Creating a Challenge for Ideal Lattices

Thomas Plantard[1] and Michael Schneider[2]

[1] University of Wollongong, Australia
thomaspl@uow.edu.au
[2] Technische Universität Darmstadt, Germany
mischnei@cdc.informatik.tu-darmstadt.de

Abstract. Lattice-based cryptography is one of the candidates in the area of post-quantum cryptography. Cryptographic schemes with security reductions to hard lattice problems (like the Shortest Vector Problem SVP) offer an alternative to recent number theory-based schemes. In order to guarantee asymptotic efficiency, most lattice-based schemes are instantiated using polynomial rings over integers. These lattices are called *ideal lattices*. It is assumed that the hardness of lattice problems in lattices over integer rings remains the same as in regular lattices. In order to prove or disprove this assumption, we instantiate random ideal lattices that allow to test algorithms that solve SVP and its approximate version. The Ideal Lattice Challenge allows online submission of short vectors to enter a hall of fame for full comparison. We adjoin a set of first experiments and a first comparison of ideal and regular lattices.

Keywords: Lattice-Based Cryptography, Ideal Lattices, Cyclotomic Rings

1 Introduction

Lattice-based cryptography is one of the candidates to replace number theory-based cryptographic schemes in the future. Many cryptographic primitives can be build on lattices. They are asymptitically efficient and provable secure, even based on worst-case problems. And lattice-based constructions are considered to withstand attacks with quantum computers. There are nonetheless numerous open problems in this research area. One of the long outstanding questions is whether ideal lattices, which are widely deployed in lattice-based cryptographic constructions, offer the same security as regular lattices. Ideal lattices are structured lattices used in cryptographic practice due to their algebraic properties, which allow faster computation and more efficient storage of cryptographic primitives. Instead of storing $\tilde{\mathcal{O}}(d^2)$ values and requiring computation time of $\tilde{\mathcal{O}}(d^2)$, ideal lattices reduce both to $\tilde{\mathcal{O}}(d)$, where d is the lattice dimension. To date, the security of the most efficient lattice-based

cryptosystems relies on the assumption that problems in ideal lattices are not easier to solve. If it turns out that ideal lattice problems are easier to solve than their regular counterparts, this would change the whole research area of lattice-based cryptography.

The Shortest Vector Problem (SVP), and more exactly its approximate version (α-SVP), is the problem considered most in cryptanalysis of lattice-based schemes. There are multiple algorithms that solve both problems and have been improved over the last years. For *regular* lattices, there exist standardized instances, e.g., the lattices of the SVP Challenge are considered as random instances [GM03]. They are widely deployed for testing SVP algorithms. For *ideal lattices* however, there is neither a standard way how to instantiate them nor a set to download sample instances or a public generator.

So far it is unclear if algorithms for lattice problems can make use of the special structure of ideal lattices. The main goal of our ideal lattice challenge is to assess whether it is possible to solve problems in ideal lattice faster than in regular lattices. A second goal is to allow people to work on standardized instances in order to compare their algorithms.

1.1 Our Contribution

As an amendment of the existing lattice challenges we introduce the Ideal Lattice Challenge for solving hard lattice problems in ideal lattices. We offer a standard way to create ideal lattices, in order to allow for testing algorithms on standard instances. This new challenge allows to draw a comparison between regular lattices in the existing challenges on the one hand and ideal lattices on the other hand. If there are algorithms that can exploit the special structure of ideal lattices, our new challenge will show that. Our ideal lattices can be created using different cyclotomic polynomials, among others the polynomial x^d+1 which is most commonly used in cryptographic constructions. It is an open question if different polynomials lead to ideal lattice problems with different hardness. This is another goal of our new challenge - assessing if the structure of these polynomials changes the hardness of the underlying lattice problems.

We add a section about first experiments on our ideal lattices in small dimensions, to assess if the choices made in generation of

the challenge were reasonable. In these experiments we already encounter a first difference of ideal lattices to regular ones: it is possible that the Gaussian heuristic, which is commonly believed to hold true for random lattices, does not hold true in the exact same way for ideal lattices. This heuristic is an important tool for estimating the security of lattice-based cryptosystems.

1.2 Notation and Definitions

A lattice $\mathcal{L}$ is a discrete additive subgroup of $\mathbb{R}^d$. Full-dimensional lattices are represented by a set of linearly independent basis vectors $\mathbf{b}_i \in \mathbb{R}^d$ for $1 \leq i \leq d$. The number d of basis vectors is the dimension of the lattice. These basis vectors can be represented as a matrix $\mathbf{B} \in \mathbb{R}^{d \times d}$ of row vectors $\mathbf{b}_i$. The lattice generated by the basis $\mathbf{B}$ is the set of all integer linear combinations of the basis vectors, i.e., $\mathcal{L}(\mathbf{B}) = \{\sum_{i=1}^{d} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$. For $d > 1$, a lattice has infinitely many bases, and by multiplication of a basis matrix with any unimodular transformation matrix it is possible to change the basis of a lattice. For representation it is common to work with basis vectors over $\mathbb{Z}$.

The volume or determinant of the lattice is denoted $\det(\mathcal{L})$. It remains the same for all bases of $\mathcal{L}$, and for any basis $\mathbf{B}$ of $\mathcal{L}$ it can be computed as $\det(\mathcal{L}) = |\det(\mathbf{B})|$. The norm of a shortest non-zero vector in a lattice $\mathcal{L}$ is denoted $\lambda_1(\mathcal{L})$. A heuristic estimation of this length can be derived using the Gaussian heuristic $GH(\mathcal{L})$:

$$\lambda_1(\mathcal{L}) \approx GH(\mathcal{L}) = \frac{\Gamma(d/2+1)^{1/d}}{\sqrt{\pi}} \cdot \det(\mathcal{L})^{1/d}\,.$$

The Euclidean norm of an element $\mathbf{x}$ is denoted $\|\mathbf{x}\|$.

Lattice Problems. For given $\alpha \geq 1$, the approximate shortest vector problem α-SVP is defined as follows. Given a basis $\mathbf{B}$ of a lattice $\mathcal{L}(\mathbf{B})$, find a non-zero vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ of norm $\|\mathbf{v}\| \leq \alpha \cdot \lambda_1(\mathcal{L}(\mathbf{B}))$. For $\alpha = 1$ this is the exact SVP. It is possible to state these problems for different norms, but we will only use the Euclidean norm in this work.

Due to the fact that $\lambda_1(\mathcal{L})$ is often unknown in practice, it is common to consider the δ-Hermite-SVP instead of the α-SVP: Given

a basis $\mathbf{B}$ of a lattice $\mathcal{L}(\mathbf{B})$, find a non-zero vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ of norm

$$\|\mathbf{v}\| \leq \delta^d \cdot \det(\mathcal{L}(\mathbf{B}))^{1/d}, \tag{1}$$

where d is the dimension of the lattice.

An important notion that derives from the Hermite-SVP is the *root Hermite factor* δ, which can be computed using (1). Given a vector $\mathbf{v}$ of length $\|\mathbf{v}\|$, the corresponding root Hermite factor is

$$\left(\frac{\|\mathbf{v}\|}{\det(\mathcal{L})^{1/d}}\right)^{1/d}.$$

It is common to use this root Hermite factor for comparison among different lattices, since it is independent of the basis.

Lattice Reduction and SVP Algorithms. Roughly speaking, lattice basis reduction is the search for short bases of a lattice. With this, the algorithms for lattice basis reduction also solve α-SVP. The most important algorithms used for reduction of lattice bases are the LLL and BKZ algorithm. LLL was presented in [LLL82]. It is a polynomial time algorithm that outputs a first basis vector which can be proven to be exponentially far from a shortest one. BKZ is a stronger generalization of LLL working in blocks of basis vectors. This algorithm was introduced in [SE94]. BKZ is parameterized by a blocksize parameter β, i.e., BKZ-β uses blocks of size β. BKZ-2 is the LLL algorithm. Higher blocksizes lead to better reduction quality (shorter vectors), at the expense of increasing runtime. More exactly, the runtime increases exponentially in β. In the lattice dimension d, BKZ behaves polynomially, but the runtime has not been proven theoretically. The most powerful variant is the BKZ 2.0 algorithm [CN11]. An implementation of BKZ 2.0 unfortunately is not publicly available.

When it comes to solve the exact shortest vector problem, the Extreme Pruning Enumeration algorithm is the strongest one in practice [GNR10,KSD+11]. It is a heuristic variant of full Enumeration [Kan83,FP85,SE94]. In theory, the probabilistic Sieving algorithms of [AKS01,NV08,MV10b] and the deterministic Voronoi cell algorithm of [MV10a] are superior. They run in single exponential time in the lattice dimension d, whereas enumeration takes double

exponential time in d. Solving SVP in practice is only manageable up to dimension $d \approx 120$ with Extreme Pruning Enumeration.

More information on lattice reduction and SVP algorithms can be found in [NV10,HPS11]. Following [GN08] the LLL algorithm reaches root Hermite factor of about 1.0219 and BKZ-20 reaches 1.0128 in practice.

Ideal Lattices. Ideal lattices are defined over the ring of integer polynomials $\mathbb{Z}[x]$. A lattice element $(f_0, \ldots, f_{d-1})$ corresponds to a polynomial $\sum_{i=0}^{d-1} f_i x^i$. A lattice $\mathcal{L}$ is an ideal lattice, if there exists a monic polynomial $g \in \mathbb{Z}[x]$ such that

$$\begin{aligned} (f_0, f_1, \ldots, f_{\deg(g)-1}) \in \mathcal{L} \quad &\iff \\ (f'_0, f'_1, \ldots, f'_{\deg(g)-1}) \in \mathcal{L}, \quad f' = xf \bmod g\,. \end{aligned}$$

In other words, multiplications of lattice elements with x in the polynomial ring remain within the lattice.

Ideal lattices were introduced in 2007 [Mic07] by Daniele Micciancio to improve the space complexity of lattice based cryptosystems. Ideal lattices allow to represent a lattice using only two polynomials. Using such lattices, classic lattice based cryptosystems can diminish their space complexity from $\tilde{\mathcal{O}}(d^2)$ to $\tilde{\mathcal{O}}(d)$. Ideal lattices also allow to accelerate computations using the polynomial structure. To improve even more the efficiency, some cryptosystems used a subclass of ideal lattices, namely principal ideal lattices. An ideal lattice $\mathcal{L}$ is a principal ideal lattice if there exist two integers $\alpha, \det \in \mathbb{N}$ such that

$$\mathcal{L} = \{(f_0, f_1, \ldots, f_{d-1}), f(\alpha) = 0 \bmod \det\}\,.$$

If the ideal lattice subclass has collapsing density in the class of lattices [BL09], however, the principal ideal lattice subclass keeps a reasonable density in the class of ideal lattices. If ideal lattices were introduced to improve space and time complexity of cryptosystems, their advantage allows a wilder utilization. As an example, they have allowed Gentry to create the first fully homomorphic encryption scheme [Gen09,SV10]. The choice of the monic polynomial g is public and generally free. Moreover, for most cryptosystems, it is advised to use $x^d + 1$ with d a power of two or $x^d - 1$. Those polynomials allow an even faster computation using FFT techniques for

polynomial multiplication. It is important to notice here that $x^d + 1$ (with d a power of two) is equal $\Phi_{2d}(x)$ (the cyclotomic polynomial of index $2d$) and that $x^d - 1$ is by definition factorisable in cyclotomic polynomials.

1.3 Related Work

There is one approach known to us where algorithms make use of the structure of ideal lattices. Micciancio and Voulgaris shortly explain how to exploit the ideal structure for their Sieving algorithms [MV10b]. Experiments with this approach where presented in [Sch11]. They show a speedup factor of d, where d is the degree of the ring polynomial, for runtime as well as for storage. Since in practice, sieving algorithms (even when exploiting the ideal structure) are weaker than Extreme Pruning Enumeration, this approach does not affect the practical hardness of SVP in ideal lattices.

Now we shortly introduce the two existing lattice challenges for regular lattices. These challenges can later be used to compare if lattice algorithms behave differently in ideal lattices.

In 2008 the team at TU Darmstadt introduced the Lattice Challenge[3] [BLR08]. The website offers the possibility to download one lattice every 25th dimension. The goal of the participant is to find a vector of Euclidean length smaller than a specified bound q, i.e., solve α-SVP for certain α. The online hall of fame shows the submitted results. To date, it is possible to solve this challenge in dimension up to 825. Here, participants make use of the so-called *sublattice-attack*, which allows to reduce the dimension of the lattice from, e.g., 825 to 240. We deal with this attack in Appendix B.

In 2010 Nicolas Gama and Michael Schneider presented a second challenge, namely the SVP Challenge[4]. The goal of this challenge is to find vectors in so-called random (Goldstein-Mayer) lattices [GM03] that are very close to a shortest vector in the lattice. Since the length of a shortest vector is not known, only heuristic estimation of the length is possible. The results of this challenge range up to dimension 120. The goal norm of this challenge is computed exactly the same way as in the Ideal Lattice Challenge, that

[3] `www.latticechallenge.org`
[4] `www.latticechallenge.org/svp-challenge`

we will introduce in the next section. This allows for a fair comparison between both challenges. Also, for the SVP and Ideal Lattice Challenges it is possible to download a lattice generator and create multiple lattices in the same dimension, which allows for probabilistic algorithms to show their strength.

2 The Ideal Lattice Challenge

Here we introduce the new challenge. We first explain how to participate in the challenge. Second, we show how we generate the ideal lattices. We hope that this will be a standard way for instantiating ideal lattices in the future.

Our new Ideal Lattice Challenge can be found online[5]. Basically there are two hall of fames for the same lattices: one SVP hall of fame and one for the approximate version α-SVP.

The way to participate in the SVP version of the Ideal Lattice Challenge is the following:

1. download the basis $\mathbf{B}$ of an ideal lattice $\mathcal{L}(\mathbf{B})$ from the website
2. find a non-zero lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B}) \setminus \{\mathbf{0}\}$ of Euclidean norm

$$\|\mathbf{v}\| \leq 1.05 \cdot GH(\mathcal{L}) \approx \lambda_1(\mathcal{L}(\mathbf{B}))$$

3. submit the vector via online submission system
4. enter the SVP hall of fame if your vector is the shortest in dimension d.

The way to participate in the α-SVP version of the Ideal Lattice Challenge is to submit a longer vector:

1. download the basis $\mathbf{B}$ of an ideal lattice $\mathcal{L}(\mathbf{B})$ from the website
2. find a non-zero lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B}) \setminus \{\mathbf{0}\}$ of Euclidean norm

$$\|\mathbf{v}\| \leq d \cdot \det(\mathcal{L}(\mathbf{B}))^{1/d}$$

3. submit the vector via online submission system
4. enter the α-SVP hall of fame if your vector is the shortest in dimension d.

[5] `www.latticechallenge.org/ideallattice-challenge/`

The website also allows to download the generator for the ideal lattices in order to generate multiple lattices in each dimension with different random seed s. Entering the hall of fame is possible with vectors of any lattice in the corresponding dimension. On the website, we publish the lattices with seed $s = 0$. Figure 1 shows the norms and the root Hermite factors that have to be reached to enter either of the hall of fames. For the norm computation we assume a determinant of exactly 2^{10d}. The root Hermite factors are computed as

$$\delta = (1.05 \cdot \frac{\Gamma(d/2+1)^{1/d}}{\sqrt{\pi}})^{1/d}$$

for the SVP hall of fame and $\delta = d^{1/d}$ for the α-SVP hall of fame.

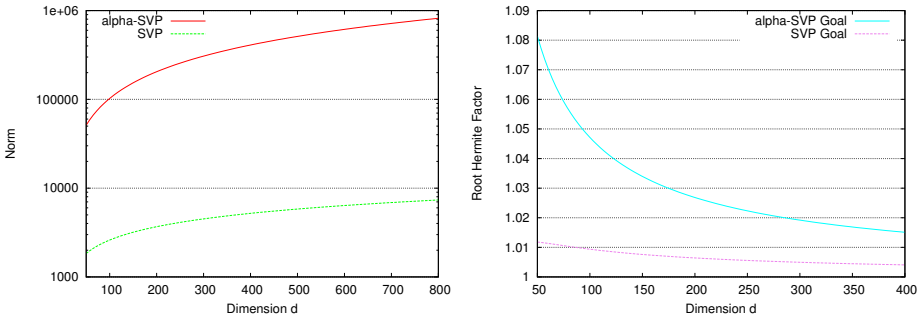

Fig. 1. Norm and root Hermite factor required to enter either of the hall of fames.

Verification. The ideal lattices have to be generated again on the server during the verification process. This might take up to a few minutes on the server. The step taking most of the time is the computation of the determinant of the ideal lattice.

2.1 The Generator For Ideal Lattices

Algorithm 1 shows the generator routine for ideal lattices for our challenge. Input to the generator is the index n of the cyclotomic polynomial $\Phi_n(x)$ and a random seed s. The index n determines the lattice dimension. More precisely, the dimension d of the lattice is the

Algorithm 1: Generator of Ideal Lattice Challenge

Input: Index n, Random Seed s
Output: Basis $\mathbf{B}$ for ideal lattice in dimension d

let $\Phi_n(x)$ be the n-th cyclotomic polynomial
let $d \leftarrow deg(\Phi_n(x))$
det $\leftarrow rand_s(2^{10d})$ ▷ *use seed s as randomness*
det $\leftarrow$ det $-(\text{det} - 1 \mod n)$
repeat
  det $\leftarrow$ det $+n$
until det *is prime*
let $g \leftarrow 1$
repeat
  $g \leftarrow g + 1$
  $\alpha \leftarrow g^{(det-1)/n}$
until α *is a root of unity of* $\Phi_n(x)$
let $\mathbf{B} =$

$$\begin{pmatrix} det & & & \\ -\alpha & 1 & & \\ -\alpha^2 & & \ddots & \\ \vdots & & & \ddots \\ -\alpha^d & & & 1 \end{pmatrix}$$

return B

degree of $\Phi_n(x)$. The random seed allows to create multiple lattices for each dimension. An example lattice can be found in Appendix A.

The determinant of the challenge ideal lattices is chosen to be of size approximately 2^{10d}, in order to be comparable to the SVP challenge. Choosing a prime determinant simplifies finding the root of the polynomial, since the factorization of the determinant is required for this. The generator picks x randomly such that $2^{10d-1} \leq x < 2^{10d}$. Then it searches for the smallest prime det such that det $\geq x$ and $(\text{det} - 1) \mid n$.

We choose cyclotomic polynomials since they cover a range of polynomials, including the ones used mosty in cryptography ($x^n + 1$ for n power of 2). This variety will give insight to the question if different types of polynomials allow for faster computations.

One interesting fact to note is that some of the indices n of the cyclotomic polynomial $\Phi_n(x)$ will lead to the same dimension d. Therefore we derive more than one challenge for some of the dimensions and there is no challenge for others.

3 Experiments

We ran a few test experiments in order to assess the hardness of the Ideal Lattice Challenge. For the SVP hall of fame, we expect very similar results to that of the SVP Challenge. This is due to the fact that the determinant of the lattices is of the same order of magnitude $\det(\mathcal{L}) \approx 2^{10d}$. Therefore one will directly be able to distinguish between SVP in ideal lattice and SVP in regular (random) lattices. Since the determinant of the lattice challenge is a bit smaller, but more important, the sublattice attack works in these lattices, we expect a difference between the results of the Lattice Challenge and the α-SVP hall of fame of the Ideal Lattice Challenge.

Recall that there are multiple lattices in some dimensions that belong to different cyclotomic polynomials.

3.1 Experiments for SVP

We applied multicore Pruned Enumeration from [KSD+11] with the polynomial bounding function and a BKZ pre-reduction blocksize of $\beta = 35$. For our experiments we use 8 AMD Opteron (2.3GHz) CPU cores. Figure 2 shows the results. Since Extreme Pruning Enumeration works probabilistic, we started the experiments with norm bound $1.06 \cdot GH(\mathcal{L})$. Otherwise, when choosing the challenge goal norm of $1.05 \cdot GH(\mathcal{L})$, the probabilistic algorithm fails in finding a vector below the bound too often. In dimension 64, in the lattice with index 128, we did not succeed to find a vector below $1.06 \cdot GH(\mathcal{L})$.

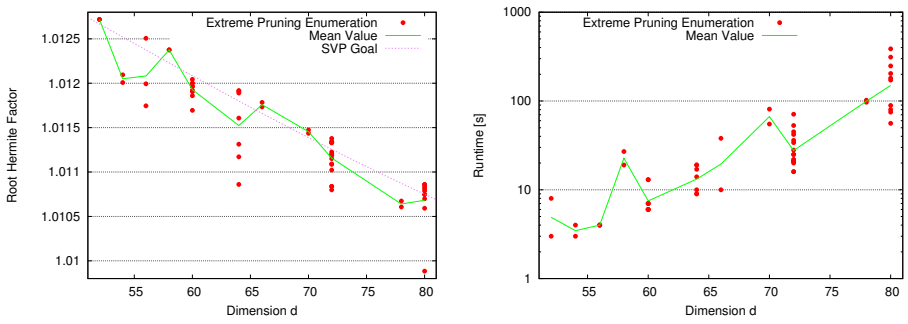

Fig. 2. Root Hermite factor and runtime of Extreme Pruning Enumeration using 8 CPU cores, including the corresponding mean values (solid green line). For the root Hermite factor, the figure also includes the goal value to enter the SVP hall of fame.

The results of Figure 2 show that it is easily possible to enter the SVP hall of fame until dimension at least 80, which on could expect from the SVP Challenge.

We also ran full Enumeration and BKZ with different blocksize parameters $\beta \in \{20, 30\}$ by the fpLLL implementation. Figure 3 shows the results. We use only one lattice per dimension, in order to make experiments in higher dimension possible.

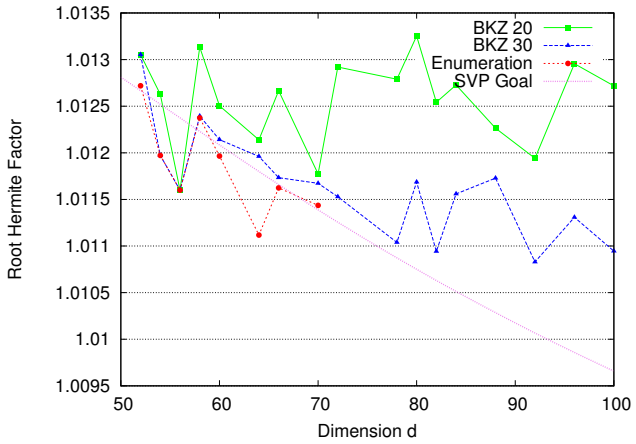

Fig. 3. Root Hermite factors reached by full Enumeration, and by BKZ with different blocksizes. The figure also includes the value that has to be reached to enter the SVP hall of fame of the Ideal Lattice Challenge.

Figure 3 shows that full Enumeration can enter the hall of fame until dimensions around 70. BKZ-20 and even BKZ-30 only reach the necessary SVP bound for very small dimensions $d < 58$.

3.2 Experiments for α-SVP

We tested the ideal lattices of our challenge in medium dimensions $d < 300$ using the NTL implementation of BKZ (using the sage interface) on a single AMD Opteron processor (2.3GHz). We fixed a blocksize $\beta \in \{2, 10, 20, 30\}$ and computed the resulting root Hermite factor, and measured the runtime. Figure 4 shows the results of our BKZ experiments.

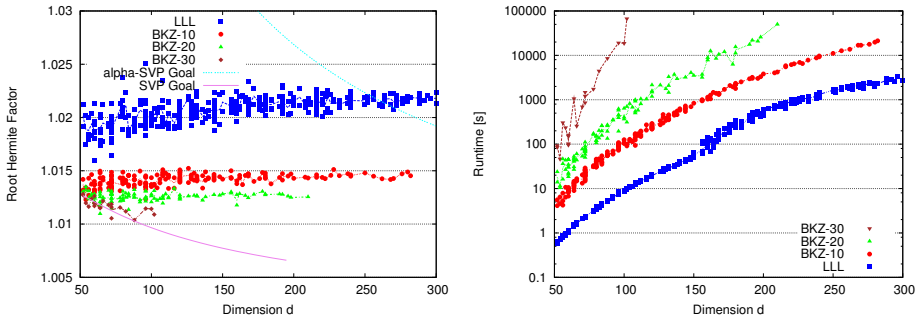

Fig. 4. Root Hermite factor and runtime of BKZ, including lines for the corresponding mean values. For the root Hermite factor, the figure also includes the value that has to be reached to enter both hall of fames of the Ideal Lattice Challenge.

With LLL ($\beta = 2$), it is possible to enter the α-SVP in dimensions up to around 250 (falling below the cyan-colored, dotted line). With BKZ-10 and BKZ-20, even higher dimensions are easy to reach. Our experiments do not exceed the point where BKZ-10 ad BKZ-20 cannot find a vector that is sufficient for the α-SVP hall of fame.

The Gaussian Heuristic. One interesting fact that we encountered during our tests is that in some dimensions, e.g. dimension 52, there is no vector below the SVP-bound (computed using the Gaussian heuristic). To our knowledge this has not happened for regular lattices in the SVP Challenge. To be more precise, the SVP Challenge hall of fame contains a vector below the GH-bound in the lattice with random seed 0 for nearly all dimensions $50 \leq d \leq 116$ (except 66, 84 and 88). It is possible that the Gaussian heuristic, which so far showed to be very accurate for regular lattices, does not hold for ideal lattices. The Gaussian heuristic is used for estimations on the runtime of cryptanalytic algorithms, like enumeration [GNR10] or BKZ 2.0 [CN11], as well as for estimation of the security of lattice-based cryptosystems. Therefore it is very important to clarify if it is also a good heuristic for ideal lattices. Appendix C presents some first experiments on this.

References

[AKS01] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *33rd Annual ACM Symposium on Theory of Computing*, pages 601–610, Crete, Greece, July 6–8, 2001. ACM Press.

[BBD08] Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors. *Post-Quantum Cryptography*. Springer, 2008.

[BL09] Johannes Buchmann and Richard Lindner. Density of ideal lattices. In Johannes A. Buchmann, John Cremona, and Michael E. Pohst, editors, *Algorithms and Number Theory*, number 09221 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[BLR08] Johannes Buchmann, Richard Lindner, and Markus Rückert. Explicit hard instances of the shortest vector problem. In Johannes Buchmann and Jintai Ding, editors, *PQCrypto*, volume 5299 of *Lecture Notes in Computer Science*, pages 79–94. Springer, 2008.

[CN11] Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20, Seoul, South Korea, December 4–8, 2011. Springer, Berlin, Germany.

[FP85] U. Fincke and Michael Pohst. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Mathematics of Computation*, 44(170):463–471, 1985.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 169–178, Bethesda, Maryland, USA, May 31 – June 2, 2009. ACM Press.

[GM03] Daniel Goldstein and Andrew Mayer. On the equidistribution of Hecke points. *Forum Mathematicum 2003, 15:2*, pages 165–189, 2003.

[GN08] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 31–51, Istanbul, Turkey, April 13–17, 2008. Springer, Berlin, Germany.

[GNR10] Nicolas Gama, Phong Q. Nguyen, and Oded Regev. Lattice enumeration using extreme pruning. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 257–278, French Riviera, May 30 – June 3, 2010. Springer, Berlin, Germany.

[HPS11] Guillaume Hanrot, Xavier Pujol, and Damien Stehlé. Algorithms for the shortest and closest lattice vector problems. In *IWCC 2011*, volume 6639 of *Lecture Notes in Computer Science*, pages 159–190. Springer, 2011.

[Kan83] Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In *STOC 1983*, pages 193–206. ACM, 1983.

[KSD+11] Po-Chun Kuo, Michael Schneider, Özgür Dagdelen, Jan Reichelt, Johannes Buchmann, Chen-Mou Cheng, and Bo-Yin Yang. Extreme enumeration on GPU and in clouds - - how many dollars you need to break SVP challenges -. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2011*, volume 6917 of *Lecture Notes in Computer Science*, pages 176–191, Nara, Japan, September 28 – October 1, 2011. Springer, Berlin, Germany.

[LLL82] Arjen Lenstra, Hendrik Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 4:515–534, 1982.

[Mic07] Daniele Micciancio. Generalized compact knapsaks, cyclic lattices and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.

[MR08] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In Bernstein et al. [BBD08], pages 147–191.

[MV10a] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In Leonard J. Schulman, editor, *42nd Annual ACM Symposium on Theory of Computing*, pages 351–358, Cambridge, Massachusetts, USA, June 5–8, 2010. ACM Press.

[MV10b] Daniele Micciancio and Panagiotis Voulgaris. Faster exponential time algorithms for the shortest vector problem. In Moses Charika, editor, *21st Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1468–1480, Austin, Texas, USA, January 17–19, 2010. ACM-SIAM.

[NV08] Phong Q. Nguyen and Thomas Vidick. Sieve algorithms for the shortest vector problem are practical. *J. of Mathematical Cryptology*, 2(2), 2008.

[NV10] Phong Q. Nguyen and Brigitte Vallée, editors. *The LLL Algorithm - Survey and Applications*. Springer, 2010.

[Sch11] Michael Schneider. *Computing Shortest Lattice Vectors on Special Hardware*. PhD thesis, Technische Universität Darmstadt, November 2011.

[SE94] Claus-Peter Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66:181–199, 1994.

[SV10] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography*, pages 420–443, 2010.

A Example Challenge

Here we present a sample for the ideal lattice challenge in dimension 4. The files have the following syntax:

```
[[1046226146011 0 0 0]
[857976943568 1 0 0]
[397363584747 0 1 0]
[224465523046 0 0 1]
]
4 5 0
[1 1 1 1 1]
```

The first rows contain the basis of the ideal lattice in Hermite normal form, in NTL format as row vectors. The upper left vector is the determinant, the entry below is the negative root of unity α. The

next row stores the dimension, the index of the cyclotomic polynomial and the random seed (in this order). The last row encodes the cyclotomic polynomial, with constant term first. E.g., the polynomial $x^4 + x$ is encoded [0 1 0 0 1]. Our example polynomial [1 1 1 1 1] encodes the polynomial $\Phi_5(x) = x^4 + x^3 + x^2 + x + 1$.

B Sublattice-Attack and Hermite Factors

The sublattice attack was first presented in [MR08]. For q-ary type lattices, it allows to lower the dimension for finding vectors of the same length. This results in reduced runtime and was excessively used in the lattice challenge. Here we show that the common sublattice attack is not applicable for the ideal lattice challenge.

We want to show that the sublattice dimension $\tilde{d}$ is bigger or equal to the lattice dimension d. For our lattices, the best sublattice dimension can be computed as

$$\tilde{d} = \sqrt{\log_2(det)/\log_2(\delta)}\,,$$

where δ is the root Hermite factor. We want to be sure that

$$d < \tilde{d} = \sqrt{\log_2(det)/\log_2(\delta)}\,.$$

Using that $\det(\mathcal{L}) \approx 2^{10d}$ we derive that

$$\delta < 2^{10/d} \tag{2}$$

should be satisfied in order to prohibit a sublattice attack. If we choose as upper bound to enter the *alpha*-SVP hall of fame the norm $d \cdot \det(\mathcal{L})^{1/d}$, then (2) is satisfied for all dimensions $d \lesssim 1050$.

Figure 5 shows the bound (2) for different lattice dimension d as solid red line. The dotted blue line is a lower bound on what lattice reduction algorithms can achieve today ($\delta = 1.009$). We gained this data from experience with the existing lattice challenges. The dashed green line in Figure 5 shows the δ-value that has to be reached to enter the hall of fame for α-SVP. So far it should not be possible to solve challenges above the intersection point around dimension $d = 737$.

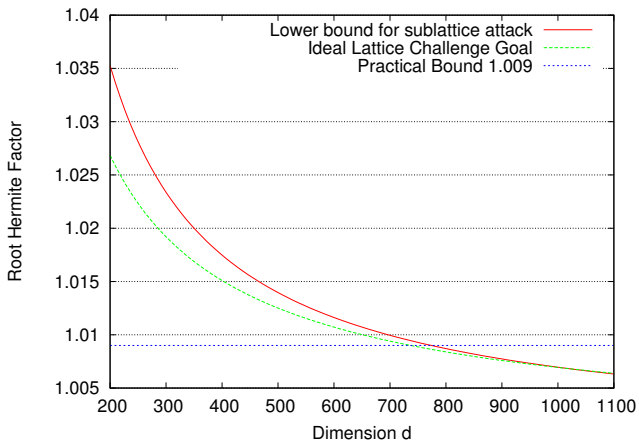

Fig. 5. Lower bound for sublattice attack and goal value to enter the α-SVP hall of fame. It should not be possible to solve challenges above $d = 737$ with today's lattice reduction algorithms. For dimensions $d \lesssim 1050$ a sublattice attack is unlikely to yield to an improvement in reduction quality.

C Ideal Lattices and the Gaussian Heuristic

In order to shed a bit more light on the question whether the Gaussian heuristic holds we performed experiments with different random seeds $0 \leq s < 20$ for all existing challenge lattices. Figure 6 shows the results. Most of the lattices contain shortest vectors that are below the SVP goal value, and so does the mean value of all lattices (over all seeds and all polynomials in the same dimension). Apparently the Gaussian heuristic gets better in higher dimensions. Still a full comparison with regular lattices remains open for future work.

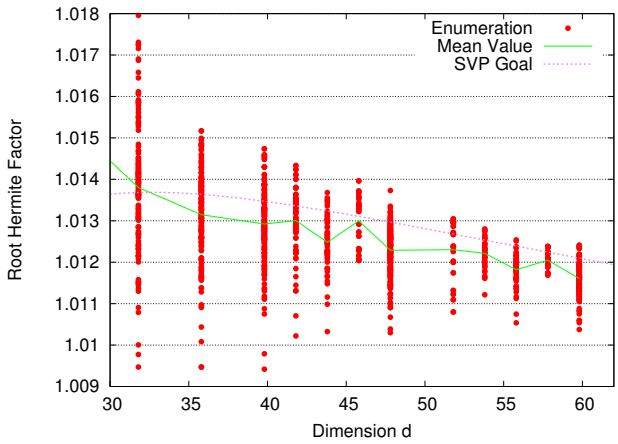

Fig. 6. Root Hermite factor reached by lattices with random seed $0 \leq s < 20$ and the corresponding mean values (solid green line). The figure also includes the value that has to be reached to enter the SVP hall of fame of the Ideal Lattice Challenge.