A new attack against search-LWE using Diophantine approximations

Robin Frot[1] and Daniel Zentai[2]

[1] robin.frot@xtendr.io
[2] daniel.zentai@xtendr.io
xtendr
Budapest, Hungary

Abstract. In this paper, we present a new attack against search-LWE instances with a small secret key. The method consists of lifting the public key to $\mathbb{Z}$ and finding a good Diophantine approximation of the public key divided by the modulus a. This is done using lattice reduction algorithms. The lattice considered, and the approximation quality needed is similar to known decision-LWE attacks for small keys. However, we do not require an in-depth analysis of the reduction algorithm (any reduction algorithm giving small enough vectors is enough for us), and our method solves the search problem directly, which is harder than the decision problem.

1 Introduction

Introduction on attacks against LWE with small key size Encryption schemes based on lattice problems have become more popular over the years due to their fully homomorphic properties. However, the strength of the underlying problems: learning with errors (LWE) or ring learning with errors (RLWE) may vary a lot depending on the choice of parameters. In this paper, we will focus on LWE instances with small key size. In [AD21], the authors give an overview of a method for solving LWE decision instances. This method stems from a careful analysis of lattice reduction algorithms (such as BKZ or Slide), showing that they behave differently depending on the presence of a small vector in the lattice. Classical techniques can then be used to perform a search-to-decision reduction. This proceeds as follows (in the case of BKZ). The BKZ reduction gives a small vector v such that

$$\|v\|_2 \ll \max(\gamma \det(\Lambda)^{\frac{1}{\dim(\Lambda)}}, \gamma^2 \lambda_1(\Lambda))$$

where γ is a constant depending on the reduction parameters, Λ a well-chosen lattice, and $\lambda_1(\Lambda)$ the norm of the smallest vector of Λ (that in this case is proportional with the size of the secret key and the error if it is an LWE instance). In particular, knowing the determinant and the norm of the small vector, it suffices to ensure a reduction such that

$$\lambda_1(\Lambda) \ll \frac{\det(\Lambda)^{\frac{1}{\dim(\Lambda)}}}{\gamma}$$

to decide in which regime we are.

Novelty of our method In this paper, we propose an attack against short key LWE instances that solves the search-LWE problem. We start with a lattice Λ, similar to the one in the above decision case, but it does not have small vectors. After reducing the lattice, we use the first few vectors of the reduced lattice to create linearly independent relations between the public key, the secret key, and the error. This can be interpreted as a simultaneous Diophantine approximation of the coefficients of the public key. Using linear algebra, we are able to recover the secret directly. In particular, this method does not require a careful analysis of the reduction algorithm, and skips the search-to-decision reduction.

Structure of the paper After recalling the basic properties of some lattice reduction algorithms and giving an overview of Diophantine approximations (Section 2), we first give an intuitive version of the attack against RLWE (Section 3) and then some refinements (Section 4). Then in Section 5, we give a generalisation of the method in a non commutatiove case, solving general LWE problems. Finally, we give some experimental results using the [RH23] lattice reduction algorithm (Section 6), and we conclude by comparing the decision attack cited above and our method.

2 Notations and preliminaries

2.1 Notations

Throughout this article, we will use the following notations.

Notations for FHE: Let $N = 2^k$ be the ring dimension for some positive integer k, q a large modulus, and t a small modulus. K will denote the cyclotomic field

$$K := \mathbb{Q}[X]/(X^N + 1),$$

$\mathcal{R}$ will denote the integer ring of K

$$\mathcal{R} := \mathbb{Z}[X]/(X^N + 1)$$

and $\mathcal{R}_q$ will denote its reduction modulo q:

$$\mathcal{R}_q := \mathcal{R}/q\mathcal{R}.$$

The distributions $\mathcal{U}_q$, χ_s, and χ_e will be respectively the uniform distribution on $\mathcal{R}_q$, the secret distribution on K, and the error distribution on K. In this paper, we will consider χ_e to be a discrete Gaussian distribution with small variance and χ_s to be a ternary distribution with prescribed Hamming weight or χ_e.

For any polynomial

$$a = \sum_{i=0}^{N-1} a_i X^i \in K$$

in reduced form, we define its norm by

$$\|a\| := \sup_i |a_i|.$$

Finally, for a polynomial $a \in K$, we define the rounded polynomial

$$\lfloor a \rceil := \sum_{i=0}^{N-1} \lfloor a_i \rceil X^i$$

where $\lfloor \cdot \rceil$ is the rounding operation over $\mathbb{Q}$.

Notations for lattice reduction: All the lattices Λ in this paper will have integer coefficients. A basis of the lattice will be represented by matrices where the column vectors are basis vectors. Until Section 6, the "approximation factors γ will be considered compared to the sup-norm of a small vector:

$$\|v\| < \gamma \det(\Lambda)^{\frac{1}{\dim(\Lambda)}}.$$

2.2 Background on RLWE and its implementations

Many encryption schemes are based on the following problem.

Definition 1 (Ring Learning With Errors (RLWE) search). *Let $s \leftarrow \chi_s$ be a secret. The search RLWE problem is to find s given a pair $(a, as+te) \in \mathcal{R}_q^2$ where $e \leftarrow \chi_e$ and $a \leftarrow \mathcal{U}_q$.*

In this article, we will consider three encryption schemes that use this type of key generation, namely BGV ([BGV14]), BFV ([Bra12], [FV12]) and CKKS ([Che+17]). They each come with specific parameter choices: in BFV and CKKS, the small modulus t is equal to 1, which is not the case for BGV. Moreover, in the case of CKKS, the large modulus is usually a very large product of primes, while in the other schemes q is usually a reasonably sized prime number. As we will explain in Section 4, it will always be possible to remove the contribution of the small modulus t in the attack (in the case of BGV), it will also be possible to reduce the large modulus q (it works in all the cases but it is particularly useful when dealing with CKKS).

Overall, all the aforementioned Fully Homomorphic Encryption (FHE) schemes will be treated the same way since the Diophantine approximation attack only deals with the public key generation and is unrelated to the encryption, evaluation, and decryption algorithms.

2.3 Lattice reduction algorithms

Let us start by defining the notion of a lattice.

Definition 2. *A k-dimensional lattice Λ is a discrete subgroup of $\mathcal{R}^n$ such that $Span_{\mathbb{R}}(\Lambda)$ is a k-dimensional vector space. We define a basis $B = (b_1, ..., b_k) \in \Lambda^k$ of Λ as a minimal generating family: $Span_{\mathbb{Z}}(B) = \Lambda$ and there exists no family of $k-1$ vectors that spans Λ. Finally, we define the minimum distance of Λ by the length of the shortest vector:*

$$\lambda_1(\Lambda) := \min_{\substack{v \in \Lambda \\ v \neq 0}} \|v\|_2$$

where the norm considered is the Euclidian norm.

In this paper, we will assume that all lattices have full rank (i.e. $k = n$). If it is not the case, the lattice can always be written with coordinates in $Span_{\mathbb{R}}(\Lambda)$ to reduce the problems to full rank lattices in dimesion k.
Given a basis of a lattice, the problem of finding a short vector is believed to be hard if the basis vectors are very skewed. The following shortest vector problem is believed to be hard for small values of $\gamma(n)$.

Definition 3 (Search γ-SVP). *Given $\gamma = \gamma(n) > 1$ and a basis B of a lattice Λ, find a non zero vector $v \in \Lambda$ such that*

$$\|v\|_2 < \gamma \lambda_1(\Lambda).$$

One can refer to [Ben23] for the complexities of SVP-related problems.

The goal of a lattice reduction algorithm is to output a reduced basis composed of smaller vectors.

One of the most used lattice reduction algorithms is the BKZ algorithm:

Theorem 1 (BKZ). *Let $B = (b_1, ..., b_n)$ be a basis of a lattice Λ, b_i^* be the i-th Gram-Schmidt vector and $\lambda_i(\Lambda)$ the i-th smallest vector in Λ. We say that a B is (β, ϵ)-BKZ reduced if for all i*

$$\|b_i^*\| \leq (1+\epsilon)\lambda_1(\Lambda(B_{[i,\max(i+\beta,n)]}))$$

where $\Lambda(B_{[i,\max(i+\beta,n)]})$ is the sub-lattice generated by the vectors is the basis with indices in the interval. There exists an algorithm that produces such a reduced basis. Moreover, one has

(i) $\|b_1\|_2 \leq \sqrt{(1+\epsilon)\gamma_\beta}^{\frac{n-1}{\beta-1}+1} det(B)^{\frac{1}{n}}$
(ii) $\|b_1\|_2 \leq ((1+\epsilon)\gamma_\beta)^{\frac{n-1}{\beta-1}} \lambda_1(\Lambda)$

(see [SE94] for the BKZ algorithm, and [LN20] for the bounds).

In practice, we used the algorithm given in [RH23] for testing:

Theorem 2 ([RH23] Theorem 2). *Let $B = (b_1, ..., b_n)$ be a basis of a lattice Λ, b_i^* be the i-th Gram-Schmidt vector and $\lambda_i(\Lambda)$ the i-th smallest vector in Λ. We say that a B is α-reduced if the following condition are satisfied:*

(i) $\|b_1\|_2 \leq 2^{\alpha n} det(B)^{\frac{1}{n}}$,

(ii) $\|b_n^*\|_2 \geq 2^{-\alpha n} det(B)^{\frac{1}{n}}$,
(iii) $\|b_i\|_2 \leq 2^{\alpha n + O(n)} \lambda_i(\Lambda)$,
(iv) $\prod_i \|b_i\|_2 \leq 2^{\alpha n^2 + O(n^2)} det(B)$.

There exists an algorithm of complexity $O(n^\omega(n+L))$ *that produces an* α*-reduced basis where* $\omega \in]2,3]$ *and* L *is the bit size of the entries.*

2.4 Simultaneous Diophantine approximations

The classical Diophantine approximation problem is to find, given a real number x, a rational number p/q such that

$$\left| x - \frac{p}{q} \right| < \frac{\phi(q)}{q}$$

for some approximation function ϕ. It has been known for a long time, that, using continued fraction approximation, it is possible to choose $\phi = q^{-1}$. For more background on simultaneous Diophantine approximation, the reader may refer to [Che13]. We are interested here in a variant of this problem.

Definition 4 ((r,ϕ)-approximation). *Let* $(a_{i,j}) \in \mathbb{R}^{n \times m}$ *be real numbers,* $r \in \mathbb{N}$ *and* ϕ *an evaluation function. An* (r,ϕ)*-approximation of* $(a_{i,j})$ *is a pair of tuples* $(q_1, .., q_m) \in \mathbb{Z}^m$ *and* $(\alpha_1, ..., \alpha_n)$ *such that*

$$\max_{1 \leq i \leq n} \left| \frac{1}{r} \sum_{j=1}^{m} q_j a_{i,j} - \alpha_i \right| < \phi(r).$$

Such approximation can be found using lattice reduction algorithms (one can refer to [BS13] and [FV23]). We will slightly differ from the aforementioned articles since most implementations of lattice reduction work best with integer coefficient lattices. In practice, the coefficients $a_{i,j}$ will be integers. Let $A \in M_{n,m}(\mathbb{R})$ the matrix consisting of the $a_{i,j}$ coefficients and define the lattice given by

$$B := \begin{pmatrix} rI_n & A \\ 0 & I_m \end{pmatrix} \in GL_{n+m}(\mathbb{R}) \tag{1}$$

for some fixed r. Any other basis $B' = (b'_1, ..., b'_n)$ of the lattice will have vectors of the shape

$$b'_i = \begin{pmatrix} \sum_{j=1}^{m} q_{j,i} a_{1,j} - r\alpha_{1,i} \\ \vdots \\ \sum_{j=1}^{m} q_{j,i} a_{n,j} - r\alpha_{n,i} \\ q_{1,i} \\ \vdots \\ q_{m,i} \end{pmatrix}.$$

Due to a direct application of Minkowski's theorem, we know the following bound

Proposition 1. *There exists a vector v as defined above such that*

$$\|v\| \leq r^{\frac{n}{m+n}}.$$

In particular, there exists tuples $(q_1, ..., q_m)$ and $(\alpha_1, ..., \alpha_m)$ such that for all i

$$\left| \frac{1}{r} \sum_{j=1}^{m} q_j a_{i,j} - \alpha_i \right| \leq r^{-\frac{m}{m+n}}$$

and

$$|q_i| \leq r^{-\frac{m}{m+n}}.$$

If the basis B' is α-reduced according to Definition 2 (i.e. the output of Ryan and Heninger reduction algorithm in [RH23]), we have

$$\frac{1}{r}\|b_1\|_2 \leq \frac{1}{r} 2^{\alpha(m+n)} \det(B)^{\frac{1}{m+n}} = 2^{\alpha(m+n)} r^{\frac{-m}{m+n}}.$$

In particular, we have

$$\max_{1 \leq i \leq n} \left| \frac{1}{r} \sum_{j=1}^{m} q_{j,1} a_{i,j} - \alpha_{i,1} \right| < C_{\alpha,m,n} r^{\frac{-m}{m+n}}.$$

for some constant only depending on α, m and n. In practice (for random $a_{i,j}$) more than one vector of the reduced basis will produce a satisfactory Diophantine approximation.

3 The basic attack

3.1 Overview of the method

Let us start with a secret key $s \leftarrow \chi_s$ as well as a public key $(a, b) \in \mathcal{R}_q^2$ where

$$\begin{aligned} a &\leftarrow \mathcal{U}_q, \\ e &\leftarrow \chi_e, \\ b &:= as + te \mod q. \end{aligned}$$

By abuse of notation, we also denote by a the lift of $a \in \mathcal{R}_q$ to K with coefficients in $[-(q-1)/2, (q-1)/2]$. The key observation of the method is that if we assume that polynomial multiplication are done in K instead of $\mathcal{R}_q$, we are able to write

$$b = as + te + qY$$

where

$$Y := -\left\lfloor \frac{as + te}{q} \right\rceil.$$

The goal is now to find two other relations

$$b_1 = \alpha s + \alpha_t e + \alpha_q Y$$
$$b_2 = \beta s + \beta_t e + \beta_q Y$$

such that $(b_1, b_2, \beta, \gamma, \beta_t, \gamma_t, \beta_q, \gamma_q) \in K^8$ are known and the matrix

$$\begin{pmatrix} a & t & q \\ \alpha & \alpha_t & \alpha_q \\ \beta & \beta_t & \beta_q \end{pmatrix} \in M_3(K)$$

is invertible.

Obviously, multiplying b by another polynomial c does not produce a linearly independent equation. This is why we will use the rounding operation in order to introduce some non-linearity. Assume that we found a polynomial $c \in K$ such that

$$ac = \alpha + \epsilon,$$
$$tc = \alpha_t + \epsilon_t,$$
$$qc = \alpha_q + \epsilon_q,$$

$(\alpha, \alpha_t, \alpha_q) \in \mathcal{R}^3$ and

$$\|\epsilon s + \epsilon_t e + \epsilon_q Y\| < \frac{1}{2}.$$

After rounding, we get

$$\begin{aligned} b_1 := \lfloor cb \rceil &= \lfloor (\alpha + \epsilon)s + (\alpha_t + \epsilon_t)e + (\alpha_q + \epsilon_q)Y \rceil \\ &= \alpha s + \alpha_t e + \alpha_q Y + \lfloor \epsilon s + \epsilon_t e + \epsilon_q Y \rceil \\ &= \alpha s + \alpha_t e + \alpha_q Y \end{aligned}$$

due to the conditions above. Since we assumed that the secret key s has small coefficients with Hamming weight $h(s)$, and the coefficients of a can be chosen in $[-(q-1)/2, (q-1)/2]$ we have

$$\|Y\| < \left\| \frac{as + te}{q} \right\| + 1 < \frac{1}{2} h(s) \|s\| + 1$$

in the case where s is ternary with low Hamming weight and

$$\|Y\| \ll N\sigma$$

if s follows a Gaussian distribution of standard deviation σ. Since the a and s are random sample, it should be noted that the expected size of Y is smaller due to cancellations in the coefficients of the product as:

$$\mathbb{E}[\|Y\|] \ll N^{\frac{1}{2}}\sigma$$

Thus, the conditions on the terms ϵ, ϵ_t, and ϵ_q are not too hard to meet with good enough Diophantine approximations.

Since Y is the largest polynomial and the errors ϵ, ϵ_t, and ϵ_q are expected to be of the same size, we will formulate the condition on the error factor the following way:

$$\|\epsilon_q Y\| < 1/2 - \eta.$$

Since we have

$$\|\epsilon_q Y\| < N\|\epsilon_q\|\|Y\|$$

and

$$\mathbb{E}[\|\epsilon_q Y\|] \ll N^{\frac{1}{2}}\|\epsilon_q\|\|Y\|,$$

we should aim for $\|\epsilon_q\| \ll (Nh(s))^{-1}$ for ternary s (resp. $\|\epsilon_q\| \ll (N^2\sigma)^{-1}$ for Gaussian s). We should still get some results on average if $\|\epsilon_q\| \ll (\sqrt{N}h(s))^{-1}$ (resp. $\|\epsilon_q\| \ll (N\sigma)^{-1}$).

3.2 Setting up the approximation

Reduction to the case $t = 1$ In the case where the error is of the shape te for some small modulus t and Gaussian distribution sample e, we set $\tilde{t}$ the inverse of t modulo q and write

$$\tilde{t}b = \tilde{t}as + (1 + kq)e + \tilde{t}qY$$

where k is such that $t\tilde{t} = 1 + kq$. By setting $\tilde{t}a := \tilde{a} + qa'$ and reducing modulo q, we get

$$\tilde{b} := \tilde{t}b \mod q = \tilde{a}s + e + q\tilde{Y}$$

for some $\tilde{Y}$. Note that this operation is just multiplying by $\tilde{t}$ in the ring $\mathcal{R}_q$. However, we chose to explain the operations over K for consistency.

Construction of the lattice Due to the previous section, we may assume that $t = 1$. As we saw in Section 2.4, it is not advisable to directly approximate the coefficient of a and q since it would give a matrix A of dimension $1 \times (N + 1)$ (see section 2.4) and would return a very poor approximation. Let us define

$$A = \begin{pmatrix} A_0 \\ qI_n \end{pmatrix}$$

where $(A_0)_{i,j} = (-1)^{\left\lfloor \frac{i+j}{N} \right\rfloor} a_{i-j[N]}$. The matrix A is in $M_{2N,N}(\mathbb{Z})$ and reducing the matrix B given by (1) produce for each of its columns some coefficients q_k,

α_k, $\alpha_{t,k}$, $\alpha_{q,k}$ and an approximation such that, for all k,

$$\left|\frac{1}{r}\sum_{i+j=k \mod N}(-1)^{\lfloor\frac{i+j}{N}\rfloor}q_i a_j - \alpha_k\right| \ll r^{-\frac{1}{3}}$$
$$\left|\frac{q_i}{r}q - \alpha_{q,k}\right| \ll r^{-\frac{1}{3}}$$
$$\left|\frac{q_i}{r}\right| \ll r^{-\frac{1}{3}}$$

The polynomial

$$Q := \sum_i \frac{q_i}{r}X^i$$

is thus likely to satisfy the conditions of the previous section. Due to the size of Q, we will always have $\alpha_1 = 0$ and it is enough to invert the matrix

$$\begin{pmatrix}\alpha & \alpha_q \\ \beta & \beta_q\end{pmatrix} \in M_3(K)$$

It should be noted that if the coefficients of Q are too close to integers, the reduction will not produce an independent linear relation. Indeed, if we have

$$Q = Q' + \epsilon'$$

where Q has integer coefficient and $\|\epsilon'\|$ is small compared to q, we get

$$\lfloor Qb \rceil = Q'as + tQ'e + qQ'Y = Q'b.$$

This behavior can occur in practice if r is poorly chosen. Minkowski's theorem gives the existence of a small vector v in our lattice with $\|v\| < r^{\frac{2}{3}}$. In particular, if the columns of A (or a small linear combination of them) are already smaller than the expected short vector, they will likely be the shortest vectors of the lattice without adding any new information. Since they are of sup norm roughly q, one must choose

$$r \ll q^{\frac{3}{2}}$$

and can expect a short vector of size

$$\|v\| \ll q,$$

or, for the approximation quality,

$$\frac{1}{r}\|v\| \ll q^{-\frac{1}{2}}.$$

Note that it is still possible to choose r to be larger than $q^{3/2}$. In that case, the lattice will have small vectors that are to be ignored and we will still get short enough vectors in the base to produce independent linear relations.

The last quantity to be computed is the approximation factor (see the notations section for the definition) needed in the lattice reduction algorithms. We will only consider the average cases in last remarks of Section 3.1. We need to find a vector v' such that $\|v'\| \ll (N\sigma)^{-1}$ in the Gaussian case (resp. $\|v'\| \ll (N^{\frac{1}{2}}h(s))^{-1}$ in the ternary case. Given the existence of the vector v above, it is necessary to reduce the lattice with an approximation factor $\gamma \ll q^{\frac{1}{2}}(N\sigma)^{-1}$ (resp. $\gamma \ll q^{\frac{1}{2}}(N^{\frac{1}{2}}h(s))^{-1}$).

As a final remark, we can see that this method only works reliably if $q^{\frac{1}{2}} \gg N$ (resp. $q \gg N$). If it is not the case, the method would only work if there exists a small vector that is smaller than the expected value, which is unlikely given that the initial matrix is random.

4 Improvements

In this section, we discuss several possible improvements of the basic attack in Section 3, showing the versatility of the method.

4.1 Large modulus and error reduction

First, let us note that the quality of approximation directly depends on the size of the modulus so it is better to keep the modulus large. However, if the modulus is really large (e.g. in the case of CKKS encryption), it is possible to reduce it in order to reduce the bit size of the lattice entries. Then, one can solve the optimization problem of finding the balance with the complexity of lattice reduction algorithm with large approximation factor and large bit size or smaller approximation factor and smaller bitsize. The following modulus reduction is inspired by the rescaling procedure of the CKKS scheme [Che+17]. Let $d \in \mathbb{N}$ be a positive integer and write

$$a = a_0 + da'$$
$$q = q_0 + dq'$$

where the coefficients $(a_0)_i$ of a_0 and q_0 are minimal in absolute value. In this case, we have

$$\left\lfloor \frac{b}{d} \right\rceil = a's + q'Y + \left\lfloor \frac{te + a_0 s + q_0 Y}{d} \right\rceil$$

and the new error e' is bounded by

$$\left\| \left\lfloor \frac{te + a_0 s + q_0 Y}{d} \right\rceil \right\| \leq \frac{h}{2} + \frac{\|te\|}{d}$$

where h is the Hamming weight of s. Moreover, we have $q' \ll q/d$.

4.2 Lattice dimension reduction

As the complexity of the lattice reduction algorithms depends mainly on the dimension of the lattice, it is important to find ways to reduce the dimension of the lattice. In the previous section, the lattices considered were of dimension $4N$ where N is the ring dimension of the FHE instance. Moreover, the quality of the reduction also crucially depends on the lattice dimension. Finally, the improvement calculations for these methods do not take into account some lucky guesses that can occur: it is possible that the reduction algorithm produces vectors that are way smaller than the expected size of the output.

Directly removing columns Recall that in Section 3.2, we reduced the problem to reducing a matrix involving $A = (A_0^T | qI_N)^T$. It is possible to remove the leftmost columns of A to get a new matrix $\tilde{A} \in M_{2N,\delta N}$. This reduces the size of the matrix to be reduced from $3N$ to $(2+\delta)N$. Note that the new matrix also has determinant r^{2N}:

$$\tilde{B} := \begin{pmatrix} rI_{2N} & \tilde{A} \\ 0 & I_{\delta N} \end{pmatrix} \in GL_{(2+\delta)N}(\mathbb{R}).$$

We can compute the quality of the approximation as in the main case. One should choose

$$r \ll q^{\frac{2+\delta}{2}}$$

and the quality of approximation is

$$\frac{1}{r}\|v\| \ll q^{-\frac{\delta}{2}}.$$

Choosing $r = q$ It may seem counter intuitive to reduce the size of r since in the previous sections it was linked to the approximation quality. However, in the case where $r = q$, we show that it is enough to reduce the matrix

$$B' = \begin{pmatrix} qId & A_0 \\ 0 & Id \end{pmatrix} \in GL_{2N}(\mathbb{Z}).$$

A reduced vector of this matrix is of the shape

$$v = \begin{pmatrix} \sum_j q_j a_{-j[N]} - q\beta_0 \\ \sum_j q_j a_{1-j[N]} - q\beta_1 \\ \vdots \\ \sum_j q_j a_{N-1-j[N]} - q\beta_{N-1} \\ q_0 \\ \vdots \\ q_{N-1} \end{pmatrix}$$

with each coefficient of size $O(q^{\frac{1}{2}})$. The volume of the lattice is indeed q^N and the dimension is $2N$. In particular, if we set

$$Q = \frac{1}{q} \sum q_i X^i,$$

we have

$$\tilde{c}b = Qas + Qe + qQY$$

with qQY having integer coefficients and Qe having small enough coefficients. After rounding, this gives a relation

$$b_1 = \alpha s + qQY.$$

This method effectively divides the dimension by $3/2$ while keeping the approximation error $q^{\frac{1}{2}}$. In the main case, Y was the largest polynomial. Since here $\epsilon_q = 0$, the strongest condition satisfied by one of the error terms comes from ϵ_1. Moreover, it make little difference to assume that s is a Gaussian distribution since it plays the same role as e. We have

$$\|Qe\| < N\sigma\|Q\|$$

and

$$\mathbb{E}[\|Qe\|] < N^{\frac{1}{2}}\sigma\|Q\|.$$

In particular, the approximation factor of the lattice reduction algorithm should be chosen to be

$$\gamma \ll \frac{q^{\frac{1}{2}}}{\sigma N}$$

or

$$\gamma \ll \frac{1}{\sigma}\left(\frac{q}{N}\right)^{\frac{1}{2}}$$

for the average case.

One can note that in this case, the lattice chosen is very close to the lattice used in [AD21], meaning that the reduction time should be similar.

4.3 Improving the approximation quality using properties of FHE schemes

In this section, we will focus on the CKKS encryption as an example, but similar results can be obtained for other encryption schemes. Along with the encryption key, the CKKS scheme comes with a relinearisation key (a_0, b_0) where a_0 is a uniformly chosen polynomial modulo $q\tilde{q}$ where $\tilde{q} \asymp q$ and

$$b_0 = a_0 s + e_0 + \tilde{q}s^2 \mod q\tilde{q}.$$

For fast calculations, the schemes also come with rotation keys (a_k, b_k) where a_k is also a uniformly chosen polynomial modulo $q\tilde{q}$ and

$$b_k = a_k s + e_k + \tilde{q} s\left(X^{5^k}\right) \mod q\tilde{q}$$

for $1 \leq k \leq N-1$.

Since we do not care about s^2 or $s\left(X^{5^k}\right)$, we can reduce these expression modulo $\tilde{q}$. If we set $a_k = \tilde{a}_k + \tilde{q}a'_k$, we get for all $0 \leq k \leq N-1$

$$\tilde{b}_k := b_k \mod \tilde{q} = \tilde{a}_k s + e_k + \tilde{q}Y_k.$$

As in Section 3.2, we can construct the matrices $\tilde{A}_k \in GL_N(\mathbb{Q})$ by

$$(\tilde{A}_k)_{i,j} := (-1)^{\lfloor \frac{i+j}{N} \rfloor} (\tilde{a}_k)_{i-j[N]}.$$

If we now define the matrices

$$A_l = (\tilde{A}_0 | ... | \tilde{A}_{l-1}) \in M_{N,lN}(\mathbb{Q})$$

$$B_l = \begin{pmatrix} qI_N & A_l \\ 0 & I_{lN} \end{pmatrix} \in GL_{(l+1)N}(\mathbb{Q})$$

In this case, Minkowski's theorem gives a small vector v such that

$$\frac{1}{\tilde{q}}\|v\| < q^{-1+\frac{1}{l}}$$

and thus the approximation factor needed in the lattice reduction is $\gamma = O(q^{1-1/l}(\sigma N)^{-1})$ or $\gamma = O(q^{1-1/l}(\sigma\sqrt{N})^{-1})$.

Another possibility is to work directly with the relinearisation key. Define

$$(A'_0)_{i,j} := (-1)^{\lfloor \frac{i+j}{N} \rfloor} (\tilde{a}_0)_{i-j[N]}$$

and

$$B' = \begin{pmatrix} q\tilde{q}I_N & 0 & A'_0 \\ 0 & q\tilde{q}I_N & \tilde{q}I_N \\ 0 & 0 & I_N \end{pmatrix} \in GL_{3N}(\mathbb{Q}).$$

In this case the small vector is bounded by

$$\frac{1}{q\tilde{q}}\|v\|(q\tilde{q})^{-1/3} \asymp q^{-\frac{2}{3}}.$$

This give the approximation factors $\gamma = O(q^{2/3}(\sigma N)^{-1})$ or $\gamma = O(q^{2/3}(\sigma\sqrt{N})^{-1})$. Note that this last estimate is a strict improvement of the basic attack since the dimension is the same and the approximation factors are relaxed.

5 A more general version

We will explain in this section how to generalize this method to any (not necessarily commutative) subring of $M_n(\mathbb{Z})$ or $M_n(\mathbb{Z}/q\mathbb{Z})$. In particular, this method also works for the classical LWE problems (still with short key and error).

Let us begin by noticing the following fact showing that the results of this section will be a strict generalization.

Proposition 2. *There exists a ring morphism $\phi : \mathcal{R} \to M_N(\mathbb{Z})$ such that $\phi(a) = A$, where*

$$(A)_{i,j} := (-1)^{\left\lfloor \frac{i+j}{N} \right\rfloor} a_{i-j[N]}.$$

Proof. It is enough to define

$$\phi(X) = \tilde{J}$$

where

$$\tilde{J} := \begin{pmatrix} 0 & \cdots & \cdots & 0 & -1 \\ 1 & \ddots & & & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}$$

and complete ϕ by linearity.

Let $A = (a_{i,j})$ be a uniformly chosen matrix in $M_n(\mathbb{Z}/q\mathbb{Z})$ and s, e two vectors in $\mathbb{Z}^N$ sample from a Gaussian distribution with small standard deviation. Let us define

$$b := As + e \in (\mathbb{Z}/q\mathbb{Z})^N.$$

As before, if we see A as a matrix with integer coefficients, we have the lift

$$b := As + e + qY \in \mathbb{Z}^N$$

for some vector $Y \in \mathbb{Z}^N$.

If M we define

$$M = \begin{pmatrix} qI_n & A^T \\ 0 & I_n \end{pmatrix}.$$

Note that the transpose is just a trick used to allow us to use left multiplication and is completely independent of the reduction. If $\tilde{M}$ is a matrix for a reduced basis of the lattice defined by M, we have the following.

Proposition 3. *With the above notations, any subset of N column vectors of $\tilde{M}$ is of the shape*

$$\begin{pmatrix} A^T Q + qA' \\ Q \end{pmatrix}$$

Proof. Any column of $\tilde{M}$ is of the shape

$$\begin{pmatrix} \sum_j a_{j,1} q_j + \alpha_1 q \\ \vdots \\ \sum_j a_{j,N} q_j + \alpha_N q \\ q_1 \\ \vdots \\ q_N \end{pmatrix}.$$

If a $2N \times N$ matrix consisting of N such columns is

$$\begin{pmatrix} \tilde{A} \\ Q \end{pmatrix}$$

with $Q = (q_{i,j})$, we have

$$(\tilde{A})_{i,j} = \sum_k a_{k,i} q_{k,j} + q\alpha i, j.$$

In particular,

$$\tilde{A} = A^T Q + qA'.$$

Heuristically, all basis vectors in $\tilde{M}$ should be of size $O(q^{\frac{1}{2}})$ using Minokowski's Theorem since we started from a random q-ary matrix. In particular, this gives

$$\begin{aligned} q^{-1} A^T Q &= A' + O(q^{\frac{1}{2}}) \\ q^{-1} Q &= O(q^{\frac{1}{2}}). \end{aligned}$$

Finally, as in Section 4.2, we have

$$\begin{aligned} \left\lfloor \frac{1}{q} Q^T b \right\rceil &= \left\lfloor \frac{1}{q} (A^T Q)^T s + Q^T Y + \frac{1}{q} Q^T e \right\rceil \\ &= (A')^T s + Q^T Y \end{aligned}$$

By finding another relation of this type (i.e. selecting another subset of N vectors), we get the system

$$\begin{cases} b_1 = A_1 s + Q_1 Y \\ b_2 = A_2 s + Q_2 Y. \end{cases}$$

This can be solved for s if $Q_1, Q_2 \in GL_N(\mathbb{Q})$ and $Q_1^{-1} A_1 - Q_2^{-1} A_2 \in GL_N(\mathbb{Q})$:

$$s = (Q_1^{-1} A_1 - Q_2^{-1} A_2)^{-1} (Q_1^{-1} b_1 - Q_2^{-1} b_2).$$

One can note that if $M = kN$ LWE samples are available, it is possible to consider the matrix

$$A = \begin{pmatrix} A_1 \\ \vdots \\ A_k \end{pmatrix} \tag{2}$$

in the above in order to recover Q_1, ..., Q_k such that

$$\frac{1}{q}\sum A_i^T Q_i = \alpha + O(q^{-\delta}).$$

with some better precision.

6 Expected and experimental results

In this section, we assume that all reductions have been performed using the matrix in Section 4.2.

6.1 Approximation needed for a successful attack

In the above section our goal was to reduce a lattice so that the sup norm of the vectors is bounded by a suitable constant depending on q, n and σ. In order to get back to more classical results on lattice reduction, let us translate this into bounds in the Euclidean norm.

Assuming that all the coefficients of the reduced vectors are of similar size, we have

$$\sqrt{2N}\|v\| \asymp \|v\|_2.$$

As seen previously, our goal is to find basis vectors such that

$$\|v\|_2 \asymp \sqrt{2N}\|v\| < \frac{\sqrt{2}q}{\sigma}$$

(in the case of Section 4.2). Since the lattice reduction yields vectors v for the lattice Λ satisfying

$$\|v\|_2 < H \det(\Lambda)^{\frac{1}{\dim(\Lambda)}} = Hq^{\frac{1}{2}},$$

our goal is to have the Hermite factor $H < \frac{\sqrt{2}}{\sigma}q^{\frac{1}{2}}$. The difficulty is more often measured in terms of the root Hermite factor

$$rhf = H^{\frac{1}{\dim(\Lambda)}},$$

and we must have

$$rhf < \left(\frac{\sqrt{2}}{\sigma}\right)^{\frac{1}{2N}} q^{\frac{1}{4N}}$$

for a successful attack.

6.2 Experimental results

We conducted experiments on an Intel i7-1355U processor at 2.25 GHz using the `flatter` reduction library (using 12 threads). The first set of tests has been conducted with $\log_2(q) = 120$ and $\sigma = 5$ with a low precision of `flatter` (with `flatter -rhf 1.02`):

$\log_2(N)$	secret key recovered	target rhf	effective rhf	wall time
6	Yes	1.373	1.017	6.58s
7	Yes	1.171	1.018	1min4s
8	Yes	1.082	1.017	12min 12s
9	Yes	1.040	1.018	2h 2min
10	Yes	1.020	1.018	10h 35min

The second set of tests has been conducted with $\log_2(q) = 30$ and $\sigma = 5$ with the highest precision possible of (non modified) `flatter` (with `flatter -rhf 1`):

$\log_2(N)$	secret key recovered	target rhf	effective rhf	wall time
6	Yes	1.073	1.0135	$10.5s$
7	Yes	1.036	1.0138	2min 9s
8	Yes	1.018	1.0144	15min 55s
9	No	1.0089	–	–

7 Comparison with the LWE-decision attack

Let us compare the root Hermit factor needed in the reduction in the decision-LWE attack of [AD21] and our attack (in the Section 4.2 version). For an instance of key dimension N and modulus q the decision attack requires a lattice of dimension $2N+1$ and of determinant q whereas our search-attack only requires a lattice of dimension $2N$ and determinant q.

Assuming the BKZ algorithm is used, a root Hermit factor

$$rhf \ll \left(\frac{q^{\frac{2N}{2N+1}}}{\lambda_1} \right)^{\frac{1}{2N+1}}$$

is required to detect a small vector of size λ_1.

In our case, we are required to have

$$rhf \ll \left(\frac{q^{\frac{1}{2}}}{\lambda_1} \right)^{\frac{1}{2N}}$$

in order to solve search-LWE for key and error of size λ_1.

These quantities being very similar shows that our method is at least as good as the decision-LWE solution. Moreover, our method is easier to use as it does not require a specifically well-studied reduction algorithm, and it skips the search-to-decision reduction step, which calls many instances of the decision algorithm.

References

[AD21] Martin Albrecht and Léo Ducas. "Lattice attacks on NTRU and LWE: a history of refinements". In: *Cryptology ePrint Archive* (2021).

[Ben23] Huck Bennett. "The Complexity of the Shortest Vector Problem". In: *ACM SIGACT News* 54.1 (2023), pp. 37–61.

[BGV14] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. "(Leveled) fully homomorphic encryption without bootstrapping". In: *ACM Transactions on Computation Theory (TOCT)* 6.3 (2014), pp. 1–36.

[Bra12] Zvika Brakerski. "Fully homomorphic encryption without modulus switching from classical GapSVP". In: *Annual Cryptology Conference*. Springer. 2012, pp. 868–886.

[BS13] Wieb Bosma and Ionica Smeets. "Finding simultaneous Diophantine approximations with prescribed quality". In: *The Open Book Series* 1.1 (2013), pp. 167–185.

[Che+17] Jung Hee Cheon et al. "Homomorphic encryption for arithmetic of approximate numbers". In: *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23*. Springer. 2017, pp. 409–437.

[Che13] Nicolas Chevallier. "Best simultaneous Diophantine approximations and multidimensional continued fraction expansions". In: *Mosc. J. Comb. Number Theory* 3.1 (2013), pp. 3–56.

[FV12] Junfeng Fan and Frederik Vercauteren. "Somewhat practical fully homomorphic encryption". In: *Cryptology ePrint Archive* (2012).

[FV23] Machiel van Frankenhuijsen and Edward K Voskanian. "A note on the quality of simultaneous Diophantine approximations obtained by the LLL algorithm". In: *arXiv preprint arXiv:2310.01561* (2023).

[LN20] Jianwei Li and Phong Q Nguyen. "A complete analysis of the BKZ lattice reduction algorithm". In: *Cryptology ePrint Archive* (2020).

[RH23] Keegan Ryan and Nadia Heninger. "Fast practical lattice reduction through iterated compression". In: *Cryptology ePrint Archive* (2023).

[SE94] Claus-Peter Schnorr and Martin Euchner. "Lattice basis reduction: Improved practical algorithms and solving subset sum problems". In: *Mathematical programming* 66 (1994), pp. 181–199.