Lattice Problems beyond Polynomial Time

Divesh Aggarwal
National University of Singapore
Singapore
divesh@comp.nus.edu.sg

Huck Bennett
Oregon State University
Corvallis, Oregon, USA
huck.bennett@oregonstate.edu

Zvika Brakerski[*]
Weizmann Institute of Science
Rehovot, Israel
zvika.brakerski@weizmann.ac.il

Alexander Golovnev
Georgetown University
Washington, DC, USA
alexgolovnev@gmail.com

Rajendra Kumar
Weizmann Institute of Science
Rehovot, Israel
rjndr2503@gmail.com

Zeyong Li[†]
National University of Singapore
Singapore
li.zeyong@u.nus.edu

Spencer Peters[‡]
Cornell University
Ithaca, NY, USA
sp2473@cornell.edu

Noah Stephens-Davidowitz[‡]
Cornell University
Ithaca, NY, USA
noahsd@gmail.com

Vinod Vaikuntanathan[§]
MIT
Cambridge, MA, USA
vinodv@csail.mit.edu

ABSTRACT

We study the complexity of lattice problems in a world where algorithms, reductions, and protocols can run in superpolynomial time. Specifically, we revisit four foundational results in this context—two protocols and two worst-case to average-case reductions. We show how to improve the approximation factor in each result by a factor of roughly $\sqrt{n/\log n}$ when running the protocol or reduction in $2^{\varepsilon n}$ time instead of polynomial time, and we show a novel protocol with no polynomial-time analog. Our results are as follows.

(1) We show a worst-case to average-case reduction proving that secret-key cryptography (specifically, collision-resistant hash functions) exists if the (decision version of the) Shortest Vector Problem (SVP) cannot be approximated to within a factor of $\widetilde{O}(\sqrt{n})$ in $2^{\varepsilon n}$ time. This extends to our setting Ajtai's celebrated polynomial-time reduction for the Short Integer Solutions (SIS) problem (1996), which showed (after improvements by Micciancio and Regev (2004, 2007)) that secret-key cryptography exists if SVP cannot be approximated to within a factor of $\widetilde{O}(n)$ in polynomial time.

(2) We show another worst-case to average-case reduction proving that *public-key* cryptography exists if SVP cannot be approximated to within a factor of $\widetilde{O}(n)$ in $2^{\varepsilon n}$ time. This extends Regev's celebrated polynomial-time reduction for the Learning with Errors (LWE) problem (2005, 2009), which achieved an approximation factor of $\widetilde{O}(n^{1.5})$. In fact, Regev's reduction is quantum, but we prove our result under a classical reduction, generalizing Peikert's polynomial-time classical reduction (2009), which achieved an approximation factor of $\widetilde{O}(n^2)$.

(3) We show that the (decision version of the) Closest Vector Problem (CVP) with a constant approximation factor has a coAM protocol with a $2^{\varepsilon n}$-time verifier. We prove this via a (very simple) generalization of the celebrated polynomial-time protocol due to Goldreich and Goldwasser (1998, 2000). It follows that the recent series of $2^{\varepsilon n}$-time and even $2^{(1-\varepsilon)n}$-time hardness results for CVP cannot be extended to large constant approximation factors γ unless AMETH is false. We also rule out $2^{(1-\varepsilon)n}$-time lower bounds for any constant approximation factor $\gamma > \sqrt{2}$, under plausible complexity-theoretic assumptions. (These results also extend to arbitrary norms, with different constants.)

(4) We show that $O(\sqrt{\log n})$-approximate SVP has a coNTIME protocol with a $2^{\varepsilon n}$-time verifier. Here, the analogous (also celebrated!) polynomial-time result is due to Aharonov and Regev (2005), who showed a polynomial-time protocol achieving an approximation factor of $\sqrt{n}$ (for *both* SVP and CVP, while we only achieve this result for SVP). This result implies similar barriers to hardness, with a larger approximation factor under a weaker complexity-theoretic conjectures (as does the next result).

(5) Finally, we give a novel coMA protocol for constant-factor-approximate CVP with a $2^{\varepsilon n}$-time verifier. Unlike our other results, this protocol has no known analog in the polynomial-time regime.

All of the results described above are special cases of more general theorems that achieve time-approximation factor tradeoffs. In

[*] Supported by the Israel Science Foundation (Grant No. 3426/21), and by the European Union Horizon 2020 Research and Innovation Program via ERC Project REACT (Grant 756482).

[†] Supported by the NUS-NCS Joint Laboratory for Cyber Security, Singapore.

[‡] Supported in part by the NSF under Grant No. CCF-2122230.

[§] Supported by DARPA under Agreement No. HR00112020023, NSF CNS-2154149, MIT-IBM Watson AI, Analog Devices, a Microsoft Trustworthy AI grant and a Thornton Family Faculty Research Innovation Fellowship. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA.

STOC '23, June 20–23, 2023, Orlando, FL, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9913-5/23/06…$15.00
https://doi.org/10.1145/3564246.3585227

particular, the tradeoffs for the first four results smoothly interpolate from the polynomial-time results in prior work to our new results in the exponential-time world.

CCS CONCEPTS

• **Security and privacy → Mathematical foundations of cryptography**.

KEYWORDS

Lattice problems, worst-case to average-case reductions, shortest vector problem, closest vector problem

ACM Reference Format:
Divesh Aggarwal, Huck Bennett, Zvika Brakerski, Alexander Golovnev, Rajendra Kumar, Zeyong Li, Spencer Peters, Noah Stephens-Davidowitz, and Vinod Vaikuntanathan. 2023. Lattice Problems beyond Polynomial Time. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3564246.3585227

1 EXTENDED ABSTRACT

A lattice $\mathcal{L} \subset \mathbb{R}^n$ is the set of all integer linear combinations of linearly independent basis vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \mathbb{R}^n$,

$$\mathcal{L} = \mathcal{L}(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) = \{z_1\boldsymbol{b}_1 + \cdots + z_n\boldsymbol{b}_n \ : \ z_i \in \mathbb{Z}\} \ .$$

The most important computational problem associated with lattices is the γ-approximate Shortest Vector Problem (γ-SVP), which is parameterized by an approximation factor $\gamma \geq 1$. Given a basis for a lattice $\mathcal{L} \subset \mathbb{R}^n$, γ-SVP asks us to approximate the length of the shortest non-zero vector in the lattice up to a factor of γ. The second most important problem is the γ-approximate Closest Vector Problem (γ-CVP), in which we are additionally given a target point $\boldsymbol{t} \in \mathbb{Q}^n$, and the goal is to approximate the minimal distance between $\boldsymbol{t}$ and any lattice point, again up to a factor of γ. Here, we define length and distance in terms of the ℓ_2 norm (though, in the sequel, we sometimes work with arbitrary norms).

We note that these problems are often referred to as γ-GapSVP and γ-GapCVP, when one wishes to distinguish them from the associated search problems. In this paper, we are only interested in the decision problems and we will therefore refer to these problems simply as γ-SVP and γ-CVP, as is common in the complexity literature.

These two problems are closely related. In particular, CVP is known to be at least as hard as SVP in quite a strong sense, as there is a simple efficient reduction [23] from SVP to CVP that preserves the approximation factor γ and rank n (as well as the norm). Moreover, historically, it has been much easier to find algorithms for SVP than for CVP and much easier to prove hardness results for CVP.

Both SVP and CVP have garnered much attention over the past twenty-five years or so, after Ajtai proved two tantalizing results. First, he constructed a cryptographic (collision-resistant) hash function and proved that it is secure if γ-SVP is hard for some approximation factor $\gamma = \mathrm{poly}(n)$ [7, 22]. This in particular implies that secret-key encryption exists under this assumption. To prove his result, he showed the first worst-case to average-case reduction in this context. Specifically, he showed that a certain *average-case* lattice problem called the Short Integer Solutions problem (SIS, corresponding to the problem of breaking his hash function) was as hard as γ-SVP, a *worst-case* problem. Second, Ajtai proved the NP-hardness of exact SVP, i.e., γ-SVP with $\gamma = 1$ (under a randomized reduction) [8], answering a long-standing open question posed by van Emde Boas [38].

Ajtai's two breakthrough papers led to *many* follow-ups. In particular, there followed a sequence of works showing the hardness of γ-SVP for progressively larger approximation factors γ [17, 24, 25, 28, 29], leading to the current state of the art: NP-hardness (under randomized reductions) for any constant γ and hardness for $\gamma = n^{c/\log\log n}$ under the assumption that NP does not have $2^{n^{o(1)}}$-time (randomized) algorithms. A different, but related, line of work showed hardness of γ-CVP for progressively larger approximation factors γ, culminating in NP-hardness for $\gamma = n^{c/\log\log n}$ [18].

A separate line of work improved upon Ajtai's worst-case to average-case reduction. Micciancio and Regev showed that Ajtai's hash function is secure if $\widetilde{O}(n)$-SVP is hard [31], improving on Ajtai's large polynomial approximation factor. Regev also improved on Ajtai's results in another (very exciting!) direction, showing a *public-key* encryption scheme that is secure under the assumption that $\widetilde{O}(n^{1.5})$-SVP is hard for a *quantum* computer [36]. To do so, Regev defined an average-case lattice problem called Learning with Errors (LWE), constructed a public-key encryption scheme whose security is (essentially) equivalent to the hardness of LWE, and showed a *quantum* worst-case to average-case reduction for LWE. Peikert later showed how to prove *classical* hardness of LWE in a different parameter regime, showing that secure public-key encryption exists if $\widetilde{O}(n^2)$-SVP is hard, even for a *classical* computer [33]. (The ideas in these works have since been extended to design *many* new and exciting cryptographic primitives. See [34] for a survey.)

One might even hope that continued work in this area would lead to one of the holy grails of cryptography: a cryptographic construction whose security can be based on the (minimal) assumption that NP $\not\subseteq$ BPP. Indeed, in order to do so, one would simply need to decrease the approximation factor achieved by one of these worst-case to average-case reductions and increase the approximation factor achieved by the hardness results until they meet! However, two seminal works showed that this was unlikely. First, Goldreich and Goldwasser showed a coAM protocol for $\sqrt{n/\log n}$-CVP, and therefore also for $\sqrt{n/\log n}$-SVP [21]. Second, Aharonov and Regev showed a coNP protocol for $\sqrt{n}$-CVP (and therefore also for $\sqrt{n}$-SVP) [6]. These results are commonly interpreted as barriers to proving hardness, since they imply that if $\sqrt{n/\log n}$-SVP (or even $\sqrt{n/\log n}$-CVP) is NP-hard, then the polynomial hierarchy would collapse to $\mathrm{NP}^{\mathrm{NP}}$, and that the hierarchy would collapse to NP for $\gamma = \sqrt{n}$. It seems very unlikely that we will be able to build cryptography from the assumption that γ-SVP is hard for some $\gamma = o(\sqrt{n})$, and so these results are typically interpreted as ruling out achieving such a "holy grail" result via this approach.

Indeed, the state of the art has been stagnant for over a decade now (in spite of much effort), in the sense that no improvement has been made to the approximation factors achieved by (1) (Micciancio and Regev's improvement to) Ajtai's worst-case to average-case reduction; (2) Regev's worst-case to average-case quantum reduction for public-key encryption or Peikert's classical reduction; (3) the best known hardness results for SVP (or CVP); (4) Goldreich and Goldwasser's coAM protocol; *or* (5) Aharonov and Regev's coNP protocol. (Of course, much progress has been made in other directions!)

However, all of the above results operate in the polynomial-time regime, showing hardness against polynomial-time algorithms and protocols that run in polynomial time (formally, protocols with polynomially bounded communication and polynomial-time verifiers). It is of course conventional (and convenient) to work in this polynomial-time setting, but as our understanding of computational lattice problems and lattice-based cryptography has improved over the past decade, the distinction between polynomial and superpolynomial time has begun to seem less relevant. Indeed, the fastest algorithms for γ-SVP run in time that is *exponential* in n, even for $\gamma = \mathrm{poly}(n)$, and it is widely believed that no $2^{o(n)}$-time algorithm is possible for $\gamma = \mathrm{poly}(n)$. This belief plays a key role in the study of lattice-based cryptography.

In particular, descendants of Regev's original public-key encryption scheme are nearing widespread use in practice. One such scheme was even recently standardized by NIST [9, 32], with the goal of using this scheme as a replacement for the number-theoretic cryptography that is currently used for nearly all secure communication.[1] In practice, these schemes rely for their security not only on the polynomial-time hardness of SVP, but on *very* precise assumptions about the hardness of γ-SVP as a function of γ. (E.g., the authors of [9] rely on sophisticated simulators that attempt to predict the optimal behavior of heuristic γ-SVP algorithms, which roughly tell us that n^k-SVP cannot be solved in time much better than $2^{0.29n/(2k+1)}$ for constant $k \geq 0$.)

Therefore, we are now more interested in the *fine-grained, superpolynomial* complexity of γ-SVP and γ-CVP. I.e., we are not just interested in what is possible in polynomial time, but rather we are interested in precisely what is possible with different superpolynomial running times, with a particular emphasis on algorithms that run in 2^{Cn} time for different constants C. And, the specific approximation factor really matters quite a bit, as the running time $2^{C_\gamma n}$ of the best known γ-SVP algorithms for polynomial approximation factors $\gamma = \mathrm{poly}(n)$ depends quite a bit on the specific polynomial γ. (This is true both for heuristic algorithms and those with proven correctness. E.g., the best known proven running time for approximation factor n^c is roughly $2^{O(n/(c+1))}$ for constant $c \geq 0$. See [4] for the current state of the art.)

Indeed, a recent line of work has extended *some* of the seminal polynomial-time results described above to the fine-grained superpolynomial setting [2, 3, 5, 13, 14]. Specifically, these works show exponential-time lower bounds for SVP and CVP, both in their exact versions with $\gamma = 1$ and for small constant approximation factors $\gamma = 1 + \varepsilon$ (under suitable variants of the Exponential Time Hypothesis). These results can be viewed as fine-grained generalizations of Ajtai's original hardness result for SVP (or, perhaps, of the subsequent results that showed hardness for small approximation factors, such as [17, 28]), and they provide theoretical evidence in favor of the important cryptographic assumption that (suitable) lattice-based cryptography cannot be broken in $2^{o(n)}$ time.

However, there are no known non-trivial generalizations of the other major results listed above to the regime of superpolynomial running times. For example, (in spite of much effort) it is not known how to extend the above fine-grained hardness results to show exponential-time lower bounds for approximation factors γ substantially larger than one—say, e.g., large constants γ (let alone the polynomial approximation factors that are relevant to cryptography)—in analogy with the celebrated hardness of approximation results that are known against polynomial-time algorithms. And, prior to this work, it was also not known how to extend the worst-case to average-case reductions and protocols mentioned above to the superpolynomial setting in a non-trivial way (i.e., in a way that improves upon the approximation factor).

1.1 Our Results

At a high level, our results can be stated quite succinctly. We generalize to the superpolynomial setting (1) Ajtai's worst-case to average-case reduction for secret-key cryptography; (2) Regev's worst-case to average-case quantum reduction for public-key cryptography and Peikert's classical version; (3) Goldreich and Goldwasser's coAM protocol; and (4) Aharonov and Regev's coNP protocol. In all of these results, in the important special case when the reductions or protocols are allowed to run in $2^{\varepsilon n}$ time, we improve upon the polynomial-time approximation factor by a factor of roughly $\sqrt{n/\log n}$ (and a factor of $\widetilde{O}(n)$ for Peikert's classical worst-case to average-case reduction). We also show a novel coMA protocol that has no known analog in the polynomial-time regime.

See Figure 1 for a diagram showing the current state of the art for both the polynomial-time regime and the $2^{\varepsilon n}$-time regime for arbitrarily small constants $\varepsilon > 0$. Below, we describe the results in more detail and explain their significance. We describe the protocols first, as our worst-case to average-case reductions are best viewed in the context of our protocols.

1.1.1 Protocols for Lattice Problems.

A coAM *protocol.* Our first main result is a generalization of Goldreich and Goldwasser's coAM protocol, as follows.

Theorem 1.1 (Informal, see the full version [1, Section 3])**.** *For every $\gamma = \gamma(n) \geq 1$, there is a* coAM *protocol for γ-*CVP *running in time $2^{O(n/\gamma^2)}$.*

*Furthermore, for every constant $\varepsilon > 0$, there exists a $\delta > 0$ such that there is a two-round private-coin (honest-verifier perfect zero knowledge) protocol for $(\sqrt{2} + \varepsilon)$-*coCVP *running in time $2^{(1/2-\delta)n}$.*

See the full version [1, Section 3] for the precise result, which is also more general in that it also applies to arbitrary norms $\|\cdot\|_K$ (with different constants), just like the original theorem of [21].

This theorem is a strict generalization of the original polynomial-time result of Goldreich and Goldwasser [21]. In fact, the protocol itself is a simple generalization of the original [21] protocol. And, just like [21] was viewed as a barrier to proving polynomial-time hardness results for approximation factors $\gamma \geq \sqrt{n/\log n}$, our result can be viewed as a barrier to proving superpolynomial hardness for smaller approximation factors γ. In particular, the theorem rules out the possibility of using a fine-grained reduction from k-SAT to prove, e.g., $2^{\Omega(n)}$ hardness for large constants γ or $2^{(1-\varepsilon)n}$-time

[1] The number-theoretic cryptography that is currently in use is known to be broken by a sufficiently large quantum computer. In contrast, lattice-based cryptography is thought to be secure not only against classical computers, but also against quantum computers, which is why it has been standardized. See [32] for more discussion.

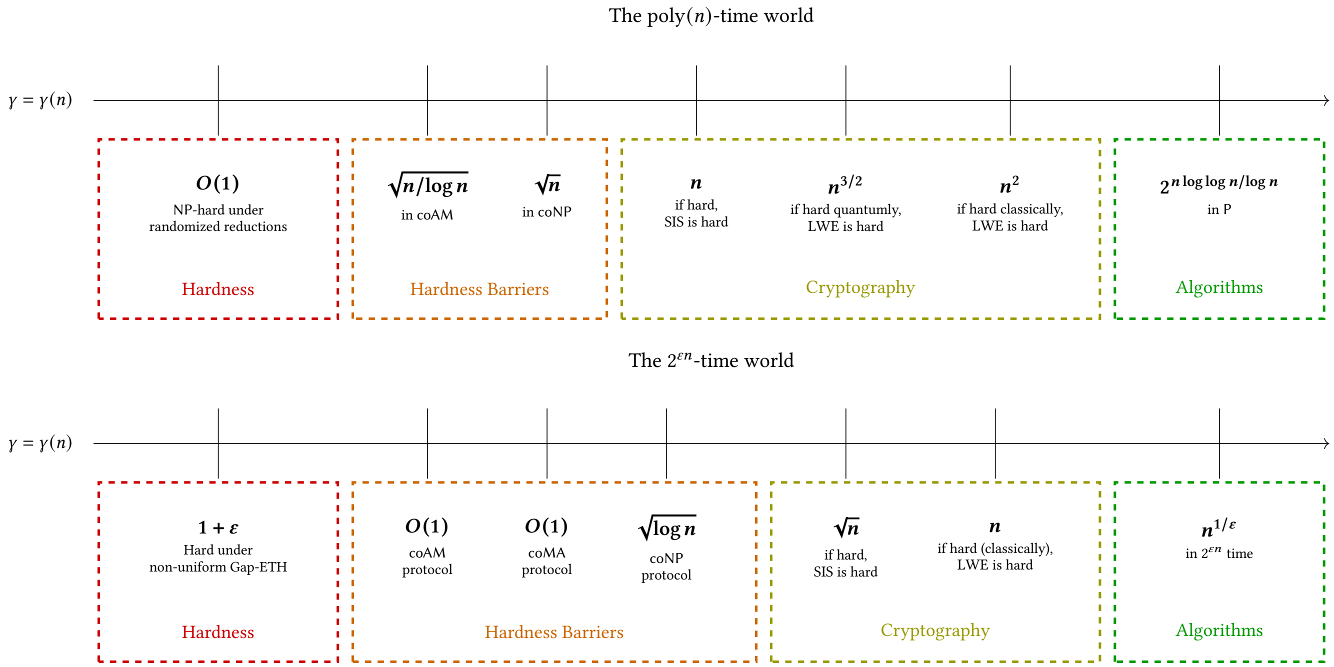


Figure 1: This figure shows the current state of the art of the complexity of γ-SVP for different approximation factors γ in two different regimes. The top row shows polynomial-time results (polynomial-time hardness, protocols, worst-case to average-case reductions, and algorithms, respectively). The bottom row shows $2^{\varepsilon n}$-time results. Note that the scales are rather extreme, and are certainly not the same in the two rows. The hardness barriers and cryptography results in the bottom row are the five new results in this paper. We have omitted some constants for simplicity. (This figure is based on a similar one appearing in [12].)

hardness for any constant $\gamma > \sqrt{2}$ (assuming AMETH and IPSETH respectively, See the full version [1, Section 9] for the precise statement).[2] We place a particular emphasis on the running time of $2^{(1-\varepsilon)n}$ because (1) the fastest known algorithm for CVP runs in time $2^{n+o(n)}$; and (2) we know a $2^{(1-\varepsilon)n}$-time lower bound for $(1+\varepsilon')$-CVP [2, 13] (under variants of SETH—though, admittedly, only in ℓ_p norms where p is not an even integer, so not for the ℓ_2 norm). Therefore, this protocol provides an explanation for why fine-grained hardness results for CVP are stuck at small constant approximation factors. See the full version [1, Section 9] for a precise discussion of these barriers to proving hardness and their relationship to known hardness results.

As we explain in more detail in Section 1.2.1, our protocol is a very simple and natural generalization of the original beautiful protocol due to Goldreich and Goldwasser. And, as we explain below, the same simple ideas behind this protocol are also used in our worst-case to average-case reduction for LWE.

A co-non-deterministic protocol. Our second main result is a variant of Aharonov and Regev's coNP protocol for $\sqrt{n}$-CVP, as follows.

Theorem 1.2 (Informal, see the full version [1, Theorem 6.2])**.** *For every $\gamma = \gamma(n) \geq 1$, there is a co-non-deterministic protocol for γ-SVP that runs in time $n^{O(n/\gamma^2)}$. In particular, there is a $2^{\varepsilon n}$-time protocol for $O_\varepsilon(\sqrt{\log n})$-SVP.*

This result is almost a strict generalization of [6], except that Aharonov and Regev's protocol works for CVP, while ours only works for SVP.

Again, this result can be viewed as a barrier to proving hardness of γ-SVP (assuming NETH; see the full version [1, Section 2.8, Section 9] for more discussions). And, just like how [6] gives a stronger barrier against proving polynomial-time hardness than [21] (collapse of the polynomial hierarchy to the first level, as opposed to the second) at the expense of a larger approximation factor γ, our Theorem 1.2 gives a stronger barrier against proving superpolynomial hardness (formally, a barrier assuming NETH rather than AMETH) than Theorem 1.1, at the expense of a larger approximation factor.

As we discuss more in Section 1.2.2, our protocol is broadly similar to the original protocol in [6], but the details and the analysis are quite different—requiring in particular careful control over the higher moments of the discrete Gaussian distribution.

[2] It might seem strange that we describe a roughly $2^{n/2}$-time protocol as ruling out roughly 2^n hardness. This is because k-coSAT is known to have a roughly $2^{n/2}$-time two-round *protocol* [39] (and even an MA protocol), but is not known to have a $2^{(1-\varepsilon)n}$-time *algorithm* (for sufficiently large k). The assumption that k-SAT has no $2^{(1-\varepsilon)n}$-time protocol for sufficiently large k is called SETH, while the assumption that k-coSAT does not have a $2^{(1/2-\varepsilon)n}$-time two-round protocol for sufficiently large k is called IPSETH. So, to prove $2^{(1-\varepsilon)n}$-time hardness of γ-CVP under SETH, it would suffice to give a ($2^{\varepsilon n}$-time, Turing) reduction from k-SAT on n variables to γ-CVP on a lattice with rank $n + o(n)$. But, for constant $\gamma > \sqrt{2}$, such a reduction together with Theorem 1.1 would imply a significantly faster protocol for k-coSAT than what is currently known, and would therefore violate IPSETH. See the full version [1, Section 2.8] for more discussion of fine-grained complexity and related hypotheses and [1, Section 9] for formal proofs ruling out such reductions under various hypotheses.

We note that we originally arrived at this protocol in an attempt to solve a different (and rather maddening) open problem. In [6], Aharonov and Regev speculated that their protocol could be improved to achieve an approximation factor of $\sqrt{n/\log n}$ rather than $\sqrt{n}$, therefore matching in coNP the approximation factor achieved by [21] in coAM. And, there is a certain sense in which they came tantalizingly close to achieving this (as we explain in Section 1.2.2). It has therefore been a long-standing open problem to close this $\sqrt{\log n}$ gap.

We have *not* successfully closed this gap between [6] and [21]. Indeed, for all running times, the approximation factor in Theorem 1.2 remains stubbornly larger than that in Theorem 1.1 by a factor of $\sqrt{\log n}$, so that in some sense the gap persists even into the superpolynomial-time regime! But, we *do* show that a suitable modification of the Aharonov and Regev coNP protocol can achieve approximation factors less than $\sqrt{n}$, at the expense of more running time. This in itself is already quite surprising, as the analysis in [6] seems in some sense tailor-made for the approximation factor $\sqrt{n}$ and no lower. For example, prior to our work, it was not even clear how to achieve an approximation factor of, say, $\sqrt{n}/10$ in co-non-deterministic time less than it takes to simply solve the problem deterministically. We show how to achieve, e.g., an approximation factor of $\sqrt{n}/C$ for any constant C in polynomial time.

A coMA *protocol.* Our third main result is a coMA protocol for CVP, as follows.

Theorem 1.3 (Informal; see the full version [1, Theorem 7.2]). *There is a* coMA *protocol for γ-*CVP *that runs in time $2^{O(n/\gamma)}$. In particular, there is a $2^{\varepsilon n}$-time protocol for $O_\varepsilon(1)$-*CVP.

Unlike our other protocols, the protocol in Theorem 1.3 has no analog in prior work. Indeed, the result is only truly interesting for running times larger than roughly $2^{\sqrt{n}}$, since for smaller running times it is completely subsumed by [6]. It is therefore unsurprising that this result was not discovered by prior work that focused on the polynomial-time regime.

This protocol too can be viewed as partial progress towards improving the approximation factor achieved by [6] by a factor of $\sqrt{\log n}$. In particular, notice that in the important special case of $2^{\varepsilon n}$ running time, the approximation factor achieved in Theorem 1.3 is better than that achieved by Theorem 1.2 by a $\sqrt{\log n}$ factor. (Indeed, since the approximation factor is constant in this case, it is essentially the best that we can hope for.) So, in the $2^{\varepsilon n}$-time world, there is no significant gap between the approximation factors that we know how to achieve in coMA and coAM, in contrast to the polynomial-time world.

As a barrier to proving exponential-time hardness of lattice problems, the coMA protocol in Theorem 1.3 lies between the co-non-deterministic protocol in Theorem 1.2 and the coAM protocol in Theorem 1.1, since a co-non-deterministic protocol implies a coMA protocol, which implies a coAM protocol (though at the expense of a constant factor in the exponent of the running time; see the full version [1, Section 2.8]). In particular, for $2^{\varepsilon n}$ running time, the approximation factor is (significantly) better than Theorem 1.2 but (just slightly) worse than Theorem 1.1. But, the complexity-theoretic assumption needed to rule out hardness in this case (MAETH) is weaker than for Theorem 1.1 (AMETH) but stronger than for Theorem 1.2 (NETH).

In fact, our coMA protocol is perhaps best viewed as a "mixture" of the two beautiful protocols from [21] and [6]. As we explain in Section 1.2.3, we think of this coMA protocol as taking the best parts from [21] and [6], and we therefore view the resulting "hybrid" protocol as quite natural and elegant.

1.1.2 Worst-Case to Average-Case Reductions.

Worst-case to average-case reductions for SIS*.* Our fourth main result is a generalization beyond polynomial time of (Micciancio and Regev's version of) Ajtai's worst-case to average-case reduction, as follows.

Theorem 1.4 (Informal; see the full version [1, Theorem 8.1]). *For any $\gamma = \gamma(n) \geq 1$, there is a reduction from γ-*SVP *to* SIS *that runs in time $2^{n^2 \cdot \mathrm{polylog}(n)/\gamma^2}$. In particular, (exponentially secure) secret-key cryptography exists if $\widetilde{O}(\sqrt{n})$-*SVP *is $2^{\Omega(n)}$ hard.*

This is a strict generalization of the previous state of the art, i.e., the main result in [31], which only worked in the polynomial-time regime, i.e., for $\gamma = \widetilde{\Theta}(n)$. (In fact, our reduction is also a generalization of the reduction due to Micciancio and Peikert [30], which itself generalizes [31] to more parameter regimes. Specifically, our result holds in the "small modulus" regime, like that of [30]. But, in this high-level description where we have not even defined the modulus, we ignore this important distinction. [31] also gives a polynomial-time reduction from SIVP to SIS, but our techniques do not seem to show a way to achieve a better approximation factor for SIVP via a superpolynomial-time reduction.)

We are particularly interested in the special case of our reduction for $\gamma = \widetilde{\Theta}(\sqrt{n})$. Indeed, as we mentioned earlier, it is widely believed that γ-SVP is $2^{\Omega(n)}$ hard for *any* approximation factor $\gamma \leq \mathrm{poly}(n)$, and even stronger assumptions are commonly made in the literature on lattice-based cryptography (both in theoretical and practical work—and even in work outside of lattice-based cryptography [16]). Therefore, we view the assumption that $\widetilde{O}(\sqrt{n})$-SVP is $2^{\Omega(n)}$ hard to be quite reasonable in this context. Indeed, if one assumes (as is common in the cryptographic literature) that the best known (heuristic) algorithms for γ-SVP are essentially optimal, then this result implies significantly better security for lattice-based cryptography than other worst-case to average-case reductions.

In fact, Theorem 1.4 follows from an improvement to just one step in Micciancio and Regev's reduction. Specifically, to achieve the best possible approximation factor, Micciancio and Regev essentially used their SIS oracle to generate the witness used in Aharonov and Regev's coNP protocol.[3] Our generalization of Aharonov and Regev's protocol uses (a larger version of) the same witness, so that we almost get our generalization of [31] for free once we have generalized [6]. There are, however, many technical details to work out, as we describe in Section 1.2.4.

[3] There are simpler ways to use a SIS oracle to solve SVP that achieve a worse approximation factor—e.g., by using SIVP as an intermediate problem. But Micciancio and Regev's clever use of the [6] protocol yields the $\widetilde{O}(n)$ approximation factor that has remained the state of the art since a preliminary version of [31] was published in 2004.

(To get the best approximation factor that we can, we actually use our coMA protocol in some parameter regimes and our co-non-deterministic protocol in others. This works similarly because the witness is the same for the two protocols.)

Worst-case to average-case reductions for LWE. Our fifth and final main result is a generalization of both Regev's quantum worst-case to average-case reduction for LWE [36] and Peikert's classical version [33]. Since LWE comes with many parameters, in this high-level overview we simply present the special case of the result for the hardest choice of parameters that is known to imply public-key encryption.

Theorem 1.5 (Informal; see the full version [1, Theorem 5.3 and 5.5]). *For any $\gamma = \gamma(n) \geq 1$, public-key encryption exists if γ-SVP is $2^{n^2 \operatorname{polylog}(n)/\gamma}$ hard for a classical computer or $2^{n^3 \operatorname{polylog}(n)/\gamma^2}$ hard for a quantum computer. In particular, (exponentially secure) public-key cryptography exists if $\widetilde{O}(n)$-SVP is $2^{\Omega(n)}$ hard, even for a classical computer.*

Again, this is a strict generalization of the prior state of the art, which matched the above result for polynomial running time. And, again, we stress that $2^{\Omega(n)}$-hardness of $\widetilde{O}(n)$-SVP is a widely believed conjecture. Indeed, if one assumes (as is common in the cryptographic literature) that the best known (heuristic) algorithms for γ-SVP are essentially optimal, then this result implies significantly better security for lattice-based public-key cryptography than prior worst-case to average-case reductions.

In particular, notice that in the important special case of running time $2^{\varepsilon n}$, our quantum reduction and classical reduction achieve essentially the same approximation factor. (Indeed, they differ by only a constant factor.) So, perhaps surprisingly, there is no real gap between classical and quantum reductions in the exponential-time regime, unlike in the polynomial-time regime.

We note that behind this result is a new generalization of the polynomial-time reduction from SVP to the Bounded Distance Decoding problem (BDD). This polynomial-time reduction was implicit in [33] and made explicit in [27], and it can be viewed as a version of the [21] coAM protocol in which Merlin is simulated by a BDD oracle. We (of course!) generalize this by allowing the reduction to run in more time in order to achieve a better approximation factor, using the same (quite simple) idea that we used to generalize the [21] coAM protocol. (See the full version [1, Section 4].)

Furthermore, to obtain the best possible approximation factor in the classical result (and, in particular, an approximation factor that matches the quantum result in the $2^{\varepsilon n}$-time setting), we also observe that Peikert's celebrated classical reduction from BDD to LWE can be made to work for a wider range of parameters if it is allowed to run in superpolynomial time. At a technical level, this involves combining basis reduction algorithms (e.g., from [19]) with the discrete Gaussian sampling algorithm from [15, 20]. The resulting improved parameters results in a significant savings in the approximation factor, and even a small savings in the polynomial-time setting. (E.g., in the exponential-time setting, this saves us a factor of $\sqrt{n}$.)

Both of these observations follow relatively easily from combining known techniques. But, they might be of independent interest.

1.2 Our Techniques

1.2.1 A coAM *Protocol.* At a high level, our coAM protocol uses the following very elegant idea due to Goldreich and Goldwasser [21]. Recall that our goal is to describe a protocol between all-powerful Merlin and computationally bounded Arthur in which Merlin (for whatever mysterious reason) wishes to convince Arthur that $\boldsymbol{t}$ is far from the lattice. In particular, if $\operatorname{dist}(\boldsymbol{t}, \mathcal{L}) > 2$ (the FAR case), Merlin should be able to convince Arthur that $\boldsymbol{t}$ is far from the lattice. On the other hand, if $\operatorname{dist}(\boldsymbol{t}, \mathcal{L}) \leq d$ (the CLOSE case, where $d < 2$ will depend on Arthur's running time), then even if all-powerful Merlin tries his best to convince Arthur that $\boldsymbol{t}$ is far from the lattice, Arthur should correctly determine that Merlin is trying to trick him with high probability.

To that end, consider the set

$$S_0 := \bigcup_{\boldsymbol{y} \in \mathcal{L}} (\mathcal{B} + \boldsymbol{y}) ,$$

which is the union of balls of radius 1 centered around each lattice point, and the set

$$S_{\boldsymbol{t}} := \bigcup_{\boldsymbol{y} \in \mathcal{L}} (\mathcal{B} + \boldsymbol{y} - \boldsymbol{t}) = S_0 - \boldsymbol{t} ,$$

which instead consists of balls centered around lattice points shifted by $\boldsymbol{t}$. See Figure 2.

Notice that $\operatorname{dist}(\boldsymbol{t}, \mathcal{L}) > 2$ (i.e., the FAR case) if and only if S_0 and $S_{\boldsymbol{t}}$ are disjoint (ignoring the distinction between open and closed balls). On the other hand, if $\operatorname{dist}(\boldsymbol{t}, \mathcal{L}) \leq d < 2$ (the CLOSE case), then the two sets must overlap, with more overlap if d is smaller. Specifically, the intersection of the two sets will contain at least a

$$p_d \approx (1 - d^2/4)^{n/2} \approx e^{-d^2 n}$$

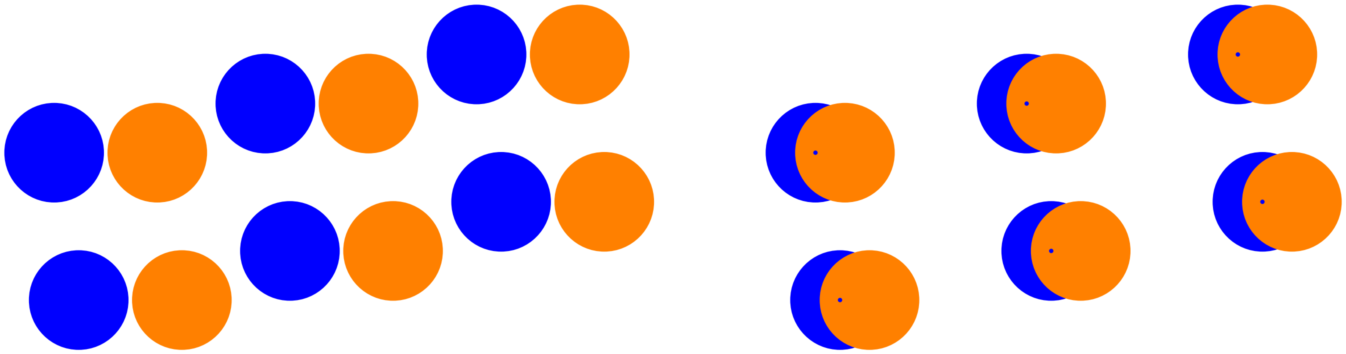
fraction of the total volume of S_0. (See the full version [1, Lemma 2.1] for the precise statement.)

So, Arthur first flips a coin. If it comes up heads, he samples a point $\boldsymbol{x} \sim S_0$ uniformly at random from S_0. Otherwise, he samples $\boldsymbol{x} \sim S_{\boldsymbol{t}}$.[4] He then sends the result to Merlin. Arthur then simply asks Merlin "was my coin heads or tails?" In other words, Arthur asks whether $\boldsymbol{x}$ was sampled from S_0 or $S_{\boldsymbol{t}}$. If we are in the FAR case where $\operatorname{dist}(\boldsymbol{t}, \mathcal{L}) > 2$, then Merlin (who, remember, is all powerful) will be able to unambiguously determine whether $\boldsymbol{x}$ was sampled from S_0 or $S_{\boldsymbol{t}}$, since they are disjoint sets. On the other hand, if $\operatorname{dist}(\boldsymbol{t}, \mathcal{L}) \leq d$, then with probability at least p_d, $\boldsymbol{x}$ will lie in the intersection of the two sets. When this happens, even all-powerful Merlin can do no better than randomly guessing Arthur's coin.

Arthur and Merlin can therefore play this game, say, n/p_d times. If we are in the FAR case, then an honest Merlin will answer correctly every time, and Arthur will correctly conclude that $\boldsymbol{t}$ is far from the lattice. If we are in the CLOSE case, then no matter what Merlin does, he is likely to guess wrong at least once, in which case Arthur will correctly conclude that Merlin is trying to fool him.

This yields a *private-coin* (honest-verifier perfect zero knowledge) protocol that runs in time roughly $1/p_d \approx e^{d^2 n}$ for γ-CVP with $\gamma = 2/d$. Similarly to the polynomial-time setting, one can then use standard generic techniques to convert any private-coin

[4] In fact, there is no uniformly random distribution over S_0 or $S_{\boldsymbol{t}}$, since they have infinite volume. In reality, we work with these sets *reduced modulo the lattice.* But, in this high-level description, it is convenient to pretend to work with the sets themselves.



$\mathrm{dist}(\mathrm{t}, \mathcal{L}) > 2$

$\mathrm{dist}(\mathrm{t}, \mathcal{L}) \le d$

Figure 2: Comparison of the sets S_0 and S_t in the FAR case and in the CLOSE case.

protocol into a true public-coin, two-round protocol (i.e., a true coAM protocol), at the expense of increasing the constant in the exponent.

1.2.2 A co-Non-Deterministic Protocol. Our co-non-deterministic protocol (as well as our coMA protocol) is based on the beautiful protocol of Aharonov and Regev [6]. The key tools are the *periodic Gaussian function* and the *discrete Gaussian distribution.* For $\boldsymbol{x} \in \mathbb{R}^n$, we define

$$\rho(\boldsymbol{x}) := e^{-\pi\|\boldsymbol{x}\|^2} \;,$$

and for a lattice $\mathcal{L} \subset \mathbb{R}^n$ and target vector $\boldsymbol{t} \in \mathbb{R}^n$, we extend this definition to the lattice coset $\mathcal{L} - \boldsymbol{t}$ as

$$\rho(\mathcal{L} - \boldsymbol{t}) := \sum_{\boldsymbol{y} \in \mathcal{L}} \rho(\boldsymbol{y} - \boldsymbol{t}) \;.$$

We can then define the periodic Gaussian function as

$$f(\boldsymbol{t}) := \frac{\rho(\mathcal{L} - \boldsymbol{t})}{\rho(\mathcal{L})} \;.$$

Very roughly speaking, we expect $f(\boldsymbol{t})$ to be a smooth approximation to the function $e^{-\pi \mathrm{dist}(\boldsymbol{t},\mathcal{L})^2}$, or at least to be relatively large when $\boldsymbol{t}$ is close to the lattice and relatively small when $\boldsymbol{t}$ is far from the lattice. See Figure 3.

Banaszczyk proved a number of important and beautiful results about the periodic Gaussian function [10]. In particular, he showed that

$$e^{-\pi\,\mathrm{dist}(\boldsymbol{t},\mathcal{L})^2} \le f(\boldsymbol{t}) \le 1 \;.$$

So, if $\boldsymbol{t}$ is close to the lattice, then $f(\boldsymbol{t})$ cannot be too small. On the other hand, if $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) \ge \sqrt{n}$, then Banaszczyk proved that $f(\boldsymbol{t}) < 2^{-n}$. So, if we could somehow approximate $f(\boldsymbol{t})$ up to an additive error of $\delta \in (2^{-n}, 1)$, then we could distinguish between the case when $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) \lesssim \sqrt{\log(1/\delta)}$ and the case when $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) \ge \sqrt{n}$, and therefore solve γ-CVP for $\gamma \approx \sqrt{n/\log(1/\delta)}$.

Of course, it is not immediately clear *how* to approximate $f(\boldsymbol{t})$, even with additional help from an all-powerful prover. However, Aharonov and Regev observed that suitably chosen short vectors from the *dual lattice* $\mathcal{L}^*$ can be used for this purpose. Specifically, they recalled from the Poisson summation formula that

$$f(\boldsymbol{t}) = \mathop{\mathbb{E}}_{\boldsymbol{w}\sim D_{\mathcal{L}^*}}[\cos(2\pi\langle \boldsymbol{w}, \boldsymbol{t}\rangle)] \;, \tag{1}$$

where $D_{\mathcal{L}^*}$ is the *discrete Gaussian distribution*, defined by

$$\Pr_{\boldsymbol{w}\sim D_{\mathcal{L}^*}}[\boldsymbol{w} = \boldsymbol{z}] := \frac{\rho(\boldsymbol{z})}{\rho(\mathcal{L}^*)}$$

for any $\boldsymbol{z} \in \mathcal{L}^*$. So, Aharonov and Regev had the prover provide the verifier with $W := (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N)$ sampled independently from $D_{\mathcal{L}^*}$. The verifier can then compute

$$f_W(\boldsymbol{t}) := \frac{1}{N}\sum_{i=1}^{N} \cos(2\pi\langle \boldsymbol{w}, \boldsymbol{t}\rangle) \;.$$

I.e., f_W is the sample approximation of Equation (1). By the Chernoff-Hoeffding bound, $f_W(\boldsymbol{t})$ will provide an approximation of $f(\boldsymbol{t})$ up to an error of roughly $1/\sqrt{N}$. So, this *almost* yields a roughly N-time non-deterministic protocol for distinguishing the FAR case when $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) \ge \sqrt{n}$ from the CLOSE case when $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) \lesssim \sqrt{\log N}$, i.e., a protocol for $\sqrt{n/\log N}$-CVP.

The one (rather maddening) issue with this protocol is that it is not clear how to maintain soundness against a cheating prover in the case when $\boldsymbol{t}$ is close to the lattice. I.e., suppose that the prover provides vectors $W := (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N)$ that are *not* sampled from the discrete Gaussian distribution. Then, $f_W(\boldsymbol{t})$ will presumably no longer be a good approximation to $f(\boldsymbol{t})$, and the verifier might therefore be fooled into thinking that $\boldsymbol{t}$ is far from the lattice when it is in fact quite close.

It seems that what we need is some sort of "test of Gaussianity" to "check that W looks like it was sampled from $D^N_{\mathcal{L}^*}$." Or, more accurately, we need some efficiently testable set of properties that (1) are satisfied by honestly sampled vectors $W = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N) \in \mathbb{R}^{n\times N}$ with high probability in the FAR case; and (2) are enough to imply that $f_W(\boldsymbol{t})$ is not too small in the CLOSE case when $\boldsymbol{t}$ is relatively close to $\mathcal{L}$. One crucial observation is that, as long as the $\boldsymbol{w}_i$ are dual lattice vectors, then it suffices in the CLOSE case to consider $f_W(\boldsymbol{u})$ for $\boldsymbol{u}$ that are relatively short. This is because the function f_W is *periodic* over the lattice, so that $f_W(\boldsymbol{t}) = f_W(\boldsymbol{u})$ where $\boldsymbol{u} := \boldsymbol{t} - \boldsymbol{y}$ for $\boldsymbol{y} \in \mathcal{L}$ a closest lattice vector to $\boldsymbol{t}$. (It is crucial to remember that $\boldsymbol{u}$ is only used for the analysis. In particular, the verifier cannot compute $\boldsymbol{u}$ efficiently.)

To create a sound protocol, Aharonov and Regev therefore studied the second-order Taylor series expansion of $f_W(\boldsymbol{u})$ around $\boldsymbol{u} = \boldsymbol{0}$,

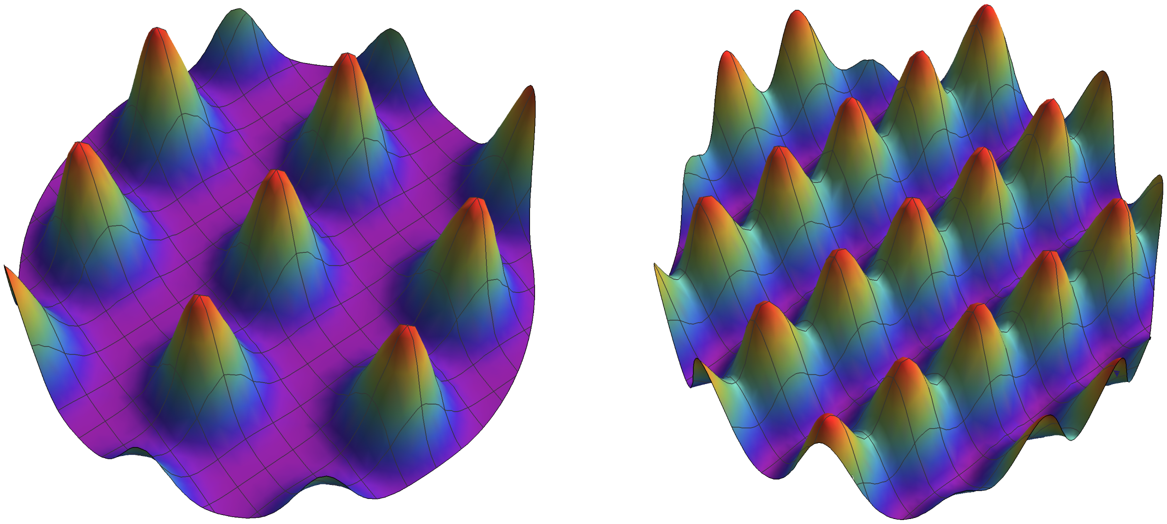


Figure 3: The periodic Gaussian function $f(t)$ for two different two-dimensional lattices $\mathcal{L}$.

i.e.,

$$\begin{aligned} f_W(\boldsymbol{u}) &= 1 - \frac{2\pi^2}{N} \cdot \sum_{i=1}^{N} \langle \boldsymbol{w}_i, \boldsymbol{u}\rangle^2 + \frac{2\pi^4}{3N} \sum_{i=1}^{N} \langle \boldsymbol{w}_i, \boldsymbol{u}\rangle^4 - \cdots \\ &\geq 1 - \frac{2\pi^2}{N} \cdot \sum_{i=1}^{N} \langle \boldsymbol{w}_i, \boldsymbol{u}\rangle^2 \\ &\geq 1 - 2\pi^2 \cdot \|WW^T/N\| \cdot \|\boldsymbol{u}\|^2 , \end{aligned}$$

where $WW^T/N = \frac{1}{N}\sum_i \boldsymbol{w}_i \boldsymbol{w}_i^T \in \mathbb{R}^{n \times n}$ and $\|WW^T/N\|$ is the spectral norm. In particular, $f_W(\boldsymbol{u})$ will be large for short $\boldsymbol{u}$, provided that WW^T/N has small spectral norm. One can show (again using the Poisson summation formula) that an honestly sampled witness W will satisfy, say, $\|WW^T/N\| \leq 1$ with high probability. And, the verifier can of course efficiently check this because the spectral norm is efficiently computable. So, Aharonov and Regev used this simple test as their "test of Gaussianity."

Putting everything together, we see that by checking that W consists of dual vectors, that $\|WW^T/N\| \leq 1$, and that, say, $f_W(\boldsymbol{t}) < 1/2$, the verifier will always reject when $\mathrm{dist}(\boldsymbol{t}, \mathcal{L})$ is smaller than some constant in the CLOSE case, regardless of W. And, it will accept (with high probability over the choice of witness W) when W is sampled honestly and $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) \geq \sqrt{n}$ in the FAR case. This yields the final approximation factor of $O(\sqrt{n})$ achieved in [6].

Notice, however, that by using this spectral-norm-based "test of Gaussianity," Aharonov and Regev only achieved an approximation factor of $O(\sqrt{n})$, rather than the approximation factor $O(\sqrt{n/\log N})$ that they would have gotten if they could have somehow guarantee that the W were sampled honestly. In particular, when $N = \mathrm{poly}(n)$, this costs a factor of roughly $\sqrt{\log n}$ in the approximation factor. (At a technical level, this factor of $\sqrt{\log n}$ is lost because the second-order approximation $\cos(x) \approx 1 - x^2/2$ is of course only accurate when $|x|$ is bounded by some small fixed constant.)

Fixing this (again, rather maddening) loss of a $\sqrt{\log n}$ factor has been a major open problem ever since. More generally, it is not at all clear how to achieve even a slightly better approximation factor using these ideas, even if we are willing to increase N and the running time of our verifier substantially. It seems relatively clear that a more demanding "test of Gaussianity" is needed.

A natural idea would be to approximate f_W via a higher-order Taylor series approximation,

$$f_W^{(k)}(\boldsymbol{u}) := 1 - \sum_{j=1}^{k} \frac{(2\pi)^{2j}}{(2j)!} \sum_{i=1}^{N} \langle \boldsymbol{w}_i, \boldsymbol{u}\rangle^{2j}/N \; .$$

It is not hard to see that $f_W^{(k)}$ is quite close to f_W provided that $\boldsymbol{u}$ is not too long. Specifically,

$$|f_W^{(k)}(\boldsymbol{u}) - f_W(\boldsymbol{u})| \lesssim \frac{1}{N} \cdot \sum_{i=1}^{N} (\langle \boldsymbol{w}_i, \boldsymbol{u}\rangle/k)^{2k} \; .$$

We know that when W is sampled honestly, this error cannot be much larger than roughly $(\|\boldsymbol{u}\|^2/k)^k$ (with high probability). Therefore, when W is sampled honestly, it must be the case that the *moments* $\sum_{i=1}^{N} \langle \boldsymbol{w}_i, \boldsymbol{u}\rangle^{2j}/N$ of W have some property that guarantees that $f_W^{(k)}(\boldsymbol{u}) \gtrsim e^{-\pi\|\boldsymbol{u}\|^2} - (\|\boldsymbol{u}\|^2/k)^k$. If we could somehow identify and test this property efficiently for sufficiently large k, then we could use this as our "test of Gaussianity," and we would be done.

However, we do not know how to test this property efficiently, and it seems quite hard to do so in general. Even just for $j = 2$, it is in general computationally hard even to approximate, say,

$$\max_{\|\boldsymbol{u}\| \leq d} \frac{1}{N} \cdot \sum_{i=1}^{N} \langle \boldsymbol{w}_i, \boldsymbol{u}\rangle^{2j} \; ,$$

as this is exactly the matrix two-to-four norm [11]. (Compare this with the case of $j = 1$, which yields the easy-to-compute spectral norm.) And, bounding the specific sum $f_W^{(k)}$ that interests us seems significantly more complicated than bounding an individual term—perhaps particularly because it is an alternating sum. It is therefore entirely unclear how to efficiently certify that the sum defining $f_W^{(k)}(\boldsymbol{u})$ is bounded whenever $\boldsymbol{u}$ is bounded.

We solve this problem by asking for additional properties of our lattice $\mathcal{L}$ in the FAR case that allow us to make this problem tractable. Specifically, we require that in the FAR case, not only do we have $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) \geq \sqrt{n}$, but we also have that $\mathcal{L}$ has no non-zero vectors shorter than $\sqrt{n}$. (Intuitively, this means that "the Gaussian peaks of $f(\boldsymbol{t})$ are well separated," as in the left example in Figure 3.) Micciancio and Regev [31] considered this more restrictive promise problem (for roughly the same reason) and observed that γ-SVP can be reduced to it. (It is this additional requirement in the FAR case that prevents us from obtaining a protocol for CVP, rather than SVP.)

Via Fourier-analytic techniques, we show that this new requirement implies that in the FAR case, the *moments* of the discrete Gaussian $D_{\mathcal{L}^*}$,

$$\mathop{\mathbb{E}}_{\boldsymbol{w} \sim D_{\mathcal{L}^*}} [w_1^{j_1} \cdots w_n^{j_n}] \ ,$$

are extremely close to the corresponding moments of the continuous Gaussian distribution as long as the j_i are non-negative integers such that $\sum j_i$ is not too large. (See the full version [1, Lemma 2.11].) We then observe that these moments for $\sum j_i \leq 2k$ completely characterize $f_W^{(k)}(\boldsymbol{u})$.

So, while in general it seems to be difficult to determine whether a given witness W has the property that $f_W^{(k)}(\boldsymbol{u})$ is not too small for all sufficiently short $\boldsymbol{u}$, we show that in our special use case, it suffices for the verifier to check that the sample moments

$$\frac{1}{N} \sum_{i=1}^{N} w_{i,1}^{j_1} \cdots w_{i,n}^{j_n}$$

are close to some specific known values for $\sum j_i \leq 2k$.

There are roughly n^{2k} such moments to check, and each can be checked in time essentially N. If these checks pass, then we can use $f_W(\boldsymbol{t})$ to distinguish the CLOSE case from the FAR case as long as in the close case we have

$$1/\sqrt{N} + (\mathrm{dist}(\boldsymbol{t}, \mathcal{L})^2/k)^k \lesssim e^{-\pi \mathrm{dist}(\boldsymbol{t}, \mathcal{L})^2} \ .$$

In particular, by setting $N = n^{O(k)}$, we will not be fooled in the CLOSE case as long as $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) \lesssim \sqrt{k}/10$, which gives our approximation factor of roughly $\sqrt{n/k}$ in time roughly $n^{O(k)}$.[5] See the full version [1, Section 6].

1.2.3 *A* coMA *Protocol.*

Our coMA protocol combines some of the beautiful ideas from [6] with some of the equally beautiful ideas from [21].

Indeed, recall that [6] and our generalization show how to generate a witness W of size roughly N such that, if W is sampled honestly, it can be used to distinguish the case when $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) \geq \sqrt{n}$ from the case when $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) \lesssim \sqrt{\log N}$. Specifically, there is a simple function $f_W(\boldsymbol{t})$ that is large in the CLOSE case but small in the FAR case, provided that the witness W is generated honestly. The difficulty, in both the original Aharonov and Regev protocol and in our version described above, is in how to handle dishonestly generated W, in which case f_W might not have this property and might therefore lead Arthur to incorrectly think that $\boldsymbol{t}$ is far from the lattice when in fact it is close.

On the other hand, [21] and our generalize work by either sampling $\boldsymbol{x}$ from a ball around a lattice point or sampling $\boldsymbol{x}$ from a ball around (a lattice shift of) $\boldsymbol{t}$. Then, in the FAR case, a random vector $\boldsymbol{x}$ sampled from a ball around a lattice point will always be closer to $\mathcal{L}$ than a random vector $\boldsymbol{x}$ sampled from a ball around $\boldsymbol{t}$. So, in the FAR case, an honest Merlin can determine whether $\boldsymbol{x}$ was sampled from one distribution or the other by checking whether $\mathrm{dist}(\boldsymbol{x}, \mathcal{L})$ is large or small. On the other hand, in the CLOSE case, there is some overlap between the distributions, so that no matter how Merlin behaves, he will not be able to consistently distinguish between the two cases. (Recall Figure 2.)

Our idea is therefore to have Arthur "use W to simulate Merlin's behavior in the coAM protocol." In particular, the witness for our protocol is exactly the same W that we use as a witness in our co-non-deterministic protocol (and therefore simply a larger version of the original [6] witness). However, Arthur's verification procedure is quite different (and, of course, it is now randomized, which is why we obtain a coMA protocol). To verify Merlin's claim that $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) \geq \sqrt{n}$, Arthur repeatedly samples $\boldsymbol{x}_0$ from a ball of radius r around $\boldsymbol{0}$ and $\boldsymbol{x}_1$ from a ball of radius r around $\boldsymbol{t}$, where r is to be set later. Arthur then computes $f_W(\boldsymbol{x}_0)$ and $f_W(\boldsymbol{x}_1)$ and rejects (i.e., guesses that he is in the CLOSE case) unless $f_W(\boldsymbol{x}_0)$ is large and $f_W(\boldsymbol{x}_1)$ is small.

Note that, at least at a high level, the completeness of our protocol in the FAR case follows from the analysis of [6]. In particular, if W is sampled honestly, then $f_W(\boldsymbol{x}_0)$ will be large as long as $r \lesssim \sqrt{\log N}$, and $f_W(\boldsymbol{x}_1)$ will be small as long as $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) - r \gtrsim \sqrt{n}$. On the other hand, the soundness of our protocol in the CLOSE case follows from the analysis of [21]. In particular, if $\mathrm{dist}(\boldsymbol{t}, \mathcal{L}) \leq d \leq r$, then regardless of our choice of f_W, Arthur will reject with probability at least

$$p_{d/r}/2 \approx (1 - d^2/(4r^2))^{n/2} \approx e^{-d^2 n/r^2} \ ,$$

as in our discussion of the coAM protocol above. By running this test, say, $n/p_{d/r}$ times, Arthur will reject with high probability in the CLOSE case.

Plugging in numbers, we take r as large as we possibly can without violating completeness, so we take $r \approx \sqrt{\log N}$. Our final protocol then has a witness of size roughly N, an approximation factor of roughly $\sqrt{n}/d$, and a running time of roughly $N \cdot e^{d^2 n/r^2} \approx N \cdot e^{d^2 n/\log N}$. The most natural setting of parameters takes $d \approx \log N/\sqrt{n}$, which gives an approximation factor of $n/\log N$ in $\mathrm{poly}(N)$ time. (However, we note that the protocol is also potentially interesting in other parameter settings; e.g., one can obtain non-trivial approximation factors with relatively small communication size N by allowing Arthur to run in more time. In contrast, our other protocols seem to require roughly as much communication as computation.) See the full version [1, Section 7].

1.2.4 *Worst-Case to Average-Case Reductions for SIS.*

Our generalization of Micciancio and Regev's worst-case to average-case reduction for SIS comes nearly for free after all the work we did to develop our variant of Aharonov and Regev's coNP protocol (and our coMA protocol). In particular, Micciancio and Regev's

[5] This description might suggest that we can take $N \leq 2^{O(k)}$, yielding a $n^{O(k)}$-time protocol with $2^{O(k)}$-sized witness. However, in this informal discussion we are ignoring the error that we incur from the fact that the sample moments $\frac{1}{N}\sum_i w_{i,1}^{j_1} \cdots w_{i,n}^{j_n}$ will deviate from their expectation. After accounting for this, we are forced to take $N \geq n^{\Omega(k)}$.

worst-case to average-case reduction essentially shows how to use a SIS oracle to sample from $D_{\mathcal{L}^*}$ (provided that $\lambda_1(\mathcal{L})$ is not too small). They then used this to generate the witness for Aharonov and Regev's protocol, allowing them to solve γ-SVP. Our co-non-deterministic protocol also uses samples from $D_{\mathcal{L}^*}$ as a witness (as does our coMA protocol). So, we are more-or-less able to use the exact same idea to obtain our generalization of Micciancio and Regev's result. (In fact we are able to work in the more general setting of Micciancio and Peikert [30], who showed a reduction that works for smaller moduli than [31].)

However, our co-non-deterministic protocol is a bit more delicate than the protocol in [6]. Specifically, our reduction really does need to produce samples from $D_{\mathcal{L}^*}$ in the FAR case. In contrast, [31] (and, to our knowledge, all other worst-case to average-case reductions for SIS) were only able to show how to use a SIS oracle to produce samples from some mixture of discrete Gaussian distributions with potentially different parameters (i.e., different standard deviations). At a technical level, this issue arises because the SIS oracle can potentially output vectors with different lengths, resulting in discrete Gaussian samples with different parameters.

We overcome this (annoying!) technical difficulty by showing how to control the parameter of the samples generated by the reduction, showing that a SIS oracle is in fact sufficient to produce samples from the distribution $D_{\mathcal{L}^*}$ itself (provided that the smoothing parameter of $\mathcal{L}^*$ is small enough). Our reduction mostly follows the elegant and well known reduction of Micciancio and Peikert [30]. And, though the proof does not require substantial new ideas, we expect that the result will be useful in future work—as a reduction directly from discrete Gaussian sampling should be quite convenient. (See the full version [1, Theorem 8.3].)

Finally, in order to get the best approximation factor that we can, we actually use our coMA protocol when the running time is large, rather than our co-non-deterministic protocol. E.g., our coMA protocol saves a factor of $\sqrt{\log n}$ in the approximation factor over our co-non-deterministic protocol in the important special case when the running time is $2^{\varepsilon n}$. And, our worst-case to average-case reduction inherits this savings. See the full version [1, Section 8].

1.2.5 Worst-Case to Average-Case Reductions for LWE. Recall that we show two worst-case to average-case reductions for LWE. One is a *quantum* reduction, following Regev [36]. The other is a *classical* reduction, following Peikert [33]. In both cases, our modifications to prior work are surprisingly simple.

In the quantum case, the *only* difference between our reduction and prior work is in a single step. Specifically, Regev's original quantum reduction is most naturally viewed as a reduction from BDD to LWE. However, BDD is not nearly as well studied as SVP. Regev therefore used elegant quantum computing tricks to obtain hardness directly from SVP. However, Peikert [33] and Lyubashevsky and Micciancio [27] later showed a simple classical reduction from SVP to BDD that is perhaps best viewed as a version of the coAM protocol from [21] in which the BDD oracle is used to simulate Merlin. (This reduction was implicit in [33] and made explicit in [27].)

Using the same ideas that we used to generalize the coAM protocol from [21], we show how to generalize this reduction from SVP to BDD—showing that a better approximation factor is achievable if the reduction is allowed more running time. By composing this reduction with Regev's reduction from BDD to LWE, we similarly show a time-approximation-factor tradeoff for LWE.

To generalize Peikert's *classical* reduction, we use the above idea and also make one other simple modification to the reduction. Specifically, Peikert showed how to use a sufficiently "nice" basis of the dual lattice $\mathcal{L}^*$ to reduce BDD to LWE, where the modulus of the LWE instance depends on how "nice" the basis is.[6] He then used the celebrated LLL algorithm [26] to efficiently find a relatively nice basis. We simply plug in generalizations of [26] that obtain better bases in more time [19, 37]. (In fact, even in the polynomial-time regime, this improves on Peikert's approximation factor by a small polylogarthmic term. See the full version [1, Section 5].)

To achieve the best parameters, we rely on the "direct-to-decision" reduction of [35] (in both the classical and quantum setting), allowing us to avoid the search-to-decision reductions that were used in work prior to [35]. (Search-to-decision reductions that do not increase the approximation factor are known for some moduli, but for other moduli, the only known search-to-decision reductions incur a loss in the approximation factor.)

ACKNOWLEDGEMENTS

NSD thanks Chris Peikert for responding to oracle queries in less than unit time. DA, HB, AG, and NSD would like to thank the organizers of the Bertinoro program on Fine Grained Approximation Algorithms and Complexity at which some of this work was completed. DA, HB, RK, SP, NSD, and VV would like to thank the Simons Institute and the organizers of the Lattices and Beyond Summer Cluster at which some of this work was done. Finally, HB, SP, and NSD would like to thank the National University of Singapore's Centre for Quantum Technologies for graciously supporting their visit, at which, again, some of this work was done.

REFERENCES

[1] Divesh Aggarwal, Huck Bennett, Zvika Brakerski, Alexander Golovnev, Rajendra Kumar, Zeyong Li, Spencer Peters, Noah Stephens-Davidowitz, and Vinod Vaikuntanathan. 2022. Lattice Problems Beyond Polynomial Time. arXiv:2211.11693 [cs.CC] https://arxiv.org/abs/2211.11693.

[2] Divesh Aggarwal, Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. 2021. Fine-Grained Hardness of CVP(P)—Everything That We Can Prove (and Nothing Else). In *SODA*. http://arxiv.org/abs/1911.02440

[3] Divesh Aggarwal and Eldon Chung. 2021. A Note on the Concrete Hardness of the Shortest Independent Vector in Lattices. *Inform. Process. Lett.* 167 (2021).

[4] Divesh Aggarwal, Zeyong Li, and Noah Stephens-Davidowitz. 2021. A $2^{n/2}$-Time Algorithm for $\sqrt{n}$-SVP and $\sqrt{n}$-Hermite SVP, and an Improved Time-Approximation Tradeoff for (H)SVP. In *Eurocrypt*. http://arxiv.org/abs/2007.09556

[5] Divesh Aggarwal and Noah Stephens-Davidowitz. 2018. (Gap/S)ETH Hardness of SVP. In *STOC*. http://arxiv.org/abs/1712.00942

[6] Dorit Aharonov and Oded Regev. 2005. Lattice Problems in NP ∩ coNP. *J. ACM* 52, 5 (2005), 749–765. Preliminary version in FOCS, 2005.

[7] Miklós Ajtai. 1996. Generating Hard Instances of Lattice Problems. In *STOC*.

[8] Miklós Ajtai. 1998. The Shortest Vector Problem in L_2 Is NP-hard for Randomized Reductions. In *STOC*.

[9] Roberto Avanzi, Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. 2021. CRYSTALS-Kyber (Version 3.02) – Submission to Round 3 of the NIST Post-Quantum Project. (2021). https://pq-crystals.org/kyber/resources.shtml.

[10] Wojciech Banaszczyk. 1993. New Bounds in Some Transference Theorems in the Geometry of Numbers. *Math. Ann.* 296, 4 (1993), 625–635.

[6] To build public-key cryptography with a larger modulus, it seems that we must take the error rate in the LWE instance to be larger, which yields a worse approximation factor for SVP. This was fundamentally the reason why Peikert's classical reduction required hardness of SVP with a larger approximation factor in order to build public-key encryption, compared to Regev's quantum reduction. Later work [15] showed how to prove classical hardness of LWE with a smaller modulus. However, the [15] reduction incurs additional loss in the approximation factor.

[11] Boaz Barak, Fernando G.S.L. Brandao, Aram W. Harrow, Jonathan Kelner, David Steurer, and Yuan Zhou. 2012. Hypercontractivity, Sum-of-Squares Proofs, and Their Applications. In *STOC*.

[12] Huck Bennett. 2022. The Complexity of the Shortest Vector Problem. Invited survey. To appear, SIGACT News Open Problems Column.

[13] Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. 2017. On the Quantitative Hardness of CVP. In *FOCS*. http://arxiv.org/abs/1704.03928

[14] Huck Bennett, Chris Peikert, and Yi Tang. 2022. Improved Hardness of BDD and SVP under Gap-(S)ETH. In *ITCS*.

[15] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. 2013. Classical Hardness of Learning with Errors. In *STOC*. http://arxiv.org/abs/1306.0281

[16] Zvika Brakerski, Noah Stephens-Davidowitz, and Vinod Vaikuntanathan. 2021. On the Hardness of Average-Case k-SUM. In *RANDOM*.

[17] Jin-Yi Cai and Ajay Nerurkar. 1999. Approximating the SVP to within a factor $(1 + 1/\dim^{\epsilon})$ is NP-hard under Randomized Reductions. *J. Comput. System Sci.* 59, 2 (1999), 221–239.

[18] Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. 2003. Approximating CVP to within Almost-Polynomial Factors Is NP-Hard. *Combinatorica* 23, 2 (2003), 205–243.

[19] Nicolas Gama and Phong Q. Nguyen. 2008. Finding Short Lattice Vectors within Mordell's Inequality. In *STOC*.

[20] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. 2008. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *STOC*. https://eprint.iacr.org/2007/432

[21] Oded Goldreich and Shafi Goldwasser. 2000. On the Limits of Nonapproximability of Lattice Problems. *J. Comput. System Sci.* 60, 3 (2000), 540–563. https://doi.org/10.1006/jcss.1999.1686 Preliminary version in STOC 1998.

[22] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. 2011. Collision-Free Hashing from Lattice Problems. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*. Springer, 30–39. See also the original version https://eccc.weizmann.ac.il/eccc-reports/1996/TR96-042/index.html, from 1996.

[23] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and Jean-Pierre Seifert. 1999. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Inform. Process. Lett.* 71, 2 (1999), 55–61.

[24] Ishay Haviv and Oded Regev. 2012. Tensor-Based hardness of the Shortest Vector Problem to within Almost Polynomial Factors. *Theory of Computing* 8, 23 (2012), 513–531.

[25] Subhash Khot. 2005. Hardness of Approximating the Shortest Vector Problem in Lattices. *J. ACM* 52, 5 (2005), 789–808.

[26] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. 1982. Factoring Polynomials with Rational Coefficients. *Math. Ann.* 261, 4 (1982), 515–534.

[27] Vadim Lyubashevsky and Daniele Micciancio. 2009. On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem. In *CRYPTO*.

[28] Daniele Micciancio. 2001. The Shortest Vector Problem Is NP-hard to Approximate to within Some Constant. *SIAM J. Comput.* 30, 6 (2001), 2008–2035.

[29] Daniele Micciancio. 2012. Inapproximability of the Shortest Vector Problem: Toward a deterministic reduction. *Theory of Computing* 8 (2012), 487–512.

[30] Daniele Micciancio and Chris Peikert. 2013. Hardness of SIS and LWE with Small Parameters. In *CRYPTO*.

[31] Daniele Micciancio and Oded Regev. 2007. Worst-Case to Average-Case Reductions Based on Gaussian Measures. *SIAM Journal of Computing* 37, 1 (2007), 267–302. Preliminary version in FOCS 2004.

[32] NIST. 2022. Selected Algorithms 2022 - Post-Quantum Cryptography. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

[33] Chris Peikert. 2009. Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem. In *STOC*.

[34] Chris Peikert. 2016. A Decade of Lattice Cryptography. *Foundations and Trends in Theoretical Computer Science* 10, 4 (2016), 283–424.

[35] Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. 2017. Pseudorandomness of Ring-LWE for Any Ring and Modulus. In *STOC*.

[36] Oded Regev. 2009. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. *J. ACM* 56, 6 (2009), Art. 34, 40. Preliminary version in STOC 2005.

[37] Claus-Peter Schnorr. 1987. A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms. *Theoretical Computer Science* 53, 23 (1987), 201–224.

[38] Peter van Emde Boas. 1981. *Another NP-Complete Problem and the Complexity of Computing Short Vectors in a Lattice*. Technical Report. University of Amsterdam, Department of Mathematics, Netherlands.

[39] Ryan Williams. 2016. Strong ETH Breaks With Merlin and Arthur: Short Non-Interactive Proofs of Batch Evaluation. In *CCC*.

Received 2022-11-07; accepted 2023-02-06