中国科学: 信息科学 2024 年 第 54 卷 第 12 期: 2727–2742
SCIENTIA SINICA Informationis
论文

格上唯一最短向量问题的细粒度复杂性

金保隆[1,2*], 薛锐[1,2]

1. 中国科学院信息工程研究所网络空间安全防御重点实验室, 北京 100085
2. 中国科学院大学网络空间安全学院, 北京 100049
* 通信作者. E-mail: jinbaolong@iie.ac.cn

收稿日期: 2024–05–13; 修回日期: 2024–07–18; 接受日期: 2024–08–06; 网络出版日期: 2024–11–22

国家自然科学基金 (批准号: 62172405) 资助项目

摘要 唯一最短向量问题 (unique shortest vector problem, uSVP) 的困难性是一些流行的格密码方案的安全基础, 对其在不同参数下的复杂性的研究也是格密码体系的重要组成部分, 然而这方面的研究进展十分缓慢, 同时对 uSVP 的细粒度复杂性的研究也十分欠缺. 本文改进了一个从最短向量问题 (shortest vector problem, SVP) 到 uSVP 的归约算法, 提高了其成功概率, 调整了参数限制, 从而允许建立不同参数条件下从 SVP 到 uSVP 的细粒度归约. 利用这个归约, 得以证明以下结果: (1) 在超多项式时间下, 得到了一个参数更大的从 SVP 到 uSVP 的归约; (2) 在强指数时间假设 (gap strong exponential time hypothesis, GapSETH) 下, 构建了从常数参数的 SVP 到常数参数的 uSVP 的亚指数时间归约, 从而证明了 uSVP 在 GapSETH 下的困难性; (3) 以上结果都可以转化成有界距离译码 (bounded distance decoding, BDD) 的相应结论: 在对应条件下求解参数小于 1 的 BDD 也是困难的; (4) 结合稳定格中的格点个数上界, 得到了参数远超常数的从 SVP 到 uSVP 的归约.

关键词 最短向量问题, 唯一最短向量问题, 计算复杂性, 细粒度复杂性, 复杂性归约

1 引言

基于格的密码体系在近几十年备受关注, 研究者们提出了许多基于格上的困难性假设的密码方案 [1~6], 并在理论研究和实际应用中流行起来. 这些密码方案的安全性主要基于短整数解问题 (short integer solution) 和带错学习问题 (learning with errors) 的困难性, 而这两个问题的困难性又基于特定参数下格上的最短向量问题 (shortest vector problem, SVP) 和唯一最短向量问题 (unique shortest vector problem, uSVP) 的难解性. 因此对 SVP 和 uSVP 的复杂性 —— 尤其是其在不同参数下的困难性 —— 的研究, 对建立完善的格密码体系极其重要. 格上的计算问题的困难性研究结果为基于格的密码方案的安全性提供了理论基础. 现有多项式时间下的困难性归约涉及的参数与实际密码方案能

引用格式: 金保隆, 薛锐. 格上唯一最短向量问题的细粒度复杂性. 中国科学: 信息科学, 2024, 54: 2727–2742, doi: 10.1360/SSI-2024-0145
Jin B L, Xue R. Fine-grained hardness of the unique shortest vector problem in lattices (in Chinese). Sci Sin Inform, 2024, 54: 2727–2742, doi: 10.1360/SSI-2024-0145

达到的参数仍有较大差距, 而在超多项式时间下二者所能达到的参数相对而言更为接近, 对于格上的困难问题在超多项式时间下的细粒度复杂性的研究也因此成为一个趋势. 本文着重关注的正是 uSVP 在超多项式时间下的细粒度复杂性.

虽然格密码已经发展了相当长时间, 但是对 uSVP 的复杂性的研究进展相当缓慢, 现有的相关结论也不甚令人满意. Kumar 和 Sivakumar [7] 给出了第一个关于 uSVP 的 NP 困难性的证明, 但并未明确给出最终得到的 uSVP 实例的参数范围. Aggarwal 和 Dubey [8] 同时给出了确定性的和随机性的从 SVP 到 uSVP 的归约. 其中, 确定性的归约可以将 SVP 实例转化成参数为 $(1+2^{-O(n^2)})$ 的 uSVP 实例, 随机性的归约则能够得到参数为 $(1+1/\operatorname{poly}(n))$ 的 uSVP 实例. 几乎在同一时间, Stephens-Davidowitz [9] 给出了一个从常数参数的 SVP 到参数不超过 $(1+O(\log n/n))$ 的 uSVP 的归约, 从而证明了在该参数范围内的 uSVP 是 NP 困难的. 这一结果是目前针对 uSVP 在多项式时间下的 NP 困难性的最好研究结果. 对于 uSVP 在更大的参数下的结果可以参考文献 [10~12], 这些文献中的结果证明了常数参数的 uSVP 至少和参数为 $O(\sqrt{n/\log(n)})$ 的 SVP 同样困难. 注意, 以上结论都是使用多项式时间归约得到的, 不难发现这里涉及的参数与格密码中所使用的参数 $(n^{O(1)})$ 仍然还有相当大的差距.

近些年关于格问题在超多项式时间下的细粒度复杂性研究得到了长足发展. 最早的此类结果出现在文献 [13] 中, 对于所有奇数 p, Bennett 等 [13] 通过构建从 k-SAT 到 CVP (closest vector problem) 的归约, 得到了 CVP 在所有 l_p 范数空间的基于 SETH (strong exponential time hypothesis) 的复杂性下界. 文献 [14, 15] 分别将该结果扩展到了 SVP 和 SIVP (shortest independent vectors problem) 上. Aggarwal 等 [16] 给出了一系列关于 CVP 及其变体在 SETH 和强指数时间假设 (gap strong exponential time hypothesis, GapSETH) 下的复杂性结果. 结合文献 [14, 16] 的结果可以得到近似求解 SVP 在 GapSETH 下的复杂性结果. Bennett 等 [17] 也探索了 BDD 的细粒度复杂性. BDD 与 uSVP 的复杂性联系十分紧密, 关于二者之间的关系可以参考文献 [11, 18]. Aggarwal 等 [19] 证明了一系列关于格问题的细粒度困难性上界. 文献 [20] 给出了一系列超多项式时间下的一些格问题之间的归约, 包括一个从参数为 $O(1/\varepsilon^{1+1/p})$ 的 CVP 到参数为 $(1+\varepsilon)$ 的 uSVP 的归约.

对于格问题的细粒度复杂性的研究已经成为了一个流行的研究领域, 然而专门针对 uSVP 的细粒度复杂性的研究还十分欠缺. 本文进一步探索这一研究领域, 通过研究 SVP 和 uSVP 在超多项式时间下的联系, 将 SVP 的复杂性结论转化成了 uSVP 在相应条件下的复杂性结论. 通过文中构建的归约, 得到了一些 uSVP 在超多项式时间下的困难性结论. 相较于多项式时间下的已有结果, 文中所给结论中 uSVP 的参数更大.

Stephens-Davidowitz [9] 给出的归约基于一种被称为稀疏化的技术, 这个技术最早出现在 Khot 的文献 [21] 中. 此技术也被 Stephens-Davidowitz 用来构建从 DGS (discrete Gaussian sampling) 到 CVP 和 SVP 的归约 [22]. 本文中的归约的核心步骤也依赖于该技术所提供的能力.

相关研究. 格理论中最重要且最受关注的计算问题是 CVP 和 SVP. 现于此列出现有的关于格问题在多项式时间下的复杂性的主要研究结果, 以供参考. CVP 是第一个被证明具有 NP 困难性的格问题, 这一结果出现在 Boas 的一篇报告中 [23], 此外, Boas 还证明了 l_∞ 范数空间的 SVP 是 NP 困难的, 并推测欧几里得空间的 SVP 也是 NP 困难的. 这一推测最终被 Ajtai [24] 证明. Ajtai 不仅证明了在欧几里得空间精确求解 SVP 是 NP 困难的, 而且指出在 2^{-n^ε} 参数内近似求解 SVP 依然是 NP 困难的.

在 Ajtai 的工作之后, 关于近似求解 SVP 的复杂性的研究得到了长足发展. 文献 [21, 25~27] 给出了常数参数下 SVP 的 NP 困难性证明. 其中文献 [21, 27] 中得到的 SVP 实例可以通过张量积来增大近似参数, 从而证明了以任意常数参数近似求解 SVP 都是 NP 困难的. 另外, 文献 [21] 的结果表明,

如果假设 NP $\not\subseteq$ RTIME($2^{\mathrm{poly}(\log n)}$), 那么对于任意常数 ε 而言, 以不超过 $2^{(\log n)^{1/2-\varepsilon}}$ 的参数近似求解 SVP 也是困难的. 而文献 [27] 的结果表明, 如果允许运行时间不超过 $2^{O(n^{\varepsilon})}$ 的算法, 那么以不超过 $n^{\delta/\log\log n}$ 的参数近似求解 SVP 也是困难的. 文献 [27] 中的归约仅需要在选择目标向量和构造满足 Sauer 定理 [28~30] 的矩阵时使用随机性, 理论上此归约所使用的随机性比文献 [21] 中的归约使用的随机性更少. Haviv 和 Regev [31] 对 Khot [21] 给出的归约作了进一步的分析, 证明该归约也可以使用标准张量积获得任意常数参数, 从而避免了 Khot 所使用的扩展张量积结构所导致的一些缺陷. Bennett 和 Peikert [32] 基于 Reed-Solomon 编码构建了一个从集合覆盖问题 (exact set cover) 到近似求解 SVP 的归约.

另一方面, 在某些计算复杂性推测为真时, 以不小于某些值的参数近似求解 SVP 不是 NP 困难的. 第一个此类结果出现在文献 [33] 中, Goldreich 和 Goldwasser [33] 为参数为 $\sqrt{n/O(\log n)}$ 的 CVP 和 SVP 的补问题构造了一个 Arthur-Merlin (AM) 协议. 这意味着以不小于此参数近似求解 CVP 和 SVP 属于复杂性类 coAM, 因此它们不太可能是 NP 困难的. Aharonov 和 Regev [34] 构造了一个非确定性的验证算法 (NP verifier), 该算法能够验证一个向量是否与给定的格距离比较远, 这意味着对应的判定问题 — 参数为 $c\sqrt{n}$ 的 GapCVP— 属于复杂性类 NP 和 coNP, 因此不太可能是 NP 困难的, 否则 NP $\subset$ coNP. Cai [35] 使用与文献 [33] 中相同的技术证明了参数不小于 $n^{1/4}$ 的 uSVP 不太可能是 NP 困难的. 更多关于 SVP, uSVP 及其变种问题的复杂性的信息, 请参阅文献 [27, 36].

上述大部分困难性归约适用于所有的 l_p 范数空间. Regev 和 Rosen [37] 证明了欧几里得空间的 SVP 是最容易的, 因此得到 SVP 在欧几里得空间的困难性结果也足以证明 SVP 在其他范数空间的困难性.

格密码体系在近几十年得到了长足发展 [1~6], 并且在 NIST (National Institute of Standards and Technology) 的后量子密码标准征集中备受关注. 在实际应用中, 基于格问题的密码系统的安全性不仅依赖于 SVP 或 uSVP 在多项式时间下的难解性, 还依赖于更加具体的困难性假设, 例如不能使用特定参数下的 BKZ 算法求解. BKZ 算法在实际应用中运行效率极高, 但是目前却没有多项式级别的 BKZ 的运行时间上界. 这在一定程度上启发了学者们对格问题的细粒度的复杂性, 尤其是 $2^{o(n)}$ 时间下的复杂性, 进行研究, 以期为基于格的密码方案提供更强的理论支持 (参考文献 [13, 14, 19]). 另外, 关于 SAT 问题的 ETH (exponential time hypothesis) 为超多项式时间下的复杂性研究提供了很好的理论基础, 并允许使用更强的计算能力.

表 1 展示了目前 uSVP [7~9, 11, 12, 20] 和 BDD [11, 12, 17, 38] 主要复杂性结果以及本文的结果. 其中粗体部分是本文的结果.

本文贡献. (1) 文献 [9] 中的归约的改进: 在归约过程中本文采用了不同素数选择策略, 并且用新方式证明了归约算法的正确性. 在改进后的归约过程中, 所有参数均可随意调整, 使得改进后的归约算法的成功概率下界比文献 [9] 给出的更大 (见定理 4 的证明之后的讨论; 表 2).

(2) 超多项式时间下的归约: 在允许运行时间为 $2^{O(n/\log n)}$ 的算法中, 求解参数为 $(1+c/\log n)$ 的 uSVP 至少和以常数参数近似求解 SVP 同样困难 (见定理 5). 文献 [9] 在多项式条件下得到的参数为 $(1+O(\log n/n))$, 二者之差为 $(c/\log n - O(\log n/n))$, 对于充分大的 n, 本文在超多项式条件下得到的 uSVP 实例参数更大.

(3) 亚指数时间下的归约: 在允许运行时间为 $2^{\varepsilon n}$ 的算法中, 求解常数参数的 uSVP 至少和以更大的常数参数近似求解 SVP 同样困难 (见定理 6). 由于以接近 1 的参数近似求解 SVP 在 GapSETH 下是困难的, 所以本文的结果表明: 参数为接近 1 的常数的 uSVP 在 GapSETH 下同样是困难的 (会有一定的参数损失).

表 1　uSVP 与 BDD 复杂性的现有主要结果和本文结果 (粗体)

Table 1　Existing main results about the complexity of uSVP and BDD, and results in this paper (in bold)

Time of reduction	uSVP		BDD	
	Factor	Complexity result	Factor	Complexity result
poly(n)	–	NP hard [7]	$O(\sqrt{\log n/n})$	Efficiently solvable [38]
	$1+2^{-O(n^2)}$	NP hard [8]	$\alpha > 1/\sqrt[p]{2}$	$\geqslant (1-\alpha^{-p}/2)^{-1/p}$-CVP [38]
	$1+1/\mathrm{poly}(n)$	NP hard [8]		–
	$1+O(\log n/n)$	NP hard [9]		–
	$c>1$	$\geqslant \sqrt{\frac{O(n)}{\log n}}$-SVP [11, 12]	$1/c<1$	$\geqslant \sqrt{\frac{O(n)}{\log n}}$-SVP [11, 12]
$2^{O(n/\log n)}$	$\mathbf{1+O(1/\log n)}$	$\boldsymbol{\geqslant \gamma}$**-SVP, constant** $\boldsymbol{\gamma}$ **(5)**	$\mathbf{(1+\frac{O(1)}{\log n})^{-1}}$	$\boldsymbol{\geqslant \gamma}$**-SVP, constant** $\boldsymbol{\gamma}$ **(5)**
Subexponential	$\boldsymbol{\gamma^{1/l}}$	$\boldsymbol{\geqslant \gamma}$**-SVP, constant** $\boldsymbol{\gamma}$ **(6)**	$\boldsymbol{\alpha<1}$	**GapSETH hard (6)**
	$1+\varepsilon$	$\geqslant O(1/\varepsilon^{1+1/p})$-CVP [20]	$\alpha > \alpha^{\dagger}_{p,C} > 1$	Non-uniform GapSETH hard [17]
Stable lattice	$\boldsymbol{n^{O(1/\log\log n)}}$	$\boldsymbol{\geqslant O(\sqrt{n})}$**-SVP (8)**	$\boldsymbol{n^{-O(1/\log\log n)}}$	$\boldsymbol{\geqslant O(\sqrt{n})}$**-SVP (8)**

表 2　文献 [9] 与本文中算法参数对比

Table 2　Comparison of some parameters in [9] and this paper

	Single run success probability	Range of prime p	Factor of SVP (γ)
Ref. [9]	$\frac{\xi_1}{p}(1-\frac{\xi_2}{p})$	$\xi_2 \leqslant p/(20\log p)$	$1 \leqslant \gamma < 2$
This paper		$1+1/c^2 \leqslant p/\xi_2 \leqslant 1+c^2$	$1 \leqslant \gamma$

(4) 更大的参数的归约: 对于一类满足特定性质的格 —— 稳定格 (stable lattice [39]), 有更紧致的格点个数上界, 使用这些格点个数上界, 结合文中给出的归约, 可以得到超过常数参数的 SVP 和 uSVP 的复杂性的关系 (见定理 8). 该结果对建立超多项式时间下的格密码体系 [19] 提供了参考.

本文结构如下: 第 2 节介绍了本文用到的记号、符号以及一些基本定义和重要结论. 第 3 节详细介绍了本文构建的归约算法及其证明, 并给出了在两种不同的参数设置下所得到的复杂性结论. 第 4 节介绍了稳定格上的 uSVP 在大参数下的复杂性结论. 第 5 节对全文进行了总结.

2　预备知识

2.1　符号与标记

本文使用小写字母来代表数字和向量, 大写字母表示矩阵. 一些例外可以通过上下文判断其意义 (比如文中会使用 N 代指数字, g 和 f 代指函数). 希腊字母用来代指数值, 它们一般不是确定的值, 而是强调在不同情况下都存在这样的值.

对于任意向量 $v \in \mathbb{R}^n$ 和正整数 $p \in \mathbb{N}^+$, v 在 l_p 范数空间的模长为 $\|v\|_p = (\sum_{i=1}^{n} |v_i|^p)^{1/p}$. 最常见的是 l_2 范数空间, 也被称为欧几里得空间. 在明确所使用的范数空间后, v 在 l_p 范数空间的模长简称为 “长度”. 本文中所涉及的内容对所有 l_p 范数空间都适用, 所以不会指定具体的范数空间.

格为一组向量的整数线性组合所组成的集合. 给定一组向量 $B = [b_1, b_2, \ldots, b_m] \in \mathbb{R}^{n\times m}$, 其生成的格为 $\mathcal{L}(B) = \{Bz = \sum z_i b_i \mid z \in \mathbb{Z}^m\}$. 文中用符号 $\mathcal{L}$ 或者 B 来代指格 $\mathcal{L}(B)$. 注意, 格中的元素都是 $\mathbb{R}^n$ 中的向量, 但在讨论格上问题时也将其称为 “格点”. 文中 (不失一般性地) 假设了所涉及的 B 都由 n 个线性无关的 n 维向量组成.

将向量 $t \in \mathbb{R}^n$ 到格 $\mathcal{L}$ 的距离定义为 $\mathrm{dist}(\mathcal{L}, t) = \min\{\|t - v\|_p \mid v \in \mathcal{L}\}$.

关于格 $\mathcal{L}$ 有一组重要的常数 $\{\lambda_i\}$, 被称为相继最小值 (successive minimums), 其具体定义为 $\lambda_i(\mathcal{L}) = \inf\{r|\dim(\mathrm{span}(\mathcal{L} \cap \mathcal{B}(0,r))) \geqslant i\}$, 其中 $\mathcal{B}(0,r) = \{v \mid \|v\|_p \leqslant r, v \in \mathbb{R}^n\}$ 指以原点作为球心半径为 r 的 n 维球体. 对同一个格而言, 在不同的范数空间这组常数的值也会有所不同. 这组常数中最重要的是 $\lambda_1(\mathcal{L}(B))$, 本文使用 $\lambda(\mathcal{L})$ 和 $\lambda(B)$ 来代指 $\lambda_1(\mathcal{L}(B))$.

如果非零格点 $v \in \mathcal{L}$ 对于任意 $u \in \mathcal{L}, u \neq \pm v$ 都有 $\forall k \in \mathbb{Z}, v \neq ku$, 那么 v 为 $\mathcal{L}$ 中的本原格点 (primitive lattice point). 将格 $\mathcal{L}$ 中所有本原格点组成的集合记为 $\mathcal{L}_{\mathrm{prim}}$, 格 $\mathcal{L}$ 中长度不超过给定值 r 的本原格点的个数记为 $\xi(\mathcal{L}, r) = \frac{1}{2}\#\{\mathcal{L}_{\mathrm{prim}} \cap \mathcal{B}(0,r)\}$.

2.2 基本定义与已有结论

定义1 (最近向量问题, CVP) 给定格基 $B \in \mathbb{R}^{n\times m}$ 和一个目标向量 $t \in \mathbb{R}^n$, CVP 要求找到一个距离 t 最近的格点. 即找到一个格点 $v \in \mathcal{L}(B)$, 满足 $\|v - t\|_p = \mathrm{dist}(\mathcal{L}(B), t)$.

定义2 (判定性最近向量问题, GapCVP) 在 GapCVP 判定过程中, 对任意实数 $\gamma \geqslant 1 \in \mathbb{R}$, 格基 $B \in \mathbb{R}^{n\times m}$ 和一个目标向量 $t \in \mathbb{R}^n$, 以及一个正实数 d,

- 如果 $\mathrm{dist}(\mathcal{L}(B), t) \leqslant d$, 那么 (B, t, d) 称为 YES 实例;
- 如果 $\mathrm{dist}(\mathcal{L}(B), t) > \gamma d$, 那么 (B, t, d) 称为 NO 实例.

CVP 是格理论中极其重要的计算问题, 许多格问题的困难性结果都依赖于 CVP 的 NP 困难性, GapCVP 是其判定版本, 后者在实际应用和理论研究中更加常见.

定义3 (最短向量问题, SVP) 给定格基 $B \in \mathbb{R}^{n\times m}$, SVP 要求找到格 $\mathcal{L}(B)$ 中的一个最短向量. 即找到一个格点 $v \in \mathcal{L}$, 满足 $\|v\|_p = \lambda(B)$.

定义4 (判定性最短向量问题, GapSVP) GapSVP 判定过程中, 对任意近似因子 $\gamma \geqslant 1 \in \mathbb{R}$, 给定格基 $B \in \mathbb{R}^{n\times m}$ 和一个正实数 d,

- 如果 $\lambda(B) \leqslant d$, 那么 (B, d) 称为 YES 实例;
- 如果 $\lambda(B) > \gamma d$, 那么 (B, d) 称为 NO 实例.

GapSVP 是一个判定性问题, 它也是格理论中的关键问题, 其困难性一定不大于同参数的近似求解 SVP. 通常文中会明确指出 SVP 实例的近似因子 γ, 并将其称为 “以参数 γ 近似求解 SVP”、“参数为 γ 的 SVP” 或者 γ-SVP.

定义5 (唯一最短向量问题, uSVP) 对任意实数 $\gamma > 1 \in \mathbb{R}$, 给定格基 $B \in \mathbb{R}^{n\times m}$, uSVP 要求在格 $\mathcal{L}(B)$ 中的最短向量唯一的情况下, 找到格 $\mathcal{L}(B)$ 中的最短向量. 即如果 $\lambda_2(B) \geqslant \gamma\lambda_1(B)$, 那么要求找到格点 $v \in \mathcal{L}$, 满足 $\|v\|_p = \lambda(B)$.

uSVP 的定义中自带参数 γ, 这个参数指出了在何范围不存在与最短格点线性无关的格点. 在后文中会使用 “参数为 γ 的 uSVP” 或 γ-uSVP 来代指符合 $\lambda_2(B) \geqslant \gamma\lambda_1(B)$ 要求的 uSVP. γ-uSVP 谕言机在接收到符合 γ-uSVP 定义的实例时会返回最短的格点, 在收到的格不符合 γ-uSVP 定义时, 不妨也要求其必须返回一个格点.

定义6 (有界距离译码, BDD) 对任意实数 $\alpha > 0 \in \mathbb{R}$, 给定格基 $B \in \mathbb{R}^{n\times m}$ 和向量 $t \in \mathbb{R}^n$, 满足 $\mathrm{dist}(\mathcal{L}(B), t) \leqslant \alpha\lambda(B)$. BDD 要求找到一个格点 $v \in \mathcal{L}(B)$ 满足 $\|t - v\| = \mathrm{dist}(\mathcal{L}(B), t)$.

定义7 (k-SAT) k-SAT 问题的实例 $\psi = \wedge_i c_i$ 是由一系列子式的合取 (conjunction) 组成的布尔函数, 并且每一个子式 $c_i = \vee_j x_{ij}$ 是至多 k 个文字的析取 (disjunction). 在判定 k-SAT 问题中,

- 如果存在一组赋值使得 ψ 的值为真, 那么 ψ 称为 YES 实例;

• 如果对所有可能的赋值, ψ 的值都为假, 那么 ψ 称为 NO 实例.

定义8 (Max-k-SAT) 对于任意给定的 k-SAT 实例 $\psi=\wedge_{i=1}^{m}c_i$ 和一个实数 $\delta\in(0,1]$, Max-k-SAT 要求判断是否存在一组赋值 a 使得 $\sum_i c_i(a)\geqslant\delta m$.

定义9 (ETH [40, 41]) 存在一个常数 $\varepsilon>0$, 不存在能在 $2^{\varepsilon n}$ 时间内解决 n 元 3-SAT 的算法.

定义10 (SETH [40, 41]) 对于任意常数 $\varepsilon>0$, 都存在一个常数 $k\geqslant 3$, 不存在能在 $2^{(1-\varepsilon)n}$ 时间内解决 n 元 k-SAT 的算法.

定义11 (GapSETH [42]) 对于任意常数 $\varepsilon>0$, 都有 $k(\varepsilon)$ 和 $\delta(\varepsilon)>0$, 不存在能在 $2^{(1-\varepsilon)n}$ 时间内区分 n 元 k-SAT 的可满足实例与只能满足 $1-\delta$ 部分子式实例的算法.

格问题 GapSETH 下的困难性可以由从 Max-k-SAT 开始到 GapCVP 的若干归约得出. 本文的超过多项式时间的归约算法也基于这些结果.

定理1 ([43]) 对于任意的格 $\mathcal{L}$ 和常数 γ 有 $\#\{v\in\mathcal{L}\mid\|v\|_p\leqslant\gamma\lambda(\mathcal{L})\}\leqslant 2\cdot\lceil 2\gamma\rceil^n-1$.

定理2 ([39]) 令 $t=10(\log(n)+2)$, 那么对于任意的格 $\mathcal{L}\subset\mathbb{R}^n$, 如果 $\forall\mathcal{L}'\subset\mathcal{L},\det(\mathcal{L}')\geqslant 1$, 则对每个向量 $u\in\mathbb{R}^n$ 有以下结论:

(1) 对任意 $r\geqslant 1$, 有 $|\mathcal{L}\cap(r\mathcal{B}_2^n+u)|\leqslant 3\mathrm{e}^{\pi t^2r^2}/2$;

(2) 对任意 $\sqrt{n/2\pi}\cdot t^{-1}\leqslant r\leqslant\sqrt{n/2\pi}\cdot t$, 存在一个常数 C, 有 $|\mathcal{L}\cap(r\mathcal{B}_2^n+u)|\leqslant(Ctr/\sqrt{n})^{n/2}$;

(3) 对任意 $r\geqslant\sqrt{n/2\pi}\cdot t$, 有 $|\mathcal{L}\cap(r\mathcal{B}_2^n+u)|\leqslant 2(2\pi er^2/n)^{n/2}$,

其中 $\mathcal{B}_2^n$ 指 l_2 范数空间的 n 维单位球. 文献 [39] 将满足定理 2 中约束的格 $\mathcal{L}$ 称为稳定格.

引理1 ([7]) 对于任意大小不超过 2^m 的有限集合 $T\neq\emptyset$, 存在高效算法生成子集系列 $T=T_0\supseteq T_1\supseteq\cdots\supseteq T_{2m}$, 使得存在仅有一个元素的子集的概率不少于 $2/3-2^{-m}$.

根据素数定理, 对于任意正整数 n, 在 $[n,2\,n]$ 上可以高效地进行素数采样. 因此存在一个常数 $1<c\leqslant 2$, 对所有充分大的整数 i, 可以在 $[c^i,c^{i+1}]$ 上高效地进行素数采样. 后文将用到此事实.

3 参数化的归约框架

3.1 稀疏定理

在介绍本文的归约算法之前, 先介绍一种筛选格点的方法, 该方法被称为稀疏引理 (sparsification lemma). 对于指定的格 $\mathcal{L}(B)$ 和一个素数 p, 稀疏引理允许对所有格点进行随机筛选, 最终得到一个子格 $\mathcal{L}'$, 该子格包含且仅包含所有满足 $\langle z,B^{-1}v\rangle\equiv 0\bmod p$ 的格点 (z 为随机选取的整数向量).

现在逐步介绍该引理及其应用. 首先对于一组向量和素数 p 有下面的引理.

引理2 ([22]) 对于任意素数 p 和一组非零向量 $x_0,x_1,\ldots,x_N\in\mathbb{Z}_p^n$, 如果 $\forall a\in\mathbb{R},x_0\neq ax_i\bmod p$ 对于所有 $i\geqslant 1$ 成立, 那么对于从 $\mathbb{Z}_p^n$ 中均匀随机选取的向量 z 有

$$\frac{1}{p}-\frac{N}{p^2}\leqslant\Pr[\langle z,x_0\rangle\equiv 0\bmod p,\ \langle z,x_i\rangle\not\equiv 0\bmod p,\forall i]\leqslant\frac{1}{p}.$$

引理 2 中并未对所涉及的一系列向量做额外约束, 仅要求 x_0 与其他向量线性无关 (模 p). 该引理应用到本原格点上就相当于对相应的格使用该引理. 对于格 $\mathcal{L}(B)$、素数 p 以及向量 $z\in\mathbb{Z}_p^n$ 使用引理 2 后被保留下来的格点是 $\mathcal{L}$ 的子格, 记为 $\mathcal{L}'$. 可以高效地计算 $\mathcal{L}'$ 的一组基, 在此给出两个已有工作中的结论, 具体计算方式参考对应文献.

引理3 ([18,22]) 对于格 $\mathcal{L}\subset\mathbb{R}^n$ 的一组基 B、素数 $p\in\mathbb{Z}^+$ 和向量 $z\in\mathbb{Z}_p^n$, 存在一个多项式时间的算法计算格 $\mathcal{L}'=\{v\in\mathcal{L}\mid\langle z,B^{-1}v\rangle\equiv 0 \bmod p\}$ 的一组基 B'.

在将引理 2 应用到格上时, 需要考察的不是格点本身, 而是格点在基 B 下的系数向量. 例如对于格 $\mathcal{L}(B)$, 素数 p, 随机向量 $z\in\mathbb{Z}_p^n$, 考虑格点 v_1,v_2. 引理 2 会根据 $\langle z,B^{-1}v_1\rangle \bmod p$ 和 $\langle z,B^{-1}v_2\rangle \bmod p$ 保留或剔除对应的格点, 生成的子格包含所有系数向量与 z 的内积为 0 (模 p) 的格点. 按照引理要求, 在模 p 下, v_1,v_2 的系数向量不能是彼此倍数, 即对任意 $a\in\mathbb{Z}_p$, 不允许 $B^{-1}v_1\equiv aB^{-1}v_2 \bmod p$. 为排除此种情况, 文献 [22] 给出了下面的引理.

引理4 ([22]) 对格 $\mathcal{L}(B)\subset\mathbb{R}^n$, 假设 $v_1,v_2\in\mathcal{L}$ 是两个本原格点, $v_1\neq\pm v_2,\|v_1\|\geqslant\|v_2\|$, 并且对任意 $p\geqslant 101$ 和 $a\in\mathbb{Z}_p$ 满足 $B^{-1}v_1\equiv aB^{-1}v_2 \bmod p$, 那么 $\xi(\mathcal{L},\|v_1\|)>p/(20\log p)$.

这样一来, 只要在使用引理 2 时额外要求对所有涉及的本原格点 v 都有 $\xi(\mathcal{L},\|v\|)\leqslant p/(20\log p)$, 就可以避免这种情况. 然而这样的限制条件导致素数 p 比本原格点个数大很多, 以至于在使用引理 2 构建从 SVP 到 uSVP 的归约时成功概率比较小. 在此给出一种新的方法处理此种情形.

引理5 对于任意格 $\mathcal{L}(B)\subset\mathbb{R}^n$, 素数 p, $v_1\neq\pm v_2\in\mathcal{L}\cap\mathcal{B}(r)$ 是两个本原格点. 如果 $r<\lambda(\mathcal{L})p/2$, 那么对于任意满足 $\|av_1\|\leqslant r,\|bv_2\|\leqslant r$ 的 $a,b\in\mathbb{Z}_p$, 有 $aB^{-1}v_1\not\equiv bB^{-1}v_2 \bmod p$.

证明 假设存在满足引理 5 的两个格点 v_1,v_2 满足 $aB^{-1}v_1\equiv bB^{-1}v_2 \bmod p$, 那么有 $B^{-1}(av_1-bv_2)\equiv 0 \bmod p$, 因此存在一个格点 $v_3\neq 0$ 使得 $av_1-bv_2=pv_3\geqslant\lambda(\mathcal{L})p$. 显然, 这与 $v_1,v_2\in\mathcal{L}\cap\mathcal{B}(r)$ 和 $r<\lambda(\mathcal{L})p/2$ 矛盾.

结合引理 2 和 4, 对任意一系列本原格点, 有以下定理.

定理3 ([22]) 对于任意的格 $\mathcal{L}(B)$, 一系列本原格点 $y_0,y_1,\ldots,y_N\in\mathcal{L}_{\text{prim}}$, 并且 $y_i\neq\pm y_0,\forall i>0$, 以及素数 $p\geqslant 101$, 如果 $\xi(\mathcal{L},\|y_i\|)\leqslant p/(20\log p)$, $\forall i$, 那么对于从 $\mathbb{Z}_p^n$ 中均匀随机选取的向量 z 有

$$\frac{1}{p}-\frac{N}{p^2}\leqslant\Pr[\langle z,B^{-1}y_0\rangle\equiv 0 \bmod p,\ \langle z,B^{-1}y_i\rangle\not\equiv 0 \bmod p,\ \forall i]\leqslant\frac{1}{p}.$$

对于任意本原格点 x, 将格中所有长度不超过 $\|x\|$ 的本原格点代入上述定理, 可以得到如下引理.

引理6 ([22]) 存在一个多项式时间的算法, 以格 $\mathcal{L}(B)$ 和素数 p 作为输入, 输出一个 $\mathcal{L}$ 的子格 $\mathcal{L}'$, 使得对于任意满足 $N:=\xi(\mathcal{L},\|x\|)-1\leqslant p/(20\log p)$, $\|x\|<\lambda_1(\mathcal{L})p$ 的格点 $x\in\mathcal{L}$, 若以 $\mathcal{L}'$ 作为输入询问 SVP 谕言机, 有下面的不等式:

$$\frac{1}{p}-\frac{N}{p^2}\leqslant\Pr[\text{SVP}(\mathcal{L}')=\pm x]\leqslant\frac{1}{p}.$$

特别地,

$$\frac{N}{p}-\frac{N^2}{p^2}\leqslant\Pr[\lambda(\mathcal{L}')\leqslant\|x\|]\leqslant\frac{N}{p}.$$

如果能够使用引理 2 构建一个子格, 使得该子格中仅剩一个长度不超过 r_1 的本原格点, 同时所有长度超过 r_1 且不超过 r_2 的格点都被筛除, 那么就可以得到一个 uSVP 实例.

引理7 ([9]) 存在一个多项式时间的随机算法, 以格 $\mathcal{L}(B)$ 和素数 p 作为输入, 输出一个 $\mathcal{L}$ 的子格 $\mathcal{L}'$, 对于任意满足 $\lambda(\mathcal{L})\leqslant r_1\leqslant r_2<\lambda(\mathcal{L})p$ 和 $\xi(\mathcal{L},r_2)\leqslant p/(20\log p)$ 的实数 r_1,r_2 有

$$\Pr[\lambda_1(\mathcal{L}')\leqslant r_1,\ \lambda_2(\mathcal{L}')>r_2]\geqslant\frac{\xi(\mathcal{L},r_1)}{p}\cdot\left(1-\frac{\xi(\mathcal{L},r_2)}{p}\right).\tag{1}$$

将引理 7 中的限制条件 $\xi(\mathcal{L},r_2)\leqslant p/(20\log p)$ 替换成 $r<\lambda_1(\mathcal{L})p$, 并额外增加一个长度区间, 根据引理 5 , 可以得到下面的引理.

引理8 存在多项式时间的随机算法, 以格 $\mathcal{L}(B)$ 和素数 p 作为输入, 输出 $\mathcal{L}$ 的子格 $\mathcal{L}'$, 对于任意满足 $\lambda(\mathcal{L})\leqslant r_1\leqslant r_2\leqslant r_3<\lambda(\mathcal{L})p/2$ 和 $r_3<2r_1$ 的实数 r_1,r_2,r_3, 有下面的不等式:

$$\Pr[r_1<\lambda_1(\mathcal{L}')\leqslant r_2,\ \lambda_2(\mathcal{L}')>r_3]\geqslant\frac{\xi(\mathcal{L},r_2)-\xi(\mathcal{L},r_1)}{p}\cdot\left(1-\frac{\xi(\mathcal{L},r_3)}{p}\right).\tag{2}$$

为了叙述简便, 在下面的内容中将使用 "稀疏定理" 来代指引理 8.

3.2 灵活划分

文献 [9] 中为了证明归约算法的成功概率下界将所有的格点做了划分, 利用这个划分也得以证明归约算法得到的 uSVP 实例的参数以及运行时间都满足要求. 需要注意的是, 文献 [9] 所使用的划分仅是为了证明而进行的虚拟划分, 并不需要实际上的操作.

通过研究发现, 使用与文献 [9] 中所使用的划分相近的方式也可以证明相同的结果, 因此可用其他方式证明该算法的正确性. 需要指出的是, 使用不同的划分所得到的最终参数、运行时间以及成功概率会有所不同. 也许使用某个或者某组划分可以给出该算法更好的证明, 比如得到更大的参数或者更大的成功概率. 鉴于这种情况给出下面的引理.

引理9 对任意正整数 $l>0$, 选定任意的区间 $[0,l]$ 上的单调非减函数 $f:\mathbb{N}\to\mathbb{R}$, 令其满足 $f(0)=a,f(l)=b$. 选定另一个任意的区间 $[a,b]$ 上的单调非减函数 $g:\mathbb{R}\to\mathbb{N}$, 令其满足 $g(a)>0,g(b)\leqslant N_\gamma\cdot g(a)$, 其中 a,b 是满足 $a<b$ 的实数, N_γ 是一个大整数. 如果正整数 k,l,N_γ 满足 $k^l\geqslant N_\gamma$, 那么存在 $1\leqslant j\leqslant l$ 使得 $g(f(j))\leqslant k\cdot g(f(j-1))$ 成立.

证明 假设 $\forall j,g(f(j+1))>k\cdot g(f(j))$. 对于任意的 $i>j$, 都满足 $g(f(i))>k^{i-j}\cdot g(f(j))$, 因此有不等式 $g(f(l))>k^l\cdot g(f(0))$, 即 $g(b)>k^l\cdot g(a)$, 这与引理中的限制条件矛盾.

3.3 归约框架

文献 [9] 提出了一个从 SVP 到 uSVP 的归约算法, 并使用该算法成功证明了参数为 $1+O(\log n/n)$ 的 uSVP 也是 NP 困难的. 基于文献 [9] 中的归约算法, 在此给出该归约的一个改进版本, 此版本更改了素数的采样方式, 去除了一些不必要的参数限制, 从而允许自由选取各项参数. 这里假设调用 γ'-uSVP 谕言机所消耗的时间可以忽略不计, 并且在不考虑算法并行化的前提下, 允许同时进行任意多项式个数的 uSVP 询问.

定理4 给定任意 $\gamma\geqslant1\in\mathbb{R}$, 定义 $N_\gamma=\sup_{\mathcal{L}\subset\mathbb{R}^n}\{\xi(\mathcal{L},\gamma\lambda(\mathcal{L}))\}$, 令正整数 h,k,l 满足以下不等式: $k>2\gamma,k^l\geqslant N_\gamma,h\leqslant\mathrm{poly}(n),l<\mathrm{poly}(n)$. 令 $\gamma'=\gamma^{1/l}<\sqrt{2}$, c 为使区间 $[c^i,c^{i+1}]_{0\leqslant i\leqslant h}$ 上可以高效地进行素数采样的常数, 并且满足不等式 $c^h\geqslant(c+1/c)N_\gamma$. 存在一个随机算法将 γ-SVP 归约到 γ'-uSVP, 该算法的运行时间上界为 $\mathrm{poly}(n)\cdot k$, 其成功概率有大于零的常数下界.

证明 此归约算法的输入为格 $\mathcal{L}$ 的一组基 B, 其被视作一个 γ-SVP 的实例. 对每个整数 $i=0,\ldots,h$ 从区间 $[c^i,c^{i+1}]$ 上进行素数采样得到 p_i. 执行下面的步骤:

(1) 以格 $\mathcal{L}$ 和每一个素数 p_i 作为输入使用引理 8, 得到一系列子格 $\{\mathcal{L}_i\}$.

(2) 以每一个 $\mathcal{L}_i$ 作为输入调用 γ'-uSVP 谕言机, 将谕言机的所有输出记为集合 $\{x_i\}$, 并将其中最短的非零格点记录下来.

(3) 重复以上两个步骤 k 次, 并输出记录到的最短的非零格点 x.

下面来证明上面的归约过程满足定理 4 的要求.

时间上界. 归约算法需要生成 h 个素数, 在每一次执行中, 需要执行 h 次稀疏定理, 然后向 γ'-uSVP 谕言机做 h 次询问. 根据引理 3 , 使用稀疏定理生成一个随机子格的过程是十分高效的, 同时, 在所要求的区间内也可以高效地进行素数采样. 另外, 由于 $N_\gamma = \sup_{\mathcal{L}\subset\mathbb{R}^n}\{\xi(\mathcal{L},\gamma\lambda(\mathcal{L}))\}$, 根据定理 1 则有 $N_\gamma \leqslant \#\{v\in\mathcal{L} \mid \|v\|\leqslant\gamma\lambda(B)\} \leqslant 2\cdot\lceil 2\gamma\rceil^n - 1$, 因此设置 $h\leqslant \mathrm{poly}(n)$ 也是合理的. 综上所述, 前面给出的归约算法前两步执行所需要的时间不会 $\mathrm{poly}(n)$, 因此完成一次归约算法 (独立重复运行前两步 k 次) 所消耗的总时间为 $\mathrm{poly}(n)\cdot k$.

成功概率. 将 $\mathcal{L}$ 中的格点按照它们的长度做出划分. 首先, 令 $r_0 = \lambda(\mathcal{L})$, 对每一个 $i = 1,\ldots,l$ 令 $r_i = \gamma'^i r_0$, 并且定义 $r_{-1} = r_0/2$. 按照引理 9 设置 $f(i) = \gamma'^i r_0, g(f(i)) = \xi(\mathcal{L}, f(i))$, 那么存在一个 $0\leqslant j\leqslant l-1$, 使得不等式 $\xi(\mathcal{L},r_{j+1})\leqslant k\cdot\xi(\mathcal{L},r_j)$ 成立. 这里不妨假定 j 是满足此条件的最小整数.

由于定理中作出了限制 $\gamma' < \sqrt{2}$, 所以 $r_{j+1}\leqslant \gamma'^2 r_{j-1}\leqslant 2r_{j-1}$. 因此, 对任意的长度在区间 $(r_{j-1},r_j]$ 中的格点 v 以及长度在区间 $(r_j,r_{j+1}]$ 中的格点 u, 有 $\forall w\geqslant 2\in\mathbb{Z}, u\neq wv$. 不妨假设 $\exists j', r_2\leqslant\gamma\lambda(\mathcal{L})\leqslant\lambda(\mathcal{L})p_{j'}/2$. (在一般情况下这个不等式并不成立, 这里不妨先假设 $p_{j'}\gg\gamma$, 后文会说明此假设的合理性).

定义事件 E_j 为 "通过应用稀疏定理得到的子格里, 在所有长度不超过 r_{j+1} 的本原格点中, 仅长度在区间 $(r_{j-1},r_j]$ 中的一个本原格点得以保留". 根据稀疏定理 8, 对于归约过程中得到的每一个子格 $\mathcal{L}'$ 以及合适的素数 $p_{j'}$, 如果 $c^{j'}\leqslant(c+1/c)\xi(\mathcal{L},r_{j+1})\leqslant c^{j'+1}$ (根据前面所做的划分, 一定存在这样的区间), 那么有下面的不等式:

$$\begin{aligned}\Pr[E_j] &= \Pr[r_{j-1}<\lambda_1(\mathcal{L}')\leqslant r_j,\ \lambda_2(\mathcal{L}')\geqslant r_{j+1}]\\ &\overset{(a)}{>}\frac{\xi(\mathcal{L},r_j)-\xi(\mathcal{L},r_{j-1})}{p_{j'}}\left(1-\frac{\xi(\mathcal{L},r_{j+1})}{p_{j'}}\right)\\ &\overset{(b)}{>}\frac{(k-1)\xi(\mathcal{L},r_j)}{kp_{j'}}\left(1-\frac{\xi(\mathcal{L},r_{j+1})}{p_{j'}}\right)\\ &\overset{(c)}{>}\frac{(k-1)c^2\xi(\mathcal{L},r_j)}{k(1+c^2)^2\xi(\mathcal{L},r_{j+1})}\overset{(d)}{\geqslant}\frac{(k-1)c^2}{k^2(1+c^2)^2}.\end{aligned}\tag{3}$$

下面逐步解释式 (3) 的各项: 根据不等式 (2) 可以得到不等式 (a). 再根据前文所述, j 为满足 $\xi(\mathcal{L},r_{j+1})\leqslant k\cdot\xi(\mathcal{L},r_j)$ 的最小值, 所以 $\xi(\mathcal{L},r_j) > k\cdot\xi(\mathcal{L},r_{j-1})$, 故 $\xi(\mathcal{L},r_j)-\xi(\mathcal{L},r_{j-1}) > \xi(\mathcal{L},r_j) - \xi(\mathcal{L},r_j)/k$, 因此得到不等式 (b).

由于 $p_{j'}\in[c^{j'},c^{j'+1}]$, 且 $c^{j'}\leqslant(c+1/c)\xi(\mathcal{L},r_{j+1})\leqslant c^{j'+1}$, 所以 $(1+1/c^2)\xi(\mathcal{L},r_{j+1})\leqslant p_{j'}\leqslant(1+c^2)\xi(\mathcal{L},r_{j+1})$. 又因为定义域为 $(1/((1+c^2)a), 1/((1+1/c^2)a))$ 的函数 $x(1-ax)$ 在 $x = 1/(2a)$ 时取得最大值, 所以 $p_{j'}$ 越接近 $2\xi(\mathcal{L},r_{j+1})$, 上述概率下界越大, 且在 $p_{j'}$ 接近 $(1+1/c^2)\xi(\mathcal{L},r_{j+1})$ 或 $(1+c^2)\xi(\mathcal{L},r_{j+1})$ 时取得最小值 (二者相等). 用 $(1+c^2)\xi(\mathcal{L},r_{j+1})$ 替换 $p_{j'}$ 得到不等式 (c). 最后, 由 $\xi(\mathcal{L},r_{j+1})\leqslant k\cdot\xi(\mathcal{L},r_j)$ 可以得到不等式 (d). 因此式 (3) 成立.

归约过程的单次运行得到满足 $\|x\| < \gamma\lambda(\mathcal{L})$ 的格点 x 的概率不小于任何一个实例成功的概率 $\Pr[E_j]$. 将这个过程独立执行 k 次后, 最终得到满足 $\|x\| < \gamma\lambda(\mathcal{L})$ 的格点 x 的概率不小于 $1-(1-\Pr[E_j])^k$. 根据 $\mathrm{e}^{-1} = \lim_{n\to\infty}(1-1/n)^n$, 利用式 (3), 对于充分大的 k 有如下不等式:

$$\Pr[\|x\|<\gamma\lambda(\mathcal{L})]\geqslant 1-(1-\Pr[E_j])^k > 1-\left(1-\frac{(k-1)c^2}{k^2(1+c^2)^2}\right)^k \approx 1-\mathrm{e}^{-c^2/(1+c^2)^2}.$$

因此定理 4 中的归约算法的最终成功概率的下界是大于零的常数.

根据定理中的设置, 如果在某一次使用稀疏定理时成功得到了一个 γ'-uSVP 实例, 此时调用 uSVP 谕言机得到的向量长度不会超过相应的 r_j, 归约算法最后输出的向量 x 的长度也不会超过 r_j, 且 $r_j \leqslant \gamma r_0 = \gamma\lambda(\mathcal{L})$, 因此最后得到的短向量是符合要求的 SVP 的近似解.

合理性说明. 现在来证明前面选择的 $p_{j'}$ 足够好. 首先, 当 $j \in \{0,1\}$ 时, 可以直接对输入的格 $\mathcal{L}$ 应用稀疏定理从而得到上面的不等式. 当 $j = 0$ 时, 可以使用稀疏定理或者引理 1 将该格以一定概率转化成符合要求的 uSVP 实例 (仅保留一个最短向量), 其成功概率不小于上面所给下界.

假设 $j \geqslant 1$, 在这种情况下有 $\xi(\mathcal{L}, r_1) > k$, 所以 $\xi(\mathcal{L}, r_{j+1}) > k$, $p_{j'} > k > 2\gamma$ ($k > 2\gamma$ 是定理中的要求之一). 因此有 $r_{j+1} \leqslant \gamma\lambda(\mathcal{L}) < \lambda(\mathcal{L})p_{j'}/2$, 满足应用稀疏定理 8 的要求. 事实上, 通常情况下 k 要远比常数级的 γ 大, 一般不少于 n 的多项式 (甚至是指数) 函数, 因此满足稀疏定理 8 要求的素数一定存在.

下面讨论上述改进后的归约参数与文献 [9] 中归约参数的关系. 文献 [9] 中使用了引理 7 计算归约算法的成功概率下界, 即每次成功的下界由不等式 (1) 界定. 上面的方案使用引理 8 计算归约算法的成功概率下界, 即每次成功的下界由不等式 (2) 界定. 当归约算法中的参数 k 足够大时, 不等式 (1) 和 (2) 的右侧都可以近似写成下面的形式:

$$\big(\xi_1/p\big)\big(1-(\xi_2/p)\big), \tag{4}$$

其中 ξ_1, ξ_2 为常数. 再次根据关于 x 的函数 $x(1-ax), x \in (0, 1/a)$ 的性质, 素数 p 越接近 $2\xi_2$ 式 (4) 的值越大. 引理 7 中涉及的素数 p 与本原格点个数 ξ_2 需要满足 $\xi_2 \leqslant p/(20\log p)$, 代入式 (4), 得到的单次成功概率下界不超过:

$$\Big(\xi_1/(\xi_2 20\log p)\Big)\Big(1-\big(1/(20\log p)\big)\Big). \tag{5}$$

而引理 8 则允许 p 充分接近 $2\xi_2$, 并且可以保证存在满足 $(1+1/c^2)\xi_2 \leqslant p \leqslant (1+c^2)\xi_2$ 的素数, 将素数取值的上下界代入式 (4), 得到的单次成功概率下界至少为

$$\Big(\xi_1/\big(\xi_2(1+c^2)\big)\Big)\Big(1-\big(1/(1+c^2)\big)\Big). \tag{6}$$

从式 (5) 和 (6) 可见, 同样参数下改进的归约算法每一轮的成功概率下界更大. 另外, 文献 [9] 中的归约算法证明中假设了 $2^a > \gamma'^n$, 这一条件隐含了 $\gamma = \gamma'^{n/a} < 2$. 本文的证明过程中并无此种限制, γ 可以根据需要随意选择, 因此该归约算法支持的参数范围比文献 [9] 中的更大.

3.4 亚指数时间的结论

第一个结果. 首先在此给出一个时间上界为 $2^{O(n/\log n)}$ 的结果. 令 $l = O(\log n)$, $k = 2^{O(n/\log n)}$, 根据格点个数上界 $|\{y \in \mathcal{L} \mid \|y\| \leqslant \gamma\lambda(\mathcal{L})\}| \leqslant 2\lceil 2\gamma\rceil^n - 1$ 调整各参数, 可以得到以下定理.

定理5 给定任意常数 $\gamma \geqslant 1$, 令正整数 $l = O(\log n)$, $k = 2^{O(n/\log n)}$, $h \leqslant \mathrm{poly}(n)$. 存在一个随机算法将 γ-SVP 归约到 γ'-uSVP. 其中 $\gamma' = \gamma^{1/l}$, 该随机算法的运行时间上界为 $2^{O(n/\log n)}$, 并且其成功概率有大于零的常数下界.

将各个参数代入定理 4 即可得到上面的结论, 因此其证明也与之前一致. 更加值得关注的是这个结论得到的近似因子 γ' 的大小, 对此可以使用下面的引理来给出 γ' 的下界.

引理10 对于任意给定的常数 γ 和 c, 令 $\gamma' = 1 + c/\log n$, 将 l 设置为 $c'\log n$, 那么总存在一个常数 c' 使得不等式 $\gamma'^l \geqslant \gamma$ 成立.

证明 对于任意给定的常数 γ 和 c, 将 c' 设置为一个大于 $\ln\gamma/c$ 的数. 记 $n'=\log n/c$, 注意到,

$$(1+c/\log n)^{\ln\gamma\log n/c}=\left((1+1/n')^{n'}\right)^{\ln\gamma}\approx \mathrm{e}^{\ln\gamma}=\gamma.$$

因此, 当 $c'>\ln\gamma/c$ 时, 对于充分大的 n 总有 $\gamma'^{l}>\gamma$, 即 $\gamma'>\gamma^{1/l}$.

作为对比, 在多项式时间下, 已经证明的能够保证 uSVP 是 NP 困难的最大近似因子为 $(1+O(\log(n)/n))$, 对于充分大的 n 这里得到的近似因子 $1+c/\log n$ 更大.

另一方面, 注意到归约算法的运行时间上界主要取决于 k. 不妨将 k 写成 $2^{n/O(\log n)}$, 注意无论 $O(\log n)$ 所隐藏的常数多么小, 对于充分大的 n 和任意的常数 ε 而言, 都有 $k\ll 2^{\varepsilon n}$.

第二个结果. 现在来探索上面的归约算法在允许亚指数时间算法情况下的表现, 不妨假定一个固定的时间上界 $2^{\delta n}$.

首先做出以下参数设置: $l=O(1)$, $k=2^{o(n)}$, 根据格点个数上界 $|\{y\in\mathcal{L}\mid\|y\|\leqslant\gamma\lambda(\mathcal{L})\}|\leqslant 2\lceil 2\gamma\rceil^n-1$ 调整各个参数, 可以得到下面的定理.

定理6 给定任意常数 $\gamma\geqslant 1$, 选择一个合适的常数 ε, 令 $l=O(1)$, $k=2^{\varepsilon n}$, $h\leqslant\mathrm{poly}(n)$. 存在一个随机算法将 γ-SVP 归约到 γ'-uSVP. 其中 $\gamma'=\gamma^{1/l}$, 该随机算法的运行时间上界为 $\mathrm{poly}(n)\cdot k<2^{\delta n}$, 并且其成功概率有大于零的常数下界.

对于任意常数 γ 和选定的 l, 上述定理得到的近似因子 γ' 是一个常数. 现在来确定上述定理对什么样的 l 能够保持成立. 当 l 是比较大的常数时, 只要调整 k 到适当大小即可使归约算法的总运行时间在 $2^{\delta n}$ 之内. 显然, 在 γ 已经确定的情况下, l 越小 ,γ' 越大, 因此需要考察能够保持定理成立的最小的 l 的值. 由于需要参数 l 和 k 满足 $k^l\geqslant N_\gamma$, 而 k 直接决定了算法的运行时间, 不妨先确定 k, 将其设置为 $2^{\varepsilon n}$, 将前面的不等式改写成关于 l 的不等式: $l>(1+\log(\gamma+1)+1/n)/\varepsilon$. 至此有结论: 定理 6 对所有大于 $(1+\log(\gamma+1))/\varepsilon$ 的正整数 l 都成立.

复杂性结论一. 根据已有结果 [16] 以接近 1 的常数近似求解 CVP 是 GapSETH 困难的, 对于任意常数 ε, 都存在一个常数 $\gamma(\varepsilon)>1$, 在 GapSETH 下不存在运行时间小于 $2^{(1-\varepsilon)n}$ 的可以在参数 γ 内近似求解 CVP 的算法. 结合文献 [14] 中从 CVP 到 SVP 的归约, 不难看出此结论同样适用于 SVP. 现在加上前面得到的从 SVP 到 uSVP 的归约 (定理 6), 可以得出结论: 求解参数为接近于 1 的常数的 uSVP 在 GapSETH 下也是困难的 (有亚指数级的复杂性下界).

3.5 BDD 的 GapSETH 困难性

现在来展示上文中关于 uSVP 的复杂性结论所揭示的 BDD 在相应情况下的复杂性结论. 本小节的结论得益于文献 [11] 中给出的从 γ-uSVP 到 $(1/\gamma)$-BDD 的归约.

复杂性结论二. 结合文献 [11] 中的从 γ-uSVP 到 $(1/\gamma)$-BDD 的归约和前面得到的关于 uSVP 的 GapSETH 困难性结论, 可以得到结论: 求解参数为小于 1 的常数的 BDD 也是 GapSETH 困难的. 需要在此强调, 仅从 BDD 实例参数的角度而言, 这个结果在一定程度上弥补了现有最好结果 [17] 的一些不足 (文献 [17] 中得到的 BDD 实例的参数不一定小于 1).

3.6 进一步改进的可能性

最后再来看一下文献 [9] 和本文使用的归约算法. 二者本质上只做两件事, 首先采样若干素数, 然后以每个素数和给定的格为输入使用引理 2. 由于引理 2 的随机性, 在应用此引理时无法预测会得到什么, 因此需要借用格点划分来提供一个明确的成功概率下界.

成功概率. 使用符号 $\mathcal{P}$ 来代表对格点的划分, 字母 p 代表采样得到的素数. 根据前文所给出的灵活划分结构, 对于常数 γ'、给定的格 $\mathcal{L}$ 和素数 p 来说, 单次使用稀疏定理得到的子格 $\mathcal{L}'$ 符合 γ'-uSVP 定义的概率为 $\max_{\mathcal{P}}\{\Pr[\exists r, \lambda_1(\mathcal{L}') \leqslant r, \lambda_2(\mathcal{L}') > \gamma' r, \mathrm{rank}(\mathcal{L}' \cap \mathcal{B}(\gamma' r)) = 1 \mid \mathcal{P}]\}$.

最终参数. 假设要求前述归约算法的单次运行成功概率不小于 $\delta(n)$, 那么所能得到的 uSVP 实例的参数下界为 $\max_{\mathcal{P}}\{\gamma' \mid \Pr[\exists r, \lambda_1(\mathcal{L}') \leqslant r, \lambda_2(\mathcal{L}') > \gamma' r, \mathrm{rank}(\mathcal{L}' \cap \mathcal{B}(\gamma' r)) = 1 \mid \mathcal{P}] \geqslant \delta\}$.

需要强调的是, 无论是本文在定理 4 中给出的成功概率下界, 还是文献 [9] 中给出的成功概率下界都只考虑了一种划分方式, 如果考虑更多的划分方式, 有机会获得更好的成功概率下界.

4 稳定格上的 uSVP

如果格 $\mathcal{L}$ 满足 $\forall \mathcal{L}' \subset \mathcal{L}, \det(\mathcal{L}') \geqslant 1$ 则称为稳定格 [39]. 根据定理 2, 在稳定格上可以得到比定理 1 中刻画得更好的格点个数上界. 本节将放松定理 4 中对 γ' 的限制, 然后讨论其结合不同的格点上界 (尤其是稳定格的格点上界) 得到的归约参数.

前文在使用稀疏定理构建从 SVP 到 uSVP 的归约时, 为了分析简便, 做出要求 $\gamma' < \sqrt{2}$. 也因为这个限制条件, 能够得到的 uSVP 实例的参数不能超过 $\sqrt{2}$, 具有一定的局限性.

根据引理 5 , 可以将参数 γ' 的上界适当地放松, 这样一来在保证定理正确性的前提下或许能适当提高 uSVP 实例的参数范围. 具体来说, 可以给出如下引理.

引理11 对于任意的格 $\mathcal{L}(B) \in \mathbb{R}^n$, 素数 p, 实数 $r_1 < r_2$ 和向量 $z \xleftarrow{U} \mathbb{Z}_p^n$, 令 c 为使区间 $[c^i, c^{i+1}]_{0 \leqslant i \leqslant h}$ 上可以高效地进行素数采样的常数, j 为满足 $c^j \leqslant (c + 1/c)\xi(\mathcal{L}, r_2) \leqslant c^{j+1}$ 的最小整数, 从区间 $[c^j, c^{j+1}]$ 上采样得到一个素数 p, 考虑下面的格: $\mathcal{L}' = \{v \in \mathcal{L} \mid \langle z, B^{-1}x \rangle \equiv 0 \bmod p\}$, 如果给定的实数满足 $r_2 < \lambda(\mathcal{L})p/2$, 那么有 $\Pr[\lambda_1(\mathcal{L}') \leqslant r_1,\ \lambda_2(\mathcal{L}') > r_2 \mid \xi(\mathcal{L}, r_2) = 1] \geqslant \frac{c^2}{(1+c^2)^2} \cdot \frac{\xi(\mathcal{L}, r_1)}{\xi(\mathcal{L}, r_2)}$.

有了引理 11 之后, 就能给出改进后理论上支持比较大的参数范围的从 SVP 到 uSVP 的归约框架, 从而突破前文给出的定理 4 对 uSVP 的参数限制.

定理7 对于合适的素数 p, 任意实数 $\gamma < p/2$, 记 $N_\gamma = \sup_{\mathcal{L} \subset \mathbb{R}^n}\{\xi(\mathcal{L}, \gamma\lambda(\mathcal{L}))\}$, 令正整数 h, k, l 满足以下不等式 $k > \gamma, k^l \geqslant N_\gamma, h \leqslant \mathrm{poly}(n), l < \mathrm{poly}(n)$. 令 $\gamma' = \gamma^{1/l}$, c 为使区间 $[c^i, c^{i+1}]_{0 \leqslant i \leqslant h}$ 上可以高效地进行素数采样的常数, 并且满足不等式 $c^h \geqslant (c + 1/c)N_\gamma$. 存在一个随机算法将 γ-SVP 归约到 γ'-uSVP, 该算法的运行时间上界为 $\mathrm{poly}(n) \cdot k$, 其成功概率有大于零的常数下界.

证明 现在来证明, 在替换了参数限制之后, 定理 4 的证明仍然适用于现在的情形.

归约算法. 这里使用的归约步骤保持不变, 照旧先采样素数, 然后使用稀疏定理, 最后调用 uSVP 谕言机并重复独立地执行这些步骤.

时间上界. 由于 h 和 N_γ 并不受新参数限制的影响, 所以算法的运行时间上界也保持不变.

成功概率. 这里保持定理 4 的概率下界证明中的格点划分方式不变. 对于满足 $\xi(\mathcal{L}, r_{j+1}) \leqslant k \cdot \xi(\mathcal{L}, r_j)$ 的整数 j, 由于限制了 $\gamma < p/2$, 因此 $r_{j+1} \leqslant \gamma\lambda(\mathcal{L}) < \lambda(\mathcal{L})p/2$. 这次, 在进行概率分析时仅考虑满足 $\xi(\mathcal{L}', r_j) = \xi(\mathcal{L}', r_{j+1}) = 1$ 的子格 $\mathcal{L}'$. 根据引理 11 在满足以上要求的子格 $\mathcal{L}'$ 中, 长度不超过 r_{j+1} 的非零格点要么是满足 $\|v\| = \lambda(\mathcal{L}')$ 的唯一一个本原格点, 要么是此本原格点的整数倍, 并且与 v 线性无关的格点的长度都大于 r_{j+1}. 根据引理 11, 可以得到单次的成功概率下界:

$$\Pr[\lambda_1(\mathcal{L}') \leqslant r_j,\ \lambda_2(\mathcal{L}') > r_{j+1} \mid \xi(\mathcal{L}', r_{j+1}) = 1] \geqslant \frac{c^2}{(1 + c^2)^2 k}.$$

剩余分析与前文中一致, 这里不再重复.

下面来讨论定理 7 能够带来的结论. 注意, 这里仅考虑允许运行时间为 $2^{\varepsilon n}$ 的算法的情形. 首先类比于定理 6, 可以给出下面结论.

定理8 对于合适的 p, 给定任意实数 $1 \leqslant \gamma \leqslant p/2$, 令正整数 $l = O(1)$, $k = 2^{\varepsilon n}$, $h \leqslant \mathrm{poly}(n)$. 存在一个随机算法将 γ-SVP 归约到 γ'-uSVP. 其中 $\gamma' = \gamma^{\frac{1}{l}}$, 该随机算法的运行时间上界为 $\mathrm{poly}(n) \cdot k$, 并且其成功概率有大于零的常数下界.

如果这里仍使用 $2\lceil 2\gamma \rceil^n - 1$ 作为格点上界, 类似于定理 6, 需要保证 $l > (1 + \log(\gamma + 1) + 1/n)/\varepsilon$. 此时得到的 uSVP 实例的参数下界为 $\gamma^{\frac{1}{l}}$, 对 γ 取极限可以得到

$$\lim_{\gamma \to \infty} \gamma' < \lim_{\gamma \to \infty} \gamma^{\varepsilon/(1+\log(\gamma+1)+1/n)} < 2^{\varepsilon \lim_{\gamma \to \infty} (\log \gamma)/(1+\log(\gamma+1)+1/n)} = 2^{\varepsilon}.$$

可以看到在这种情况下即使在理论上放松了对 γ' 的限制, 但得到的 uSVP 实例的参数上界仍然远小于 $\sqrt{2}$. 可见限制 uSVP 实例参数的主要因素不是前面选择的参数限制, 而是使用的格点上界. 如果使用更紧致的本原格点个数上界, 则可以得到参数更大的结果.

对一般的格, 所有现有方法都只能粗略估算长度在某一范围内的格点个数, 无论是上界还是下界都十分模糊. 然而对于稳定格, 根据定理 2, 在欧几里得空间, 有以下不等式:

(1) 当 $r = O(\frac{\sqrt{n}}{\log(n)})$ 时, $|\mathcal{L} \cap (r\mathcal{B}_2^n)| \leqslant 2^{O(n)}$;

(2) 当 $r = O(\sqrt{n})$ 时, $|\mathcal{L} \cap (r\mathcal{B}_2^n)| \leqslant 2^{O(\frac{n}{2}\log\log(n))}$;

(3) 当 $r = O(\log(n)\sqrt{n})$ 时, $|\mathcal{L} \cap (r\mathcal{B}_2^n)| \leqslant 2^{O(n\log\log(n))}$.

下面考虑最短向量长度为常数级的稳定格 $\mathcal{L}$, 使用上述格点个数上界, 能够得到在这样的稳定格中关于 SVP 和 uSVP 复杂性之间联系的结论.

对于 $r = O(\frac{\sqrt{n}}{\log(n)})$, 有不等式 $|\mathcal{L} \cap r\mathcal{B}_2^n| \leqslant 2^{O(n)}$. 要想以不超过 $r/\lambda(\mathcal{L})$ 的近似因子找到 $\lambda(\mathcal{L})$ 的近似解, 那么只需找出 $\mathcal{L}$ 中长度不超过 r 的任一格点. 根据上面的结论, 这样的格点有至多 $2^{O(n)}$ 个.

现在考虑在有 uSVP 谕言机的情况下近似求解 SVP, 要想利用 uSVP 谕言机找到长度不超过 r 的格点, 就需要从格 $\mathcal{L}$ 中提取 uSVP 实例, 并且其最短向量的长度应该不超过 r. 同时, 所使用的 uSVP 谕言机也要尽可能的弱 (即生成的 uSVP 实例的参数尽可能的大).

令 $k = 2^{\varepsilon n}$, 由于长度不超过 $r = O(\frac{\sqrt{n}}{\log(n)})$ 的格点不多于 $2^{O(n)}$, 因此存在一个常数 $l = O(1)$ 使得 $k^l \geqslant 2^{O(n)}$. 根据定理 7, 此时得到的 uSVP 实例的参数不小于 $(r/\lambda(\mathcal{L}))^{1/l} = O((n/\log^2 n)^{1/(2l)})$. 由于 l 是一个常数, 此处得到的 uSVP 实例参数在理论上远超常数.

对 $r = O(\sqrt{n})$ 以及 $r = O(\log(n)\sqrt{n})$, 有不等式 $|\mathcal{L} \cap r\mathcal{B}_2^n| \leqslant 2^{O(n\log\log(n))}$. 此时将 l 设置为 $O(\log\log(n))$, k 仍然设置为 $2^{\varepsilon n}$. 代入定理 8, 有下面的关系式:

$$\gamma' = \gamma^{\frac{1}{l}} = (C_1\sqrt{n})^{\frac{1}{C_2\log\log(n)}} = 2^{\frac{\log(C_1)}{C_2\log\log(n)} + \frac{\log(n)}{2C_2\log\log(n)}} \approx n^{1/O(\log\log(n))},$$

其中 C_1, C_2 为 O 符号隐藏的常数. 同样的, 当 $r = O(\log(n)\sqrt{n})$ 时, 可以得到

$$\gamma' = (C_1\log(n)\sqrt{n})^{\frac{1}{C_2\log\log(n)}} = 2^{\frac{\log(C_1)}{C_2\log\log(n)} + \frac{\log(n)}{2C_2\log\log(n)} + \frac{1}{C_2}} \approx n^{1/O(\log\log(n))}.$$

需要特别指出的是, 根据文献 [19], 上面讨论的参数范围的 SVP 不再是 GapSETH 困难的了. 但仍然可以假设现实中不存在能够在亚指数时间内求解该参数范围的 SVP 实例的算法. 需要在此强调, 这里给出的有关 uSVP 的结论也可以使用文献 [11] 中的归约转化成关于 BDD 的结论.

5 总结

本文对一个从 SVP 到 uSVP 的归约进行了改进. 通过改变归约算法中的素数采样策略, 提高了归约算法的成功概率; 其次, 使用新的参数限制条件保证了算法的正确性. 从本原格点个数的角度, 定理 4 使用的限制条件允许选取更接近本原格点个数的素数, 不仅提高了归约的成功概率, 而且为得到参数更大的从 SVP 到 uSVP 的归约奠定了基础.

在超多项式时间条件下, 本文得到了常数参数的从 SVP 到 uSVP 的归约, 这样的参数比在多项式条件下的对应结果好很多. 结果表明: 相较于在多项式条件下, 在超多项式条件下 SVP 和 uSVP 二者的复杂性联系更加紧密.

结合稳定格上的格点个数上界, 进一步得到了参数远超常数级别的从 SVP 到 uSVP 的归约, 这一结果对建立超多项式时间下的格密码体系提供了理论参考.

同时, 本文得到的所有关于 uSVP 的复杂性结论都可以几乎无损失地转换成 BDD 在对应条件下的复杂性结论. 尤其是结合 uSVP 在 GapSETH 下的困难性结果得到的对应条件下 BDD 的困难性结果, 能够保证得到的 BDD 实例的参数小于 1. 从这个角度, 该结果比现有的关于 BDD 在 GapSETH 下的最好的困难性结果 [17] 更好.

致谢 审稿人提出了许多建设性意见, 提高了文章的可读性, 作者表示衷心感谢.

参考文献

1 Ajtai M. Generating hard instances of lattice problems (extended abstract). In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing. New York: Association for Computing Machinery, 1996. 99–108
2 Ajtai M, Dwork C. A public-key cryptosystem with worst-case/average-case equivalence. In: Proceedings of the 29th Annual ACM Symposium on Theory of Computing. New York: Association for Computing Machinery, 1997. 284–293
3 Regev O. New lattice-based cryptographic constructions. J ACM, 2004, 51: 899–942
4 Ajtai M, Dwork C. The first and fourth public-key cryptosystems with worst-case/average-case equivalence. Electronic Colloquium on Computational Complexity, TR07-097. 2007
5 Regev O. On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing. New York: Association for Computing Machinery, 2005. 84–93
6 Peikert C. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing. New York: Association for Computing Machinery, 2009. 333–342
7 Kumar R, Sivakumar D. On the unique shortest lattice vector problem. Theor Comput Sci, 2001, 255: 641–648
8 Aggarwal D, Dubey C K. Improved hardness results for unique shortest vector problem. Inf Process Lett, 2016, 116: 631–637
9 Stephens-Davidowitz N. Search-to-decision reductions for lattice problems with approximation factors (slightly) greater than one. In: Approximation, Randomization, and Combinatorial Optimization. Dagstuhl: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. 1–18
10 Lyubashevsky V. The n^c-unique shortest vector problem is hard. Cryptology ePrint Archive, 2008. https://eprint.iacr.org/2008/504
11 Lyubashevsky V, Micciancio D. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In: Proceedings of the 29th Annual International Cryptology Conference. Berlin: Springer, 2009. 577–594
12 Jin B, Xue R. Improved lower bound for the complexity of unique shortest vector problem. Cybersecurity, 2023, 6: 38
13 Bennett H, Golovnev A, Stephens-Davidowitz N. On the quantitative hardness of CVP. In: Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science, 2017. 13–24
14 Aggarwal D, Stephens-Davidowitz N. (Gap/S)ETH hardness of SVP. In: Proceedings of the 50th Annual ACM SIGACT

Symposium on Theory of Computing. New York: Association for Computing Machinery, 2018. 228–238
15 Aggarwal D, Chung E. A note on the concrete hardness of the shortest independent vector in lattices. Inf Process Lett, 2021, 167: 106065
16 Aggarwal D, Bennett H, Golovnev A, et al. Fine-grained hardness of CVP(P): everything that we can prove (and nothing else). In: Proceedings of the 32nd Annual ACM-SIAM Symposium on Discrete Algorithms, 2021. 1816–1835
17 Bennett H, Peikert C, Tang Y. Improved hardness of BDD and SVP under Gap-(S)ETH. In: Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS 2022), 2022. 1–12
18 Bai S, Stehlé D, Wen W. Improved reduction from the bounded distance decoding problem to the unique shortest vector problem in lattices. In: Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming, 2016. 1–12
19 Aggarwal D, Bennett H, Brakerski Z, et al. Lattice problems beyond polynomial time. In: Proceedings of the 55th Annual ACM Symposium on Theory of Computing. New York: Association for Computing Machinery, 2023. 1516–1526
20 Aggarwal D, Chen Y, Kumar R, et al. Dimension-preserving reductions between SVP and CVP in different p-norms. In: Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 2021. 2444–2462
21 Khot S. Hardness of approximating the shortest vector problem in lattices. In: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, 2003. 126–135
22 Stephens-Davidowitz N. Discrete Gaussian sampling reduces to CVP and SVP. In: Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms, 2016. 1748–1764
23 Boas P. Another NP-Complete Partition Problem and the Complexity of Computing Short Vectors in a Lattice. Technical Report, 1981. https://staff.fnwi.uva.nl/p.vanemdeboas/vectors/mi8104c.html
24 Ajtai M. The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In: Proceedings of the 30th Annual ACM Symposium on Theory of Computing. New York: Association for Computing Machinery, 1998. 10–19
25 Micciancio D. The shortest vector in a lattice is hard to approximate to within some constant. In: Proceedings of the 39th Annual Symposium on Foundations of Computer Science, 1998. 92–98
26 Khot S. Hardness of approximating the shortest vector problem in high l_p norms. In: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, 2003. 290–297
27 Micciancio D. Inapproximability of the shortest vector problem: toward a deterministic reduction. Theory Comput, 2012, 8: 487–512
28 Sauer N. On the density of families of sets. J Combinatorial Theor Ser A, 1972, 13: 145–147
29 Shelah S. A combinatorial problem; stability and order for models and theories in infinitary languages. Pac J Math, 1972, 41: 247–261
30 Vapnik V N, Chervonenkis A Y. On the uniform convergence of relative frequencies of events to their probabilities. In: Measures of Complexity: Festschrift for Alexey Chervonenkis. Cham: Springer International Publishing, 2015. 11–30
31 Haviv I, Regev O. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In: Proceedings of the 39th Annual ACM Symposium on Theory of Computing. New York: Association for Computing Machinery, 2007. 469–477
32 Bennett H, Peikert C. Hardness of the (approximate) shortest vector problem: a simple proof via reed-Solomon codes. In: Approximation, Randomization, and Combinatorial Optimization. Dagstuhl: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2023. 1–20
33 Goldreich O, Goldwasser S. On the limits of nonapproximability of lattice problems. J Comput Syst Sci, 2000, 60: 540–563
34 Aharonov D, Regev O. Lattice problems in NP ∩ coNP. J ACM, 2005, 52: 749–765
35 Cai J Y. A relation of primal-dual lattices and the complexity of shortest lattice vector problem. Theor Comput Sci, 1998, 207: 105–116
36 Bennett H. The complexity of the shortest vector problem. SIGACT News, 2023, 54: 37–61
37 Regev O, Rosen R. Lattice problems and norm embeddings. In: Proceedings of the 38th Annual ACM Symposium on Theory of Computing. New York: Association for Computing Machinery, 2006. 447–456

38 Liu Y, Lyubashevsky V, Micciancio D. On bounded distance decoding for general lattices. In: Approximation, Randomization, and Combinatorial Optimization. Berlin: Springer, 2006. 450–461
39 Regev O, Stephens-Davidowitz N. A reverse Minkowski theorem. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing. New York: Association for Computing Machinery, 2017. 941–953
40 Impagliazzo R, Paturi R. On the complexity of k-SAT. J Comput Syst Sci, 2001, 62: 367–375
41 Impagliazzo R, Paturi R, Zane F. Which problems have strongly exponential complexity? J Comput Syst Sci, 2001, 63: 512–530
42 Manurangsi P. Approximation and hardness: beyond P and NP. Dissertation for Ph.D. Degree. Berkeley: University of California, Berkeley, 2019
43 Betke U, Henk M, Wills J M. Successive-minima-type inequalities. Discrete Comput Geom, 1993, 9: 165–175

Fine-grained hardness of the unique shortest vector problem in lattices

Baolong JIN[1,2*] & Rui XUE[1,2]

1. *Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100085*, China*;
2. *School of Cybersecurity, University of Chinese Academy of Sciences, Beijing* 100049*, China*
* Corresponding author. E-mail: jinbaolong@iie.ac.cn

Abstract The intractability of uSVP (unique shortest vector problem) is the foundation of the security of some popular lattice-based cryptographic schemes. The study of its computational complexity plays an important part in the lattice-based cryptography. However, the research progress on the difficulties of uSVP under different parameters is very slow. There is also a lack of research on the fine-grained hardness of uSVP. In this work, we presented an improved reduction that allows for a fine-grained reduction of SVP (shortest vector problem) instances to uSVP instances. Our reduction has a potentially greater success probability and allows more flexible parameters. Using this reduction, we demonstrated the following results: (1) we presented a reduction from SVP to uSVP with greater parameters, in a super polynomial setting; (2) assuming the GapSETH (gap strong exponential time hypothesis), we presented a reduction from SVP to uSVP with subexponential running time and constant parameters, which proved the intractability of uSVP under GapSETH; (3) all results above can be transferred into the hardness results of BDD (bounded distance decoding), hence BDD with a parameter strictly smaller than 1 is also intractable assuming GapSETH; (4) by combining our reduction with an upper bound for the number of lattice points in stable lattices, we presented several reductions from SVP to uSVP with parameters beyond constant.

Keywords shortest vector problem, unique shortest vector problem, computational complexity, fine-grained complexity, complexity reduction