A Broadcast Attack against NTRU Using Ding's Algorithm

Yanbin Pan, Yingpu Deng
Key Laboratory of Mathematics Mechanization
Academy of Mathematics and Systems Science,Chinese Academy of Sciences
Beijing 100190, China
{panyanbin,dengyp}@amss.ac.cn

Abstract

Very recently, Ding proposed an ingenious algorithm to solve LWE problem with bounded errors in polynomial time. We find that it can be easily used to give a broadcast attack against NTRU, the most efficient lattice-based public-key cryptosystem known to date.

Keywords: Broadcast attack, NTRU, lattice-based cryptosystems, Ding's Algorithm.

1 Introduction

In 1988, Hästad [3] proposed the first broadcast attack against public key cryptosystems. The attack enables an attacker to recover the plaintext sent by a sender to multiple recipients, without requiring any knowledge of the recipient's secret key.

In 2009, Plantard and Susilo [13] 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, Ding proposed an ingenious algorithm to solve LWE problem with bounded errors in polynomial time.

We find that it can be easily used to give a broadcast attack against NTRU. As we know, it is the first broadcast attack against NTRU.

We have to point out that some other lattice-based cryptosystems, such as [12], can not resist the broadcast attack either.

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

2 Preliminaries

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

2.1 NTRU

The NTRU cryptosystem proposed by Hoffstein, Pipher, Silverman [7] is the most practical scheme known to date. It features reasonably short, easily created keys, high speed, and low memory requirements. By the results of Coppersmith and Shamir [1], the security of NTRU is related, but not equivalent, to the hardness of some lattice problems. To date, the chosen-ciphertext attacks against NTRU may be the most dangerous. Most of the ciphertext-only attacks [9, 5, 4] against NTRU rely on the underlying lattice's special cyclic structure.

For the completeness, we give a simple description of the NTRU cryptosystem. For more details see [7].

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. 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 subsection. Every element in R can be represented as a polynomial or a vector. For example, for $f \in R$, we can represent f as

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

We work in the ring R.
Key Generation:

Step 1 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 $f * F_p = 1 \bmod p$.

Step 2 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 $r \in \mathcal{L}_r$, then compute the ciphertext:

$$c = h * r + m \bmod q$$

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. We may use totally different ways to attack NTRU instead of a uniform one.

As in [10], we summarize 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	$L(d_r, d_r)$	-	[7]
NTRU-2001	$2^k \in [\frac{N}{2}, N]$	$2 + x$	$1 + p * F$	$\mathcal{B}(d_g)$	$\mathcal{B}$	$\mathcal{B}(d_r)$	$\mathcal{B}(d_F)$	[8]
NTRU-2005	prime	2	$1 + p * F$	$\mathcal{B}(d_g)$	$\mathcal{B}$	$\mathcal{B}(d_r)$	$\mathcal{B}(d_F)$	[6]

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 $0\}$.

2.2 Ding's Algorithm for LWE

The Learning with Errors (LWE) problem was first introduced by Regev [14]. Since it is hard as worst-case lattice problems, the LWE problem has many applications in constructing cryptosystems with security proofs.

We can describe the LWE problem as follows as in [2].

First, we have a parameter n, a prime modulus q , and an error probability distribution κ on the finite field $\mathbb{F}_q$ with q elements.

Let $\prod_{\mathbf{s},\kappa}$ on $\mathbb{F}_q$ be the probability distribution obtained by selecting an element $\mathbf{a}$ in $\mathbb{F}_q^n$ randomly and uniformly, choosing $e \in \mathbb{F}_q$ according to κ, and outputting $(\mathbf{a}, < \mathbf{a}, \mathbf{s} > +e)$, where + is the addition that is performed in $\mathbb{F}_q$.

We say that an algorithm solves LWE with modulus q and error distribution κ, if, for any $\mathbf{s}$ in $\mathbb{F}_q^n$, with an arbitrary number of independent samples from $\prod_{\mathbf{s},\kappa}$, it outputs $\mathbf{s}$ (with high probability).

Very recently, Ding [2] proposed an ingenious algorithm to solve LWE problem with bounded errors, in which the error probability distribution κ is on a proper subset $ES = \{e_1, e_2, \cdots, e_D\}$ of the whole finite field $\mathbb{F}_q$, in polynomial time with fixed $D(D < q)$.

The main steps of Ding's Algorithm are

- Nonlinearization: For any sample $(\mathbf{a}, < \mathbf{a}, \mathbf{s} > +e)$, if we let $b =< \mathbf{a}, \mathbf{s} > +e$, we have a linear equation
$$\sum_{i=1}^{n} \mathbf{a}_i \mathbf{s}_i + e = b. \tag{1}$$
Then we can also have a corresponding nonlinear equation
$$\prod_{k=1}^{D} (\sum_{i=1}^{n} \mathbf{a}_i \mathbf{s}_i + e_k - b) = 0 \tag{2}$$
where $e_k \in ES$.

- Linearization: Notice that the total degree of every monomial of $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n$ in (2) is at most D, we assign each of such monomials a new variable $\mathbf{y}_i$, then we transform (2) to a linear equation:
$$\sum_{i} \mathbf{c}_i \mathbf{y}_i = 0 \tag{3}$$
where $\mathbf{c}_i$ is the corresponding coefficient of $\mathbf{y}_i$ in (2).

- Solving. Let $Q = \binom{n+D}{n}$, we know that the number of $\mathbf{y}_i$'s is at most $Q - 1$. So when we have enough, $O(n^D)$, linear equations like (3), we can find $\mathbf{y}_i$ by solving a set of linear equations.

3 A Broadcast Attack against NTRU

3.1 A Broadcast Attack against NTRU

We find Ding's Algorithm can be easily used to give a broadcast attack against NTRU.

Suppose there is a sender and k recipients. All these recipients use NTRU cryptosystems with the same N, q, p but different public and private keys. The

sender encrypts the same message m with every public key of these recipients and independent $r \in \mathcal{L}_r$ respectively. Then he sends the k ciphertexts to these recipients respectively. The broadcast attack is to recover m with these k ciphertexts without requiring any knowledge of the recipient's secret key. More precisely, the attacker wants to recover m from

$$\begin{aligned} h_{(1)} * r_{(1)} + m &= c_{(1)} \bmod q \\ h_{(2)} * r_{(2)} + m &= c_{(2)} \bmod q \\ &\vdots \\ h_{(k)} * r_{(1)} + m &= c_{(k)} \bmod q \end{aligned}$$

where $h_{(i)}$ is the i-th recipient's public key and the attacker knows none of these $r_{(i)}$'s.

Before giving the attack, we need transform NTRU to its linear form.

3.1.1 The Linear Form of NTRU over $\mathbb{Z}_q$

In NTRU, a polynomial $f = \sum_{i=0}^{N-1} f_i x^i \in R$ can also be represented as a vector

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

and the multiplication of f and g can be represented as

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

That is, $\mathbf{t}$ is the corresponding vector of $f * g$ in R.

Then, for the equation in R

$$c = h * r + m \bmod q,$$

we have the linear form

$$\mathbf{c} = H\mathbf{r} + \mathbf{m} \bmod q \tag{4}$$

where

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

or we have N linear equations

$$\mathbf{c}_i = \sum_{j=0}^{N-1} H_{i+1,j+1}\mathbf{r}_j + \mathbf{m}_i \bmod q$$

where $i = 0, \cdots N-1$.

This is a little different from (1). We would like to point out that it is not necessary for q to be a prime. However, if we want to recover $\mathbf{m}$, we need $\mathbf{m}$ to be the same as $\mathbf{s}$ in (1), but not $\mathbf{r}$. So we can't use Ding's Algorithm directly.

If H is invertible in $\mathbb{Z}_q^{N\times N}$, obviously we can easily get the equation below from (4):

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

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

$$\widehat{H}\mathbf{m} + \mathbf{r} = \mathbf{b} \bmod q \tag{5}$$

3.1.2 The Pseudo-Inverse of H

Usually, H is invertible in $\mathbb{Z}_q^{N\times N}$ with high probability in NTRU-2001 and NTRU-2005. However, H is not invertible in NTRU-1998.

Luckily, for any public key h in NTRU-1998, we can usually find a polynomial $h' \in R$ with overwhelming probability such that for any $r \in L(d_r, d_r)$,

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

We call h is pseudo-invertible as in [11].

We can find h' in polynomial time as follows. 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

$$\phi : R_q \to P_1 \times P_2.$$

Since $h(1) = 0 \bmod q$, we have $\phi(h) = (0, \bar{h})$ where $\bar{h}$ denotes the reduction of h modulo $x^{N-1} + x^{N-2} + \cdots + 1$. With high probability, $\bar{h}$ is invertible in P_2. We denote its inverse in P_2 by $\widetilde{h}$. Considering the polynomial $h' = \phi^{-1}((1, \widetilde{h}))$ in R_q, it satisfies $h' * h * r = r \bmod q$ for $r \in L(d_r, d_r)$.

Let

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

we have

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

from (4).

Similarly, let $\widehat{H} = H'$, $\mathbf{b} = H'\mathbf{c}$, then we also have

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

3.1.3 A Broadcast Attack against NTRU

Obviously, we have N linear equations from (5):

$$\sum_{j=0}^{N-1} \widehat{H}_{i+1,j+1}\mathbf{m}_j + \mathbf{r}_i = \mathbf{b}_i \bmod q$$

where $i = 0, \cdots N-1$. This is totally as same as (1). So we can use Ding's Algorithm to recover $\mathbf{m}$ if we have enough linear equations, since $\mathbf{r}_i$ is usually in $\{0, 1\}$ or $\{-1, 0, 1\}$. Usually, by Ding's result, we need $O(N^2)$ and $O(N^3)$ linear equations respectively to complete the attack for $ES = \{0, 1\}$ and $ES = \{-1, 0, 1\}$. Since we can obtain N linear equations from one recipient, we expect that we need $O(N)$ and $O(N^2)$ recipients respectively to complete the attack.

3.2 Improving the Attack

Two natural ideas to improve the attack is obvious:

- decreasing the number of variables,
- increasing the number of equations.

In fact, from the view of the effect, increasing the number of equations is usually equivalent to decreasing the number of variables.

3.2.1 With the Description of ES

In NTRU, ES is usually $\{-1, 0, 1\}$ or $\{0, 1\}$.

- if $ES = \{-1, 0, 1\}$, we can add N equations: $\mathbf{m}_i^3 - \mathbf{m}_i = 0$ $(i = 0, \cdots, N-1)$.
- if $ES = \{0, 1\}$, we can add N equations: $\mathbf{m}_i^2 - \mathbf{m}_i = 0$ $(i = 0, \cdots, N-1)$.

3.2.2 With Known Bits

If we know some bits, either the message bits or the random bits, we can also improve the attack.

- If we know some bits of $\mathbf{m}$, for example, $\mathbf{m}_0, \mathbf{m}_2, \cdots, \mathbf{m}_{k-1}$, then we can obviously eliminate those monomials containing at least one of these known bits.

- If we know some bits of $\mathbf{r}$, for example, $\mathbf{r}_0, \mathbf{r}_2, \cdots, \mathbf{r}_{k-1}$, then for $i = 0, \cdots, k-1$, we have k linear equations:

$$\sum_{j=0}^{N-1} \widehat{H}_{i+1,j+1}\mathbf{m}_j + \mathbf{r}_i - \mathbf{b}_i = 0 \bmod q \tag{6}$$

  For each of these equations, we also have N equations for $ES = \{0, 1\}$:

$$\begin{gathered}
\mathbf{m}_0(\textstyle\sum_{j=0}^{N-1} \widehat{H}_{i+1,j+1}\mathbf{m}_j + \mathbf{r}_i - \mathbf{b}_i) = 0 \bmod q \\
\mathbf{m}_1(\textstyle\sum_{j=0}^{N-1} \widehat{H}_{i+1,j+1}\mathbf{m}_j + \mathbf{r}_i - \mathbf{b}_i) = 0 \bmod q \\
\vdots \\
\mathbf{m}_{N-1}(\textstyle\sum_{j=0}^{N-1} \widehat{H}_{i+1,j+1}\mathbf{m}_j + \mathbf{r}_i - \mathbf{b}_i) = 0 \bmod q
\end{gathered}$$

  Together with (6), we have $N + 1$ equations for each linear equation. Notice that though we have $(N + 1)k$ new equations, in general they are not linearly independent when we transform them to linear equations by involving new variables $\mathbf{y}_i$. We conjecture that the effect when we use these new equations is as the same as when we know k bits of $\mathbf{m}$.

  For $ES = \{-1, 0, 1\}$, we can similarly have another N^2 equations

$$\mathbf{m}_i\mathbf{m}_l(\sum_{j=0}^{N-1} \widehat{H}_{i+1,j+1}\mathbf{m}_j + \mathbf{r}_i - \mathbf{b}_i) = 0 \bmod q$$

  for $i, l = 0, \cdots, N-1$.

However, how can we obtain these bits we need? A natural idea is guessing the bits. We next show that "guessing" is not a bad idea for NTRU.

For any vector $\mathbf{v} = \mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_{N-1})^T$, we denote by $\mathbf{v}^{(r)}$ its r-cycle:

$$\mathbf{v}^{(r)} = (\mathbf{v}_{N-r}, \mathbf{v}_{N-r+1}, \cdots, \mathbf{v}_{N-1}, \mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_{N-r-1})^T$$

where $r \in \{1, \cdots, N-1\}$ and $\mathbf{v}^{(0)} = \mathbf{v}$.

For NTRU, since we have

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

we can conclude that for $i = 0, \cdots, N-1$,

$$H\mathbf{r}^{(i)} + \mathbf{m}^{(i)} = \mathbf{c}^{(i)} \bmod q.$$

We take $\mathbf{r}$ as an example to show that how we guess its some bits. For example, we guess $\mathbf{r}_0 = 0, \mathbf{r}_2 = 0, \cdots, \mathbf{r}_{k-1} = 0$. Then only if there is an i-cycle of $\mathbf{r}$ such that

$$\begin{array}{rcl} \mathbf{r}_0^{(i)} & = & \mathbf{r}_0 \\ \mathbf{r}_1^{(i)} & = & \mathbf{r}_1 \\ & \vdots & \\ \mathbf{r}_{k-1}^{(i)} & = & \mathbf{r}_{k-1} \end{array}$$

can we use the corresponding equation $H\mathbf{r}^{(i)} + \mathbf{m}^{(i)} = \mathbf{c}^{(i)} \bmod q$ instead of the origin one for any recipient to continue the attack. Of course, we don't know what i is. However, we can commit the attack for every $i \in \{0, 1, \cdots, N-1\}$, then check whether we get the correct message.

By [9], we know that the probability that there is an i-cycle we need is approximately equal to $1 - (1 - \prod_{j=0}^{d_r-1}(1 - \frac{k}{N-j}))^N$ for $\mathcal{L}(r) = \mathcal{B}(d_r)$ which is very close to 1 for small k. More analysis see [9].

3.2.3 With the Clue from the Parameters

We can easily get

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

when we take $h(x), r(x), m(x)$ and $c(x)$ as polynomials. Since we know $h(x), c(x)$, and by $\mathcal{L}_r$ we also know that $r(1) = 0$ in NTRU-1998 or $r(1) = d_r$ in NTRU-2001 and NTRU-2005, we have

$$\sum_{j=0}^{N-1} \mathbf{m}_j + h(1)r(1) - c(1) = 0 \bmod q.$$

As above, we also have N new equations for $ES = \{0, 1\}$

$$\mathbf{m}_i(\sum_{j=0}^{N-1} \mathbf{m}_j + h(1)r(1) - c(1)) = 0 \bmod q$$

for $i = 0, \cdots, N-1$.

For $ES = \{-1, 0, 1\}$, we have another N^2 equations

$$\mathbf{m}_i \mathbf{m}_l (\sum_{j=0}^{N-1} \mathbf{m}_j + h(1)r(1) - c(1)) = 0 \bmod q$$

for $i, l = 0, \cdots, N-1$.

3.3 A Toy Example

Suppose $N = 3, q = 5$, and the sender wants to send the message $\mathbf{m} = (0, 1, 0)^T$ to the recipient whose public key H satisfies

$$\widehat{H} = H^{-1} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{pmatrix} \bmod q,$$

(in fact, we don't really generate a NTRU public key and we just give an example to illustrate the algorithm here) and he chooses a random vector $\mathbf{r} = (0, 1, 1)^T$ from $\mathcal{B}(2)$. Finally, he gets the ciphertext $(2, 2, 4)^T$.

Suppose the attacker obtains the ciphertext and without loss of generality, he knows $\mathbf{r}_0 = 0$.

As above, the attacker can generate the equations below:

$$\begin{aligned}
\mathbf{m}_i^2 - \mathbf{m}_i &= 0 \quad i = 0, 1, 2 \\
\mathbf{m}_0 + 2\mathbf{m}_1 + 3\mathbf{m}_2 - 2 &= 0 \\
\mathbf{m}_i(\mathbf{m}_0 + 2\mathbf{m}_1 + 3\mathbf{m}_2 - 2) &= 0 \quad i = 0, 1, 2 \\
(3\mathbf{m}_0 + \mathbf{m}_1 + 2\mathbf{m}_2 - 1)(3\mathbf{m}_0 + \mathbf{m}_1 + 2\mathbf{m}_2 - 2) &= 0 \\
(2\mathbf{m}_0 + 3\mathbf{m}_1 + \mathbf{m}_2 - 4)(2\mathbf{m}_0 + 3\mathbf{m}_1 + \mathbf{m}_2 - 3) &= 0 \\
\mathbf{m}_0 + \mathbf{m}_1 - \mathbf{m}_2 - 1 &= 0 \\
\mathbf{m}_i(\mathbf{m}_0 + \mathbf{m}_1 - \mathbf{m}_2 - 1) &= 0 \quad i = 0, 1, 2
\end{aligned}$$

Taking every monomial appearing in the equations as a new variable, then we have 13 linear equations and only 9 variables. Very luckily, we can solve the set of linear equations.

4 Conclusion

Using Ding's Algorithm, we give a broadcast attack against NTRU, the most efficient lattice-based public-key cryptosystem known to date. We also give some ways to improve the attack. However, the lager N is, the larger the size of the set of the linear equations is, and the more ciphertexts of the recipients we need. This leads a difficult task to the attacker to commit the broadcast attack.

References

[1] D. Coppersmith, 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. 52C-61.

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

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

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

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

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

[7] J. Hoffstein, J. Pipher, 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.

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

[9] A. May, 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.

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

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

[12] 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

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

[14] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proc. of 37th STOC*, D.S. Johnson, U. Feige, Eds. New York, USA: ACM, 2005, pp. 84–93.