An algebraic algorithm for breaking NTRU with multiple keys

Shi Bai[1*], Hansraj Jangir[1*], Tran Ngo[1*] and William Youmans[1*]

[1]Department of Mathematics and Statistics, Florida Atlantic University, Boca Raton, Florida, United States.

*Corresponding author(s). E-mail(s): shih.bai@gmail.com; hjangir2020@fau.edu; ngotbtran@gmail.com; youmansw@fau.edu;

Abstract

We describe a heuristic polynomial-time algorithm for breaking the NTRU problem with multiple keys when given a sufficient number of ring samples. Following the linearization approach of the Arora-Ge algorithm (*ICALP '11*), our algorithm constructs a system of linear equations using the public keys. Our main contribution is a kernel reduction technique that extracts the secret vector from a linear space of rank $\boldsymbol{n}$, where $\boldsymbol{n}$ is the degree of the ring in which NTRU is defined. Compared to the algorithm of Kim-Lee (*Designs, Codes and Cryptography, '23*), our algorithm does not require prior knowledge of the Hamming weight of the secret keys. Our algorithm is based on some plausible heuristics. We demonstrate experiments and show that the algorithm works quite well in practice, with close to cryptographic parameters.

Keywords: Lattice-based cryptography, cryptanalysis, NTRU problem with multiple keys, linearization.

1 Introduction

Lattices have attracted substantial research attention due to their capacity to create efficient cryptographic schemes that are believed to be resistant to quantum adversaries. Fundamental average-case computational problems in lattice-based cryptography include the Short Integer Solution problem (SIS) [3, 30], the Learning With Errors problem (LWE) [36, 37] and the NTRU problem [19, 20].

The NTRU cryptosystem [19, 20], originally proposed by Hoffstein, Pipher and Silverman in 1996, and the corresponding NTRU problem have formed the basis of many cryptosystems in recent years. Let $R = \mathbb{Z}[x]/\langle p(x)\rangle$ be a quotient ring where $p(x)$ has degree n. The NTRU problem states that it is difficult to compute a short vector in the R-module $\{(\mathbf{x}, \mathbf{y}) \in R^2 \mid \mathbf{hx} - \mathbf{y} = 0 \pmod q\}$ given the promise that a short solution $(\mathbf{g}, \mathbf{f})$ exists. Usually, the polynomials $\mathbf{g}, \mathbf{f}$ are the secret keys of the system. There has been much follow-up research on the analysis, design, and implementation of the variant NTRU problems [5, 8–10, 12, 13, 15, 21, 22, 27, 28, 40]. Notably, the assumed hardness of the NTRU problem underlies the security of Falcon [35], a selected algorithm in the NIST post-quantum cryptography standardization process; NTRU [11], a Round 3 finalist; and NTRU Prime [9, 10], an alternate Round 3 candidate. It is therefore evident that NTRU is an attractive foundation for cryptosystems which plays an important role in constructing post-quantum schemes.

1.1 Prior and related work

Following the groundbreaking work of [19, 20], the NTRU assumption has been used extensively in cryptography. The NTRU cryptosystem remained unbroken after more than two decades of cryptanalysis. Lattice reduction and meet-in-the-middle are the two popular methods in evaluating the security of NTRU-based schemes in practice.

Coppersmith and Shamir [14] noticed that recovering a short enough vector in some lattice defined by the public key $\mathbf{h}$ is sufficient to break the NTRU cryptosystem. Asymptotically, this requires a strong lattice reduction such as the Block Korkine-Zolotarev (BKZ) reduction [18, 38] with large blocksize. In practice, parameters have been updated to reflect recent advances in lattice reduction algorithms [24]. Odlyzko described a meet-in-the-middle algorithm in [23] by partially enumerating the candidate polynomials for $\mathbf{f}$ and $\mathbf{g}$. In practice, the best algorithm for solving the NTRU problem is the combination of these two ideas, e.g., the so-called hybrid lattice and meet-in-the-middle approach of Howgrave-Graham [25].

It has been realized that overstretched (e.g., when the modulus q is large) NTRU variants can be much easier to solve, by exploiting the subfield structure [5, 12]. It has been shown that the resulting complexity improvement does not require any algebraic structure [28], but it is due to the existence of a dense sublattice. A recent work of Ducas and Wessel van Woerden [15] shows that the critical point of being "overstretched" is about $q = n^{2.484+o(1)}$. It is noted that these works do not break NTRU encryption in general as q is often chosen to be smaller.

On the provable side, there is evidence that the NTRU problem cannot be too easy to solve. Stehlé and Steinfeld [40] have shown that, when the support of $\mathbf{f}, \mathbf{g}$ are sufficiently large, the distribution of $\mathbf{h} = \mathbf{f}/\mathbf{g} \pmod q$ can be statistically close to the uniform distribution over the invertible elements in the ring. Recently, Pellet-Mary and Stehlé [34] demonstrated an efficient reduction from the worst-case approximate shortest vector problem over ideal lattices to the average-case of some variant NTRU problem.

Various approaches have been explored to extend the NTRU assumption. One line of research focuses on the NTRU problem with multiple keys [1, 2, 31, 33, 39]. More precisely, multiple samples of the form $\mathbf{h}_i = \mathbf{f}_i/\mathbf{g} \pmod q$ are given, with a fixed

denominator polynomial $\mathbf{g}$. The problem asks to recover the secret $\mathbf{g}$ (or $\mathbf{f}_i$). This is referred to as the "NTRU Learning Problem" in the work [33, Definition 4.4.4] and has also been discussed online in [32]. Nitaj [31] has considered a special case where two samples $\mathbf{h}_1, \mathbf{h}_2$ are given where $\|\mathbf{f}_1 - \mathbf{f}_2\|$ is small. In such cases, they showed that the secret vector can be embedded as the shortest vector in some lattice, though it is not clear whether a stronger lattice reduction is needed for actually recovering the secret. Singh and Padhye [39] further generalized this idea and applied it to the NTRU problem with n public keys.

Recently, Kim and Lee [27] described an interesting subfield algorithm for the NTRU problem with multiple keys. They showed that, for ternary secrets, under the assumption that the Hamming weight of the keys $\mathbf{f}_i$ are fixed and known, there exists a polynomial-time algorithm solving the NTRU problem with multiple keys. For the ring $\mathbb{Z}[x]/\langle x^n - 1\rangle$, their algorithm recovers the coefficient vector of $\mathbf{g}\bar{\mathbf{g}}$ in the real subfield and then leverages the Gentry-Szydlo algorithm for extracting the solution $\mathbf{g}$.

1.2 Contribution

We describe a polynomial-time algorithm for solving the multiple-key NTRU problem given sufficiently many samples, without assuming any prior knowledge of the Hamming weight of the secret. Our algorithm leverages the linearization technique of Arora and Ge [7]. Our main contribution is a kernel reduction (or subspace reduction) algorithm that extracts the target secret from a linear space of rank n.

It is known that the LWE problem (with small errors) is prone to an algebraic attack such as the Arora-Ge method [7]. It is folklore that the multiple-key NTRU problem can be rephrased as an LWE-like problem, expressed as $\mathbf{h}_i\mathbf{g} - \mathbf{f}_i = 0 \pmod q$. Thus, it appears plausible to use the Arora-Ge method to break the multiple-key NTRU problem. This observation has already been discussed in [27]. However, there is a known obstacle in using such a method for NTRU [26]. To see the issue, consider the ring $R = \mathbb{Z}[x]/\langle x^n - 1\rangle$ with binary polynomials $\mathbf{f}_i$. Notice that the rotations of the secret polynomials $(\mathbf{f}_i \cdot x^k, \mathbf{g} \cdot x^k)$ also satisfy the public key equation $\mathbf{h}_i(\mathbf{g} \cdot x^k) - \mathbf{f}_i \cdot x^k = 0 \pmod q$ since $\mathbf{f}_i \cdot x^k$ is again binary. Therefore, the kernel of the linearized system will contain (the linearized version of) these vectors. Indeed, it can be shown that the kernel of the linearized version has rank n given sufficiently many samples, hence recovering the actual secret is non-trivial.

We circumvent this issue by using a so-called kernel reduction (or subspace reduction) technique. The main idea is to pin down a particular rotation of the secret vector $\mathbf{g}$ based on its runs of zero coefficients, progressively reducing the dimension of the subspace where the (linearized) secret vector lives and eventually extracting the secret. Together with the Arora-Ge [7] algorithm, this gives a polynomial-time algorithm for solving the NTRU problem with multiple keys. Our algorithm does not require any knowledge of the Hamming weight of $\mathbf{f}_i$ and only requires that $\mathbf{g}$ has at least one zero coefficient, which is almost always satisfied in practical schemes. The Arora-Ge step works better when the support of $\mathbf{f}_i$ is small (e.g., binary or ternary), but a larger support is possible if more samples are given. The algorithm does not require any specific distribution on $\mathbf{f}_i$. Furthermore, the algorithm does not restrict $\mathbf{g}$

to be binary or ternary in general, as long as its support is small. All of these requirements are commonly satisfied in practical systems, as long as the number of samples given is sufficient. The algorithm is also applicable to common rings as discussed in Section 5. As an example, when $\mathbf{f}_i$ is binary (resp. ternary), a number of $O(n)$ (resp. $O(n^2)$) samples $\mathbf{h}_i = \mathbf{f}_i/\mathbf{g} \pmod q$ is sufficient to recover the secret polynomial $\mathbf{g}$, up to multiplication by some x^j, in polynomial time using the Arora-Ge algorithm. The algorithm is provable based on some plausible heuristics.

In Section 6, we demonstrate with concrete experiments that the algorithm almost always works in practice, for close to cryptographic-size parameters.

1.3 Comparison and discussion

Our work focuses on the NTRU problem given multiple samples sharing a common secret denominator $\mathbf{g}$. This is the same problem discussed in [27, 32]. Kim and Lee [27] described an algorithm for solving this problem when the secrets $\mathbf{f}_i$'s are ternary/binary and have known Hamming weight (such information, for example, can be obtained via a side channel). By comparison, we describe a polynomial-time algorithm for solving this problem, without any knowledge of the Hamming weight of the secrets $\mathbf{f}_i$.

In a nutshell, both algorithms use linear algebra to recover some information about the secret. Consider the ring $R = \mathbb{Z}[x]/\langle x^n - 1\rangle$ where n is a prime. The algorithm of Kim and Lee recovers $\mathbf{g}\bar{\mathbf{g}}$ by looking at equations of the form $\mathbf{h}_i\bar{\mathbf{h}}_i\mathbf{g}\bar{\mathbf{g}} = \mathbf{f}_i\bar{\mathbf{f}}_i \pmod q$ where $\bar{\mathbf{g}} := \mathbf{g}(1/x)$. The values of $\mathbf{f}_i\bar{\mathbf{f}}_i$ are known if the Hamming weights of $\mathbf{f}_i$ are given. Then the secret $\mathbf{g}$ can be recovered from $\mathbf{g}\bar{\mathbf{g}}$ by invoking the Gentry-Szydlo algorithm [17]. By comparison, our algorithm follows the linearization approach of Arora-Ge [7]. We propose a kernel reduction method (described in Section 4.4) to recover $\mathbf{g} \cdot x^k$ from a linear space of rank n. Our algorithm has the main advantage that it does not require the Hamming weight of $\mathbf{f}_i$ to be given in prior. Furthermore, our algorithm recovers the secret $\mathbf{g}$ (up to a rotation) directly, thus it does not require the Gentry-Szydlo step whose implementation is non-trivial. By comparison, the algorithm [27] requires the Gentry-Szydlo step to complete, e.g., they write *"we emphasize that the GS algorithm is necessary for solving NTRU with multiple keys"*. Both algorithms run in polynomial time in n and the bit size of the input. The algorithm of Kim and Lee [27] has the advantage of requiring fewer samples than ours when the secrets $\mathbf{f}_i$ are ternary. For example, when the secrets $\mathbf{f}_i$ are ternary and sampled from the ring $R = \mathbb{Z}[x]/\langle x^n - 1\rangle$, our algorithm needs $O(n^2)$ ring samples $\mathbf{h}_i$ while their algorithm only uses $O(n)$ ring samples. When the secrets $\mathbf{f}_i$ are binary, both algorithms need $O(n)$ ring samples.

Furthermore, our algorithm does not rely on any ring structure and hence works for most rings including the original NTRU rings $\mathbb{Z}[x]/\langle x^n \pm 1\rangle$ and NTRU Prime [9, 10] rings $\mathbb{Z}[x]/\langle x^p - x - 1\rangle$. In fact, the obstacle mentioned in Section 1.2 does not always occur in all of these rings, thus the original Arora-Ge [7] method may already work. By comparison, the method of [27] is presented for the ring $\mathbb{Z}[x]/\langle x^n - 1\rangle$ and $\mathbb{Z}[x]/\langle x^p - x - 1\rangle$, but should also work for any ring that admits a suitable conjugate of x.

As our method is based on Arora-Ge [7]'s linearization technique, which aims to solve a polynomial system, it is natural to ask whether a Gröbner basis method works instead. Albrecht, Cid, Faugère and Perret [4] considered such an approach for the

case of LWE and show that the number of required samples could indeed be reduced. We leave this question for future work.

2 Preliminaries

2.1 Notation

We denote by log the base 2 logarithm. For prime $q \geq 2$ we write the integers mod q as $\mathbb{Z}_q$. For $n \geq 1$ we define $[n]$ as the set $\{0, \ldots, n-1\}$.

We represent vectors and matrices with bold lowercase and uppercase letters respectively. A column vector $\mathbf{a}$ of length n is written $(a_1, \ldots, a_n)^T$ and we write $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n]$ for the matrix whose n columns are given by the $\mathbf{a}_i$'s. We use $|$ to represent horizontal concatenation. We write $\text{span}(\mathbf{B})$ for the span of a set of vectors $\mathbf{B}$ and $\ker(\mathbf{A})$ for the right kernel of a matrix $\mathbf{A}$.

Let $R = \mathbb{Z}[x]/\Phi$ for some polynomial Φ of degree n. For $q \in \mathbb{Z}$ we write R_q for $\mathbb{Z}_q[x]/\Phi$ and $R_q^\times$ for the multiplicative subgroup of R_q. An element $\mathbf{f}$ of R will be written as $\mathbf{f} = \sum_{i=0}^{n-1} f_i x^i$. Define $\phi(\mathbf{f}) = (f_0, \ldots, f_{n-1})^T$ to be the coefficient vector of $\mathbf{f}$. If it is clear from the context we will identify $\mathbf{f}$ with its coefficient vector. We say $\mathbf{f}$ has ternary coefficients if all $f_i \in \{-1, 0, 1\}$, and binary coefficients if all $f_i \in \{0, 1\}$. We write $\mathbf{hw}(\mathbf{f})$ for the Hamming weight of $\mathbf{f}$, i.e. the number of non-zero coefficients, and $\text{Constant}(\mathbf{f})$ for the constant term of $\mathbf{f}$.

Given a support set S and a distribution D over S, we denote by $s \leftarrow D$ the process of sampling $s \in S$ from the distribution D. With $s \leftarrow U(S)$ we denote sampling s according to the uniform distribution over S.

For $n \geq 1$ and $r > 0$, we let $V_n(r)$ denote the volume of the n-dimensional ball of radius r. We also let v_n denote the volume of an n-dimensional unit ball where $v_n = \pi^{n/2}/\Gamma(1 + n/2) \approx \left(\frac{2\pi e}{n}\right)^{n/2}/\sqrt{n\pi}$.

2.2 Lattices

A lattice $\mathcal{L}$ is an additive discrete subgroup of $\mathbb{R}^m$. It can be represented as the set of all integer linear combinations of n linearly independent basis vectors $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$. Let $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{m \times n}$ be the matrix whose columns are given by the $\mathbf{b}_i$. The lattice $\mathcal{L}$ generated by $\mathbf{B}$ is defined as $\mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^n\}$ and we call $\mathbf{B}$ a basis for $\mathcal{L}$. We say $\mathcal{L}$ has full rank if $m = n$.

The Euclidean norm of a shortest non-zero vector in $\mathcal{L}$ is denoted by $\lambda_1(\mathcal{L})$ which is called the minimum of the lattice $\mathcal{L}$. The analysis of lattice algorithms often relies on heuristic assumptions such as the so-called Gaussian Heuristic (GH). Let $\mathcal{S}$ be a measurable set in the span of $\mathcal{L}$. The Gaussian Heuristic states that the number of lattice points in $\mathcal{S}$ is $|\mathcal{L} \cap \mathcal{S}| \approx \text{Vol}(\mathcal{S})/\text{Vol}(\mathcal{L})$. When $\mathcal{S}$ is an n-dimensional ball of radius r, the latter quantity is about $(v_n \cdot r^n)/\text{Vol}(\mathcal{L})$. Taking $v_n \cdot r^n \approx \text{Vol}(\mathcal{L})$, we see that $\lambda_1(\mathcal{L})$ is about $\text{GH}(\mathcal{L}) \coloneqq v_n^{-1/n} \cdot \text{Vol}(\mathcal{L})^{1/n} \approx \sqrt{n/(2\pi e)} \cdot \text{Vol}(\mathcal{L})^{1/n}$. In practice we assume that $\text{GH}(\mathcal{L})$ is a decent approximation for $\lambda_1(\mathcal{L})$. Let $\mathbf{B}$ be a basis for $\mathcal{L}$. We define the root Hermite factor of the basis $\mathbf{B}$ as $\delta(\mathbf{B}) = (\|\mathbf{b}_1\|/\text{Vol}(\mathcal{L})^{1/n})^{1/n}$. We say an algorithm admits a root Hermite factor of δ if any input basis can reach the

target root Hermite factor δ after being processed by the algorithm. A larger root Hermite factor is preferred from the cryptanalysis point of view.

2.3 Multiple-key NTRU problem

We review the definition of the NTRU problem and the variant multiple-key NTRU problem.

Definition 1 ($\mathrm{NTRU}_{\Phi,q,B}$ instance)**.** *Let $q \geq 2$ be an integer, $B \leq \sqrt{q}$ be a positive real number, and $R = \mathbb{Z}[x]/\Phi$. An element $\boldsymbol{h} \in R_q$ is called an $\mathrm{NTRU}_{\Phi,q,B}$ instance if there exist $(\boldsymbol{f}, \boldsymbol{g}) \in R_q \times R_q^\times$ such that $\boldsymbol{h} = \boldsymbol{f}/\boldsymbol{g} \pmod q$ and $\|\boldsymbol{f}\|_\infty, \|\boldsymbol{g}\|_\infty \leq B$.*

Definition 2 ($\mathrm{NTRU}_{\Phi,q,B,D}$ search and decision problem)**.** *Let q, B, R be as defined in Definition 1. Let D be a distribution over $\mathrm{NTRU}_{\Phi,q,B}$ instances. The search $\mathrm{NTRU}_{\Phi,q,B,D}$ problem asks, given an $\boldsymbol{h}$ sampled from D, to compute $(\boldsymbol{f}, \boldsymbol{g}) \in R_q \times R_q^\times$ such that $\boldsymbol{h} = \boldsymbol{f}/\boldsymbol{g} \pmod q$ and $\|\boldsymbol{f}\|_\infty, \|\boldsymbol{g}\|_\infty \leq B$. The decisional $\mathrm{dNTRU}_{\Phi,q,B,D}$ problem asks to distinguish between samples from D and from $U(R_q)$.*

Now we define the multiple-key NTRU (or m-NTRU for short) problem which involves several ring samples $\mathbf{h}_i$.

Definition 3 ($\mathrm{m\text{-}NTRU}^E_{\Phi,q,B}$ instance)**.** *Let $q \geq 2$ be an integer, $B \leq \sqrt{q}$ be a positive real number, and $R = \mathbb{Z}[x]/\Phi$. Let E be a finite subset of $\mathbb{Z}$ containing 0 such that $|E| > 1$. Let m be a positive integer. A set of polynomials $\{\boldsymbol{h}_i\}_{1 \leq i \leq m}$ with $\boldsymbol{h}_i \in R_q$ is called a $\mathrm{m\text{-}NTRU}^E_{\Phi,q,B}$ instance if there exist polynomials $\boldsymbol{f}_i \in R_q$ with support E and $\boldsymbol{g} \in R_q^\times$ such that $\boldsymbol{h}_i = \boldsymbol{f}_i/\boldsymbol{g} \pmod q$ and $\|\boldsymbol{g}\|_\infty \leq B$, $\forall i \leq m$.*

Note this problem is defined with a shared polynomial $\mathbf{g}$ over all the m samples.

Definition 4 ($\mathrm{m\text{-}NTRU}^E_{\Phi,q,B,D}$ search and decision problem)**.** *Let q, B, R, E, m be as defined in Definition 3. Let D be a distribution over $\mathrm{m\text{-}NTRU}^E_{\Phi,q,B}$ instances. The search $\mathrm{m\text{-}NTRU}^E_{\Phi,q,B,D}$ problem asks, given a set of polynomials $\{\boldsymbol{h}_i\}_{1 \leq i \leq m}$ sampled from D, to compute polynomials $\boldsymbol{f}_i \in R_q$ with support E and $\boldsymbol{g} \in R_q^\times$ such that $\boldsymbol{h}_i = \boldsymbol{f}_i/\boldsymbol{g} \pmod q$ and $\|\boldsymbol{g}\|_\infty \leq B$. The decisional $\mathrm{m\text{-}dNTRU}^E_{\Phi,q,B,D}$ problem asks to distinguish between samples from D and from $U(R_q^m)$.*

Such variant NTRU problems have been studied before [1, 2, 31, 33, 39]. The complexity of our algorithm is dominated by the size of the support of the polynomials $\mathbf{f}_i$, so its cardinality should be small for a polynomial running-time.

In practical schemes, the coefficients of both $\mathbf{g}$ and $\mathbf{f}$ are usually ternary ($E = \{-1, 0, 1\}$) or binary ($E = \{0, 1\}$). Therefore in this work we will assume the cardinality of E is a small constant. In general, our results place no restrictions on the size B (except for certain rings, which can be seen in Theorem 1) because our algorithm only requires that $\mathbf{g}$ has at least one zero coefficient. We will also assume the $\mathbf{f}_i$ and $\mathbf{g}$ are independently generated. Lastly, we will omit B and D from the notation if they are clear from the context.

3 Algorithms for NTRU with multiple keys

We review known and folklore algorithms for solving the multiple-key NTRU problem.

3.1 Lattice reduction

One standard method to evaluate the security of the NTRU problem is lattice reduction on NTRU lattices [14]. Given a NTRU public key $\mathbf{h}$, one can form the NTRU lattice defined by $\Lambda_q(\mathbf{h}) := \{(\mathbf{x}, \mathbf{y}) \in R^2 \mid \mathbf{hx} - \mathbf{y} = 0 \pmod q\}$. The coefficient vector of $(\mathbf{g}, \mathbf{f})$ is usually a shorter vector compared to the Gaussian heuristic. A basis of $\Lambda_q(\mathbf{h})$ is:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{H} & q\mathbf{I} \end{bmatrix},$$

where $\mathbf{H}$ is the multiplication matrix associated to $\mathbf{h}$, e.g.,

$$\mathbf{H} = [\phi(\mathbf{h}), \phi(\mathbf{h} \cdot x), \ldots, \phi(\mathbf{h} \cdot x^{n-1})].$$

Here, the i-th column of $\mathbf{H}$ is the coefficient vector of $\mathbf{h} \cdot x^{i-1}$.

This can be extended to the multiple-key NTRU problem, where the coefficient vector of $(\mathbf{g}, \mathbf{f}_1, \ldots, \mathbf{f}_m)$ is a short vector in the lattice

$$\Lambda_q(\mathbf{h}_1, \ldots, \mathbf{h}_n) := \begin{bmatrix} \mathbf{I} & & & \mathbf{0} \\ \mathbf{H}_1 & q\mathbf{I} & & \\ \vdots & & \ddots & \\ \mathbf{H}_m & & & q\mathbf{I} \end{bmatrix}.$$

This lattice has rank $(m+1)n$ and determinant q^{mn}. Concrete security can be estimated using standard methods such as [6, 16].

Consider a simple example where $(\mathbf{g}, \mathbf{f}_1, \ldots, \mathbf{f}_m)$ are sampled uniformly with binary coefficients in the ring $\mathbb{Z}[x]/\langle x^n - 1\rangle$. Its expected Euclidean norm is $\sqrt{n(m+1)/2}$. Using the estimate for solving the unique SVP by Gama and Nguyen [16, Section 3.3], we compute the ratio of the Gaussian heuristic estimate with the secret vector length, which is $\gamma \approx q^{m/(m+1)}$.

Note that one can also drop samples by using any $k \leq m$ polynomials instead of m. Thus without loss of generality we can think of m as varying from 1 to its upper bound. Heuristically one can recover the secret as soon as $\gamma \approx \delta^{(m+1)n}$ where δ is the root Hermite factor of the algorithm used. As n is usually large, it is often preferred to have a smaller lattice rank in the lattice reduction (for the root Hermite factor to be large). This means the optimal m is usually small in such lattice attacks (for the multiple-key NTRU problem). To see this, we take $n = 256, q = 769$ and compute the blocksize (of a BKZ-type algorithm) required as a function of $m \leq 256$ – the number of samples given – using the method from [6]. The required blocksizes are plotted in Figure 1. It can be seen that, for such parameters, a smaller number of samples is preferred, e.g., it actually degenerates to the original NTRU case with just one ring sample.

Nitaj [31] has considered a special case where two samples $\mathbf{h}_i = \mathbf{f}_i/\mathbf{g}$ are given and satisfies a norm condition $\|\mathbf{f}_1 - \mathbf{f}_2\| < \min(\|\mathbf{f}_1\|, \|\mathbf{f}_2\|)$. In such cases, they showed that one can form a lattice containing a shortest vector $(\mathbf{g}, \mathbf{f}_1 - \mathbf{f}_2)$. Then by using lattice reduction one can recover the secret. It is likely that a strong lattice reduction such as

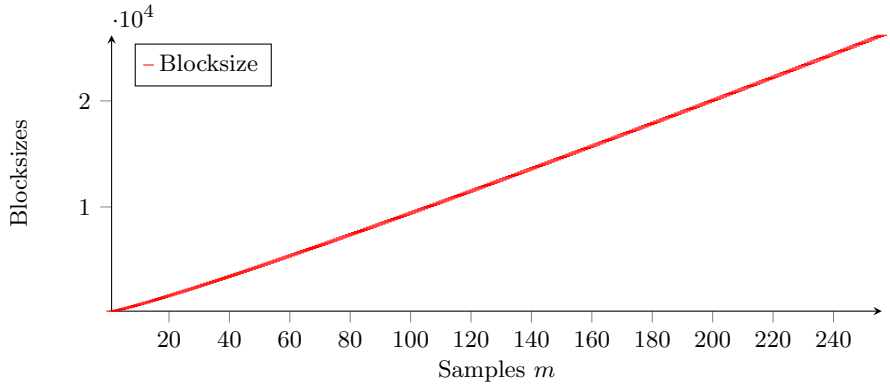

Fig. 1: Blocksize required as a function of number of samples used. Parameters: $n = 256, q = 769$ and binary secrets.

BKZ is needed to recover the secret, similar to the above example. It is also mentioned that the norm condition is only satisfied in rare cases.

3.2 Linearization

It is folklore [27] that the multiple-key NTRU problem can be rephrased as an LWE-like problem and is therefore prone to an algebraic attack such as Arora-Ge [7]. Let $R = \mathbb{Z}[x]/\langle x^n - 1\rangle$ and assume $\mathbf{f}_i \in R$ have ternary support. For any $j \in [n]$ we have

$$f_{i,j}(f_{i,j} - 1)(f_{i,j} + 1) = f_{i,j}^3 - f_{i,j} = 0$$

where $\mathbf{f}_i = \sum_{j=0}^{n-1} f_{i,j} x^j$. As $\mathbf{h}_i\mathbf{g} - \mathbf{f}_i = 0 \pmod q$ we can rewrite this as

$$\prod_{b\in\{-1,0,1\}} (\text{Constant}(x^{-j} \cdot \mathbf{h}_i\mathbf{g}) - b) = 0 \tag{1}$$

for any $j \in [n]$. We use $F_{i,j}(\mathbf{z})$ to represent this equation with $\mathbf{g}$ replaced by some unknown $\mathbf{z} \in R_q$ where we identify $\mathbf{z}$ with its coefficients in $\mathbb{Z}_q^n$. Then $\{F_{i,j}(\mathbf{z})\}_{i,j}$ is a polynomial system with a promised solution $\mathbf{g}$. Notice that $F_{i,j}(\mathbf{z})$ is a cubic multivariate polynomial in terms of the unknown coefficients of $\mathbf{z}$, and using the linearization technique [7] we can write it as a system of linear equations in $O(n^3)$ variables, assigning a variable to each unique monomial. Thus we expect $O(n^2)$ public keys $\mathbf{h}_i$ to determine the solution, as each key generates n equations by rotations x^{-k}.

This idea stems from the Arora-Ge and Gröbner basis methods for binary (or small error) LWE [4, 7]. For example, one can write $\mathbf{h}_i\mathbf{g} - \mathbf{f}_i = 0 \pmod q$ and rephrase this as an LWE instance $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \pmod q$ with $\mathbf{b} = \mathbf{0}$. However, there is a known obstacle [26] in using such a method for the case of NTRU in several popular rings. Note that, for R as defined above, the rotations of the secret polynomials $(\mathbf{f}_i \cdot x^k, \mathbf{g} \cdot x^k)$ also satisfy the public key equation $\mathbf{h}_i(\mathbf{g} \cdot x^k) - \mathbf{f}_i \cdot x^k = 0 \pmod q$ and $\mathbf{f}_i \cdot x^k$ is again ternary. Therefore, all coefficient vectors of the form $\mathbf{g} \cdot x^k$ for all $k \in [n]$ together

with their linear combinations are in the kernel of the linearized system. Given an invertible $\mathbf{g}$, these vectors are linearly independent and hence span a subspace of rank at least n. Computing the kernel of the linearized system will produce a basis which spans this subspace, but not necessarily disclosing the exact $\mathbf{g} \cdot x^k$ (since the basis will contain vectors of the form $\{\sum_{i=0}^{n-1} z_i(\mathbf{g} \cdot x^i)\}$ for $z_i \in \mathbb{Z}$, represented in their linearized vectors). Thus it is not immediately clear whether a shortest vector can be recovered from the kernel efficiently [26].

Remark 1. *It is plausible to use a lattice reduction algorithm on a lattice defined from the kernel basis to recover the secret. The lattice on the kernel of the linearized system has determinant $q^{\tau(n)-n}$ and rank $\tau(n)$, where $\tau(n)$ is defined in Section 4.2. Using the approach from [16], we see that the root Hermite factor required is asymptotically $q^{1/\tau(n)}$. This is quite small as $\tau(n)$ is quadratic or cubic in n for common parameters. In comparison to the standard NTRU lattice, the required root Hermite factor is asymptotically $q^{1/O(n)}$.*

3.3 Kim-Lee algorithm

Kim and Lee [27] described two algorithms for breaking NTRU encryption given multiple keys, which run in polynomial time in n and $\log q$. Their algorithms require the Hamming weight of the keys $\mathbf{f}_i$ to be fixed and known, and needs n samples $\mathbf{h}_i$.

Their first algorithm considers the ring $R = \mathbb{Z}[x]/\langle x^n - 1\rangle$ as in the original NTRU proposal [19, 20] and exploits the existence of a maximal real subfield. We assume n is a prime which is the most common case for such rings. This algorithm starts by considering $\mathbf{h}_i\bar{\mathbf{h}}_i\mathbf{g}\bar{\mathbf{g}} = \mathbf{f}_i\bar{\mathbf{f}}_i \pmod q$. The main observation is that, when $\mathbf{f}_i$'s have ternary coefficients and $\mathbf{hw}(\mathbf{f}_i) = W$ is given, the constant term of $\mathbf{f}_i\bar{\mathbf{f}}_i$ is precisely W. Therefore, one gets one linear equation with respect to the unknown variables in $\mathbf{g}\bar{\mathbf{g}}$ from a single ring sample. Collecting n such ring samples is sufficient to recover $\mathbf{g}\bar{\mathbf{g}}$. Recovering the actual $\mathbf{g} \cdot x^k$ for some k then requires the Gentry-Szydlo algorithm [17] which, while being polynomial time, is non-trivial to implement in practice. This algorithm can also be applied to the case where the secrets $\mathbf{f}_i$ are binary. In such cases, the algorithm still requires n ring samples. This approach is versatile, and could be adapted to other rings that admit a suitable conjugate of x.

Their second algorithm focuses on a variant of the Streamlined NTRUPrime [9, 10] problem in the ring $R = \mathbb{Z}[x]/\langle x^p - x - 1\rangle$, where p is prime. As this ring has no proper subfield the first algorithm does not apply. Interestingly, they observed that they can multiply a row vector $(1, \ldots, 1)$ to both sides of the equation $\mathbf{h}_i\mathbf{g} = \mathbf{f}_i \pmod q$. On the right-hand side, the inner product of the vector $(1, \ldots, 1)$ with the coefficients of $\mathbf{f}_i$ reveals the number of coefficients which are 1, subtracted by the number of coefficients which are -1. For the left-hand side, they observe that the matrix formed by $(1, \ldots, 1) \cdot \phi(\mathbf{h}_i)$ across several i is heuristically non-singular. Thus one can use linear algebra to obtain $\mathbf{g}$ directly when $O(n)$ public keys $\mathbf{h}_i$ are given. This algorithm requires knowing the number of coefficients that are 1 and -1. They also commented that this method cannot be applied to the original NTRU encryption with multiple keys since the matrix obtained (as described above) will be singular.

4 Solving multiple-key NTRU with kernel reduction

In this section we describe how to leverage the Arora-Ge algorithm to solve NTRU with multiple keys in the ring $\mathbb{Z}[x]/\Phi$ for $\Phi \in \{x^n+1, x^n-1\}$ using our kernel reduction algorithm. We discuss application to other rings in Section 5.

We first give our main results in Section 4.1 and consider the linearization procedure in Section 4.2. Then we analyze the kernel of the linearized system in Section 4.3 based on some heuristic assumptions. Finally, we describe our kernel reduction algorithm in Section 4.4 and prove its correctness.

4.1 Main results

Theorem 1 (Under **Heuristics** 1, 2, 3)**.** *Let n and q be positive integers with $q > 3$ a prime. Let $\Phi \in \{x^n + 1, x^n - 1\}$. Let E be a finite subset of $\mathbb{Z}$ containing 0 where $d = |E|$ is constant. Let $\{\boldsymbol{h}_i\}_{i\in[m]}$ be an m-NTRU$^E_{\Phi,q}$ instance with $m = O(n^{d-1})$ such that $\boldsymbol{h}_i = \boldsymbol{f}_i/\boldsymbol{g} \pmod q$ and at least one coefficient of $\boldsymbol{g}$ is zero. For either of the following cases:*

1. *n is prime, or*
2. *n is composite, and the entries of $\boldsymbol{g}$ are sampled uniformly from a constant size support (e.g. $\boldsymbol{g}$ is binary or ternary),*

there is a heuristic algorithm that recovers $\boldsymbol{g} \cdot x^k$ for some $k \in [n]$ in time polynomial in n and the bit length of q with probability $1 - o(1)$.

Observe that when n is a prime, the theorem does not place any restriction on the size and distribution of $\mathbf{g}$. However, when n is a composite, the theorem requires $\mathbf{g}$ to be sampled uniformly from some small support due to some probabilistic argument used in Lemma 6 and Heuristic 3. In most NTRU schemes the secrets have tiny support such as binary and ternary, which is covered by Theorem 1.

4.2 Linearization

We consider the polynomial system given for the ternary case in Equation (1). Let $\{h_i\}_{i\in[m]}$ be a m-NTRU$^E_{\Phi,q}$ instance for $\Phi \in \{x^n + 1, x^n - 1\}$ and define $R = \mathbb{Z}[x]/\Phi$. Let $\mathrm{Coeff}_j(\mathbf{z}) = z_j$ be the function which extracts the coefficient of x^j for some $\mathbf{z} \in R_q$ and $j \in [n]$. We view the coefficients of $\mathbf{z}$ as unknowns $z_0, \ldots, z_{n-1}$, so $\mathrm{Coeff}_j(\mathbf{z})$ returns an element in $\mathbb{Z}_q[z_0, \ldots, z_{n-1}]$. Consider the polynomial system given by

$$F_{i,j}(\mathbf{z}) = \prod_{b\in E} (\mathrm{Coeff}_j(\mathbf{h}_i\mathbf{z}) - b) \tag{2}$$

for $i \in [m]$. $F_{i,j}(\mathbf{z})$ is a multivariate polynomial in $\mathbb{Z}_q[z_0, \ldots, z_{n-1}]$ of degree $d = |E|$. By treating each distinct monomial as a new variable, we can view this as a linear equation in approximately n^d variables in $\mathbb{Z}_q$. Following [7] we refer to this process as "linearization", and we will denote the linearization of $F_{i,j}$ by $L_{i,j}$.

We use $\tau(n, E)$ to denote the number of unique monomials occuring in the polynomials $\{F_{i,j}(\mathbf{z})\}_{i,j}$. This gives the number of variables in the linearized system. Note that as we assume the support E contains 0 and $|E| > 1$, $F_{i,j}(\mathbf{z})$ will only have

monomials of degree in $\{1, \ldots, d\}$. Therefore we have

$$\tau(n, E) \leq \sum_{k=1}^{d} \binom{n+k-1}{k}.$$

For convenience, we will use the shortcut notation $\tau(n)$ in place of $\tau(n, E)$ when the support E is clear in the context. Note that it has order $O(n^d)$ when d is a constant.

4.2.1 Ordering

Due to the nature of our algorithm, it will be important to fix a particular ordering on the monomials of $F_{i,j}(\mathbf{z})$ and thus on the variables occuring in its linearization $L_{i,j}(\mathbf{z})$. As noted above, $F_{i,j}(\mathbf{z})$ contains monomials of degree in $\{1, \ldots, d\}$. Therefore it is always possible to order the monomials in the following way:

1. Order all monomials of degree $d > 1$ according to the lexicographical ordering.
2. Order all monomials of degree $d = 1$ according to the lexicographical ordering.
3. Monomials of degree $d > 1$ have higher orders than monomials of degree $d = 1$.

In the case of binary and ternary supports $E = \{0, 1\}$ and $E = \{-1, 0, 1\}$ this ordering coincides with the standard graded lexicographical ordering. In general it can be different, and can be described as a lexicographical ordering on higher degree terms followed by a lexicographical ordering on degree 1 terms.

Example 1. *Take $n = 2$ and $E = \{0, 1, 2\}$. Let $\{h_i\}_{i\in[m]}$ be an* m-NTRU$^E_{\Phi,q}$ *instance. The polynomials $F_{i,j}(\boldsymbol{z})$ given by Equation (2) have degree 3 and two variables. The monomials of $F_{i,j}(\boldsymbol{z})$ can be ordered as:*

$$z_0^3 > z_0^2 z_1 > z_0^2 > z_0 z_1^2 > z_0 z_1 > z_1^3 > z_1^2 > z_0 > z_1.$$

Define $z_{i_1,i_2,i_3} := z_{i_1} z_{i_2} z_{i_3}$ for $i_1, i_2, i_3 \in [n]$. Then this monomial ordering induces the following ordering on the linearized variables of $L_{i,j}$:

$$z_{0,0,0} > z_{0,0,1} > z_{0,0} > z_{0,1,1} > z_{0,1} > z_{1,1,1} > z_{1,1} > z_0 > z_1.$$

Abusing notation, we will sometimes use $L_{i,j}(\mathbf{z})$ to denote the coefficient vector of the linearization of $F_{i,j}$ under this monomial order. This is interpreted as a row vector.

4.2.2 Linearized vectors

After fixing the order, we set up some notations for easier exposition. Given some $\mathbf{y} = \sum_{i=0}^{n-1} y_i x^i \in R_q$ we define the following column vector, which orders the products of coefficients of $\mathbf{y}$ using the above ordering:

$$\varphi(\mathbf{y}) := (\mathbf{y}^{(0)} \mid \cdots \mid \mathbf{y}^{(n-1)} \mid y_0, \ldots, y_{n-1})^T,$$

where the row vector $\mathbf{y}^{(i)}$ is defined as $\mathbf{y}^{(i)} := (y_i^d, y_i^{d-1} y_{i+1}, \ldots)$, which is lexicographically ordered. Note that the subvector $\mathbf{y}^{(i)}$ collects all the monomials that contain the variable y_i and hence the following subvectors $\mathbf{y}^{(j)}$ for $j \geq i + 1$ do not contain

the variable y_i anymore. The length of the vector $\varphi(\mathbf{y})$ is $\tau(n)$. We also define N_i as the length of the subvector $\mathbf{y}^{(i)}$. Note that in the ternary case we have $N_i = \binom{n-i+1}{2}$, and in the binary case we have $N_i = n - i$.

We will refer to $\varphi(\mathbf{y})$ as the *"linearized vector"* of $\mathbf{y}$. We are mostly interested in applying $\varphi(\cdot)$ on vectors of the form $\mathbf{g} \cdot x^k$.

Let $\mathbf{H}_i$ denote the multiplication matrix associated to $\mathbf{h}_i$. That is, $\mathbf{H}_i = [\phi(\mathbf{h}_i), \phi(\mathbf{h}_i \cdot x), \ldots, \phi(\mathbf{h}_i \cdot x^{n-1})]$. Denote by $[\mathbf{H}'_i \ -\mathbf{H}_i]$ the block matrix whose j-th row corresponds to the coefficients of the linearized polynomials $L_{i,j}(\mathbf{z})$ under our monomial ordering for $i \in [m]$. Now denote by $\mathbf{A}$ the block matrix constructed by concatenating the blocks corresponding to all m ring samples:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_m \end{bmatrix} = \begin{bmatrix} \mathbf{H}'_1 & -\mathbf{H}_1 \\ \mathbf{H}'_2 & -\mathbf{H}_2 \\ \vdots & \vdots \\ \mathbf{H}'_m & -\mathbf{H}_m \end{bmatrix}. \tag{3}$$

Note that $\mathbf{A}$ is formed in a row-wise way, and has mn rows and $\tau(n)$ columns.

4.3 Kernel

We are interested in understanding the right kernel of $\mathbf{A}$ constructed in Equation (3). We know the kernel contains $\varphi(\mathbf{g} \cdot x^k)$ for $k \in [n]$. This is because each $\mathbf{g} \cdot x^k$ is a solution for each $F_{i,j}(\mathbf{z})$, so they must have corresponding linearized vectors in $\ker(\mathbf{A})$. We now wish to know when the $\varphi(\mathbf{g} \cdot x^k)$ generate $\ker(\mathbf{A})$.

4.3.1 Lower bound on the kernel

First, we show that the kernel rank must be at least n.

Lemma 2. *Let $\boldsymbol{g} \in R_q^\times$. The linearized vectors $\{\varphi(\boldsymbol{g} \cdot x^k)\}_{k \in [n]}$ are linearly independent. In other words, the linearized system $\boldsymbol{A}$ as in Equation (3) has a right kernel of rank at least n.*

Proof. Consider the multiplication matrix $\mathbf{G} = [\phi(\mathbf{g}), \phi(\mathbf{g} \cdot x), \ldots, \phi(\mathbf{g} \cdot x^{n-1})]$ associated to $\mathbf{g}$. As $\mathbf{g}$ is invertible so is $\mathbf{G}$, hence $\mathbf{G}$ has full rank. Now we define

$$\mathbf{B} = [\varphi(\mathbf{g}), \varphi(\mathbf{g} \cdot x), \ldots, \varphi(\mathbf{g} \cdot x^{n-1})],$$

whose columns are the linearized vectors of $\mathbf{g} \cdot x^k$. Note $\mathbf{B}$ contains $\mathbf{G}$ as an $n \times n$ submatrix in the last n rows, so $\mathbf{B}$ has rank n as well. It follows that the vectors $\varphi(\mathbf{g} \cdot x^k)$ are all linearly independent, and $\mathbf{B}$ is a basis for a dimension n subspace of $\ker(\mathbf{A})$. □

We will show in Section 4.3.3 that the linearized vectors $\varphi(\mathbf{g} \cdot x^k)$ generate all of $\ker(\mathbf{A})$ with non-negligible probability for a suitably large number of samples m. The proof proceeds analogously to [4, 7]. However, [4, 7] considered the case of LWE where the input samples are genuinely uniform and their linearized vectors are mutually independent by definition of LWE. In our case, the linearized row vectors in $\mathbf{A}_i$ are

formed from a fixed $\mathbf{h}_i$ and thus cannot be independent in theory. To circumvent this we made several heuristic assumptions that we summarize in Section 4.3.2.

4.3.2 Heuristic assumptions

First, we will assume that the coefficients of each $\mathbf{h}_i$ are uniform in $\mathbb{Z}_q^n$. Note that the randomness of the public keys $\mathbf{h}_i$ is induced from the distribution of the private keys $\mathbf{f}_i$ and $\mathbf{g}$.

Heuristic 1. *For any $i \in [m]$, the coefficients of $\boldsymbol{h}_i$ are uniformly distributed in $\mathbb{Z}_q^n$.*

We will use this heuristic to justify that the rows of the multiplication matrix corresponding to $\mathbf{h}_i$ are uniform. In theory, this heuristic cannot be true for practical parameters. For example, the distribution of $\mathbf{h}_i$ cannot be statistically indistinguishable from uniform [40] given a small support on $\mathbf{f}_i$ and $\mathbf{g}$. However, the notion of statistical indistinguishability is perhaps overwhelming from a cryptanalysis point of view.

Furthermore, we will assume that the linearized row vectors within each block matrix $\mathbf{A}_i$ behave like independent vectors.

Heuristic 2. *For any fixed i, the linearized vectors $L_{i,j}$ are mutually independent between distinct j's.*

In practice, these assumptions seem mild. Our experiments support that (see Section 6) using just $m \approx \tau(n)/n$ (close to the theoretical minimum required) number of samples $\mathbf{h}_i$ is almost always sufficient.

4.3.3 Upper bound on the kernel

We will need the following lemma on the zeros of a multivariate polynomial, due to Schwartz and Zippel.

Lemma 3 (Schwartz-Zippel)**.** *Let $P \in K[z_1, \ldots, z_n]$ be a non-zero polynomial of total degree $d \geq 0$ over an integral domain K. Let X be a finite subset of K. Then*

$$\Pr_{\boldsymbol{x} \leftarrow U(X^n)}[P(\boldsymbol{x}) = 0] \leq \frac{d}{|X|}.$$

Theorem 4 (Under heuristics 1, 2)**.** *Let $\{h_i\}_{i \in [m]}$ be an* m-NTRU$^E_{\Phi,q}$ *instance where ϕ, q, E are as defined in Theorem 1. Let $d = |E|$. The resulting linearized system $\boldsymbol{A}$ as in Equation (3) has a right kernel with rank n with probability at least $1 - d^{mn} \cdot q^{\tau(n)-mn}$.*

Proof. By Lemma 2, the matrix $\mathbf{B} = [\varphi(\mathbf{g}), \varphi(\mathbf{g} \cdot x), \ldots, \varphi(\mathbf{g} \cdot x^{n-1})]$ is a basis for a dimension n subspace S of $\ker(\mathbf{A})$. We now lower bound the probability that S is exactly $\ker(\mathbf{A})$. The proof is similar to [4, 7].

Fix some $\mathbf{s} \in \mathbb{Z}_q^{\tau(n)} \setminus S$. We will view $L_{i,j}(s)$ as a polynomial in the coefficients of $\mathbf{h}_i$. In other words, we view it as some polynomial $P_{i,j}(z_1, \ldots, z_n)$ where $P_{i,j}(\phi(\mathbf{h}_i)) = L_{i,j}(\mathbf{s})$. We first fix i and j. By Heuristic 1 we can apply Lemma 3 to find

$$\Pr_{\mathbf{h}_i \leftarrow U(R_q)}[L_{i,j}(\mathbf{s}) = 0] = \Pr_{\phi(\mathbf{h}_i) \leftarrow U(\mathbb{Z}_q^n)}[P_{i,j}(\phi(\mathbf{h}_i)) = 0] \leq \frac{d}{q}.$$

Note that the mutual independence of the $\mathbf{f}_i$ and $\mathbf{g}$ implies the $\mathbf{h}_i$ are mutually independent as well. Then by Heuristic 2 it follows that

$$\Pr_{\mathbf{h}_i \leftarrow U(R_q)}[L_{i,j}(\mathbf{s}) = 0,\ \forall i \in [m],\ \forall j \in [n]] \leq \left(\frac{d}{q}\right)^{mn}.$$

By a union bound, the probability that the linearized system has some solution $\mathbf{s} \in \mathbb{Z}_q^{\tau(n)} \setminus S$ is less than $(d/q)^{mn} q^{\tau(n)} = d^{mn} q^{\tau(n)-mn}$. □

Let d be a constant and $d < q$. The probability in Theorem 4 is asymptotically almost sure when $m = c \cdot \tau(n)/n$ for some small constant $c \approx 1 + 1/\Theta(\log n)$. In practice, the number of samples m can be set to be precisely $\lceil \tau(n)/n \rceil$. Indeed, experimental results of Section 6 show that all instances succeeded with $m = \lceil \tau(n)/n \rceil$, so it seems likely that the slightly larger m predicted is just an artifact of the proof. The following informal argument seems to indicate just this, and has the additional benefit of treating $\mathbf{A}$ as a block matrix, simplifying the proof. However, this requires that we assume the (linearized) blocks $\mathbf{A}_i$ are uniformly random and mutually independent, a stronger assumption than the one used in the above proof.

We use the same notation as in the proof of Theorem 4. First recall that each block $\mathbf{A}_i = [\mathbf{H}_i' \ -\mathbf{H}_i]$ of $\mathbf{A}$ corresponding to a sample $\mathbf{h}_i$ must have rank n. As $S \subseteq \ker(\mathbf{A}) \subseteq \ker(\mathbf{A}_i)$ we can write $\ker(\mathbf{A}_i) = S \oplus K_i$ for some K_i with $S \cap K_i = \{0\}$. Since $\ker(\mathbf{A}_i)$ has dimension $\tau(n) - n$ we have $\dim(K_i) = \tau(n) - 2n$. Let $X = \mathbb{Z}_q^{\tau(n)} \setminus S$. Then

$$\Pr_{\mathbf{v} \leftarrow U(X)}[\mathbf{A}_i \mathbf{v} = 0] = \frac{|K_i \setminus \{0\}|}{|X|} = \frac{q^{\tau(n)-2n} - 1}{q^{\tau(n)} - q^n}.$$

Assuming $\mathbf{A}_i$ are uniformly random and independent, the events $\{\mathbf{A}_1 \mathbf{v} = 0\}, \ldots, \{\mathbf{A}_m \mathbf{v} = 0\}$ are mutually independent as well, so $\Pr[\mathbf{A}\mathbf{v} = 0] = \prod_{i=1}^{m} \Pr[\mathbf{A}_i \mathbf{v} = 0]$. By a union bound, the final probability is bounded above by

$$\frac{\left(q^{\tau(n)-2n} - 1\right)^m}{\left(q^{\tau(n)} - q^n\right)^{m-1}} \leq \frac{\left(q^{\tau(n)-2n} - 1\right)^m}{q^{2n(m-1)} \left(q^{\tau(n)-2n} - 1\right)^{m-1}} \leq q^{\tau(n)-2mn},$$

which is $q^{-\tau(n)}$ when $m = \tau(n)/n$. This indicates that $\lceil \tau(n)/n \rceil$ samples are enough for the procedure to succeed with high probability, which further supports our experimental findings in Section 6.

We also note that Albrecht, Cid, Faugère and Perret have shown a direct proof (see [4, Theorem 8]) of the linear independence of the linearized system for the case of LWE, using the determinant of some Macaulay matrix. Such an argument requires a larger q which is not applicable to our parameters used in experiments.

4.4 Our kernel reduction algorithm

By Theorem 4 we know that, given sufficiently many ring samples, linearization will produce a system $\mathbf{A}$ whose kernel has rank n with non-negligible probability and contains the linearized vectors $\varphi(\mathbf{g} \cdot x^k)$ for $k \in [n]$. As we remarked earlier however, extracting such a solution from the kernel of this linear system seems non-trivial.

We overcome this by iteratively reducing the kernel of $\mathbf{A}$. Recall $\mathbf{g}$ is known to have at least one zero coefficient. Since multiplication by x^k in R corresponds to a rotation of the coefficient vector up to sign, we also know that there must be a rotation that places a zero coefficient at the 0-th term. This translates to a run of zeros at the start of the corresponding linearized vector, and we can reduce the dimension of the kernel by restricting it to only such solutions. However, there is a possible obstruction in the case that n is not prime, hence the separation between prime and composite n in Theorem 1.

4.4.1 Zero patterns

To discuss the possible obstruction to kernel reduction we introduce the following terminology.

Definition 5 (Zero pattern)**.** *Let $\boldsymbol{g} \in R_q$ where $R_q = \mathbb{Z}_q[X]/\Phi$ with $\Phi \in \{x^n+1, x^n-1\}$ and q is a prime. We say that $\boldsymbol{g}$ admits a zero pattern if there exists an integer k with $1 < k < n$ such that $\boldsymbol{g}$ and $\boldsymbol{g} \cdot x^k$ have zero coefficients in exactly the same locations.*

It is clear that when n is a prime $\mathbf{g}$ can not admit a zero pattern unless it is the zero element, since n has no proper divisor. We state this without proof in Lemma 5.

Lemma 5. *When n is a prime, $\boldsymbol{g}$ does not admit a zero pattern.*

However, when n is composite, some $\mathbf{g}$ may admit a zero pattern. We demonstrate some examples below.

Example 2. *Let $R = \mathbb{Z}[x]/\langle x^6 - 1\rangle$. In this ring the polynomial $\boldsymbol{g} = x^5 + x^4 + x^2 - x$ admits a zero pattern since it has the same zero locations as $\boldsymbol{g} \cdot x^3 = x^5 - x^4 + x^2 + x$. Similarly, take $R = \mathbb{Z}[x]/\langle x^6 + 1\rangle$ and $\boldsymbol{g} = x^5 + x^3 + x$. Then $\boldsymbol{g}$ and $\boldsymbol{g} \cdot x^2 = x^5 + x^3 - x$ have the same locations of zero coefficients.*

Fortunately, we observe that the density of such bad $\mathbf{g}$'s is negligible for most common parameters. More precisely, when n is not too small and $\mathbf{g}$ is sampled uniformly from a support of small size then it will admit a zero pattern with low probability.

Lemma 6. *Let n be a composite integer and $R = \mathbb{Z}[x]/\Phi$ for $\Phi \in \{x^n + 1, x^n - 1\}$. Let $\mathcal{G}$ be the size of the support on each coefficient of $\boldsymbol{g}$. Denote G as the set of all $\boldsymbol{g}$'s sampled with such bounded support (e.g., $|G| = \mathcal{G}^n$) and let $G_0 \subseteq G$ contain all those $\boldsymbol{g}$'s admitting a zero pattern. Then $|G_0|/|G| = o(1)$.*

Proof. Let $\phi(\mathbf{g})$ denote the coefficient vector of $\mathbf{g}$. If $\mathbf{g}$ admits a zero pattern then for some non-trivial divisor d of n, $\mathbf{g}$ and $\mathbf{g} \cdot x^k$ have the same coefficients zero. This implies $\phi(\mathbf{g})$ can be partitioned into n/k segments of length k, each segment having l zeros at the same indices, for some $1 \leq l < k$. We will refer to this as a (k, l)-*zero pattern*.

A vector of length k can have l entries zero in $\binom{k}{l}$ unique ways. As all segments of $\phi(\mathbf{g})$ must have the same indices zero, there are a total of $\binom{k}{l}$ such (k, l)-zero patterns. Each (k, l)-zero pattern fixes nl/k coefficients to be zero. As the remaining $n - nl/k$ entries must be non-zero there are at most $\binom{k}{l}(\mathcal{G}-1)^{n-nl/k}$ possible $\mathbf{g}$ with a (k, l)-zero

pattern. Then the number of $\mathbf{g} \in G$ with any zero pattern is at most

$$\sum_{\substack{k|n \\ k\neq 1,n}} \sum_{l=1}^{k-1} \binom{k}{l} (\mathcal{G}-1)^{n-nl/k} = \sum_{\substack{k|n \\ k\neq 1,n}} \left((\mathcal{G}-1)^n \left(\left(1 + \frac{1}{(\mathcal{G}-1)^{\frac{n}{k}}} \right)^k - 1 \right) - 1 \right) \quad (4)$$

$$\leq n^{o(1)} (\mathcal{G}-1)^n \left(\left(1 + \frac{1}{(\mathcal{G}-1)^2} \right)^{n/2} - 1 \right) \quad (5)$$

where Equation (4) follows from the binomial theorem. Equation (5) follows from the observation that the term in the summation is maximized at $k = n/2$ and that the number of divisors of n is in $n^{o(1)}$[**?** , Theorem 315].

Since $\mathcal{G}$ is a constant, we have that $(\mathcal{G}-1)\sqrt{1+1/(\mathcal{G}-1)^2} = \sqrt{\mathcal{G}^2 - 2\mathcal{G} + 2} = \mathcal{G} - \epsilon$ for some positive constant $\epsilon < 1$. We consider the density of such $\mathbf{g}$ over the support of $\mathbf{g}$ which is less than

$$n^{o(1)} \left((\mathcal{G} - \epsilon)/\mathcal{G} \right)^n = o(1).$$

Thus the density of $\mathbf{g} \in G$ admitting a zero pattern is $o(1)$. □

Note that, given an m-NTRU$^E_{\Phi,q,B}$ instance where $\|\mathbf{g}\|_\infty \leq B$ for some constant B, $\mathcal{G}$ can be set to $2B+1$.

For our purposes $\mathbf{g}$ is sampled from (a subset of) $R_q^\times$ rather than R_q, so Lemma 6 does not apply directly. As the proof of Lemma 6 follows from a combinatorial argument which does not account for the ring structure of R_q, it is not immediately clear how to adapt it to $R_q^\times$. For this reason, we make the following additional heuristic assumption that the density result of Lemma 6 carries over to $\mathbf{g} \in G \cap R_q^\times$.

Heuristic 3. *Let R, G, G_0 be as in Lemma 6 and q be a prime. The density $|G_0 \cap R_q^\times|/|G \cap R_q^\times| \approx |G_0|/|G| = o(1)$.*

The high success rates of our experiments in Section 6 indicate that this should be a mild assumption. Also, this assumption can be shown to be true in certain cases. For example, when $R = \mathbb{Z}[x]/\langle x^n - 1 \rangle$ and $\mathbf{g}$ is sampled from $R_q^\times$ with binary support, it cannot have a zero pattern regardless of whether n is prime or composite. This is because it will no longer be invertible, as its associated multiplication matrix $[\phi(\mathbf{g}), \phi(\mathbf{g}\cdot x), \ldots, \phi(\mathbf{g}\cdot x^{n-1})]$ will not be invertible. Moreover, one may also prove this for certain rings where one can lower bound the number of small invertible elements, using methods such as [**? ?**]. We leave such discussions for future work.

In the end, if a given $\mathbf{g}$ admits a zero pattern a potential fix is to re-randomize $\mathbf{g}$ in the hope that it will no longer have a zero pattern. This idea has been used in [**?**] in a different context. For example, one can sample a small $\mathbf{r}$ and write $\mathbf{h}_i(\mathbf{g}+\mathbf{r}) - \mathbf{f}_i = \mathbf{h}_i\mathbf{r} \pmod q$. Instead of finding the kernel of the linearized system, now we look for the pre-image of the linearized vector of $\mathbf{h}_i\mathbf{r}$. It is possible that $\mathbf{g}+\mathbf{r}$ still has at least one zero but no longer has a zero pattern, in which case Algorithm 1 will succeed. One could also divide $\mathbf{h}_i$ by a random invertible polynomial $\mathbf{r}$ and reconstruct our polynomial system which will now have solutions $\mathbf{g}\mathbf{r}\cdot x^k$, which may eliminate the zero pattern. We leave the analysis of such re-randomization for future work.

4.4.2 The algorithm

With the following lemma we will be ready to state the kernel reduction algorithm for all n.

Lemma 7. *Let $V = W_1 \oplus W_2$ be a finite-dimensional vector space. If there exists W such that $W_1 \subseteq W \subseteq V$ and $W \cap W_2 = \{0\}$ then $W = W_1$.*

Proof. By Grassmann's formula $\dim(W)+\dim(W_2) = \dim(W+W_2)+\dim(W\cap W_2) = \dim(W + W_2)$. Observe that $W + W_2 \subseteq V$ and $\dim(V) = \dim(W_1) + \dim(W_2)$. Then $\dim(W_1) \leq \dim(W) \leq \dim(W_1)$ so $W = W_1$. □

Theorem 8 (Under **Heuristic 3**)**.** *Let $\{\boldsymbol{h}_i\}_{i\in[m]}$ be an* m-NTRU$^E_{\Phi,q}$ *instance satisfying the assumptions of Theorem 1. On input the resulting linearized system $\boldsymbol{A}$, Algorithm 1 outputs $\boldsymbol{g}\cdot x^k$ for some $k \in [n]$ with probability $1 - o(1)$, and runs in time polynomial in n and the bit size of q.*

Proof. Theorem 1 assumes that $m = O(n^{d-1})$. Therefore, by Theorem 4, $\ker(\mathbf{A})$ has rank n with probability $1 - o(1)$. We will thus assume $\ker(\mathbf{A})$ has rank n.

Let $\mathbf{B}$ be any basis matrix for $\ker(\mathbf{A})$. By Lemma 2, $\ker(\mathbf{A})$ is generated by $\varphi(\mathbf{g}\cdot x^k)$ for $k \in [n]$. Write $\mathbf{y} = \mathbf{g} \cdot x^k$ for some k. We will view $\mathbf{B}$ as a block matrix where each block corresponds to the subvector $\mathbf{y}^{(i)}$ of $\varphi(\mathbf{y})$:

$$\mathbf{B} = \left[\, \mathbf{B}_0^T, \mathbf{B}_1^T \cdots, \mathbf{B}_n^T \,\right]^T. \tag{6}$$

For any $i \in [n]$, the block $\mathbf{B}_i$ is the submatrix with N_i rows (where N_i is defined in Subsubsection 4.2.2), and $\mathbf{B}_n$ is the final $n \times n$ submatrix. Note that we have $\mathbf{y}^{(i)} \in \mathrm{span}(\mathbf{B}_i)$ for $i \in [n]$.

Recall that we require $\mathbf{g}$ to have at least one coefficient zero. Let $I = \{i_1, \ldots, i_r\} \subseteq [n]$ be the indices such that $g_{i_k} = 0$. Note that $r = n - \mathbf{hw}(\mathbf{g})$. It must hold that $\mathrm{Coeff}_0(\mathbf{g} \cdot x^{-i_k}) = 0$ for each $i_k \in I$. Let $\mathbf{y} = \mathbf{g} \cdot x^{-i_k}$ for some $i_k \in I$. Then the first block $\mathbf{y}^{(0)}$ of the linearized vector $\varphi(\mathbf{y})$ must be 0, i.e.

$$\varphi(\mathbf{y}) = (0, \ldots, 0 \mid \mathbf{y}^{(1)} \mid \cdots \mid \mathbf{y}^{(n-1)}, y_0, \ldots, y_{n-1})^T. \tag{7}$$

Let $\mathbf{X}$ be a basis matrix for $\ker(\mathbf{B}_0)$ and set $\mathbf{B}' = \mathbf{B} \cdot \mathbf{X}$, $S = \mathrm{span}(\mathbf{B}')$. Then Equation (7) is equivalent to the observation that $\varphi(\mathbf{g} \cdot x^{-i_k}) \in S \subseteq \ker(\mathbf{A})$. This holds for all indices $i_k \in I$, so

$$\{\varphi(\mathbf{g} \cdot x^{-i_k}) \mid i_k \in I\} \subseteq S \subseteq \ker(\mathbf{A}).$$

Thus $\dim(S) \geq r$. As $S \cap \{\varphi(\mathbf{g}\cdot x^{-i}) \mid i \notin I\} = \emptyset$, by Lemma 7 we see that $\{\varphi(\mathbf{g}\cdot x^{-i_k}) \mid i_k \in I\}$ must be a basis for S, so $\dim(S) = r$. Thus we have reduced the dimension of the space where we will search for a solution from n to $r = n - \mathbf{hw}(\mathbf{g})$, and will iterate this procedure until the dimension is 1, if possible. In that case $\mathbf{B}'$ is just $\varphi(\mathbf{g} \cdot x^{-i_k})$ so we can recover the coefficients of $\mathbf{g} \cdot x^{-i_k}$ from the last n entries of $\mathbf{B}'$.

Algorithm 1 Extracting a solution $\mathbf{g} \cdot x^k$ by a kernel reduction algorithm.

Require: Linearized system $\mathbf{A}$ constructed from m samples $\mathbf{h}_i = \mathbf{f}_i/\mathbf{g}$, $d = |E|$ where E is the support of $\mathbf{f}_i$.
Ensure: A solution $\mathbf{g} \cdot x^k$ for some $k \in [n]$ or false if a solution can't be found.

```
B ← basis matrix for ker(A)
if rank(B) > n then                                   ▷ Not enough samples.
    return false
else if rank(B) = 0 then                              ▷ No solutions exist.
    return false
else if rank(B) = 1 then                              ▷ There is a unique solution.
    g̃ ← Σ_{i=τ(n)−n}^{τ(n)} b_i x^i where B = (b_1, ..., b_{τ(n)})^T
    return g̃
end if
J ← {0}
X ← basis matrix for ker(B_J)
if rank(X) = 1 then
    B' = B · X
    g̃ ← Σ_{i=τ(n)−n}^{τ(n)} b_i x^i where B' = (b_1, ..., b_{τ(n)})^T
    return g̃
end if
for i ∈ {1, ..., n − 1} do
    J' ← J ∪ {i}
    X' ← basis matrix for ker(B_{J'})
    if rank(X') = 0 then
        continue
    else if rank(X') = 1 then
        B' = B · X'
        g̃ ← Σ_{i=τ(n)−n}^{τ(n)} b_i x^i where B' = (b_1, ..., b_{τ(n)})^T
        return g̃
    else if 1 < rank(X') < rank(X) then
        J ← J'
        X ← X'
    else if rank(X') = rank(X) then                   ▷ There is a zero pattern.
        return false
    end if
end for
```

We now assume $r > 1$. For $J = \{j_1, \ldots, j_s\} \subseteq [n]$ we denote by $\mathbf{B}_J$ the matrix

$$\mathbf{B}_J = \left[\mathbf{B}_{j_1}^T, \mathbf{B}_{j_2}^T, \cdots, \mathbf{B}_{j_s}^T \right]^T. \tag{8}$$

Let $\mathbf{X}$ be a basis matrix for $\ker(\mathbf{B}_{\{0,1\}})$ and $\mathbf{B}' = \mathbf{B} \cdot \mathbf{X}$. By the same argument used above we see that

$$\{\varphi(\mathbf{g} \cdot x^{-i_k}) \mid \{i_k, i_k + 1\} \subseteq I\}$$

is a basis for $S = \text{span}(\mathbf{B}')$. In other words, S is now spanned by the linearized vectors corresponding to $\mathbf{g} \cdot x^{-i_k}$ where $\mathbf{g}$ has two consecutive zero coefficients, at indices i_k and $i_k + 1$. If only one such vector exists then $\dim(S) = 1$ and we are done. Otherwise, three possibilities remain:

- If $\dim(S) = 0$ then $\mathbf{g}$ does not have any two consecutive zero coefficients. In this case, we update $\mathbf{B}_{\{0,1\}}$ to $\mathbf{B}_{\{0,2\}}$ and repeat the procedure. In other words, we now test to see if $\mathbf{g}$ has two zero coefficients separated by one non-zero coefficient.
- If $1 < \dim(S) < r$ then $\mathbf{g}$ does have some pairs of consecutive zero coefficients. In this case we reduced the dimension, and can continue by updating $\mathbf{B}_{\{0,1\}}$ to $\mathbf{B}_{\{0,1,2\}}$.
- If $\dim(S) = r$ then every zero coefficient of $\mathbf{g}$ is followed by another, so $\mathbf{g} = 0$ and we can abort the algorithm early.

We continue in this manner, removing a submatrix $\mathbf{B}_j$ if $\dim(S) = 0$, and adding an additional submatrix $\mathbf{B}_{j+1}$ at every step. If $\dim(S)$ is unchanged at any step then $\mathbf{g}$ has a zero pattern, and the algorithm can be aborted. This procedure is repeated at most $n - 1$ times in the worst case scenario, when $\mathbf{g}$ has $n - 1$ zero coefficients.

Thanks to the ordering described in Section 4.2.1 this procedure converges on a solution $\mathbf{g} \cdot x^k$ that has all of its zero coefficients weighted towards the low order terms. Such an ordering is unique if and only if $\mathbf{g}$ does not have a zero pattern, as discussed in Section 4.4.1. By Lemma 5 if n is prime $\mathbf{g}$ will not have a zero pattern. Otherwise, by Lemma 6 and Heuristic 3, $\mathbf{g}$ has a zero pattern with probability $o(1)$. In either case, the algorithm succeeds with probability $1 - o(1)$.

In the end, the main computation in the algorithm is the kernel computation and matrix multiplication, which take time polynomial in n and the bit size of q.

□

Proof of Theorem 1. This follows directly from Algorithm 1 on input the linearized system described in Equation (3). □

5 Application to NTRU variants

We have so far focused on NTRU with multiple keys over the rings $\mathbb{Z}[x]/\langle x^n \pm 1\rangle$, since the kernel reduction algorithm is generally only required for these rings. In particular, the original Arora-Ge method may already function in particular rings with specific parameters without the need to invoke our kernel reduction procedure. To see this, consider an example where $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ and $\mathbf{f}_i$ has binary support. The original Arora-Ge method may already work in this case since the rotations $\mathbf{f}_i \cdot x^k$ are unlikely to be binary anymore.

In this subsection, we summarize and clarify the applicability of the original Arora-Ge method and our kernel reduction step to several popular NTRU variants. These observations are also verified in experiments in Section 6.

- As discussed above, take $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ and $\mathbf{f}_i$ with *binary* support. The original Arora-Ge method may already work in this case.
- Take $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ and consider $\mathbf{f}_i$ with *ternary* support. The rotations $\mathbf{f}_i \cdot x^k$ are again ternary. Thus the original Arora-Ge method may fail to work and our kernel reduction step is needed.
- Take $R = \mathbb{Z}[x]/\langle x^n - 1\rangle$ and $\mathbf{f}_i$ with *binary (or ternary)* support. The rotations $\mathbf{f}_i \cdot x^k$ are again *binary (resp. ternary)*. Thus the kernel reduction step is needed.
- For the same reason, the original Arora-Ge method is likely to work for the NTRU Prime [9, 10] ring $\mathbb{Z}[x]/\langle x^p - x - 1\rangle$ and the NTTRU [29] ring $\mathbb{Z}[x]/\langle x^n - x^{n/2} + 1\rangle$.

To extend the criterion to general rings, we consider the obstacle discussed in Section 3.2. Assume our m-NTRU instance has support E. Then the non-uniqueness of the solution is due to the existence of $\mathbf{r}$ such that $\mathbf{f}_i \cdot \mathbf{r}$ still has support E for $i \in [m]$, since then $\mathbf{g} \cdot \mathbf{r}$ is also a valid solution: $\mathbf{h}_i(\mathbf{g} \cdot \mathbf{r}) - (\mathbf{f}_i \cdot \mathbf{r}) = 0 \pmod q$. On the other hand, if there does not exist any such $\mathbf{r}$ then the linearized system will have any additional solutions $\varphi(\mathbf{g} \cdot \mathbf{r})$. In such cases, the original Arora-Ge method will already work.

When $R = \mathbb{Z}[x]/\Phi$ for $\Phi \in \{x^n + 1, x^n - 1\}$ and $\mathbf{f}_i$ have ternary coefficients then we of course have the rotations $\mathbf{r} = x^j$ for all j producing valid solutions, but there may be other possibilities for $\mathbf{r}$ depending on the particular set $\{\mathbf{f}_i\}_{i \in [m]}$. Theorem 4, although stated in terms of the linearization, can also be interpreted as saying that such "bad" $\mathbf{r}$ exists with low probability.

Finally, our work considers prime moduli q. The NTRU submission [11] is instantiated over the ring $R_q = \mathbb{Z}_q[x]/\langle x^n - 1\rangle$ and specifies a power of two q and prime n. Our results don't hold for this choice of parameters for a couple of reasons. First, because $\mathbb{Z}_q$ is not an integral domain Lemma 3 does not apply. Second, due to the existence of zero divisors in $\mathbb{Z}_q$ the premise of the kernel reduction method no longer holds. It is noted in [11] that it is possible to use prime q to achieve better size vs. security trade-offs at the cost of being slightly less efficient, and in this context our attack does apply.

6 Implementation and experiments

In this section we report on our implementation and experiments for the algorithms described in Theorem 1 and Algorithm 1. Our algorithm is implemented in `C++` using the `FLINT` library [41] compiled with `OpenMP` support for computing the kernel of a matrix over a finite field. The source code is available at https://github.com/wjyoumans/arora-ge-ntru. These experiments are mostly run on systems with `Intel Xeon E5-2660` and `AMD EPYC-75511` cores.

6.1 $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$ and $\mathbb{Z}_q[x]/\langle x^n - 1\rangle$

In the first set of experiments, we consider two cases where the underlying rings are $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$ and $\mathbb{Z}_q[x]/\langle x^n - 1\rangle$ respectively. In practice the exponent n is usually taken to be a power-of-two or a prime in such rings, but for the purpose of verifying our algorithm we considered more general n.

In the first experiment we focus on ring $R_q = \mathbb{Z}_q[x]/\langle x^n - 1\rangle$ where $\mathbf{f}_i, \mathbf{g}$ are sampled uniformly with binary coefficients such that $\mathbf{g}$ contains at least one zero entry. The

dimension n ranges from 32 to 320 and we fix $q = 769$ in these experiments (thus the ratio between q and n varies). For each dimension n, we generate ≥ 16 instances of the multiple key NTRU problem with different seeds, and for each instance we generate exactly $m = \lceil \tau(n)/n \rceil$ samples $\mathbf{h}_i$. The results are tabulated in Table 1. The first column denotes the ring dimension. The second column "#samples m" denotes the number of samples $\mathbf{h}_i$ used. The third column $\tau(n)$ follows the discussion in Section 4.2. the dimensions of the linearized system. The fourth column reports the rank of the kernel of this system (initial rank before kernel reduction). The fifth column denotes the number of seeds/instances used. The sixth column denotes the number of succeeded experiments, in terms of whether the actual secret $\mathbf{g}$ can be recovered. The last column records the average running-time per instance (in seconds).

There are several observations. First, one can see that the rank of the kernel is always equal to the degree n. This implies that the chosen m is sufficiently large. Note that m is chosen to be precisely $\lceil \tau(n)/n \rceil$ according to the discussion following Theorem 4. Second, all experiments succeeded in recovering the actual secret $\mathbf{g}$, which shows the effectiveness of the kernel reduction algorithm described in Algorithm 1. As the secrets have binary support, one needs $m \approx n/2$ ring samples resulting in a matrix with dimensions $\tau(n) \approx n^2/2$.

Dim n	#samples m	$\tau(n)$	initial kernel rank	#trials	#succ.	ave. time (s)
32	18	560	32	64	64	0.093
48	26	1224	48	64	64	0.533
64	34	2144	64	64	64	1.846
80	42	3320	80	64	64	3.766
96	50	4752	96	64	64	12.27
128	66	8384	128	64	64	38.54
160	82	13040	160	64	64	111.5
192	98	18720	192	64	64	403.9
224	114	25424	224	32	32	617.0
256	130	33152	256	32	32	1785
288	146	41904	288	32	32	2299
320	154	46664	320	16	16	4696

Table 1: Experiments in $R_q = \mathbb{Z}_q[x]/\langle x^n - 1 \rangle$ where the secrets $\mathbf{f}_i, \mathbf{g}$ have binary coefficients and $q = 769$.

In the second experiment, we considered the ring $R_q = \mathbb{Z}_q[x]/\langle x^n + 1 \rangle$ where $\mathbf{f}_i, \mathbf{g}$ are sampled uniformly with ternary coefficients such that $\mathbf{g}$ contains at least one zero entry. The dimension n ranges from 16 to 64 resulting in matrix dimension ranging from 832 to 45824. We also fix $q = 769$. For each dimension n, we generate ≥ 16 instances of the multiple key NTRU problem. The results are given in Table 2. The columns follow similar notations. As the secrets $\mathbf{f}_i$ are now ternary, $\tau(n) \approx n^3/6$ so one needs $m \approx n^2/6$ ring samples. This is why the columns "#samples m" and "$\tau(n)$" are larger than those in Table 1. We stop the experiments at $n = 64$, which correspond to a matrix dimension of 45824.

In the third experiment, we considered both rings $\mathbb{Z}_q[x]/\langle x^n \pm 1 \rangle$ but focus on very small moduli q. This is motivated by the factor of q appearing in the probability

Dim n	#samples m	$\tau(n)$	initial kernel rank	#trials	#succ.	ave. time (s)
16	52	832	16	64	64	0.145
24	110	2624	24	64	64	2.256
32	188	6016	32	64	64	13.26
40	288	11520	40	64	64	64.62
48	419	19648	48	32	32	262.6
56	552	30912	56	16	16	658.4
64	716	45824	64	16	16	3183

Table 2: Experiments in $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$ where the secrets $\mathbf{f}_i, \mathbf{g}$ have ternary coefficients and $q = 769$.

described in Theorem 4. More specifically, we choose q from $13, 19, 29, 31$. We fix $n = 64$ (or 32) for the ring $\mathbb{Z}_q[x]/\langle x^n - 1\rangle$ (resp. $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$), and take $m = 34$ (resp. 188). Note that these m are chosen to be $\lceil \tau(n)/n \rceil$. For each set of parameters, we generate multiple instances as indicated by the column "# trials". In the end, we tabulate the number of succeeded experiments in the last column. One can see that most of the experiments still succeeded even when the moduli are tiny. The only two exceptions are $q = 11$ and $q = 13$ in the ring $\mathbb{Z}_q[x]/\langle x^n - 1\rangle$ where the linearized system did not have sufficient rank (indicating more samples are needed). These moduli are very small compared to what is used in practice.

	Dim n	q	$\tau(n)$	#trials	#succ.
$\mathbb{Z}_q[x]/\langle x^n - 1\rangle$ Binary $\mathbf{f}_i, \mathbf{g}$	64	11	2144	128	126
	64	13	2144	64	60
	64	19	2144	64	63
	64	29	2144	64	64
	64	31	2144	64	64
$\mathbb{Z}_q[x]/\langle x^n + 1\rangle$ Ternary $\mathbf{f}_i, \mathbf{g}$	32	11	6016	128	128
	32	13	6016	64	64
	32	19	6016	64	64
	32	29	6016	64	64
	32	31	6016	64	64

Table 3: Experiments in two rings $\mathbb{Z}_q[x]/\langle x^n \pm 1\rangle$ with various small moduli $q = 13, 19, 29, 31$.

6.2 $\mathbb{Z}_q[x]/\langle x^p - x - 1\rangle$ and $\mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$

We consider some more experiments for the NTRU Prime [9, 10] ring of the form $\mathbb{Z}[x]/\langle x^p - x - 1\rangle$ and the NTTRU [29] ring of the form $\mathbb{Z}[x]/\langle x^n - x^{n/2} + 1\rangle$. As discussed in Section 5, the original Arora-Ge method is likely to already work in such rings and hence the kernel reduction algorithm is not needed. The main purpose of these experiments is to verify the discussions made in Section 5 such that the initial kernel (in these rings) is likely to be already have dimension 1.

In the fourth experiment of Table 4 we consider the ring $R_q = \mathbb{Z}_q[x]/\langle x^p - x - 1\rangle$. In the fifth experiment of Table 5 we consider the ring $R_q = \mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$. In both experiments, $\mathbf{f}_i, \mathbf{g}$ are sampled uniformly with binary coefficients such that $\mathbf{g}$ contains at least one zero entry. We also fix $q = 769$ in both experiments. In Table 4, we choose prime dimensions p ranging from 37 to 131. In Table 5, we choose dimensions n ranging from 32 to 128. For each dimension p, we generate 32 instances of the multiple key NTRU problem with different seeds, and for each instance we generate exactly $m = \lceil \tau(n)/n \rceil$ samples $\mathbf{h}_i$. The results are tabulated in Table 4 and Table 5. The columns have similar notations as the previous experiments.

One can see that in all instances ths linearized systems have an initial kernel of rank 1, which mean the original Arora-Ge method is already able to recover the solution.

Dim p	#samples m	$\tau(n)$	initial kernel rank	#trials	#succ.	ave. time (s)
37	20	740	1	32	32	0.102
53	28	1484	1	32	32	0.631
67	35	2345	1	32	32	1.928
83	43	3569	1	32	32	3.900
97	50	4850	1	32	32	12.87
131	67	8777	1	32	32	40.35

Table 4: Experiments in $R_q = \mathbb{Z}_q[x]/\langle x^p - x - 1\rangle$ where the secrets $\mathbf{f}_i, \mathbf{g}$ have binary coefficients and $q = 769$. There is no need to perform the kernel reduction step.

Dim p	#samples m	$\tau(n)$	initial kernel rank	#trials	#succ.	ave. time (s)
32	18	560	1	32	32	0.103
48	26	1224	1	32	32	0.592
64	34	2144	1	32	32	1.577
80	42	3320	1	32	32	3.815
96	50	4752	1	32	32	12.34
128	66	8384	1	32	32	37.82

Table 5: Experiments in $R_q = \mathbb{Z}_q[x]/\langle x^n - x^{n/2} + 1\rangle$ where the secrets $\mathbf{f}_i, \mathbf{g}$ have binary coefficients and $q = 769$. There is no need to perform the kernel reduction step.

Acknowledgements

The work of SB, HJ and TN is supported in part by the National Science Foundation grant 2044855 & 2122229. The work of WY is generously sponsored by National Security Agency grant H98230-22-1-0328.

The authors would like to thank Changmin Lee for helpful discussions about the kernel reduction algorithm. We thank the anonymous reviewers for their careful reading of our manuscript and insightful suggestions. The authors would also like

to acknowledge the use of the services provided by Research Computing at Florida Atlantic University.

References

[1] Agrawal S (2019) Indistinguishability obfuscation without multilinear maps: New methods for bootstrapping and instantiation. In: Ishai Y, Rijmen V (eds) EUROCRYPT 2019, Part I, LNCS, vol 11476. Springer, Heidelberg, pp 191–225, https://doi.org/10.1007/978-3-030-17653-2_7

[2] Agrawal S, Pellet-Mary A (2020) Indistinguishability obfuscation without maps: Attacks and fixes for noisy linear FE. In: Canteaut A, Ishai Y (eds) EUROCRYPT 2020, Part I, LNCS, vol 12105. Springer, Heidelberg, pp 110–140, https://doi.org/10.1007/978-3-030-45721-1_5

[3] Ajtai M (1996) Generating hard instances of lattice problems (extended abstract). In: 28th ACM STOC. ACM Press, pp 99–108, https://doi.org/10.1145/237814.237838

[4] Albrecht MR, Cid C, Faugère JC, et al (2015) Algebraic algorithms for lwe problems. ACM Commun Comput Algebra 49(2):62. https://doi.org/10.1145/2815111.2815158, URL https://doi.org/10.1145/2815111.2815158

[5] Albrecht MR, Bai S, Ducas L (2016) A subfield lattice attack on overstretched NTRU assumptions - cryptanalysis of some FHE and graded encoding schemes. In: Robshaw M, Katz J (eds) CRYPTO 2016, Part I, LNCS, vol 9814. Springer, Heidelberg, pp 153–178, https://doi.org/10.1007/978-3-662-53018-4_6

[6] Albrecht MR, Göpfert F, Virdia F, et al (2017) Revisiting the expected cost of solving uSVP and applications to LWE. In: Takagi T, Peyrin T (eds) ASIACRYPT 2017, Part I, LNCS, vol 10624. Springer, Heidelberg, pp 297–322, https://doi.org/10.1007/978-3-319-70694-8_11

[7] Arora S, Ge R (2011) New algorithms for learning in presence of errors. In: Aceto L, Henzinger M, Sgall J (eds) ICALP 2011, Part I, LNCS, vol 6755. Springer, Heidelberg, pp 403–415, https://doi.org/10.1007/978-3-642-22006-7_34

[8] Bai S, Beard A, Johnson F, et al (2022) Fiat-shamir signatures based on module-NTRU. In: Nguyen K, Yang G, Guo F, et al (eds) ACISP 22, LNCS, vol 13494. Springer, Heidelberg, pp 289–308, https://doi.org/10.1007/978-3-031-22301-3_15

[9] Bernstein DJ, Chuengsatiansup C, Lange T, et al (2017) NTRU prime: Reducing attack surface at low cost. In: Adams C, Camenisch J (eds) SAC 2017, LNCS, vol 10719. Springer, Heidelberg, pp 235–260, https://doi.org/10.1007/978-3-319-72565-9_12

[10] Bernstein DJ, Brumley BB, Chen MS, et al (2020) NTRU Prime. Tech. rep., National Institute of Standards and Technology, available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions

[11] Chen C, Danba O, Hoffstein J, et al (2020) NTRU. Tech. rep., National Institute of Standards and Technology, available at https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions

[12] Cheon JH, Jeong J, Lee C (2016) An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without a low level encoding of zero. Cryptology ePrint Archive, Report 2016/139, https://eprint.iacr.org/2016/139

[13] Cheon JH, Kim D, Kim T, et al (2019) A new trapdoor over module-NTRU lattice and its application to ID-based encryption. Cryptology ePrint Archive, Report 2019/1468, https://eprint.iacr.org/2019/1468

[14] Coppersmith D, Shamir A (1997) Lattice attacks on NTRU. In: Fumy W (ed) EUROCRYPT'97, LNCS, vol 1233. Springer, Heidelberg, pp 52–61, https://doi.org/10.1007/3-540-69053-0_5

[15] Ducas L, van Woerden WPJ (2021) NTRU fatigue: How stretched is overstretched? In: Tibouchi M, Wang H (eds) ASIACRYPT 2021, Part IV, LNCS, vol 13093. Springer, Heidelberg, pp 3–32, https://doi.org/10.1007/978-3-030-92068-5_1

[16] Gama N, Nguyen PQ (2008) Predicting lattice reduction. In: Smart NP (ed) EUROCRYPT 2008, LNCS, vol 4965. Springer, Heidelberg, pp 31–51, https://doi.org/10.1007/978-3-540-78967-3_3

[17] Gentry C, Szydlo M (2002) Cryptanalysis of the revised NTRU signature scheme. In: Knudsen LR (ed) EUROCRYPT 2002, LNCS, vol 2332. Springer, Heidelberg, pp 299–320, https://doi.org/10.1007/3-540-46035-7_20

[18] Hanrot G, Pujol X, Stehlé D (2011) Analyzing blockwise lattice algorithms using dynamical systems. In: Rogaway P (ed) CRYPTO 2011, LNCS, vol 6841. Springer, Heidelberg, pp 447–464, https://doi.org/10.1007/978-3-642-22792-9_25

[19] Hoffstein J, Pipher J, Silverman JH (1996) NTRU: A new high speed public key cryptosystem. draft from CRYPTO '96 rump session, put online in 2016 at https://web.securityinnovation.com/hubfs/files/ntru-orig.pdf

[20] Hoffstein J, Pipher J, Silverman JH (1998) NTRU: A ring-based public key cryptosystem. In: Buhler JP (ed) Algorithmic Number Theory. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 267–288

[21] Hoffstein J, Pipher J, Silverman JH (2001) NSS: An NTRU lattice-based signature scheme. In: Pfitzmann B (ed) EUROCRYPT 2001, LNCS, vol 2045. Springer, Heidelberg, pp 211–228, https://doi.org/10.1007/3-540-44987-6_14

[22] Hoffstein J, Howgrave-Graham N, Pipher J, et al (2003) NTRUSIGN: Digital signatures using the NTRU lattice. In: Joye M (ed) CT-RSA 2003, LNCS, vol 2612. Springer, Heidelberg, pp 122–140, https://doi.org/10.1007/3-540-36563-X_9

[23] Hoffstein J, Silverman JH, Whyte W (2006) Meet-in-the-middle attack on an ntru private key. Technical report, NTRU Cryptosystems, July 2006. Report #04, available at http://www.ntru.com.

[24] Hoffstein J, Pipher J, Schanck JM, et al (2017) Choosing parameters for NTRUEncrypt. In: Handschuh H (ed) CT-RSA 2017, LNCS, vol 10159. Springer, Heidelberg, pp 3–18, https://doi.org/10.1007/978-3-319-52153-4_1

[25] Howgrave-Graham N (2007) A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In: Menezes A (ed) CRYPTO 2007, LNCS, vol 4622. Springer, Heidelberg, pp 150–169, https://doi.org/10.1007/978-3-540-74143-5_9

[26] Kim J, Lee C (2023) Personal communication.

[27] Kim J, Lee C (2023) A polynomial time algorithm for breaking NTRU encryption with multiple keys. Des Codes Cryptogr 91(8):2779–2789. https://doi.org/10.1007/s10623-023-01233-5, URL https://doi.org/10.1007/s10623-023-01233-5

[28] Kirchner P, Fouque PA (2017) Revisiting lattice attacks on overstretched NTRU parameters. In: Coron JS, Nielsen JB (eds) EUROCRYPT 2017, Part I, LNCS, vol 10210. Springer, Heidelberg, pp 3–26, https://doi.org/10.1007/978-3-319-56620-7_1

[29] Lyubashevsky V, Seiler G (2019) NTTRU: Truly fast NTRU using NTT. IACR TCHES 2019(3):180–201. https://doi.org/10.13154/tches.v2019.i3.180-201, https://tches.iacr.org/index.php/TCHES/article/view/8293

[30] Micciancio D, Regev O (2004) Worst-case to average-case reductions based on Gaussian measures. In: 45th FOCS. IEEE Computer Society Press, pp 372–381, https://doi.org/10.1109/FOCS.2004.72

[31] Nitaj A (2014) Cryptanalysis of NTRU with two public keys. Int J Netw Secur 16(2):112–117. URL http://ijns.jalaxy.com.tw/contents/ijns-v16-n2/ijns-2014-v16-n2-p112-117.pdf

[32] Peikert C (2015) Multiple ntru public keys for the same private key? URL https://crypto.stackexchange.com/questions/30893/multiple-ntru-public-keys-for-the-same-private-key

[33] Peikert C (2016) A decade of lattice cryptography. Found Trends Theor Comput Sci 10(4):283–424. https://doi.org/10.1561/0400000074, URL https://doi.org/10.1561/0400000074

[34] Pellet-Mary A, Stehlé D (2021) On the hardness of the NTRU problem. In: Tibouchi M, Wang H (eds) ASIACRYPT 2021, Part I, LNCS, vol 13090. Springer, Heidelberg, pp 3–35, https://doi.org/10.1007/978-3-030-92062-3_1

[35] Prest T, Fouque PA, Hoffstein J, et al (2022) FALCON. Tech. rep., National Institute of Standards and Technology, available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022

[36] Regev O (2005) On lattices, learning with errors, random linear codes, and cryptography. In: Gabow HN, Fagin R (eds) 37th ACM STOC. ACM Press, pp 84–93, https://doi.org/10.1145/1060590.1060603

[37] Regev O (2006) Lattice-based cryptography (invited talk). In: Dwork C (ed) CRYPTO 2006, LNCS, vol 4117. Springer, Heidelberg, pp 131–141, https://doi.org/10.1007/11818175_8

[38] Schnorr C (1987) A hierarchy of polynomial time lattice basis reduction algorithms. Theoretical Computer Science 53(2-3):201–224

[39] Singh S, Padhye S (2017) Cryptanalysis of ntru with n public keys. In: 2017 ISEA Asia Security and Privacy (ISEASP), pp 1–6, https://doi.org/10.1109/ISEASP.2017.7976980

[40] Stehlé D, Steinfeld R (2011) Making NTRU as secure as worst-case problems over ideal lattices. In: Paterson KG (ed) EUROCRYPT 2011, LNCS, vol 6632. Springer, Heidelberg, pp 27–47, https://doi.org/10.1007/978-3-642-20465-4_4

[41] team TF (2023) FLINT: Fast Library for Number Theory. Version 2.9.0, https://flintlib.org