Polynomial Time Bounded Distance Decoding near Minkowski's Bound in Discrete Logarithm Lattices

Léo Ducas* [1] and Cécile Pierrot[1]

[1]Cryptology Group, CWI, Amsterdam, The Netherlands

Abstract

We propose a concrete family of dense lattices of arbitrary dimension n in which the lattice Bounded Distance Decoding (BDD) problem can be solved in deterministic polynomial time. This construction is directly adapted from the Chor-Rivest cryptosystem (IEEE-TIT 1988).

The lattice construction needs discrete logarithm computations that can be made in deterministic polynomial time for well-chosen parameters. Each lattice comes with a deterministic polynomial time decoding algorithm able to decode up to large radius. Namely, we reach decoding radius within $O(\log n)$ Minkowski's bound, for both ℓ_1 and ℓ_2 norms.

Keywords: Dense lattices. Bounded Distance Decoding (BDD). Minkoswki's bound.
Mathematics Subject Classification (2010) 94B35. 94B65. 11H31. 11H71.

1 Introduction

Sphere Packing. Given a large number of equal non-overlapping spheres, the question of finding the most efficient way to pack them together is quite an old problem. Arranging the spheres so that their centers form an Euclidean lattice (a.k.a quadratic form) helps to find solutions. For instance, in two dimensions and with the Euclidean norm, Kepler already conjectured in 1610 that the familiar hexagonal lattice solves the packing problem but the first proof was only given in 1940 by Tóth [Tot40]. However all ball arrangements is not of a lattice nature, and, with arbitrary norm and dimension, the question whether or not the best density is achieved on the lattice arrangements is still open. Yet, a classical method to find (maybe not the best but) a solution to this sphere packing problem is to aim at constructing dense lattices.

Intuitively, the density of a lattice is the proportion of the space that is occupied by maximum radius non-overlapping spheres centered in the lattice points. For instance the density of the hexagonal lattice in the plane is $\pi/\sqrt{12} \approx 0.907$. Forgetting about spheres, the density $\bar{\lambda}_1^{(p)}(\mathcal{L})$ of a lattice $\mathcal{L}$ of rank n (in the ℓ_p-norm) can be measured by the normalized length of its shortest vector:

$$\bar{\lambda}_1^{(p)}(\mathcal{L}) := \frac{\lambda_1^{(p)}(\mathcal{L})}{\mathrm{Vol}(\mathcal{L})^{1/n}}, \quad \text{where } \lambda_1^{(p)}(\mathcal{L}) := \min_{x \in \mathcal{L}\setminus\{0\}} ||x||_p. \tag{1}$$

*Supported by a Veni Innovational Research Grant from NWO under project number 639.021.645.

Minkowski's theorem provides an upper bound on the density of any lattice, generically:

$$\bar{\lambda}_1^{(p)}(\mathcal{L}) \leq \mathcal{M}_n^{(p)} \tag{2}$$

where $\mathcal{M}_n^{(p)} = 2\cdot\mathrm{Vol}(\mathcal{B}_n^{(p)})^{-1/n}$ and $\mathrm{Vol}(\mathcal{B}_n^{(p)})$ denotes the volume of the unit ball in ℓ_p-norm in $\mathbb{R}^n$. Note that this *Minkowski's bound* $\mathcal{M}_n^{(p)}$ depends on dimension and norm only. In particular, for the ℓ_1-norm (resp. ℓ_2-norm), the density of any n-dimensional lattice is upperbounded by $\mathcal{M}_n^{(1)}$ (resp. $\mathcal{M}_n^{(2)}$) where:

$$\mathcal{M}_n^{(1)} = (n!)^{1/n} \qquad\qquad \sim n/e \tag{3}$$

$$\mathcal{M}_n^{(2)} = 2\cdot\Gamma\left(\frac{n}{2}+1\right)^{1/n}\Big/\sqrt{\pi} \qquad\qquad \sim\sqrt{2n/\pi e}. \tag{4}$$

For the ℓ_2 norm, we know this bound to be tight up to constant factors. Indeed, there exists sequences of lattices for which $\bar{\lambda}_1^{(2)}(\mathcal{L}) = \Theta(\sqrt{n})$, to be compared with Minkowski's bound which has equivalent $\sqrt{2n/\pi e}$ in this case. It is for example known that random lattices have with high probability a first normalized minima very close to this bound [Ajt06]. Explicit constructions are also known, from Martinet [Mar78] and from Shioda [Shi91, Elk94], the latter being known as Mordell-Weil lattices.

A simpler family $A_n^{(k)}$ of lattices was given by Craig (called repeated difference lattices in [CS13, Chapter 8, Section 6]), and for $k = \Theta(n/\log n)$, the minimal distance of these lattices are logarithmically close to Minkowski's bound in both ℓ_1 and ℓ_2 norms:

$$\bar{\lambda}_1^{(1)}(A_n^{(k)}) \geq \Theta(n/\log n) \tag{5}$$

$$\bar{\lambda}_1^{(2)}(A_n^{(k)}) \geq \Theta(\sqrt{n}/\log n). \tag{6}$$

Bounded Distance Decoding. The Bounded Distance Decoding (BDD) problem is the algorithmic facet of sphere packing.

Definition 1 (Bounded Distance Decoding problem in ℓ_p-norm.)**.** *For a full-rank lattice $\mathcal{L} \subset \mathbb{R}^n$, and a bounded decoding radius $r^{(p)} \leq \lambda_1^{(p)}(\mathcal{L})/2$, given a target:*

$$t = v + e,$$

where $v \in \mathcal{L}$ and $||e||_p < r^{(p)}$, recover v and e.

As for the density we will note $\bar{r}^{(p)}$ the normalized decoding radius $\bar{r}^{(p)} = r^{(p)}/\mathrm{Vol}(\mathcal{L})^{1/n}$. Note that the condition $r^{(p)} \leq \lambda_1^{(p)}(\mathcal{L})/2$ guarantees the unicity of the solution, but we cannot insure it beyond this radius. Indeed, let x be a short vector such that $||x||_p = \lambda_1^{(p)}$ then, for an error $e = x/2$, namely for an instance $t = 3x/2$, we cannot tell if t comes from the lattice vectors x or $2x$.

The Bounded Distance Decoding problem plays a crucial role in communication over a noisy channel, as it allows to separate a codeword $v \in \mathcal{L}$ from the noise e introduced by the channel. For the square lattice $\mathbb{Z}^n$, this problem is trivial since we just need to round each coordinate to the nearest integer. Yet this is far from one could hope for in term of error tolerance, since the best radius $\bar{r}^{(2)} = r^{(2)} = \lambda_1^{(2)}(\mathbb{Z}^n)/2 = 1/2$ is constant. One instead hopes to get a decoding radius as close as possible to half of Minkowski's bound, namely of the order of magnitude of $\sqrt{n}$. Unfortunately, efficient decoding algorithm are not known for very dense lattices as random ones, Martinet's, Mordell-Weil's or Craig's lattices.

Currently, the best normalized decoding radius achievable in polynomial time was given by Micciancio and Nicolosi [MN08] over Barnes-Wall lattices BW_N (of dimension $n = 2^N$). It reaches the maximal decoding radius in ℓ_2-norm and can even be efficiently extended to list-decoding [GP12], but remains quite far from Minkowski's bound. Indeed, the maximal decoding radius $r^{(2)} = \lambda_1^{(2)}(\mathrm{BW}_N)/2$ is only such that:

$$\bar{r}^{(2)} = \Theta(\sqrt[4]{n}).$$

While strict BDD close to Minkowski's bound was still an open problem, a relaxed variant allowing a small probability of failure over the randomess of the error term e was recently solved by Yan et al. [YLLW14] using construction D over Polar-codes.

1.1 Contribution

In this paper, we show how to construct dense lattices admitting efficient BDD algorithm for a radius near Minkowski's bound for both ℓ_1 and ℓ_2 norms. Namely, for the ℓ_2-norm we reach:

$$\bar{r}^{(2)} = \Theta\left(\frac{\sqrt{n}}{\ln n}\right),$$

to be compared with the best known radius over Barnes-Wall lattices that is $\Theta(\sqrt[4]{n})$ and with the theoretical Minkowski's bound that is equivalent to $\Theta(\sqrt{n})$. For the ℓ_1-norm we reach:

$$\bar{r}^{(1)} = \Theta\left(\frac{n}{\ln n}\right),$$

to be compared with the theoretical Minkowski's bound that is equivalent to $\Theta(n)$. Constructing a lattice and running the associated decoding algorithm have both polynomial time complexity. Moreover, we emphasize that neither the construction nor the decoding algorithm make use of a quantum computer: their construction rely on discrete logarithm computations, which can be made easy by appropriate parametrization.

The construction is not so new, since it is directly inspired from deprecated knapsack-based cryptosystems first proposed in 1988 by Chor and Rivest [CR88, Len91]. Their construction is based on discrete logarithm over finite-field extension, yet a very similar construction was proposed by Okamoto *et al.* [OTU00] relying merely on discrete logarithm of modular integers. For ease of presentation, we base ourself on the later.

The core idea behind those cryptosystems is that the subset-prime-product problem over the integers is an easy problem. More precisely, if $p_1, \ldots, p_m \leq B$ are primes, given $t = \prod p_i^{e_i} \mod m$ for positive integers e_i, and assuming $B^{||e||_1} \leq m$, recovering e can easily be done by trial divisions. Taking discrete logarithms allows to convert this to a subset-sum problem, that was the underlying hard problem of these protocols.

Ignoring the cryptographic countermeasures making this subset-sum instance hard to an adversary, we re-interpret this construction as a lattice error-correction scheme. This construction is done in Paragraph 3.1. Discrete logarithm computations occur while constructing such a lattice, but they can be made easy by appropriate choice of m, and do not need any quantum computer algorithm.

In order to decode such a lattice, the only remaining technicality to be dealt with is the fact that, in these previous protocols, the decoded error e is assumed to have positive integer coefficients. The integrality condition is easily solved by rounding. The positivity condition can also be removed by rational reconstruction, *i.e.* solving the shortest-vector problem in dimension 2. Our decoding algorithm is detailed in Paragraph 3.2. In Paragraph 3.3 we propose concrete well-chosen parameters to build a family of dense lattices with polynomial time decoding algorithms. Finally in Section 4 we discuss about some generalizations of the scheme that could help in practice to obtain better decoding radius – but all these slight changes do not interfere with our asymptotic results.

2 Preliminaries

Notation. In the sequel, if $x \in \mathbb{R}^n$ is a vector, then x_i denotes its i-th coordinate. The ℓ_p-norm of a vector x is noted $||x||_p$, and is defined by $||x||_p = (\sum |x_i|^p)^{1/p}$. Moreover we denote by $\lfloor x \rceil$ the coefficient-wise rounding to $\mathbb{Z}^n$.

Useful inequalities. We make use of the following statements:

$$\forall x \in \mathbb{R}^n, \qquad ||x||_1 \leq \sqrt{n} \cdot ||x||_2. \tag{7}$$

$$\forall x \in \mathbb{R}^n, \qquad ||\lfloor x \rceil||_1 \leq 2||x||_1. \tag{8}$$

$$\forall x \in \mathbb{Z}^n, \qquad ||x||_1 \leq ||x||_2^2. \tag{9}$$

Inequality (7) is an application of Cauchy-Schwarz's inequality in the Euclidean space $\mathbb{R}^n$. Besides, Inequality (8) comes from inequality $|\lfloor y \rceil| \leq 2|y|$ for any real number y; whereas Inequality (9) is a generalization of $|s| \leq s^2$ for any (possibly negative) integer s.

Factoring.

Definition 1. *Let B be a natural integer. An integer is B-smooth if all its factors are lower or equal to B.*

Factoring is considered as a hard problem in cryptography. Yet, when we know we have a smooth number, it becomes much more easy to factorize it.

Proposition 1 (Factorisation by trial division)**.** *If B is a natural integer and n a B-smooth integer with k factors, then one can find the whole factorization of n with complexity:*

$$O(B \cdot k \cdot (\log n)^2).$$

Proof. Let n be the target number we want to factorize. With these hypotheses, the simplest algorithm consists in trying all prime p lower than B and to see if it factors n or not. If it does, we pursue with the quotient n/p. The algorithm ends when we have found all its k factors (with multiplicity). Each trial division costs $O((\log n)^2)$ bit operations. Note that this exponent 2 depends on the underlying multiplication algorithm, so it can be improved with fast algorithms. □

Remark 1. *If B gets large, algorithms such as Pollard-ρ [Pol78], or even Lenstra's Elliptic Curve Method [Len87] will eventually become much faster for factorization. Yet for our application B will be no larger than $\Theta(n \log n)$ where n is the dimension of the considered lattice: it is plausible that trial division remains the fastest method for parameters of interest.*

Discrete Logarithm. As for factoring, computing discrete logarithms in a group that has a smooth order is easier than in a generic group of the same order of magnitude. It comes from the fact that Pohlig-Hellman algorithm [PH78] helps to reduce the problem to computing discrete logarithms in all the subgroups, so that the main parameter to discuss with the hardness of computing discrete logarithms is not the size of the group itself, but the size of its largest subgroup of prime order. This remark will help to construct a family of lattices without being bothered by these computations. More precisely:

Proposition 2. *For any group G of order $\prod p_i^{a_i}$, where p_i are prime numbers, using a combination of Pohlig-Hellman [PH78] and Pollard-ρ [Pol78] algorithms permits to compute discrete logarithms in G in:*

$$O\left(\sum a_i(\ln(|G|) + \sqrt{p_i})\right)$$

group operations.

3 Discrete logarithm based family of lattices

3.1 Settings and construction

Let us fix n a natural integer and m an integer such that $(\mathbb{Z}/m\mathbb{Z})^*$ is a cyclic group.[1] Recall that the size of the group $(\mathbb{Z}/m\mathbb{Z})^*$ is $\varphi(m)$, where φ denotes Euler's totient function. Now let B be a natural integer depending on n such that the set of prime numbers $\mathcal{F} \subset \mathbb{N}$ defined as:

$$\mathcal{F} = \{p \in \mathbb{N} \,|\, p \text{ prime}, p \text{ does not divide } m \text{ and } p \leqslant B\},$$

has exactly n elements $p_1, \cdots, p_n$. Recall that the k-th prime number is asymptotically equivalent to $k \log k$. So asymptotically we have $B \sim n \log n$. Now, consider the group morphism:

$$\begin{aligned} \psi : \mathbb{Z}^n &\to (\mathbb{Z}/m\mathbb{Z})^* \\ (x_1, \cdots, x_n) &\mapsto \prod_{i=1}^{n} p_i^{x_i} \mod m. \end{aligned}$$

The lattice of multiplicative relation between $p_1, \cdots, p_n$ is defined as:

$$\mathcal{L} := \ker \psi = \left\{ x = (x_1, \cdots, x_n) \in \mathbb{Z}^n \mid \prod_{i=1}^{n} p_i^{x_i} = 1 \mod m \right\}.$$

As $\mathcal{L}$ is a full rank sublattice of $\mathbb{Z}^n$, we have $\mathrm{Vol}(\mathcal{L}) = |\mathbb{Z}^n/\mathcal{L}| = |\operatorname{Im} \psi|$. In consequence $\mathrm{Vol}(\mathcal{L}) \leq \varphi(m)$, with equality if and only if ψ is surjective.

Calling g a generator of the multiplicative cyclic group $(\mathbb{Z}/m\mathbb{Z})^*$, we see that $\mathcal{L}$ is a lattice of dimension n that can be rewritten as:

$$\mathcal{L} = \left\{ x = (x_1, \cdots, x_n) \in \mathbb{Z}^n \mid \sum_{i=1}^{n} x_i \log_g p_i = 0 \mod \varphi(m) \right\}. \tag{10}$$

If one of the prime $p \in \mathcal{F}$ is a generator of $(\mathbb{Z}/m\mathbb{Z})^*$ we are even able to explicit a basis of this lattice. Indeed, in this case without loss of generality we can assume p_n is a generator of $(\mathbb{Z}/m\mathbb{Z})^*$. Thus $\log_g p_n$ is invertible modulo $\varphi(m)$. An explicit basis of $\mathcal{L}$ can be constructed as the row vectors of the following matrix:

$$\begin{pmatrix} 1 & & & & -\log_g p_1 / \log_g p_n \\ & 1 & & & -\log_g p_2 / \log_g p_n \\ & & \ddots & & \vdots \\ & & & 1 & -\log_g p_{n-1} / \log_g p_n \\ & & & & \varphi(m) \end{pmatrix}$$

[1]Later we relax this assumption and discuss about m.

where blank entries should be read as 0. We note that:

$$\mathrm{Vol}(\mathcal{L}) = \varphi(m).$$

This is not surprising: if one of the primes of $\mathcal{F}$ is a generator of $(\mathbb{Z}/m\mathbb{Z})^*$ then ψ is surjective. Explicit bases can also be efficiently constructed without such an assumption but requires more care. We emphasize that the volume of $\mathcal{L}$ is then not necessarily equal to $\varphi(m)$, but cannot be larger.

Remark 2. *Thus to explicitly construct the basis we need to compute all the discrete logarithms in $\mathcal{F}$ modulo $\varphi(m)$. We will discuss about complexity and efficiency in Section 3.3, while choosing some relevant parameters.*

3.2 Decoding algorithm

Beware that in this Section 3.2 all the radius are without any consideration of the volume – namely, they are not normalized yet.

3.2.1 Positive discrete errors in ℓ_1-norm

In this paragraph we present how to recover x from a given vector $t = x + e$ where:

- $x \in \mathcal{L}$
- e is a positive discrete bounded error, namely $e \in \mathbb{N}^n$ such that $||e||_1 \leqslant r_{\mathbb{N}}^{(1)}$, for some yet to determine bound $r_{\mathbb{N}}^{(1)}$.

The first step of the decoding algorithm consist in computing the following product modulo m:

$$\prod_{i=1}^{n} p_i^{t_i} = \prod_{i=1}^{n} p_i^{x_i} \prod_{i=1}^{n} p_i^{e_i} = \prod_{i=1}^{n} p_i^{e_i} \mod m.$$

From $||e||_1 \leqslant r_{\mathbb{N}}^{(1)}$ and $p \leqslant B$ for any $p \in \mathcal{F}$ we get $\prod_{i=1}^{n} p_i^{e_i} \leqslant B^{r_{\mathbb{N}}^{(1)}}$. Efficient decoding can be ensured up to the following ℓ_1 radius:

$$r_{\mathbb{N}}^{(1)} = \frac{\ln m}{\ln B}. \tag{11}$$

Indeed, in that case, $B^{r_{\mathbb{N}}^{(1)}} = m$ so the product $\prod_{i=1}^{n} p_i^{e_i}$, which is lower than m, can be computed in $\mathbb{Z}$, and not only modulo m. Then we factorize this integer, which is easy since it's a smooth number. From this factorization we recover the error vector e.

3.2.2 Discrete errors in ℓ_1-norm

If e is again a discrete bounded error such that $||e||_1 \leqslant r_{\mathbb{Z}}^{(1)}$, but without any constraint on the sign of its coefficients, namely $e \in \mathbb{Z}^n$, we need to slightly change the decoding algorithm. Here we compute again $f = \prod_{i=1}^{n} p_i^{t_i} \mod m$, yet this is no longer equal to a product but to a fraction of the form:

$$f = \prod_{i \text{ s.t. } e_i > 0}^{n} p_i^{e_i} \cdot \prod_{i \text{ s.t. } e_i < 0} p_i^{e_i} = u/v \mod m.$$

To recover $u = \prod_{i \text{ s.t. } e_i>0}^{n} p_i^{e_i}$ and $v = \prod_{i \text{ s.t. } e_i<0} p_i^{-e_i}$ not only modulo m but in $\mathbb{Z}$, we use the following lemma.

Lemma 1. *If u, v are positive coprime integers and invertible modulo m such that $u, v < \sqrt{m/2}$, and if $f = u/v \bmod m$, then $\pm(u, v)$ are the shortests vector of the 2-dimensional lattice $L = \{(x, y) \in \mathbb{Z}^2 | x - fy = 0 \bmod m\}$ for the ℓ_2-norm.*

In particular, given f and m, one can recover (u, v) in polynomial time.

Proof. Let us assume that their exists a non-zero vector $(u', v') \in L$ shorter than (u, v). First note that (u', v') is $\mathbb{R}$-linearly independent of (u, v), since u and v are coprime.

Now consider the lattice L' generated by (u, v) and (u', v'). Because L' is a full-rank sublattice of L we have $\mathrm{Vol}(L') \geq \mathrm{Vol}(L)$. Since $(f, 1)$ and $(m, 0)$ form a basis of L we have $\mathrm{Vol}(L) = m$. It leads to:

$$\mathrm{Vol}(L') \geq m.$$

On the other hand by Hadamard inequality we have:

$$\mathrm{Vol}(L') \leq ||(u, v)||_2 \cdot ||(u', v')||_2 \leq ||(u, v)||_2{}^2 < m.$$

By contradiction this concludes that $\pm(u, v)$ are the shortest vectors of L. The vector (u, v) is then easy to recover in polynomial time, using Gauss' algorithm. $\square$

Having recovered the vector (u, v), it remains to recover e by factorization of u and v. In conclusion, we have a decoding algorithm for any integer errors up to ℓ_1 radius:

$$r_{\mathbb{Z}}^{(1)} = \frac{\ln(m/2)}{2 \cdot \ln B}. \tag{12}$$

Indeed, an error bounded by this radius ensures u and v to be strictly lower than $\sqrt{m/2}$.

3.2.3 Continuous error in ℓ_1-norm

The generalization to continuous error is rather straightforward, and consists simply in first rounding the target t coordinate-wise, to reduce the problem to the discrete case.

Indeed, let $t = x + e$ where $x \in \mathcal{L}$ and e is a small error. Set $t' = \lfloor t \rceil$, and note that since $x \in \mathcal{L} \subset \mathbb{Z}^n$, we have $t' = x + \lfloor e \rceil$. Applying the previous decoding algorithm to t' will yield the correct answer x if $||\lfloor e \rceil||_1 \leq r_{\mathbb{Z}}^{(1)}$.

Recalling Inequality (8): $||\lfloor x \rceil||_1 \leq 2||x||_1$, we conclude that this algorithm provides a ℓ_1 decoding radius $r^{(1)} = r_{\mathbb{Z}}^{(1)}/2$, namely:

$$r^{(1)} = \frac{\ln(m/2)}{4 \cdot \ln B}. \tag{13}$$

3.2.4 Continuous error in ℓ_2-norm

Inequality (7) ensures that the above algorithm also decodes errors up to ℓ_2 radius $r^{(2)} = r^{(1)}/\sqrt{n}$, namely:

$$r^{(2)} = \frac{\ln(m/2)}{4\sqrt{n} \cdot \ln B}. \tag{14}$$

Remark 3 (Generalization to ℓ_p-norm)**.** *Even if the Euclidean norm (or ℓ_2-norm) is of major interest in practice while dealing with BDD, note that the key of our algorithm is to decode in ℓ_1-norm. Actually we could generalize this decoding to any norm, by relying on inequality $||x||_1 \leq n^{(p-1)/p}||x||_p$ that is true for any $x \in \mathbb{R}^n$. As in Paragraph 3.2.4 above, it yields a decoding algorithm for errors up to ℓ_p radius:*

$$r^{(p)} = \frac{\ln(m/2)}{4n^{(p-1)/p} \cdot \ln B}.$$

Remark 4. *As for lattice construction, the asymptotic complexity of this algorithm is detailed later for some well-chosen parameters.*

3.3 A family of lattices approaching Minkowski's bound.

Proposition 3. *For every natural integer n, we are able to construct in polynomial time an n-dimensional lattice with a polynomial time algorithm decoding errors up to normalized decoding radius:*

$$\begin{aligned} \bar{r}^{(2)} &= \Theta\left(\frac{\sqrt{n}}{\ln n}\right), \quad \textit{for the Euclidean norm,} \\ \textit{and} \quad \bar{r}^{(1)} &= \Theta\left(\frac{n}{\ln n}\right), \quad \textit{for the } \ell_1\textit{-norm.} \end{aligned}$$

In both ℓ_1 and ℓ_2 norms we reach Minkowski's bound up to logarithmic factors. Indeed the normalized radius of Proposition 3 need to be compared with Minkowski's bounds $\mathcal{M}_n^{(2)}$ and $\mathcal{M}_n^{(1)}$ that are equivalent to:

$$\begin{aligned} &\Theta(\sqrt{n}), \quad \text{for the Euclidean norm,} \\ \text{and} \quad &\Theta(n), \quad \text{for the } \ell_1\text{-norm,} \end{aligned}$$

where c is a constant given by Equation (3). Moreover, we only use classical algorithms so there is no need to have a quantum computer to construct the lattice or decode it.

Proof. Efficient construction. Let q be a prime number such that $q \geqslant 3$, and $n > 0$ a natural integer. Take:

$$m = q^n.$$

Theorem 1. *The group $(\mathbb{Z}/m\mathbb{Z})^*$ is cyclic if and only if m is $1, 2, 4, q^k$ or $2q^n$, where q is an odd prime and $n > 0$ a natural integer. For all other values of m the group is not cyclic.*

Theorem 1 (see for instance [Sha93, page 92] for more details) indicates that $G = (\mathbb{Z}/m\mathbb{Z})^*$ is a cyclic group of order $\varphi(m) = (q-1)q^{n-1}$. Now consider the lattice $\mathcal{L}$ defined by (10). We define B as the $(n+2)$-th prime number, so that $\mathcal{F}$ precisely has order n. In the sequel we use an equivalent for B which is easier to manipulate than the formal definition. We have $B \sim (n+2)\log(n+2)$ so:

$$B \sim n \log n. \tag{15}$$

To explicitly construct $\mathcal{L}$ we need to find n discrete logarithms modulo $\varphi(m)$. Namely if g is a generator of G and if $\mathcal{F} = \{p_1, \cdots, p_n\}$, then we compute all $\log_g p_i$ for $i = 1, \cdots, n$. To compute one of these discrete logarithms we use a combination of Pohlig-Hellman [PH78] and Pollard-Rho [Pol78] algorithms. Proposition 2 underlines that for any group G' of order $\prod t_i^{a_i}$ this can be done in $O(\sum_{t_i/|G'|} a_i(\ln(|G'|) + \sqrt{t_i}))$ group operations. Plugging with our value $|G| = \varphi(m) = (q-1)q^{n-1}$ we obtain one of these discrete logarithms in quadratic time with respect to n. Thus, we are able to construct the lattice $\mathcal{L}$ in cubic time. Again, it's the extremely high smoothness of m that makes this computation feasible.

Decoding a large ball. From Section 3.2 we can decode up to ℓ_1-radius (resp. ℓ_2-radius) $r^{(1)}$ (resp. $r^{(2)}$) given as in Equation (13) (resp. Equation (14)). To conclude we just need to compute the two corresponding normalized radius $\bar{r}^{(i)} = r^{(i)}/\mathrm{Vol}(\mathcal{L})^{1/n}$ for $i = 1, 2$. From $\mathrm{Vol}(\mathcal{L}) \geq \varphi(m)$ we are able to decode up to:

$$\bar{r}^{(2)} = \frac{\ln(m/2)}{4((q-1)q^{n-1})^{1/n}\sqrt{n} \cdot \ln B} \quad \text{and} \quad \bar{r}^{(1)} = \frac{\ln(m/2)}{4((q-1)q^{n-1})^{1/n} \cdot \ln B}.$$

From $\log m \sim n$ and $B \sim n \ln n$ we have:

$$\bar{r}^{(2)} = \Theta\left(\frac{\sqrt{n}}{\ln n}\right) \quad \text{and} \quad \bar{r}^{(1)} = \Theta\left(\frac{n}{\ln n}\right).$$

To deal with complexity, we recall that our decoding algorithm is simply:

1. Rounding a n-dimensional vector (namely t)
2. Computing a product modulo m (namely f)
3. Finding the shortest vector of a 2-dimensional integer lattice (namely (u, v))
4. and Factoring two B-smooth integers (namely u and v).

The first step is linear in n. The second one is linear as well. Thanks to Lemma 1 the third one is polynomial. Concerning the last step, in order to factor u for instance, plugging $k = ||e||_1$ in Proposition 1 make the complexity become $O(B \cdot ||e||_1 \cdot (\log u)^2)$. From (15), $||e||_1 < r^{(1)}$, $r^{(1)} = O(n)$, $u < m$ and $\log m = O(n)$ we get at worst a complexity in $O(n^4 \ln(n))$ for factoring the two B-smooth integers. Thus, our whole decoding algorithm for both ℓ_1 and ℓ_2 norms runs in polynomial time. □

4 Generalizations

For a fixed n, in the previous discussion the size of the normalized radius up to which we are able to decode varies with the ratio:

$$\frac{\ln m}{\varphi(m)}.$$

Here comes some variants that could help in practice to increase a bit this quantity. First one can notice that, according to Theorem 1, choosing $m = 2q^n$ instead of $m = q^n$ maintain the cyclicity and the size of the group $(\mathbb{Z}/m\mathbb{Z})^*$, sligthly improving the ratio $\ln m/\varphi(m)$.

4.1 Construction in a non-cyclic multiplicative group

In Section 3.1 we make the assumption that m is chosen such that $(\mathbb{Z}/m\mathbb{Z})^*$ is a cyclic group. Indeed, we can deal with more general constructions where m has no special form, except that it is not divisible by 8. Let's write its factorisation:

$$m = \prod_{i=1}^{k} p_i^{e_i}$$

where p_i are prime numbers and e_i natural integers. If $(\mathbb{Z}/m\mathbb{Z})^*$ is not cyclic then there is no generator, and talking about discrete logarithm may seem meaningless. Yet thanks to the Chinese Reminder Theorem:

$$(\mathbb{Z}/m\mathbb{Z})^* \simeq \prod_{i=1}^{k} (\mathbb{Z}/p_i^{e_i}\mathbb{Z})^*.$$

For all $p_i > 2$ we know from Theorem 1 that $(\mathbb{Z}/p_i^{e_i}\mathbb{Z})^*$ is cyclic. If $p_i = 2$ then as soon as $e_i = 0$ or $e_1 = 1$ the corresponding group is still cyclic. In these cyclic groups we can define a discrete logarithm modulo their order. To construct our lattice we just need to define the discrete logarithm of en element in $(\mathbb{Z}/m\mathbb{Z})^*$ as a k-dimensional vector where its coordinates are the discrete logarithms of its image in each cyclic group. Namely, the function log is here defined as a morphism:

$$\log : (\mathbb{Z}/m\mathbb{Z})^* \mapsto \bigoplus_{i=1}^{k} (\mathbb{Z}/\varphi(p_i^{e_i})\mathbb{Z}).$$

To put it in a nutshell, our construction works as soon as m is not divisible by 8. This permits to increase a bit the previous ratio: indeed, for two integers m of the same order of magnitude, $\varphi(m)$ decreases as m gets more and more smooth.

4.2 Adapting the construction to Finite Fields

We have described a construction based on the multiplicative group $(\mathbb{Z}/m\mathbb{Z})^\times$, but the original construction of Chorr and Rivest was working over the multiplicative group of a finite field extension $\mathbb{F}_{p^d}^\times$.

The drawback is that while the decoding algorithm remains polynomial time, the explicit construction of the lattice (computation of discrete logarithm) is not polynomial time anymore; though it can heuristically be made quasi-polynomial time [BGJT14]. This drawback can be cicumvented by resorting to product of finite fields, as recently done in the cryptosystem of Li et al. [LLXY17].

Nevertheless, asymptotically we were not able to decode better radii with such constructions.

Acknowledgment

We would like to thank Cong Ling, Dan Shepherd and Chaoping Xing for their interesting discussions and precious comments about this work.

References

[Ajt06] Miklós Ajtai. Generating random lattices according to the invariant distribution. *Draft of March 2006*, 2006.

[BGJT14] Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*, pages 1–16, 2014.

[CR88] Benny Chor and Ronald L. Rivest. A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Trans. Information Theory*, 34(5):901–909, 1988.

[CS13] John Horton Conway and Neil James Alexander Sloane. *Sphere packings, lattices and groups*, volume 290. Springer Science & Business Media, 2013.

[Elk94] Noam D Elkies. Mordell-Weil lattices in characteristic 2: I. construction and first properties. *International Mathematics Research Notices*, 1994(8):343–361, 1994.

[GP12] Elena Grigorescu and Chris Peikert. List decoding Barnes-Wall lattices. In *Computational Complexity (CCC), 2012 IEEE 27th Annual Conference on*, pages 316–325. IEEE, 2012.

[Len87] Hendrik W Lenstra. Factoring integers with elliptic curves. *Annals of Mathematics*, 126(3):649–673, 1987.

[Len91] Hendrik W Lenstra. On the Chor-Rivest knapsack cryptosystem. *Journal of Cryptology*, 3(3):149–155, 1991.

[LLXY17] Zhe Li, San Ling, Chaoping Xing, and Sze Ling Yeo. On the closest vector problem for lattices constructed from polynomials and their cryptographic applications. Cryptology ePrint Archive, Report 2017/1002, 2017. `https://eprint.iacr.org/2017/1002`.

[Mar78] Jacques Martinet. Tours de corps de classes et estimations de discriminants. *Inventiones mathematicae*, 44(1):65–73, 1978.

[MN08] Daniele Micciancio and Antonio Nicolosi. Efficient bounded distance decoders for Barnes-Wall lattices. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 2484–2488. IEEE, 2008.

[OTU00] Tatsuaki Okamoto, Keisuke Tanaka, and Shigenori Uchiyama. Quantum public-key cryptosystems. In *Advances in Cryptology, CRYPTO 2000*, pages 147–165. Springer, 2000.

[PH78] Stephen C. Pohlig and Martin E. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance (corresp.). *IEEE Transactions on Information Theory*, 24(1):106–110, 1978.

[Pol78] John Pollard. Monte Carlo methods for index computations mod p. In *Mathematics of Computation*, pages 918–924, 1978.

[Sha93] Daniel Shanks. *Solved and Unsolved Problems in Number Theory*. New York: Chelsea, 1993. 4th edition, ISBN: 978-0828412971.

[Shi91] Tetsuji Shioda. Mordell-Weil lattices and sphere packings. *American Journal of mathematics*, 113(5):931–948, 1991.

[Tot40] László Fejes Toth. Uber einen geometrischen satz. *Math Z.*, 46:79–83, 1940.

[YLLW14] Yanfei Yan, Ling Liu, Cong Ling, and Xiaofu Wu. Construction of capacity-achieving lattice codes: Polar lattices. *ArXiv preprint arXiv:1411.0187*, 2014.