A Successful Subfield Lattice Attack on a Fully Homomorphic Encryption Scheme*

Martha Norberg Hovd[1,2]

[1] Simula UiB, Norway
[2] University of Bergen, Norway

Abstract

We present the application of a known subfield lattice attack on a fully homomorphic encryption scheme based on NTRU. We show that the scheme is vulnerable to the attack due to a particular parameter having to satisfy a derived lower bound. We also show that, due to the structure of the scheme, the attack is successful in all practical instantiations of the scheme.

1 Introduction

Fully homomorphic encryption (FHE) schemes are encryption schemes with the following property: for any function f defined over the message space, $\mathrm{Dec}(\mathrm{Eval}(f, c)) = f(\mathrm{Dec}(c))$, where $c = \mathrm{Enc}(m)$ for a message m, and Eval is an evaluation algorithm. The first such scheme was presented by Gentry 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 $\mathrm{Dec}(\mathrm{Enc}(m_1)\mathrm{Enc}(m_2)) = m_1 m_2$ for two (possibly distinct) messages m_1, m_2.

All these starting schemes add bounded randomness to the plaintext to obscure it, and decryption is guaranteed to be correct so long as the randomness stays within the bounds prescribed during set-up, meaning that an encryption of m will actually decrypt to m. This bounded randomness is also referred to as 'noise'. The problem is that as operations are performed on a ciphertext, the noise may grow until it no longer respects the required bounds. At this point, the noise is said to have become unmanageable, as we no longer have any guarantee of correct decryption. These schemes which allow for a limited amount of homomorphic operations to be performed are merely somewhat homomorphic.

In order to have a fully homomorphic scheme the noise in the ciphertexts must be reduced, which is usually achieved through a combination of operations. These operations may stunt the growth of noise, or reduce it slightly, but it is not enough to provide an FHE scheme. To create an FHE scheme, bootstrapping is applied: a homomorphic evaluation of the decryption algorithm. Bootstrapping reduces the noise sufficiently to allow for homomorphic evaluation of any function, but it is a very time-consuming procedure. It is therefore preferable to construct an FHE 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 scheme' is based on a previous scheme, but with different parameter settings, which may result in a less secure scheme. We show one such example in this article, namely that the NTRU-based FHE scheme RC by Rohloff and Cousins [8] is vulnerable to an attack by Albrecht et al. [1]. The RC scheme has different

*This paper appeared at the NISK 2018 conference.

parameter settings compared to the standard NTRU scheme to accommodate for the noise-reducing operations needed to perform homomorphic operations. In particular, this means that the attack by Albrecht et al. does not break the original NTRU encryption scheme.

2 Preliminaries

2.1 Notation

All vectors are row vectors and will be denoted with bold lower case letters: $\mathbf{v}, \mathbf{w}$, whilst 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 $\mathbf{a} = [a_1, a_2, \ldots, a_n]$, whereas sets will be denoted by $\{0, 1, \ldots\}$.

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 arrays of the same length k with elements from a polynomial ring R. We then define the inner product of them 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 coefficient vector $[a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}]$.

The modular reduction $p = r \bmod q$ reduces p modulo q to $r \in (-q/2, q/2]$. We also write $p \equiv r \bmod q$ if we wish to stress that p is equivalent to r modulo q: $p = r + kq$, for $k \in \mathbb{Z}$. The notation generalizes to vectors and polynomials.

The Euclidean norm of an integer 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 an integer 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.

Lemma 2.1. *The following bound holds for any two elements $a, b \in \mathbb{Z}[x]/(x^n + 1)$:*

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

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 the polynomial is reduced with respect to $x^n + 1$, every product $a_i b_j x^{i+j}$ with $i + j \geq n$ is reduced to $-a_i b_j x^{i+j-n}$ in the resulting polynomial ring element. Therefore, every coefficient of $a * b$ is a sum of n terms $a_i b_j$, and so 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 6.14 of Hoffstein et al. [6] shows, any two bases of a lattice are related by an integer matrix with determinant ± 1. It follows from this result that for any two basis matrices $\mathbf{B}, \mathbf{B}'$ we have: $|\det(\mathbf{B})| = |\det(\mathbf{B}')|$. In other words, the determinant of basis matrices is a lattice invariant, defined as the determinant of the lattice.

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, which we use to 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 suffices 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 nonzero 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 6.25 [6]))**.** *Every lattice* $\mathcal{L}$ *of dimension* η *has at least one nonzero vector* $\mathbf{v} \in \mathcal{L}$ *satisfying* $\|\mathbf{v}\| \leq \sqrt{\eta}\det(\mathcal{L})^{1/\eta}$.

Another result by Hermite is that for a given dimension η the Hermite's constant γ_η is the smallest value such that every lattice $\mathcal{L}$ of dimension η contains a nonzero vector $\mathbf{v} \in \mathcal{L}$ satisfying $\|\mathbf{v}\| \leq \sqrt{\gamma_\eta}\det(\mathcal{L})^{1/\eta}$. It follows that $\gamma_\eta \leq \eta$ [6]. Hermite's constant is generally not known. However, we may use the inequality 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 nonzero 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^n, where 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 [7] and its generalisation, BKZ [9], 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 [3].

2.3 An Introduction to NTRU and its Security

The original NTRU encryption scheme 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 [5]. We present the idea of a key recovery attack on NTRU here because it is the basis of a security argument for the RC scheme.

We present enough details of the NTRU-based scheme RC here to discuss a possible key recovery attack on it, and defer a full presentation to Section 4. The scheme follows the general structure of NTRU quite closely, the main difference is that the RC scheme is defined over the polynomial ring $R = \mathbb{Z}[x]/(x^n + 1)$, for $n = 2^k$. The secret key of the scheme is a polynomial $f \leftarrow \chi$, for a distribution χ over R, and f must be invertible modulo q. The public key is defined as $h = f^{-1} * g \bmod q$, for the polynomial $g \leftarrow \chi$.

One way to try to find the secret key f given only the public information q, n and h is to reformulate the problem into one based on 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 as 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. Coef-

ficient $n+1+k$, for $k \in \{0, 1, \dots n-1\}$ is expressed as:

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

where the fact that x^n is equivalent to -1 in $R = \mathbb{Z}[x]/(x^n+1)$ has been applied. Hence, $[f, -u] \cdot \mathbf{B}_{\text{NTRU}} = [f, g]$, which means that $[f, g]$ belongs to $\mathcal{L}_{\text{NTRU}}$, since 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 $\bar{f} * h \equiv \bar{g} \bmod q$ 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 for the lattice $\mathcal{L}_{\text{NTRU}}$. We stress again that the vector $[f, g]$ being among the shortest vectors in the lattice $\mathcal{L}_{\text{NTRU}}$ is a condition for this strategy to work.

3 Subfield Lattice Attack

There may be more efficient attacks than applying LLL or BKZ on the lattice basis, depending on the properties of the scheme. We present one such attack here, by Albrecht et al. [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 $n' \leq n$ is also 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 Section 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 $[x', y']$ has 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 Albrecht et al. [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 $\|[x', y']\| \leq \beta^{\Theta(n'/\beta)}\|\mathbf{v}_0\|$ when applied to the basis of the lattice $\mathcal{L}'_{\text{NTRU}}$, where $\|\mathbf{v}_0\|$ denotes 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 which with high probability is 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 of Albrecht et al. [1], we conclude that for bound (1) to hold, and therefore for the attack to succeed, it suffices that

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

Once the vector $[x', y']$ is 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 finds 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

We now state the RC encryption scheme by Rohloff and Cousins [8]. The scheme is defined over the polynomial ring $R = \mathbb{Z}[x]/(x^n + 1)$, for n a power of 2. The scheme has the integer parameters q, p, chosen such that $q \gg p \geq 2$ and $\gcd(p, q) = 1$. Given these integers, the rings $R_p = \mathbb{Z}_p[x]/(x^n + 1)$ and $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ are defined as the message and ciphertext space, respectively. In addition, the probability distribution χ over R_q is defined, which will typically be some discrete Gaussian distribution. The scheme consists of the following operations:

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

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

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

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

EvalAdd(c_0, c_1, d_0, d_1): Output: $c = c_0 + c_1 \bmod q$, $d = \max(d_0, d_1)$.

EvalMult(c_0, c_1, d_0, d_1): Output: $c = c_0 * c_1 \bmod q$, $d = d_0 + d_1$.

The two latter 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 $f^k * b = m \bmod p$ for any power $k \geq 0$, whilst this is not necessarily the case for f^{-1}, as there is no guarantee that $f^{-1} = 1 \bmod p$. Therefore, the decryption procedure will decrypt any ciphertext of degree at most the given d, assuming $f^d * c = f^k * b \bmod q$, which is why d is set as $\max(d_0, d_1)$ in EvalAdd.

The polynomials f, g, r and e must be chosen so that they ensure correct decryption, so 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 derive bounds the coefficients of these polynomials should satisfy 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 to be met on the coefficients of the polynomials f, g, r and e which ensures correct decryption of a freshly generated ciphertext.

Proposition 4.1. *If every coefficient of the polynomials f, g, r and e is strictly less than $\sqrt{\frac{q}{4pn}}$, any freshly generated ciphertext will be decrypted correctly.*

Proof. The decryption of $c = pr * h + e \bmod q$ 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, every coefficient of this polynomial should have absolute value less than $q/2$, or else the result is $b = pr * g + f * e - q \sum_{i=0}^{n-1} a_i x^i$ where some $a_i \neq 0$ and hence, $b \bmod p$ need not equal m. Therefore, if the inequality $\|pr * g + f * e\|_\infty < q/2$ is satisfied, any freshly generated ciphertext is decrypted correctly. Using the triangle inequality and Lemma 2.1, we may compute:

$$\begin{aligned}\|pr * g + f * e\|_\infty &\leq \|pr * g\|_\infty + \|f * e\|_\infty \\ &\leq pn\|r\|_\infty\|g\|_\infty + n\|f\|_\infty\|e\|_\infty \\ &\leq pn\|r\|_\infty\|g\|_\infty + pn\|f\|_\infty\|e\|_\infty \leq 2pnB^2, \qquad (3)\end{aligned}$$

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 assumption is true if the polynomials r, g, f and e are sampled from a distribution χ such that any coefficient is strictly less than $\sqrt{\frac{q}{4pn}}$. □

4.2 Noise Reductions

The RC scheme 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; we therefore only focus on these two operations in the following.

Note that this, strictly speaking, only makes the scheme presented here leveled 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 Section 4.1, and refer the interested reader to Rohloff and Cousins for details on the bootstrapping procedure [8].

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 $\chi \to \bar{a} \equiv 1 \bmod p$. Given the hint, the actual key switching is the following 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:* $\mathrm{Dec}(f_1, c_1, d) = m$. *If every coefficient of f_1, f_2, g, r, e and $\bar{a}$ is strictly less than $(\frac{q}{2^{d+1}p^d n^{2d}})^{\frac{1}{2d+1}}$, then* $\mathrm{Dec}(f_2, c_2, 1) = m$, *with $a_{1\to 2}$ and c_2 generated according to the above KeySwitch procedure.*

Proof. Decryption of c_2 results in:

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

If the inequality $\|\bar{a} * \bar{b}_1\|_\infty < q/2$ holds, decryption is guaranteed to be correct, i.e., $b_2 = \bar{a} * b_1 = \bar{a} * m = m \bmod p$.

Seeing as c_1 is a ciphertext of degree d, it must be the result of $d-1$ multiplications so, without loss of generality, let $\bar{b}_1 = f_1^d * (pr * g * f_1^{-1} + e)^d \bmod q$. If $\|\bar{a} * \bar{b}_1\|_\infty < q/2$ holds, it is the case that:

$$\begin{aligned}\|\bar{a} * f_1^d * (pr * g * f_1^{-1} + e)^d\|_\infty &= \|\bar{a} * f_1^d * \sum_{i=0}^{d} \binom{d}{i} p^i r^i * g^i * f_1^{-i} * e^{d-i}\|_\infty \\ &= \|\bar{a} * \sum_{i=0}^{d} \binom{d}{i} p^i r^i * g^i * f_1^{d-i} * e^{d-i}\|_\infty\end{aligned}$$

By Lemma 2.1 :

$$\begin{aligned}&\leq n^{2d}\|\bar{a}\|_\infty \sum_{i=0}^{d} \binom{d}{i} p^i \|r\|_\infty^i \|g\|_\infty^i \|f\|_\infty^{d-i} \|e\|_\infty^{d-i} \\ &\leq p^d n^{2d} B^{2d+1} \sum_{i=0}^{d} \binom{d}{i} = 2^d p^d n^{2d} B^{2d+1} < q/2.\end{aligned}$$

Here, B is a bound on the largest coefficient in $\bar{a}, r, g, f$ and e, and the condition of B 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, by essentially dividing the ciphertext by q'. This operation will reduce the underlying noise of the ciphertext by a factor of approximately q'. The operation works by adding Δ, a small multiple of p equivalent to $-c$ modulo q', to the ciphertext c, so $c + \Delta$ is divisible by q'. This should only cause a slight increase in the noise of the ciphertext, and thus ensure that the underlying message is preserved. Seeing as $q'|q$, it follows that $\gcd(q', p) = 1 \Rightarrow \exists v$ s.t. $v = (q')^{-1} \bmod p$. The procedure 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 is easily compensated for by either multiplying with q' in the final step of the decryption procedure or ensuring that $q' \equiv 1 \bmod p$.

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

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]:

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

If the inequality $\|vc - ps\|_\infty < \bar{q}/2$ is satisfied, decryption is correct:

$$f * c' = vf * c - pf * s = v(pr * g(qf' + 1) + f * e) - pf * s \in R$$

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

$$\equiv vpg * r + vf * e - pf * s \bmod \bar{q}$$

If the inequality $\|vf*e+vpg*r-pf*s\|_\infty < \bar{q}/2$, is respected, we will have: $(f*c' \bmod \bar{q}) \bmod p = vm$, and decryption of c' will be correct. Thus, the following expression should be satisfied for correct decryption:

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

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 encryptions of messages m_0, m_1, respectively, 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. The condition for correct decryption is that the polynomials $f_1, f_2, g, r_0, r_1, e_0, e_1$ and $\bar{a}$ are 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 the drawn polynomials should satisfy so the noise added during multiplication and switching keys is sufficiently lowered by switching the modulus. The ciphertext ComposedEvalMult(c_0, c_1) outputs is of 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:

$$\begin{aligned} f_2 * c &= f_2 * \frac{1}{q'}(a_{1\to 2} * c_0 * c_1 + \Delta) \\ &= \frac{1}{q'} f_2 * (\bar{a} * f_2^{-1} * f_1^2 * (pr_0 * g * f_1^{-1} + e_0)(pr_1 * g * f_1^{-1} + e_1) + \Delta) \\ = \ldots &\equiv \frac{1}{q'}(p^2\bar{a} * r_0 * r_1 * g^2 + p\bar{a} * r_0 * g * f_1 * e_1 \\ &\quad + p\bar{a} * r_1 * g * f_1 * e_0 + \bar{a} * f_1^2 * e_0 * e_1 + f_2 * \Delta) = b' \equiv \bar{b} \bmod \bar{q}. \end{aligned}$$

If the inequality $\|b'\|_\infty < \bar{q}/2$ holds, the equality $b' = \bar{b}$ also holds. 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:

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

If $\frac{1}{q'}(4p^2n^4B^5 + nB\frac{pq'}{2})$ is less than $\frac{q}{2q'}$, ComposedEvalMult outputs a ciphertext guaranteed to be decrypted correctly. □

Given this final bound on all the coefficients of the noise-inducing polynomials, we may use it to derive the final bound on q. This bound will depend on other parameters of the scheme and the probability distribution χ, and if the bound is satisfied decryption will be correct.

Suppose any of the polynomials affecting the noise level are drawn from a discrete Gaussian distribution with parameter r, and set w as an assurance measure so that it is highly improbable for any polynomial drawn from this distribution to have an 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 set the condition that

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

should be satisfied for decryption to be correct after a call to ComposedEvalMult.

Assuming that $4p^2n^{1.5}r^5w^5 < q'$ holds, it follows from the condition above that $1 + \frac{1}{2}p\sqrt{n}rw < q/2q'$. Furthermore, $q/q' \geq q_1$ for q_1 the smallest factor of q and thus also the smallest possible ciphertext modulus. In theory, q_1 could be significantly smaller than the other factors of q, as q_1 can be set as the final ciphertext modulus, which would not be subjected to a modulus switching. We would therefore only require q_1 to be large enough to decrypt ciphertexts that have undergone D modulus switchings. A more practical approach however, is to set the following universal bound for any factor of q, as Rohloff and Cousins, and we do:

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

We may therefore conclude that if all factors of q satisfies bound (7), the noise is sufficiently reduced to ensure 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 $\bar{q}$. Hence, q should satisfy the following lower bound, as not doing so might result in an incorrect decryption:

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

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 applicable to the RC scheme, and that the attack will be successful.

We note first that Albrecht et al. state in particular that Heuristic 3.1 holds for the Gaussian distribution [1], which is the distribution suggested for the RC scheme [8]. The attack is therefore applicable to the RC scheme.

However, as emphasized in the final paragraph of Section 2.3, an attack based on solving the 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. This assumption must hold for the attack to produce a vector which can be used as a secret key. The assumption does in all likelihood hold, as the following proposition shows:

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 $\sqrt{\eta}\det(\mathcal{L})^{1/\eta}$. For $\mathcal{L}_{\text{NTRU}}$, we get $\|\mathbf{v}_0\| \leq \sqrt{2n}(q^n)^{1/2n} = \sqrt{2nq}$.

We may 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 RC scheme, and it will produce a vector usable as a secret key with overwhelming probability.

Regarding the success of the attack: recall bound (2) of Section 3.2:

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

satisfaction of which ensures that the attack succeeds. The bound being satisfied is more likely as q grows larger with respect to n, i.e., the more factors q consists of, allowing for more CompEvalMult operations to be performed, the more likely the bound is to be satisfied.

If D modulus switchings are possible, then q will be of size $(4p^2r^5w^5n^{1.5})^{D+1}$, in accordance with bound (8). Allowing for D modulus switchings is desireable as it allows for at most D multiplications to be performed before needing to bootstrap. The success of the attack therefore hinges on whether the parameters also result in q satisfying bound (2). As the next subsection shows, the attack is successful for an extensive range of parameters, as q does satisfy bound (2) more often than not, and that setting the parameters in such a way that the attack fails results in an impractical encryption scheme.

5.2 Results

Albrecht et al. carried out experiments to test their attack on actual systems [1], which is neccessary due to a lack of understanding of the performance of the basis reduction algorithms LLL and BKZ. The experiments were carried out on NTRU bases over the ring $R = \mathbb{Z}[x]/(x^n + 1)$, for n a power of 2, which means that the experimental results are transferable to the RC scheme. We may therefore use the experimental data given by Albrecht et al. [1] to

judge how successful such an attack may be on the RC scheme. We set the following values: $p = 2, r = w = 6$, which are the parameter values Rohloff and Cousins suggest [8].

For example, a successful attack was carried out in 3.5 hours for $n = 2^{11}$ when $\log(q) \geq 165$, which corresponds to $D = 3$, for $q = (4p^2r^5w^5n^{1.5})^{D+1}$ for the RC scheme. To achieve the same success by running BKZ on the full lattice (that is, not exploiting the possibility of using the sub-field strategy), an attacker 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 before. This attack took 120 hours, whereas 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 RC 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

We have shown that the subfield lattice attack described by Albrecht et al. [1] can be applied to the NTRU-based fully homomorphic encryption scheme RC by Rohloff and Cousins [8]. 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 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 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 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.

References

[1] M. R. Albrecht, S. Bai, and L. 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 Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, pages 153–178, 2016.

[2] Y. Chen and P. 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*, pages 1–20, 2011.

[3] N. Gama and P. 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*, pages 31–51, 2008.

[4] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

[5] J. Hoffstein, J. Pipher, and J. 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*, pages 267–288, 1998.

[6] J. Hoffstein, J. Pipher, J. H. Silverman, and J. H. Silverman. *An introduction to mathematical cryptography*. Springer New York, second edition, 2014.

[7] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische annalen*, 261:515–534, 1982.

[8] K. Rohloff and D. B. 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*, pages 221–234, 2014.

[9] C.-P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical computer science*, 53(2-3):201–224, 1987.