Solving LWR via BDD Strategy: Modulus Switching Approach

Huy Quoc Le[1(✉)], Pradeep Kumar Mishra[1], Dung Hoang Duong[2], Masaya Yasuda[3,4]

[1] Graduate School of Mathematics, Kyushu University,
744 Motooka, Nishi-ku, Fukuoka-shi, Fukuoka-ken, 819-0395, Japan.
q-le@math.kyushu-u.ac.jp
p-mishra@math.kyushu-u.ac.jp
[2] School of Computing and Information Technology, University of Wollongong,
Northfields Avenue, Wollongong NSW 2522, Australia.
hduong@uow.edu.au
[3] Institute of Mathematics for Industry, Kyushu University,
744 Motooka, Nishi-ku, Fukuoka-shi, Fukuoka-ken, 819-0395, Japan.
[4] JST, CREST,
4-1-8 Honcho, Kawaguchi, Saitama, 332-0012, Japan.
yasuda@imi.kyushu-u.ac.jp

Abstract. The typical approach in attacking an $\mathrm{LWR}_{m,n,q,p}(\chi_s)$ instance parameterized by four integers m, n, q, p ($q \geq p$) and a probability distribution χ_s is just by simply regarding it as a Learning with Errors (LWE) modulo q instance and then trying to adapt known LWE attacks to this LWE instance. In this paper, we show that for an $\mathrm{LWR}_{m,n,q,p}(\chi_s)$ instance whose parameters satisfy a certain sufficient condition, one can use the BDD strategy to recover the secret with higher advantages if one transforms the LWR instance to an LWE modulo q' instance with q' chosen appropriately instead of an LWE modulo q instance. The optimal modulus q' used in our BDD attack is quite close to p as well as typically smaller than q. Especially, our experiments confirm that our BDD attack is much better in solving search-LWR in terms of root Hermite factor, success probability and even running time either in case the ratio $\log(q)/\log(p)$ is big or/and the dimension n is sufficiently large.

Key words: Learning with Errors (LWE) · Learning with Rounding (LWR) · Bounded Distance Decoding (BDD) strategy · Modulus switching · Lattice basis reduction · Babai's Nearest Plane (BNP) algorithm.

1 Introduction

The LWR problem introduced by Banerjee, Peikert and Rosen in [6] is a derandomization variant of the well-known LWE problem in which a (q, p)-modulo rounding function (denoted by $\lfloor \cdot \rceil_{q,p}$) is used to hide the secret instead of using an error e drawn from some distribution. Specifically, for $x \in \mathbb{Z}_q$ we have $\lfloor x \rceil_{q,p} =$

$\lfloor (p/q) \cdot x \rceil \in \mathbb{Z}_p$, where $\lfloor x \rceil$ rounds the real number x to the nearest integer. Let $n \geq 1$, $q \geq p \geq 2$, and given a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$, an LWR sample is the pair $(\mathbf{a}, c) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$, in which the vector $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ uniformly at random and $c := \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_{q,p} \in \mathbb{Z}_p$.

Several LWR-based cryptosystems have recently been submitted to NIST for Post-Quantum Cryptography Standardization, for instance Round2 [4] and Lizard [12]. In their proposals, for efficiency goal, the secret is sampled from (even sparse) small sets. For example, in [12], the authors consider the binary secrets $\{0,1\}^*$ or trinary secrets $\{-1,0,1\}^*$ or even the secret $\mathbf{s}$ is sampled from some uniform distribution over $\{-1,0,1\}^*$ in which $\mathbf{s}$ contains h nonzeros for some fixed integer $h > 0$.

A typical approach in attacking LWR is to transfom an LWR instance to an LWE modulo q. More specifically, the following method, called *q-reduction*, is used to lift an LWR instance $(\mathbf{a}, c = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_{q,p}) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$ to an LWE instance modulo q of the form $(\mathbf{a}, c' = \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where $c' = \left\lfloor \frac{q}{p} c \right\rceil$ and $e \in \left(-\frac{q}{2p}, \frac{q}{2p} \right]$. Most of techniques for solving LWR are adapted from attacks against an LWE modulo q instance [4,12,13] such as SIS strategy, BDD strategy, uSVP strategy, algebraic strategy [3] and so on (cf. [2] for more details). In this work, we just focus on the BDD strategy which aims to reduce a search-LWE instance to the closest vector problem (CVP), then use a practical lattice basis reduction algorithm (such as Lenstra-Lenstra-Lovász (LLL) algorithm [19] or the blockwise Korkine-Zolotarev (BKZ) algorithm [23]) along with the Babai's Nearest Plane (BNP) algorithm [5] to solve this CVP instance.

Some questions can be raised here that if it is possible to transform an $\mathrm{LWR}_{m,n,q,p}(\chi_s)$ instance to an LWE instance modulo q' (called *q′-approach*) other than an LWE modulo q (called *q-approach*), with $q' < q$ or even $q' > q$, that if the q'-approach is better for specific attacks (such as the BDD strategy we are going to focus on) than the q-approach and that if so, how to choose the optimal q'. Note that, in the case of $q' < q$, the transformation is the so-called *modulus switching* technique. The technique allows to transform an LWE modulo q instance to an LWE modulo q' instance with q' is typically chosen as

$$q' \approx \frac{\sigma_s}{\sigma} \cdot \sqrt{\frac{n}{12}} \cdot q, \tag{1}$$

where n is the length of the secret and σ_s, σ are standard deviations of the secret and the error of the original LWE mod q instance, respectively (cf. [2, Lemma 2] for more details). The modulus switching technique was used for the first time aiming to speed up the homomorphic encryption operations [10]. Then the technique was also used to evaluate the classical hardness of LWE problem [9]. Recently, the technique was modified to combine with BKW algorithm on LWE [1]. Until now, however, the effect of the technique on other attacks against LWE and LWR has not been studied carefully.

Our contribution. We evaluate the impact of modulus switching on the BDD strategy in solving search LWR problem. We achieved the following:

1. We reduce an $\mathrm{LWR}_{m,n,q,p}(\chi_s)$ instance to an LWE modulo q instance and obtain an induced error e. Some previous works stated that the error *heuristically* follows a uniform distribution over $[-q/2p, q/2p)$ (see, e.g., [12, Sec. 4.2.1]) then its variance is roughly $\frac{q^2}{12p^2}$. We will confirm that the error actually follows a discrete uniform distribution and hence we compute its variance that is more precisely equal to $\frac{q^2+2pq}{12p^2}$, which is much larger than $\frac{q^2}{12p^2}$ if $q \gg p$. This helps us to estimate q' (mentioned below) more exactly than previous works.
2. We also determine the successful range in which LWR instances can be solved by the BDD strategy using practical lattice reduction algorithms (e.g., LLL or BKZ) accompanied with the BNP algorithm [5].
3. We theoretically and experimentally convince that one can transform an $\mathrm{LWR}_{m,n,q,p}(\chi_s)$ instance to an LWE modulo $q' \approx \sqrt{\frac{(m-n)(n\sigma_s^2+1)p^2q^2}{n(q^2+2q)}}$ which can be solved more "efficiently" by the BDD strategy under a sufficient condition. By "efficiently", we mean that we will have either higher success probability and smaller root Hermite factor or/and smaller running time in solving this LWE instance depending on how much the size of α is. Our approach is especially suitable for LWR instances with short secret, i.e., σ_s small.

Our main technical tool is a heuristic evaluation on the success probability of the BDD strategy. To the best of our knowledge, this work is the first attempt to evaluate carefully the modulus switching's effect to the BDD strategy on LWR problem. We expect that our work will provide with a different perspective in exploiting modulus switching technique not only to solve LWR (even LWE) (by the BDD strategy or even other strategies) but also in other application scenarios.

Notation. If $A = \{a_1, \cdots, a_m\}$ with $a_i \in \mathbb{R}$ then $k \cdot A = \{k \cdot a_1, \cdots, k \cdot a_m\}$ for any $k \in \mathbb{R}$. The logarithm of base 2 (resp., the natural logarithm) of a positive real number x will be written as $\log(x)$ (resp., $\ln(x)$). We use $\mathcal{U}(A)$ to indicate the uniform distribution over the set A. The rounding operation $\lfloor a \rceil$ outputs the integer closest to a and in the case of a tie, it outputs the integer next to a. For any positive integer q, we denote by $\mathbb{Z}_q = \{0, 1, \cdots, q-1\}$ the set of integers modulo q. We write $x \leftarrow \chi$ to say that the random variable x follows the probability distribution χ or x is sampled from the distribution χ. For a real number k, the notation $y \leftarrow k \cdot \chi$ means that $y = k \cdot x$ for some x that follows the probability distribution χ.

2 Preliminaries

2.1 Lattices

The lattice $\mathcal{L} = \mathcal{L}(\mathbf{A})$ generated by the column matrix $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_m] \in \mathbb{Z}^{n \times m}$ of m linearly independent vectors is defined to be the set of all linear integral combinations of $\mathbf{a}_i$'s, i.e., $\mathcal{L}(\mathbf{A}) = \{\mathbf{A}.\mathbf{x} : \mathbf{x} \in \mathbb{Z}^m\} = \{\sum_{i=1}^{m} x_i \mathbf{a}_i : x_i \in \mathbb{Z}\}$. We call the matrix $\mathbf{A}$ *a basis* of $\mathcal{L}$ and call each $\mathbf{a}_i$ *a basis vector*. The rank of the lattice is the number of basis vector (i.e., m). The dimension of the lattice is the number of entries in each basis vector (i.e., n). If $m = n$, the lattice is called to be *full-rank*. Note that, every lattice has infinitely many bases up to a unimodular matrix of determinant ± 1. Hence, if $\mathbf{A}$ and $\mathbf{B}$ are two different bases of the lattice $\mathcal{L}$, then $\det(\mathbf{A}^T\mathbf{A}) = \det(\mathbf{B}^T\mathbf{B})$. We call $\det(\mathcal{L}(\mathbf{A})) := \sqrt{\det(\mathbf{A}^T\mathbf{A})}$ the *determinant* (or *volume*) of the lattice $\mathcal{L}(\mathbf{A})$.

The Gram-Schmidt matrix $\mathbf{A}^* = \{\mathbf{a}_1^*, \cdots, \mathbf{a}_m^*\}$ for a basis $\mathbf{A} = \{\mathbf{a}_1, \cdots, \mathbf{a}_m\}$ is defined by setting $\mathbf{a}_1^* = \mathbf{a}_1$ and $\mathbf{a}_i^* = \mathbf{a}_i - \sum_{j=1}^{i-1} \frac{\langle \mathbf{a}_i, \mathbf{a}_j^* \rangle}{\|\mathbf{a}_j^*\|^2}, i = 2, \cdots, m$. Note that $\det(\mathcal{L}(\mathbf{A})) = \prod_{i=1}^{m} \|\mathbf{a}_i^*\|$. The *fundamental parallelepiped* asscociated with a basis $\mathbf{A} = \{\mathbf{a}_1, \cdots, \mathbf{a}_m\}$ is $\mathcal{P}_{1/2}(\mathbf{A}) = \left\{\sum_{i=1}^{m} x_i \mathbf{a}_i : x_i \in \left[-\frac{1}{2}, \frac{1}{2}\right)\right\}$. We define the fundamental parallelepiped $\mathcal{P}_{1/2}(\mathbf{A}^*)$ for the Gram-Schmidt matrix $\mathbf{A}^*$ in the same way.

For integers q, m, n ($m \geq n$), given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, we consider the so-called *q-ary lattice* $\Lambda_q(\mathbf{A}) = \{\mathbf{u} \in \mathbb{Z}^m : \mathbf{u} = \mathbf{A}\mathbf{s} \bmod q \text{ for some } \mathbf{s} \in \mathbb{Z}^n\}$. It is well known that $\det(\Lambda_q(\mathbf{A})) = q^{m-n}$ with high probability.

2.2 Lattice Basis Reduction Algorithms and Root Hermite Factor

A basis of a lattice can be reduced using the so-called *lattice basis reduction* (LBR) algorithms to obtain a new basis consisting of short and nearly orthogonal lattice vectors. Such two algorithms typically used in practice are LLL [19] and BKZ [23]. The former is a polynomial-time algorithm and the latter is a block version of the former with exponential complexity.

Let $\mathcal{L}$ be a lattice of rank m and $\mathbf{A}$ be a reduced lattice basis obtained using some LBR algorithm, say $\mathcal{A}$, the *root Hermite factor* (rHF) $\delta_{\mathcal{A}}$ of $\mathcal{A}$ with respect to $\mathbf{A}$ is the constant given by

$$\delta_{\mathcal{A}} = \left(\frac{||\mathbf{u}_1||}{\det(\mathcal{L})^{1/m}}\right)^{\frac{1}{m}}, \tag{2}$$

where $\mathbf{u}_1$ is the shortest non-zero vector of $\mathbf{A}$. Gama and Nguyen in [15] attempted to estimate the rHF of LLL and BKZ for random matrices. Namely, they estimated that the rHF of LLL is $\delta_{\text{LLL}} \approx 1.0219$ on average in high dimension ≥ 100 while that of BKZ with blocksize $\beta = 20$ [1] is $\delta_{\text{BKZ}} \approx 1.0128$.

Unfortunately, however, these experimental results of [15] for random matrices may be not perfectly fit for q-ary lattices. That is the reason why Kudo et

[1] BKZ of blocksize 20 is usually used in practice because of its time/quality trade-off property.

al. in [17] conducted intensively an experiment on q-ary lattices to estimate the quantity $\min_{i=1}^{m} \|\mathbf{b}_i^*\|$ from which they defined an alternative measure as follows:

$$c_{\mathcal{A}} := \left(\frac{\min_{i=1}^{m} \|\mathbf{b}_i^*\|}{\det(\Lambda_q(\mathbf{A}))^{1/m}} \right)^{\frac{1}{m}}, \tag{3}$$

where $\mathbf{b}_i^*$'s are Gram-Schmidt vectors of a basis of the q-ary lattice $\Lambda_q(\mathbf{A})$, say $\mathbf{B} = \{\mathbf{b}_1, \cdots, \mathbf{b}_m\}$, that is already reduced by some LBR algorithm $\mathcal{A}$. Note that $c_{\mathcal{A}} \leq 1$ since $\min_{i=1}^{m} \|\mathbf{b}_i^*\| \leq (\prod_{i=1}^{m} \|\mathbf{b}_i^*\|)^{1/m} = \det(\Lambda_q(\mathbf{A}))^{1/m}$. Especially, Kudo et al. [17] estimated that $c_{\mathrm{LLL}} = 0.9775$ whereas using BKZ with blocksize $\beta = 20$, they got $c_{\mathrm{BKZ}} = 0.9868$ (cf. [17, Table 1]).

If we still denote the rHF for q-ary lattices by $\delta_{\mathcal{A}}$ then it seems that $\delta_{\mathcal{A}} \approx 1/c_{\mathcal{A}}$. For instance, with $c_{\mathrm{LLL}} = 0.9775$ and $c_{\mathrm{BKZ}} = 0.9868$, we have $1/c_{\mathrm{LLL}} = 1.0230$ and $1/c_{\mathrm{BKZ}} = 1.0139$, respectively, that are quite close to the rHF for random matrices mentioned above.

We will use (3) to reach an important heuristic that is useful for our work (see Subsection 2.4).

2.3 Probability

Variance of Random Variables. We denote the variance of a random variable X by σ_X. For $a, b \in \mathbb{Z}$, the variance of a random variable X following the discrete uniform distribution $\mathcal{U}(\{a, a+1, \cdots, b-1, b\})$ is $\sigma_X^2 = ((b-a+1)^2 - 1)/12$. If X follows the continuous uniform distribution $\mathcal{U}(a, b)$ then $\sigma_X^2 = (b-a)^2/12$. Assuming that $Z = X + Y$ where X, Y are independent random variables then $\sigma_Z^2 = \sigma_X^2 + \sigma_Y^2$. Finally, for every random variable X and for every constant $k \in \mathbb{R}$, let $Y = kX$, then we have $\sigma_Y^2 = k^2 \sigma_X^2$.

Gaussian Distribution. For any real numbers $\mu, x \in \mathbb{R}$ and any real number $s > 0$, the one dimensional continuous Gaussian distribution $\mathcal{D}_{\mu,\sigma}$ of mean μ and variance σ^2 is defined by its probability density function (pdf) $\mathcal{D}_{\mu,\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \forall x \in \mathbb{R}$.

Convolution of Two Distributions. Let X and Y be continuously distributed independent random variables with pdfs f_X and f_Y. Then the pdf of the random variable $Z = X + Y$ is the convolution of f_X and f_Y given by

$$f_Z(z) = (f_X * f_Y)(z) = \int_{-\infty}^{+\infty} f_X(t) f_Y(z-t) dt = \int_{-\infty}^{+\infty} f_X(z-t) f_Y(t) dt. \tag{4}$$

2.4 Search-LWE and BDD Strategy

LWE problem proposed in [22] has been playing important role in lattice-based cryptography. The hardness of most of the lattice-based cryptosystems is based on the LWE problem which has two versions: one is decision version and another is search version. Here, we recall the search version of the LWE problem.

Definition 1 (Search-LWE Problem). *Given positive integers n, q, a fixed secret vector $\mathbf{s}$ whose each component is sampled from some distribution χ_s and another probability distribution χ_e on $\mathbb{Z}$, the search-LWE$_{m,n,q,\chi_s,\chi_e}$ problem is to find the secret $\mathbf{s}$ from m LWE samples of the form $(\mathbf{a}, c = \langle \mathbf{a}, \mathbf{s} \rangle + e \ mod \ q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where $\mathbf{a} \in \mathbb{Z}_q^n$ is drawn uniformly at random, and the error term $e \leftarrow \chi_e$.*

If we have such m samples $(\mathbf{a}_i, c_i = \langle \mathbf{a}, \mathbf{s} \rangle + e_i)$ for $i = 1, \cdots, m$, we can collect them as $(\mathbf{A}, \mathbf{c} = \mathbf{A}.\mathbf{s} + \mathbf{e})$ in which $\mathbf{A}$ is an $(m \times n)$- matrix whose i-th row is $\mathbf{a}_i$, $\mathbf{c} = (c_1, c_2, \cdots, c_m)^T$, and $\mathbf{e} = (e_1, e_2, \cdots, e_m)^T$.

BDD Strategy. The BDD strategy for solving search-LWE proposed by Lindner and Peikert [20] is based on the close relation between search-LWE problem and BDD problem. Given a lattice and a target vector unusually close to the lattice, BDD problem asks to find the lattice vector closest to the target.

Let $(\mathbf{A}, \mathbf{c} = \mathbf{A}\mathbf{s} + \mathbf{e} \mod q) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ be a search-LWE$_{m,n,q,\chi_s,\chi_e}$ instance. Also let $\Lambda_q(\mathbf{A}) = \{\mathbf{u} \in \mathbb{Z}^m : \mathbf{u} = \mathbf{A}\mathbf{s} \bmod q \text{ for some } \mathbf{s} \in \mathbb{Z}^n\}$ be the q-ary lattice spanned by columns of $\mathbf{A}$ and we call it *the associated q-ary lattice* of the search-LWE problem $(\mathbf{A}, \mathbf{c})$. To solve search-LWE problem via the BDD strategy, we reduce it to a CVP. In fact, if the error $\mathbf{e}$ is sufficiently short then $\mathbf{c}$ is closest to some lattice point $\mathbf{u} = \mathbf{A}\mathbf{s} \bmod q \in \Lambda_q(\mathbf{A})$ since we have $\mathbf{e} = \mathbf{c} - \mathbf{A}\mathbf{s}$. Thus, finding the secret $\mathbf{s}$ is equivalent to finding $\mathbf{u}$, i.e., solving a CVP problem over q-ary lattice $\Lambda_q(\mathbf{A})$.

The most basic tools used in solving search-LWE via BDD strategy are some basis reduction algorithm, say $\mathcal{A}$ (e.g., LLL or BKZ), and the BNP algorithm. The BNP algorithm takes as input the vector $\mathbf{c}$ and a basis $\mathbf{B} = \{\mathbf{b}_1, \cdots, \mathbf{b}_m\}$ of $\Lambda_q(\mathbf{A})$ that is already reduced by $\mathcal{A}$. Finally, the BNP algorithm outputs the lattice point $\mathbf{u} \in \Lambda_q(\mathbf{A})$ such that $\mathbf{e} = \mathbf{c} - \mathbf{u} \in \mathcal{P}_{1/2}(\mathbf{B}^*)$.

From now on, by "BDD strategy" or "BDD attack", we mean the BDD strategy associated with the BNP algorithm.

Success Probability of the BDD Strategy. The success probability of the BDD strategy in solving search-LWE is measured by the probability of the event that the error $\mathbf{e}$ lies in $\mathcal{P}_{1/2}(\mathbf{B}^*)$. Depending on which distribution the error $\mathbf{e}$ follows, we have some formulas to compute the probability in literature: (i) if $\mathbf{e}$ is uniform then we can estimate the probability by

$$\Pr\left[\mathbf{e} \in \mathcal{P}_{1/2}(\mathbf{B}^*)\right] = \prod_{i=1}^{m} \left(\frac{\rceil \|\mathbf{b}_i^*\|}{2\sigma_e \sqrt{3}} \right), \tag{5}$$

(ii) in the case of a Gaussian error $\mathbf{e}$, we can use the formula taken from [20]

$$\Pr\left[\mathbf{e} \in \mathcal{P}_{1/2}(\mathbf{B}^*)\right] = \prod_{i=1}^{m} \operatorname{erf}\left(\frac{\rceil \|\mathbf{b^*}_i\|}{2\sigma_e \sqrt{2}} \right), \tag{6}$$

where σ_e^2 is the variance of the error $\mathbf{e}$ according to its distribution and $\operatorname{erf}(\cdot)$ is the Gaussian error function $\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp(-t^2) dt, z \in [0, +\infty]$. However,

(5) and (6) are really not helpful for our work [2]. Therefore, we have to look for another way to estimate the success probability for the BDD strategy regardless of the error's distribution. Fortunately, we can use a heuristic analysis appeared in [17] as follows:

We have $\Pr\left[\mathbf{e} \in \mathcal{P}_{1/2}(\mathbf{B}^*)\right] = \Pr\left[|\langle \mathbf{e}, \mathbf{b}_i^* \rangle| < \|\mathbf{b}_i^*\|^2/2, \forall i = 1, ..., m\right]$. Using the heuristics $|\langle \mathbf{e}, \mathbf{b}_i^* \rangle| \approx \|\mathbf{e}\| \cdot \|\mathbf{b}_i^*\|/\sqrt{m}$ and $\|\mathbf{e}\| \approx \sigma_e \cdot \sqrt{m}$, we have $2\sigma_e \leq \|\mathbf{b}_i^*\|$ for all $i = 1, ..., m$, which is equivalent to

$$2\sigma_e \leq \min_{i=1,...,m} \|\mathbf{b}_i^*\|. \tag{7}$$

Combining (7) with (3) yields the following heuristic that will be very useful for our work:

Heuristic 1. *Let $c_{\mathcal{A}}$ is defined as in* (3)*. Heuristically, the success probability for the BDD strategy in solving search-LWE problem $(\mathbf{A}, \mathbf{c} = \mathbf{As} + \mathbf{e} \mod q) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m$ can be measured by the probability $\Pr\left[2\sigma_e \leq c_{\mathcal{A}}^m \cdot \det(\Lambda_q(\mathbf{A}))^{1/m}\right]$, namely, $\Pr\left[2\sigma_e \leq c_{\mathcal{A}}^m \cdot q^{(m-n)/m}\right]$ which is equivalent to*

$$\Pr\left[\frac{q^{m-n}}{\sigma_e^m} \geq \frac{2^m}{c_{\mathcal{A}}^{m^2}}\right], \tag{8}$$

where $\mathcal{A}$ is the LBR algorithm used within the BDD strategy.

BDD Strategy and Root Hermite Factor. It is conventional that the quality of the reduced basis (which is characterized by the rHF obtained using by some LBR algorithm) has the most significant effect on the success probability of the BNP algorithm (see, e.g., [2, Section 5.4], [7, 16]), hence the BDD strategy. Namely, smaller rHF means that the corresponding basis is reduced better, hence the BNP algorithm may return the closest vector more precisely, so the efficacy of the BDD strategy may be higher.

Assume the BDD strategy uses the LBR algorithm named $\mathcal{A}$. And suppose that we want to compare the efficacy of the BDD strategy in solving a search-LWE problem $(\mathbf{A}_1, \mathbf{c}_1)$ with that in solving a search-LWE problem $(\mathbf{A}_2, \mathbf{c}_2)$. Then instead of success probability, we can compare the rHFs of $\mathcal{A}$ with respect to the reduced bases of the associated q-ary lattices $\Lambda_q(\mathbf{A}_1)$ and $\Lambda_q(\mathbf{A}_2)$.

2.5 Search LWR and a Reduction from LWR to LWE

Let p, q be two moduli such that $2 \leq p \leq q$. Recall that, the (q, p)-modulo rounding operation, denoted by $\lfloor . \rceil_{q,p}$, is defined as follows: for $x \in \mathbb{Z}_q$ $\lfloor x \rceil_{q,p} = \lfloor (p/q) \cdot x \rceil \in \mathbb{Z}_p$. We can extend the operation for vectors, matrices by taking it component-wise, such as for $\mathbf{x} = (x_1, \cdots, x_n) \in \mathbb{Z}_q^n$, we have $\lfloor \mathbf{x} \rceil_{q,p} = (\lfloor x_1 \rceil_{q,p}, \cdots, \lfloor x_n \rceil_{q,p})$.

[2] We cannot use either (5) or (6) if the error $\mathbf{e}$ has a complex behavior. Such a kind of error is the q'-error that we will see in Section 5.

Definition 2 (LWR Sample). *For a secret vector* $\mathbf{s} \leftarrow \chi_s$ *where* χ_s *is some distribution of variance* σ_s^2*, an* $LWR_{m,n,q,p}(\chi_s)$ *sample is obtained by choosing a vector* $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ *uniformly at random and outputting* $(\mathbf{a}, c = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_{q,p}) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$.

Assuming that we have such m LWR samples then we can write them as $(\mathbf{A}, \mathbf{c} = \lfloor \mathbf{As} \rceil_{q,p}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_p^m$, where $\mathbf{A}$ is a matrix whose rows are $\mathbf{a}_i$ and $\mathbf{c}$ is a column vector whose elements are c_i.

Definition 3 (Search-LWR). *Given* m *LWR samples* $(\mathbf{A}, \mathbf{c} = \lfloor \mathbf{As} \rceil_{q,p}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_p^m$*, the search-*$LWR_{m,n,p,q}(\chi_s)$ *problem is to find the secret* $\mathbf{s}$.

Reducing $\mathbf{LWR_{m,n,q,p}(\chi_s)}$ to LWE Modulo q. The following reduction, called *q-reduction*, is used to transform an LWR instance consisting of samples of the form $(\mathbf{a}, c = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_{q,p}) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$ to an LWE modulo q instance of the form $(\mathbf{a}, c_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ in which

$$\begin{aligned} c_1 &= \left\lfloor \frac{q}{p} \cdot c \right\rceil \bmod q \\ &= \left\lfloor \frac{q}{p} . \left(\frac{p}{q} \cdot (\langle \mathbf{a}, \mathbf{s} \rangle + qu) + e_1 \right) \right\rceil \bmod q, \text{with } e_1 \in \left(-\frac{1}{2}, \frac{1}{2} \right], u \in \mathbb{Z}, \\ &= (\langle \mathbf{a}, \mathbf{s} \rangle + e) \bmod q = \langle \mathbf{a}, \mathbf{s} \rangle \bmod q + e, \text{ where } e := \lfloor (q/p) \cdot e_1 \rceil . \end{aligned} \tag{9}$$

For short, we call the error e *q-error*.

Note that, in the reduction above, we used the following assumption:

Assumption 1. *We assume that in our work, the error induced by reduction from an LWR instance to a corresponding LWE instance is not changed by a modulo operation. Formally, for an LWE sample* $(\mathbf{a}, (\langle \mathbf{a}, \mathbf{s} \rangle + e) \bmod q) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$*, we assume that*

$$(\langle \mathbf{a}, \mathbf{s} \rangle + e) \bmod q = \langle \mathbf{a}, \mathbf{s} \rangle \bmod q + e.$$

This assumption was also used in many previous works relating to LWE such as [8, 10, 11, 17, 18]. We still use this assumption in Section 5.

3 Distribution of the q-Error

The q-reduction above has been typically considered in recent works relating to the LWR problem (see, e.g., [4,12,14]). In these works, it is heuristically assumed that e is continuously uniform over $(-q/(2p), q/(2p)]$. If so, the variance is just $\sigma^2 \approx q^2/(12p^2)$. In this section, we show that e is actually distributed according to a discrete uniform distribution over the set $\{-\lfloor \lfloor q/2 \rfloor / p \rceil, \cdots, \lfloor \lfloor q/2 \rfloor / p \rceil\}$, hence its variance is actually $\sigma^2 \approx (q^2 + 2pq)/(12p^2)$ which is significantly greater than $q^2/(12p^2)$ in the case $q \gg p$.

To begin with, we state a simple lemma on rounding operation via discrete uniform distribution.

Lemma 1. *Given a nonzero real number b. Set $b_0 := -\lfloor b \rceil$, $b_1 := -\lfloor b \rceil + 1$, $\cdots, b_{t-1} := \lfloor b \rceil - 1$ and $b_t := \lfloor b \rceil$ and define the set $B := \{b_0, \cdots, b_t\}$. Let x be a real number taken uniformly at random from $[-b, b]$. Then*

$$\Pr[\lfloor x \rceil = b_i | b_i \in B] = \Pr[\lfloor x \rceil = b_j | b_j \in B], \text{ for all } i, j \in \{1, ..., t-1\}.$$

In particular,

$$\Pr[\lfloor x \rceil = b_0] = \Pr[\lfloor x \rceil = b_t] \leq \Pr[\lfloor x \rceil = b_i | b_i \in B], \forall i \in \{1, ..., t-1\}.$$

Proof. The idea for the proof is easy. Firstly, note that, for $1 \leq i \leq t-1$, we have

$$\Pr[\lfloor x \rceil = b_i] = \Pr[x \in [b_i - 1/2, b_i + 1/2)],$$

which implies the first statement in the lemma. Secondly, since $b_0 = -\lfloor b \rceil$ and $b_t = \lfloor b \rceil$, so $b_0 - 1/2 \leq -b < b_0 + 1/2$ and $b_t - 1/2 \leq b < b_t + 1/2$,

$$\Pr[\lfloor x \rceil = b_0] = \Pr[x \in [-b, b_0 + 1/2)] \leq \Pr[x \in [b_0 - 1/2, b_0 + 1/2)],$$

and

$$\Pr[\lfloor x \rceil = b_t] = \Pr[x \in [b_t - 1/2, b]] \leq \Pr[x \in [b_t - 1/2, b_t + 1/2)].$$

□

Now we give the theorem describing the behavior of q-error e.

Theorem 1. *Set $b := \frac{\lfloor q/2 \rfloor}{p}$ and $A := \{-\lfloor b \rceil + 1, \cdots, \lfloor b \rceil - 1\}$. Also let e be the q-error defined as in (9). Then we have:*

$$\Pr[e = a | a \in A] = \frac{1}{2b}, \text{ and } \Pr[e = -\lfloor b \rceil] = \Pr[e = \lfloor b \rceil] = \frac{b - \lfloor b \rceil + \frac{1}{2}}{2b} \leq \frac{1}{2b}.$$

However, less precisely, we can say that e is uniform over

$$B := \{-\lfloor b \rceil, -\lfloor b \rceil + 1, \cdots, \lfloor b \rceil - 1, \lfloor b \rceil\}.$$

Then the variance of q-error e is

$$\sigma^2 \approx \frac{\left(2\left\lfloor \frac{\lfloor q/2 \rfloor}{p} \right\rceil + 1\right)^2 - 1}{12} \approx \frac{q^2 + 2qp}{12p^2}. \tag{10}$$

Proof. First, we show that the error e_1 appears in (9) is distributed uniformly over $\frac{1}{q} \cdot \{-\lfloor q/2 \rfloor, \cdots, \lfloor q/2 \rfloor\}$. Note that, we fix the secret $\mathbf{s}$ which is sampled from some probability distribution χ_s beforehand. It is true that if we take $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random then $(\langle \mathbf{a}, \mathbf{s} \rangle \bmod q)$ is also uniform over $\mathbb{Z}_q$. Hence, $\frac{p}{q} \cdot (\langle \mathbf{a}, \mathbf{s} \rangle \bmod q)$ is also uniform over $\frac{p}{q} \cdot \mathbb{Z}_q = \{0, p/q, \cdots, p \cdot (q-1)/q\}$. Suppose that $\langle \mathbf{a}, \mathbf{s} \rangle \bmod q = k$, for some $k \leftarrow \mathcal{U}(\{0, \cdots, q-1\})$. There always exist integers w and v such that $kp = qw + v$, $-\lfloor q/2 \rfloor \leq v \leq \lfloor q/2 \rfloor$, and $0 \leq w \leq p$. Certainly, v is uniform over the set $\{-\lfloor q/2 \rfloor, \cdots, \lfloor q/2 \rfloor\}$. Thus,

$(p/q) \cdot (\langle \mathbf{a}, \mathbf{s} \rangle \bmod q) = kp/q = w + v/q$, where $-\lfloor q/2 \rfloor/q \leq v/q \leq \lfloor q/2 \rfloor/q$. Consequently, $\lfloor (p/q) \cdot (\langle \mathbf{a}, \mathbf{s} \rangle \bmod q) \rceil = w \in \{0, \cdots, p\}$, and hence

$$e_1 := \frac{p}{q}(\langle \mathbf{a}, \mathbf{s} \rangle \bmod q) - \left\lfloor \frac{p}{q}(\langle \mathbf{a}, \mathbf{s} \rangle \bmod q) \right\rceil = \frac{v}{q}$$

is uniform over $(1/q) \cdot \{-\lfloor q/2 \rfloor, \cdots, \lfloor q/2 \rfloor\}$. As a result, $(q/p) \cdot e_1 = v/p$ is uniform over $\{-\lfloor q/2 \rfloor/p, \cdots, \lfloor q/2 \rfloor/p\}$.

Recall that, the q-error $e = \lfloor (q/p) \cdot e_1 \rceil$. Applying Lemma 1 to $b := \lfloor q/2 \rfloor/p$, $B := \{-\lfloor b \rceil, -\lfloor b \rceil + 1, \cdots, \lfloor b \rceil - 1, \lfloor b \rceil\}$ and $x := (q/p) \cdot e_1$ and $e := \lfloor x \rceil$, the theorem follows. The variance of e is computed using the discrete uniform distribution over $B \subset \mathbb{Z}$. □

4 Estimating the Successful Range for BDD Strategy in Solving LWR

Our purpose in this section is to find a condition of q so as to the $\mathrm{LWR}_{m,n,q,p}(\chi_s)$ instance can be solved by the BDD strategy. The condition depends only on dimension n, the used LBR algorithm $\mathcal{A}$ (through its constant $c_{\mathcal{A}}$ defined as in (3)) as well as the bit ratio between q and p. According to the q-reduction, we say that a search-$\mathrm{LWR}_{m,n,q,p}(\chi_s)$ instance is solvable by the BDD strategy if the corresponding LWE modulo q can be solved by the strategy.

Let $(\mathbf{a}, c = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_{q,p}) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$ be an $\mathrm{LWR}_{m,n,q,p}(\chi_s)$ instance and its corresponding LWE instance defined as in (9), the q-error e has variance σ defined as in (10). By Heuristic 1, we need $2\sigma \leq c_{\mathcal{A}}^m \cdot q^{(m-n)/m}$ happens with probability 1. Note that $\sigma = \sqrt{\frac{q^2+2qp}{12p^2}} \leq \frac{q}{2p}$ as $pq \leq q^2$, yielding that $2\sigma \leq \frac{q}{p}$. To estimate the successful range, we should consider the following slightly stronger condition

$$\frac{q}{p} \leq c_{\mathcal{A}}^m \cdot q^{(m-n)/m},$$

from which we obtain $\frac{q^n}{p^m} \leq c_{\mathcal{A}}^{m^2}$. Given $0 < \alpha < 1$ such that $p = q^\alpha$, from the previous equation we get

$$(m\alpha - n)\log(q) \geq -m^2 \log(c_{\mathcal{A}}). \tag{11}$$

It is easy to see that (11) just makes sense as long as $m > \frac{n}{\alpha}$. So with this condition, we can rewrite (11) as $\log(q) \geq \frac{-m^2 \log(c_{\mathcal{A}})}{m\alpha - n}$. Hence we have that

$$\log(q) \geq \frac{-4n \log(c_{\mathcal{A}})}{\alpha^2},$$

since $\min_m \left\{ \frac{-m^2 \log(c_{\mathcal{A}})}{m\alpha - n} \right\} = \frac{-4n \log(c_{\mathcal{A}})}{\alpha^2}$ obtained at $m = \frac{2n}{\alpha}$. For given α, let $q_{\min}$ be the integer such that

$$\log(q_{\min}) = \lceil -4n \log(c_{\mathcal{A}})/\alpha^2 \rceil, \tag{12}$$

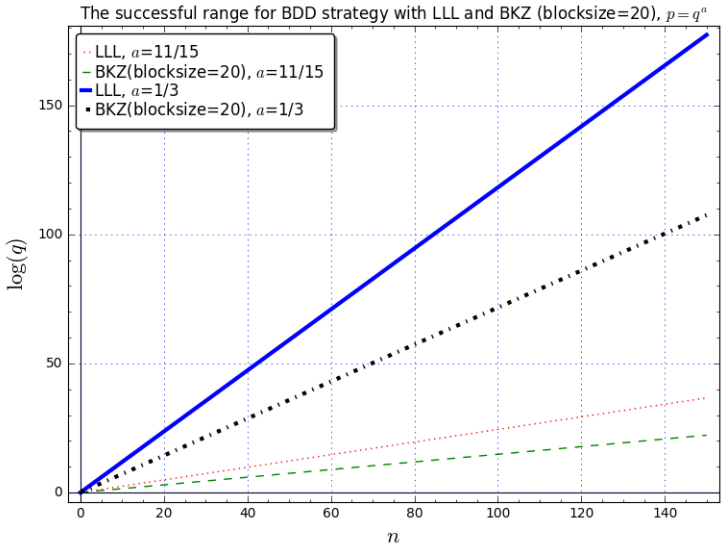

Fig. 1: We plot the graphs of the lines $\log(q_{\min}) = \lceil -4n\log(c_{\mathcal{A}})/\alpha^2 \rceil$ corresponding to $\alpha = 11/15$ and $\alpha = 1/3$. For LLL, we use $c_{\mathrm{LLL}} = 0.9775$ and for BKZ of block size 20 we use $c_{\mathrm{BKZ}} = 0.9868$.

then $\log(q_{\min})$ is a function in n whose graph is a straight line (called *the boundary line*). The line divides the plane into two half-planes: the upper half-plane indicates the successful range in which LWR instances are solvable by the BDD strategy, while the lower half-plane indicates the failure range in which LWR instances are unsolvable by the BDD strategy, (see Fig. 1).

Remark 1. It is easy to see from (12) that either n grows or/and α decreases makes the value $\log(q_{\min})$ increase. This seems to mean that large n and/or smaller α should be chosen for LWR-based cryptosystems. However, as we will see later in our experiments, larger n and/or smaller α provide our mudulus switching approach with more advantages (see Sections 5, 6).

Optimal Number of LWR Samples for BDD Strategy. The optimal number of LWR samples m should be chosen such that the right hand side of $2\sigma \leq c_{\mathcal{A}}^m q^{\frac{m-n}{m}}$ is maximum. So the optimal value of m should be:

$$m = \left\lfloor \sqrt{\frac{n\log(q)}{-\log(c_{\mathcal{A}})}} \right\rceil. \tag{13}$$

Remind that, the optimal number of samples typically used in attacking LWE problems (e.g., see [20, 21]) is $m = \left\lfloor \sqrt{\frac{n\log(q)}{\log(\delta_{\mathcal{A}})}} \right\rceil$, which along with (13) again convince us that $\delta_{\mathcal{A}} \approx 1/c_{\mathcal{A}}$ (see Subsection 2.2).

5 Modulus Switching for BDD Strategy on LWR

We will analyze the so-called q'-reduction which reduces an $\mathrm{LWR}_{m,n,q,p}(\chi_s)$ instance to an LWE modulo q' instance. Then we estimate the optimal q' and an associated condition such that the BDD strategy on LWE modulo q' instance is more efficient than the BDD strategy on LWE modulo q instance.

5.1 Reducing $\mathbf{LWR}_{m,n,q,p}(\chi_s)$ to LWE Modulo q'

Let $(\mathbf{a}, c = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_{q,p}) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$ be an LWR sample. We reduce this LWR sample to the LWE sample modulo q' of the form $(\mathbf{a}', c' = \langle \mathbf{a}', \mathbf{s} \rangle + e') \in \mathbb{Z}_{q'}^n \times \mathbb{Z}_{q'}$ with $c' = \lfloor (q'/q) \cdot c \rceil$ clarified below where $\mathbf{a}' = \lfloor (q'/q) \cdot \mathbf{a} \rceil$. We call the error e' q'-*error*. We now take a closer look into the process of generating e'. Recall that, $c = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_{q,p} = (p/q) \cdot \langle \mathbf{a}, \mathbf{s} \rangle + p \cdot u + e_1$ for some $u \in \mathbb{Z}$, where $e_1 \leftarrow \frac{1}{q} \cdot \mathcal{U}(T)$ with $T = \{-\lfloor q/2 \rfloor, ..., \lfloor q/2 \rfloor\}$ (see (9) and the proof of Theorem 1).

Now with q'-reduction we will obtain

$$c_2 := \frac{q'}{p} \cdot c = \left\langle \left\lfloor \frac{q'}{q} \mathbf{a} \right\rceil, \mathbf{s} \right\rangle + \left\langle \frac{q'}{q} \mathbf{a} - \left\lfloor \frac{q'}{q} \mathbf{a} \right\rceil, \mathbf{s} \right\rangle + q' \cdot u + e_2,$$

where $e_2 := \frac{q'}{p} \cdot e_1 \leftarrow \frac{q'}{pq} \cdot \mathcal{U}(T)$. Hence

$$c' := \lfloor c_2 \rceil \bmod q' = \left\langle \left\lfloor \frac{q'}{q} \mathbf{a} \right\rceil, \mathbf{s} \right\rangle \bmod q' + \left\langle \frac{q'}{q} \mathbf{a} - \left\lfloor \frac{q'}{q} \mathbf{a} \right\rceil, \mathbf{s} \right\rangle + e_2 + e_3,$$

where $e_3 \in (-\frac{1}{2}, \frac{1}{2}]$. The q'-error is $e' := e_2 + e_3 + e_4$, with $e_4 := \left\langle \frac{q'}{q} \mathbf{a} - \left\lfloor \frac{q'}{q} \mathbf{a} \right\rceil, \mathbf{s} \right\rangle = \sum_1^n \left(\frac{q'}{q} a_i - \left\lfloor \frac{q'}{q} a_i \right\rceil \right) \cdot s_i$.

Distribution of q'-Error. The behavior of q'-error is mainly affected by that of e_2 (following a uniform distribution (cf. Section 3)) and e_4 (following a Gaussian distribution via Central Limit Theorem (cf. [2, Lemma 2])). Then, using the Equation (4) on the convolution of two distributions, the probability density function of e' can be approximated by

$$\begin{aligned} f(y) &= \frac{p\sqrt{6}}{q'\sqrt{\pi n \sigma_s^2}} \int_{-\frac{q'}{2p}}^{\frac{q'}{2p}} \exp\left(-\frac{6(x-y)^2}{n\sigma_s^2} \right) dx = \frac{p}{q'\sqrt{\pi}} \int_{\frac{\sqrt{6}\left(-\frac{q'}{2p} - y\right)}{\sqrt{n}\sigma_s}}^{\frac{\sqrt{6}\left(\frac{q'}{2p} - y\right)}{\sqrt{n}\sigma_s}} \exp(-\zeta^2) d\zeta \\ &= \frac{p}{2q'} \cdot \left[\mathrm{erf}\left(\frac{\sqrt{6}\left(\frac{q'}{2p} - y\right)}{\sqrt{n}\sigma_s} \right) + \mathrm{erf}\left(\frac{\sqrt{6}\left(\frac{q'}{2p} + y\right)}{\sqrt{n}\sigma_s} \right) \right]. \end{aligned} \tag{14}$$

Its derivative is

$$f'(y) = \frac{\sqrt{6}p}{2\sqrt{\pi n} q' \sigma_s} \cdot \left[-\exp\left(\frac{-6\left(\frac{q'}{2p} - y\right)^2}{n\sigma_s^2} \right) + \exp\left(\frac{-6\left(\frac{q'}{2p} + y\right)^2}{n\sigma_s^2} \right) \right].$$

The function $f(y)$ is symmetric through origin and it has convex bell-shaped curve reaching its highest value $h(q') = (p/q') \cdot \mathrm{erf}\left((\sqrt{6}q')/(2p\sqrt{n}\sigma_s)\right)$ at $y = 0$. Note that the functions $f'(y)$ and $h(q')$ tend to 0 as q' increases. Thus, if $q' \gg p$, the error e' will tend to follow a uniform distribution. By contrast, when $q' \approx p$, the error e' will tend to be distributed via a Gaussian distribution.

The behavior of q'-error e' is complex, therefore we cannot use (5) or (6) to estimate the success probability of the BDD strategy in solving the LWE modulo q'. Also, on the other hand, we cannot compare the success probability of the BDD strategy on LWE modulo q' with that on LWE modulo q using the formulas (5) or (6). This is why we need to use Heuristic 1.

Variance of q'-Error. We consider the variances of e_2, e_3 and e_4. The variance of e_2 will be $\sigma_2^2 := \left(q'^2/(q^2p^2)\right) \cdot \left(\left((2\lfloor q/2 \rfloor + 1)^2 - 1\right)/12\right)$, the variance of e_3 is $\sigma_3^2 = \frac{1}{12}$. The variance σ_4^2 of e_4 can be approximated as sum of n summands in which each summand is uniform on $[-\frac{1}{2}\sigma_s, \frac{1}{2}\sigma_s]$ where σ_s is the variance of the secret $\mathbf{s}$. Hence $\sigma_4^2 = \frac{n}{12}\sigma_s^2$. We assume that e_2, e_3 and e_4 are three independent random variables, then the variance of e' is estimated by

$$\sigma'^2 := \sigma_2^2 + \sigma_3^2 + \sigma_4^2 \approx \frac{1}{12}\left(n\sigma_s^2 + \frac{q'^2}{q^2p^2}\left(q^2 + 2q\right) + 1\right) = \frac{1}{12}\left(Mq'^2 + N\right), \quad (15)$$

where $M := (q^2 + 2q)/(p^2q^2)$ and $N := n\sigma_s^2 + 1$.

5.2 Optimizing q' for BDD Strategy

Assume that we have m LWR samples each of which is a pair $(\mathbf{a}, c = \lfloor \langle \mathbf{a}, \mathbf{s} \rangle \rceil_{q,p}) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$. We reduce the LWR instance to:

- The LWE modulo q instance consists of m LWE samples of the form $(\mathbf{a}, c_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ where $c_1 = \langle \mathbf{a}, \mathbf{s} \rangle \bmod q + e$ where *q-error* e has variance of $\sigma^2 \approx \frac{q^2+2qp}{12p^2}$. We call this approach *q-approach*.
- The LWE modulo q' instance consists of m LWE samples of the form $(\mathbf{a}', c') \in \mathbb{Z}_{q'}^n \times \mathbb{Z}_{q'}$ where $c' = \langle \mathbf{a}', \mathbf{s} \rangle \bmod q' + e'$ and $\mathbf{a}' = \lfloor (q'/q) \cdot \mathbf{a} \rceil$, and the q'-error e' has variance of $\sigma'^2 \approx \frac{1}{12}\left(Mq'^2 + N\right)$, with $M = (q^2 + 2q)/(p^2q^2)$ and $N = n\sigma_s^2 + 1$. We call this approach *q'-approach*.

For short, we say that the success probability of the BDD strategy in solving the LWE modulo q instance (resp., the LWE modulo q instance) as *the success probability of the q-approach* (resp., the success probability of the q'-approach). Notice that, in this section, the number of samples m is arbitrary as long as $m > n$. However, m will be optimally chosen via (13) in our experiments.

Our goal is to choose q' such that the success probability of the q'-approach is highest and certainly higher than that of the q-approach. The key idea is to use Heuristic 1. According to the heuristic, the success probability of the q-approach is

$$\Pr\left[\frac{q^{m-n}}{\sigma^m} \geq \frac{2^m}{c_{\mathcal{A}}^{m^2}}\right]. \quad (16)$$

Similarly, that of the q'-approach is

$$\Pr\left[\frac{q'^{m-n}}{\sigma'^m} \geq \frac{2^m}{c_{\mathcal{A}}^{m^2}}\right]. \tag{17}$$

Set $P' := \frac{q'^{m-n}}{\sigma'^m}$ and $P := \frac{q^{m-n}}{\sigma^m}$. The Equations (16) and (17) say that if $P' \geq P$ then the success probability of the q'-approach will be higher than that of the q-approach. Hence in order to achieve our goal, we can choose q' such that $P' \geq P$ and

$$P' \text{ is maximum.} \tag{18}$$

The condition $P' \geq P$ is equivalent to

$$\frac{q'^{m-n}}{\sqrt{(Mq'^2+N)^m}} \geq \frac{q^{m-n}}{\sqrt{(\frac{q^2+2qp}{p^2})^m}}. \tag{19}$$

The Equation (19) can be rewritten as

$$q'^{2(m-n)} \cdot (q^2+2qp)^m \geq q^{2(m-n)} \cdot p^{2m} \cdot (Mq'^2+N)^m. \tag{20}$$

Now, it is the time to state the main theorem of this section.

Theorem 2. (i) *The optimal choice for q' satisfying* (18) *is*

$$q' \approx \sqrt{\frac{(m-n)(n\sigma_s^2+1)p^2q^2}{n(q^2+2q)}}. \tag{21}$$

(ii) *The sufficient condition under which the q'-approach has success probability higher than that of the q-approach is that*

$$(n\sigma_s^2+1)p^2 \leq q^2+2q. \tag{22}$$

Proof. Set $M := (q^2+2q)/(p^2q^2)$ and $N := n\sigma_s^2+1$ as in (15).

For (i), we have $P' = \frac{q'^{m-n}}{\sigma'^m} = \frac{q'^{m-n}\sqrt{12^m}}{\sqrt{(Mq'^2+N)^m}}$. Define $t := q'^2$, then $P'^2 = \frac{t^{m-n}12^m}{(Mt+N)^m}$. It is easy to see that $t_0 := \frac{(m-n)N}{nM}$ maximizes P'^2. So we should choose $q' \approx \sqrt{t_0} = \sqrt{\frac{(m-n)N}{nM}}$.

For (ii), by plugging $q'^2 = t_0$ into (20) and after some calculations, we get

$$[(m-n)N]^{m-n} \cdot (q^2+2qp)^m \cdot n^n \geq q^{2(m-n)} \cdot p^{2m} \cdot (mN)^m \cdot M^{m-n}. \tag{23}$$

Now taking the natural logarithm on (23) we have

$$\begin{aligned}&(m-n)\ln(m-n) + (m-n)\ln(N) + m\ln(q^2+2qp) + n\ln(n)\\ &\geq 2(m-n)\ln(q) + 2m\ln(p) + m\ln(m) + m\ln(N) + (m-n)\ln(M).\end{aligned}$$

Using the facts that $m\ln(q^2+2qp) \geq m\ln(q^2+2q)$, and that $(m-n)\ln(m-n)+n\ln(n) \geq m\ln(m)$ [3] along with some simple calculations, we imply the sufficient condition for (20) to hold is that $\ln(N) + [\ln(p^2) - \ln(q^2+2q)] \leq 0$ which is equivalent to

$$(n\sigma_s^2+1)p^2 \leq q^2+2q.$$

□

Remark 2. The Equation (21) looks like the same as the Equation (1). The main difference between them is that the Equation (21) gets involved with the number of samples m. This suggests that in order to apply the modulus switching technique for the BDD strategy on LWE, one should find the new q' rather than using the one in (1).

Remark 3. Recall that, if (22) holds then so does the (20), hence (19) also holds. Thus the gap

$$\texttt{GAP} := (q^2+2q) - (n\sigma_s^2+1)p^2 \tag{24}$$

can be used to estimate the gap between two approaches. Obviously, the bigger the gap GAP is, the better the q'-approach is in comparison with the q-approach. The Equation (24) yields that: (i) the q'-approach is more suitable for short secret LWR instances which have σ_s small; (ii) in the case that p close to q, i.e., $\alpha \approx 1$, the q'-approach is not more efficient than the q-approach and (iii) by contrast, in the case p is much smaller than q, i.e., $\alpha \ll 1$, the q'-approach is much more efficient than the q-approach.

Remark 4. By Remark 1, given α, if n increases then $\log(q_{\min})$ also increases. This is likely to make the gap GAP increase. We can see this trend from the columns entitled "GAP (24)" in the Tables 2-5.

6 Implementation and Experimental Results

We implemented the BDD strategy on LWR problem to evaluate the efficacy of the q'-approach in comparison with the q-approach. In our experiments, we used SageMath version 8.1 [24] to implement the BDD strategy. The LBR algorithm used in our experiments is LLL [19]. We used the the function".LLL()" to call the floating point implementation of LLL in the *fplll* library which is included in SageMath with the default reduction parameter 0.99. By using such an LLL algorithm, we have the corresponding constant mentioned in (3) is $c_{\mathrm{LLL}} = 0.9775$ (see Section 2.2).

We introduce some parameters and notations presented in our experimental results: α is the bit ratio of p and q, i.e., $\alpha = \log(p)/\log(q)$; n is the dimension of secret; $\log(q_{\min})$ is the smallest bit size of q computed by (12) given α, n; $\log(q')$, $\log(q)$, $\log(p)$ are the bit size of modulo q' computed by (21), of moduli q and p, respectively; m is the optimal number of LWR samples for BDD attack

[3] It is easy to check that the funtion $x\ln(x)$ is concave over $(0, +\infty)$.

Table 1: **How to proceed with our experiments?**

1. First, choose α, then choose n and compute $\log(q_{\min})$ by (12).
2. Next, choose $\log(q)$ to be around $\log(q_{\min})$, from which we have q and p. After that we compute q' by (21) and compute m by (13).
3. For each tuple $(\alpha, n, q_{\min}, q, p, q')$, sample 10 LWR instances or 5 LWR instances up to α and n. (For $\alpha = 11/15$, we sample 10 LWR instances. For the rest of α's, we sample 5 LWR instances.)
4. For each LWR instance, transform it to LWE modulo q and LWE modulo q'.
5. Finally, run BDD attack on these two LWE instances.

computed by (13) (we use the same number of samples m in both the q-approach and q'- approach); the columns entitled "`GAP`", "succ(q)", "succ(q')" , "rHF(q)" and "rHF(q')" represent the size of the gap `GAP` in (24), the success probability of the q-approach and of the q'-approach, the rHF of the q-approach and of the q'-approach, respectively. Note that, rHF(q) and rHF(q') are computed using the formula (3).

The experimental results are summarized in the Tables 2 - 5 each of which corresponds to one value α. The small secret $\mathbf{s}$ is drawn uniformly at random over $\{-1, 0, 1\}^n$. The first $\alpha = 11/15$ is inspired from choosing parameters in the work of [4], the last one $\alpha = 1/3$ comes from [13] whilst two middle ones $\alpha = 2/3$ and $\alpha = 1/2$ are additionally suggested by us. We refer to Table 1 for generating parameters, sampling LWR instances, as well as running BDD attack on the corresponding LWE instances.

We highlight some noticeable things from our experimental results:

- In all cases, the rHF of the q-approach is always bigger than that of the q'-approach. Interestingly, the rHF of the q'-approach becomes smaller once α declines while the rHF of the q-approach does not seem to change, namely, rHF(q') ≈ 1.0201 for all considered α's. Recall that, smaller root Hermite factor means that the LWE modluo q' instance is more easily solved by BDD strategy than the LWE modluo q instance (see Section 2.4).
- When α is close to 1, such as $\alpha = 11/15$, the q'-approach does not outweigh the q-approach much (see Table 2). In constrast, when α is closer to 0 than 1, e.g., $\alpha = 1/3$, q' approach is much efficient than the q-approach in terms of success probability, rHF and even running time (although we do not add the runtime data in the tables due to lacking of space) since q' is quite close to p (see Table 5). For example, with $\alpha = 1/3, n = 100$, we $\log(q') = 59$ that is much less than $\log(q) = 119$.
- The bit size $\log(q')$ is quite close to $\log(p)$. Namely, in all considered cases, we have $\log(q') - \log(p)$ equals to 3 or 4. It seems that the difference $\log(q') - \log(p)$ increases (but slowly) once either n increases or/and α decreases.
- With fixed α, the successful range for BDD strategy seems to be widen when n grows. For instance, consider $\alpha = 11/15$: in case $n = 60$ we have $\log(q_{\min}) = 15$, if choose $\log(q) = 15$ then succ(q)=0%; by contrast, with

Table 2: **Compare q'-approach with the q-approach** ($\alpha = 11/15$)

$(n, \log(q_{\min}))$	$\log(q)$	$\log(p)$	$\log(q')$	m	GAP	succ(q)	succ(q')	rHF(q)	rHF(q')
(60,15)	**15**	11	**14**	166	9.5776e8	**0%**	**60%**	1.0202	1.0200
	17	12	**15**	176	1.6716e10	100%	100%	1.0203	1.0197
(80,20)	**17**	12	**15**	204	1.6567e10	0%	0%	1.0209	1.0200
	18	13	**16**	209	6.6267e10	**20%**	**80%**	1.0205	1.0200
	20	15	**19**	221	1.0603e12	**80%**	**100%**	1.0209	1.0200
	22	16	**20**	232	1.7435e13	100%	100%	1.0205	1.0200
(100,25)	**20**	15	**19**	247	1.0507e12	**20%**	**60%**	1.0210	1.0203
	25	18	**22**	276	1.1228e15	100%	100%	1.0212	1.0195

Table 3: **Compare the q'-approach with the q-approach** ($\alpha = 2/3$)

$(n, \log(q_{\min}))$	$\log(q)$	$\log(p)$	$\log(q')$	m	GAP	succ(q)	succ(q')	rHF(q)	rHF(q')
(60,18)	**16**	11	**14**	171	4.1791e9	80%	80%	1.0203	1.0199
	17	11	**14**	176	1.7064e10	**20%**	**60%**	1.0201	1.0188
	18	12	**15**	181	6.8256e10	**100%**	**100%**	1.0202	1.0191
(80,24)	**21**	14	**18**	226	4.3882e12	**20%**	**100%**	1.0203	1.0187
	23	15	**19**	237	7.0330e13	**40%**	**100%**	1.0215	1.0180
	24	16	**20**	242	2.8132e14	**100%**	**100%**	1.0208	1.0183
(100,30)	**24**	16	**20**	270	2.8128e14	**0%**	**80%**	1.0211	1.0184
	26	17	**21**	281	4.5028e15	**0%**	**100%**	1.0211	1.0179
	28	19	**23**	292	7.2045e16	100%	100%	1.0213	1.0182

Table 4: **Compare the q'-approach with the q-approach** ($\alpha = 1/2$)

$(n, \log(q_{\min}))$	$\log(q)$	$\log(p)$	$\log(q')$	m	GAP	succ(q)	succ(q')	rHF(q)	rHF(q')
(60,32)	**26**	13	**17**	218	4.5036e15	**0%**	**100%**	1.0210	1.0142
	30	15	**19**	234	1.1529e18	**20%**	**100%**	1.0201	1.0140
(80,43)	**27**	17	**21**	256	1.8014e16	**0%**	**80%**	1.0208	1.0146
	35	18	**22**	292	1.1806e21	**0%**	**100%**	1.0209	1.0140
(100,53)	**31**	16	**20**	307	4.5036e15	**0%**	**60%**	1.0214	1.0143
	37	19	**23**	336	1.8889e22	**0%**	**100%**	1.0213	1.0138

Table 5: **Compare the q'-approach with the q-approach** ($\alpha = 1/3$)

$(n, \log(q_{\min}))$	$\log(q)$	$\log(p)$	$\log(q')$	m	GAP	succ(q)	succ(q')	rHF(q)	rHF(q')
(40,48)	**38**	13	**17**	215	7.5558e22	**0%**	**100%**	1.0212	1.0096
(60,71)	**49**	16	**20**	299	3.1691e29	**0%**	**100%**	1.0215	1.0091
(80,95)	**61**	20	**24**	386	5.3169e36	**0%**	**100%**	1.0214	1.0088
(100,119)	**59**	20	**24**	424	3.3231e35	**0%**	**100%**	1.0213	1.0091

$n = 80$, we have $\log(q_{\min}) = 20$, hence if $\log(q) = 20$ then succ(q)=80%. Similarly, for $n = 100$, $\log(q_{\min}) = 25$ then succ(q)=100% with if $\log(q) = 25$.
- With fixed α, it seems that when n grows the difference succ(q')−succ(q) between two approaches also increases. Take $\alpha = 2/3$ for example, we can compare the difference succ(q')−succ(q) for $n = 60$ with the difference succ(q')−succ(q) for respect to $n = 100$ (see Table 3). This suggests that for large n, q'-approach actually outweighs q-approach.
- Also, fixed α and n, the difference succ(q')−succ(q) seems to depend on choosing q around $q_{\min}$. If q is much less than $q_{\min}$ then both succ(q) and succ(q') are 0%. For example, for $\alpha = 11/15, n = 80$, we have $\log(q_{\min}) = 20$, if we choose $\log(q) = 17$ then $\log(q') = 15$ and succ(q)=succ(q')=0%. However, for the case q is sufficiently bigger than $q_{\min}$, the success probability of two approaches is 1. Certainly, in that case, depending on which one is smaller (typically, q' is smaller than q), we choose the corresponding approach to reduce the running time of BDD attack.
- Remarkably, when α is close to 0 (e.g., $\alpha = 1/3$), the q'-approach significantly widens the successful range. For instance, with $\alpha = 1/3$, $n = 80$ and $\log(q_{\min}) = 95$, if we choose $\log(q) = 61$, then $\log(q') = 24$ and succ(q)=0% but succ(q')=100% (see Table 5).

The experimental phenomena listed above can be theoretically explained by our theoretical results. Our experimental results, in turn, also strongly support our reasonings. From our experiments, we can conclude that for LWR-based cryptosystems, one should choose two moduli q and p that are not so far from each other.

7 Conclusion

In this paper, we concentrated on applying the idea of the modulus switching technique for solving LWR instances. To do that, we scrutinized the behavior of LWR errors in order to evaluate their variances more precisely. Furthermore, the successful range in which a search LWR instance can be solved by the BDD strategy associated with Babai's Nearest Plane algorithm was also determined. Based on the successful range, we take LWR instances that are consistent with our experiments. Experimental results support our theoretical result that applying modulus switching technique for the BDD strategy on (small secret) search-LWR$_{m,n,q,p}(\chi_s)$ will be very efficient, especially if the bit ratio $\log(p)/\log(q)$ is close to 0 and/or n is sufficiently large. While our experiments were just proceeded with toy examples in which n is quite small, we believe that our modulus switching approach still performs very well for large n's used in practice.

Although our work does not give any warning to current LWR-based cryptosystems, it suggests that the modulus switching technique should be carefully considered in security analyses of prospective LWR-based (maybe even LWE-based) cryptosystems due to its remarkable effects. Considering effects of the modulus switching on other attacking strategies against LWR problem (even LWE) and analyzing its runtime effectiveness are our future works.

Acknowledgments. This work was supported by JST CREST Grant Number JPMJCR14D6, Japan. This work was also supported by JSPS KAKENHI Grant Number 16H02830. We would like to thank the anonymous reviewers for their careful reading as well as very helpful comments and suggestions.

References

1. Albrecht, M.R., Faugère, J.C., Fitzpatrick, R., Perret, L.: Lazy modulus switching for the bkw algorithm on lwe. In: Krawczyk, H. (ed.) Public-Key Cryptography – PKC 2014. pp. 429–445. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
2. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Cryptology ePrint Archive, Report 2015/046 (2015), https://eprint.iacr.org/2015/046
3. Arora, S., Ge, R.: New algorithms for learning in presence of errors. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) Automata, Languages and Programming. pp. 403–415. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
4. Baan, H., Bhattacharya, S., Garcia-Morchon, O., Rietman, R., Tolhuizen, L., Torre-Arce, J.L., Zhang, Z.: Round2: Kem and pke based on glwr. Submission to NIST proposal, Round 1 (2017), https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions
5. Babai, L.: On lovász' lattice reduction and the nearest lattice point problem. Combinatorica **6**(1), 1–13 (March 1986). https://doi.org/10.1007/BF02579403, https://doi.org/10.1007/BF02579403
6. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology – EUROCRYPT 2012. pp. 719–737. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
7. Bischof, C., Buchmann, J., Dagdelen, Ö., Fitzpatrick, R., Göpfert, F., Mariano, A.: Nearest planes in practice. In: Ors, B., Preneel, B. (eds.) Cryptography and Information Security in the Balkans. pp. 203–215. Springer International Publishing, Cham (2015)
8. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. pp. 309–325. ITCS '12, ACM, New York, NY, USA (2012). https://doi.org/10.1145/2090236.2090262, http://doi.acm.org/10.1145/2090236.2090262
9. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing. pp. 575–584. STOC '13, ACM, New York, NY, USA (2013). https://doi.org/10.1145/2488608.2488680, http://doi.acm.org/10.1145/2488608.2488680
10. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) lwe. In: Proceedings of the 2011 IEEE 52Nd Annual Symposium on Foundations of Computer Science. pp. 97–106. FOCS '11, IEEE Computer Society, Washington, DC, USA (2011). https://doi.org/10.1109/FOCS.2011.12, http://dx.doi.org/10.1109/FOCS.2011.12
11. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-lwe and security for key dependent messages. In: Rogaway, P. (ed.) Advances in Cryptology – CRYPTO 2011. pp. 505–524. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

12. Cheon, J.H., Kim, D., Lee, J., Song, Y.: Lizard public key encryption. Submission to NIST proposal, Round 1 (2017), https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions
13. Duc, A., Tramr, F., Vaudenay, S.: Better algorithms for lwe and lwr. Cryptology ePrint Archive, Report 2015/056 (2015), https://eprint.iacr.org/2015/056
14. Fang, F., Li, B., Lu, X., Liu, Y., Jia, D., Xue, H.: (deterministic) hierarchical identity-based encryption from learning with rounding over small modulus. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security. pp. 907–912. ASIA CCS '16, ACM, New York, NY, USA (2016). https://doi.org/10.1145/2897845.2897922, http://doi.acm.org/10.1145/2897845.2897922
15. Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: Smart, N. (ed.) Advances in Cryptology – EUROCRYPT 2008. pp. 31–51. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
16. Göpfert, F., van Vredendaal, C., Wunderer, T.: A hybrid lattice basis reduction and quantum search attack on lwe. In: Lange, T., Takagi, T. (eds.) Post-Quantum Cryptography. pp. 184–202. Springer International Publishing, Cham (2017)
17. Kudo, M., Yamaguchi, J., Guo, Y., Yasuda, M.: Practical analysis of key recovery attack against search-lwe problem. In: Ogawa, K., Yoshioka, K. (eds.) Advances in Information and Computer Security. pp. 164–181. Springer International Publishing, Cham (2016)
18. Laine, K., Lauter, K.: Key recovery for lwe in polynomial time. Cryptology ePrint Archive, Report 2015/176 (2015), https://eprint.iacr.org/2015/176
19. Lenstra, A.K., Lenstra, H.W., Lovasz, L.: Factoring polynomials with rational coefficients. In: Mathematische Annalen. vol. 261 (12 1982)
20. Lindner, R., Peikert, C.: Better key sizes (and attacks) for lwe-based encryption. In: Kiayias, A. (ed.) Topics in Cryptology – CT-RSA 2011. pp. 319–339. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
21. Micciancio, D., Regev, O.: Lattice-based cryptography. In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Post-Quantum Cryptography. pp. 147–191. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
22. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. J. ACM **56**(6), 34:1–34:40 (Sep 2009). https://doi.org/10.1145/1568318.1568324, http://doi.acm.org/10.1145/1568318.1568324
23. Schnorr, C.P., Euchner, M.: Lattice basis reduction: Improved practical algorithms and solving subset sum problems. Mathematical Programming **66**(1), 181–199 (Aug 1994). https://doi.org/10.1007/BF01581144, https://doi.org/10.1007/BF01581144
24. Stein, W., et al.: Sage Mathematics Software (Version 8.1). The Sage Development Team (2018), `http://www.sagemath.org`