A Successful Subfield Lattice Attack on a Fully Homomorphic Encryption Scheme

Martha Norberg Hovd*

Simula@UiB
University of Bergen
martha.hovd@uib.no

Abstract

We present the application of a known subfield lattice attack on a fully homomorphic encryption scheme based on NTRU. It is known that for this attack to be successful, a parameter of the scheme must satisfy a lower bound. We derive a second lower bound on the same parameter and show that this bound must be respected if the scheme is to be functional, and furthermore that, in all practical instances of the scheme, the derived second lower bound is greater than the lower bound required for the attack to be applicable. Hence, the scheme is shown to be susceptible to the subfield lattice attack, and furthermore that this susceptibility is inevitable given the current structure of the scheme.

1 Introduction

Fully homomorphic encryption (FHE) schemes are encryption schemes with the following property: for any function f, $\mathrm{Dec}(f(c)) = f(\mathrm{Dec}(c))$, where $c = \mathrm{Enc}(m)$ for a message m. The first such scheme was presented in 2009 [4], and several schemes have been presented since. They mostly follow the same structure and have the same starting point: an encryption scheme where both multiplication and addition of **freshly generated** ciphertexts are homomorphic: $\mathrm{Dec}(\mathrm{Enc}(m_1) + \mathrm{Enc}(m_2)) = m_1 + m_2$, and $\mathrm{Dec}(\mathrm{Enc}(m_1)\mathrm{Enc}(m_2)) = m_1 m_2$.

All these schemes add bounded randomness to the plaintext to obscure it; this randomness is also referred to as 'noise'. The problem is that as more operations are performed on a ciphertext, this noise may grow until it no longer respects the required bounds. At this point, the noise is said to have become unmanageable, as we have no guarantee of correct decryption and the scheme is merely somewhat homomorphic.

In order to have a fully homomorphic scheme, we must reduce the noise, which is usually achieved through a combination of operations. These may stunt the growth of noise, or reduce it slightly. When these no longer suffice, bootstrapping is applied: a homomorphic evaluation of the decryption algorithm. Bootstrapping reduces the noise sufficiently to allow for homomorphic evaluation of any function - it is, however, a very time-consuming procedure. It is, therefore, preferable to construct a fully homomorphic scheme by relying on other strategies and using bootstrapping only as a last resort, as a scheme heavily dependent on bootstrapping is very impractical.

In some cases, the somewhat homomorphic 'starting schemes' are based on previously known schemes, but with different parameter settings, which may result in a less secure scheme. We present one such case, namely a FHE scheme based on NTRU, presented in [7]. We also present an employable attack, described fully in [1]. However, the article [1] only describes the attack, without noting that it may be applied to the NTRU-based fully homomorphic encryption scheme

*The author presented this paper at the NISK 2018 conference.

Proceedings of the 11th Norwegian Information Security Conference, Sep. 19-20, 2018.
NISK Journal, e-ISSN 1894-7735. https://ojs.bibsys.no/index.php/NISK

discussed here. We show this, and also how the susceptibility of this attack is inevitable. This is done by rigorously deriving a lower bound on a crucial parameter of the encryption scheme, and showing that this bound must be satisfied if the scheme is to be functional. Furthermore, it is noted that another lower bound on the same parameter must be met for the attack to be applicable, and finally that this lower bound is typically smaller than the lower bound derived from the scheme itself. It follows that the scheme may be attacked in all practical instances of it.

It is important to note that it is the derived lower bound which makes the scheme vulnerable to the subfield lattice attack, and furthermore that this lower bound is a result of the additional noise-reducing operations of the scheme. In particular, this means that the original NTRU encryption scheme does not share the same vulnerability to this attack.

2 Preliminaries

2.1 Notation

All vectors are row vectors and will be denoted with bold lower case letters: $\mathbf{v}, \mathbf{w}$, while matrices will be denoted using bold upper case letters: $\mathbf{A}, \mathbf{B}$. Elements of either a vector, a matrix or a (polynomial) ring will be denoted with a lower case letter in italics: a, b. Vectors will be written as $\mathbf{a} = [a_1, a_2, \dots, a_n]$, whereas sets will be denoted by $\{0, 1, \dots\}$.

Multiplication of integers, or an integer and a vector or polynomial is denoted by simple juxtaposition: $ab, a\mathbf{v}, af(x)$. Multiplication of a vector and a matrix will be denoted by a single dot: $\mathbf{v} \cdot \mathbf{A}$, and finally, the multiplication of two polynomials will be denoted by an asterisk: $f * g$. Furthermore, this polynomial multiplication always takes place in some polynomial ring, and the main motivation of the multiplicative notation is to serve as a reminder of this during computations. It should be clear from the context whether or not a given element is a polynomial, and any polynomial f will therefore, with very few exceptions, not be written $f(x)$.

Let $\mathbf{v}, \mathbf{w}$ be vectors of the same length k over a polynomial ring R. We then define the dot product of these two vectors as $\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{i=1}^{k} v_i * w_i \in R$. In addition, we have the following notation: for any two polynomials $a = \sum_{i=0}^{n-1} a_i x^i,\ b = \sum_{i=0}^{n-1} b_i x^i$, let $[a, b]$ denote the vector $[a_0, \dots, a_{n-1}, b_0, \dots, b_{n-1}]$.

We operate with the two following modular reductions: $p = r \ \underline{\mathrm{mod}}\ q$ reduces p modulo q to $r \in (-q/2, q/2]$, whilst $p = r \mod q$ denotes the modular reduction to $r \in [0, q-1]$. Note that the only difference in the notation is the underlining of the first mod. In both cases, we may also write $p \equiv r \ \underline{\mathrm{mod}}\ q$ or $p \equiv r \mod q$ if we wish to stress that p is equivalent to r modulo q: $p = r + kq$, for $k \in \mathbb{Z}$. This generalizes to vectors and polynomials.

The Euclidean norm of a vector $\mathbf{v}$ is denoted by $\|\mathbf{v}\| = \sqrt{v_1^2 + v_2^2 + \cdots + v_n^2}$, whilst $\|\cdot\|_\infty$ denotes the infinity norm: $\|\mathbf{v}\|_\infty = \max_i\{|v_i|\}$. Supposing f is a polynomial, $\|f\|, \|f\|_\infty$ refers to calculating either norm of the coefficient vector of f.

For a probability distribution χ, $x \leftarrow \chi$ refers to drawing x according to χ. Furthermore, any logarithm log will be to the base 2.

Finally, throughout this paper, the following lemma will prove quite useful. Here, $R = \mathbb{Z}[x]/f(x)$ for a polynomial f of degree n.

Lemma 2.1. *The following bound holds for any two elements $a, b \in R$:*

$$\|a * b\|_\infty \leq n\|a\|_\infty\|b\|_\infty.$$

Proof. Seeing as $a_i \leq \|a\|_\infty,\ b_i \leq \|b\|_\infty\ \forall i \in \{0, 1, \ldots, n-1\}$, it follows that $a_i b_j \leq \|a\|_\infty\|b\|_\infty$. Since every coefficient of $a*b$ is the sum of n terms $a_i b_j$, it holds that $\|a*b\|_\infty \leq n\|a\|_\infty\|b\|_\infty$. □

2.2 Lattices

Definition 2.2. *Let $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_\eta\}$ be a set of linearly independent vectors, with $\mathbf{v}_i \in \mathbb{R}^m$ $\forall i \in \{1, \ldots, \eta\}$. The lattice $\mathcal{L}$ generated by $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_\eta$ is the set of linear combinations of these vectors with coefficients in $\mathbb{Z}$:*

$$\mathcal{L} = \{a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \cdots + a_\eta\mathbf{v}_\eta : a_1, a_2, \ldots, a_\eta \in \mathbb{Z}\}.$$

A basis for the lattice $\mathcal{L}$ is any set of independent vectors that generates $\mathcal{L}$, and any two such sets will have the same dimension. Suppose $m = \eta$, we may then represent a basis by a square matrix (where the basis vectors form the rows of the matrix) and so we may calculate the determinant of it. There are of course many possible bases of a lattice $\mathcal{L}$, but as Proposition 7.14 of [6] shows, any two bases of a lattice are related by an integer matrix with determinant ± 1. It follows from this that for any two basis matrices $\mathbf{B}, \mathbf{B}'$ we have: $|\det(\mathbf{B})| = |\det(\mathbf{B}')|$. Based on this, we have the following lattice invariant:

Definition 2.3. *Let $\mathcal{L}$ be a lattice of dimension η with basis $B = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_\eta\}$, where $\mathbf{v}_i \in \mathbb{R}^\eta \quad \forall i \in \{1, 2, \ldots, \eta\}$. The determinant of $\mathcal{L}$ is defined as*

$$\det(\mathcal{L}) = |\det(\mathbf{B})|.$$

Any vector $\mathbf{v} \in \mathcal{L}$ has a (Euclidean) length. Given this property, we may formulate the shortest vector problem of a lattice $\mathcal{L}$ [6]:

The shortest vector problem (SVP): Find a shortest nonzero vector in a lattice $\mathcal{L}$, i.e. find a nonzero vector $\mathbf{v} \in \mathcal{L}$ that minimizes $\|\mathbf{v}\|$.

It may be shown that solving SVP is NP-hard under the randomized reduction hypothesis [6]. Due to this proven hardness, SVP is used in cryptographic settings, so that breaking an encryption scheme requires solving SVP for a certain instance. However, solving SVP precisely is not always necessary; in some cases, it may suffice to compute merely an approximation of the vectors in question; that is, solving the following problem [6]:

Approximate-SVP: Let $\psi(\eta)$ be a function of the lattice dimension η of a lattice $\mathcal{L}$, with $\|\mathbf{v}_0\|$ the length of the shortest vectors in $\mathcal{L}$. Find a vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| \leq \psi(\eta)\|\mathbf{v}_0\|$.

Of course, the length of the shortest vector $\mathbf{v}_0 \in \mathcal{L}$ is not always given, but an upper bound on $\|\mathbf{v}_0\|$ is always given by the following theorem:

Theorem 2.4 (Hermite's Theorem (Theorem 7.25 [6]))**.** *Every lattice $\mathcal{L}$ of dimension η has at least one vector $\mathbf{v} \in \mathcal{L}$ satisfying $\|\mathbf{v}\| \leq \sqrt{\eta}\det(\mathcal{L})^{1/\eta}$.*

Another result by Hermite is that for every lattice $\mathcal{L}$ of dimension η there exists a Hermite constant, γ_η, such that $\|\mathbf{v}_0\| \leq \sqrt{\gamma_\eta}\det(\mathcal{L})^{1/\eta}$. This constant is generally not known. However, we may use the expression to rephrase the approximate-SVP into the Hermite Shortest Vector Problem [3]:

HSVP: Given a lattice $\mathcal{L}$ and an approximation factor $\alpha > 0$, find a non-zero vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| \leq \alpha \det(\mathcal{L})^{1/\eta}$.

The approximation factor α may be expressed as δ^η, in which case δ is known as the Hermite root factor.

Of course, a solution to any of these problems is seldom apparent given a basis B for a lattice, and the most efficient way of solving any of the presented problems is to find a basis which contains the solution of either stated problem. This is known as basis reduction, and the main algorithms are LLL and a generalisation of it, BKZ, both of which are HSVP-algorithms [3].

LLL works by swapping two vectors in the basis and performing a reduction, whereas BKZ works similarly, only with more than two vectors. The number of vectors BKZ works with is known as the block size, denoted by β. The larger β is, the more precise the result of BKZ will be. Although the algorithms are not fully understood, it is known that BKZ outperforms LLL. BKZ also performs much better, both with respect to time and the resulting approximation factors, than any theoretical bound predicts.

2.3 An Introduction to NTRU and its Security

The original NTRU encryption scheme [5] is defined over the polynomial ring $\mathbb{Z}[x]/(x^N - 1)$ for an integer N. The integer $q > 1$ is an additional parameter of the scheme, as most operations are performed modulo q.

A complete description of the NTRU-based FHE scheme to be presented in section 4 may be found in [7]. The scheme follows the general structure of NTRU quite closely. The secret key of the scheme is a polynomial $f \leftarrow \chi$, for a distribution χ over R, and we require f to be invertible modulo q. The public key is defined as $h = f^{-1} * g \bmod q$, for $g \leftarrow \chi$. The main difference in the setup is that this scheme is defined over the polynomial ring $R = \mathbb{Z}[x]/(x^n + 1)$, where we require $n = 2^k$.

A possible attack is to try to find the secret key f based solely on the public information q and h, and one way to do this is to reformulate the problem into one related to lattices. This is done by constructing a $2n \times 2n$ basis matrix for a lattice $\mathcal{L}_{\text{NTRU}}$. For an NTRU public key polynomial $h(x) = h_0 + \cdots + h_{n-1}x^{n-1}$, the basis matrix of the lattice $\mathcal{L}_{\text{NTRU}}$ is:

$$\mathbf{B}_{\text{NTRU}} = \begin{bmatrix} 1 & 0 & \dots & 0 & h_0 & h_1 & \dots & h_{n-1} \\ 0 & 1 & \dots & 0 & -h_{n-1} & h_0 & \dots & h_{n-2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -h_1 & -h_2 & \dots & h_0 \\ 0 & 0 & \dots & 0 & q & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & q \end{bmatrix}.$$

Recall that $h = g * f^{-1}$ and $f * f^{-1} = 1 + qf'$, so we must have $f * h = g + qu$ for some polynomial $u = g * f'$.

Proposition 2.5. *For the polynomials f, g and u described above, we have: $[f, -u] \cdot \mathbf{B}_{\text{NTRU}} = [f, g]$.*

Proof. The n first coefficients of the resulting vector of $[f, -u] \cdot \mathbf{B}_{\text{NTRU}}$ are obviously f. Coefficient $n+1+k$, for $k \in \{0, 1, \dots n-1\}$ is expressed as:

$$\sum_{\substack{i,j=0 \\ i+j=k}}^{n-1} f_i h_j - \sum_{\substack{i,j=0 \\ i+j=k+n}}^{n-1} f_i h_j - q u_k = g_k + q u_k - q u_k = g_k,$$

where the fact that $x^n \equiv -1$ in $R = \mathbb{Z}[x]/(x^n+1)$ has been applied. Hence, $[f, -u] \cdot \mathbf{B}_{\text{NTRU}} = [f, g]$, meaning $[f, g]$ belongs to $\mathcal{L}_{\text{NTRU}}$, as the vector may be expressed as a linear combination of the basis vectors of $\mathcal{L}_{\text{NTRU}}$ using only integers. □

Supposing $[f, g]$ is among the shortest vectors in the lattice $\mathcal{L}_{\text{NTRU}}$, it follows that if an adversary is able to solve SVP in $\mathcal{L}_{\text{NTRU}}$, she is able to compute f based solely on public information, and thus break the scheme. Furthermore, any pair of polynomials $[\bar{f}, \bar{g}]$ with sufficiently small coefficients satisfying the relation $\bar{f} * h = \bar{g} \bmod q$ will also suffice, as will probably any solution to approximate-SVP for an approximation factor smaller than $\sqrt{n}$ [6]. Thus, recovering the secret key f of the encryption scheme reduces to solving approximate-SVP or HSVP for the lattice $\mathcal{L}_{\text{NTRU}}$. We stress again that for this strategy to work, we require the vector $[f, g]$ to be among the shortest vectors in the lattice $\mathcal{L}_{\text{NTRU}}$.

3 Subfield Lattice Attack

There may be more efficient attacks than merely applying LLL or BKZ on the lattice basis, depending on the properites of the scheme. We present one such attack here, described fully in [1].

3.1 Algebraic Background

Let $\mathbb{K} = \mathbb{Q}[\omega]$ be a field, for a root of unity ω of order $2n$, for n a power of 2, and let $\mathbb{L}$ be a subfield of $\mathbb{K}$ such that $\mathbb{L} = \mathbb{Q}[\omega']$, for ω' a root of unity of order $2n'$, where n' also is a power of 2, and define $\rho = n/n'$. These fields will have rings of integers $\mathbb{Z}[\omega]$ and $\mathbb{Z}[\omega']$, respectively. These rings of integers may be shown to be isomorphic to the polynomial rings $R = \mathbb{Z}[x]/(x^n+1)$ and $R' = \mathbb{Z}[x]/(x^{n'}+1)$.[1]

We know from Galois theory that there is a Galois group G' of automorphisms $\{\varphi_i\}$ on $\mathbb{K}$ that fixes $\mathbb{L}$ pointwise [1]. Using these automorphisms, we may define the norm function $\mathrm{N}_{\mathbb{K}/\mathbb{L}} : \mathbb{K} \to \mathbb{L}$, as $\mathrm{N}_{\mathbb{K}/\mathbb{L}}(a) = \prod_{\varphi_i \in G'} \varphi_i(a)$.

3.2 The Attack

Given an instance of an NTRU-based encryption scheme, with $sk = f$ and $pk = h = f^{-1} * g$, we define $f' = \mathrm{N}_{\mathbb{K}/\mathbb{L}}(f), g' = \mathrm{N}_{\mathbb{K}/\mathbb{L}}(g), h' = \mathrm{N}_{\mathbb{K}/\mathbb{L}}(h)$ and a new lattice $\mathcal{L}'_{\text{NTRU}}$ defined by h' and q as described in Subsection 2.3. The approach of the attack is to find a short vector $[x', y'] \in \mathcal{L}'_{\text{NTRU}}$ by performing LLL on the basis B'_{NTRU} and lift this vector up to $[x, y]$ in the original lattice, using the canonical inclusion map. If the vector $[x', y']$ satisfies certain properties, the vector $[x, y]$ will be short in $\mathcal{L}_{\text{NTRU}}$, and might therefore function as a secret key.

The actual attack rests on the following heuristic:

Heuristic 3.1 (Heuristic 1 [1]). *For any n and any $f, g \in R$ with reasonable isotropic distribution of variance σ^2 and any constant $c > 0$, there exists a constant C such that $\|f'\| \leq (\sigma n^C)^\rho$ and $\|g'\| \leq (\sigma n^C)^\rho$, except with probability $\mathcal{O}(n^{-c})$.*

Moreover, Theorem 1 of [1] assures us of the existence of a lattice reduction algorithm with block-size β which is able to find a vector $[x', y'] \in R'$ such that $\|[x', y']\| \leq \beta^{\Theta(2n'/\beta)}\|\mathbf{v}_0\|$ when applied to the basis of the lattice $\mathcal{L}'_{\text{NTRU}}$, with $\|\mathbf{v}_0\|$ the length of the shortest vectors in the lattice. When combined with the observation that $\|\mathbf{v}_0\| \leq \|[f', g']\|$ and Heuristic 3.1, we conclude that there exists a lattice reduction algorithm able to find a vector $[x', y'] \in R'$ such that

$$\|[x', y']\| \leq \beta^{\Theta(n/\beta\rho)}\|[f', g']\| \leq \beta^{\Theta(n/\beta\rho)}(n\sigma)^{\Theta(\rho)}.$$

Furthermore, we also have the following theorem:

Theorem 3.2 (Theorem 2 [1]). *Let $f', g' \in R'$ be such that $\langle f' \rangle$ and $\langle g' \rangle$ are coprime ideals[1] and $h' * f' = g' \bmod q$ for some $h' \in R'$. If $[x', y'] \in \mathcal{L}'_{\text{NTRU}}$ has length satisfying*

$$\|[x', y']\| < \frac{q}{\|[f', g']\|}, \tag{1}$$

then $[x', y'] = v[f', g']$ for some $v \in R'$.

Based on the result derived from Heuristic 3.1 and Theorem 1 from [1], we conclude that for bound (1) to hold, and therefore for the attack to succeed, we must require

$$\beta^{\Theta(n/\beta\rho)}(n\sigma)^{\Theta(\rho)} \leq q. \tag{2}$$

Once the vector $[x', y']$ has been found, we lift $x', y' \in R'$ to R using the canonical inclusion map $L : \mathbb{L} \to \mathbb{K}$:

$$\begin{aligned} x &= L(x') = L(v) * L(f'), \\ y &= L(y') * h/L(h') \bmod q = L(v) * L(g') * h/L(h') \bmod q, \end{aligned}$$

Here, v is as in Theorem 3.2. For simplicity, we set $\tilde{f} = L(f')/f$, $\tilde{g} = L(g')/g$ and $\tilde{h} = L(h')/h$; we then have

$$\begin{aligned} x &= L(v) * \tilde{f} * f \bmod q \\ y &= L(v) * L(g')/\tilde{h} = L(v) * g * \tilde{g}/\tilde{h} = L(v) * \tilde{f} * g \bmod q \\ \Rightarrow [x, y] &= u * [f, g] \in \mathcal{L}_{\text{NTRU}} \quad \text{with } u = L(v) * \tilde{f} \in R. \end{aligned}$$

In other words: the subfield attack yields a (small) multiplicative of $[f, g]$ under certain reasonable assumptions.

4 A Fully Homomorphic Encryption Scheme based on NTRU

4.1 The Somewhat Homomorphic Encryption Scheme

As mentioned in Section 2.3, the encryption scheme presented here is fully described in [7] and is defined over the polynomial ring $R = \mathbb{Z}[x]/(x^n + 1)$, for n a power of 2. We also have the

[1] The authors of [1] note that the probability of $\langle f' \rangle$ and $\langle g' \rangle$ being coprime is roughly 3/4, and also that it does not seem strictly necessary for the attack to be successful in practice.

integer parameters q, p, chosen such that $q \gg p \geq 2$ and $\gcd(p, q) = 1$. Given these integers, we define the rings $R_p = \mathbb{Z}_p[x]/(x^n + 1)$ and $R_q = \mathbb{Z}_q[x]/(x^n + 1)$. In addition, we have a probability distribution χ, which will typically be some discrete Gaussian distribution. The scheme consists of the following operations:

KeyGen: Draw $f \leftarrow \chi$ such that $f \equiv 1 \mod p$ and $\exists f^{-1} \underline{\mod} q$. Draw $g \leftarrow \chi$ as well, and output $pk = h = g * f^{-1} \underline{\mod} q$ and $sk = f$.

Enc($pk = h, m \in R_p$): Draw $e, r \leftarrow \chi$ such that $e \equiv m \mod p$.
Output $c = pr * h + e \underline{\mod} q$, $d = 1$.

Dec($sk = f, c \in R_q$, d): Compute $\bar{b} = f^d * c \underline{\mod} q$ and lift this to the integer polynomial $b \in R$ with coefficients in $(-q/2, q/2]$. Output $m = b \underline{\mod} p$.

EvalAdd(c_0, c_1, d_0, d_1): Output: $c = c_0 + c_1 \underline{\mod} q$, $d = \max(d_0, d_1)$.

EvalMult(c_0, c_1, d_0, d_1): Output: $c = c_0 * c_1 \underline{\mod} q$, $d = d_0 + d_1$.

The two last operations are the homomorphic operations, and it is also these that necessitate the notion of the degree d of a ciphertext, which denotes the power of f^{-1} in the ciphertext. Note that $f^k * b = m \underline{\mod} p$ for any power $k \geq 0$, whilst this is not necessarily the case for f^{-1}, as there is no guarantee that $f^{-1} = 1 \underline{\mod} p$. Therefore, the decryption procedure will decrypt any ciphertext of degree at most the given d, assuming $f^d * c = f^k * b \underline{\mod} q$, which is the reason $d = \max(d_0, d_1)$ in EvalAdd.

The polynomials f, g, r and e must be chosen so that they ensure correct decryption - meaning χ should have parameters ensuring that these polynomials are "short enough". What precisely this entails will be discussed at some length throughout this section. Essentially: we will derive bounds on the coefficients of these polynomials to ensure correct decryption, even after the noise reducing operations have been performed on a ciphertext. The resulting bounds will be used to derive a final bound on q.

We start with the lower bound imposed on the coefficients of the polynomials f, g, r and e to ensure correct decryption of a freshly generated ciphertext.

Proposition 4.1. *In order for the presented encryption scheme to correctly decrypt a freshly generated ciphertext, we require every coefficient of the polynomials f, g, r and e to be strictly less than $\sqrt{\frac{q}{4pn}}$.*

Proof. The decryption of $c = pr * h + e \underline{\mod} q$ proceeds as follows, when viewed as an operation in R, as opposed to R_q:

$$\begin{aligned}\bar{b} = f * c = f * (pr * h + e) = pf * r * g * f^{-1} + f * e \\ = pq * r * g * f' + pr * g + f * e,\end{aligned}$$

where $f * f^{-1} = qf' + 1$. Consider the polynomial $pr * g + f * e$ in R. To ensure correct decryption, we need every coefficient of this polynomial to be of absolute value less than $q/2$, or else we get $b = pr * g + f * e - q \sum_{i=0}^{n-1} a_i x^i$ where some $a_i \neq 0$ and hence, $b \underline{\mod} p$ need not equal m. We therefore require $\|pr * g + f * e\|_\infty < q/2$. Using the triangle inequality and Lemma 2.1, we may compute:

$$\begin{aligned}\|pr * g + f * e\|_\infty &\leq \|pr * g\|_\infty + \|f * e\|_\infty \\ &\leq pn\|r\|_\infty\|g\|_\infty + n\|f\|_\infty\|e\|_\infty\end{aligned}$$

$$\leq pn\|r\|_\infty\|g\|_\infty + pn\|f\|_\infty\|e\|_\infty \leq 2pnB^2, \tag{3}$$

for B a bound on the largest coefficient of r, g, f and e. If we assume (3) is less than $q/2$, then any fresh ciphertext will decrypt correctly. This requires the polynomials r, g, f and e to be sampled from a distribution χ which ensures that any coefficient is strictly less than $\sqrt{\frac{q}{4pn}}$. □

4.2 Noise reductions

The scheme in [7] uses key switching, ring reduction, modulus switching and bootstrapping as strategies to reduce the noise of a ciphertext, and thus turn the somewhat homomorphic scheme into a fully homomorphic encryption scheme. However, only key switching and modulus switching are being performed after every multiplication; hence we shall only focus on these two operations.

Note that this, strictly speaking, only makes the scheme presented here somewhat homomorphic, as we need bootstrapping to make it truly fully homomorphic. Nevertheless, we refer to the scheme presented here as a fully homomorphic scheme, mainly to separate it from the "starting scheme" presented in Subsection 4.1, and refer the interested reader to [7] to study the bootstrapping procedure.

4.2.1 Key Switching

Key switching converts a ciphertext of degree at most d encrypted under f_1 into a ciphertext of degree 1 encrypted under the secret key f_2. This procedure requires a hint, namely $a_{1\to 2} = \bar{a} * f_1^d * f_2^{-1} \bmod q$, for $\chi \to \bar{a} \equiv 1 \mod p$. Based on this hint, the actual key switching is the procedure

KeySwitch$(c_1, a_{1\to 2})$: Output: $c_2 = a_{1\to 2} * c_1 \bmod q$.

Proposition 4.2. *Suppose c_1 is an encryption of m under f_1 of degree d which decrypts correctly:* $\mathrm{Dec}(f_1, c_1, d) = m$. *In order to ensure* $\mathrm{Dec}(f_2, c_2, 1) = m$, *for $a_{1\to 2}$ and c_2 generated according to the above procedure, we require every coefficient of f_1, f_2, g, r, e and $\bar{a}$ to be strictly less than* $(\frac{q}{2^{d+1}p^d n^{2d}})^{\frac{1}{2d+1}}$.

Proof. Decryption of c_2 results in:

$$\begin{aligned}\bar{b}_2 &= f_2 * c_2 = f_2 * a_{1\to 2} * c_1 = f_2 * \bar{a} * f_1^d * f_2^{-1} * c_1 \\ &\equiv \bar{a} * f_1^d * c_1 \equiv \bar{a} * \bar{b}_1 \bmod q\end{aligned}$$

In order to have correct decryption, we require $\|\bar{a} * \bar{b}_1\|_\infty < q/2$, to ensure that $b_2 = \bar{a} * b_1 = \bar{a} * m = m \bmod p$. Seeing as c_1 is a ciphertext of degree d, it must be the result of $d-1$ multiplications, wlog let $\bar{b}_1 = f_1^d * (pr * g * f_1^{-1} + e)^d \bmod q$. For the assumption $\|\bar{a} * \bar{b}_1\|_\infty < q/2$ to hold, we must have:

$$\begin{aligned}\|\bar{a} * f_1^d * (pr * g * f_1^{-1} + e)^d\|_\infty &= \|\bar{a} * f_1^d * \sum_{i=0}^{d} \binom{d}{i} p^i r^i * g^i * f_1^{-i} * e^{d-i}\|_\infty \\ &= \|\bar{a} * \sum_{i=0}^{d} \binom{d}{i} p^i r^i * g^i * f_1^{d-i} * e^{d-i}\|_\infty \\ \text{By Lemma 2.1} \qquad &\leq n^{2d}\|\bar{a}\|_\infty \sum_{i=0}^{d} \binom{d}{i} p^i \|r\|_\infty^i \|g\|_\infty^i \|f\|_\infty^{d-i} \|e\|_\infty^{d-i}\end{aligned}$$

$$\leq p^d n^{2d} B^{2d+1} \sum_{i=0}^{d} \binom{d}{i} \leq 2^d p^d n^{2d} B^{2d+1} < q/2.$$

Here, B is a bound on the largest coefficient in $\bar{a}, r, g, f$ and e, and the requirement of this being strictly less than $(\frac{q}{2^{d+1}p^d n^{2d}})^{\frac{1}{2d+1}}$ to ensure correct decryption after switching keys immediately follows. □

It follows that key switching should be performed after every multiplication to minimize this bound. In the case $d = 2$ we have:

$$B^5 < \frac{q}{8p^2n^4}. \tag{4}$$

4.2.2 Modulus Switching

Modulus switching converts a ciphertext from modulus q to a smaller modulus, $\bar{q} = q/q'$ for some factor q' of q, which reduces the underlying noise by a factor of approximately q'. Seeing as $q'|q$, it follows that $\gcd(q', p) = 1 \Rightarrow \exists v$ s.t. $v = (q')^{-1} \bmod p$. The procedure $\text{ModSwitch}(c, q, q')$ is performed as follows:

1. Compute a short $\varrho \in R$ such that $\varrho = c \bmod q'$.
2. Compute a short $\Delta \in R$ such that $\Delta = (q'v - 1)\varrho \bmod (pq')$.
3. Let $\varrho' = c + \Delta \bmod q$. Note that q' divides ϱ' by construction.
4. Output $c' = (\varrho'/q') \in R_{\bar{q}}$.

Note that the final step indirectly multiplies ϱ with v, which may easily be compensated for by either multiplying with q' in the final step of the decryption procedure or ensuring that $q' \equiv 1 \mod p$.

Proposition 4.3. *Suppose c is an encryption of degree 1 of the message m under the secret key f. Let $c' = \text{ModSwitch}(c, q, q')$; then, in order to have $v\text{Dec}(f, c, 1) = \text{Dec}(f, c', 1)$, we require every coefficient of f, g, r and e to be less than or equal to B, which again needs to satisfy $\frac{1}{q'}(2pnB^2 + nB\frac{pq'}{2}) < \frac{q}{2q'}$*

Proof. Let $\bar{q} = q/q'$. As $\varrho = c \bmod q'$ and $v = (q')^{-1} \bmod p$, we may write

$$\varrho = c - q'l \qquad \text{for } l \in R, \qquad q'v = 1 + pk \qquad \text{for } k \in \mathbb{Z}.$$

Following the procedure, we have[2]:

$$\begin{aligned}
(q'v - 1)\varrho &= (pk + 1 - 1)(c - q'l) = pk(c - q'l) = pkc - pq'kl. \\
\Rightarrow \Delta &= pkc - pq's \text{ for } s \in R, \quad \text{as} \quad R \ni \Delta = (q'v - 1)\varrho \equiv pkc \bmod pq'. \\
\varrho' &= c + \Delta \bmod q = c + pkc - pq's = (1 + pk)c - pq's \\
&\equiv q'vc - pq's \bmod q. \\
c' &= \varrho'/q' \equiv vc - ps \bmod \bar{q}.
\end{aligned}$$

[2]Throughout this proof, pk denotes p multiplied with k, not the public key.

In order to guarantee correct decryption, we require $\|vc - ps\|_\infty < \bar{q}/2$, so that:

$$\begin{aligned} f * c' &= vf * c - pf * s = v(pr * g(qf' + 1) + f * e) - pf * s \in R \\ &\equiv vpg * r + vf * e - pf * s \bmod \bar{q} \end{aligned}$$

If we are to have equality rather than equivalence, we must have $\|vf*e+vpg*r-pf*s\|_\infty < \bar{q}/2$, so that: $(f * c' \bmod \bar{q}) \bmod p = vm$. Thus, for the decryption of c' to be successful, we need to have the following:

$$\|f * c'\|_\infty = \|f * (c + \Delta)/q'\|_\infty \leq \frac{1}{q'}(\|f * c)\|_\infty + \|f * \Delta\|_\infty)$$

We use $c = p * r * g * f^{-1} + e$ as well as Lemma 2.1 and derive:

$$\begin{aligned} \frac{1}{q'}(\|pg * r + f * e\|_\infty + \|f * \Delta\|_\infty) &\leq \frac{1}{q'}(2pnB^2 + nB\|\Delta\|_\infty) \\ &\leq \frac{1}{q'}(2pnB^2 + nB\frac{pq'}{2}) < q/2q'. \end{aligned} \tag{5}$$

□

4.3 ComposedEvalMult and the Growth of q

The operation ComposedEvalMult is simply the sequential execution of EvalMult, KeySwitch and ModSwitch.

Proposition 4.4. *Suppose c_0, c_1 are two ciphertexts encrypted under the public key $h = g*f_1^{-1}$, both of degree 1. Correctness of ComposedEvalMult means*

$$\mathrm{Dec}(f_2, \mathrm{ComposedEvalMult}(c_0, c_1), 1) = \mathrm{Dec}(f_1, c_0, 1) * \mathrm{Dec}(f_1, c_1, 1),$$

where f_2 is the new secret key after KeySwitch has been performed. To achieve this, we require the polynomials $f_1, f_2, g, r_0, r_1, e_0, e_1$ and $\bar{a}$ to all be drawn from a distribution χ so that their largest coefficient is smaller than B, and that B satisfies

$$\frac{1}{q'}(4p^2n^4B^5 + nB\frac{pq'}{2}) < \frac{q}{2q'}.$$

Proof. Based on the proofs of propositions 4.2 and 4.3, it follows that

$$f_2 * \mathrm{ComposedEvalMult}(c_0, c_1) \equiv \bar{b} \bmod \bar{q},$$

where $\bar{b} = m_0 * m_1 \bmod p$. What needs to be calculated is the bound we require on the drawn polynomials so that the noise added during multiplication and switching keys is sufficiently lowered by switching the modulus. The ciphertext $\mathrm{ComposedEvalMult}(c_0, c_1)$ outputs is on the form $c = \frac{1}{q'}(a_{1\to 2} * c_0 * c_1 + \Delta)$ for a factor q' of q. We have the following:

$$\begin{aligned} f_2 * c &= f_2 * \frac{1}{q'}(a_{1\to 2} * c_0 * c_1 + \Delta) \\ &= \frac{1}{q'} f_2 * (\bar{a} * f_2^{-1} * f_1^2 * (pr_0 * g * f_1^{-1} + e_0)(pr_1 * g * f_1^{-1} + e_1) + \Delta) \end{aligned}$$

$$= \ldots \equiv \frac{1}{q'}(p^2\bar{a} * r_0 * r_1 * g^2 + p\bar{a} * r_0 * g * f_1 * e_1 + p\bar{a} * r_1 * g * f_1 * e_0 + \bar{a} * f_1^2 * e_0 * e_1 + f_2 * \Delta) = b' \equiv \bar{b} \bmod \bar{q}.$$

We need $\|b'\|_\infty < \bar{q}/2$, so $b' = \bar{b}$. To achieve a bound on the coefficients of the polynomials, we use Lemma 2.1 and set

$$\|\bar{a}\|_\infty = \|r_0\|_\infty = \|r_1\|_\infty = \|g\|_\infty = \|f_1\|_\infty = \|f_2\|_\infty = \|e_0\|_\infty = \|e_1\|_\infty = B,$$

and we compute:

$$\begin{aligned}\|b'\|_\infty &\leq \frac{1}{q'}(p^2n^4B^5 + 2pn^4B^5 + n^4B^5 + nB\|\Delta\|_\infty) \\ &\leq \frac{1}{q'}(4p^2n^4B^5 + nB\frac{pq'}{2}),\end{aligned} \tag{6}$$

which has to be smaller than $\frac{q}{2q'}$ if ComposedEvalMult is to be correct. □

Given this final bound all the coefficients of the noise-inducing polynomials need to satisfy, we may use this to at last derive a bound on q. This bound will depend on other parameters of the scheme and the probability distribution χ.

Suppose any of the polynomials affecting the noise level are drawn from a discrete Gaussian distribution of parameter r, and set w as an assurance measure; meaning it should be practically impossible for any polynomial drawn from this distribution to have a Euclidean length greater than rw. It follows that we may set a bound on the infinity norm of any such distributed polynomial as $\frac{rw}{\sqrt{n}}$. Using this bound and expression (6), we require

$$\frac{1}{q'}(4p^2n^4(\frac{rw}{\sqrt{n}})^5 + n\frac{rw}{\sqrt{n}}\frac{pq'}{2}) = \frac{1}{q'}(4p^2n^{1.5}r^5w^5 + \frac{1}{2}pq'\sqrt{n}rw) < q/2q'$$

for decryption to still be correct after a call to ComposedEvalMult.

Suppose $4p^2n^{1.5}r^5w^5 < q'$, we then have $1 + \frac{1}{2}p\sqrt{n}rw < q/2q'$, where the value $q/q' \geq q_1$, a single factor of q, and thus the smallest possible ciphertext modulus. In theory, q could have a factor q_1 significantly smaller than the rest, as we could set this as the final ciphertext modulus, which would not be subjected to a modulus switching. We would therefore only require such a factor to be large enough to decrypt ciphertexts that have been subjected to D modulus switchings. A more practical approach however, is to set the following universal bound for any factor of q, as the authors of [7] do:

$$q_i > 4p^2r^5w^5n^{1.5}. \tag{7}$$

We may therefore conclude that ensuring any factor of q satisfies bound (7) results in an encryption scheme which reduces the noise sufficiently to guarantee correct decryption of any freshly generated ciphertext and output of ComposedEvalMult, given that the input ciphertexts has at most the same noise level as any freshly generated ciphertexts for the current ciphertext modulus $\bar{q}$. As a result, q should satisfy the following lower bound, as anything else might result in decryption failure:

$$q > (4p^2r^5w^5n^{1.5})^{D+1}. \tag{8}$$

5 Subfield Lattice Attack on the NTRU-based Fully Homomorphic Encryption Scheme

5.1 Applicability and Success of the Attack

It remains to be shown that the attack of section 3 is at all applicable to the scheme in section 4, and also that it will be successful.

We note first that the authors of [1] state in particular that Heuristic 3.1 holds for the Gaussian distribution, which is the distribution suggested by the authors of [7] for the encryption scheme. It is therefore possible to apply the attack on this scheme.

However, as noted in the last paragraph of subsection 2.3, an attack based on solving SVP or approximate-SVP for the lattice $\mathcal{L}_{\text{NTRU}}$ rests on the assumption that $[f, g]$ is among the shortest vectors in this lattice. We therefore need this assumption to hold, if the attack described in section 3 is to produce a vector which may be used as a secret key. As the following proposition shows, the necessary assumption holds:

Proposition 5.1. *With overwhelming probability, the vector $[f, g]$ is one of the shortest vectors in the lattice $\mathcal{L}_{\text{NTRU}}$.*

Proof. Recall Theorem 2.4: the length of the shortest vector in any lattice $\mathcal{L}$ is at most $\sqrt{\eta}\det(\mathcal{L})^{1/\eta}$. In the case of $\mathcal{L}_{\text{NTRU}}$, we get $\|\mathbf{v}_0\| \leq \sqrt{2n}(q^n)^{1/2n} = \sqrt{2nq}$.

We may in addition calculate a bound on $\|[f, g]\|$, using the upper bound $\|f\|_\infty, \|g\|_\infty < \sqrt{\frac{q}{4pn}}$, derived in the proof of Proposition 4.1:

$$\|[f, g]\| = \sqrt{f_0^2 + \dots f_{n-1}^2 + g_0^2 + \dots + g_{n-1}^2} \leq \sqrt{2n\left(\sqrt{\frac{q}{4pn}}\right)^2} = \sqrt{\frac{q}{2p}}.$$

Comparing the two bounds, we have: $\sqrt{\frac{q}{2p}} \Big/ \sqrt{2nq} = \sqrt{\frac{1}{4pn}} \ll 1$. Thus, seeing as the bound on $\|[f, g]\|$ is much smaller than the Hermite bound, it is highly probable that $[f, g]$ is one of the shortest vectors in $\mathcal{L}_{\text{NTRU}}$. □

Thus, the attack is applicable to the scheme at hand, and will produce a vector usable as a secret key should it be successful.

With regards to the success of the attack: recall bound (2) of subsection 3.2:

$$\beta^{\Theta(n/\beta\rho)}(n\sigma)^{\Theta(\rho)} \leq q,$$

which needs to be satisfied if the attack is to succeed. This is obviously more likely as q grows in relation to n - in other words, the more factors q consists of, allowing for more CompEvalMult operations to be performed.

If we wish to allow for D modulus switchings to be applied - meaning at most D multiplications are possible without calling on bootstrapping to reduce the noise - q will be of size $(4p^2r^5w^5n^{1.5})^{D+1}$, in accordance with bound (8). The success of the attack therefore depends on whether or not these parameters force q to satisfy bound (2). As the next subsection shows, the attack is successful for an extensive range of parameters, and setting the parameters in such a way that the attack fails results in a rather impractical encryption scheme.

5.2 Results

The authors of [1] also carried out experiments to test their attack on actual systems, which is a necessity due to lack of understanding of the nature of the basis reduction algorithms LLL and BKZ. The experimental attacks were carried out on NTRU bases over the ring $R = \mathbb{Z}[x]/(x^n+1)$, for n a power of 2 - meaning the experimental results are transferable to the scheme presented here. We may therefore employ the experimental data given in [1] on the scheme to judge how successful such an attack may be. We set the following values: $p = 2, r = w = 6$, which are the parameter values the authors of [7] suggest.

For example, a successful attack was carried out in 3.5 hours for $n = 2^{11}$ when $\log(q) \geq 165$, which in the given scheme corresponds to $D = 3$, for $q = (4p^2r^5w^5n^{1.5})^{D+1}$. To achieve the same success by running BKZ on the full lattice (that is, not exploiting the possibility of using the sub-field strategy), one would have to run BKZ with block size 27 to achieve $\delta = 1.0141$. For this block-size, BKZ is still considered practical, and the subfield lattice attack might therefore not be too big an improvement in this specific instance [2].

The highest dimension the attack was carried out in was $n = 2^{12}$, with success for $\log(q)$ as low as 190, yet again corresponding to $D = 3$, with the same parameter values as previous. This attack took 120 hours; a direct attack on the full lattice would require running BKZ with block size 131 to achieve $\delta = 1.0081$, an attack that seems unfeasible at this point, as $\beta = 131$ is much too large a block-size to be practical [2].

It follows from these utilizations of the attack that the scheme must be considered insecure if the scheme is also to make meaningful use of the noise reduction strategies presented. Note also that the subfield attacks used LLL to reduce the subfield basis. Therefore it seems reasonable to expect better attacks if BKZ was used on these bases instead, as BKZ consistently outperforms LLL.

6 Conclusions

It has been shown that a subfield lattice attack described in [1] may be applied to an NTRU-based fully homomorphic encryption scheme presented in [7]. It has also been shown that the attack requires the integer parameter q of the encryption scheme to satisfy a lower bound in order to be successful. At the same time, utilization of necessary operations that reduce the level of noise in a ciphertext *also* requires q to satisfy a second lower bound, which is typically much larger than the one required for the attack to be applicable. For this to not be the case and the scheme to be safe from the attack, the parameters of the scheme make it very impractical, and essentially unusable, as it would result in a scheme overly dependent on bootstrapping. Thus, we must conclude that the susceptibility of the described attack is inevitable, for all intents and purposes, if the scheme is to make meaningful use of its noise reducing operations.

Further work may include carrying out the subfield lattice attack with BKZ instead of LLL, as this may improve the attack further. It is also possible that this may affect the security of other schemes based on NTRU.

References

[1] Martin R. Albrecht, Shi Bai, and Léo Ducas. “A Subfield Lattice Attack on Overstretched NTRU Assumptions - Cryptanalysis of Some FHE and Graded Encoding Schemes”. In: *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Confer-*

ence, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I. 2016, pp. 153–178. URL: http://dx.doi.org/10.1007/978-3-662-53018-4_6.

[2] Yuanmi Chen and Phong Q. Nguyen. "BKZ 2.0: Better Lattice Security Estimates". In: *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*. 2011, pp. 1–20. URL: https://doi.org/10.1007/978-3-642-25385-0_1.

[3] Nicolas Gama and Phong Q. Nguyen. "Predicting Lattice Reduction". In: *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*. 2008, pp. 31–51. URL: http://dx.doi.org/10.1007/978-3-540-78967-3_3.

[4] Craig Gentry. "A fully homomorphic encryption scheme". PhD thesis. 2009. ISBN: 9781109444506. URL: https://search.proquest.com/docview/305003863?accountid=8579.

[5] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. "NTRU: A Ring-Based Public Key Cryptosystem". In: *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*. 1998, pp. 267–288. URL: https://doi.org/10.1007/BFb0054868.

[6] Jeffrey Hoffstein et al. *An introduction to mathematical cryptography*. Second edition. Springer, 2014, pp. 373–454. ISBN: 9781493917105.

[7] Kurt Rohloff and David Bruce Cousins. "A Scalable Implementation of Fully Homomorphic Encryption Built on NTRU". In: *Financial Cryptography and Data Security - FC 2014 Workshops, BITCOIN and WAHC 2014, Christ Church, Barbados, March 7, 2014, Revised Selected Papers*. 2014, pp. 221–234. URL: http://dx.doi.org/10.1007/978-3-662-44774-1_18.