J. Buchmann and J. Ding (Eds.): PQCrypto 2008, LNCS 5299, pp. 79–94, 2008. © Springer-Verlag Berlin Heidelberg 2008

Explicit hard instances of the shortest vector problem

– Revised Version –

Johannes Buchmann, Richard Lindner, Markus Rückert, and Michael Schneider

Technische Universität Darmstadt, Department of Computer Science
Hochschulstraße 10, 64289 Darmstadt, Germany
buchmann,rlindner,rueckert,mischnei@cdc.informatik.tu-darmstadt.de

Abstract. Building upon a famous result due to Ajtai, we propose a sequence of lattice bases with growing dimension, which can be expected to be hard instances of the shortest vector problem (SVP) and which can therefore be used to benchmark lattice reduction algorithms.

The SVP is the basis of security for potentially post-quantum cryptosystems. We use our sequence of lattice bases to create a challenge, which may be helpful in determining appropriate parameters for these schemes.

Keywords: Lattice reduction, lattice-based cryptography, challenge

1 Introduction

For the construction of post-quantum cryptosystems, it is necessary to identify computational problems, whose difficulty can be used as a basis of the security for such systems, and that remain difficult even in the presence of quantum computers. One candidate is the problem of approximating short vectors in a lattice (shortest vector problem — SVP). The quantum-hardness of this problem was analyzed by Ludwig [27] and Regev [36]. They both find that the computational advantage gained with quantum computers is marginal. There are several cryptographic schemes whose security is based on the intractability of the SVP in lattices of sufficiently large dimension (e.g. [19,20,3,37]). To determine appropriate parameters for these cryptosystems, it is necessary to assess the practical difficulty of this problem as precisely as possible.

In this paper, we present a sequence of lattice bases with increasing dimension, which we propose as a world wide challenge. The construction of these lattices is based both on theoretical and on practical considerations. On the theoretical side, we apply a result of Ajtai [2]. It states that being able to find a sufficiently short vector in a random lattice from a certain set, which also contains our challenge lattices, implies the ability to solve supposedly hard problems (cf. [38]) in all lattices with a slightly smaller dimension than that of the random lattice. Furthermore, we invoke the pigeon hole principle, which guarantees the existence of a short trinary vector in each challenge lattice. On the practical side, using an analysis by Gama and Nguyen [16], we argue that finding this vector is hard for the lattices in our challenge. We also present first experimental results that confirm the analysis.

Our challenge at http://www.latticechallenge.org can be considered as an analogue of similar challenges for the integer factoring problem [39] and the problems of computing discrete logarithms in the multiplicative group of a finite field [30], or in the group of points on an elliptic curve over a finite field [11].

Our aim is to evaluate the current state-of-the-art in practical lattice basis reduction by providing means for an immediate and well-founded comparison. As a first application of the proposed challenge, we compare the performance of LLL-type reduction methods — LLL [26], Nguyen and Stehlé's fpLLL [33], Koy and Schnorr's segment LLL (sLLL) [24] — and block-type algorithms —

Schnorr's BKZ [41,40], Koy's primal-dual (PD) [23], Ludwig's practical random sampling [1] (PSR) [28]. To our knowledge, this is the first comparison of these algorithms.

Related work. Lattice reduction has been subject to intense studies over the last decades, where a couple of methods and reduction schemes, in particular the LLL algorithm by Lenstra, Lenstra, and Lovász [26], have been developed and successively improved. Especially, the block Korkine Zolotarev algorithm (BKZ), due to Schnorr [40,41], has become the standard method when strong lattice basis reduction is required. In theory, the best algorithm for finding short vectors is the block algorithm *slide reduction* presented in [15]. Compared to other algorithms like Schnorr's, the asymptotical complexity remains the same, the asymptotical approximation factor achievable remains $2^{\mathcal{O}(n \log \log n / \log n)}$, while the constants are lowered using slide reduction.

There have been several approaches to measure the effectiveness of known lattice reduction algorithms, especially in the context of the NTRU cryptosystem [20]. Some of them, as in [21,22], base their analysis on cryptosystems while others, like [34,16], make a more general approach using random lattices.

To our knowledge, there has never been a unified challenge, one that is independent of a specific cryptosystem, for lattice reduction algorithms. In all previous challenges, the solution was always known to the creator. There exist some classes of lattices, upon which you could build a challenge, e.g. the random lattices described in [13]. The authors fix a number N and choose one of the lattices with dimension n and determinant N. For implementation issues it is common to use a prime N. It can be shown that these lattices are uniformly distributed in the set of all modulo lattices of dimension n with respect to the natural probability measure on this set. This construction leads to lattices with short vectors of approximate length $\mathrm{vol}(L)^{1/n}\sqrt{n/(2\pi e)}$ [16]. However, for a challenge it is more application oriented to use modular lattices since most lattice-based cryptographic schemes work in modular lattices, e.g. the SWIFFT hash function [29], the GPV signature scheme [17], or the NTRU cryptosystem [20].

Organization. In Section 2, we provide a brief introduction to lattices and state some fundamental definitions. In Section 3, we define a family of lattices and prove two properties, which are fundamental for our explicit construction presented in Section 4. Then, we give first experimental results comparing the performance of various lattice reduction algorithms in Section 5. Finally, Section 6 introduces the actual lattice challenge.

2 Preliminaries

Let $\mathbb{R}^n$ denote the n-dimensional real vectorspace. We write the vectors of this space in boldface to distinguish them from numbers. Any two vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$ have an inner product $\langle \mathbf{v}, \mathbf{w} \rangle = \mathbf{v}^T \mathbf{w}$. Any $\mathbf{v} \in \mathbb{R}^n$ has a length given by the Euclidean norm $\|\mathbf{v}\|_2 = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle} = \sqrt{v_1^2 + \cdots + v_n^2}$. In addition to the Euclidean norm, we also use the maximum norm $\|\mathbf{v}\|_\infty = \max_{i=1,\ldots,n}\{|v_i|\}$.

A lattice in $\mathbb{R}^n$ is a set $L = \{\sum_{i=1}^m x_i \, \mathbf{b}_i \,|\, x_i \in \mathbb{Z}\}$, where $\mathbf{b}_1, \ldots, \mathbf{b}_m$ are linearly independent over $\mathbb{R}$. The matrix $B = [\mathbf{b}_1, \ldots, \mathbf{b}_m]$ is called a *basis* of the lattice L and we write $L = L(B)$. The number of linearly independent vectors in the basis is the dimension of the lattice. If $\dim(L(B)) = n$ the lattice is full-dimensional.

An m-dimensional lattice $L = L(B)$ has many different bases, namely all the matrices in the orbit $B\,\mathrm{GL}_m(\mathbb{Z}) = \{BT \,|\, T \in \mathrm{GL}_m(\mathbb{Z})\}$. If the lattice is full-dimensional and integral, that is $L \subseteq \mathbb{Z}^n$, then there exists a unique basis $B = (b_{i,j})$ of L, which is in Hermite normal form (HNF), i.e.

i. $b_{i,j} = 0$ for all $1 \leq j < i \leq m$
ii. $b_{i,i} > b_{i,j} \geq 0$ for all $1 \leq i < j \leq m$

Furthermore, the volume $\mathrm{vol}(L)$ of a full-dimensional lattice is defined as $|\det(B)|$, for any basis B of L. For every m-dimensional lattice L there is a dual (or polar, reciprocal) lattice $L^* = \{\mathbf{x} \in \mathbb{R}^m \,|\, \forall \mathbf{y} \in L : \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}\}$. For any full-dimensional lattice $L = L(B)$, it holds that

[1] A practical variant of Schnorr's random sampling reduction [42].

$L^* = L((B^{-1})^T)$. The length of the shortest lattice vector, denoted with $\lambda_1 = \lambda_1(L)$, is called first successive minimum.

3 Foundations of the challenge

In this section, we define a family of sets containing lattices, where each set will have two important properties:

1. All lattices in the set contain non-obvious short vectors;
2. Being able to find a short vector in a lattice chosen uniformly at random from the set, implies being able to solve difficult computational problems in all lattices of a certain smaller dimension.

The family of lattice sets. Let $n \in \mathbb{N}$, $n \geq 50$, $c_1, c_2 \in \mathbb{R}_{>0}$, such that

$$c_1 \geq 2.1 \quad \text{and} \quad c_2 \leq c_1 \ln(2) - \frac{\ln(2)}{50 \ln(50)} \tag{1}$$

Furthermore, let

$$m = \lfloor c_1 n \ln(n) \rfloor \ , \tag{2}$$
$$q = \lfloor n^{c_2} \rfloor \ , \tag{3}$$

and $\mathbb{Z}_q = \{0, \ldots, q-1\}$. For a matrix $X \in \mathbb{Z}_q^{n \times m}$, with column vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$, let

$$L(c_1, c_2, n, X) = \left\{ (v_1, \ldots, v_m) \in \mathbb{Z}^m \;\middle|\; \sum_{i=1}^{m} \mathbf{x}_i v_i \equiv \mathbf{0} \pmod{q} \right\} .$$

All lattices in the set $L(c_1, c_2, n, \cdot) = \{L(c_1, c_2, n, X) | X \in \mathbb{Z}_q^{n \times m}\}$ are of dimension m and the family of lattices $\mathfrak{L}$ is the set of all $L(c_1, c_2, n, \cdot)$, such that c_1, c_2, n are chosen according to (1).

In the following theorems, we prove that all lattices in the sets of the family $\mathfrak{L}$ have the desired properties.

Existence of short vectors. We prove that all lattices in $L(c_1, c_2, n, \cdot)$ of the family $\mathfrak{L}$ contain a vector with Euclidean norm less than $\sqrt{m}$.

Theorem 1. *Let $n \in \mathbb{N}, n \geq 50, c_1, c_2 \in \mathbb{R}_{>0}$, and $q, m \in \mathbb{N}$ be as described above. Then, any lattice in $L(c_1, c_2, n, \cdot) \in \mathfrak{L}$ contains a vector with Euclidean norm less than $\sqrt{m}$.*

Proof. Let $L(c_1, c_2, n, X) \in L(c_1, c_2, n, \cdot) \in \mathfrak{L}$. We use the pigeon hole principle in order to show that $L(c_1, c_2, n, X)$ contains a trinary vector of length less than or equal to $\sqrt{m}$.

Consider the set of all vectors $\mathbf{z} \in \{0, 1\}^m$. Obviously, it contains 2^m vectors. By the choice of c_1, c_2, m, and q above, we have $2^m > q^n$ because

$$\begin{aligned} n \ln(q) &\leq c_2\, n \ln(n) \\ &\overset{*}{<} \ln(2) \lfloor c_1\, n \ln(n) \rfloor \\ &= m \ln(2) \,. \end{aligned}$$

For a proof of inequality $*$, we refer the reader to Appendix A. In consequence, there is a collision, i.e. two *distinct* vectors $\mathbf{z}, \mathbf{z}' \in \{0, 1\}^m$ with

$$\sum_{i=1}^{m} \mathbf{x}_i\, z_i \equiv \sum_{i=1}^{m} \mathbf{x}_i\, z'_i \pmod{q}$$

and therefore $\mathbf{z} - \mathbf{z}' \in L(c_1, c_2, n, X)$. The vector $\mathbf{z} - \mathbf{z}'$, however, is at least as short as $\sqrt{m}$ in the Euclidean norm because

$$\|\mathbf{z} - \mathbf{z}'\|_2 \leq \sqrt{m}\; \|\mathbf{z} - \mathbf{z}'\|_\infty = \sqrt{m} \,.$$

□

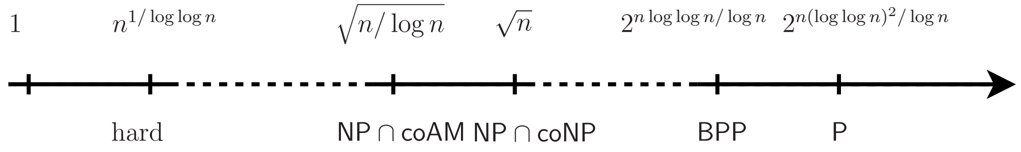


Fig. 1. The complexity of γ-SVP for increasing γ (some constants omitted).

Hardness of finding short vectors. In the following, we show that being able to find short vectors in an m-dimensional lattice chosen uniformly at random from $L(c_1, c_2, n, \cdot) \in \mathfrak{L}$, implies being able to solve (conjectured) hard lattice problems for *all* lattices of dimension n.

In his seminal work [2], Ajtai proved the following theorem that connects average-case instances of certain lattice problems to worst-case instances. The problems are defined as follows.

Lattice problems. Let $L \subseteq \mathbb{Z}^n$ be an n-dimensional lattice and $\gamma \geq 1$. We define the

- Approximate shortest length problem (γ-SLP):
  Find $l \in \mathbb{R}$, such that $l \leq \lambda_1(L) \leq \gamma\, l$.
- Approximate shortest vector problem (γ-SVP):
  Find a vector $\mathbf{v} \in L \setminus \{\mathbf{0}\}$, such that for all $\mathbf{w} \in L$: $\|\mathbf{v}\|_2 \leq \gamma \|\mathbf{w}\|_2$.
- Approximate shortest basis problem (γ-SBP):
  Find a basis B of L, such that for all $C \in B\,\mathrm{GL}_m(\mathbb{Z})$:

$$\max_{i=1,2,\ldots,n} \|\mathbf{b}_i\|_2 \leq \gamma \max_{i=1,2,\ldots,n} \|\mathbf{c}_i\|_2 \;.$$

Theorem 2 ([2, Theorem 1]). *Let $c > 1$ be an absolute constant. If there exists a probabilistic polynomial time (in n) algorithm $\mathcal{A}$ that finds a vector of Euclidean norm at most $\beta(n) = \sqrt{m}$ in a random m-dimensional lattice from $L(c_1, c_2, n, \cdot) \in \mathfrak{L}$ with probability $\geq 1/2$ then there exists*

1. *an algorithm $\mathcal{B}_1$ that solves the γ-SLP;*
2. *an algorithm $\mathcal{B}_2$ that solves the SVP, provided that the shortest vector is γ-unique* [2]*;*
3. *an algorithm $\mathcal{B}_3$ that solves the γ-SBP.*

Algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ solve the respective problem (each with $\gamma = n^c$) with probability exponentially close to 1 in all *lattices of dimension n, i.e. especially in the worst-case. $\mathcal{B}_1, \mathcal{B}_2$, and $\mathcal{B}_3$ run in probabilistic polynomial time in n.*

As for the constant c in Theorem 2, there have been several improvements to Ajtai's reduction with $c \geq 8$ [10]. The first improvement ($c = 3.5 + \epsilon$) is due to Cai and Nerurkar [10], whereas the most recent works by Micciancio [31] and Micciancio and Regev [32], improve c to almost [3] 1.

Following an improved worst-case to average-case reduction by Gentry, Peikert, and Vaikuntanathan [17], we argue that Theorem 2 holds for our choice of parameters. Their reduction demands that for $\beta(n) = \sqrt{m}$, $q = \lfloor n^{c_2} \rfloor$ has to grow at least slightly faster than $\sqrt{2.1}\, n \ln(n)$. More precisely,

$$\lfloor n^{c_2} \rfloor \geq \sqrt{m}\, \omega(\sqrt{n \ln(n)}) \text{ , i.e. } \lfloor n^{c_2} \rfloor \geq \sqrt{c_1}\, n \ln(n)\, (\sqrt{n \ln(n)})^{\epsilon} \qquad (\epsilon > 0)\,.$$

With our choice of c_2, we satisfy this lower bound for all relevant $n \geq 50$. The resulting approximation factor γ is $\tilde{O}(n)$.

Asymptotic and practical hardness of the above problems depends on the choice of γ. A recent survey [38] by Regev states the currently known "approximability" and "inapproximability" results. As for the complexity of lattice problems, it focuses on the works of Lagarias, Lenstra, and Schnorr

[2] A shortest vector $\mathbf{v} \in L$ is γ-*unique* if for all $\mathbf{w} \in L$ with $\|\mathbf{w}\|_2 \leq \gamma \|\mathbf{v}\|_2 \Rightarrow \mathbf{w} = \pm\mathbf{v}$.

[3] Omitting poly-logarithmic terms in the resulting approximation factor.

[25], Banaszczyk [7], Goldreich and Goldwasser [18], Ajtai, Kumar, and Sivakumar [4], Aharonov and Regev [1], and Peikert [35]. Since it is very helpful and descriptive, we adopted Figure 1 from the survey.

On the left, there are provably NP-hard problems, followed by a gap for which the hardness is unknown. In the center, there are problems conjectured not to be NP-hard because their NP-hardness would contradict the general perception that $\mathsf{coNP} \neq \mathsf{NP}$. Finally, on the right, there are problems that can be solved in probabilistic polynomial time.

We emphasize that the problems in Theorem 2 are *not* believed to be NP-hard because $\gamma > \sqrt{n}$. Nevertheless, there is no known algorithm that efficiently solves worst-case instances of lattice problems for sufficiently large dimensions n, with an approximation factor polynomial in n. So Theorem 2 strongly supports our claim that computing short vectors in the lattice family is hard. This is also supported by a heuristic argument of Gama and Nguyen [16], which we refer to in Section 4.

4 Construction of explicit bases

Ajtai's construction in [2] defines all lattices implicitly. In this section, we show how to generate explicit integral bases for these lattices.

For any $m \geq 500$, we now construct a lattice L_m of dimension m, which is our hard instance of the SVP. The lattice L_m is of the form $L(c_1, c_2, n, X)$, where the parameters c_1, c_2, n, X are chosen as a function of the dimension m as follows.

We start with a desired lattice dimension m, set $c_1 = 2.1$, $c_2 = c_1 \ln(2) - \ln(2)/(50 \ln(50))$, and choose $n = n(m)$ such that (2) holds, i.e. find an $n \in \mathbb{N}$ such that $\lfloor c_1\, n \ln(n) \rfloor$ is a close to m as possible. With $m = 500$, for example, we get $c_1 = 2, c_2 = 1.45207, n = 59$, and $q = 372$.

Having selected the set $L(c_1, c_2, n, \cdot)$, we "randomly" pick a lattice from it. We use the digits of π as a source of "randomness" [4]. This approach is supported by the conjectured normalcy of π in [5,6]. We write

$$3.\pi_1\, \pi_2\, \pi_3\, \pi_4\, \ldots\,,$$

so π_i, for $i \geq 1$, is the ith decimal digit of π in the expansion after the decimal point. In order to compensate for potential statistical bias, we define

$$\pi_i^* = \pi_{2\,i} + \pi_{2\,i-1} \mod 2 \qquad \text{for } i \geq 1\,.$$

Now, we use the sequence $(\pi_1^*, \pi_2^*, \pi_3^*, \pi_4^*, \ldots)$ as a substitute for a sequence of uniformly distributed random bits.

Let $\ell = 0$. The matrix $X = (x_{i,j}) \in \mathbb{Z}_q^{n \times m}$ is chosen via

$$\begin{aligned} x_{i,j} &= \sum_{l=k}^{k+\lfloor \log_2(q) \rfloor} 2^{l-k}\, \pi_l^* \qquad \text{for } 1 \leq i \leq n, 1 \leq j \leq m\,, \\ \text{with} \quad k &= k(i,j) = ((i-1)\, m + (j-1) + \ell)\, (\lfloor \log_2(q) \rfloor + 1) + 1\,, \\ \text{If} \quad & x_{i,j} \geq q \text{ recompute } x_{i,j} \text{ with } \ell = \ell + 1\,. \end{aligned}$$

With that, we have selected a "random" element $L(c_1, c_2, n, X)$, for which we will now generate an integral basis.

Let I_m be the m-dimensional identity matrix. We start with the matrix

$$Y_1 = (X^T \mid q\, I_m) = \left(\begin{array}{ccc|cccc} x_{1,1} & \cdots & x_{n,1} & q & 0 & \cdots & 0 \\ x_{1,2} & \cdots & x_{n,2} & 0 & q & & \vdots \\ \vdots & \ddots & \vdots & \vdots & & \ddots & 0 \\ x_{1,m} & \cdots & x_{n,m} & 0 & \cdots & 0 & q \end{array}\right).$$

[4] The digits of π can be obtained from `ftp://pi.super-computing.org/`.

m	n	q	δ_m
200	29	132	1.0125
250	34	167	1.0110
300	40	211	1.0095
350	45	251	1.0087
400	50	293	1.0079
450	54	327	1.0075
500	59	372	1.0069
550	64	419	1.0065
600	68	458	1.0062
650	73	507	1.0058
700	77	548	1.0055

Table 1. Lattice parameters with the necessary δ_m.

Let Y_2 be the Hermite normal form of Y_1, we compute the transformation matrix T_1, which satisfies

$$Y_2 \, T_1 = Y_1 = (X^T \mid qI_m) \, .$$

We set T_2 to be equal to T_1, but without the n leading columns. This guarantees that

$$Y_2 \, T_2 = q \, I_m \, . \tag{4}$$

Finally, we set the basis to $B = T_2^T$.

Now, we have to show that B is an integral basis of $L(c_1, c_2, n, X)$. Clearly, B is an integral matrix because the transformation T_1, given by the HNF computation, is in $\mathbb{Z}^{m \times (n+m)}$ and T_2 is the same matrix with the n leading columns removed.

By the uniqueness of inverses, (4) shows that $B = ((Y_2/q)^{-1})^T$. This implies that B is a basis for the dual lattice of $L(Y_2/q)$ (cf. Section 2). Since Y_2 is an integral transformation of Y_1, they span the same lattice. Thus, $L(Y_2/q) = L(Y_1/q)$.

By the defining property of the dual lattice, we have that for any $\mathbf{v} \in L(B)$ and $\mathbf{w} \in L(Y_1/q)$, it holds that $\langle \mathbf{v}, \mathbf{w} \rangle \in \mathbb{Z}$. So especially for all columns $\mathbf{x}$ of X^T, it holds that $\langle \mathbf{v}, \mathbf{x}/q \rangle \in \mathbb{Z}$, or equivalently $\langle \mathbf{v}, \mathbf{x} \rangle \in q\mathbb{Z}$. This implies $\langle \mathbf{v}, \mathbf{x} \rangle \bmod q = 0$, which in turn gives us $L(B) \subseteq L(c_1, c_2, n, X)$.

Now let $\mathbf{v} \in L(c_1, c_2, n, X)$, so for any column $\mathbf{x}$ of X^T we have that the inner product $\langle \mathbf{v}, \mathbf{x} \rangle \bmod q = 0$, or equivalently $\langle \mathbf{v}, \mathbf{x}/q \rangle \in \mathbb{Z}$. Since we know $L(c_1, c_2, n, X) \subseteq \mathbb{Z}^m$, it also holds that $\langle \mathbf{v}, \mathbf{e} \rangle \in \mathbb{Z}$ for any column $\mathbf{e}$ of the identity matrix I_m. Since $\mathbf{v}$ has an integral inner product with each column vector in Y_1/q, this means $\mathbf{v}$ is in the dual lattice of $L(Y_1/q)$, which we know to be $L(B)$. Finally, we have $L(B) = L(c_1, c_2, n, X)$.

For a small example of such a basis, refer to Appendix C.

The choice of parameters. We now argue that our choice of the parameters leads to m-dimensional lattices $L_m = L(c_1, c_2, n, X)$, in which vectors of norm less than $\sqrt{m}$ are hard to find.

We have chosen c_2, such that Theorem 1 guarantees the existence of lattice vectors with norm less than $\sqrt{m}$ in L_m. Then, the hardness of lattice problems in a large dimension m would be based on the worst-case hardness of lattice problems in a very small dimension n. As n decreases, our hardness argument becomes less meaningful because even worst-case lattice problems in small dimensions are believed to be easy.

Table 1 shows how m and n are related for the selected lattices L_m. For a graphical overview, up to $m = 2000$, refer to Appendix B. Thus, in order to apply Theorem 2 as a strong indication for hardness, we keep $n(m)$ close to m in the above construction. We choose a pseudo-random X to get a random element in $L(c_1, c_2, n, \cdot)$, as required by Theorem 2.

To give another argument for the hardness of SVP in our lattices, we use a result by Gama and Nguyen [16]. They analyze the hardness of finding vectors $\mathbf{v}$ in a random lattice L of dimension d

such that

$$\|\mathbf{v}\| < \delta^d \mathrm{vol}(L)^{1/d},$$

where δ is a predetermined constant (Hermite factor). They state that computing such vectors is difficult for $\delta = 1.01$ and "totally out of reach" for $\delta = 1.005$ and $d \geq 500$. The distribution from which Gama and Nguyen choose the random lattices for their analysis is different from the distribution we have described at the beginning of this section, so their statement can only be seen as an indication of hardness.

Following the analysis in [8], the shortest vector we are likely to find in in a d-dimensional sublattice of the challenge lattice L_m has length $\delta^d q^{n/d}$. Considering this length as a function of the dimension d, the minimum $\delta^{2\sqrt{n\log(q)/\log(\delta)}}$ is obtained for $d = \sqrt{n\log(q)/\log(\delta)}$. Setting the target length to $\sqrt{m}$, we can compute a δ_m, which satisfies

$$\sqrt{m} \leq \delta_m^{2\sqrt{n\log(q)/\log(\delta_m)}}.$$

Such δ_m are listed in column 3 of Table 1.

In combination with the analysis of Gama and Nguyen, the table suggests that while finding a vector shorter than $\sqrt{m}$ in L_{250} is still possible, the respective problem in L_{700} will be very hard in practice. As the dimension increases, the necessary δ_m falls below 1.005. We believe that finding short vectors in the corresponding lattices will require entirely new algorithms.

5 Experiments with lattice reduction algorithms

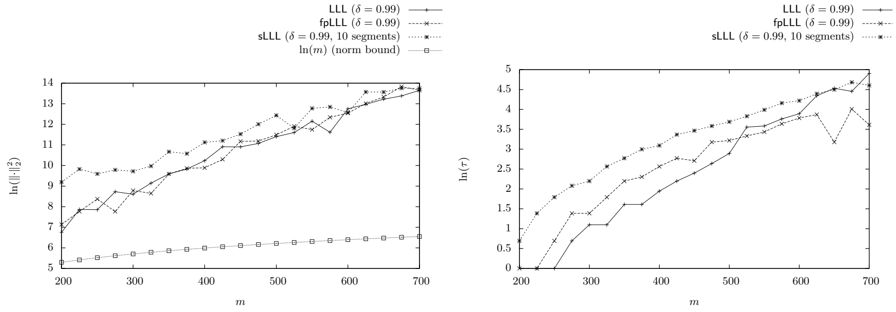
As a first application of our explicit construction of lattices L_m, we show how various lattice reduction algorithms perform on them. Basically, there are two types of algorithms: the LLL-type and the block-type. Building upon LLL, block-type algorithms are typically stronger, in the sense that they are able to find significantly shorter vectors. Block-type algorithms, however, are impractical for large block sizes because their running time increases at least exponentially in this parameter. All experiments were run on a single core AMD Opteron at 2.6 GHz, using Shoup's NTL [43] in version 5.4.2 and GCC 4.3.1. .

Toy challenges. For Theorem 1 to work, we need $n(m) \geq 50$, which is why we refer to lattices L_m with $m < 400$ as toy challenges. This does not imply that solving the challenge is easy in those smaller dimensions. We just cannot simply prove the existence of a sufficiently short lattice vector. In practice, however, we have seen that such vectors can be found.

Implementations. For LLL and BKZ, we used the famous floating-point implementations integrated in the NTL. We thank Filipović and Koy for making available their implementations of sLLL and PD, which were part of the diploma thesis [14]. We also thank Ludwig for making available and updating his implementation of PSR that was part of his PhD thesis [28]. Finally, we thank Cadé and Stehlé for making available their implementation of fpLLL. It was obtained from [44] in version 3.0.3.

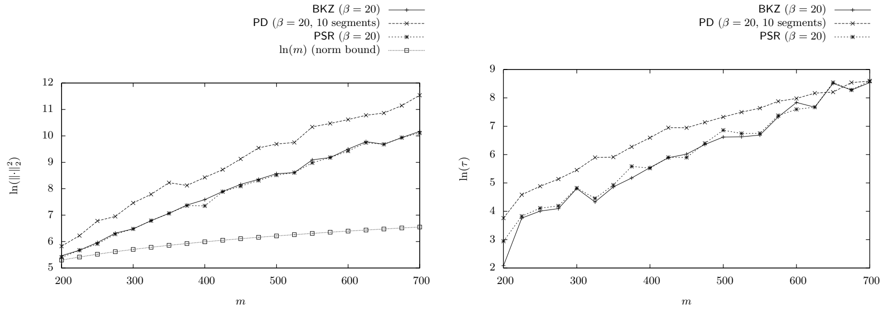
Figure 2 and Figure 3 depict the performance, i.e. the length of the shortest obtained vector and the logarithmic running time in seconds, for LLL-type and block-type methods, respectively. The boxed line in the left figures shows the norm bound $\sqrt{m}$ that has to be undercut. In contrast to our initial construction in [9], the same algorithms are not able to solve even the easiest problem in dimension $m = 200$.

While being arguably efficient with our choice of parameters, sLLL performs slightly worse than fpLLL and LLL with respect to approximation quality. This, however, is to be expected in this segment-wise algorithm. For larger dimensions, however, the approximation results of all LLL-type algorithms seem to converge. Thanks to Damien Stehlé, who pointed out the correct parameters for fpLLL, the running time performance of fpLLL significantly surpasses that of LLL and sLLL in higher dimensions. In Figure 3a, observe that BKZ and PSR perform better than PD, which is



(a) Shortest vectors (b) Running time

Fig. 2. Performance of LLL-type lattice reduction with comparable parameters.



(a) Shortest vectors (b) Running time

Fig. 3. Performance of block-type lattice reduction with comparable parameters.

mostly due to the internal sLLL step in PD. Their running time, displayed in Figure 3b, seems to converge in higher dimensions.

While the approximation performance of block-type algorithms can be further improved using larger block sizes, this approach is limited by the resulting running time. Extrapolating to higher dimensions, it becomes obvious that finding sufficiently short vectors in L_m requires a significantly larger effort. This coincides with our observation on the Hermite factor in Section 4.

To conclude, we have reviewed the current state-of-the-art performance of lattice reduction algorithms, using reasonable parameters. We did not, however, explore the limits of the block-type methods. This assessment, we leave to the contestants of the actual lattice challenge that is defined in the next section.

6 The challenge

In Section 4, we have constructed challenge lattices L_m of dimension m, for $m \geq 500$. The results in Section 3 together with the pseudo-random choice of L_m guarantee the existence of vectors $\mathbf{v} \in L_m$ with $\|\mathbf{v}\|_2 < n(m)$, which are hard to find. For a toy example, refer to Appendix C.

As stated before, we want the lattice challenge to be *open* in the sense that it does not terminate when the *first* short vector is found. Having proven the existence of just one solution might suggest that there are no more, but during practical experiments, we found that many successively shorter vectors exist. For example in Figure 4, we display that in dimension $m = 200$ BKZ with increasing block size subsequently finds smaller and smaller lattice vectors.

We propose the following challenge to all researchers and students.

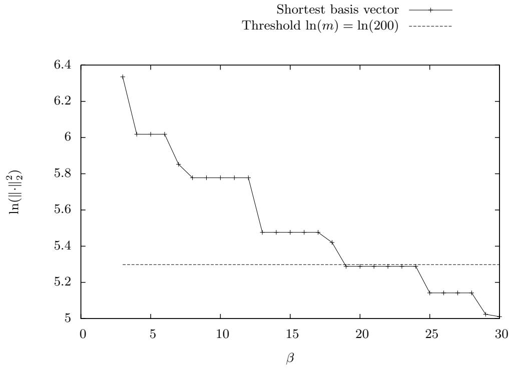


Fig. 4. Shortest vectors found by β-BKZ in dimension $m = 200$.

Lattice Challenge

The contestants are given lattice bases of lattices L_m, together with a norm bound ν. Initially, we set $\nu = \lceil \sqrt{m} \rceil$.

The goal is to find a vector $\mathbf{v} \in L_m$, with $\|\mathbf{v}\|_2 < \nu$.
Each solution $\mathbf{v}$ to the challenge decreases ν to $\|\mathbf{v}\|_2$.
The challenge is hosted at `http://www.latticechallenge.org`.

Acknowledgements

We thank Oded Regev, Damien Stehlé, Phong Q. Nguyen, Nicolas Gama, and Moon Sung Lee for their helpful remarks and suggestions. Furthermore, we thank the PQCrypto 2008 program committee and the anonymous reviewers for their valuable comments.

References

1. D. Aharonov and O. Regev. Lattice problems in $\mathsf{NP} \cap \mathsf{coNP}$. *J. ACM*, 52(5):749–765, 2005.
2. M. Ajtai. Generating hard instances of lattice problems. In *Proceedings of the Annual Symposium on the Theory of Computing (STOC)*, pages 99–108. ACM Press, 1996.
3. M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the Annual Symposium on the Theory of Computing (STOC)*, pages 284–293. ACM Press, 1997.
4. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the Annual Symposium on the Theory of Computing (STOC)*, pages 601–610. ACM Press, 2001.
5. D. Bailey and R. Crandall. On the random character of fundamental constant expansions. *Experimental Mathematics*, 10(2):175–190, 2001.
6. D. Bailey and R. Crandall. Random generators and normal numbers. *Experimental Mathematics*, 11(4):527–546, 2002.
7. W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–635, 1993.
8. D. J. Bernstein, J. Buchmann, and E. Dahmen, editors. *Post-Quantum Cryptography*. Springer-Verlag, 2008.
9. J. Buchmann, R. Lindner, and M. Rückert. Explicit hard instances of the shortest vector problem (extended version). Cryptology ePrint Archive, Report 2008/333, 2008. `http://eprint.iacr.org/2008/333`.

10. J. Cai and A. Nerurkar. An improved worst-case to average-case connection for lattice problems. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 468–477, 1997.
11. Certicom Corp. The Certicom ECC Challenge. `http://www.certicom.com/index.php/the-certicom-ecc-challenge`.
12. D. Coppersmith and A. Shamir. Lattice attacks on NTRU. In *Advances in Cryptology — Eurocrypt 1997*, pages 52–61, 1997.
13. A. M. Daniel Goldstein. On the equidistribution of hecke points. *Forum Mathematicum 2003, 15:2*, pages 165–189, 2003.
14. B. Filipović. Implementierung der Gitterbasenreduktion in Segmenten. Master's thesis, Johann Wolfgang Goethe-Universität Frankfurt am Main, 2002.
15. N. Gama and P. Q. Nguyen. Finding short lattice vectors within mordell's inequality. In *STOC*, pages 207–216, 2008.
16. N. Gama and P. Q. Nguyen. Predicting lattice reduction. In N. P. Smart, editor, *Advances in Cryptology — Eurocrypt 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 31–51. Springer-Verlag, 2008.
17. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the Annual Symposium on the Theory of Computing (STOC) 2008*, pages 197–206. ACM Press, 2008.
18. O. Goldreich and S. Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000.
19. O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology — Crypto 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 112–131. Springer-Verlag, 1997.
20. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In J. Buhler, editor, *Algorithmic Number Theory Symposium — ANTS*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer-Verlag, 1998.
21. J. Hoffstein, J. H. Silverman, and W. Whyte. Estimated breaking times for NTRU lattices. Technical Report 012, Version 2, NTRU Cryptosystems, 2003. `http://ntru.com/cryptolab/tech_notes.htm`.
22. N. Howgrave-Graham, H. J., J. Pipher, and W. Whyte. On estimating the lattice security of NTRU. Technical Report 104, Cryptology ePrint Archive, 2005. `http://eprint.iacr.org/2005/104/`.
23. H. Koy. Primale-duale Segment-Reduktion. `http://www.mi.informatik.uni-frankfurt.de/research/papers.html`, 2004.
24. H. Koy and C.-P. Schnorr. Segment LLL-reduction of lattice bases. In J. H. Silverman, editor, *CaLC*, volume 2146 of *Lecture Notes in Computer Science*, pages 67–80. Springer, 2001.
25. J. C. Lagarias, H. W. L. Jr., and C.-P. Schnorr. Korkin-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10(4):333–348, 1990.
26. A. Lenstra, H. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
27. C. Ludwig. A faster lattice reduction method using quantum search. In *Algorithms and Computation*, volume 2906 of *Lecture Notes in Computer Science*, pages 199–208. Springer-Verlag, 2003.
28. C. Ludwig. *Practical Lattice Basis Sampling Reduction*. PhD thesis, Technische Universität Darmstadt, 2005. `http://elib.tu-darmstadt.de/diss/000640/`.
29. V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. Swifft: A modest proposal for fft hashing. In *Fast Software Encryption (FSE) 2008*, Lecture Notes in Computer Science, pages 54–72. Springer-Verlag, 2008.
30. K. S. McCurley. The discrete logarithm problem. In C. Pomerance, editor, *Cryptology and computational number theory*, pages 49–74, Providence, 1990. American Mathematical Society.
31. D. Micciancio. Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. *SIAM Journal on Computing*, 34(1):118–169, 2004.
32. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.
33. P. Q. Nguyen and D. Stehlé. Floating-point LLL revisited. In R. Cramer, editor, *Advances in Cryptology — Eurocrypt 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 215–233. Springer-Verlag, 2005.
34. P. Q. Nguyen and D. Stehlé. LLL on the average. In F. Hess, S. Pauli, and M. E. Pohst, editors, *ANTS*, volume 4076 of *Lecture Notes in Computer Science*, pages 238–256. Springer-Verlag, 2006.
35. C. Peikert. Limits on the hardness of lattice problems in ℓ_p norms. In *IEEE Conference on Computational Complexity*, pages 333–346. IEEE Computer Society, 2007.

36. O. Regev. Quantum computation and lattice problems. *SIAM J. Comput.*, 33(3):738–760, 2004.
37. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th annual ACM symposium on Theory of computing*, pages 84–93. ACM Press, 2005.
38. O. Regev. On the complexity of lattice problems with polynomial approximation factors, 2007. A survey for the LLL+25 conference.
39. RSA Security Inc. The RSA Challenge Numbers. `http://www.rsa.com/rsalabs/node.asp?id=2093`.
40. C. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53:201–224, 1987.
41. C. Schnorr. Block reduced lattice bases and successive minima. *Combinatorics, Probability and Computing*, 4:1–16, 1994.
42. C. Schnorr. Lattice reduction by random sampling and birthday methods. In *STACS 2003: 20th Annual Symposium on Theoretical Aspects of Computer Science*, volume 2607 of *Lecture Notes in Computer Science*, pages 146–156. Springer-Verlag, 2003.
43. V. Shoup. Number theory library (NTL) for C++. `http://www.shoup.net/ntl/`.
44. D. Stehlé. Damien Stehlé's homepage at école normale supérieure de Lyon. `http://perso.ens-lyon.fr/damien.stehle/english.html`.

A Completing the proof of Theorem 1

We want to show there exists an n_0, such that

$$c_2 n \ln(n) < \ln(2) \lfloor c_1 n \ln(n) \rfloor$$

for all $n \geq n_0$. Recall that $c_2 < c_1 \ln(2)$, so there exists an $\epsilon > 0$, such that $c_2 = c_1 \ln(2) - \epsilon$. We choose n_0 to be the smallest positive integer, such that $\ln(2) \leq \epsilon\, n_0 \ln(n_0)$. We prove the inequality.

$$\begin{aligned} c_2 n \ln(n) &= (c_1 \ln(2) - \epsilon) n \ln(n) \\ &\leq \ln(2) c_1 n \ln(n) - \epsilon\, n \ln(n) \\ &\leq \ln(2)(c_1 n \ln(n) - 1) \\ &< \ln(2) \lfloor c_1 n \ln(n) \rfloor \end{aligned}$$

This completes the proof.

B Ratio between m, n, and q

In order to get an idea of the ratios $n : m$ and $q : m$ in our challenge lattices, refer to Figure 5.

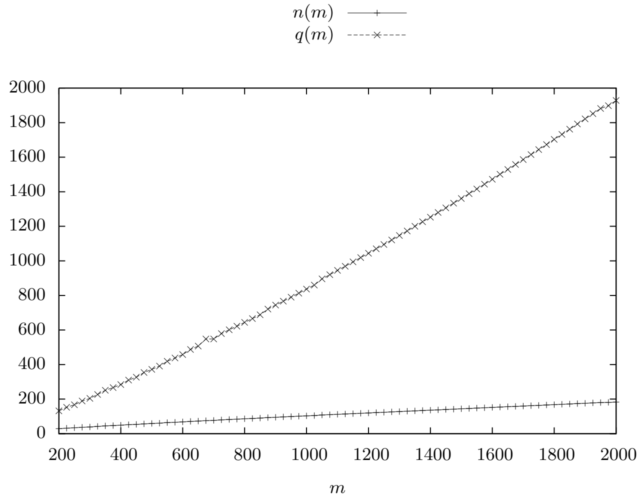


Fig. 5. Ratio between challenge dimension m, reference dimension n, and the modulus q.

C Challenge example

The following low-dimensional example gives an idea of what the challenge lattices, and the short vectors in them, essentially look like. Its block structure is similar to the one found by Coppersmith and Shamir for NTRU lattices [12]. This is not surprising because both belong to the class of modular lattices.

Example 1. The transposed challenge basis for $m = 20, n = 6, q = 13$ looks like:

```
[
[1 0 0 0 0 0 0 0 0 0 0 0 0 0  -4  -4  -2   0 -11  -7]
[0 1 0 0 0 0 0 0 0 0 0 0 0 0  -2  -6  -8   0  -6   0]
[0 0 1 0 0 0 0 0 0 0 0 0 0 0  -5  -7 -12  -6 -10  -1]
[0 0 0 1 0 0 0 0 0 0 0 0 0 0  -4  -6  -1  -1   0  -3]
[0 0 0 0 1 0 0 0 0 0 0 0 0 0  -1 -10  -4  -4 -10   0]
[0 0 0 0 0 1 0 0 0 0 0 0 0 0  -3  -8   0  -1  -4  -7]
[0 0 0 0 0 0 1 0 0 0 0 0 0 0  -6  -5   0  -4  -6  -4]
[0 0 0 0 0 0 0 1 0 0 0 0 0 0  -4  -9  -8  -2  -9  -1]
[0 0 0 0 0 0 0 0 1 0 0 0 0 0  -8  -2  -5  -1  -2  -5]
[0 0 0 0 0 0 0 0 0 1 0 0 0 0 -11 -12  -2 -11 -10  -5]
[0 0 0 0 0 0 0 0 0 0 1 0 0 0  -9  -1  -7  -7   0  -6]
[0 0 0 0 0 0 0 0 0 0 0 1 0 0  -2  -2  -7  -2  -5  -6]
[0 0 0 0 0 0 0 0 0 0 0 0 1 0   0 -12  -7  -5  -2  -6]
[0 0 0 0 0 0 0 0 0 0 0 0 0 1 -11  -4 -11  -3  -4  -2]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0  13   0   0   0   0   0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0   0  13   0   0   0   0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0   0   0  13   0   0   0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0   0   0   0  13   0   0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0   0   0   0   0  13   0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0   0   0   0   0   0  13]
]
```