Trends in Mathematics
Information Center for Mathematical Sciences
Volume 8, Number 1, June, 2005, Pages 197–205

A NEW LATTICE ATTACK ON NTRU CRYPTOSYSTEM

DAEWAN HAN

Abstract. In this article a new lattice attack on NTRU cryptosystem is presented. In contrast with the previous lattice attacks, We can apply our attack to the NTRUEncrypt of high degree such that we can present the upper bound of the breaking times for the systems of degree 251. We describe some interesting properties of newly defined lattices and the attack in addition.

1. Introduction

NTRU, which consists of NTRUEcrypt and NTRUSign, is a recent high performance public-key cryptosystem. Because the system is very fast and requires small amount of memory, it is suitable for enhancing the security in constrained devices.

In [2] Coppersmith and Shamir presented the lattice attack(CS attack) on NTRU cryptosystem. They defined a lattice determined by parameters of the system and showed that recovering the secret key from the public key is reduced to finding a shortest vector of the lattice. Although a few improved attacks are suggested thereafter[3, 9, 14], basic principles of them are not different from those of CS attack.

Since CS attack was presented, authors of NTRU made numerous experiments in order to estimate the breaking times for the system of high degree[6]. Based on the experimental results, they claimed that the attacks are exponential with the degree N and presented the lower bound of the breaking times for the system of degree 251(NTRU251), which are recommended as standards[7] .

For the present it is accepted that better lattice reduction algorithms are required in order to break NTRU with the previous lattice attacks[10]. As an heuristic evidence, there is no experimental results which break the system of degree greater than 150 to our knowledge.

In this article we present a new lattice attack on NTRU cryptosystem. For the attack we define some basic lattices from parameters of NTRU and construct new lattices by intersecting them. By using the new lattices and the practical lattice reduction algorithms, we can experiment on the more higher degree than the previous attacks in practice. As a result, we can present the upper bound of the breaking times for NTRU251. We verified from the experiments that with the high probability we can break it on a single PC within about 2^{184} seconds. Because

Key words and phrases. NTRU cryptosystem, Lattice attacks.

N	q	p	d_F	d_g	d_r
251	239	2	72	72	72

TABLE 1. Parameters of NTRU251

we can apply the attack in parallel, the breaking times are reduced in proportion to the amount of computational power.

The last of this article is organized as follows: In section 2 we introduce NTRUEncrypt briefly and describe the CS attack for the comparison with the new attack. The exposition of our attack and some experimental results are given in section 3, and finally we conclude in section 4.

2. Preliminaries

The previous lattice attacks on NTRU are described as key recovery attacks. But we shall describe them as message recovery attacks on NTRUEncrypt in this article. As a matter of fact, key recovery attacks are special cases of message recovery attacks[6].

To set the grounds of our discussion, we briefly introduce NTRUEncrypt and CS attack in this section.

2.1. **Introduction of NTRUEncrypt. Parameters:** Let N be an odd prime. We will be working over the ring $\mathcal{R} = \mathbf{Z}[x]/(x^N - 1)$. The ring $\mathcal{R}$ is identified with the set of integer polynomials of degree less than N. Multiplication in $\mathcal{R}$ is denoted by $*$.

The sets D_F, D_g, and D_r are subsets of $\mathcal{R}$ that have dF,dg, and dr coefficients equal to 1 and the rest of the coefficients equal to 0 respectively. Two parameters p and q are chosen so that they are relatively prime. Table 1 shows parameters of NTRU251.

Key generation: The private key is chosen to be of the form $\mathbf{f} = 1 + p * F$ for randomly selected $F \in D_F$ in such a way that $\mathbf{f}$ is invertible modulo q. The inverse will be denoted by $\mathbf{f}_q$. The public key is set to

$$\mathbf{h} \equiv p * \mathbf{f}_q * \mathbf{g} \pmod{q}. \tag{1}$$

using a random polynomial $g \in D_g$.

Encryption: To encrypt a message $\mathbf{m}$ which is a binary polynomial in $\mathcal{R}$, we choose a random $\mathbf{r} \in D_r$ and compute the ciphertext

$$\mathbf{e} = \mathbf{r} * \mathbf{h} + \mathbf{m} \pmod{q}. \tag{2}$$

2.2. **CS Attack on NTRU.** We describe Coppersmith and Shamir's lattice attack in view point of the message recovery attack on NTRUEncrypt. We shall not use the balancing constant λ, because it is not essential and is 1 in almost all cases.

The NTRU lattice L is the lattice of dimension $(2N + 1)$ generated by the row vectors of a matrix of the following form, where $\mathbf{h} = (h_0, \cdots, h_{N-1})$ is the public

key, q is the big modulus, $\mathbf{e} = (e_0, \cdots, e_{N-1})$ is a ciphertext corresponding to a message $\mathbf{m}$, and c is a dummy nonzero constant:

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{N-1} & 0 \\ 0 & 1 & \cdots & 0 & h_{N-1} & h_0 & \cdots & h_{N-2} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & h_1 & h_2 & \cdots & h_0 & 0 \\ 0 & 0 & \cdots & 0 & q & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & q & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & q & 0 \\ 0 & 0 & \cdots & 0 & e_0 & e_1 & \cdots & e_{N-1} & c \end{pmatrix}. \tag{3}$$

The lattice L contains the relatively short vector $\mathbf{v} = (-\mathbf{r}, \mathbf{m}, c)$, which is with high probability the shortest vector of L . An attacker uses lattice reduction algorithms to find $\mathbf{v}$ from L, then he can recover the message $\mathbf{m}$.

This attack works very well for the low degrees. However, numerous experimental results says that it is exponential with the degree N if we use the existing reduction algorithms[6]. To our knowledge there are no experimental results which break NTRUEncrypt of degree greater than 150 to date. For the present it is accepted that better lattice reduction algorithms are required in order to break NTRUEncrypt with the previous lattice attacks[10].

3. The New Lattice Attack

3.1. **Construction of New Lattices.** We first review the definition of intersection of lattices. Although it can be defined in more general lattices, We consider only the lattices of full rank in $\mathbb{Z}^n$ for some integer n.

Let $\mathcal{L}$ and $\mathcal{M}$ be lattices in $\mathbb{Z}^n$. Consider the intersection of $\mathcal{L}$ and $\mathcal{M}$:

$$\mathcal{L} \cap \mathcal{M} := \{\mathbf{v} | \mathbf{v} \in \mathcal{L} \text{ and } \mathbf{v} \in \mathcal{M}\}.$$

Then it is also a lattice of rank n, and it can be computed within $O(n^3)$ computations (if we ignore the bit size of the elements in the lattices)[1].

Now we define basic lattices for the attack. We use notations $\mathbf{h}, \mathbf{m}, \mathbf{e}, q$ and d_r the same as section 2. Let $\mathbf{a}$ be $\mathbf{e} - \mathbf{m}$ and I be a set $\{0, \cdots, N-1\}$.

For each $j \in I$, define $\mathcal{L}_j$ as a lattice of degree $n = (N+2)$ generated by row vectors of the following matrix:

$$M_j = \begin{pmatrix} 1 & 0 & \cdots & 0 & h_{0j} & 1 \\ 0 & 1 & \cdots & 0 & h_{1j} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & h_{(N-1)j} & 1 \\ 0 & 0 & \cdots & 0 & q & 0 \\ 0 & 0 & \cdots & 0 & a_j & d_r \end{pmatrix},$$

where $h_{ij} = h_{(-i+j) \bmod N}$ and a_j is j-th coefficient of $\mathbf{a}$.

Suppose $\mathcal{L}_j = \mathcal{L}_{j+k}$ for some j and $k \geq 1$. Then $\mathbf{h} = \mathbf{h} * x^k \bmod (x^N - 1)$. Since N is prime, this is possible only when $h_i = h_0$ for all $1 \leq i \leq N-1$. We can easily verified that such an h cannot be a public key of NTRUEncrypt. Thus, $\mathcal{L}_j$ are different lattices each other for all $j \in I$.

For any non-empty subset J of I, let's define a lattice $\mathcal{L}_J$ as followings:

$$\mathcal{L}_J = \bigcap_{j \in J} \mathcal{L}_j.$$

Then $\mathcal{L}_J$ has the following property.

Proposition 1. *Let J_t be a set of $t (1 \leq t \leq N)$ elements in I and q be a prime. Then,*

$$Det(\mathcal{L}_{J_t}) = d_r q^t,$$

where $Det(\mathcal{L}_{J_t})$ is the determinant of the lattice $\mathcal{L}_{J_t}$.

The proof of proposition 1 is given in the appendix.

If we intersect lattices more and more, the determinant of the resulting lattice will increase. Proposition 1 shows that this intuitional argument holds true very well for our special lattices.

3.2. **Heuristic Exposition of a New Attack.** Now we explain our attack heuristically. It is easily shown that for all $j \in I$ $\mathcal{L}_j$ contains $(\mathbf{r}, 0, 0)$, and thus $\mathcal{L}_{J_t}$ contains $(\mathbf{r}, 0, 0)$ for any $J_t \subset I$. The lengths of basis of $\mathcal{L}_{J_t}$ will be usually longer than those of $\mathcal{L}_{J_s}$ if $t > s$, because the determinant of the former is larger than that of the latter by proposition 1. Since the length of $(\mathbf{r}, 0, 0)$ is fixed as $\sqrt{d_r}$, if we increase t more and more then $(\mathbf{r}, 0, 0)$ can be a shortest vector of $\mathcal{L}_{J_t}$, and we may expect that for sufficiently large t the lattice reduction algorithms can produce $(\mathbf{r}, 0, 0)$ as one of the vectors of output basis.

The main drawback of this scenario is that we should predict m_j in advance in order to construct $\mathcal{L}_j$. However, if we can find $(\mathbf{r}, 0, 0)$ with $\mathcal{L}_{J_t}$ for relatively small t as compared with N, this attack will be more efficient than the brute force attack.

The value t, the number of bits needed to predict, depends on the efficiency of the reduction algorithms used in the attack. Thus we cannot estimate t generally. However, we could verified through the various experiments that t is relatively uniform for a fixed parameter and a fixed reduction algorithm, and that it is so small that the total complexity of the attack is less than the trivial brute force attack.

3.3. **Experimental Results.** First we made experiments for the purpose of estimating the breaking times of NTRU251.

We selected seven integers $67, 107, 139, 167, 191, 221$ and 251 for N, some of which were in the early version of standards. For each N we set q and d_r such that q/N and d_r/N are as equal as possible to those of NTRU251. Then we found out t by the following algorithm.

(1) For a fixed N, q and d_r, give values randomly to $F, \mathbf{g}, \mathbf{r}, \mathbf{m}$ and calculate all the other polynomials from them.

(2) Select an initial value appropriately as a candidate of t.
(3) Construct $\mathcal{L}_{J_t}$ and obtain a reduced basis with a lattice reduction algorithm.
(4) If the reduced basis contains $(\mathbf{r}, 0, 0)$ then output t and stop. Otherwise, increase t and execute from step 3.

We implemented the above algorithm on a Pentium VI 2.04 GHz PC. We used the BKZ-LLL algorithm[12] of NTL package[11] for the lattice reduction, and we set the LLL constant $\delta = 0.99$ and block size $k = 20$. We tested 10 times per a parameter set, changing the values of each parameter randomly. We counted t only when we succeeded in all 10 cases. Table 2 shows the results.

TABLE 2. Results of the first experiment

N	67	107	139	167	191	221	251
q	61	101	131	157	181	211	239
d_r	19	31	40	48	55	63	72
t	24	49	75	102	124	151	177
T_{int}(sec)	5.2	73.8	352	1117.3	2499.7	5872	12287.6
T_{red}(sec)	1.9	17.9	57.9	144.1	195.4	317.2	460.8
T_{one}(sec)	7.1	91.7	409.9	1261.4	2695.1	6189.2	12748.4
T_{tot}(sec)	$2^{26.8}$	$2^{55.5}$	$2^{83.3}$	$2^{110.6}$	$2^{132.4}$	$2^{158.3}$	$2^{183.7}$

In table 2 T_{int} is the average time for constructing the lattice $\mathcal{L}_{J_t}$ and T_{red} is the average time for the reduction algorithm to find the target vector. T_{one} is the sum of $T_{int} + T_{red}$ which will be the breaking time of the attack in case we predict all t bits of messages correctly. Finally T_{tot} is the total time for the attack to succeed estimated by the following method:

$$\begin{aligned} T_{tot} &= T_{one} \times \text{the number of prediction} \\ &= T_{one} \times \sum_{i=0}^{d_r} {}_tC_i. \end{aligned}$$

Experimental results says that with the high probability we can break NTRU251 on a single PC within about 2^{184} seconds. Because we can execute the attack algorithms in parallel, the breaking times are reduced in proportion to the amount of computational power.

The complexity of the brute force attack for NTRU251 is about 2^{213} encryption. Thus our attack is more efficient than it. Although meet-in-the-middle attack needs less computations than our attack[13], it needs impractically large memories.

The complexity of the attack seems to rely mainly on t, and t relies on the efficiency of lattice reduction algorithms. If we restrict the reduction algorithms to BKZ-LLL, which is the most efficient algorithm in view point of approximation factor to our knowledge, we can expect that if block size k increases then t decreases. However, if k increases then the reduction time also increases. Thus we cannot say definitely what is better.

Second experiment was made to observe the relations between the complexity of the attack and the block size of BKZ-LLL.

We selected more integers for N, and prepared a set of parameters for each N. Then we found out t as we increased the block size k. The results are shown in Table 3.

TABLE 3. t according to the block size k

N	101	111	121	131	141	151	161	171	181	191	201	211	221	231	241	251
$k = 5$	42	46	55	62	75	87	98	106	111	123	134	142	151	160	173	178
$k = 10$	40	47	56	58	73	79	88	98	108	117	127	135	145	154	166	175
$k = 15$	41	47	55	62	69	77	82	87	100	114	122	127	135	143	153	165
$k = 20$	40	45	54	64	71	78	82	88	97	108	117	127	138	149	159	166
$k = 25$	38	46	51	56	61	72	79	89	96	104	119	120	136	145	154	167

We verified that t tends to decrease as k increases. However, if k is greater than 20, then we need so much times to reduce the basis that the total complexity of the attack increases reversely. If $k \geq 30$, the breaking times are too long to complete the experiments.

Table 4 shows the lattice reduction time for $k = 15, 20$ and 25.

TABLE 4. Lattice reduction times(seconds) according to the block size k

N	111	131	151	171	191	211	231	251
$k = 10$	6	8	17	21	33	48	61	84
$k = 15$	6	14	29	42	66	81	114	138
$k = 20$	15	31	69	143	216	254	332	522
$k = 25$	196	2,294	16,254	42,387	66,025	137,886	187,293	258,725

From the experiments we can conclude that $k = 20$ is the best block size to break NTRUEncrypt in our attack.

4. CONCLUSION

In this article we have presented a new lattice attack on NTRU cryptosystem by using the intersection of lattices, and presented an upper bound of the breaking time for NTRU251 with our attack.

Although our attack is more efficient than the trivial brute force attack and has some merits compared with the previous lattice attacks, it is not powerful enough to threaten the practically used NTRU cryptosystems. However, we expect that our approach can be used to reveal the unknown structure of NTRU and to estimate the security of it more precisely.

References

[1] H. Cohen, A Course in Computational Algebraic Number Theory, Springer-Verlag, 1993.

[2] D. Coppersmith, A. Shamir, Lattice Attacks on NTRU, Advances in Cryptology - EUROCRYPT '97, LNCS 1233, Springer-Verlag, 1997.

[3] C. Gentry, Key Recovery and Message Attacks on NTRU-Composite, Advances in Cryptology - EUROCRYPT 2001, LNCS 2045, Springer-Verlag, 2001.

[4] J. Hoffstein, J. Pipher, and J. H. Silverman, NTRU: A ring-based public key cryptosystem. ANTS III, LNCS 1423, Springer-Verlag, 1998.

[5] J. Hoffstein and J.H. Silverman, Optimizations for NTRU. In Public-Key Cryptogrphy and Computational Number Theory. DeGruyter, 2002.

[6] J. Hoffstein, J.H. Silverman, and W. Whyte, Estimated Breaking Times for NTRU Lattices, Technical Report #12(Version 2), NTRU Cryptosystems.

[7] IEEE Standard P1363.1/D4, Standard specifications for public key cryptography : Techniques based on hard problems over lattices, IEEE. Available from `http://grouper.ieee.org/group/1363`.

[8] A.K. Lenstra, H.W. Lenstra Jr,. L. Lovśz, Factoring Polynomials with Rational Coefficients, Mathematische Ann. 261(1982), 513-534.

[9] A. May and J.H. Silverman, Dimension Reduction Methods for Convolution Modular Lattices, CaLC 2001, LNCS 2146, Springer-Verlag, 2001.

[10] P.Q. Nguyen, J. Stern, The Two Faces of Lattices in Cryptology, CaLC 2001, LNCS 2146, Springer-Verlag, 2001.

[11] NTL - A Number Theory Library, Victor Shoup, available at `http://shoup.net/ntl`.

[12] C.P. Schnorr, A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms, Theoretical Computer Science 53, 201-224, 1987.

[13] N. Howgrave-Graham, J.H. Silverman, W. Whyte, A Meet-In-The-Middle Attack on an NTRU Private Key, Technical Report #4(Version 2), NTRU Cryptosystems.

[14] J.H. Silverman, Dimension-Reduced Lattices, Zero-Forced Lattices, and the NTRU Public Key Cryptosystem, Technical Report #13(Version 1), NTRU Cryptosystems.

Appendix A. Proof of proposition 1

Proof. We shall denote J_t, $\mathcal{L}_{J_t}$ and d_r by J, $\mathcal{L}_J$ and d respectively for the notational convenience. $\mathcal{L}_J$ can be represented as follows:

$$\begin{aligned}\mathcal{L}_J = \{(v_0,\cdots,v_{N+1}) | \exists s \in \mathbb{Z} \text{ s.t. } & \sum_{i=0}^{N-1} v_i + sd = v_{N+1}, \\ & \sum_{i=0}^{N-1} v_i h_{ij} + sa_j = v_N (\text{mod } q) \text{ for all } j \in J\}.\end{aligned} \tag{4}$$

Now consider the following linear map

$$\begin{aligned}\phi : \mathcal{L}_J &\rightarrow \mathbb{Z}^{N+2} \\ (v_0,\cdots,v_{N+1}) &\mapsto (v_0,\cdots,v_N, v_{N+1} - \sum_{i=0}^{N-1} v_i).\end{aligned}$$

Let the image of ϕ be L. Then L is also a lattice of rank $N+2$ and is represented by

$$\begin{aligned} L &= \{(v_0,\cdots,v_N,sd)|\exists s\in\mathbb{Z} \text{ s.t. } \sum_{i=0}^{N-1} v_ih_{ij}+sa_j = v_N(\text{mod } q) \text{ for all } j\in J\} \\ &= \{(v_0,\cdots,v_N,d\cdot v_{N+1})|\sum_{i=0}^{N-1} v_ih_{ij}+v_{N+1}a_j = v_N(\text{mod } q) \text{ for all } j\in J\}. \end{aligned}$$

Since the determinant of the matrix representing the linear map ϕ is 1, $\mathcal{L}_J$ and L have the same determinant.

Consider another lattice L' represented by

$$L' = \{(v_0,\cdots,v_{N+1})|\sum_{i=0}^{N-1} v_ih_{ij}+v_{N+1}a_j = v_N(\text{mod } q) \text{ for all } j\in J\}.$$

Then $\text{Disc}(L) = d\times \text{ Disc}(L')$. Thus it is enough to show that $\text{Disc}(L') = q^t$ in order to prove the proposition.

Let L'' be a subspace of $\mathbb{Z}_q^{N+2}$ represented by

$$L'' = \{(v_0,\cdots,v_{N+1})\in\mathbb{Z}_q^{N+2}|\sum_{i=0}^{N-1} v_ih_{ij}+v_{N+1}e_j = v_N \text{ for all } j\in J\}.$$

Since L'' is a solution space of t linear equations, the rank of L'' is $N+2-t$. Thus L'' is generated by row vectors of an $(N+2-t)\times(N+2)$ matrix M. Since q is prime, $\mathbb{Z}_q$ is a field. Thus we can make M as the row echelon form as follows:

$$M = \begin{pmatrix} 1 & * & 0 & * & 0 & * & \cdots & 0 & * \\ 0 & 0 & 1 & * & 0 & * & \cdots & 0 & * \\ 0 & 0 & 0 & 0 & 1 & * & \cdots & 0 & * \\ & \ddots & & & \ddots & & \ddots & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & * \end{pmatrix}.$$

Now L' is generated by the row vectors of the following matrix:

$$M' = \begin{pmatrix} M \\ q\cdot Id \end{pmatrix} = \begin{pmatrix} 1 & * & 0 & * & 0 & * & \cdots & 0 & * \\ 0 & 0 & 1 & * & 0 & * & \cdots & 0 & * \\ 0 & 0 & 0 & 0 & 1 & * & \cdots & 0 & * \\ & \ddots & & & \ddots & & \ddots & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & * \\ q & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & q & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ & & & & \ddots & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & q \end{pmatrix}.$$

In each row of the matrix M the first nonzero element is 1. Let's call it the leading 1 and the column which contains leading 1 the leading column. There are exactly

$(N+2-t)$ leading columns in M. Finally let's call $(N+2-t+j)$-th row of M' a leading row if j-th column of M is a leading column.

The leading rows of M' can be generated by rows which are not leading rows. Thus we can remove the leading rows of M' in order to generate the lattice L'. Thus L' is indeed generated by the following $(N+2)\times(N+2)$ matrix:

$$M'' = \begin{pmatrix} 1 & * & 0 & * & 0 & * & \cdots & 0 & * \\ 0 & 0 & 1 & * & 0 & * & \cdots & 0 & * \\ 0 & 0 & 0 & 0 & 1 & * & \cdots & 0 & * \\ & \ddots & & & \ddots & & \ddots & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & * \\ 0 & q & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & q & 0 & 0 & \cdots & 0 & 0 \\ & & & & \ddots & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & q \end{pmatrix}.$$

It is easy to know that the determinant of M'' is q^t. Therefore $\mathrm{Disc}(L') = q^t$, which completes the proof.

□

National Security Research Institute, 161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, Korea

E-mail address: dwh@etri.re.kr