Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 170 (2022)

The Complexity of the Shortest Vector Problem*

Huck Bennett†

January 31, 2023

Abstract

Computational problems on point lattices play a central role in many areas of computer science including integer programming, coding theory, cryptanalysis, and especially the design of secure cryptosystems. In this survey, we present known results and open questions related to the *complexity* of the most important of these problems, the Shortest Vector Problem (SVP).

*This work is set to appear in the SIGACT News Open Problems Column.

†Oregon State University, huck.bennett@oregonstate.edu.

ISSN 1433-8092

Contents

1 Introduction **1**
1.1 Definitions . . . 2

2 Foundational Complexity **3**
2.1 Early Results, Algorithms, and NP-hardness . . . 3
2.2 Hardness of Approximation . . . 5
2.3 Fine-Grained Hardness . . . 5

3 Showing Hardness: A Technical Summary **7**
3.1 Showing Small-Factor Hardness of Approximation: A First Attempt . . . 8
3.2 The Ajtai-Micciancio and Khot Reductions . . . 8
3.3 Amplifying Hardness . . . 10
3.4 Constructing Locally Dense Lattices . . . 11

4 Additional Complexity **12**
4.1 GapSVP with Polynomial Approximation Factors . . . 12
4.2 Variants of SVP and Reductions Between Them . . . 15
4.3 Complexity for Cryptography . . . 16

1 Introduction

Intuitively, a lattice is a regular ordering of points in n-dimensional space.[1] Lattices are classically studied mathematical objects, and in the last 40 years have found a large number of applications in computer science. These include algorithms for integer programming (starting with [Len83, Kan87]; see [Dad12] for a more recent reference), coding theory (especially coding in the Gaussian channel; see, e.g., [CS99, Chapter 3, Section 4.1]), cryptanalysis (e.g., attacks on knapsack cryptography [LO85] and low-exponent RSA [Cop01, Bon99]), and especially in the development of *lattice-based cryptography*, that is, cryptosystems whose security is based on the apparent intractability of computational problems on lattices.

Lattice-based cryptosystems have many attractive properties including being *post-quantum*, i.e., apparently being secure even against attackers equipped with quantum computers, provable security assuming the hardness of certain worst-case computational problems on lattices, and functionality that allows for constructing exciting and exotic primitives like fully-homomorphic encryption (FHE); see [Pei16] for a survey. Because of these properties (and more), in July 2022 the National Institute of Standards and Technology (NIST) standardized three lattice-based cryptosystems as the culmination of their years-long post-quantum cryptography standardization project [Nat22]. Furthermore, both the sole *primary* post-quantum key exchange protocol (CRYSTALS Kyber) and primary digital signature protocol (CRYSTALS Dilithium) standardized by NIST are lattice-based [ABD+17].

The focus of this survey is on the *Shortest Vector Problem* (SVP), which is a search problem, and its decision variant, GapSVP. The Shortest Vector Problem is the most important computational problem on lattices. In SVP, the goal is to find a shortest non-zero vector in an input lattice, and in GapSVP the goal is to determine whether the length of such a vector is above or below a given threshold. (We formally specify how a lattice is represented and define these problems shortly.)

As with any important computational problem, it is natural to ask about the *complexity* of SVP: Is it NP-hard? If so, is it hard to approximate? And, precisely how long does it take to solve? These questions are particularly important given the likely central role of lattice-based cryptography in applications in the near future. In particular, the security of lattice-based cryptography in practice relies on understanding the fine-grained hardness of SVP, and proving that GapSVP is sufficiently hard to approximate would prove that lattice-based cryptography is secure.

In this work, we summarize the complexity of (Gap)SVP in a wide range of regimes, discuss its connections to cryptography and other areas, and present a variety of open problems related to these things. We refer the reader to Figures 1, 3 and 4, which summarize the complexity of approximate GapSVP and show that the problem is interesting with a wide range of approximation factors. Finally, we note that many of the open problems we give are likely very challenging, and making any progress on them would still be very interesting (even without fully resolving them).

Additional surveys and open problems on lattice complexity. The book of Micciancio and Goldwasser [MG02] is the classic text on the complexity of lattice problems, although there have been many developments since it was published around 20 years ago. More recent resources on related topics include a blog post on the fine-grained complexity of lattice problems by Bennett, Golovnev, and Stephens-Davidowitz [BGS20], and a collection of open problems related to algorithms and complexity aspects of lattice problems more generally from the Simons Institute's Summer 2022 Lattices Program [Sim22].

Acknowledgements. The author would like to thank Bill Gasarch, Sasha Golovnev, Chris Peikert, Mike Rosulek, and Noah Stephens-Davidowitz for helpful comments and for pointing out typos. He would especially like to thank Noah for his detailed comments and for sharing his refined Latex link-coloring scheme, and Bill for inviting him to write for the SIGACT News Open Problems Column. Additionally, the author

[1] Unfortunately, the term "lattice" is overloaded to mean at least two different things in computer science and mathematics. This article is about the geometric objects (sometimes called *point lattices*, https://en.wikipedia.org/wiki/Lattice_(group)), and not about partial orders (https://en.wikipedia.org/wiki/Lattice_(order)).

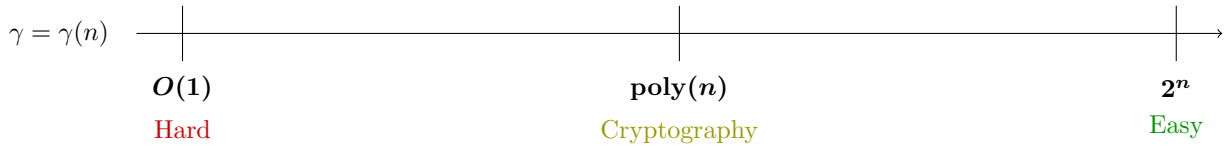

Figure 1: A simplified summary of the complexity of γ-GapSVP on lattices of dimension n for constant, polynomial, and exponential approximation factors $\gamma = \gamma(n)$. The problem is NP-hard (under randomized reductions) for $\gamma = O(1)$, and is solvable in polynomial time for $\gamma = 2^n$. If it is hard for $\gamma = n^c$ for a sufficiently large constant $c > 0$, then lattice-based cryptography is provably secure. For a more detailed summary, see Figure 3.

would like to thank the Simons Institute for hosting the Summer 2022 Lattices Program and Vinod Vaikuntanathan for inviting him to give a talk on the complexity of SVP there. That talk was the seed of this survey.

1.1 Definitions

Formally, a *lattice* $\mathcal{L}$ is the set of all integer linear combinations of some n linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \mathbb{R}^n$, and the matrix $\mathbf{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) \in \mathbb{R}^{n \times n}$ that has these vectors as its columns is called a *basis* of $\mathcal{L}$. That is, the lattice $\mathcal{L}$ generated by basis $\mathbf{B}$ is

$$\mathcal{L} = \mathcal{L}(\mathbf{B}) := \Big\{ \sum_{i=1}^{n} a_i \boldsymbol{b}_i : a_1, \ldots, a_n \in \mathbb{Z} \Big\} . \tag{1}$$

The value n is the *dimension* of $\mathcal{L}$, and is the important lattice parameter when discussing runtimes and (possibly super-constant) approximation factors $\gamma = \gamma(n)$ of computational problems on lattices.[2]

The ℓ_p norm of a vector $\boldsymbol{x} \in \mathbb{R}^n$ is defined as $\|\boldsymbol{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$ for $1 \le p < \infty$, and as $\|\boldsymbol{x}\|_\infty := \max_{i \in [n]} |x_i|$ for $p = \infty$. By default, we will work with the Euclidean norm ℓ_2, and will simply write $\|\boldsymbol{x}\|$ instead of $\|\boldsymbol{x}\|_2$.

We define $\lambda_1(\mathcal{L})$ to be the length of a shortest non-zero vector in $\mathcal{L}$. Equivalently, $\lambda_1(\mathcal{L})$ is the minimum distance between any two distinct vectors in $\mathcal{L}$, and so is also called the *minimum distance* of $\mathcal{L}$. Formally, we define

$$\lambda_1(\mathcal{L}) := \min_{\boldsymbol{v} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\boldsymbol{v}\| = \min_{\substack{\boldsymbol{u}, \boldsymbol{v} \in \mathcal{L}, \\ \boldsymbol{u} \neq \boldsymbol{v}}} \|\boldsymbol{u} - \boldsymbol{v}\| .$$

We next formally define the (approximate) decision and search versions of the Shortest Vector Problem. See also the illustration of approximate GapSVP in Figure 2.

Definition 1.1 (Decisional Shortest Vector Problem)**.** For $\gamma = \gamma(n) \ge 1$, the decision version of the γ-approximate Shortest Vector Problem (γ-GapSVP) is defined as follows. On input a basis $\mathbf{B} \in \mathbb{Q}^{n \times n}$ of a lattice $\mathcal{L}$ and a distance threshold $r > 0$, decide whether the input satisfies

- (YES instance) $\lambda_1(\mathcal{L}) \le r$, or
- (NO instance) $\lambda_1(\mathcal{L}) > \gamma r$

when one of these cases is promised to hold.

[2] One may define lattices more generally in terms of bases $\mathbf{B} \in \mathbb{R}^{m \times n}$ for $m > n$. In this case, m is the (ambient) *dimension* of the lattice, and n is its *rank*. The dimension m typically plays a secondary role, and often we can assume essentially without loss of generality that $m = n$ by projecting, which we do in most of this survey.

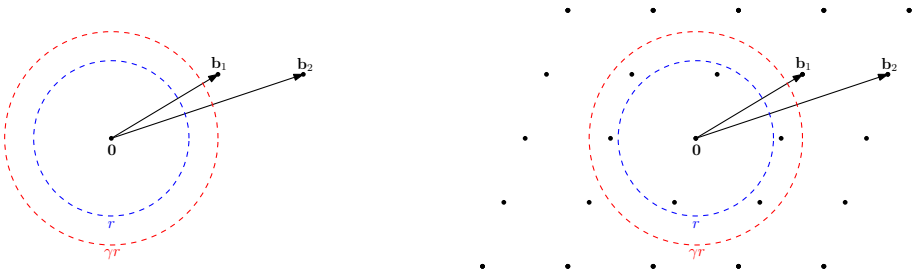

Figure 2: **Left:** An instance of γ-GapSVP consisting of a basis $\mathbf{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2)$ and $r > 0$. The goal is to decide whether there exists a non-zero lattice vector of norm at most r (i.e., there exists such a vector inside the blue circle), or if all non-zero lattice vectors are of norm greater than γr (i.e., all such vectors are strictly outside the red circle) when one of these cases is promised to hold. **Right:** Looking at lattice points in $\mathcal{L}(\mathbf{B})$ near the origin shows that the instance is a YES instance of γ-GapSVP because the non-zero lattice vectors $\pm(2\boldsymbol{b}_1 - \boldsymbol{b}_2)$ are inside the blue circle of radius r.

Definition 1.2 (Search Shortest Vector Problem)**.** For $\gamma = \gamma(n) \geq 1$, the search version of the γ-approximate Shortest Vector Problem (γ-SVP) is defined as follows. On input a basis $\mathbf{B} \in \mathbb{Q}^{n \times n}$ of a lattice $\mathcal{L}$, output a non-zero vector $\boldsymbol{v} \in \mathcal{L}$ satisfying $\|\boldsymbol{v}\| \leq \gamma \cdot \lambda_1(\mathcal{L})$.

We also define the decision version of the Closest Vector Problem, GapCVP, a closely related problem that will arise when talking about GapSVP. For a vector $\boldsymbol{t}$ and a lattice $\mathcal{L}$, define the distance between $\boldsymbol{t}$ and $\mathcal{L}$ as $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) := \min_{\boldsymbol{v} \in \mathcal{L}} \|\boldsymbol{t} - \boldsymbol{v}\|$.

Definition 1.3 (Decisional Closest Vector Problem)**.** For $\gamma = \gamma(n) \geq 1$, the decision version of the γ-approximate Closest Vector Problem (γ-GapCVP) is defined as follows. On input a basis $\mathbf{B} \in \mathbb{Q}^{n \times n}$ of a lattice $\mathcal{L}$, a vector $\boldsymbol{t} \in \mathbb{Q}^n$, and a distance threshold $r > 0$, decide whether the input satisfies

- (YES instance) $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) \leq r$, or
- (NO instance) $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) > \gamma r$

when one of these cases is promised to hold.

The vector $\boldsymbol{t}$ in a GapCVP instance is often called the *target vector*.

In the exact case when $\gamma = 1$, we simply write the above problems as GapSVP, SVP, and GapCVP respectively.[3] Finally, we note that although the definitions of $\lambda_1(\mathcal{L})$ and $\mathrm{dist}(\boldsymbol{t}, \mathcal{L})$, and Definitions 1.1 to 1.3 all use the Euclidean norm, one can generalize these definitions to arbitrary norms, including to ℓ_p norms with $p \neq 2$. We will sometimes discuss such generalizations.

2 Foundational Complexity

We start by surveying foundational complexity results related to (Gap)SVP, including NP-hardness, hardness of approximation, and fine-grained hardness. We again refer the reader to Figures 3 and 4, which summarize many of the results described here and in Section 4. Specifically, Figure 3 captures the complexity of γ-GapSVP in a "polynomial-time world," and Figure 4 captures the complexity of γ-GapSVP in an "exponential-time world."

2.1 Early Results, Algorithms, and NP-hardness

The formal study of the complexity of lattice problems began around 40 years ago with work of van Emde Boas [vEB81], who proved hardness of the GapCVP in the ℓ_2 norm. van Emde Boas also proved that GapSVP

[3] We emphasize that, somewhat confusingly, the "Gap" qualifier in GapSVP indicates that the problem is a decision problem, and not that it is an approximation problem. In particular, there are exact and approximate versions of both GapSVP (the decision version of SVP) and SVP (the search version of SVP).

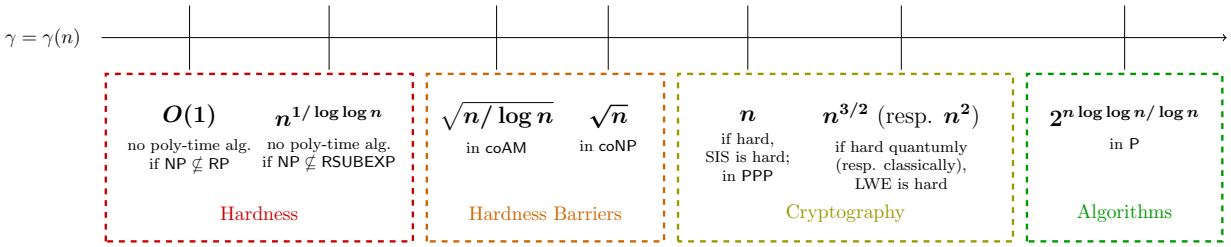

Figure 3: A summary of the complexity of γ-GapSVP on lattices of dimension n for approximation factors $\gamma = \gamma(n)$, with some constants and lower-order terms omitted for clarity. The figure shows polynomial-time algorithms, worst-case to average-case reductions, and protocols, as well as hardness results ruling out polynomial-time algorithms.

The problem has no polynomial-time algorithm for any constant approximation factor $\gamma = O(1)$ or even near-polynomial factor $\gamma = n^{1/\log\log n}$ under standard complexity assumptions (shown in the red box labeled "Hardness"). On the algorithmic side, the problem is solvable in polynomial time for slightly sub-exponential $\gamma = 2^{O(n\log\log n/\log n)}$ (shown in the green box labeled "Algorithms").

The two most important problems in lattice-based cryptography, the Shortest Integer Solutions Problem (SIS) and the Learning With Errors Problem (LWE) are provably hard *on average*—with parameters that allow for constructing private-key and public-key cryptography, respectively—assuming that γ-GapSVP is hard in the worst case for certain polynomial approximation factors $\gamma = \text{poly}(n)$ (shown in the yellow box labeled "Cryptography").

Finally, γ-GapSVP is in coAM for $\gamma = O(\sqrt{n/\log n})$ and in coNP for $\gamma = O(\sqrt{n})$ (shown in the orange box labeled "Hardness Barriers"). These results are often interpreted as barriers to showing (NP-)hardness for the somewhat larger polynomial approximation factors γ relevant for the hardness of SIS and LWE, since NP-hardness of γ-GapSVP for $\gamma = \Omega(\sqrt{n/\log n})$ would result in an unexpected complexity theoretic consequence, namely, in the polynomial hierarchy collapsing.

Increasing either of the approximation factors γ in the "Hardness" box, or decreasing any of the approximation factors γ in the "Hardness Barriers," "Cryptography," or "Algorithms" boxes while keeping the corresponding assumption or consequence the same is an interesting open problem. See also Figure 4 for a summary of the complexity of γ-GapSVP in an "exponential-time world." An adapted version of this figure appears in [ABB+22].

in the ℓ_∞ norm is NP-hard, and conjectured that GapSVP in the ℓ_2 norm was NP-hard. Shortly thereafter, work of Lenstra, Lenstra, and Lovász [LLL82] complemented this work on hardness by giving a polynomial-time algorithm for $2^{O(n)}$-SVP—the famous LLL algorithm. Subsequently, the BKZ reduction algorithm of Schnorr [Sch87], slide reduction algorithm of Gama and Nguyen [GN08], and dual BKZ reduction algorithm of Miccancio and Walter [MW16] improved this by giving polynomial-time algorithms for γ-SVP with slightly subexponential approximation factor $\gamma = 2^{O(n\log\log n/\log n)}$ (see the "Algorithms" box in Figure 3).

Seventeen years after [vEB81], landmark work of Ajtai [Ajt98] largely resolved van Emde Boas's conjecture in the affirmative by showing that GapSVP is NP-hard, but with the caveat that the hardness reduction he used was *randomized*.[4] As a consequence, Ajtai showed that GapSVP is not in RP (or in P) assuming that NP $\neq$ RP, which is a stronger assumption than P $\neq$ NP.

In the following sections, we survey major improvements made to Ajtai's original result in terms of hardness of approximation and fine-grained hardness. However, despite progress on those fronts, no one has been able to give a derandomized hardness reduction even for exact GapSVP. Indeed, it remains a major open question to show that GapSVP is "properly" NP-hard.

Open Problem 2.1 (Deterministic NP-hardness)**.** *Prove that (exact or approximate)* GapSVP *in the* ℓ_2 *norm is* NP*-hard under a deterministic Karp reduction.*

In fact, it would be interesting to show such hardness in any ℓ_p norm for $p < \infty$, which is also unknown (as mentioned above, [vEB81] showed such hardness in the ℓ_∞ norm). See Sections 3.2 and 3.4 and Open Problems 3.4 and 3.5 for further discussion about derandomizing hardness reductions for GapSVP.

[4] Additionally, γ-GapSVP is in NP for any approximation factor $\gamma = \gamma(n) \geq 1$. The witness certifying that an instance $(\mathbf{B}, r)$ is a YES instance is simply a non-zero vector $\boldsymbol{v} \in \mathcal{L}(\mathbf{B})$ satisfying $\|\boldsymbol{v}\| \leq r$. So, showing deterministic NP-hardness of GapSVP would imply that it is NP-complete.

2.2 Hardness of Approximation

In the years following Ajtai's seminal work, a sequence of results showed (randomized) hardness of approximation of γ-GapSVP for successively larger approximation factors γ and in different ℓ_p norms [CN98, Mic98, Kho03, Kho04, RR06, HR07, Mic12]. The following result, which is stated as the main theorem in [HR07], is essentially the culmination of work on hardness of approximation for GapSVP so far.

Theorem 2.2 (Hardness of Approximation for GapSVP, [Kho04], [HR07, Theorem 1.1])**.** *For every $1 \leq p \leq \infty$, there is no randomized polynomial-time algorithm for γ-GapSVP on lattices of rank n in the ℓ_p norm with the following approximation factors $\gamma = \gamma(n)$ unless the corresponding assumption is violated:*

1. *Any constant $\gamma \geq 1$, unless $\mathsf{NP} \subseteq \mathsf{RP}$;*
2. *$\gamma = \gamma(n) = 2^{(\log n)^{1-\varepsilon}}$ for any constant $\varepsilon > 0$, unless $\mathsf{NP} \subseteq \mathsf{RTIME}[2^{\mathrm{poly}(\log n)}]$;*
3. *$\gamma = \gamma(n) = n^{c/\log\log n}$ for some constant $c > 0$, unless $\mathsf{NP} \subseteq \mathsf{RSUBEXP} = \cap_{\delta>0}\mathsf{RTIME}[2^{n^\delta}]$.*

We note that [Kho04] already essentially showed Theorem 2.2, Item 1, and also showed weaker versions of Items 2 and 3. Additionally, we note that Dinur [Din02] showed the hardness result in Item 3 for GapSVP in the ℓ_∞ norm under a much weaker assumption. Namely, she showed that $n^{c/\log\log n}$-GapSVP in the ℓ_∞ norm with some constant $c > 0$ is NP-hard under a deterministic reduction.

A natural question is whether even stronger hardness of approximation—specifically, hardness of γ-GapSVP for some polynomial approximation factor $\gamma = n^\varepsilon$—holds for GapSVP in some ℓ_p norm.

Open Problem 2.3 (Hardness of n^ε-GapSVP)**.** *Prove that there is no polynomial-time algorithm for γ-GapSVP with $\gamma = \gamma(n) = n^\varepsilon$ for some constant $\varepsilon > 0$ under a standard complexity assumption.*

One generic (if not completely standard) complexity assumption that might be useful for showing such hardness is the *Projection Games Conjecture* (PGC), a strong form of the Unique Games Conjecture formalized by Moshkovitz [Mos12]. Indeed, [Mos12] notes that PGC implies super-polynomial hardness of n^ε-GapCVP in the ℓ_2 norm, and asks whether a similar result holds for n^ε-GapSVP. Mukhopadhyay [Muk22] recently made progress in this direction by addressing the hardness of GapSVP in the ℓ_∞ norm under PGC.

2.3 Fine-Grained Hardness

The goal of *fine-grained hardness* as a field of study is to prove strong, precise lower bounds on the time complexity of certain computational problems under well-studied assumptions such as the Exponential Time Hypothesis (ETH) and Strong Exponential Time Hypothesis (SETH). In the context of lattice problems, the goal is generally to prove (conditional) *exponential* lower bounds on lattice problems—e.g., results of the form "γ-GapSVP has no $2^{\Omega(n)}$-time algorithm assuming ETH" or "γ-GapSVP has no $2^{n/20}$-time algorithm assuming SETH," as opposed to "γ-GapSVP has no $\mathrm{poly}(n)$-time algorithm assuming $\mathsf{RP} \neq \mathsf{NP}$," which is all that follows from the NP-hardness (under randomized reductions) of γ-GapSVP.[5]

Fine-grained hardness is closely related to the concept of *concrete security* in cryptography, where the goal is to understand the precise amount of time and space needed to break a cryptosystem using a (possibly heuristic) attack algorithm, and so is especially well-motivated by the need to understand the security of lattice-based cryptography in practice. Most notably for our setting, the "core-SVP" methodology introduced in [ADPS16] analyzes the concrete security of lattice-based cryptosystems in terms of the cost of an exact SVP oracle call (in terms of the rank n of the underlying lattice), and is the standard way of estimating such security. We also refer the reader to [ACD+18, Wal17], which use this methodology to analyze the concrete security of many leading lattice-based cryptosystems and discuss the concrete security of lattice-based cryptography.

The fastest known classical heuristic algorithms for SVP run in roughly $(3/2)^{n/2} \approx 2^{0.292n}$ time, and the fastest known quantum heuristic algorithms run in roughly $2^{0.265n}$ time [Laa15, LMvdP15, BDGL16].

[5] For more on the topics discussed in this section, we refer the reader to [BGS20], a survey focusing specifically on the fine-grained hardness of lattice problems.

These best known heuristic algorithms are all so-called "sieving algorithms." The fastest known provable classical algorithm for GapSVP runs in $2^{n+o(n)}$ time [ADRS15], and the fastest known provable quantum algorithm for GapSVP runs in either $2^{0.950n+o(n)}$ or $2^{0.835n+o(n)}$ time depending on the precise quantum model of computation used [ACKS21]. (Of course, closing the runtime gaps between (both classical and quantum) heuristic and provable algorithms is also a very interesting question, but this survey focuses on complexity rather than algorithms.)

Additionally, [ADPS16] asserts that "it is very plausible that the best quantum SVP algorithm would run in time greater than $[(4/3)^{n/2} \approx 2^{0.2075n}]$." This assertion assumes that sieving algorithms are essentially optimal for solving SVP (the lower bound of $(4/3)^{n/2}$ comes from the space complexity of these algorithms). An important goal for fine-grained complexity is to try to *prove* that this lower bound holds against *all* algorithms.

Indeed, because the core-SVP methodology and time-complexity estimates mentioned above are widely used, the difference between a $2^{0.265n}$-time versus $2^{n/20}$-time versus $2^{\sqrt{n}}$-time algorithm for exact or near-exact SVP means the difference between many real-world lattice-based cryptosystems being secure, insecure with current parameters, and effectively broken in practice. (Variants of this observation already appear throughout the literature on the fine-grained hardness of lattice problems; it serves as important motivation for the whole research area. For example, [BGS20] makes a similar observation.)

Assumptions for fine-grained hardness. The *Exponential Time Hypothesis (ETH)* asserts that there is no $2^{o(n)}$-time algorithm for the 3-SAT problem on formulas with n variables, and the *Strong Exponential Time Hypothesis (SETH)* asserts that for every $\varepsilon > 0$ there exists $k \in \mathbb{Z}^+$ such that there is no $2^{(1-\varepsilon)n}$-time algorithm for the k-SAT problem on formulas with n variables.[6] These well-studied assumptions (and their variants) are very useful for proving (conditional) fine-grained hardness results about lattice problems. In particular, as we discuss shortly, recent work has shown $2^{\Omega(n)}$-time lower bounds on algorithms for lattice problems assuming variants of ETH, and even 2^{cn}-time lower bounds with explicit constants $c > 0$ assuming variants of SETH.

Known results on fine-grained hardness. Work of Bennett, Golovnev, and Stephens-Davidowitz [BGS17] initiated the study of the fine-grained hardness of lattice problems by showing roughly 2^n hardness of GapCVP in most ℓ_p norms assuming SETH, and follow-up work by Aggarwal, Bennett, Golovnev, and Stephens-Davidowitz [ABGS21] extended this to all p *excluding the even integers* $2\mathbb{Z}$.

Theorem 2.4 (Fine-grained complexity of GapCVP, [BGS17, ABGS21])**.** *For every $p \geq 1$ there is no $2^{o(n)}$-time algorithm for* GapCVP *in the ℓ_p norm assuming ETH.*[7] *Furthermore, for every $p \in [1, \infty] \setminus 2\mathbb{Z}$ and constant $\varepsilon > 0$, there is no $2^{(1-\varepsilon)n}$-time algorithm for* GapCVP *in the ℓ_p norm assuming SETH.*

Additionally, [BGS17, ABGS21] also showed analogous results for γ-GapCVP with a small constant approximation factor $\gamma > 1$ assuming "Gap" versions of ETH and SETH. The $p \notin 2\mathbb{Z}$ caveat is disappointing because the Euclidean case of $p = 2$ is of the most interest, but recent work of Aggarwal and Kumar [AK22] shows that it is very likely inherent at least for a large class of proof techniques ([BGS17, ABGS21] showed why their specific proof techniques did not work for $p \in 2\mathbb{Z}$).

In elegant follow-up work, Aggarwal and Stephens-Davidowitz [AS18] extended the results of [BGS17] for GapCVP by showing fine-grained hardness results for GapSVP. Specifically, they gave very efficient reductions from the hard (approximate) GapCVP instances in [BGS17] and elsewhere to (approximate) GapSVP instances. We note that the results in [AS18] are not as quantitatively strong and do not apply for as wide a range of ℓ_p norms as those of [BGS17]. However, because GapSVP is no harder than GapCVP in a fine-grained sense (see [GMSS99]) and perhaps substantially easier, this discrepancy is natural.

[6] ETH is a stronger assumption than $\mathsf{P} \neq \mathsf{NP}$, which is equivalent to 3-SAT not having a poly(n)-time algorithm, and, as the name implies, SETH is a stronger assumption than ETH, although this is slightly less obvious. There are also stronger variants variants of ETH and SETH (including randomized, non-uniform, and "Gap" variants), many of which also play an important role in the fine-grained hardness of lattice problems.

[7] Although formalized and published for the first time in [BGS17], this result on hardness of GapCVP under ETH was also known in folklore.

Theorem 2.5 (Fine-grained complexity of GapSVP, [AS18]). *For every $p \geq 1$ there exists $\gamma > 1$ such that there is no $2^{o(n)}$-time algorithm for γ-GapSVP in the ℓ_p norm assuming non-uniform Gap-ETH. Furthermore, for every $p > p_0 \approx 2.1397$, $p \notin 2\mathbb{Z}$, there is no $2^{C_p n}$-time algorithm for* GapSVP *in the ℓ_p norm for some constant $C_p \in (0,1)$ assuming randomized SETH.*

Additionally, [BPT22] extended the second result by showing roughly $2^{C_p n}$-hardness of γ-GapSVP in the ℓ_p norm for $p > p_0 \approx 2.1397$, $p \notin 2\mathbb{Z}$ with a small constant approximation factor $\gamma > 1$ assuming randomized Gap-SETH.

Finally, we note that close quantum analogs of the results in Theorems 2.4 and 2.5 hold assuming the quantum analog of SETH. Quantum SETH essentially says that Grover search, which gives a quadratic speed-up over classical brute-force search, is the optimal quantum algorithm for solving k-SAT for large k (see [BPS21] for a formal definition). In particular, in the quantum setting, the roughly 2^n-time lower bound in Theorem 2.4 becomes a roughly $2^{n/2}$-time lower bound, and the $2^{C_p n}$-time lower bound in Theorem 2.5 becomes a $2^{C_p n/2}$-time lower bound.

We also refer the reader to Figure 4, which compares the exponential hardness of GapSVP shown in Theorem 2.5 with exponential-time protocols, reductions, and algorithms for GapSVP.

Open problems on fine-grained hardness. The first natural open question related to fine-grained hardness is whether it is possible to *prove* a lower bound matching the "plausible runtime barrier" of $2^{0.2075n}$ even for quantum SVP algorithms widely assumed in the cryptanalysis literature under some standard complexity assumption.

Open Problem 2.6 (Exponential hardness in ℓ_2). *Prove that there is no $2^{0.2075n}$-time quantum algorithm for exact* SVP *in the ℓ_2 norm under a standard complexity assumption.*

Of course, a natural such starting complexity assumption would be SETH, but [AK22] gives a substantial barrier to showing such hardness. However, it does not fully rule out such reductions, even barring any surprising complexity-theoretic consequences. So, one could try to make progress on Open Problem 2.6 either by trying to mitigate the barrier in [AK22], or by showing hardness under a different assumption. We also note that any progress along the lines of Open Problem 2.6 (i.e., showing 2^{cn}-hardness of SVP for any explicit constant $c > 0$ under a standard assumption) would be very interesting.

Another interesting question already asked in [ABGS21, BGS20] is whether it is possible to prove greater-than-2^n hardness for (exact) GapSVP or GapCVP in some other norm. For example, using the intuition mentioned in [ABGS21, BGS20] that the best running time of an algorithm for SVP in some norm is bounded by the maximum possible number of shortest non-zero vectors in that norm (i.e., the lattice *kissing number* in that norm), one might expect algorithms for SVP in the ℓ_∞ norm to take at least roughly 3^n time.[8] (We note that lattice problems in the ℓ_2 norm are the easiest among all ℓ_p norms in a precise sense [RR06].)

Open Problem 2.7 (Greater-than-2^n hardness in other norms). *Prove that that for some constant $c > 1$, there is no 2^{cn}-time algorithm for exact* GapSVP *or exact* GapCVP *in some norm under a standard complexity assumption.*

3 Showing Hardness: A Technical Summary

In this section, we sketch how to prove (randomized) NP-hardness of γ-GapSVP for some small approximation factor $\gamma > 1$, and then how to amplify this approximation factor to obtain the results in Theorem 2.2. This section is the most technical in the survey; the reader may skip it and proceed to Section 4. Additionally, we emphasize that this sketch omits various subtleties and details in an attempt to convey the main ideas.

[8] The vectors $\{-1, 0, 1\}^n \setminus \{\mathbf{0}\}$ are all shortest non-zero vectors in the integer lattice $\mathbb{Z}^n$ in the ℓ_∞ norm. So, the ℓ_∞ kissing number is at least $3^n - 1$. Moreover, a packing argument shows that the kissing number in any norm is at most 3^n, so this is essentially tight.

3.1 Showing Small-Factor Hardness of Approximation: A First Attempt

In this section, we sketch how to prove NP-hardness of γ-GapSVP for a small constant approximation factor $\gamma > 1$ (at largest $\gamma \approx \sqrt{2}$ for γ-GapSVP in the ℓ_2 norm).

Nearly all of the many different NP-hardness proofs for GapSVP start in roughly the same way. Namely, they reduce from a variant of GapCVP (recall Definition 1.3) called GapCVP$'$, and use gadget lattices called locally dense lattices as advice. We start by defining GapCVP$'$.

Definition 3.1. For $\gamma = \gamma(n) \geq 1$, γ-GapCVP$'$ is the decision problem defined as follows. On input a basis $\mathbf{B} \in \mathbb{Q}^{m \times n}$ of a lattice $\mathcal{L}$, a target vector $\boldsymbol{t} \in \mathbb{Q}^m$, and a distance threshold $r > 0$, decide whether

- (YES instance) There exists $\boldsymbol{x} \in \{0,1\}^n$ such that $\|\mathbf{B}\boldsymbol{x} - \boldsymbol{t}\| \leq r$, or
- (NO instance) $\mathrm{dist}(w\boldsymbol{t}, \mathcal{L}) > \gamma r$ for all $w \in \mathbb{Z} \setminus \{0\}$

when one of these cases is promised to hold.

We note that GapCVP$'$ has stronger promises in both the YES and NO cases than "plain" GapCVP. Namely, in the YES case, the target vector is promised to be close to a *binary* combination of basis vectors, and in the NO case, the target vector is promised to be far from all non-zero integer multiples $w\boldsymbol{t}$ of the target $\boldsymbol{t}$ as opposed to just $\boldsymbol{t}$. Despite these stronger promises, Arora, Babai, Stern, and Sweedyk showed that GapCVP$'$ is NP-hard to approximate to within any constant factor γ.

Theorem 3.2 ([ABSS93])**.** *For any constant $\gamma \geq 1$, γ-GapCVP$'$ is NP-hard.*

A first attempt to reduce an instance $(\mathbf{B}, \boldsymbol{t}, r)$ of γ-GapCVP$'$ to an instance $(\mathbf{B}', r')$ of γ'-GapSVP for some approximation factors $\gamma, \gamma' \geq 1$ works by "appending $-\boldsymbol{t}$ to $\mathbf{B}$." In fact, to ensure linear independence of the columns of $\mathbf{B}'$, we first add some small orthogonal component $s > 0$ to $-\boldsymbol{t}$, and then append the resulting vector to $\mathbf{B}$. Putting everything together, this yields the following attempted reduction:

$$\mathbf{B}, \boldsymbol{t}, r \;\mapsto\; \mathbf{B}' := \begin{pmatrix} \mathbf{B} & -\boldsymbol{t} \\ \mathbf{0} & s \end{pmatrix} \;, r' := \sqrt{r^2 + s^2}\;. \tag{2}$$

This construction of $\mathbf{B}'$ from $\mathbf{B}$, $\boldsymbol{t}$, and s is called *Kannan's embedding*. It is easy to see that the transformation in Equation (2) maps YES instances of γ-GapCVP$'$ to YES instances of γ'-GapSVP by noting that if $\|\mathbf{B}\boldsymbol{x} - \boldsymbol{t}\| \leq r$ for $\boldsymbol{x} \in \{0,1\}^n$ then $\|\mathbf{B}' \cdot (\boldsymbol{x}^T, 1)^T\| \leq r'$. Furthermore, by setting s to be sufficiently small and γ to be sufficiently large, we have that if the input instance of γ-GapCVP$'$ is a NO instance then $\|\mathbf{B}' \cdot (\boldsymbol{x}^T, w)^T\| > \gamma r \approx \gamma r'$ for any $\boldsymbol{x} \in \mathbb{Z}^n$ and $w \in \mathbb{Z} \setminus \{0\}$. So, if we only needed to consider non-zero w, we could take $\gamma' \approx \gamma$ and the reduction would essentially work. However, the trouble with this approach comes from analyzing $\|\mathbf{B}' \cdot (\boldsymbol{x}^T, 0)^T\| = \|\mathbf{B}\boldsymbol{x}\|$. Indeed, if $\mathcal{L}(\mathbf{B})$ happens to contain "overly short" vectors, then so will $\mathcal{L}(\mathbf{B}')$ regardless of whether $\boldsymbol{t}$ is close to $\mathcal{L}(\mathbf{B})$.

3.2 The Ajtai-Micciancio and Khot Reductions

There are two related ways to overcome the issue caused by the lattice $\mathcal{L}(\mathbf{B})$ in the input GapCVP$'$ instance having "overly short" vectors, which we call the "Ajtai-Micciancio reduction" after [Ajt98, Mic98], and the "Khot reduction" after [Kho04]. The key to both of these reductions is the use of gadget lattice-target pairs called *locally dense lattices*, a term that was coined by Micciancio.

Definition 3.3 (Locally Dense Lattices)**.** For constant α satisfying $1/2 \leq \alpha < 1$ and $n', G \in \mathbb{Z}^+$, an (α, G)-*locally dense lattice* is specified by a basis matrix $\mathbf{A} \in \mathbb{Q}^{n' \times n'}$ and a vector $\boldsymbol{u} \in \mathbb{Q}^{n'}$ such that

$$\left|\mathcal{L}(\mathbf{A}) - \boldsymbol{u}) \cap (\alpha\lambda_1(\mathcal{L}(\mathbf{A})) \cdot \mathcal{B}_2^{n'})\right| \geq G\;,$$

where $\mathcal{B}_2^{n'}$ denotes the closed unit Euclidean ball in n' dimensions.

That is, there are at least G vectors in the lattice "shift" $\mathcal{L}(\mathbf{A}) - \boldsymbol{u}$ of norm $\alpha\lambda_1(\mathcal{L}(\mathbf{A}))$, or, equivalently, $\mathcal{L}(\mathbf{A})$ has at least G vectors at distance at most $\alpha\lambda_1(\mathcal{L}(\mathbf{A}))$ to $\boldsymbol{u}$ (intuitively, there are at least G vectors in $\mathcal{L}(\mathbf{A})$ "close" to $\boldsymbol{u}$). We call α the *relative distance* of the locally dense lattice. It is the ratio of the "close distance" $\alpha\lambda_1(\mathcal{L})$ to the minimum distance $\lambda_1(\mathcal{L})$ of the lattice.[9]

We note that nearly all of the many hardness reductions for GapSVP are variants of one of these two reductions and use locally dense lattices. The sole exception is [Kho03], which gives a reduction from the Label Cover Problem rather than GapCVP$'$ to GapSVP.[10]

We describe the reductions below as if they have access to a suitable locally dense lattice already; for now one may think of them as being provided as auxiliary input or advice. In Section 3.4 we discuss (efficient, randomized) constructions of locally dense lattices.

The Ajtai-Micciancio reduction. The Ajtai-Micciancio reduction modifies Equation (2) by "stacking" the basis-target pair $(\mathbf{B}, \boldsymbol{t})$ in the input GapCVP$'$ instance and the basis-target pair $(\mathbf{A}, \boldsymbol{u})$ defining a locally dense lattice, and then applies Kannan's embedding. Additionally, it right-multiplies $\mathbf{B}$ by a linear transformation $\mathbf{T}$ such that all possible coefficient vectors $\boldsymbol{x} \in \{0,1\}^n$ of a close vector $\mathbf{B}\boldsymbol{x}$ to $\boldsymbol{t}$ in the input GapCVP$'$ instance are contained in $\mathbf{T}(\mathbf{Y})$, where $\mathbf{Y} := \{\boldsymbol{y} \in \mathbb{Z}^{n'} : \|\mathbf{A}\boldsymbol{y} - \boldsymbol{u}\| \leq \alpha \cdot \lambda_1(\mathcal{L}(\mathbf{A}))\}$ is the set of coefficient vectors of vectors $\boldsymbol{v} = \mathbf{A}\boldsymbol{y} \in \mathcal{L}(\mathbf{A})$ within distance $\alpha \cdot \lambda_1(\mathcal{L}(\mathbf{A}))$ of $\boldsymbol{u}$. Ajtai and Micciancio show how to sample suitable such linear transformations $\mathbf{T}$ efficiently using randomness [Ajt98, Mic98].

That is, the output basis $\mathbf{B}'$ becomes

$$\mathbf{B}' := \begin{pmatrix} \mathbf{B}\mathbf{T} & -\boldsymbol{t} \\ \mathbf{A} & -\boldsymbol{u} \\ \mathbf{0} & s \end{pmatrix} . \tag{3}$$

One can then check that if the input GapCVP$'$ instance is a YES instance with $\|\mathbf{B}\boldsymbol{x} - \boldsymbol{t}\| \leq r$ for $\boldsymbol{x} \in \{0,1\}^n$, if $T\boldsymbol{y} = \boldsymbol{x}$ for $\boldsymbol{y} \in \mathbb{Z}^{n'}$, and if $\boldsymbol{y}$ is one of the at least G coefficient vectors such that $\|\mathbf{A}\boldsymbol{y} - \boldsymbol{u}\| \leq \alpha \cdot \lambda_1(\mathcal{L}(\mathbf{A}))$, then $\mathbf{B}'(\boldsymbol{y}^T, 1)^T$ is a short non-zero vector in $\mathcal{L}(\mathbf{B}')$, implying that $\lambda_1(\mathcal{L}(\mathbf{B}'))$ is small. On the other hand, one can check that if the GapCVP$'$ instance is a NO instance then $\lambda_1(\mathcal{L}(\mathbf{B}')) \geq \min\{\lambda_1(\mathcal{L}(\mathbf{A})), \min_{w \neq 0} \operatorname{dist}(w\boldsymbol{t}, \mathcal{L}(\mathbf{B})\}$ is large.

The Khot reduction. The Khot reduction first modifies Equation (2) by forming a block-diagonal matrix from $\mathbf{B}$ and $\mathbf{A}$ (which is a basis of the direct sum lattice $\mathcal{L}(\mathbf{B}) \oplus \mathcal{L}(\mathbf{A})$), and then applying Kannan's embedding. That is, the intermediate basis output by the first step in Khot's reduction is

$$\mathbf{B}_{\text{int}} := \begin{pmatrix} \mathbf{B} & \mathbf{0} & -\boldsymbol{t} \\ \mathbf{0} & \mathbf{A} & -\boldsymbol{u} \\ \mathbf{0} & \mathbf{0} & s \end{pmatrix} . \tag{4}$$

Finally, Khot's reduction computes and outputs a basis $\mathbf{B}'$ of a uniformly random sublattice $\mathcal{L}'$ of $\mathcal{L}(\mathbf{B}_{\text{int}})$ of prime index $q \approx G$ (i.e., such that $|\mathcal{L}(\mathbf{B}_{\text{int}})/\mathcal{L}'| = q$). This process is called *sparsification*, and computing $\mathbf{B}'$ is efficient. Intuitively, sparsification has the effect of keeping each point in a finite subset $S \subseteq \mathcal{L}(\mathbf{B}_{\text{int}})$ in $\mathcal{L}'$ with probability $1/q$. We next explain why this is useful.

Let A be the number of vectors in $\mathcal{L}(\mathbf{B})$ of norm at most r', and recall that these vectors are what cause problems in the initial reduction attempted in Equation (2). Let G be the number of "close" vectors in the locally dense lattice defined by $(\mathbf{A}, \boldsymbol{u})$.[11] Then, if $A \ll G$, with high probability $\mathcal{L}'$ will contain no short vectors in the NO case (the only such vectors correspond to short vectors in $\mathcal{L}(\mathbf{B})$ and will be removed by

[9] We note that by the triangle inequality and the definition of $\lambda_1(\mathcal{L})$, there do not exist locally dense lattices with $\alpha < 1/2$ and $G > 1$ simultaneously (even with respect to norms other than ℓ_2). So, we have chosen to define locally dense lattices only for $\alpha \geq 1/2$.

[10] The reduction in [Kho03] came before the better-known reduction in [Kho04], which essentially subsumes its results. Nevertheless, [Kho03] is interesting because it uses different techniques from all other hardness reductions for GapSVP.

[11] Khot introduced the mnemonic of A for the number of "annoying" vectors in the NO case and G for the number of "good" vectors in the YES case in the intermediate lattice $\mathcal{L}(\mathbf{B}_{\text{int}})$.

sparsification), but at least one short vector of the form $\mathbf{B}' \cdot (\boldsymbol{x}^T, \boldsymbol{y}^T, 1)^T$ in the YES case (where $\mathbf{B}\boldsymbol{x}$ is a close vector to $\boldsymbol{t}$ in the input γ-GapCVP$'$ instance and $\mathbf{A}\boldsymbol{y}$ is one of the at least G close vectors in $\mathcal{L}(\mathbf{A})$ to $\boldsymbol{u}$ guaranteed by the definition of a locally dense lattice).

Use of randomness. Both the Ajtai-Micciancio reduction and the Khot reduction use randomness in two places. First, because all efficient constructions of locally dense lattices (in the ℓ_2 norm) are randomized, both reductions use randomness to construct the pair $\mathbf{A}, \boldsymbol{u}$ (in most cases only finding $\boldsymbol{u}$ requires randomness; see also Section 3.4). Additionally, the Ajtai-Micciancio reduction requires randomness to sample the linear transformation $\mathbf{T}$, and the Khot reduction requires randomness for sparsification (which makes its use of randomness seem more inherent). Furthermore, although very elegant, the Khot reduction has two-sided error whereas the Ajtai-Micciancio reduction only has one-sided error (assuming that construction of the basis $\mathbf{A}$ of the locally dense lattice is deterministic).

We also note that deterministic NP-hardness of approximation *is* in fact known for the decision version of the Minimum Distance Problem (GapMDP), which is the analog of GapSVP on linear error correcting codes. Indeed, although the original reduction showing such NP-hardness of approximation for GapMDP by Dumer, Micciancio, and Sudan [DMS99] was randomized, it was successfully derandomized by Cheng and Wan [CW09].[12] Austrin and Khot [AK11] and Micciancio [Mic14] subsequently showed simplified such deterministic reductions.

In fact, [DMS99] used an analog of the Ajtai-Micciancio to show hardness, and [CW09] successfully derandomized it. So, one might optimistically hope that similar techniques to those used to show hardness of approximation for GapMDP also work for showing deterministic hardness of GapSVP. While some techniques do not seem to carry over (this can be made precise in certain ways; see [BP22]), connections between hardness reductions for (approximate) GapMDP and GapSVP certainly warrant further study.

3.3 Amplifying Hardness

The same approach for hardness amplification leads to all three results in Theorem 2.2, namely *tensoring*. The *tensor product* $\mathcal{L}_1 \otimes \mathcal{L}_2$ of two lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ is defined as $\mathcal{L}_1 \otimes \mathcal{L}_2 := \mathcal{L}(\mathbf{B}_1 \otimes \mathbf{B}_2)$, where $\mathbf{B}_1$ and $\mathbf{B}_2$ are bases of $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively, and $\mathbf{B}_1 \otimes \mathbf{B}_2$ is their Kronecker product. It is easy to show that $\lambda_1(\mathcal{L}_1 \otimes \mathcal{L}_2) \leq \lambda_1(\mathcal{L}_1) \cdot \lambda_1(\mathcal{L}_2)$, but in general this inequality may or may not be tight or even close to tight; see, e.g., [HR07, Lemma 2.3].

The approach of [Kho04, HR07, Mic12] for showing the results in Theorem 2.2 uses roughly the following strategy. It starts by reducing γ'-GapCVP$'$ (for some large constant γ') to γ-GapSVP for some relatively small constant approximation factor $\gamma > 1$ using one of the reductions in Section 3.2, and additionally argues that if the GapCVP$'$ instance has certain properties, then the underlying lattice $\mathcal{L}$ in the γ-GapSVP instance tensors nicely. It then outputs the k-fold tensor product $\mathcal{L}^{\otimes k}$ of $\mathcal{L}$ with itself for some $k = k(n)$ (where n is the rank of $\mathcal{L}$), which is a γ^k-GapSVP instance (on a lattice of rank n^k). Varying the value of $k = k(n)$ leads to the three hardness results in Theorem 2.2.

Khot [Kho04] introduced this general strategy, but performed hardness amplification using a variant of the tensor product that he called the "augmented tensor product." This already essentially sufficed to show Theorem 2.2, Item 1, but led to lower factors γ in Items 2 and 3. The key additional insight of Haviv and Regev [HR07] was that the hard lattices output by Khot's reduction in fact behave nicely under the standard tensor product.

Finally, as part of an attempt to derandomize the reductions in [Kho04, HR07], Micciancio [Mic12] introduced a new distance measure $\tau(\boldsymbol{x})$ for integer vectors $\boldsymbol{x}$. While not a norm, the quantity $\tau(\boldsymbol{x})$ lower bounds the ℓ_2 norm $\|\boldsymbol{x}\|$ of $\boldsymbol{x}$ and is meant to represent a combination of the ℓ_2 norm and Hamming weight of $\boldsymbol{x}$. Micciancio then showed that the quantity $\tau(\mathcal{L}) := \min_{\boldsymbol{x} \in \mathcal{L} \setminus \{\mathbf{0}\}} \tau(\boldsymbol{x})$ tensors nicely in the sense that $\tau(\mathcal{L})\lambda_1(\mathcal{L}') \leq \lambda_1(\mathcal{L} \otimes \mathcal{L}') \leq \lambda_1(\mathcal{L})\lambda_1(\mathcal{L}')$ for arbitrary integer lattices $\mathcal{L}, \mathcal{L}'$. Using τ, Micciancio gave

[12] However, in contrast to GapSVP, deterministic NP-hardness of *exact* GapMDP was known since the original hardness proof of Vardy [Var97].

hardness reductions showing the results in Theorem 2.2 with one-side rather than two-sided error.[13] (The use of randomness in all of these reductions occurs in the initial reduction to γ-GapSVP for small γ. Amplifying γ via tensoring is deterministic.)

3.4 Constructing Locally Dense Lattices

Of course, the reductions described in Section 3.2 (and hence Section 3.3) are only useful if locally dense lattices actually exist and are efficient to construct. Indeed, roughly speaking, whether a locally dense lattice exists, is efficiently constructible using randomness, or is efficiently constructible deterministically corresponds to whether the Ajtai-Micciancio reduction is non-uniform, randomized, or deterministic, respectively. Moreover, the specific parameters α and G of the locally dense lattice impact how strong a hardness result the reductions yield. The largest approximation factor γ for which it is possible to show hardness either via the Ajtai-Micciancio or Khot reductions (before any tensoring) is $\gamma \approx 1/\alpha$, and the difference between G being at least 2^{n^ε} for constant $\varepsilon > 0$, $2^{\Omega(n)}$, and 2^{cn} for an explicit constant $c > 0$ roughly corresponds to the difference in being able to show NP-hardness, $2^{\Omega(n)}$-hardness under ETH, and $2^{c'n}$-hardness for an explicit constant $c' > 0$ under SETH (assuming appropriate hardness of GapCVP$'$).

Fortunately, constructions of locally dense lattices are known from a variety of different base lattices, including the Schnorr-Adleman prime number lattice [Ajt98, CN98, Mic98], a variant of Construction A applied to BCH codes [Kho04, HR07], Construction D applied to a tower of BCH codes [Mic12], the integer lattice $\mathbb{Z}^n$ [AS18, BP20, BPT22], exponential kissing number lattices (constructed in [Vlă19], used in [AS18, BPT22]), and Construction A applied to Reed-Solomon codes [BP22]. These various constructions each have different merits, such as being more apparently amenable to derandomization, having smaller α, having larger G, or working in different ℓ_p norms.[14] In particular, for every constant $\varepsilon > 0$, efficient randomized constructions of locally dense lattices of rank n with $\alpha = \sqrt{2} - \varepsilon$ and $G = 2^{n^\delta}$ for some constant $\delta > 0$ are known [Mic98, BP22]. This value of α is also known to be optimal for the ℓ_2 norm (i.e., as small as possible) for subexponentially large $G = G(n)$.

We conclude with two open problems that were originally asked by and would derandomize the constructions in [Mic98] and [BP22], respectively. Solutions to these problems would (nearly) be sufficient to show deterministic NP-hardness of GapSVP as asked for in Open Problem 2.1. We note that these problems—which are about the density of smooth integers and syndromes of parity check matrices of Reed-Solomon codes, respectively—are interesting math problems in their own right and not even clearly related to lattices a priori. The first problem is likely very hard.

Open Problem 3.4 (Determinstic local density from prime number lattices, [Mic98, Conjecture 1])**.** *Show that for any constant $\varepsilon > 0$ there exists a constant $d > 0$ such that for all sufficiently large n there exists an odd integer in the range $[n, n + n^\varepsilon]$ that is square-free and $(\log^d n)$-smooth (i.e., all prime factors of n occur with multiplicity 1 and are at most $\log^d n$).*

For a prime q and integer $1 \le k \le q$, let $H = H(k, q) \in \mathbb{F}_q^{\{0,1,\dots,k-1\} \times \mathbb{F}_q}$ be the Vandermonde matrix with $H_{i,j} = j^i$. That is, the ith row of H, indexed by $i \in \{0, 1, \dots, k-1\}$, consists of the ith powers of the elements $j \in \mathbb{F}_q$. (We define 0^0 to be 1.) The matrix $H(k, q)$ is a parity-matrix of the Reed-Solomon code over $\mathbb{F}_q$ with block length q and codimension k.

Open Problem 3.5 (Deterministic local density from Reed-Solomon lattices, [BP22])**.** *Show that there exists $\varepsilon > 0$ such that for all sufficiently large primes q and $k := \lceil q^\varepsilon \rceil$ there exists an efficiently computable $\boldsymbol{s} \in \mathbb{F}_q^k$ such that subexponentially many vectors $\boldsymbol{x} \in \{0,1\}^q$ of Hamming weight at most $1.99k$ such that $H\boldsymbol{x} = \boldsymbol{s}$, where $H = H(k, q)$.*

[13] Despite their reduction having two-sided error, Haviv and Regev [HR07] argue that, by combining their reduction with self-reducibility properties of SAT, one can turn a fast algorithm for γ-GapSVP into a fast algorithm for SAT with *one-sided error*. Hence, they get the hardness results in Theorem 2.2 under assumptions of the form "$\mathsf{NP} \not\subseteq \mathsf{RTIME}[f(n)]$" as opposed to stronger assumptions of the form "$\mathsf{NP} \not\subseteq \mathsf{BPTIME}[f(n)]$."

[14] Locally dense lattices with respect to general ℓ_p norms are defined analogously to Definition 3.3, but with λ_1 replaced by $\lambda_1^{(p)}$, the lattice's minimum distance in the ℓ_p norm, and with $\mathcal{B}_2^n$ replaced by $\mathcal{B}_p^n$, the closed ℓ_p unit ball.

That is, in coding theory language, Open Problem 3.5 asserts that for some constant $\delta > 0$ there are at least 2^{q^δ} many received words $\boldsymbol{x} \in \{0,1\}^q \subseteq \mathbb{F}_q^q$ of Hamming weight at most $1.99k$ with syndrome $\boldsymbol{s}$.

We note that, assuming for simplicity that $1.99k$ is an integer, an averaging argument shows that there exists some $\boldsymbol{s} \in \mathbb{F}_q^k$ such that at least $\binom{q}{1.99k}/q^k$ vectors $\boldsymbol{x} \in \{0,1\}^q$ of Hamming weight $1.99k$ satisfy $H\boldsymbol{x} = \boldsymbol{s}$, and that

$$\binom{q}{1.99k}/q^k \geq q^{0.99k}/(1.99k)^{1.99k} \approx q^{0.99k}/q^{1.99\varepsilon k} = q^{(0.99-1.99\varepsilon)q^\varepsilon}$$

is subexponentially large in q for all ε satisfying $0 < \varepsilon \leq 0.49$. Furthermore, it is not hard to sample $\boldsymbol{s} \in \mathbb{F}_q^k$ such that with good probability at least, say, $\binom{q}{1.99k}/(100q^k)$ vectors $\boldsymbol{x} \in \{0,1\}^q$ of Hamming weight $1.99k$ satisfy $H\boldsymbol{x} = \boldsymbol{s}$. However, proving that such an $\boldsymbol{s}$ can be found *deterministically* appears to be quite challenging.

We also note that even a "black box" answer to Open Problem 3.5 is not quite enough to show deterministic NP-hardness of GapSVP by itself, the issue being that computing the linear transformation T in the Ajtai-Micciancio reduction still requires randomness. However, if one could additionally show that the vectors $\boldsymbol{x} \in \{0,1\}$ with $H\boldsymbol{x} = \boldsymbol{s}$ are sufficiently structured then this would likely allow for computing T deterministically too. Finally, we note in passing that [BP22] shows that a positive resolution of Open Problem 3.5 would yield list-decoding lower bounds for Reed-Solomon codes, improving on the state-of-the-art in [GR05].

4 Additional Complexity

Finally, we discuss a number of important additional aspects of the complexity of SVP. In Section 4.1, we discuss the relationship between the complexity of γ-GapSVP and the security of lattice-based cryptography, and also describe related results on the structural complexity of γ-GapSVP. In Section 4.2 we discuss variants of SVP and reductions between them. Finally, in Section 4.3 we discuss additional connections between γ-GapSVP and cryptography. (There is also overlap between these topics.)

4.1 GapSVP with Polynomial Approximation Factors

In this section we discuss the complexity of γ-GapSVP with polynomial approximation factors $\gamma = \gamma(n)$, which corresponds to the "Cryptography" and and "Hardness Barriers" boxes in Figure 3.

The two key problems in lattice-based cryptography are the Short Integer Solutions Problem (SIS), from which it is possible to construct many "minicrypt" primitives such as private-key encryption, and the Learning With Errors Problem (LWE), from which it is possible to construct many "cryptomania" primitives such as public-key encryption and much more (such as fully homomorphic encryption). We do not define the SIS or LWE problems here or discuss their properties in detail, and refer the reader to the survey of Peikert [Pei16] for these things. Rather, here we note that a beautiful line of work on worst-case to average-case reductions shows that these problems (and the cryptosystems built from them) are provably hard *on average* (which implies that the cryptosystems built from them are provably secure) assuming that γ-GapSVP with a sufficiently large polynomial approximation factor $\gamma = \gamma(n)$ is hard *in the worst case*.

These hardness reductions are one of the hallmark attractive features of lattice-based cryptography, and from a complexity standpoint give even more motivation for studying approximate GapSVP with a range of approximation factors. Such reductions began with work of Ajtai [Ajt96], which gave a worst-case to average-case reduction from γ-GapSVP to SIS, albeit with a large polynomial approximation factor γ. This was later improved to $\gamma = \tilde{O}(n)$ by Micciancio and Regev [MR04]. Furthermore, work of Regev [Reg05] introduced the LWE problem and gave an efficient quantum reduction from γ-GapSVP to LWE with $\gamma = \tilde{O}(n^{3/2})$. Follow-up work by Peikert [Pei09] and Brakersi, Langlois, Peikert, Regev, and Stehlé [BLP+13] partially "de-quantized" Regev's reduction and gave an efficient *classical* reduction from γ-GapSVP to LWE with $\gamma = \tilde{O}(n^2)$.[15]

[15] The notation $\tilde{O}(f(n))$ is shorthand for $O(f(n) \cdot \mathrm{poly}(\log(f(n))))$.

In fact, the full story is a bit more complicated. Both SIS and LWE are parameterized by a modulus $q = q(n)$, SIS additionally has a size bound $\beta = \beta(n)$, LWE additionally has an error distribution with width parameter $\alpha = \alpha(n)$, and the settings of these parameters determine the smallest value of γ for which worst-case to average-case reductions from γ-GapSVP yield hardness. However, there are meaningful parameter settings of SIS and LWE (in particular, ones that yield provably secure private- and public-key cryptography, respectively) for which it suffices to take $\gamma = \gamma(n)$ to be the factors stated above.

We first ask whether it is possible to show hardness of SIS or LWE assuming that γ-GapSVP is hard for a smaller approximation factor γ (which is a weaker hypothesis).

Open Problem 4.1 (Hardness of SIS or LWE from smaller approximation factors)**.** *Give a classical polynomial-time worst-case to average-case reduction from* γ-GapSVP *to* SIS *with* $\gamma = O(n^{1-\varepsilon})$ *or to* LWE *with* $\gamma = O(n^{2-\varepsilon})$ *for some constant* $\varepsilon > 0$*. Or, give a quantum polynomial-time such reduction to* LWE *with* $\gamma = O(n^{3/2-\varepsilon})$*.*

Barriers to hardness. Given the reductions mentioned above showing that, if γ-GapSVP is hard in the worst-case for sufficiently large polynomial γ, then lattice-based cryptography is provably secure, it is a natural question whether one can in fact prove such hardness. Indeed, these reductions show that if γ-GapSVP is NP-hard for $\gamma = n^{2+\varepsilon}$ for $\varepsilon > 0$, then LWE and hence lattice-based cryptography is secure. This would in turn answer a huge open question in theoretical cryptography by showing how to obtain secure encryption solely from the assumption $\mathsf{P} \neq \mathsf{NP}$.

Unfortunately, it seems very unlikely that a result of this form is true because of two results placing γ-GapSVP for $\gamma \approx \sqrt{n}$ in the complexity classes coAM and coNP. More specifically, work of Goldreich and Goldwasser [GG98] shows that $O(\sqrt{n/\log n})$-GapSVP is in coAM, and work of Aharonov and Regev [AR04] shows that $O(\sqrt{n})$-GapSVP is in coNP. We also note that Peikert [Pei07] showed similar results in different ℓ_p norms. (These results all hold for approximate GapCVP with the same approximation factors as well.)

These are very interesting results in their own right, and contribute to our understanding of the rich structural complexity of GapSVP (as shown in Figure 3). In the context of cryptographic applications, these results are often considered "hardness barriers" in the sense that they (conditionally) rule out NP-hardness of γ-GapSVP for factors $\gamma = \Omega(\sqrt{n})$ including the $\gamma = \Omega(n)$ factors needed to show hardness of SIS or LWE. Indeed, if, say, n-GapSVP were NP-hard, this would imply that $\mathsf{NP} \subseteq \mathsf{coNP}$ and $\mathsf{NP} \subseteq \mathsf{coAM}$, and the polynomial hierarchy would collapse.[16]

We also ask whether it is possible to improve our understanding of the structural complexity of approximate GapSVP and strengthen either of these "barrier" results.

Open Problem 4.2 (Improved coNP or coAM hardness)**.** *Prove that* γ-GapSVP *is in* coNP *for* $\gamma = o(\sqrt{n})$ *or in* coAM *for* $\gamma = o(\sqrt{n/\log n})$*.*

In particular, it would be interesting to improve the $\gamma = O(\sqrt{n})$ approximation factor for which we know that γ-GapSVP is in coNP to $\gamma = O(\sqrt{n/\log n})$, which is the smallest approximation factor for which we that γ-GapSVP is in coAM.

Finally, we note yet another interesting complexity result for approximate SVP with a polynomial approximation factor. Specifically, work of Sotiraki, Zampetakis, and Zirdelis [SZZ18] places n-SVP (and a variant of SIS) in PPP, the complexity class of total search problems that are proven to be total using pigeonhole arguments.

Protocols and reductions in super-polynomial time. Recent work by Aggarwal, Bennett, Brakerski, Golovnev, Kumar, Li, Peters, Stephens-Davidowitz, and Vaikuntanathan [ABB+22] addresses Open Problems 4.1 and 4.2, but with a twist: it considers protocols and worst-case to average-case reductions that run in *super-polynomial time*, and specifically ones that run in $2^{\varepsilon n}$ time for some small constant $\varepsilon > 0$.

[16] There is a subtlety in showing that n-GapSVP being NP-hard implies that $\mathsf{NP} \subseteq \mathsf{coNP}$ and $\mathsf{NP} \subseteq \mathsf{coAM}$ in that γ-GapSVP for $\gamma > 1$ is *promise problem* and not a total problem. This subtlety is addressed in [AR04], which shows that this claim does in fact hold.

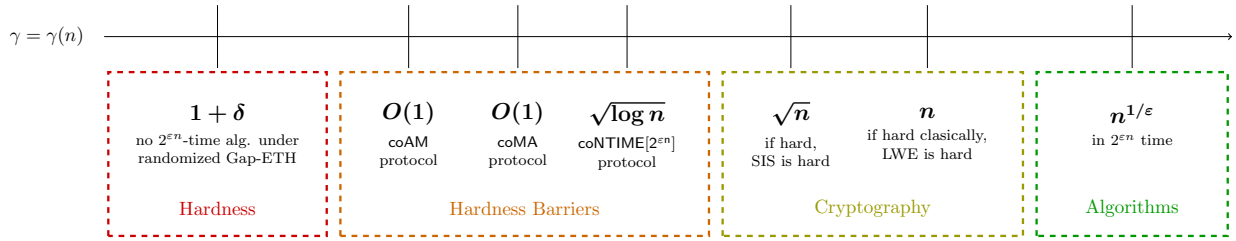

Figure 4: The complexity of γ-GapSVP on lattice of dimension n when algorithms, protocols, and reductions are allowed to run in $2^{\varepsilon n}$ time for small constant $\varepsilon > 0$, with some constants and lower-order terms omitted for clarity. The results corresponding to all of the protocols and reductions in the "Hardness Barriers" and "Cryptography" boxes are new and appear in [ABB+22]. They save either a roughly $\sqrt{n}$ or roughly n factor in the approximation factor compared to the corresponding protocol or reduction run in poly(n) time; see Figure 3 for comparison. The approximation factors in the cryptography box are sufficient to show hardness of SIS and LWE with parameters that imply private-key and public-key cryptography, respectively.

As is the case for Figure 3, increasing the approximation factor γ in the "Hardness" box, or decreasing any of the approximation factors γ in the "Hardness Barriers," "Cryptography," or "Algorithms" boxes while keeping the corresponding assumption or consequence the same is an interesting open problem. This figure appears in [ABB+22].

In particular, [ABB+22] shows that running in $2^{\varepsilon n}$ time allows for saving a factor of roughly $\sqrt{n}$ in the approximation factor $\gamma = \gamma(n)$ in the coAM and co-non-deterministic protocols for γ-GapSVP mentioned above, in the (classical) worst-case to average-case reduction from γ-GapSVP to SIS, and in the quantum worst-case to average-case reduction from γ-GapSVP to LWE. Additionally, [ABB+22] shows that such a $2^{\varepsilon n}$ running time allows for saving a factor of roughly n in the approximation factor γ in the classical worst-case to average-case reduction from γ-GapSVP to LWE. Finally, it gives a novel coMA protocol for γ-GapSVP that only improves on the corresponding co-non-deterministic protocol when run in nearly exponential time (and saves a factor of roughly n in the approximation factor compared to its own polynomial-time version when run in this way).

See Figure 4 for a summary of the results obtained in [ABB+22] as well as algorithms and hardness results for γ-GapSVP in the $2^{\varepsilon n}$-time regime. (The best $2^{\varepsilon n}$-time algorithms for γ-GapSVP use basis reduction [GN08, MW16, ALNS20] and work for approximation factors $\gamma \approx n^{1/\varepsilon}$. The best-known (conditional) hardness of approximation result in the $2^{\varepsilon n}$-time regime is the one from [AS18] mentioned in Theorem 2.5.) Additionally, see Figure 3 for a comparison with approximation factors for protocols, reductions, and algorithms in the polynomial-time regime.

Analogous to Open Problems 4.1 and 4.2, it is interesting to ask whether any of the approximation factors γ in Figure 4 for γ-GapSVP in the $2^{\varepsilon n}$-time regime can be improved. We specifically ask this question for the protocols and worst-case to average-case reductions in Figure 4, since all of the corresponding results are new.

Open Problem 4.3 (Improved protocols and reductions in $2^{\varepsilon n}$)**.** *Lower one of the approximation factors γ for the $2^{\varepsilon n}$-time protocols (given in the "Hardness Barriers" box) or worst-case to average-case reductions (given in the "Cryptography" box) in Figure 4.*

Finally, motivated by the coMA protocol given in [ABB+22], it is interesting to consider whether there is a polynomial-time coMA protocol for γ-GapSVP that achieves a better approximation factor γ than the polynomial-time Aharonov-Regev coNP protocol, which achieves an approximation factor of $\gamma = \sqrt{n}$. (We again note that this is not the case for the coMA protocol in [ABB+22].)

Open Problem 4.4 (A coMA protocol)**.** *Give a polynomial-time* coMA *protocol for γ-GapSVP on lattices of dimension n with $\gamma = \gamma(n) = o(\sqrt{n})$.*

4.2 Variants of SVP and Reductions Between Them

Search-to-decision reductions. One natural question is whether it is possible to give a dimension-preserving reduction from γ'-SVP to γ-GapSVP that preserves the approximation factor γ as much as possible. The best-known such reductions appear in work of Cheng, Hu and Pan, and Stephens-Davidowitz [Che13, HP13, Ste16]. In particular, [Ste16] gives a reduction that, for $\gamma \leq 1 + O(\log n/n)$, runs in polynomial time and has $\gamma' = \gamma^{n/\log n} \leq O(1)$. However, one might hope for a much better reduction, e.g., one that preserves both the dimension n and approximation factor γ' of the input search instance exactly (or with γ depending polynomially on n and γ').

Open Problem 4.5 (Search-to-decision reductions)**.** *Give a reduction from* γ'-SVP *on lattices of dimension* n *to* γ-GapSVP *on lattices of dimension* n *for* $\gamma' = \mathrm{poly}(\gamma, n)$.

Using the reductions from γ-GapSVP for $\gamma = \mathrm{poly}(n)$ mentioned in Section 4.1, Open Problem 4.5 would in particular show security of lattice-based cryptography based on the worst-case hardness of $\mathrm{poly}(n)$-approximate *search* SVP. We note that trying to find better search-to-decision reductions for lattice problems as asked for by Open Problem 4.5 is a well-known open problem. In particular, Open Problem 4.5 is similar to [Sim22, Problems 6 and 7].

Unique SVP**.** Another variant of SVP is the γ-*approximate unique Shortest Vector Problem* (γ-uSVP) is the variant of search SVP on lattices $\mathcal{L}$ such that, if $\boldsymbol{v} \in \mathcal{L}$ satisfies $\|\boldsymbol{v}\| = \lambda_1(\mathcal{L})$, then all vectors in $\mathcal{L}$ linearly independent of $\boldsymbol{v}$ have norm greater than $\gamma \cdot \lambda_1(\mathcal{L})$. As for γ-GapSVP, γ-uSVP with sufficiently large polynomial $\gamma = \gamma(n)$ is used as a worst-case problem on which to base hardness of lattice-based cryptography.

Despite being a search problem, γ-uSVP is actually *easier* than γ-GapSVP in the sense that there is an efficient, dimension-preserving reduction from γ-uSVP to γ-GapSVP [LM09]. One can also show that $(2-\varepsilon)$-uSVP in the ℓ_∞ norm is NP-hard for any $\varepsilon > 0$. However, relatively little is known about the hardness of γ-uSVP in the ℓ_2 norm. The exact version ($\gamma = 1$) was shown to be NP-hard by Kumar and Sivakumar [KS01], and this was improved slightly to $\gamma = 1 + 1/\mathrm{poly}(n)$ by Aggarwal and Dubey [AD16] (see also [Ste16] for an alternate hardness proof with $\gamma = 1 + 1/\mathrm{poly}(n)$). Finally, we note that Bennett, Ganju, Peetathawatchai, and Stephens-Davidowitz [BGPS21] recently showed hardness of γ-uSVP for all constant $\gamma \geq 1$ (and even superconstant $\gamma = \gamma(n)$) but under non-standard assumptions. (All of these hardness results for uSVP are under randomized reductions.) However, γ-uSVP is still not known to be NP-hard even for a fixed constant $\gamma > 1$ (say, $\gamma = 1.01$).

Open Problem 4.6 (Hardness of uSVP)**.** *Prove that* γ-uSVP *in the* ℓ_2 *norm is* NP*-hard (under randomized reductions) for some constant* $\gamma > 1$.

Parameterized GapSVP**.** We conclude this section by discussing the *parameterized* complexity of GapSVP. In the parameterized version of γ-GapSVP in the ℓ_p norm, the input consists of an integer basis $\mathbf{B}$ and a distance parameter $r > 0$ such that $k := r^p$ is a positive integer. The task is the same as for normal GapSVP, i.e., to decide whether the ℓ_p minimum distance of $\mathcal{L}(\mathbf{B})$ is at most r or greater than γr. However, the runtimes of algorithms for parameterized GapSVP are written as functions not only of n but of $k = r^p$.

A *fixed-parameter tractable* (FPT) algorithm for a problem with parameter k is one running in time $O(f(k) \cdot \mathrm{poly}(n))$ for some (possibly fast-growing) function $f(k)$ not depending on the input size n. The set of problems with FPT algorithms forms the complexity class FPT, which is typically considered to be the class of "tractable" parameterized problems. On the other hand, the canonical complexity class of "intractable" parameterized problems is called W[1]-*hard*. A parameterized problem is W[1]-hard if there is an FPT reduction from every problem in W[1] to it, and a problem is in W[1] if it is reducible via an FPT reduction to the k-Clique problem on graphs. (The k-Clique problem is the canonical W[1]-*complete* problem—it is both in W[1] and W[1]-hard.) It is a major open question whether FPT = W[1], which is the analog of the P versus NP question in the parameterized world. See [DF99, DF13] for detailed surveys about parameterized complexity theory.

It was a well-known open question whether GapSVP in the ℓ_2 norm was W[1]-hard (with respect to the parameter $k = r^p$). Indeed, it was one of the few remaining open questions in Downey and Fellows's classic book [DF99]. However, recent work by Bhattacharyya, Bonnet, Egri, Ghoshal, Karthik C. S., Lin, Manurangsi, and Marx [BBE+21] resolved this question in the affirmative and even showed hardness of approximation for parameterized GapSVP. Specifically, [BBE+21] showed that γ-GapSVP in the ℓ_p norm is W[1]-hard (under randomized reductions) for all $p > 1$ and *some* approximation factor $\gamma = \gamma(p)$ with $1 < \gamma < 2$ for all such p. This was in turn strengthened in even more recent work by Bennett, Cheraghchi, Guruswami, and Ribeiro [BCGR22], which showed W[1]-hardness of γ-GapSVP in the ℓ_p norm for all $p > 1$ and *all* constant approximation factors $\gamma \geq 1$, as well as W[1]-hardness of γ-GapSVP in the ℓ_1 norm for any constant $\gamma < 2$, again under randomized reductions. Indeed, up to its use of randomness, [BCGR22] provided a nearly complete picture of the parameterized complexity of GapSVP in ℓ_p norms, but still left open the question of showing inapproximability of GapSVP in the ℓ_1 norm to within arbitrary constant factors.

Open Problem 4.7 (Parameterized inapproximability in the ℓ_1 norm)**.** *Prove that for every constant $\gamma \geq 1$, parameterized γ-GapSVP in the ℓ_1 norm is W[1]-hard (possibly under randomized reductions).*

While many of the other open problems in this survey are likely very hard, Open Problem 4.7 seems to be very much within reach; the techniques used in [BCGR22] fell just short of proving it.

Finally, we note that [BCGR22], building on work of Manurangsi [Man20], proved that parameterized γ-GapSVP in the ℓ_p norm on lattices of rank n requires $n^{\Omega(k)}$ time to solve for $\gamma = \gamma(p) < 2^{1/p}$ under the randomized Gap-ETH assumption.

4.3 Complexity for Cryptography

Hardness on Algebraically Structured Lattices. We next discuss *algebraically structured lattices*, an important class of lattices for cryptography in that they often make cryptosystems more time- and space-efficient. Such algebraically structured lattices include ideal and module lattices, which appear in cryptosystems based on the Ring-SIS, Ring-LWE, and Module-LWE problems [Mic02, LPR10, LS15], and in the NTRU cryptosystem [HPS98]. Notably, The CRYSTALS cryptosystems [ABD+17] recently standardized by NIST are based on Module-LWE. Additionally, for sufficiently large γ there are worst-case to average-case reductions from γ-SVP on ideal lattices to Ring-SIS and Ring-LWE, and from γ-SVP on module lattices to Module-LWE [Mic02, LM06, PR06, SSTX09, LPR10, LS15, PRS17]. (This is analogous to the situation for "plain" LWE, but there the reduction is from γ-GapSVP on arbitrary lattices.) Furthermore, attacks on the cryptosystems built from these problems are typically based on algorithms for solving exact or near exact SVP on algebraically structured lattices. (The main potential downside of using algebraically structured lattices for cryptography is that it might be possible to exploit the added structure for cryptanalysis.)

Consider a ring of the form $R = \mathbb{Z}[x]/(p(x))$ for an irreducible polynomial $p(x)$ of degree n.[17] Here n is the *degree* of R. The elements of R are polynomials $q(x) = \sum_{i=0}^{n-1} a_i x^i$ of degree at most $n-1$ with integer coefficients $a_0, \ldots, a_{n-1}$. An *ideal* $\mathcal{I} \subseteq R$ is an additive subgroup of R that is closed under multiplication by elements of R. One can then construct a lattice from an ideal $\mathcal{I} \subseteq R$ via the *coefficient embedding*, which simply maps an element $q(x) = \sum_{i=0}^{n-1} a_i x^i \in \mathcal{I}$ to its coefficient vector $(a_0, \ldots, a_{n-1})^T \in \mathbb{Z}^n$.[18] A lattice constructed from an ideal of a ring R of this form via the coefficient embedding (or another embedding) is called an *ideal lattice*. Note that applying the coefficient embedding to an ideal of degree n yields a lattice of dimension n (the degree of an ideal is inherited from its base ring R). In particular, this implies that computational problems on ideals of degree n (viewed as ideal lattices) are no harder than such problems on general lattices of dimension n.

[17] More generally, one can take R to be (any subring of) the ring of integers $\mathcal{O}_K$ of an algebraic number field K.

[18] One may also construct a lattice from an ideal using the *canonical embedding*, which maps $q(x)$ to the vector $(q(\alpha_1), \ldots, q(\alpha_n))^T \in \mathbb{C}^n$, where $\alpha_1, \ldots, \alpha_n$ are the n complex roots of $p(x)$. We have defined lattices to be real-valued, but one may easily define lattices in complex space as well. In particular, one may extend the definition of the Euclidean norm to complex space; the norm of $\boldsymbol{x} \in \mathbb{C}^n$ is $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^* \boldsymbol{x}} = (\sum_{i=1}^n \overline{x_i} x_i)^{1/2}$. The canonical embedding has some particularly nice properties such as being able to compute the embedding of the product $q_1(x)q_2(x)$ for $q_1(x), q_2(x) \in R$ by computing the coordinate-wise product of the embeddings of $q_1(x)$ and $q_2(x)$.

We start by asking the natural question of whether (exact) GapSVP is NP-hard on ideal lattices corresponding to some natural family of base rings, e.g., power-of-2 cyclotomics—polynomial rings of the form $\mathbb{Z}[x]/(x^n+1)$ with n a power of 2—which are frequently used in cryptography.

Open Problem 4.8 (Hardness on Ideal Lattices)**.** *Prove that* GapSVP *is* NP*-hard on ideal lattices corresponding to some family of base rings (e.g., power-of-*2 *cyclotomics).*

One may also ask about hardness of approximation of GapSVP on ideal lattices. However, unlike for general lattices, γ-GapSVP on ideal lattices is easy for $\gamma > \sqrt{n}$. Also unlike for general lattices, worst-case to average-case reductions reduce from approximate *search* SVP on ideal lattices. So, the easiness of γ-GapSVP for relatively small γ on ideal lattices does not preclude getting meaningful cryptographic hardness from worst-case problems on ideal lattices. (See, e.g., [Ste18] for a discussion of these issues.)

One may also consider (Gap)SVP on a broader class of algebraically structured lattices called *module lattices*. Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m \in R^\ell$ for some $\ell \in \mathbb{Z}^+$. The module they generate over R is

$$\mathcal{M} := \left\{ \sum_{i=1}^{m} a_i \boldsymbol{y}_i : a_1, \ldots, a_m \in R \right\},$$

and one can apply the coefficient embedding (or another embedding) coordinate-wise to vectors in $\mathcal{M}$ to get an (ℓn)-dimensional lattice, which is called a module lattice. Another key parameter of $\mathcal{M}$ is its *rank* k, which is the dimension of its span as a vector space over $\mathbb{Q}[x]/(p(x))$. Note that $k \leq \ell$. When $k = \ell = 1$, $\mathcal{M}$ is simply an ideal of R, and so module lattices can be seen as a natural generalization of ideal lattices by considering larger k. Moreover, SVP on module lattices with $k > 1$ is potentially much harder than when $k = 1$. Indeed, there is some evidence for this since various highly non-trivial algorithms have been found for (approximate) SVP on ideal lattices, but not on module lattices [CGS14, CDPR16, CDW17, DPW19]. See [LS15, AD17, MS20] for more detailed summaries.

So, it is natural to start by asking for a relaxed version of Open Problem 4.8 for module lattices of low rank k.

Open Problem 4.9 (Hardness on Module Lattices)**.** *Prove that* GapSVP *is* NP*-hard on module lattices of constant rank $k \in \mathbb{Z}^+$ over some family of base rings (e.g., power-of-2 cyclotomics).*

Finally, we note that one can ask essentially all of the other same questions about solving (Gap)SVP on ideal and module lattices that we have for general lattices. For example, can we lower bound the runtime of the fastest algorithm for GapSVP on such lattices?

Number-theoretic cryptography versus lattice-based cryptography. The factoring problem and the discrete logarithm problem are likely the two most important number-theoretic problems used in cryptography. Indeed, they underlie the RSA cryptosystem and Diffie-Hellman key exchange, respectively. In the form of the factoring problem underlying RSA, the goal is to factor a number N of the form $N = pq$ for large primes $p \neq q$ into p and q. In the discrete logarithm problem, the goal is, given a generator g of a cyclic group $G = \langle g \rangle$ of order n and an element $g^a \in G$ for some $a \in \mathbb{Z}_n$ as input, to find a. A natural question is whether these problems can be related to lattice problems by reducing them to γ-GapSVP for sufficiently large γ.

Open Problem 4.10 (Number-theoretic cryptography versus lattice-based cryptography)**.** *Give an efficient classical reduction from factoring or the discrete logarithm problem to* γ-GapSVP *for polynomially large $\gamma = \gamma(n)$. In particular, show such a reduction to* γ-GapSVP *with $\gamma = n^{2+\varepsilon}$ for some constant $\varepsilon > 0$ (which then also gives a classical reduction to* LWE*).*

Intuitively, showing a reduction of the form in Open Problem 4.10 would prove that lattice-based cryptography is at least as secure as number-theoretic cryptography even against classical adversaries. Indeed, because the RSA and Diffie-Hellman problems efficiently reduce to the factoring and discrete logarithm problems, respectively, and because $n^{2+\varepsilon}$-GapSVP efficiently classically reduces to LWE, giving a reduction of

the form asked for in Open Problem 4.10 would imply an efficient reduction from the RSA problem or Diffie-Hellman problem to LWE. (We note that *quantum* reductions from factoring and the discrete logarithm problem of the form asked about in Open Problem 4.10 trivially exist because of Shor's algorithm [Sho94].)

Early research on prime number lattices (which eventually led to the original NP-hardness proofs for GapSVP!) happened in large part due to attempts at showing a reduction from factoring to lattice problems. Recently, Schnorr [Sch21a, Sch21b] made additional attempts to show results roughly along these lines—in fact, with the goal of then solving the output lattice problem quickly, and hence *breaking* RSA—but they were not successful.

Hardness of approximation relevant for cryptography. Next, we state a very natural problem: showing that there is no efficient algorithm for γ-GapSVP with γ large enough that it reduces to LWE under a standard assumption. This would imply that there is no efficient algorithm for LWE, and hence give some provable security for much of lattice-based cryptography.

Open Problem 4.11 (Super-polynomial hardness of LWE)**.** *Prove that there is no (possibly randomized or quantum) polynomial-time algorithm for* $n^{3/2+\varepsilon}$-GapSVP *for some constant* $\varepsilon > 0$ *(or for* LWE*) under a standard complexity assumption.*

We note that Open Problem 4.11 is in some sense a duplicate of the hardness of approximation result in Open Problem 2.3, but we view the regimes (which ask for hardness of γ-GapSVP with $\gamma = n^{\varepsilon}$ and $\gamma = n^{3/2+\varepsilon}$, respectively) as quite different. Indeed, as mentioned in the discussion around Open Problem 2.3, results showing hardness of γ-GapCVP and of γ-GapSVP in the ℓ_∞ norm for $\gamma = n^{\varepsilon}$ under a plausible hardness assumption already exist. More importantly, these and other hypothetical hardness proofs for these problems with $\gamma = n^{\varepsilon}$ for small $\varepsilon > 0$ do not need to bypass the "hardness barrier" results discussed in Section 4.1. Additionally, Open Problem 4.11 explicitly asks to rule out *quantum* algorithms and not just classical algorithms.

Finally, we conclude with what we believe to be the "holy grail" problem of lattice complexity, which asks to show exponential fine-grained hardness of GapSVP as in Open Problem 2.6 and hardness of approximation of GapSVP as in Open Problem 4.11 *simultaneously* and *unconditionally*. In fact, because [ABB+22] gives a $2^{\varepsilon n}$-time worst-case to average-case reduction from $\tilde{O}(n)$-GapSVP to LWE with parameters that allow for constructing public-key cryptography (see Figure 4), and, because we are asking for an exponential runtime lower bound on γ-GapSVP, it suffices to show hardness with $\gamma = n^{1+\varepsilon}$ (rather than $\gamma = n^{3/2+\varepsilon}$ as in Open Problem 4.11).

Open Problem 4.12 (The holy grail)**.** *Prove unconditionally that there is no (possibly randomized or quantum)* 2^{cn}*-time algorithm for* $n^{1+\varepsilon}$-GapSVP *for some explicit constants* $c, \varepsilon > 0$*.*

Resolving Open Problem 4.12 would imply strong, unconditional security of LWE with cryptographically relevant parameters, and would therefore prove strong security of lattice-based cryptography. However, resolving Open Problem 4.12 would also mean proving a statement much stronger than $\mathsf{P} \neq \mathsf{NP}$, and so it is unlikely to happen anytime in the near future. But, it's good to have dreams.

References

[ABB+22] D. Aggarwal, H. Bennett, Z. Brakerski, A. Golovnev, R. Kumar, Z. Li, S. Peters, N. Stephens-Davidowitz, and V. Vaikuntanathan. Lattice problems beyond polynomial time, 2022. Preprint. 4, 13, 14, 18

[ABD+17] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. Cryptographic suite for algebraic lattices (CRYSTALS). https://pq-crystals.org/, 2017. 1, 16

[ABGS21] D. Aggarwal, H. Bennett, A. Golovnev, and N. Stephens-Davidowitz. Fine-grained hardness of CVP(P)—everything that we can prove (and nothing else). In *SODA*. 2021. 6, 7

[ABSS93] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *J. Comput. Syst. Sci.*, 54(2):317–331, 1997. Preliminary version in FOCS 1993. 8

[ACD+18] M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, F. Virdia, and T. Wunderer. Estimate all the {LWE, NTRU} schemes! In *SCN*. 2018. 5

[ACKS21] D. Aggarwal, Y. Chen, R. Kumar, and Y. Shen. Improved (provable) algorithms for the Shortest Vector Problem via bounded distance decoding. In *STACS*. 2021. 6

[AD16] D. Aggarwal and C. K. Dubey. Improved hardness results for unique Shortest Vector Problem. *Inf. Process. Lett.*, 116(10):631–637, 2016. 15

[AD17] M. R. Albrecht and A. Deo. Large modulus Ring-LWE $\geq$ Module-LWE. In *ASIACRYPT*. 2017. 17

[ADPS16] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange - A new hope. In *USENIX*. 2016. 5, 6

[ADRS15] D. Aggarwal, D. Dadush, O. Regev, and N. Stephens-Davidowitz. Solving the Shortest Vector Problem in 2^n time using discrete Gaussian sampling: Extended abstract. In *STOC*. 2015. 6

[Ajt96] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*. 1996. 12

[Ajt98] M. Ajtai. The Shortest Vector Problem in L_2 is NP-hard for randomized reductions (extended abstract). In *STOC*, pages 10–19. 1998. 4, 8, 9, 11

[AK11] P. Austrin and S. Khot. A simple deterministic reduction for the gap minimum distance of code problem. *IEEE Trans. Inf. Theory*, 2014. Preliminary verison in ICALP 2011. 10

[AK22] D. Aggarwal and R. Kumar. Why we couldn't prove SETH hardness of CVP for even norms!, 2022. Available at https://arxiv.org/abs/2211.04385. 6, 7

[ALNS20] D. Aggarwal, J. Li, P. Q. Nguyen, and N. Stephens-Davidowitz. Slide reduction, revisited—filling the gaps in SVP approximation. In *CRYPTO*. 2020. 14

[AR04] D. Aharonov and O. Regev. Lattice problems in $\mathsf{NP} \cap \mathsf{coNP}$. *J. ACM*, 52(5):749–765, 2005. Preliminary version in FOCS 2004. 13

[AS18] D. Aggarwal and N. Stephens-Davidowitz. (Gap/S)ETH hardness of SVP. In *STOC*. 2018. 6, 7, 11, 14

[BBE+21] A. Bhattacharyya, E. Bonnet, L. Egri, S. Ghoshal, K. C. S., B. Lin, P. Manurangsi, and D. Marx. Parameterized intractability of even set and shortest vector problem. *J. ACM*, 68(3), 2021. 16

[BCGR22] H. Bennett, M. Cheraghchi, V. Guruswami, and J. Ribeiro. Parameterized inapproximability of the Minimum Distance Problem over all fields and the Shortest Vector Problem in all ℓ_p norms, 2022. Preprint. 16

[BDGL16] A. Becker, L. Ducas, N. Gama, and T. Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In *SODA*. 2016. 5

[BGPS21] H. Bennett, A. Ganju, P. Peetathawatchai, and N. Stephens-Davidowitz. Just how hard are rotations of $\mathbb{Z}^n$? Algorithms and cryptography with the simplest lattice. *IACR Cryptol. ePrint Arch.*, page 1548, 2021. 15

[BGS17] H. Bennett, A. Golovnev, and N. Stephens-Davidowitz. On the quantitative hardness of CVP. In *FOCS*. 2017. 6

[BGS20] H. Bennett, A. Golovnev, and N. Stephens-Davidowitz. Fine-grained hardness of lattice problems: Open questions, 2020. Available at https://blog.simons.berkeley.edu/2020/05/fine-grained-hardness-of-lattice-problems-open-questions/. 1, 5, 6, 7

[BLP+13] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *STOC*. 2013. 12

[Bon99] D. Boneh. Twenty years of attacks on the RSA cryptosystem. *Not. of the Am. Math. Soc.*, 46(2):203–213, 1999. 1

[BP20] H. Bennett and C. Peikert. Hardness of bounded distance decoding on lattices in ℓ_p norms. In *CCC*. 2020. 11

[BP22] H. Bennett and C. Peikert. Hardness of the (approximate) Shortest Vector Problem: A simple proof via Reed-Solomon codes. *CoRR*, abs/2202.07736, 2022. 10, 11, 12

[BPS21] H. Buhrman, S. Patro, and F. Speelman. A framework of quantum strong exponential-time hypotheses. In M. Bläser and B. Monmege, editors, *STACS*. 2021. 7

[BPT22] H. Bennett, C. Peikert, and Y. Tang. Improved hardness of BDD and SVP under Gap-(S)ETH. In *ITCS*. 2022. 7, 11

[CDPR16] R. Cramer, L. Ducas, C. Peikert, and O. Regev. Recovering short generators of principal ideals in cyclotomic rings. In *EUROCRYPT*. 2016. 17

[CDW17] R. Cramer, L. Ducas, and B. Wesolowski. Short Stickelberger class relations and application to Ideal-SVP. In *EUROCRYPT*. 2017. 17

[CGS14] P. Campbell, M. Groves, and D. Shepherd. Soliloquy: A cautionary tale. ETSI/IQC 2nd Quantum-Safe Crypto Workshop, 2014. 17

[Che13] K. Cheng. Some complexity results and bit unpredictable for short vector problem. Cryptology ePrint Archive, Paper 2013/052, 2013. https://eprint.iacr.org/2013/052. 15

[CN98] J. Cai and A. Nerurkar. Approximating the SVP to within a factor $(1 + 1/\dim^{\varepsilon})$ is NP-hard under randomized reductions. *J. Comput. Syst. Sci.*, 59(2):221–239, 1999. Preliminary version in CCC 1998. 5, 11

[Cop01] D. Coppersmith. Finding small solutions to small degree polynomials. In *Cryptography and Lattices, International Conference*. 2001. 1

[CS99] J. Conway and N. J. A. Sloane. *Sphere packings, lattices, and groups*. Springer, 1999. 1

[CW09] Q. Cheng and D. Wan. A deterministic reduction for the gap minimum distance problem. *IEEE Trans. Inf. Theory*, 58(11):6935–6941, 2012. Preliminary version in STOC 2009. 10

[Dad12] D. Dadush. *Integer Programming, Lattice Algorithms, and Deterministic Volume Estimation*. Ph.D. thesis, Georgia Institute of Technology, 2012. 1

[DF99] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999. 15, 16

[DF13] R. G. Downey and M. Fellows. *Fundamentals of Parameterized Complexity*. Texts in computer science. Springer, 2013 edition, 2013. 15

[Din02] I. Dinur. Approximating SVP_∞ to within almost-polynomial factors is NP-hard. *Theor. Comput. Sci.*, 285(1):55–71, 2002. 5

[DMS99] I. Dumer, D. Micciancio, and M. Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Trans. Inf. Theory*, 49(1):22–37, 2003. Preliminary version in FOCS 1999. 10

[DPW19] L. Ducas, M. Plançon, and B. Wesolowski. On the shortness of vectors to be found by the Ideal-SVP quantum algorithm. In *CRYPTO*. 2019. 17

[GG98] O. Goldreich and S. Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000. Preliminary version in STOC 1998. 13

[GMSS99] O. Goldreich, D. Micciancio, S. Safra, and J. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Inf. Process. Lett.*, 71(2):55–61, 1999. 6

[GN08] N. Gama and P. Q. Nguyen. Finding short lattice vectors within Mordell's inequality. In *STOC*. 2008. 4, 14

[GR05] V. Guruswami and A. Rudra. Limits to list decoding Reed-Solomon codes. *IEEE Trans. Inf. Theory*, 52(8):3642–3649, 2006. Preliminary version in STOC 2005. 12

[HP13] G. Hu and Y. Pan. Improvements on reductions among different variants of SVP and CVP. In *International Workshop on Information Security Applications*. 2013. 15

[HPS98] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*. 1998. 16

[HR07] I. Haviv and O. Regev. Tensor-based hardness of the Shortest Vector Problem to within almost polynomial factors. *Theory Comput.*, 8(1):513–531, 2012. Preliminary version in STOC 2007. 5, 10, 11

[Kan87] R. Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987. 1

[Kho03] S. Khot. Hardness of approximating the Shortest Vector Problem in high ℓ_p norms. *J. Comput. Syst. Sci.*, 72(2):206–219, 2006. Preliminary version in FOCS 2003. 5, 9

[Kho04] S. Khot. Hardness of approximating the Shortest Vector Problem in lattices. *J. ACM*, 52(5):789–808, 2005. Preliminary version in FOCS 2004. 5, 8, 9, 10, 11

[KS01] R. Kumar and D. Sivakumar. On the unique shortest lattice vector problem. *Theor. Comput. Sci.*, 255(1-2):641–648, 2001. 15

[Laa15] T. Laarhoven. Sieving for shortest vectors in lattices using angular locality-sensitive hashing. In *CRYPTO*. 2015. 5

[Len83] H. W. Lenstra. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983. 1

[LLL82] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, December 1982. 4

[LM06] V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In *ICALP*. 2006. 16

[LM09] V. Lyubashevsky and D. Micciancio. On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In *CRYPTO*. 2009. 15

[LMvdP15] T. Laarhoven, M. Mosca, and J. van de Pol. Finding shortest lattice vectors faster using quantum search. *Des. Codes Cryptogr.*, 77(2-3):375–400, 2015. 5

[LO85] J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *J. ACM*, 32(1):229–246, 1985. 1

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. *J. ACM*, 60(6):43:1–43:35, 2013. Preliminary version in EUROCRYPT 2010. 16

[LS15] A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3):565–599, 2015. 16, 17

[Man20] P. Manurangsi. Tight running time lower bounds for strong inapproximability of maximum k-coverage, unique set cover and related problems (via t-wise agreement testing theorem). In *SODA*. 2020. 16

[MG02] D. Micciancio and S. Goldwasser. *Complexity of lattice problems a cryptographic perspective*. Springer, 2002. 1

[Mic98] D. Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. *SIAM J. Comput.*, 30(6):2008–2035, 2000. Preliminary version in FOCS 1998. 5, 8, 9, 11

[Mic02] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Comput. Complex.*, 16(4):365–411, 2007. Preliminary version in FOCS 2002. 16

[Mic12] D. Micciancio. Inapproximability of the Shortest Vector Problem: Toward a deterministic reduction. *Theory Comput.*, 8(1):487–512, 2012. 5, 10, 11

[Mic14] D. Micciancio. Locally dense codes. In *CCC*. 2014. 10

[Mos12] D. Moshkovitz. The projection games conjecture and the NP-hardness of $\ln n$-approximating set-cover. *Theory Comput.*, 11:221–235, 2015. Preliminary version in APPROX 2012. 5

[MR04] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. Preliminary version in FOCS 2004. 12

[MS20] T. Mukherjee and N. Stephens-Davidowitz. Lattice reduction for modules, or how to reduce modulesvp to modulesvp. In *CRYPTO*. 2020. 17

[Muk22] P. Mukhopadhyay. The Projection Games Conjecture and the hardness of approximation of super-SAT and related problems. *J. Comput. Syst. Sci.*, 123:186–201, 2022. 5

[MW16] D. Micciancio and M. Walter. Practical, predictable lattice basis reduction. In *EUROCRYPT*. 2016. 4, 14

[Nat22] National Institute of Standards and Technology. Post-quantum cryptography project. https://csrc.nist.gov/projects/post-quantum-cryptography/, 2022. 1

[Pei07] C. Peikert. Limits on the hardness of lattice problems in ℓ_p norms. *Computational Complexity*, 17(2):300–351, May 2008. Preliminary version in CCC 2007. 13

[Pei09] C. Peikert. Public-key cryptosystems from the worst-case Shortest Vector Problem: extended abstract. In *STOC*. 2009. 12

[Pei16] C. Peikert. A decade of lattice cryptography. *Found. Trends Theor. Comput. Sci.*, 10(4):283–424, 2016. 1, 12

[PR06] C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *TCC*. 2006. 16

[PRS17] C. Peikert, O. Regev, and N. Stephens-Davidowitz. Pseudorandomness of Ring-LWE for any ring and modulus. In *STOC*. 2017. 16

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009. Preliminary version in STOC 2005. 12

[RR06] O. Regev and R. Rosen. Lattice problems and norm embeddings. In *STOC*. 2006. 5, 7

[Sch87] C. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987. 4

[Sch21a] C. Schnorr. Fast factoring integers by SVP algorithms. *IACR Cryptol. ePrint Arch.*, page 232, 2021. 18

[Sch21b] C. Schnorr. Fast factoring integers by SVP algorithms, corrected. *IACR Cryptol. ePrint Arch.*, page 933, 2021. 18

[Sho94] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. Preliminary version in ANTS 1994. 18

[Sim22] Simons Institute Summer Cluster: Lattices and Beyond. Top 10 problems on algorithms and complexity aspects of lattices, 2022. Simons Institute Wiki Post. Available at https://wiki.simons.berkeley.edu/doku.php?id=lat22:start. 1, 15

[SSTX09] D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In *ASIACRYPT*, pages 617–635. 2009. 16

[Ste16] N. Stephens-Davidowitz. Search-to-decision reductions for lattice problems with approximation factors (slightly) greater than one. In *APPROX*. 2016. 15

[Ste18] N. Stephens-Davidowitz. Ring SIS and ideal lattices, 2018. Lecture notes. 17

[SZZ18] K. Sotiraki, M. Zampetakis, and G. Zirdelis. PPP-completeness with connections to cryptography. In *FOCS*. 2018. 13

[Var97] A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Trans. Inf. Theory*, 43(6):1757–1766, 1997. 10

[vEB81] P. van Emde Boas. Another NP-complete partition problem and the complexity of computing short vectors in a lattice. Technical Report, 1981. Available at https://staff.fnwi.uva.nl/p.vanemdeboas/vectors/mi8104c.html. 3, 4

[Vlă19] S. Vlăduţ. Lattices with exponentially large kissing numbers. *Mosc. J. Comb. Number Theory*, 8(2):163–177, 2019. 11

[Wal17] M. Walter. *On the Concrete Security of Lattice-Based Cryptography*. Ph.D. thesis, University of California, San Diego, 2017. 5

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il