An Efficient Broadcast Attack against NTRU

Jianwei Li[1]*, Yanbin Pan[2]†, Mingjie Liu[1], Guizhen Zhu[1]
[1]Institute for Advanced Study, Tsinghua University
Beijing 100084, China
{lijianwei10, liu-mj07, zhugz08}@mails.tsinghua.edu.cn
[2]Key Laboratory of Mathematics Mechanization
Academy of Mathematics and Systems Science,Chinese Academy of Sciences
Beijing 100190, China
{panyanbin}@amss.ac.cn

November 24, 2011

Abstract

The NTRU cryptosystem is the most practical scheme known to date. In this paper, we first discuss the ergodic-linearization algorithm against GGH, then naturally deduce a new and uniform broadcast attack against several variants of NTRU: for every recipient's ciphertext, isolate out the blinding value vector, then do derandomization directly and entirety by using inner product, afterwards by using some properties of circular matrix together with linearization we obtain three linear congruence equations of the form $a^T Y = s \bmod q'$ with $N + [\frac{N}{2}]$ variables. Hence only if the number of the independent recipients' ciphertexts/public-keys pairs reaches $N + [\frac{N}{2}] - 2$ can we work out these variables and recover the plaintext in $O(N^3)$ arithmetic operations successfully. The experiment evidence indicates that our algorithm can efficiently broadcast attack against NTRU with the highest security parameters. To the best of our knowledge, this is the most efficient broadcast attack against NTRU. This is an algebraic broadcast attack, which is based on the special structure of the blinding value space $\mathcal{L}_r$.

Key words: Broadcast attack, NTRU, GGH, derandomization, linerization, circular matrix.

1 Introduction

In 1998, Hoffstein, Pipher and Silverman [1] presented a public key cryptosystem based on polynomial algebra called NTRU, denoted by NTRU-1998. The security of NTRU comes from the interaction of the polynomial mixing system with the independence of reduction modulo p and q. The NTRU cryptosystem is the most practical scheme known to date. It features reasonably short, easily created keys, high speed, and low memory requirements. In 2001, Hoffstein and Silverman [2] put forward another instance of NTRU, denoted by NTRU-2001, by employing different parameter sets. In 2005, Howgrave-Graham, Silverman and Whyte [3] gave the third instance of NTRU, denoted by NTRU-2005.

The best attack known against NTRU is based on lattice reduction, but this does not imply that lattice reduction is necessary to break NTRU. Coppersmith and Shamir [4] pointed out that the security of

*Supported by the National Natural Science Foundation of China (Grant No.60910118, and 61133013).
†Supported in part by the National Natural Science Foundation of China (Grant No.11071285, and 60821002) and in part by 973 Project (No.2011CB302401).

NTRU is related, but not equivalent, to the hardness of some lattice problems. Jaulmes and Joux [5] showed that they are able to conduct a chosen-ciphertext attack that recovers the secret key from a few ciphertexts/cleartexts pairs with good probability. This is very dangerous. Most of the ciphertext-only attacks [6, 7, 8] against NTRU rely on the underlying lattice's special cyclic structure.

In 1988, Hästad [9] proposed the first broadcast attack against public key cryptosystems. The scenario of a broadcast attack is as follows. A single message is encrypted by the sender directed for multiple recipients who have different public keys. By observing the ciphertexts only, an attacker can derive the plaintext without requiring any knowledge of any recipient's secret key.

In 2009, Plantard and Susilo [10] first considered the broadcast attack against the lattice-based public-key cryptosystems and also gave some heuristic attacks. However, they showed that NRTU may resist their broadcast attacks, since half of its "message" is random.

Very recently, Pan and Deng [11] give the first broadcast attack against NTRU using the ergodic-linearization algorithm [12, 13, 14]. The main idea of ergodic-linearization technique is used in practical cryptanalysis: use interpolation formula to even out the noise to derive a set of precise nonlinear equations, then introduce new variables for the monomials and obtain a linear system in the new variables. And Pan and Deng [11] pointed out that some other lattice-based cryptosystems, such as [15], can not resist the broadcast attack either.

In this paper, we find that for NTRU the inner product $(\mathbf{r}, \mathbf{r}) = \mathbf{r}^T\mathbf{r}$ is a constant, where $\mathbf{r}$ is a vector consisting of the coefficients of $r(x) \in \mathcal{L}_r$. Hence, we eliminate the blinding value vector $\mathbf{r}$ directly and entirety by doing inner product. Afterwards by using some properties of the circular matrix together with linearization we obtain three linear congruence equations of the form $\mathbf{a}^T\mathbf{Y} = s \bmod q'$ with $N + [\frac{N}{2}]$ variables from every recipient's ciphertext. It can be easily used to give a very efficient broadcast attack against several variants of NTRU: NTRU-1998, NTRU-2001 with an odd d_g, NTRU-2001 with $q = d_r$, NTRU-2005 with $gcd(q, d_g) = 1$ and NTRU-2005 with $q \mid d_r$. Since the number of variables is small, the experiment evidence indicates that one can efficiently broadcast attack against NTRU with the big parameters. Our algorithm is based on the special structure of the blinding value space $\mathcal{L}_r$. It's also a ciphertext-only attacks. Besides, we find that the error vector in the original GGH cryptosystem and the modified error vector in GGH-2009 have the same special structure. Hence, we first discuss the broadcast attacks against the original GGH and GGH-2009 by amending the ergodic-linearization algorithm [12, 13, 14], then naturally deduce the broadcast attack against NTRU.

The remainder of the paper is organized as follows. Section 2 gives some preliminaries. In Section 3, we describe the broadcast attack against NTRU. Section 4 gives a short conclusion.

2 Preliminaries

We denote the integer ring by $\mathbb{Z}$ and denote the residue class ring $\mathbb{Z}/q\mathbb{Z}$ by $\mathbb{Z}_q$. We use bold letters to denote vectors, in column notation. If $\mathbf{v}$ is a vector, then we denote the i-th entry of $\mathbf{v}$ by v_{i-1}.

2.1 GGH

We briefly review the original GGH cryptosystem, for more details see [16]. A GGH cryptosystem comprises of the following algorithms.

Setup: Generate a "good basis" $\mathbf{R} \in \mathbb{Z}^{N\times N}$ and compute a "bad basis" $\mathbf{B} \in \mathbb{Z}^{N\times N}$ of a lattice $\mathcal{L}$, such that $\mathcal{L}(\mathbf{R}) = \mathcal{L}(\mathbf{B})$. Provide $\mathbf{B}$ as public key and keep $\mathbf{R}$ as private basis.

Encryption: To encrypt a message vector $\mathbf{m} \in \mathbb{Z}^N$, use the bad basis to compute

$$\mathbf{c} = \mathbf{Bm} + \mathbf{r}. \tag{2.1}$$

where $\mathbf{r}$ is an error vector uniformly chosen from $\{-\sigma, \sigma\}^N$.
Decryption: Use the good basis to compute

$$\mathbf{m} = \mathbf{B}^{-1}\mathbf{R}\lceil\mathbf{R}^{-1}\mathbf{c}\rfloor.$$

and $\mathbf{r} = \mathbf{c} - \mathbf{Bm}$.

Notice that the original GGH cryptosystem is semantically insecure, because one can check if a ciphertext $\mathbf{c}$ corresponds to a plaintext $\mathbf{m}$ by computing $\mathbf{c} - \mathbf{Bm}$. Furthermore, it's obvious that $r_i^2 = \sigma^2$ for $i = 0, 1, \cdots, N-1$ and $\mathbf{r}^T\mathbf{r} = N\sigma^2$. This fact can be used to discuss the broadcast attack against the original GGH cryptosystem, which differs from the general method in [10]. The original GGH cryptosystem was attacked and broken severely by Nguyen in 1999 [17], and Nguyen pointed out that for safety, one can choose the entries of the error vector $\mathbf{r}$ at random in $[-\sigma, \cdots, \sigma]$ instead of $\{\pm\sigma\}$. Afterwards the other propositions were made using the same principle [18, 19, 20].

In addition, Yanbin Pan et al in [15] presented a new lattice-based public-key cryptosystem mixed with a knapsack and used the module strategy, which is also a GGH-type cryptosystem, denoted by GGH-2009. It has reasonable key size and quick encryption and decryption. Its decryption algorithm is: for any message $\mathbf{m} \in \{0, 1\}^N$, first we uniformly choose a vector $\mathbf{r}$ from $\{0, 1\}^N$, then compute the ciphertext: $\mathbf{c} = \mathbf{Bm} + \mathbf{r} \bmod p$, where p is a prime satisfying certain conditions. For more details see [15]. We find that the modified error vector in GGH-2009 has the same special structure as that in the original GGH cryptosystem.

2.2 NTRU

We give a simple description of the NTRU-1998 cryptosystem, for more details see [1].

The NTRU cryptosystem depends on three integer parameters (N, p, q) and four sets $\mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_r, \mathcal{L}_m$ of polynomials of degree $N-1$ with small integer coefficients, where $\mathcal{L}_f, \mathcal{L}_g$ is called Private Key spaces, $\mathcal{L}_m$ is called Plaintext space, $\mathcal{L}_r$ is called Blinding Value space. In addition, N must be an odd prime, otherwise the lattice attacks can be improved due to non-trivial factors of $X^N - 1$ (see [21]). We choose p, q such that $gcd(p, q) = 1$ and p is much smaller than q. Denote the ring $\mathbb{Z}[x]/(x^N - 1)$ by R and the multiplication in R by $*$ in this paper.
We work over the ring R.
Key Generation:
Step1. Choose $f \in \mathcal{L}_f, g \in \mathcal{L}_g$ such that there exists $F_q, F_p \in R$ satisfying $f * F_q = 1 \bmod q$ and $f * F_p = 1 \bmod p$.
Step2. Let $h = p * F_q * g \bmod q$.
Public Key: h, p, q.
Private Key: f, F_p.
Encryption: To encrypt $m \in \mathcal{L}_m$, we first choose an $r \in \mathcal{L}_r$ randomly, then compute the ciphertext:

$$c = h * r + m \bmod q. \tag{2.2}$$

Decryption: First we compute

$$\begin{aligned} a &= f * c && \bmod q \\ &= pg * r + f * m && \bmod q \end{aligned}$$

then we choose the coefficients of a in the interval from $-\frac{q}{2}$ to $\frac{q}{2}$. By the fact that all the coefficients of $pg * r + f * m$ may be in the interval from $-\frac{q}{2}$ to $\frac{q}{2}$, we almost get

$$a = pg * r + f * m.$$

Then we recover the message m by computing $m = F_p * a \bmod p$.

Since there exists several variants of NTRU, this has made the analysis of NTRU a tricky task, as in [22]. However, in this paper, we give a uniform broadcast attack against NTRU. Mol and Yung in [22] summarized the main instantiations of NTRU in the table below:

Variant	q	p	$\mathcal{L}_f$	$\mathcal{L}_g$	$\mathcal{L}_m$	$\mathcal{L}_r$	F	Ref
NTRU-1998	$2^k \in [\frac{N}{2}, N]$	3	$L(d_f, d_f - 1)$	$L(d_g, d_g)$	L_m	$L(d_r, d_r)$	-	[1]
NTRU-2001	$2^k \in [\frac{N}{2}, N]$	$2 + x$	$1 + p * F$	$\mathcal{B}(d_g)$	$\mathcal{B}$	$\mathcal{B}(d_r)$	$\mathcal{B}(d_F)$	[2]
NTRU-2005	prime	2	$1 + p * F$	$\mathcal{B}(d_g)$	$\mathcal{B}$	$\mathcal{B}(d_r)$	$\mathcal{B}(d_F)$	[3]

where

- $L_m = \{m \in R$: m has coefficients lying between $-\frac{1}{2}(p-1)$ and $\frac{1}{2}(p-1)\}$,
- $L(d_1, d_2) = \{F \in R : F$ has d_1 coefficients equal 1, d_2 coefficients equal -1, the rest $0\}$,
- $\mathcal{B}$ denotes the set of all polynomials with binary coefficients,
- $\mathcal{B}(d) = \{F \in R : F$ has d coefficients equal 1, the rest $0\}$.

Remark 1: Let us focus attention on the blinding value space $\mathcal{L}_r$. We find that the inner product

$$\begin{aligned} \mathbf{r}^T\mathbf{r} &= r_0^2 + r_1^2 + \cdots + r_{N-1}^2 \\ &= \begin{cases} 2d_r & for\ NTRU - 1998; \\ d_r & for\ NTRU - 2001\ and\ NTRU - 2005. \end{cases} \end{aligned}$$

is a constant, where $\mathbf{r} = (r_0, r_1, \cdots, r_{N-1})^T$ is a vector corresponding to $r(x) \in \mathcal{L}_r$. Note that increasing the number of recipient's ciphertext can't change plaintext vector $\mathbf{m}$, but increases the number of blinding value vector $\mathbf{r}$ respectively. It's is a heavy curse of recovering the plaintext vector $\mathbf{m}$. Hence, we should eliminate the blinding value vector or error vector $\mathbf{r}$.

2.3 The Linear Form of NTRU and Circular Matrix

In NTRU, for a polynomial $f \in R$, we can represent f as

$$f = \sum_{i=0}^{N-1} f_i x^i.$$

It is equivalent to

$$\mathbf{f} = (f_0, f_1, \cdots, f_{N-1})^T.$$

It's easy to verify the corresponding vector of $f * g$ in R is

$$\begin{pmatrix} f_0 & f_{N-1} & \cdots & f_1 \\ f_1 & f_0 & \cdots & f_2 \\ \vdots & \vdots & \ddots & \vdots \\ f_{N-1} & f_{N-2} & \cdots & f_0 \end{pmatrix} \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{N-1} \end{pmatrix}$$

In particular, even if f_i or g_j are functions about x, the formula above also holds.

Thus, we have the equivalent linear form of the formula (2.2)

$$\mathbf{c} = \mathbf{H}\mathbf{r} + \mathbf{m} \bmod q. \tag{2.3}$$

where

$$\mathbf{H} = \begin{pmatrix} h_0 & h_{N-1} & \ldots & h_1 \\ h_1 & h_0 & \ldots & h_2 \\ \vdots & \vdots & \ddots & \vdots \\ h_{N-1} & h_{N-2} & \ldots & h_0 \end{pmatrix}$$

is a circular matrix in $\mathbb{Z}_q^{N\times N}$. Obviously, $\mathbf{H}^T$ is also a circular matrix over $\mathbb{Z}_q^{N\times N}$.

As needed, we only consider the circular matrix over $\mathbb{Z}_q^{N\times N}$. For simplicity, we give the following convention: $h_{N+i} = h_i, \mathbf{H}_{N+i,N+j} = \mathbf{H}_{i,j}$. Clearly, if $\mathbf{H} \in \mathbb{Z}_q^{N\times N}$ is a circular matrix if and only if $\mathbf{H}_{i,j} = \mathbf{H}_{N-j+i+1,1}$, for $i, j \in \{1, 2, \cdots, N\}$. And there are the following fundamental lammas (see [23]).

Lemma 2.1 *If $\mathbf{H} \in \mathbb{Z}_q^{N\times N}$ is an invertible circular matrix over $\mathbb{Z}_q^{N\times N}$, then $\mathbf{H}^{-1}$ is also a circular matrix over $\mathbb{Z}_q^{N\times N}$.*

Lemma 2.2 *If $\mathbf{G}, \mathbf{H} \in \mathbb{Z}_q^{N\times N}$ are circular matrixs, then $\mathbf{GH}$ is also a circular matrix. In particular, $\mathbf{H}^T\mathbf{H}$ is a symmetric circular matrix.*

We show how to get $\mathbf{H}^{-1}$ and $\mathbf{H}^T\mathbf{H}$ in $O(N^2)$ arithmetic operations respectively in Appendix A. Of course, if N is taken to be large, then it might be faster to use Fast Fourier Transforms to compute products $\mathbf{H}^T\mathbf{H}$ in $O(N \log N)$ operations. However, it doesn't impact the final complexity. What's more, it's easy to prove the following theorem.

Theorem 2.3 *If $\mathbf{G}, \mathbf{H} \in \mathbb{Z}_q^{N\times N}$ are circular matrixes, which are corresponding to $g = \sum_{i=0}^{N-1} g_i x^i$ and $h = \sum_{i=0}^{N-1} h_i x^i$ respectively, then $\mathbf{GH} = \mathbf{I} \bmod q$ if and only if $g * h = 1$ over $\mathbb{Z}_q[x]/(x^N - 1)$, where $\mathbf{I}$ is an identity matrix of order N.*

2.4 The Proportion of the Matrices of Rank n in $\mathbb{Z}_q^{(n+l)\times n}$

Here, we discuss a general problem: how big is l, then the linear system $\mathbf{L}\times\mathbf{Y} = \mathbf{S} \bmod q$ only has a single solution with very high probability? Of course, the vector $\mathbf{S} \in \mathbb{Z}_q^{n+l}$, the (random) matrix $\mathbf{L} \in \mathbb{Z}_q^{(n+l)\times n}$ and the modulus q are known and we know that there is at least one solution. Clearly, that there is a single solution is equivalent to that the rank of $\mathbf{L}$ equals to n. Now, we distinguish two cases:

• For the case $l = 0$, if the matrix $\mathbf{L}$ is invertible modulo q, then there is only one solution.

Nguyen in [17] gave the following result estimating the proportion of invertible matrices modulo q among all matrices:

Theorem 2.4 *Let q be a power of prime p. Consider the ring of $n \times n$ matrices with entries in $\mathbb{Z}_q$. Then the proportion of invertible matrices (i.e., with determinant coprime to q) is equal to:*

$$\prod_{k=1}^{n}(1 - p^{-k}).$$

• For the case $l = 1$ *or* 2, it's easy to obtain the following generalization of Theorem 3 in [17]:

Theorem 2.5 *Let $\mathbb{F}_q$ be the finite field with q elements, where q is a prime power. The proportion of matrices of rank n in the set of $(n + l) \times n$ matrices with entries in $\mathbb{F}_q$ is equal to:*

$$\prod_{k=l+1}^{n+l}(1 - q^{-k}), \quad l = 1, 2.$$

For the sake of completeness, detailed proof is provided in Appendix B. Note that the above proportions converge quickly to their limit. The result of numerical experiment shows that the proportion can be considered as constant for high dimensions (higher than 50). Table 1 gives numerical results for the case $l = 0,\ 1,$ *and* 2.

Table 1. The proportion of the matrices of rank n in $\mathbb{Z}_q^{(n+l)\times n}$

q	3	59	64	128	197	251	256	367	587
$l=0$	0.560	0.983	0.289	0.289	0.995	0.996	0.289	0.997	0.998
$l=1$	0.840	0.9997	0.9997	0.9999	0.99997	0.99998	0.99999	0.999993	0.999997
$l=2$	0.945	0.999995	0.999996	1.000	1.000	1.000	1.000	1.000	1.000

It shows that if $l = 0$, the random matrix $\mathbf{L}$ is invertible mod$q$ with non-negligible probability, and with very high probability for $p > 59$. And we see that for the case $l = 1$, there is a single solution with very high probability.

3 The Broadcast Attacks against NTRU

3.1 Analyse the ergodic-linearization algorithm against GGH

We first analyse the ergodic-linearization algorithm [12, 13, 14] against the original GGH, which naturally deduce the broadcast attacks against NTRU.

For the formula (2.1), we do the ergodic on the error set to get

$$(\sum_{j=0}^{N-1} \mathbf{B}_{i+1,j+1} m_j - c_i - \sigma)(\sum_{j=0}^{N-1} \mathbf{B}_{i+1,j+1} m_j - c_i + \sigma) = 0, i = 0, 1, \cdots, N-1.$$

It's equivalent to do square:

$$(\sum_{j=0}^{N-1} \mathbf{B}_{i+1,j+1} m_j - c_i)^2 = \sigma^2, i = 0, 1, \cdots, N-1. \tag{3.1}$$

Then we assign $m_0^2, m_0 m_1, \cdots, m_0 m_{N-1}, m_1^2, m_1 m_2, \cdots, m_1 m_{N-1}, \cdots, m_{N-1}^2, m_0, m_1, \cdots, m_{N-1}$ new variables $\{y_i\}_1^{\frac{N^2+3N}{2}}$. This linearization will produce N linear equations from every recipient's ciphertext in the form of

$$\mathbf{a_i}^T \mathbf{Y} = \sigma^2 - c_i^2,\ i = 0, 1, \cdots, N-1,$$

where $\mathbf{Y} = (y_1, y_2, \cdots, y_{\frac{N^2+3N}{2}})^T$. What's more, $\mathbf{a_0}, \mathbf{a_1}, \cdots, \mathbf{a_{N-1}}$ are linearly independent, unless there exists $c_i = 0$ to make them linearly dependent possibly. Hence, we need at least $\lceil \frac{N+3}{2} \rceil$ recipients' ciphertexts and the corresponding public keys to obtain a system of linear congruence equations $\mathbf{L} \times \mathbf{Y} = \mathbf{S}$, where $\mathbf{L}$ is a $\frac{N^2+3N}{2} \times \frac{N^2+3N}{2}$ matrix, and $\mathbf{S}$ is a constant vector. And we can find $\mathbf{m}$ by solving the above set of linear equations over $\mathbb{Z}$.

Another way is to take the sum of the equation (3.1) for $i = 0, 1, \cdots, N-1$, which is equivalent to do the inner product

$$(\mathbf{c} - \mathbf{Bm})^T (\mathbf{c} - \mathbf{Bm}) = \mathbf{r}^T \mathbf{r}.$$

Note that $\mathbf{r}^T \mathbf{r} = N\sigma^2$, we get

$$\mathbf{m}^T \mathbf{B}^T \mathbf{Bm} - 2\mathbf{c}^T \mathbf{Bm} = N\sigma^2 - \mathbf{c}^T \mathbf{c}. \tag{3.2}$$

Then we treat (3.2) in the same way as above. This linearization will produce one linear equation from every recipient's ciphertext in the form of $\mathbf{a}^T \mathbf{Y} = N\sigma^2 - \mathbf{c}^T \mathbf{c}$. Hence, we need at least $\frac{N^2+3N}{2}$ recipients' ciphertexts/public-keys pairs to obtain a system of linear congruence equations $\mathbf{L} \times \mathbf{Y} = \mathbf{S}$.

Obviously, for another encoding method $\mathbf{Br} + \mathbf{m} = \mathbf{c}$, first we get

$$det(\mathcal{L}(\mathbf{R}))\mathbf{r} = \widetilde{\mathbf{B}}\mathbf{c} - \widetilde{\mathbf{B}}\mathbf{m},$$

then we can do something similar.

For GGH-2009, we modify the decryption equation to get

$$2\mathbf{c} - \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = 2\mathbf{B}\mathbf{m} + 2\mathbf{r} - \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \bmod p.$$

Let $\hat{\mathbf{c}} = 2\mathbf{c} - (1, 1, \cdots, 1)^T$, $\hat{\mathbf{B}} = 2\mathbf{B}$ and $\hat{\mathbf{r}} = 2\mathbf{r} - (1, 1, \cdots, 1)^T$, then we have

$$\hat{\mathbf{c}} = \hat{\mathbf{B}}\mathbf{m} + \hat{\mathbf{r}} \bmod p, \tag{3.3}$$

where $\hat{\mathbf{r}} \in \{-1, 1\}^N$. Then we can treat (3.3) in the same way as that for the original GGH.

Remark 2: Obviously, doing the inner product is worse than doing square, because the public key $\mathbf{B}$ has no good global property as the matrix $\hat{\mathbf{H}}$ in NTRU, which is an invertible circular matrix. Nonetheless, this method naturally deduces the broadcast attack against NTRU.

3.2 How to do the Broadcast Attacks against NTRU

According to Subsection 2.3, it's equivalent to consider the linear form of NTRU over $\mathbb{Z}_q$. If $\mathbf{H}$ is invertible in $\mathbb{Z}_q^{N\times N}$, obviously we can easily get the equation below from (2.3)

$$\mathbf{H}^{-1}\mathbf{m} + \mathbf{r} = \mathbf{H}^{-1}\mathbf{c} \bmod q.$$

Let $\hat{\mathbf{H}} = \mathbf{H}^{-1}, \mathbf{b} = \mathbf{H}^{-1}\mathbf{c} \bmod q$, then we have

$$\hat{\mathbf{H}}\mathbf{m} + \mathbf{r} = \mathbf{b} \bmod q. \tag{3.4}$$

Usually, $\mathbf{H}$ is invertible in $\mathbb{Z}_q^{N\times N}$ with high probability in NTRU-2001 with an odd d_g and NTRU-2005 with $gcd(q, d_g) = 1$ (the proportion of invertible elements is close to 1, which can be computed as in [24]). Hence, we can easily choose $\mathbf{H}$ which is invertible, then get an invertible circular matrix $\hat{\mathbf{H}}$ and it requires $O(N^2)$ arithmetic operations from Lemma 2.1. Another way to estimate whether $\mathbf{H}$ is invertible in $\mathbb{Z}_q^{N\times N}$ or not is to observe whether $gcd(det(\mathbf{L}), q) = 1$ or not.

However, for NTRU-2001 with an even d_g and NTRU-2005 with $q \mid d_g$, $\mathbf{H}$ is not invertible. We need some extra restrictions: $q \mid d_r$, to get an invertible $\mathbf{H}$. In addition, $\mathbf{H}$ is not invertible in NTRU-1998. Luckily, the public key h is "pseudo-invertible" mod$q$ with overwhelming probability. More precisely, there is the following result [25].

Lemma 3.1 *For any public key h in NTRU-1998, there exists a polynomial $h' \in R$ with overwhelming probability such that for any $r \in \mathcal{L}_r$*

$$h' * h * r = r \bmod q.$$

It requires $O(N^2)$ arithmetic operations.

It also holds true for NTRU-2001 with $q = d_r$ and NTRU-2005 with $q \mid d_r$.

One takes NTRU-2005 with $q \mid d_r$ as an example to explain how to find h' in polynomial time as follows. If $gcd(q, d_g) = 1$, then $\mathbf{H}$ is invertible in $\mathbb{Z}_q^{N\times N}$ with high probability, as mentioned in [24]. Hence, we can assume that $q \mid d_g$. Since $R_q = \mathbb{Z}_q[x]/(x^N - 1)$ is isomorphic to $P_1 \times P_2$ where $P_1 = \mathbb{Z}_q[x]/(x - 1)$ and $P_2 = \mathbb{Z}_q[x]/(x^{N-1} + x^{N-2} + \cdots + 1)$, we have

$$\phi : R_q \widetilde{\rightarrow} P_1 \times P_2.$$

Since $h(1) = 0 \bmod q$, we have $\phi(h) = (0, \bar{h})$ (therefore h is not invertible in R_q), where $\bar{h}$ denotes the reduction of h modulo $x^{N-1} + x^{N-2} + \cdots + 1$. Note that $x^{N-1} + x^{N-2} + \cdots + 1$ is an irreducible polynomial, the proportion that the (random) $\bar{h}$ is invertible in P_2 with very high probability (it's equal to $1 - q^{1-N}$). We denote its inverse in P_2 by $\tilde{h}$, then $\bar{h} * \tilde{h} = 1$ over P_2. Using Extended Euclidean Algorithm for $x^{N-1} + x^{N-2} + \cdots + 1$ and $\bar{h}$ in $\mathbb{Z}_q[x]$, we can get $\tilde{h}$ with $O(N^2)$ arithmetic operations (see [26], pp.71-72, Corollary 4.6). Meanwhile, using the above algorithm, we compute polynomials u and v satisfying $(x^{N-1} + x^{N-2} + \cdots + 1)u + (x - 1)v = 1$. Then the Chinese Remainder Theorem tells us that

$$\begin{aligned} h' &= \phi^{-1}((1, \tilde{h})) \\ &= (x^{N-1} + x^{N-2} + \cdots + 1)u + (x-1)v\tilde{h} \\ &= 1 + (x-1)v(\tilde{h} - 1) \qquad\qquad \bmod (x^N - 1) \end{aligned}$$

in R_q and it uses $O(N^2)$ arithmetic operations. Since $\phi(h' * h) = (1, \tilde{h})(0, \bar{h}) = (0, 1)$, we can set

$$h' * h = \omega(x)(x^{N-1} + x^{N-2} + \cdots + 1) + 1 \bmod q,$$

where $\omega(x)$ satisfies $N\omega(1) + 1 = 0 \bmod q$. Hence, together with $q \mid d_r$, for $r \in \mathcal{B}(d_r)$ we have

$$\begin{aligned} & h' * h * r \\ = \; & (1, x, \cdots, x^{N-1}) \begin{pmatrix} \omega(x)+1 & \omega(x) & \dots & \omega(x) \\ \omega(x) & \omega(x)+1 & \dots & \omega(x) \\ \vdots & \vdots & \ddots & \vdots \\ \omega(x) & \omega(x) & \dots & \omega(x)+1 \end{pmatrix} \mathbf{r} \bmod q \\ = \; & (1, x, \cdots, x^{N-1})\mathbf{r} = r \bmod q. \end{aligned}$$

Let

$$\mathbf{H}' = \begin{pmatrix} h'_0 & h'_{N-1} & \dots & h'_1 \\ h'_1 & h'_0 & \dots & h'_2 \\ \vdots & \vdots & \ddots & \vdots \\ h'_{N-1} & h'_{N-2} & \dots & h'_0 \end{pmatrix}$$

then we have

$$\mathbf{H}'\mathbf{m} + \mathbf{r} = \mathbf{H}'\mathbf{c} \bmod q$$

from (2.3). Note that $h'' = \phi^{-1}((1, \bar{h}))$ is the invertible element of h', then $\mathbf{H}'$ is an invertible circular matrix from Theorem 2.3. Similarly, let $\hat{\mathbf{H}} = \mathbf{H}', \mathbf{b} = \mathbf{H}'\mathbf{c} \bmod q$, then we also get the formula (3.4).

With the analysis above, it's natural to get the follow theorem.

Theorem 3.2 *Given a uniformly random instance of NTRU-1998, NTRU-2001 with an odd d_g, NTRU-2001 with $q = d_r$, NTRU-2005 with $gcd(q, d_g) = 1$ and NTRU-2005 with $q \mid d_r$, i.e. for any message* $\mathbf{m}$*, ciphertext* $\mathbf{c}$*, public key* $\mathbf{H}$ *(or polynomial h) and the corresponding blinding value vector* $\mathbf{r}$*, there exists a polynomial time algorithm that on input* $\mathbf{H}$ *or h outputs an invertible circular matrix* $\hat{\mathbf{H}}$ *with very high probability, such that*

$$\hat{\mathbf{H}}\mathbf{m} + \mathbf{r} = \mathbf{b} \bmod q,$$

where $\mathbf{b} = \hat{\mathbf{H}}\mathbf{c} \bmod q$*. It requires $O(N^2)$ arithmetic operations.*

Based on Theorem 3.2, we can obtain the following main theorem.

Theorem 3.3 *For the NTRU-1998 (also NTRU-2001 with an odd d_g, NTRU-2001 with $q = d_r$, NTRU-2005 with $gcd(q, d_g) = 1$ and NTRU-2005 with $q \mid d_r$) cryptosystem with enough (reaches $O(\frac{3N}{2})$) independent recipients' ciphertexts and corresponding public keys known, there exists a polynomial time algorithm to recover the plaintext successfully.*

Proof. Algorithm consist of three steps.
Step 1. Separating $\mathbf{r}$ from $\mathbf{Hr}$ and Derandomization
By the Theorem 3.2, we can get

$$\mathbf{r} = \mathbf{b} - \hat{\mathbf{H}}\mathbf{m} \bmod q.$$

Then, we do the inner product

$$(\mathbf{b} - \hat{\mathbf{H}}\mathbf{m})^T(\mathbf{b} - \hat{\mathbf{H}}\mathbf{m}) = \mathbf{r}^T\mathbf{r} \bmod q.$$

Note that $\mathbf{r}^T\mathbf{r} = d$ is a constant, we get

$$\mathbf{m}^T\hat{\mathbf{H}}^T\hat{\mathbf{H}}\mathbf{m} - 2\mathbf{b}^T\hat{\mathbf{H}}\mathbf{m} = d - \mathbf{b}^T\mathbf{b} \bmod q. \tag{3.5}$$

From Lemma (2.2), $\hat{\mathbf{H}}^T\hat{\mathbf{H}}$ is a symmetric circular matrix.
Step 2. Linearization
We linearize the equations (3.5). Let $d - \mathbf{b}^T\mathbf{b} = s$, $\mathbf{b}^T\hat{\mathbf{H}} = (w_0, w_1, \cdots, w_{N-1})$ and

$$\hat{\mathbf{H}}^T\hat{\mathbf{H}} = \begin{pmatrix} a_0 & a_{N-1} & \dots & a_1 \\ a_1 & a_0 & \dots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{N-1} & a_{N-2} & \dots & a_0 \end{pmatrix}$$

where $a_i = a_{N-i}$, for $i \in \{0, 1, \cdots, N-1\}$. From (3.5), we can get

$$\begin{aligned} & a_0(m_0^2 + m_1^2 + \cdots + m_{N-1}^2) \\ & + a_1(m_1m_0 + m_2m_1 + \cdots + m_0m_{N-1}) \\ & + \cdots\cdots\cdots\cdots \\ & + a_{N-1}(m_{N-1}m_0 + m_0m_1 + \cdots + m_{N-2}m_{N-1}) \\ & - 2w_0m_0 - 2w_1m_1 - \cdots - 2w_{N-1}m_{N-1} = s \bmod q \end{aligned} \tag{3.6}$$

Let $x_i = m_im_0 + m_{i+1}m_1 + \cdots + m_{N-1}m_{N-i-1} + m_0m_{N-i} + \cdots + m_{i-1}m_{N-1}$, for $i = 0, 1, \cdots, N-1$. Note that N is an odd prime, $a_i = a_{N-i}$ and $x_i = x_{N-i}$ for $i \in \{0, 1, \cdots, N-1\}$, then the formula (3.6) is equivalent to

$$a_0x_0 + 2a_1x_1 + \cdots + 2a_{[\frac{N}{2}]}x_{[\frac{N}{2}]} - 2w_0m_0 - 2w_1m_1 - \cdots - 2w_{N-1}m_{N-1} = s \bmod q. \tag{3.7}$$

Of course, even if N is even, we can easily obtain the same result.

Furthermore, NTRU is easily seen to be semantically insecure, since $r(1) = 0$ in NTRU-1998 or $r(1) = d_r$ in NTRU-2001 and NTRU-2005. From formula (2.2), we get

$$h(1)r(1) + m(1) = c(1) \bmod q,$$

For several variants of NTRU, we distinguish three cases:
• For NTRU-1998, we can easily get

$$m_0 + m_1 + \cdots + m_{N-1} = c(1) \bmod q. \tag{3.8}$$

and

$$(m_0 + m_1 + \cdots + m_{N-1})^2 = (c(1))^2 \bmod q.$$

The formula above is equivalent to

$$x_0 + 2x_1 + \cdots + 2x_{[\frac{N}{2}]} = c(1)^2 \bmod q. \tag{3.9}$$

By combining the formulae (3.7) and (3.9), we can get

$$2(a_1 - a_0)x_1 + \cdots + 2(a_{[\frac{N}{2}]} - a_0)x_{[\frac{N}{2}]} - 2w_0m_0 - 2w_1m_1 - \cdots - 2w_{N-1}m_{N-1} = s - a_0c(1)^2 \bmod 2^k. \tag{3.10}$$

What's more, there is a positive integer $u \geq 1$ such that 2^u divides the greatest common divisor of all the coefficients but 2^{u+1} can not. If $u \geq k$, then the equation (3.10) is noneffective. And in fact $u = 1$ holds for very high probability. We just use those samples in which $u = 1$, so we have

$$(a_1 - a_0)x_1 + \cdots + (a_{[\frac{N}{2}]} - a_0)x_{[\frac{N}{2}]} - w_0m_0 - w_1m_1 - \cdots - w_{N-1}m_{N-1} = \frac{s - a_0c(1)^2}{2} \bmod 2^{k-1}.$$

Notice that the formula (3.8) can be obtained from any other recipient's ciphertext. Hence we need at least $N + [\frac{N}{2}] - 1$ recipients' ciphertexts and the corresponding public keys to obtain a system of linear congruence equations

$$\mathbf{L} \times \mathbf{Y} = \mathbf{S} \bmod 2^{k-1},$$

where $\mathbf{L}$ is a $(N + [\frac{N}{2}]) \times (N + [\frac{N}{2}])$ matrix, $Y = (x_1, x_2, \cdots, x_{[\frac{N}{2}]}, m_0, m_1, \cdots + m_{N-1})^T$ and $\mathbf{S}$ is a constant vector. However, in the practical experiments, we take $\mathbf{L} \in \mathbb{Z}_q^{Q \times (N + [\frac{N}{2}])}$ to guarantee that the rank of $\mathbf{L}$ equals to $N + [\frac{N}{2}]$, where $Q = N + [\frac{N}{2}] + l, l \in \mathbb{N}$. Fortunately, in practice scheme $q = 128,\ 256$ or other larger number of the form 2^k, Table 1 in Section 2.4 indicates that even if we take $l = 1$, the rank of the random matrix $\mathbf{L}$ equals to $N + [\frac{N}{2}]$ with very high probability (close to 1).

• For NTRU-2001 with an odd d_g and NTRU-2001 with $q = d_r$, since $m_i^2 = m_i$ holds for $m_i \in \{0, 1\}$, we have

$$2a_1x_1 + 2a_2x_2 + \cdots + 2a_{[\frac{N}{2}]}x_{[\frac{N}{2}]} + (a_0 - 2w_0)m_0 + (a_0 - 2w_1)m_1 + \cdots + (a_0 - 2w_{N-1})m_{N-1} = s \bmod q. \tag{3.11}$$

Similar to NTRU-1998, we can easily get

$$m_0 + m_1 + \cdots + m_{N-1} = c(1) - d_rh(1) \bmod q. \tag{3.12}$$

and

$$(m_0 + m_1 + \cdots + m_{N-1})^2 = (c(1) - d_rh(1))^2 \bmod q.$$

The formula above is equivalent to

$$2x_1 + 2x_2 + \cdots + 2x_{[\frac{N}{2}]} = c(1) - d_rh(1) - 1)(c(1) - d_rh(1)) \bmod q$$

By combining the formulae (3.11) and (3.12), we can get

$$2a_1x_1 + 2a_2x_2 + \cdots + 2a_{[\frac{N}{2}]}x_{[\frac{N}{2}]} - 2w_0m_0 - 2w_1m_1 - \cdots - 2w_{N-1}m_{N-1} = s - a_0(c(1) - d_rh(1)) \bmod q.$$

There is a positive integer $u \geq 1$ such that 2^u divides the greatest common divisor of all the coefficients but 2^{u+1} can not. Similar to NTRU-1998, $u = 1$ holds for very high probability. We just use those samples in which $u = 1$, so we have

$$a_1x_1 + a_2x_2 + \cdots + a_{[\frac{N}{2}]}x_{[\frac{N}{2}]} - w_0m_0 - w_1m_1 - \cdots - w_{N-1}m_{N-1} = \frac{s - a_0(c(1) - d_rh(1))}{2} \bmod 2^{k-1}.$$

and

$$x_1 + x_2 + \cdots + x_{[\frac{N}{2}]} = \frac{c(1) - d_r h(1) - 1)(c(1) - d_r h(1))}{2} \bmod 2^{k-1}. \tag{3.13}$$

Notice that the two formulae (3.12) and (3.13) can be obtained from any other recipient's ciphertext. Hence, we need at least $N + [\frac{N}{2}] - 2$ recipients' ciphertexts/public-keys to obtain a system of linear congruence equations $\mathbf{L} \times \mathbf{Y} = \mathbf{S} \bmod 2^{k-1}$, where $\mathbf{L}$ is a $(N + [\frac{N}{2}]) \times (N + [\frac{N}{2}])$ matrix, $Y = (x_1, x_2, \cdots, x_{[\frac{N}{2}]}, m_0, m_1, \cdots + m_{N-1})^T$ and $\mathbf{S}$ is a constant vector. However, in the practical experiments, we take $\mathbf{L} \in \mathbb{Z}_q^{Q \times (N + [\frac{N}{2}])}$ to guarantee that the rank of $\mathbf{L}$ equals to $N + [\frac{N}{2}]$, where $Q = N + [\frac{N}{2}] + l, l \in \mathbb{N}$. Table 1 in Section 2.4 indicates that even if we take $l = 1$, the rank of $\mathbf{L}$ equals to $N + [\frac{N}{2}]$ with very high probability (close to 1).

• For NTRU-2005 with $gcd(q, d_g) = 1$ and NTRU-2005 with $q \mid d_r$, since $m_i^2 = m_i$ holds for $m_i \in \{0, 1\}$, similar to NTRU-2001, we have

$$m_0 + m_1 + \cdots + m_{N-1} = c(1) - d_r h(1) \bmod q, \tag{3.14}$$

$$2x_1 + 2x_2 + \cdots + 2x_{[\frac{N}{2}]} = c(1) - d_r h(1) - 1)(c(1) - d_r h(1)) \bmod q \tag{3.15}$$

and

$$2a_1x_1 + 2a_2x_2 + \cdots + 2a_{[\frac{N}{2}]}x_{[\frac{N}{2}]} - 2w_0m_0 - 2w_1m_1 - \cdots - 2w_{N-1}m_{N-1} = s - a_0(c(1) - d_r h(1)) \bmod q. \tag{3.16}$$

Since q is a odd prime, there exist the inverse of 2 mod q. Thus, the formulae (3.15) and (3.16) are equivalent to

$$x_1 + x_2 + \cdots + x_{[\frac{N}{2}]} = s' \bmod q. \tag{3.17}$$

and

$$a_1x_1 + a_2x_2 + \cdots + a_{[\frac{N}{2}]}x_{[\frac{N}{2}]} - w_0m_0 - w_1m_1 - \cdots - w_{N-1}m_{N-1} = s'' \bmod q. \tag{3.18}$$

Notice that the two formulae (3.14) and (3.17) can be obtained from any other recipient's ciphertext. Hence, we need at least $N + [\frac{N}{2}] - 2$ recipients' ciphertexts/public-keys to obtain a system of linear congruence equations $\mathbf{L} \times \mathbf{Y} = \mathbf{S} \bmod q$, where $\mathbf{L}$ is a $(N + [\frac{N}{2}]) \times (N + [\frac{N}{2}])$ matrix, $Y = (x_1, x_2, \cdots, x_{[\frac{N}{2}]}, m_0, m_1, \cdots + m_{N-1})^T$ and $\mathbf{S}$ is a constant vector. However, in the practical experiments, we take $\mathbf{L} \in \mathbb{Z}_q^{Q \times (N + [\frac{N}{2}])}$ to guarantee that the rank of $\mathbf{L}$ equals to $N + [\frac{N}{2}]$, where $Q = N + [\frac{N}{2}] + l, l \in \mathbb{N}$. NTRU-2005 in [3] takes $q = 197,\ 251,\ 367$ or larger primes in practice scheme, Table 1 indicates that even if we take $l = 0$, the rank of $\mathbf{L}$ equals to $N + [\frac{N}{2}]$ with very high probability (close to 1).

Step 3. Solving the system of linear congruence equations

We use Gaussian elimination to solve

$$\mathbf{L} \times \mathbf{Y} = \mathbf{S} \bmod q'.$$

and the output $(m_0, m_1, \cdots + m_{N-1})$ is the plaintext m. It requires $O(N^3)$ arithmetic operations (see [27], pp.47-48, Algorithm 2.2.1). More accurately, since $\mathbf{L} \in \mathbb{Z}_q^{(N + [\frac{N}{2}] + l) \times (N + [\frac{N}{2}])}$ and the rank of $\mathbf{L}$ equals to

$N + [\frac{N}{2}]$, we apply Gaussian elimination to $(\mathbf{L}|\mathbf{S})$ mod q' and get

$$\underbrace{\left(\begin{array}{cccc|c} u_{11} & u_{12} & \cdots & u_{1n} & v_1 \\ & u_{22} & \cdots & u_{2n} & v_2 \\ & & \ddots & \vdots & \vdots \\ & & & u_{nn} & v_n \\ \hline & & & & 0 \\ & \mathbf{0} & & & \vdots \\ & & & & 0 \end{array}\right)}_{n} \begin{array}{c} \left.\begin{array}{c} \\ \\ \\ \\ \end{array}\right\} n \\ \left.\begin{array}{c} \\ \\ \\ \end{array}\right\} l \end{array} \quad \underbrace{\qquad}_{1} \tag{3.19}$$

where $n = N + [\frac{N}{2}]$ and $u_{ii} \neq 0, i = 1, 2, \cdots, n$. Thus, we use back substitution method to solve $m_{N-1}, m_{N-2}, \cdots, m_1$, not the whole $\mathbf{Y}$.

Another way is to use the Hermite Normal Form. For $\mathbf{L} = (\mathbf{L}_1, \mathbf{L}_2, \cdots, \mathbf{L}_{N+[\frac{N}{2}]})$, we set $\overline{\mathbf{L}} = (\mathbf{L}_{[\frac{N}{2}]+1}, \mathbf{L}_{[\frac{N}{2}]+2}, \cdots, \mathbf{L}_{N+[\frac{N}{2}]}, \mathbf{L}_1, \mathbf{L}_2, \cdots, \mathbf{L}_{[\frac{N}{2}]})$ and $\overline{\mathbf{Y}} = (m_0, m_1, \cdots, m_{N-1}, x_1, x_2, \cdots, x_{[\frac{N}{2}]})^T$. We compute the Hermite Normal Form $\mathbf{B}$ of $\overline{\mathbf{L}}^T$: $\mathbf{B} = \overline{\mathbf{L}}^T\mathbf{U}$, where $\mathbf{U} \in \mathbb{Z}_{q'}^{(n+l)\times(n+l)}$ is a unimodular matrix (see [27], pp.69, Algorithm 2.4.6), then get $\mathbf{B}^T\overline{\mathbf{Y}} = \mathbf{U}^T\mathbf{S}$ mod q' by multiplying $\mathbf{U}^T$, finally by iteration solve $m_0, m_1, \cdots, m_{N-1}$ in turn, not the whole $\overline{\mathbf{Y}}$. □

Specifically, we have the following result ($l \in \mathbb{N}$):

Variant	N	the number of variables	Recipients	Binary length	Time
NTRU-1998	N	$N + [\frac{N}{2}] = O(3N/2)$	$N + [\frac{N}{2}] - 1 + l$	$O(\log N + 2\log(q-1) + 1)$	$O(N^3)$
NTRU-2001	N	$N + [\frac{N}{2}] = O(3N/2)$	$N + [\frac{N}{2}] - 2 + l$	$O(\log N + 2\log(q-1) + 1)$	$O(N^3)$
NTRU-2005	N	$N + [\frac{N}{2}] = O(3N/2)$	$N + [\frac{N}{2}] - 2 + l$	$O(\log N + 2\log(q-1) + 1)$	$O(N^3)$

Pan and Dent in [11] give the following result:

Variant	N	the number of variables	Recipients	Time
NTRU-1998	N	$O(N^3/6)$	$O(N^2/6)$	$O(N^9)$
NTRU-2001	N	$O(N^2/2)$	$O(N/2)$	$O(N^6)$
NTRU-2005	N	$O(N^2/2)$	$O(N/2)$	$O(N^6)$

Remark 3: Compare the two tables above, our method is very efficient in the number of variables and time complexity. In particular, for NTRU-1998 our method is very efficient and better than that in [11]. We eliminate the blinding value vector $\mathbf{r}$ directly and entirety by doing the inner product from every recipient's ciphertext, differ from eliminating $\mathbf{r}$ by using ergodic in [11]. Clearly, our algorithm also holds in the case that N is even. However, it doesn't work against NTRU with encryption padding.

4 Experimental Results

All experiments were performed on a Windows XP system with a 2.93 GHz Pentium 4 processor and 4 GByte RAM using Shoup's NTL library version 5.4.1 [28].

We implemented the broadcast attacks against three variants of NTRU. For NTRU-1998 and NTRU-2001, we adopted the algorithms for $u = 1$. In our experiments, we always obtained an matrix $\mathbf{L}$, whose the rank equals to $N + [\frac{N}{2}]$. And the number of recipients is just a little more than the number of variables (denoted by $\mathbf{T}$). Since the number of variables is small, the experiment evidence indicates that our algorithm can efficiently broadcast attack against NTRU with the big parameters. Some results against NTRU with the highest security parameters are listed below:

Variant	N	q	p	d_f	d_g	d_r	$\mathbf{T}$	Recipients	Rank($\mathbf{L}$)	Result
NTRU-1998	503	256	3	216	76	55	754	757	754	success
NTRU-2001	503	256	3	216	75	55	754	764	754	success
NTRU-2005	503	257	2	216	72	55	754	757	754	success

5 Conclusion

In this paper, we first discuss Ding's algorithm against GGH, then naturally deduce new and uniform broadcast attacks against several variants of NTRU, which is based on the special structure of blinding value space $\mathcal{L}_r$. From which we can see two main lines to study the algebraic broadcast attacks: one is decreasing the number of variables; the other is increasing the number of equations. Now, the main question is that how to do the broadcast attacks against NTRU, GGH and other cryptosystems more efficiently if the error vectors lack of the special structure.

References

[1] J. Hoffstein, J. Pipher, and J.H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem, in Proc. of Algorithmic Number Theory (Lecture Notes in Computer Science), J.P. Buhler, Ed. Berlin, Germany: Springer-Verlag, 1998, vol. 1423, pp. 267-288.

[2] J. Hoffstein, and J.H. Silverman. Optimizations for NTRU. Technical report, NTRU Cryptosystems (June 2000), available at http://citeseer.ist.psu.edu/693057.html.

[3] N. Howgrave-Graham, J.H. Silverman, and W. Whyte. Choosing Parameter Sets for NTRUEncrypt with NAEP and SVES-3. Technical Report, NTRU Cryptosystems 2005.

[4] D. Coppersmith, and A. Shamir. Lattice attacks on NTRU, in Proc of EuroCrypt'97 (Lecture Notes in Computer Science), W. Fumy, Ed. Berlin, Germany: Springer, 1997, Vol. 1233 pp. 52-61.

[5] E. Jaulmes, and A. Joux. A Chosen-Ciphertext Attack against NTRU. Advances in Cryptology-CRYPTO 2000, Lecture Notes in Computer Science, 2000, Volume 1880/2000, 20-35.

[6] A. May, and J.H. Silverman. Dimension Reduction Methods for Convolution Modular Lattices, in Proc of Cryptography and Lattices (Lecture Notes in Computer Science), J.H. Silverman, Ed. Berlin, Germany: Springer- Verlag, 2001, vol. 2146, pp. 110-125.

[7] N. Howgrave-Graham, J.H. Silverman, and W. Whyte. A Meet- In-The-Meddle Attack on an NTRU Private Key. Technical Report, available at http: //www.ntru.com/cryptolab/tech notes.htm 004.

[8] N. Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In Proc. of CRYPTO 2007, pp. 150-169, 2007.

[9] J. Hästad. Solving simultaneous modular equations of low degree. SIAM J. Comput. 17 (1988) 336-341.

[10] T. Plantard, and W. Susilo. Broadcast attacks against lattice-based cryptosystems. (ACNS 2009).

[11] Y. Pan, and Y. Deng. A broadcast attack against NTRU using Ding's Algorithm, available at http://eprint.iacr.org/2010/598.

[12] G. V. Bard. Algebraic Cryptanalysis. Springer, 2009.

[13] S. Arora, and R. Ge. Learning Parities with Structured Noise, TR10-066, April 2010.

[14] J. Ding. Solving LWE Problem with Bounded Errors in Polynomial Time, available at http://eprint.iacr.org/2010/558.

[15] Y. Pan, Y. Deng, Y. Jiang, and Z. Tu. A New Lattice-Based Cryptosystem Mixed with a Knapsack. Cryptology ePrint Archive, Report 2009/337, available at http: //eprint.iacr.org/2009/337.

[16] O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reductions problems. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 112-131. Springer, Heidelberg (1997).

[17] P. Q. Nguyen. Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from crypto 1997. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 288-304. Springer, Heidelberg (1999).

[18] R. Fischlin, and J. P. Seifert. Tensor-based trapdoors for cvp and their application to public key cryptography. In: IMA Int. Conf. pp. 244-257 (1999).

[19] D. Micciancio. Improving lattice based cryptosystems using the Hermite normal form. In: Silverman, J.H. (ed.) CaLC 2001. LNCS, vol. 2146, pp. 126-145. Springer, Heidelberg (2001).

[20] S. H. Paeng, B. E. Jung, and K. C. Ha. A lattice based public key cryptosystem using polynomial representations. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 292-308. Springer, Heidelberg (2003).

[21] C. Gentry. Key Recovery and Message Attacks on NTRU-Composite. In Proc. of Eurocrypt '01, LNCS 2045, pages 182-194. Springer-Verlag, 2001.

[22] P. Mol, and M. Yung. Recovering NTRU Secret Key from Inversion Oracle, In Proc of PKC 2008. 2008, 18-36.

[23] P. J. Davis. Circulant Matrices. New York: John Wiley and Sons Co, 1979.

[24] J. Hoffstein, and J. H. Silverman. Invertibility in truncated polynomial rings. Technical report, NTRU Cryptosystems, October 1998. Report #009, version 1, available at http://www.ntru.com.2002.

[25] P. Nguyen, and D. Pointcheval. Analysis and Improvements of NTRU Encryption Padding. In Proc. of Crypto'02, Berlin: Springer-Verlag, 2002, vol. 2442, pp. 210-225.

[26] Joachim von zur Gathen, and Jurgen Gerhard. Modern computer algebra (2nd ed). Cambridge, UK; New York, NY, USA: Cambridge University Press, 2003,pages 255-256.

[27] H. Cohen. A course in computational algebraic number theory. New York : Springer-Verlag, c1993.

[28] V. Shoup. NTL: A library for doing number theory. Available at http://www.shoup.net/ntl/

Appendix A: How to get $\mathbf{H}^{-1}$ and $\mathbf{H}^T\mathbf{H}$

We set $\mathbf{g} = (g_0, g_1, \cdots, g_{N-1})^T$ satisfying

$$\begin{pmatrix} h_0 & h_{N-1} & \dots & h_1 \\ h_1 & h_0 & \dots & h_2 \\ \vdots & \vdots & \ddots & \vdots \\ h_{N-1} & h_{N-2} & \dots & h_0 \end{pmatrix} \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{N-1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \bmod q,$$

then $\mathbf{g}$ has a unique solution over $\mathbb{Z}_q^N$ since $\mathbf{H} \in C^{N\times N}$ is invertible over $\mathbb{Z}_q^{N\times N}$. It requires $O(N^2)$ arithmetic operations (see [27], pp.47-49, Algorithm 2.2.1). Then

$$\mathbf{H}^{-1} = \begin{pmatrix} g_0 & g_{N-1} & \cdots & g_1 \\ g_1 & g_0 & \cdots & g_2 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{pmatrix}.$$

Because, for any vector $\mathbf{v} = (v_0, v_1, \cdots, v_{N-1})^T$, we denote by $\mathbf{v}^{(i)}$ its i-cycle:

$$\mathbf{v}^{(i)} = \begin{cases} \mathbf{v} & i = 0; \\ (v_{N-i}, v_{N-i+1}, \cdots, v_{N-1}, v_0, v_1, \cdots, v_{N-i-1})^T, & i \in \{1, 2, \cdots, N-1\}. \end{cases}$$

Then $\mathbf{H}\mathbf{g}^{(i)} = \mathbf{E}_{i+1} \bmod q$, where $\mathbf{E}_i$ is a column vector whose i-th entry is 1 and else are 0.

If $\mathbf{G}, \mathbf{H} \in \mathbb{Z}_q^{N\times N}$ are circular matrixs, for $i, j \in \{1, 2, \cdots, N\}$, we have

$$\begin{aligned} (\mathbf{GH})_{i,j} &= g_{i-1}h_{N-j+1} + g_{i-2}h_{N-j+2} + \cdots + g_i h_{N-j} \\ &= \sum_{l=0}^{N-1} g_l h_{N-j+i-l}, \end{aligned}$$

$$\begin{aligned} (\mathbf{GH})_{N-j+i+1,1} &= g_{N-j+i}h_0 + g_{N-j+i-1}h_1 + \cdots + g_{N-j+i+1}h_{N-1} \\ &= \sum_{l=0}^{N-1} g_l h_{N-j+i-l}. \end{aligned}$$

Hence, $(\mathbf{GH})_{i,j} = (\mathbf{GH})_{N-j+i+1,1}$, for $i, j \in \{1, 2, \cdots, N\}$, i.e. $\mathbf{GH}$ is also a circular matrix. In particular, $\mathbf{H}^T\mathbf{H}$ is a symmetric circular matrix. Hence, $(\mathbf{H}^T\mathbf{H})_{i,1} = (\mathbf{H}^T\mathbf{H})_{1,i} = (\mathbf{H}^T\mathbf{H})_{N+2-i,1}$, for $i \in \{1, 2, \cdots, N\}$. It's sufficient to calculate $\{(\mathbf{H}^T\mathbf{H})_{1,1}, (\mathbf{H}^T\mathbf{H})_{2,1}, \cdots, (\mathbf{H}^T\mathbf{H})_{[\frac{N}{2}]+1,1}\}$, which requires $(2N-1)([\frac{N}{2}]+1)$ arithmetic operations.

Appendix B: Proof of Theorem 2.5

Theorem 2.5 is equivalent to consider the set of $n \times (n+l)$ matrices with entries in $\mathbb{F}_q$. We count the number of matrices of the form $(\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{n+1})$ of rank n, where $\mathbf{b}_i \in \mathbb{Z}_q^n$. Denote by B_k the subspace spanned by $\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_k$, with the convention B_0 being the nullspace. Recall that a k-dimensional subspace has cardinality q^k. For each family $(\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{n+1})$ of rank n, there exists a unique i such that $\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{i-1}$ are linearly independent, $\mathbf{b}_i \in B_{i-1}$, and for all $j > i, \mathbf{b}_j \notin B_{j-1}$. There are $\prod_{k=0}^{i-2}(q^n - q^k)$

possibilities for $\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{i-1}$. There are q^{i-1} choices for $\mathbf{b}_i$. And there are $\prod_{k=i-1}^{n-1}(q^n - q^k)$ possibilities for $\mathbf{b}_{i+1}, \mathbf{b}_{i+2}, \cdots, \mathbf{b}_{n+1}$. It follows that the total number of families is:

$$\sum_{i=1}^{n+1} q^{i-1} \prod_{k=0}^{n-1}(q^n - q^k) = q^{n(n+1)} \prod_{k=2}^{n+1}(1 - q^{-k}).$$

Now, consider a family $(\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{n+2})$ of rank n. There exists a unique (i, j) with $i < j$ such that $\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{i-1}$ are linearly independent, $\mathbf{b}_i \in B_{i-1}$, for all $i < t < j$, $\mathbf{b}_t \notin B_{t-1}$, $\mathbf{b}_j \in B_{j-1}$ and for all $t > j$, $\mathbf{b}_t \notin B_{t-1}$. That way, we know the dimension of B_t for all t, and therefore, the number of $(\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{n+2})$ corresponding to a given (i, j) is:

$$\prod_{k=0}^{i-2}(q^n - q^k) \times q^{i-1} \times \prod_{k=i-1}^{j-3}(q^n - q^k) \times q^{j-2} \times \prod_{k=j-2}^{n-1}(q^n - q^k).$$

It follows that the total number of families of rank n is:

$$\prod_{k=0}^{n-1}(q^n - q^k) \times \sum_{i=1}^{n+1} \sum_{j=i+1}^{n+2} q^{i+j-3}.$$

Then computer the double sum:

$$\sum_{i=1}^{n+1} \sum_{j=i+1}^{n+2} q^{i+j-3} = \sum_{i=1}^{n+1} \frac{q^{i+n} - q^{2i-2}}{q-1} = \frac{(q^{n+2}-1)(q^{n+1}-1)}{(q-1)(q^2-1)}.$$

Therefore, the number of families is:

$$q^{n(n+2)} \prod_{k=3}^{n+2}(1 - q^{-k}).$$

Note: The proof above is modelled on the proof of Theorem 3 in [17].