(x-y).$$

特别地，当 K 是虚二次域时， $\mathfrak{M}_K =$最大值$_{x\in\mathbb{C}}$最小值$_{y\in\mathcal{O}_K} |x-y|^2$。如果 $\mathfrak{M}_K < 1$ ，我们称 K 为范数欧几里得的。

例 *2*（ *[14],*命题 *3*，见 *[32]*）。以下域是范数欧几里得的：

– $\mathbb{Q}(\zeta_k)$，其中 $k \in \{1,3,4,5,7,8,9,11,12,15,16,20,24\}$；– $\mathbb{Q}(i\sqrt{d})$，其中 $d \in \{1,2,3,7,11\}$。

定义 311 （$\mathcal{O}_K$-*LLL*约化基）设 $K = \mathbb{Q}(i\sqrt{d})$ 为一个虚二次域。我们将 K 与其在 $\mathbb{C}$ 中的嵌入等同，使得 $\sqrt{-d}$ 映射为虚部为正的复数。

我们考虑 $\mathbb{C}^m$ 和 $\langle\cdot,\ \cdot\rangle$ 的点积，并设 $\|x\| = \sqrt{\langle x,\ x\rangle}$，它是一个范数。

设 b_1， b_2， ……， b_m 是 $\mathbb{C}^m$中的线性无关向量。我们称 L : $= \oplus_{j=1}^m \mathbb{Z}[\zeta_k]b_j$为 $\mathbb{Z}[\zeta_k]$-格，并称 $\{b_j\}$ 为一组基。令 $\{b_1^* = b_1, b_2^*, \ldots, b_m^*\}$ 为 $\{b_j\}$ 关于点积的格拉姆-施密特正交化。设$\mu_{i,\ j} =\langle b_i,\ b_j^*\rangle / \left\|b_j^*\right\|^2$。若以下两个条件成立，我们称 $\{b_j\}$关于 $\mathfrak{M}_K < \delta < 1$是 $\mathcal{O}_K$-*LLL*约化的：

1. $\|\mu_{i,j}\| \le \mathfrak{M}_K$ *for* $1 \le j < i \le m$; *(size reduced)*
2. $\|b_i^*\|^2 + \|\mu_{i,i-1}\|^2 \left\|b_{i-1}^*\right\|^2 \ge \delta \left\|b_{i-1}^*\right\|^2$. *(Lovász condition)*

在近期的两项研究中，金和李 [32] 以及卡缪 [14] 将LLL算法推广到了所有范数欧几里得环 $\mathbb{Z}[\zeta_k]$ 和 $\mathcal{O}_{\mathbb{Q}(i\sqrt{D})}$。特别地，当 K 是一个范数欧几里得虚二次域时， [14, 定理 1.3.8] 指出， L 的任意 $\mathcal{O}_K$-约化基 b_1， ……， b_m 都满足

$$\|b_j\| \le \left(\frac{1}{\delta - \mathfrak{M}_K}\right)^{j-1} (\det L)^{1/m}.$$

$\mathcal{O}_K$-LLL算法的一个直接应用是我们在算法4中提出的 Buchmann-Pohst 算法的 $\mathcal{O}_K$ 变体。

定理312（将 $\mathcal{O}_K$格上的定理3.1改编自 [13]）算法*4*正确，并且会在经典计算机上经过 $O\left((k+m)^6(m+$对数映射$(D^{1/m}/\mu))\right)$ 次运算后终止。

证明。 该证明直接照搬了在$\mathbb{Z}$-格情形下的正确性证明。由方程3.3可知，若 $b_1, \ldots, b_m$ 是一个格的约化基，且其行列式为 $L \le D$，则 max$(\|b_1\|, \ldots, \|b_m\|) \le B$。在 [13, 命题2.2] 的证明中，利用这一点来说明由这些列生成的格的连续极小值

算法4 Buchmann-Pohst 关于 $\mathcal{O}_K$

输入： 一个 $\mathcal{O}_K$-格 $L \subset \mathbb{C}^m$，由 $g_1, \ldots, g_k \in$的近似值 $\widetilde{g_1}, \ldots, \widetilde{g_k} \in \mathbb{C}^m$ 给出，这些近似值 $\mathcal{O}_K$-张成 L； $\lambda_1(L)$的下界 μ；det L 的上界 D **输出：** 在 $\mathcal{O}_K$ 上关于 L的基 $b_1, \ldots, b_m$ 的近似值

1: $B \leftarrow c_K^m D^{1/m}$; $C \leftarrow (B/\mu)^m \gamma_m^{1/2}$; $\tilde{M} \leftarrow (k\sqrt{m}/2 + \sqrt{k})C$; $q \leftarrow \lceil \log_2 \left((\sqrt{mk}+2)\tilde{M} 2^{(k-1)/2}/\mu \right) \rceil$

2：将左侧矩阵进行 $\mathcal{O}_K$-LLL 约化，得到右侧矩阵

$$\begin{pmatrix} \boxed{\lfloor \widetilde{g_1} 2^q \rfloor} & \cdots & \boxed{\lfloor \widetilde{g_k} 2^q \rfloor} \\ \boxed{1} & & \\ & \ddots & \\ & & 1 \end{pmatrix} \qquad \left(\begin{array}{ccc} \boxed{\widetilde{c_1}} & \cdots & \boxed{\widetilde{c_k}} \\ \boxed{\mathbf{m}_1} & \cdots & \boxed{\mathbf{m}_k} \end{array} \right)$$

3：**返回** $\widetilde{c}_{k-m+1}, \widetilde{c}_2, \ldots, \widetilde{c_k}$

˜算法4中矩阵的范数上界为 M。我们使用 [14, 中 $\mathcal{O}_K$-约化基范数的上界（命题1.3）8] ，而非 [38, (1.12)] ，以得到等价的 [13, 式（9）]:

$$\|\widetilde{c_j}\| \leq 2^{(k-1)/2}\tilde{M} \qquad 1 \leq j \leq k-m. \tag{4}$$

对于任意向量 $(\widetilde{c}, \mathbf{m}) \in \oplus_{j=1}^k \mathbb{Z}(\tilde{c_j}, \mathbf{m}j)$，若 $\mathbf{m} = (\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(k)})$ 不是 $b_1, \ldots, b_k$ 的关系，即 $\mathbf{m} \notin L^*$，也就是说 $\sum_{j=}^k \mathbf{m}^{(j)} \widetilde{b}_j \neq 0.$，那么可以逐字照搬 [13, Th 3.1] 的证明中的论证，从而得到:

$$\|\widetilde{c_j}\| > 2^{(k-1)/2}\tilde{M} \qquad 1 \leq j \leq k. \tag{5}$$

由于方程 (4) 和 (5) 对于 $1 \leq j \leq k-m$ 相互矛盾，唯一的可能性是 $\mathbf{m} \in L^*$。我们对矩阵进行相同的变换，只是用 $\backsim b_j$ 替代 b_j，并将此情形下替代 $\widetilde{c_j}$ 的向量称为 c_j。我们已经证明了 $c_1 = c_2 = \cdot = c_{k-m} = 0$。

由于对矩阵各列进行的 LLL算法变换是可逆的，因此这些变换保持矩阵任意行子集的秩不变。该矩阵前 m 行的秩为 $\dim L = m$。由于 $c_1 = c_2 = \cdot = c_{k-m} = 0$，向量 $c_{k-m+1}, \ldots, c_k$ 构成 L 的一组基。

在 $\mathbb{Z}[i\sqrt{d}]$ 作为一项实用改进，实现中的大O记号中隐藏的常数与 $\log \delta$ 成反比，而后者与 $\log(\delta - \mathfrak{M}_K)$ 成反比。因此，差异仅为几个百分点。时间对秩的依赖是四次方的，所以在相同精度下，我们用 $2^4 = 16$ 倍更少的复数运算替代了大量高精度实数运算。最后，利用 Karatsuba技巧，复数乘法的代价是 3 次实数乘法，因此总体增益为 $16/3 \approx 5.33$ 倍。对 $\mathbb{Z}[i\sqrt{d}]$-LLL 的 [23] 实现表明，该约化

$\mathcal{O}_K$-格比在遗忘其代数结构、仅使用 $\mathbb{Z}$-格的底层结构时快 ≈ 5 倍：他们使用了 $\mathcal{O}_K = \mathbb{Z}[i]$，基向量的坐标为 512 位，并且$\delta = 0.99$ 都在 $\mathbb{Z}$ 和 $\mathbb{Z}[i]$ 上。[11]

密码学中的后果。 鉴于我们目前的贡献，由自同构带来的加速比为 5.33。将其扩展到更大类的$\mathcal{O}_K$-环仍是一个开放问题（例如， [37] 研究能否通过在维数为 $[\mathcal{O}_K$ 的空间中求解最近向量问题实例，将 LLL 算法扩展到任意 $\mathcal{O}_K$ $\mathbb{Z}]$。）。如果 Buchmann-Pohst 能够得到扩展，那么可以达到的最佳加速比为 $[\mathcal{O}_K :\ \mathbb{Z}]^3$。事实上，LLL算法对维数的依赖是四次的，而 $\mathcal{O}_K$ 上的算术运算成本是二次的；如果采用快速算术运算，则为准线性。

由自同构带来的加速比与自同构的数量呈多项式关系。这是一种常见情况，因为在因数分解、有限域中的离散对数以及椭圆曲线上的离散对数的经典算法中，自同构也会带来类似的加速（参见例如 [30,第 4 节、3],[5, 第 5 节3]）。例如， [20] 中利用曲线自同态对 ECDSA 发起的攻击，在一个指数复杂度的算法中实现了多项式加速。密码学界对此的反应是建议使用具有自同态的椭圆曲线，因为当时人们认为它们的影响是良性的，例如 [25]。类似地，如果认为给定次数和行列式的类群的安全性足够高，我们建议使用具有自同构的域来加速协议；例如，基于有限域中离散对数的密码系统 XTR 就采用了这种方法 [39]。

4 一种使用分圆单位的新算法

当 $K = \mathbb{Q}(\zeta_m)$ 对任意整数 m 成立时，我们将分圆单位群定义为由 -1、ζ_m 和 $\zeta_m^j - 1$ （其中 $j = 1, 2, \ldots, m-1$）生成的 K^* 的子群 C，再与单位群 $\mathcal{O}_K^*$取交。我们沿用 [18, 第 3], 节中的记号；特别是， m 和 k 的含义与其他章节中的不同。分解因子 $m = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$，并对每个指标 i，令 $m_i = m/p_i^{\alpha_i}$。

对于 $j = 1$、$\ldots$、$m-1$，我们定义

$$v_j = \begin{cases} 1 - \zeta_m^j, \text{ if for all } i \text{ we have } m_i \nmid j, \\ \frac{1-\zeta_m^j}{1-\zeta_m^{m_i}}, \text{ otherwise for the unique } i \text{ such that } m_i \mid j. \end{cases}$$

4.1 无条件结果

引理 41（定理 4.2，见 [35]） 格 $M :=$ 对数映射C 具有生成元系 $\{b_j =$对数映射$(v_j) \mid j = 1$、2，$\ldots\ldots$，$m-1\}$。

[11] 在 [32] 中的 $\mathbb{Z}[\zeta_k]$-LLL 情形下也获得了类似的加速：一个例子在 $\mathbb{Z}[\zeta_k]$ 上耗时 20 秒，而在 $\mathbb{Z}$上耗时 75 秒。

引理 42（引理 3.5，见 [18]） 对于任意整数 $j \in \{1, 2, \ldots, m-1\}$，有 $\left\|1-\zeta_m^j\right\|_2 = O(\sqrt{m})$。因此 $\|B_M\|_2 = O(\sqrt{m})$。

定理 43 算法 *1* 在 $\mathbb{Q}(\zeta_m)$ 中以 $\mathrm{poly}(m)$时间计算单位群的一个基，并使用 $O(m^2$ 对数映射 $m)$ 量子比特。

证明。 算法 1 唯一的量子步骤是对偶格采样器，其参数为 $\delta = \|B_L\| / \lambda_1^*(L)$ 和 $\eta = 1/k^2$，其中

$$k = \log_2(\sqrt{m}\,\mathrm{Lip}(f)) + \log_2(\det L))$$

在算法的步骤 1 中给出。在本节中，m 不一定等于 $\mathbb{Z}[\zeta]^*$ 的秩 m'，但 $O(m') = O(m)$，因此我们交替使用 m 和 m'。根据引理 32，所使用的采样器采用了 $O(Qm)$ 个量子比特，其中 Q 如下。我们将 $a \ll b$ 写作 $a = O(b)$。我们使用定理 36：log Lip$(f) \ll m^2 + m\log D$，然后 $k \ll m^2 + m\log D$。我们还使用引理 42：$\log 1/(\delta\lambda_1^*(L)) \ll \log(1/\|B_M\|) \ll \log m$，以及 $\log D = (m-2)\log m \ll m\log m$。

$$\begin{aligned} Q &\ll m\log(m\log\tfrac{1}{\eta})) + \log\tfrac{1}{\delta\lambda_1^*(L)} + \log(\mathrm{Lip}(f)) \\ &\ll (m\log m + m\log\log k) + \log m + m^2 + m\log D \\ &\ll m\log m + m\log\log D + \log m + m^2 + m\log D \\ &\ll m^2 + m\log D \\ &\ll m^2\log m. \end{aligned}$$

这应与一般数域所需的 $O(m^5 + m^4\log D)$ 个量子比特（推论 37）以及若不利用分圆单位而直接使用算法 3 时分圆域所需的 $O(m^5\log m)$ 个量子比特（例 1）相比较。

注 44 只要单位格含有一个具有短基的子格，我们的改进就适用，并不限于分圆域，或更一般地说，阿贝尔域。例如，*Kihel [31]* 提出了一族具有二面体伽罗瓦群的数域，其中存在一个由短单位构成的满秩子群。尽管预计所有这些情形都能加速，但分圆域情形更加特殊。事实上，参数 $\delta \le \|B_M\| / \lambda_1^*(L)$ 在很大程度上取决于指数 $[L : M]$。根据 *[36, 例 8.5]*（另见 *[17, 定理 2.8]*），在素数导子分圆域的情形下，$[L : M] = h^+(m)$。根据猜想 *1*，这个值很小，因此在这种情形下，满秩子群的影响更为显著。

4.2 最近一个关于分圆域的猜想所带来的推论

记 $h(m)$ 为 $\mathbb{Q}(\zeta_m)$ 的类群，记 $h^+(m)$ 为其最大实子域 $K := \mathbb{Q}(\zeta_m + \zeta_m^{-1})$ 的类群，并令 $h^-(m) = h(m)/h^+(m)$。民间猜想认为，$h^-(m)$ 很大但容易计算，而 $h^+(m)$ 很小但难以计算。最近的一项研究汇总了已有猜想，并进一步推进了数值计算，因此可以对这一民间猜想给出精确表述。

猜想 *1*（*[18]* 中的假设*2*）。对于所有整数 m，

$$h^+(m) \le \mathrm{poly}(m)$$

对于一个固定的多项式多项式。

该猜想的推论：无需量子计算机即可计算实三次域的类数

Marie-NicoleGras [26] 提出了一种算法，可根据 h 的上界，在多项式时间内计算循环三次域的类数。如果 m 是循环三次域的导子，那么猜想 1 意味着 h 关于 m 是多项式级的。另请注意，Schoof [47] 为 $\mathbb{Q}(\zeta_m)^+$（其中 m 为素数）提出了一种算法；当已知 $h^+(m)$ 的较小上界时，该算法速度更快。

将单位群计算归约为隐藏子群问题

定理 45 在猜想 *1* 的假设下，可以通过一个隐藏子群问题算法，在多项式时间和空间内计算 $K = \mathbb{Q}(\zeta_m)^+$ 的单位群及其类群。

证明。令 $N = \text{poly}(m)!$，并注意到猜想 1 蕴含 $\mathcal{O}_K^* \subset C^{1/N}$。令 $m' = \varphi(m)/2-1$，并令 $\varepsilon_1, \ldots, \varepsilon_{m'}$ 为 $C/\langle\zeta_m, -1\rangle$ 的一组基。注意，可以利用引理 41 的向量 v_j 通过厄米标准形算法计算它。由于 K是全实的，其唯一的单位根为 ± 1。对于任意 $c \in C$，我们称 c的向量化为 vect(c)，并将其记作 vect(c)；它是 $(m'+1)$ 元组 $(e_0, e_1, \ldots, e_{m'}) \in \mathbb{Z}/2\mathbb{Z} \times (\mathbb{Z}/N\mathbb{Z})^{m'}$，满足 $c = (-1)^{e_0} \prod_{j=1}^{m'} \varepsilon_j^{e_j}$。

我们设

$$\begin{aligned} f : \mathbb{Z}/2\mathbb{Z} \times (\mathbb{Z}/N\mathbb{Z})^{m'} &\to \left\{ \begin{array}{c} \text{canonical representations} \\ \text{of Kummer extensions} \end{array} \right\} \\ (e_0, e_1, \ldots, e_{m'}) &\mapsto K(\zeta_N, \sqrt[N]{(-1)^{e_0} \prod_{j=1}^{m'} \varepsilon_j^{e_j}})) \end{aligned} \tag{6}$$

我们声称， f 的周期为

$$\text{period}(f) = \{\text{vect}(u) \mid u \in (\mathcal{O}_K)^*\}.$$

事实上，令 (e_j) 和 (z_j) 满足 $f(e_0, \ldots, e_{m'}) = f(e_0 + z_0, \ldots, e_m + z_{m'})$，并设 $\varepsilon = (-1)^{e_0} \prod_j \varepsilon_j^{e_j}$ 和 $\zeta = (-1)^{z_0} \prod_j \varepsilon_j^{z_j}$。

我们有 $K(\zeta_N, \sqrt[N]{\varepsilon}) = K(\zeta_N, \sqrt[N]{\varepsilon\zeta})$。令 $d := [K(\zeta_N, \sqrt[N]{\varepsilon}) : K(\zeta_N)]$。那么，库默尔扩张 [34, 的同构判据（第 58] 页）指出， $\sqrt[N]{\zeta}$ 属于 $K(\zeta_N)$。但是 $\sqrt[N]{\zeta} \in \mathbb{R}$，且 $K(\zeta_N)$ 的最大实子域为 K，因此 $\sqrt[N]{\zeta} \in \mathcal{O}_K^*$。反过来，若 $\sqrt[N]{\zeta} \in \mathcal{O}_K^*$，显然 $(z_0, \ldots, z_{m'})$ 是 f 的一个周期。

最后，为了计算算法的复杂度，我们记 $a = \mathcal{P}(b)$ 为$a = b^{O(1)}$，并记 $a = \text{polylog}(b)$ 为 $a = (\log b)^{O(1)}$。根据 [33], ，隐藏子群问题的代价为 polylog(N)。由于对于任意 n， $n! \leq n^n$，我们有

$$\text{time} = \text{polylog}(N) = \mathcal{P}(\text{poly}(m) \log(\text{poly}(m))) = \mathcal{P}(\mathcal{P}(m)) = \mathcal{P}(m).$$

5 结论与开放问题

1. 单位群算法的发展路径与类群等其他问题的算法相似。例如，Buchmann 算法会同时计算这两个群，而类群的量子算法 [9] 是 Hallgren 等人提出的算法的推广 [22]。研究类群的量子比特数量如何随类数上界、类群中离散对数的计算等因素变化，并对其进行精确估算，是很有意义的。

2. 小对偶基的思想可以推广到：a) 所有阿贝尔域；b) 自同构群为单群且已知单位的伽罗瓦域，例如 [31]。3. 能否利用任意阶的自同构，取决于能否针对 $\mathcal{O}_K$ -格设计 Buchmann-Pohst 算法。可以探索这样一种算法：它在平均意义上减小范数，但可能在局部增大范数。4. 量子算法具有概率性，因此其输出未经验证。能否设计一种经典算法，证明给定的一组单位生成的子群是 ℓ-饱和的？5. 量子计算机技术非常新，目前尚不清楚给定数量的逻辑量子比特对应多少物理量子比特。隐藏子群问题算法所用的量子门是否错误更少，因而需要的纠错也更少？精确分析这一问题超出了本文的讨论范围。

一种隐藏单位的替代函数

我们考虑 K 为全实的情形。和之前一样，我们称 n 为它的次数。设 P 是定义 K 的多项式，并设 α_1，....., α_n 是 P 在 $\mathbb{R}$ 中的 n 根。令

$$\begin{aligned} w : \{P \in \mathbb{R}[x] \mid \deg P \leq n-1\} &\to \mathbb{R}^n \\ P &\mapsto (P(\alpha_1), \ldots, P(\alpha_n)). \end{aligned}$$

显然， w 是线性同构。

假设 disc(f) 无平方因子，则 $\mathcal{O}_K = \mathbb{Z}[x]/P$。我们定义

$$\begin{aligned} f : \quad \mathbb{R}^n &\to (\mathbb{R}_n[x] \bmod \mathbb{Z}_n[x]) \times (\mathbb{R}_n[x] \bmod \mathbb{Z}_n[x]) \\ (x_1, \ldots, x_n) &\mapsto (w^{-1}(e^{x_1}, \ldots, e^{x_n}), w^{-1}(e^{-x_1}, \ldots, e^{-x_n})) \bmod \mathbb{Z}_n[x]^2. \end{aligned}$$

我们断言， f 的周期集合构成一个格 Λ，使得

$$2 \operatorname{Log} \mathcal{O}_K^* \subset \Lambda \subset \operatorname{Log} \mathcal{O}_K^*.$$

确实，令 ε 属于 $\mathcal{O}_K^*$，并设 $(x_1, \ldots, x_n)$ =对数映射(ε^2)。那么 $w^{-1}(e^{x_1}, \ldots, e^{x_n})$ 是 ε^2 在 K 的正规基下的表示。由于 ε^2 属于 $\mathcal{O}_K = \mathbb{Z}[x]/\langle P \rangle$，其坐标均为整数。类似地， $w^{-1}(e^{-x_1}, \ldots, e^{-x_n})$ 是 ε^{-2} 在正规基下的表示。由于 $\varepsilon^{-2} \in_O O_K^*$，其坐标均为整数。

反过来，设 $(x_1, \ldots, x_n)$ 满足 $f(x_1, \ldots, x_n) = (0, \ldots, 0)$，并令$\varepsilon = w^{-1}(e^{x_1}, \ldots, e^{x_n})$。由于 $f(x_1, \ldots, x_n) = 0$， ε 在正规基下的坐标均为整数，因此 $\varepsilon \in \mathcal{O}_K$。类似地，$\varepsilon^{-1} \in \mathcal{O}_K$，所以 ε 是单位。

最后，给定 Λ，求解一个关于 $\mathbb{Z}/2\mathbb{Z}$ 的线性方程组，以求出 $\operatorname{Log} \mathcal{O}_K^*$。由于$K$ 是全实的，其单位根为 ± 1 ，而 $\operatorname{Log} \mathcal{O}_K^*$ 可以完全确定$\mathcal{O}_K^*$。

B 关于量子安全性的讨论

NIST 后量子挑战 [43] 对量子计算机计算资源的定义持开放态度：

> “在这里，计算资源可以采用多种不同的指标来衡量（例如，经典基本操作的数量、量子电路规模等）”

Mosca 等人开展的一项研究 [41] 显示，部分专家认为，中期内不应考虑使用数十亿个量子比特的攻击，而应防范使用数百个量子比特的攻击。经典算法也存在类似情况：2015年，标准化机构（NIST、ANSSI 等）建议使用 2048 位 RSA 密钥。然而，在 [1] 作者准确估算出，利用他们在百万个核心上运行的数域筛法实现，可以破解 RSA 1024；他们还调查发现，在抽样的百万台+ 服务器中，98% 使用 RSA 1024。此外，超过 30% 的服务器支持 RSA 768，而该密钥自 2009 年起就已被破解。

因此，有必要更精确地描述量子资源。我们提出了一个关于时间复杂度和量子比特数量的分类，如表 1 所示（有关针对各种公钥原语的量子攻击综述，参见例如 [8]）。从这个角度来看，基于高次数非分圆数域的类群的可验证延迟函数 [50] 比 RSA 更安全。

	多项式 time	次指数 time	指数 time
$O(m)$ 个量子比特	因数分解 离散对数 椭圆曲线离散对数 IQC		格 纠错码
$O(m^b$ 对数映射 $m)$ 量子比特	高次数类群计算		
超多项式空间		同源	

表 1：根据量子攻击所需资源对密码学原语进行分类：因数分解、离散对数、椭圆曲线（EC）离散对数、虚二次域的整环的类群（IQC）、超奇异同源、格、纠错密码系统和类群（CGP）计算。

参考文献

1. David Adrian、Karthikeyan Bhargavan、Zakir Durumeric、Pierrick Gaudry、Matthew Green、J AlexHalderman、Nadia Heninger、Drew Springall、Emmanuel

Thomé、Luke Valenta 等。《不完善的前向保密：Diffie-Hellman 在实践中如何失效》。载于第22届ACM SIGSAC计算机与通信安全会议论文集，2015年。2. László Babai。《关于 Lovász格约简与最近格点问题》。Combinatorica，6(1)：1–13，1986年。3. Wojciech Banaszczyk。《数论几何中若干转移定理的新界》。Mathematische Annalen，296(1)：625–635，1993年。4. Gustavo Banegas、Daniel J Bernstein、Iggy van Hoof 和 Tanja Lange。《二元椭圆曲线的具体量子密码分析》。IACR密码硬件与嵌入式系统汇刊，第451–472页，2021年。5. Razvan Barbulescu、Pierrick Gaudry、Aurore Guillevic 和 François Morain。《改进非素数阶有限域上离散对数问题的数域筛法》。载于密码学进展—Eurocrypt 2015，计算机科学讲义第9056卷。施普林格，2015年。6. Stephane Beauregard。《使用 2n+ 3 个量子比特实现 Shor算法的电路》。《量子信息与计算》，3(2)：175–185，2003年。7. Daniel J Bernstein、Tanja Lange、Chloe Martindale 和 Lorenz Panny。《用于 CSIDH 的量子电路：优化同源映射的量子求值》。载于密码学进展—Eurocrypt 2019，第409–441页。施普林格，2019年。8. J-F Biasse、X Bonnetain、E Kirshanova、A Schrottenloher 和 F Song。《用于攻击经典密码学和后量子密码学中困难性假设的量子算法》。2022年。9. Jean-François Biasse 和 Fang Song。《针对依赖于在 $\mathbb{Q}(\zeta_{p^n})$ 中寻找理想短生成元之困难性的方案的量子攻击》。技术报告，技术报告 CACR 2015-12，2015年。10. Jean-François Biasse 和 Fang Song。《用于计算任意次数数域的类群并求解主理想问题的高效量子算法》。载于第二十七届ACM-SIAM离散算法年会论文集，第893–902页。SIAM，2016年。11. Jean-François Biasse 和 Fang Song。《针对依赖于在 $\mathbb{Q}(\zeta_{2^k})$ 中寻找理想短生成元之困难性的方案的量子攻击》。《数学密码学杂志》，13(3)，2019年。12. Johannes Buchmann。《一种用于确定代数数域的类群和调节子的亚指数算法》。巴黎数论研讨班，1989(1990)：27–41，1988年。13. Johannes Buchmann 和 Michael Pohst。《根据一组生成向量计算格基》。载于 Eurocal'87：欧洲计算机代数会议，计算机科学讲义第378卷，1989年。14. Thomas Camus。《代数格的算法方法》。博士论文，格勒诺布尔阿尔卑斯大学，2017年。15. Guilhem Castagnos 和 Fabien Laguillaumie。《基于……的线性同态加密》。载于密码学专题—CT-RSA 2015，计算机科学讲义第9048卷，2015年。16. H Cohen。《计算代数数论教程》。数学研究生教材，1996年。17. Ronald Cramer、Léo Ducas、Chris Peikert 和 Oded Regev。《恢复分圆环中主理想的短生成元》。载于密码技术理论与应用国际年会论文集，第559–585页。施普林格，2016年。

18. Ronald Cramer、Léo Ducas 和 Benjamin Wesolowski。在量子多项式时间内求解分圆理想格中的中等长度短向量。《ACM学报》，68(2):1–26，2021年。19. Koen de Boer、Léo Ducas 和 Serge Fehr。连续隐藏子群问题的量子复杂度。载于《密码学进展—Eurocrypt 2020》，计算机科学讲义第12106卷，2020年。20. Iwan Duursma、Pierrick Gaudry 和 François Morain。利用自同构加速曲线上的离散对数计算。载于《密码学进展—Asiacrypt'99》，计算机科学讲义第1716卷，1999年。21. Kirsten Eisenträger、Sean Hallgren、Alexei Kitaev 和 Fang Song。一种计算任意次数数域单位群的量子算法。载于第四十六届ACM计算理论年会论文集，第293–302页，2014年。22. Kirsten Eisenträger、Sean Hallgren、Alexei Kitaev 和 Fang Song。提交给STOC 2014的《一种计算任意次数数域单位群的量子算法》的长篇版本，2019年。在线可查：https://www.cse.psu.edu/~sjh26/units-stoc-submission.pdf。23. Philippe Elbaz-Vincent 和 Etienne Marcatel。将fplll库扩展至厄米格。《ACM计算机代数通讯》，55(2)，2021年。24. Claus Fieker 和 Damien Stehlé。数域上的格的短基。载于算法数论—ANTS IX，第157–173页。施普林格，2010年。25. Robert P Gallant、Robert J Lambert 和 Scott A Vanstone。具有高效自同态的椭圆曲线上的快速点乘。载于密码学进展-CRYPTO 2001，计算机科学讲义第2139卷。施普林格，2001年。26. Marie Nicole Gras。循环三次数域扩张的类数和单位的数值计算方法与算法。《纯粹与应用数学杂志》，1975年。27. Sean Hallgren。用于计算数域单位群和类群的快速量子算法。载于第三十七届ACM计算理论年会论文集，2005年。28. Sean Hallgren。用于求解佩尔方程和主理想问题的多项式时间量子算法。《ACM学报》（JACM），54(1):1–19，2007年。29. Thomas Häner、Martin Roetteler 和 Krysta Svore。使用基于Toffoli门的模乘法，以 $2n+2$ 个量子比特进行因式分解。《量子信息与计算》，17(7-8)，2017年。30. Antoine Joux、Reynald Lercier、Nigel Smart 和 Frederik Vercauteren。中等素数情形下的数域筛法。载于密码学进展-CRYPTO 2006，计算机科学讲义第4117卷。施普林格，2006年。31. Omar Kihel。次数为10的复二面体扩张的单位群。《波尔多数论杂志》，13(2)，2001年。32. Taechan Kim 和 Changmin Lee。欧几里得环上的格约化及其在密码分析中的应用。载于IMA密码学与编码国际会议，第371–391页。施普林格，2017年。33. A Yu Kitaev。量子测量与阿贝尔稳定子问题。arXiv预印本 quant-ph/9511026，1995年。34. Helmut Koch。代数数论，《数学科学百科全书》第62卷。施普林格，1997年。35. Radan Kučera。分圆域中Stickelberger理想及圆单位群的基。《数论杂志》，40(3)，1992年。

36. C Lawrence。《分圆域导论》，数学研究生教材第83卷。1997年。37. Changmin Lee、Alice Pellet-Mary、Damien Stehlé 和 Alexandre Wallet。一种用于模格的LLL算法。载于密码学进展—Asiacrypt 2019，计算机科学讲义第11922卷，2019年。38. Arjen K Lenstra、Hendrik Willem Lenstra 和 László Lovász。具有有理系数的多项式的因式分解。Mathematische Annalen，261，1982年。39. Arjen K Lenstra 和 Eric R Verheul。XTR公钥系统。载于密码学进展-CRYPTO 2000，计算机科学讲义第1880卷，2000年。40. Xia Liu、Huan Yang 和 Li Yang。Shor算法的CNOT计数优化量子电路。arXiv预印本 arXiv:2112.11358，2021年。41. Michele Mosca 和 Marco Piani。2021年量子威胁时间线报告，2022年。在线可查：
https://info.quintessencelabs.com/hubfs/Quantum-Threat-Timeline-Report-2021-full-report-final%20(

42. Huguette Napias. 欧几里得环或整环上的LLL算法的一般化。《波尔多数论杂志》，8(2):387–396，1996年。43. NIST. 后量子密码标准化流程的提交要求和评估标准，2016年。在线获取：
https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-f

44. J Proos. 椭圆曲线的Shor离散对数量子算法。《量子信息与计算》，3(4)，2003年。45. Martin Roetteler、Michael Naehrig、Krysta M Svore和Kristin Lauter. 计算椭圆曲线离散对数的量子资源估算。载于《密码学进展—Asiacrypt 2017》，计算机科学讲义第10624卷，2017年。46. Arthur Schmidt和Ulrich Vollmer. 计算数域单位群的多项式时间量子算法。载于《第三十七届ACM计算理论年会论文集》，第475–480页，2005年。47. René Schoof. 素数导子实分圆域的类数。《计算数学》，72(242)，2003年。48. Peter W Shor. 量子计算算法：离散对数与因式分解。载于《第三十五届计算机科学基础年会论文集》，第124–134页。IEEE，1994年。49. Sri Aravinda Krishnan Thyagarajan、Guilhem Castagnos、Fabian Laguillaumie和Giulio Malavolta. 类群中的高效CCA定时承诺。载于《2021年ACM SIGSAC计算机与通信安全会议论文集》，2021年。50. Benjamin Wesolowski. 高效的可验证延迟函数。载于《密码学进展—Eurocrypt 2019》，计算机科学讲义第11478卷，2019年。51. Christof Zalka. 使用更少（纯）量子比特的Shor算法。arXiv预印本quant-ph/0601097，2006年。