A New Attack on NTRU Public Key Cryptosystem Depend on Using Public Key and Public Information

Dr. Abdul Monem S.Rahma* & Dr. Qasim Mohammed Hussein**

Received on:18/3/2009
Accepted on:2/7/2009

Abstract

This paper proposed a new method to attack the NTRU cryptosystem [Hof00a, Hof98b]. It tried to exploit the public information about the parameters of NTRU cryptosystem and using the public key, to recover the private keys without delving in the detail of encryption and decryption. It depended on extant knowledge about the number of 1's and -1's in the private keys coefficients, f and g, which are used in public key generation. It tried to make use of the polynomial that has private key coefficients properties. The proposed attack was able to recover a unique polynomial that represented the private key f and corresponding to private key g, or their rotations. But, this attack remains expensive in time and depends on the way of how to start in choice the permutation

Keywords: cryptosystem , NTRU , Public Key , Attack

طريقة مهاجمة جديدة لنظام التشفير المعلن (NTRU) بالاعتماد على المفتاح المعلن والمعلومات المعلنة

الخلاصة

يتناول البحث طريقة مقترحة جديدة لمهاجمة نظام التشفير المعلن نترو (NTRU) من خلال استثمار المعلومات المعلنة عن معلمات النظام ، مع استخدام المفتاح العام المعلن في ايجاد ومعرفة المفتاحين الخاصين المستخدمان في توليد ذلك المفتاح العام دون الخوض في تفاصيل التشفير والحل. تعتمد الطريقة المقترحة على معرفة عدد الواحدات والاصفار في قيم معاملات المفتاحين الخاصين f و g . تستفاد الطريقة المقترحة من اجراء عمليات التباديل (Permutations) على متعددة حدود مختارة (f ') كمرشحة للمفتاح الخاص f والتي تحتوي عدد من الواحدات والاصفار مساوية لعددها في المفتاح الخاص f. تمكنت الطريقة المقترحة من اعطاء متعددتي حدود وحيدتين تمثلان المفتاح الخاص f ومناظره المفتاح الخاص g او احد احتمالات تدويرهما اعتمادا على طريقة البدء في اختيار التباديل للمتعددة الحدود ' f. على الرغم من نجاح هذه الطريقة الا انها تستغرق و قت كبير يعتمد على كيفية تحديد طريقة اختيار بداية اجراء التبادي.

*** Computer Science Department, University of Technology / Baghdad**
**** Computer Science Department, College of Mohammed Kaznzan/ Baghdad**

https://doi.org/10.30684/etj.28.6.3
2412-0758/University of Technology-Iraq, Baghdad, Iraq

Introduction

NTRU is a public key cryptosystem proposed by J. Hoffstein, J. Pipher and J. Silverman [Hof00a, Hof98b]. The first version of the NTRU encryption system was presented at the Crypto ’96 conference. NTRU is based on

polynomial additions and multiplications in the ring of truncated polynomials $\mathbf{Z[x]/(x^N-1)}$ and uses reduction modulo two integer **p** and **q**, where the parameter **N** determines the security of system.

The security of the NTRU cryptosystem is based on the hardness of finding the shortest vector and closest vector problem in lattice of high dimension.

NTRU is significantly faster than other public key systems, such as RSA and ECC. Also the key generation is fast and easy with reasonable small key size. These features make NTRU cryptosystem suitable for user applications where the encryption/decryption speed plays an important role, such as wireless communications. Moreover, the computations are a very simple and low memory requirement, which makes it suitable for devices with restricted resources, such as smart cards and other embedded system. Moreover, NTRU cryptosystem has the property of working with quite small numbers, making it suitable for 8-bit processors.[Ntr05b]

This paper presents a new attack against the NTRU cryptosystem. The method exploited the public information about the parameters of NTRU cryptosystem and public key to recover the private keys without delving in the detail of encryption and decryption. It depended on knowing information about the number of 1’s and -1’s in the private keys f and g that are used in public key generation. Also it tried to make use of polynomial coefficients properties that resulted from convolution multiplication of public key and a polynomial , that contains a certain numbers of **1, 0,** and **-1** coefficients that are equal to the numbers of **1 , 0,** and **-1** in private key **g** coefficients respectively.

1. Description of NTRU Public Key Cryptosystem:

1.1 NTRU Public Key Cryptosystem Parameters

The parameter must be selected in order to define the space in which operations are performed in it. The NTRU Public Key Cryptosystem is specified by the three parameters: **N, p, q** [Hof98b, Why04].

N **-** The polynomials in the truncated polynomial ring have degree ***N*-1**. It identifies the dimension of the convolution polynomial ring used. All the polynomial multiplications are reduced **modulo N**.

q **-** The coefficients of the truncated polynomials will be reduced **modulo *q***.

p **-** It is used for key generation and in decryption, the coefficients of the message are reduced **modulo *p***.

Additional public parameters are numbers, denoted $\mathbf{d_f}$, $\mathbf{d_g}$, $\mathbf{d_m}$ and $\mathbf{d_r}$. These specify the space of allowable private keys **f** and g, the allowable messages, and the form of the random polynomial r used in encryption. In order to ensure NTRU cryptosystem security, it is essential that **p** and **q** have no common factors, **gcd(p,q)=1**.

1.2 NTRU Public Key Generation

To create a public key, the user must [Hof98b]:

- Chooses a secrete key, two "small" polynomials **f** and **g** in the ring of truncated polynomials **R**. the polynomial **f** should be invertible in $\mathbf{R_p}$ and $\mathbf{R_q}$. The polynomial **f** has $\mathbf{d_f}$ coefficients equal to **+1**, d_f **–1** coefficients equal to **-1**, and the rest equal to **0**. The polynomial **g** has d_g coefficients equal to **+1,** $\mathbf{d_g}$ coefficients equal to **-1**, and the rest equal to **0**. The coefficients will be randomly distributed **modulo q**.
- Compute the inverse of **f modulo q** and the inverse of **f modulo p**. such that $\mathbf{f*f_q = 1}$ **(modulo q)** and $\mathbf{f*f_p = 1}$ **(modulo p)**.

Where * denote a polynomial multiplication in R, which is the cyclic convolution product of two polynomials.

Find the public key, **h**, $\mathbf{h= p*F_q*g}$ **(modulo q)** , the polynomials f and g are private, while the polynomial h is public. The parameters N, q, and p are also public.

1.3 NTRU Encryption

The encrypted message m is computed as [Hof98b]:

- The message must be converted into a polynomial **m** whose coefficients are chosen**(modulo *p*).**
- Select a random small polynomial, **r Є R**.
- Compute the cipher text , **e**, as a polynomial:

$$\mathbf{e = r*h + m}\ (\text{modulo } q)$$

1.4 NTRU Decryption

To recover the original message, the decryption procedure requires three steps [Hof98b]:

Use the private key **f** to compute the polynomial **a = f*e (modulo *q*).** Shift the coefficients of **a** to lie between **-*q*/2** and ***q*/2**. Computes the polynomial b = **a(modulo *p*)**.Finally use the private polynomial $\mathbf{f_p}$ to compute $\mathbf{c = f_p*b}$ **(modulo *p*)**.

The polynomial **c** will be the original message **m**.

1.5 NTRU cryptosystem parameters Selection

The parameters of the NTRU scheme must be selected to balance a number of competing considerations: minimizing decryption failure, eliminating the possibility of brute force searches, maintaining speed and efficiency [Pip02].To get a good security level for the NTRU cryptosystem, the author of NTRU cryptosystem gave different sets of parameters given in [Hof98b, How03c, Sil98]

2. Attacks on NTRU cryptosystem:

A cryptographic attacking does not mean "total break of the target system ". It may apply to recover a small amount of information about the parameter set, encrypted message, or keying material. Additionally, an attack may simply be an analysis of approaches to attacking a system [Ntr05b]. There are many paper concerned with the attack methods against NTRU cryptosystem presented by the cryptographer.

First the brute force attack, which can be eased by the meet-in-the-middle principle [How03c, How07], may be used against the private key or against a single message. However, for a suitable choice of parameters this attack will not succeed in a reasonable time. The key and message spaces must therefore be chosen to give an adequate level of combinatorial security.

The multiple transmission attack can be used against NTRU, which will provide the content of a message that has been transmitted several times. Thus multiple transmissions are not advised. It is also one of the reasons why NTRU recommends a preprocessing scheme [Lin06, Hof98a].

In addition, NTRU cryptosystem is vulnerable to attacks based on decryption failures [Mes03, Sil03, Yu04]. In these attacks, an attacker attempts to create validly encrypted messages $\boldsymbol{e = r + h * m}$ such that the width of $\boldsymbol{p * r * g + f * m}$ is greater than $\boldsymbol{q}$. Decryption failures leak information and can allow the recovery of the private key ***f.*** The padding scheme should ensure that ***r*** and ***m*** are drawn uniformly at random from some set, and that they cannot be directly selected by the sender [How03a].

In reaction attack, there is no chance to succeed against NTRU if any given public/private key pair is used for only one message, or at most a few messages. This attack can only succeed if a single private key is used for the decryption of a large number of messages [Hof00c, Hof00b]. This is a situation that will rarely arise in an NTRU based system, since the ease of NTRU key generation key means that keys will be changed frequently [Hof00d];also, it is possible to break the NTRU system using adaptive chosen-ciphertext attack [Jau00, Mes05]. Therefore, NTRU cryptosystem should only be used with a padding scheme.

Finally, several attacks make use of the LLL algorithm of Lenstra-Lenstra- Lovasz [Cop97] which produces a reduced basis for a given lattice. They can either recover the secret key from the public key or decipher one given message. However the time required is exponential in the degree of the polynomials. For most lattices, it is indeed very difficult to find extremely short vectors. Thus for suitably large degrees, this attack is expected to fail and does fail in practice.

3. The Proposed Attacking On NTRU Public Key Cryptosystem

The paper has focused on the problem of recovering the private keys, the polynomials **f** and **g**, from knowing the public key and public information of NTRU cryptosystem. The project tried to exploit the public information about the parameters of NTRU cryptosystem and the extent of knowledge about the number of **1's** and **-1's** in the private keys f and g coefficients, which are used in public key generation. And then it tried to make use of the definition of the public key **h**, **f*h = g modulo q**, to recover polynomials that have private key coefficient properties. Then these polynomials will be tested to find which of them would represent the private keys, without delving in the detail of encryption and decryption.

3.1 General Description of the Proposed Attack

In the NTRU cryptosystem, the private keys contain a certain number of coefficients whose values are**1**, **–1**, or **0**. The proposed attack tried to exploit these values to recover the private keys, **f** and **g**, from knowing the public information and public key which generated by those private keys.

The attack started by taking a polynomial with order **N** as a candidate private key **f'**. This polynomial should contain a certain number of coefficients **0**, **1**,and **-1** equal to the numbers of these coefficients in private key **f**, making permutation for this polynomial, performed a convolution multiplication with public key for each possibility of permutation to generate a polynomial **g'**, where the interest is to obtain a polynomial **g'** that has two properties:

First: The sum of its coefficients is equal to a certain summation. This depends on the numbers of coefficients with value **1** or **-1**.

Second: It contains a certain number of **1, 0**, and **-1** coefficient is equal to the number of **1, 0**, and **-1** private key g coefficients which is equal to $\mathbf{d_g}$.

This attack is done without delving in the detail of encryption and decryption processes.

3.2 The Principles of the Proposed Attack

The proposed attack is based on the following principles:

Firstly, in brute force attack for the private key **f** with order **N**, the number of probabilities of the permutation of its coefficients is $\mathbf{3^N}$, since the value of each coefficient in **f** is **-1**, **0**, or **1**. But there are reasonable number of the ordering that does not contain a certain number of **1's** and **-1's** which is equal to $\mathbf{d_f}$ and $\mathbf{d_f}$ **-1** respectively. So we can neglect these orders from our computations where this attack is used.

Secondly, multiplying the polynomial, public key **h**, with any other polynomial whose coefficients in **{-1, 0, 1}** but it was not used, or its rotations, in generating this public key **h modulo q**, will produce the polynomial **X**. The coefficients of **X** contain different values distributed on **q** not belonging to **{-1,0,1}** only.

Finally, For a polynomial **f** with coefficient:

$\mathbf{f_0 + f_1x + \ . \ . \ . \ + f_{N-1}x^{N-1}}$ Î **R**.

If we have **f * h ≡ p g modulo q**;

Then $\mathbf{f{*}x^i\ {*}h \equiv p\ g\ {*}\ x^i}$ **modulo q**;

where the polynomial $\mathbf{f{*}x^i}$ **in R** is the rotation of **f**. So there are several solutions to represent all the rotations, equal to **N**, of f and corresponding **g**. The finding of any one of rotations will fulfill the other rotations of that case. If one has found one rotation of **f**, hence he has only **N** possibilities left for the private key **f**.

3.3 The Assumptions of the Proposed Attack

The idea of this attack depended on the following assumptions:

1. Interested in a polynomial **f¢** in $\mathbf{Z[x]/\ (x^N\text{-}1)}$ that satisfies the following:

1.1. $\mathbf{p^{-1}f}$ **¢*h = g¢modulo q = { -1 ,0,1}**….(1) Where $\mathbf{p^{-1}}$ is the inverse of the integer **p modulo q**.

1.2. The polynomial **f¢** coefficients contains a number of **1's** equal to $\mathbf{d_f}$, a number of **1's** in private key **f**, number of **-1's** equal to $\mathbf{d_f}$ **-1** and the rest is **0**.

2. Making use of the property of the **g¢** coefficients, this must be:

2.1. Their coefficients values are in **{-1, 0,1}.**

2.2. It has $\mathbf{d_g}$ coefficients equal to **1,** $\mathbf{d_g}$ coefficients equal to **-1**, and the rest is **0**.

These assumptions were used to reduce the possibilities in search space of **f¢**, by eliminating the cases of **f¢** that do not satisfy the above assumptions.

On the other hand, for the private key g that contains $\mathbf{d_g}$ **1's** and $\mathbf{d_g}$ **-1's** coefficients, the summation of **g's** coefficients can be found by the formula

$$\mathbf{sum\_g = dg.\ 1 + d_g.\ (q\text{-}1)} \quad .... (2)$$

Where $\mathbf{0 \le gi < q, i = 0 \ldots (N\text{-}1)}$.

There are **N** rotations for each case of **f¢** that yielded **g¢** rotation, which satisfy formula (2) and take the same **sum_g**. It may be given the same **sum_g** but may not satisfy formula (1) and the reverse is correct.

3.4 The Algorithm of the Proposed Attack

```
Input :  The public key ,h, the parameters p, q, and N. The
         integers $d_f$, $d_f$ –1, $d_g$, the number of 1's and -1's in
         polynomial f, the number of 1's in g, respectively.
Output: The private keys f and g.
step 1.  Find the summation of g coefficients, sum_g, using
         formula (2).
step 2.  Compute $h = p^{-1} * h$ modulo q.
step 3.  Loop.
step 4.  Take a possibility of f¢ permutation that its
         coefficients satisfy: Number of 1's equal to $d_f$ and
         number of -1's equal to $d_f$ -1.
step 5.  Find polynomial g¢ by performing the convolution
         multiplication between f¢ and h modulo q.
step 6.  Find the summation of the coefficients of g¢, sum_g¢,
         using formula (2). If sum_g not equal to sum_g¢, goto
         step3.
step 7.  Count the frequency of 1's and of -1's in g¢ coefficients.
         If they are not equal to dg, goto step3.
step 8.  f=f¢, g = g¢ , after replace every coefficients equal to
         (q-1) in g¢ to -1. And exit.
```

4. The Probabilities and Performance Analysis of the Proposed Attack

Suppose the polynomial **f** of order **N** has $\mathbf{n_1}$ identical to coefficients of value **1**, $\mathbf{n_2}$ identical to coefficients of value **-1**, and $\mathbf{n_3}$ identical to coefficients of value **0**, the permutation with repetition and unordered selection are allowed to the number nr of possibilities of **f**. The $\mathbf{n_r}$ can be calculated as:

$$\mathbf{N!}$$

$$\mathbf{n_r} = \text{¾ ¾ ¾ ¾ ¾}$$

$$\mathbf{n_1!\ .\ n_2!\ .\ n_3!}$$

Since the rotations of each case of **f¢** are distributed in the permutation possibilities, the probability of finding any one of these rotations will be considered less than the total number of possibilities as illustrated in table **1**, where **N=11, N=13, N=15, and N=17**.

Experimentally, it was found that the first solution, private key or its rotation appeared in less than the first **0.1** of permutation possibilities. For example, we experimented many polynomials for **N=13** to find the distribution of rotations of **f¢** , which was yielded the private key or its rotation, in the total possibilities of permutation.

In the worst case, the first rotation of **f¢** which represents the rotation of private key f appeared after checking 6015 possibilities from the permutation. The best case of finding the first rotation was obtained after 139 possibilities.

Also, to make this algorithm suitable for any integer value of P, one can eliminate the second step of the above algorithm, and modify the formula (2) to:

$Sum\_g = d_g.p + d_g.(q-p)$ …..(3)

Where $0 \le g_i < q$.

The values of the sum_g for different numbers of parameters were found in the table (2) by using formula 2 or 3.

This proposed method generated a unique polynomial that satisfied the conditions of the algorithm. This polynomial may be the exact private key f, or its rotation, depending on the way of start in selecting the permutation of the candidate private key. So, NTRU, like most public key cryptosystem, should not be used as originally described.

The attack is still very expensive in time space. It can be used in combination with other attack methods to get an efficient attack method against NTRU public key cryptosystem.

5. Experimental results

Here is an example that explains our attack for recovering the private keys. Suppose,

$N = 13, q=32, p=3, d_f = 4, d_g = 3$, are taken and the private keys chosen are:

$$\mathbf{f = -1 + x^3 - x^4 + x^5 + x^7 - x^{10} + x^{12}}$$

$$\mathbf{g = 1 - x^2 + x^4 - x^5 + x^9 - x^{11}}$$

For simplicity, the polynomials are represented in vector form. So instead of writing

$$\mathbf{f = -1 + x^3 - x^4 + x^5 + x^7 - x^{10} + x^{12}}$$

It is written as

f = [-1 0 0 1 -1 1 0 1 0 0 -1 0 1]

To compute the public key h, f_q is found and the private key f modulo q is inversed.

$$\mathbf{fq = 24 + 13x + 16x^2 + 10x^3 + 11x^4 + 19x^5 + 28x^6 + 18\ x^7 + 10x^8 + 17x^9 + x^{10} + 6x^{11} + 20\ x^{12}.}$$

The public key

$h = pf_q{*}g$

$$\mathbf{= 28 + 22x + 10x^2 + 30x^3 + 7x^4 + 23x^5 + x^6 + 10x^7 + 6x^8 + 11x^9 + 11x^{10} + 9x^{11} + 24x^{12}.}$$

To find the private keys using the above algorithm and make use of public information about system parameter **(N, q, p, d_f, and d_g)** :

First, find the summation of g coefficients, **sum_g**, using formula (2). The value of **sum_g = 96**.Then, remove the effect of p by **$h = p^{-1}h$**.

h is taken as input to the algorithm, and **f¢** is supposed as **[1 0 -1 1 0 0 1 0 -1 0 -1 0 1]**. After executing the algorithm of the proposed attack, the following results are obtained:

The number of possibilities that satisfied step 4 in the algorithm 1, the polynomials that contain d_f coefficients equal to **1** and **d_f-1** coefficients equal to **-1** and the rest **0**, were 60060,The numbers of possibilities that satisfied step 6 in the algorithm, number of polynomials, that have coefficients summation equal to the result of formula 2, were 143. They included 11 polynomials as shown below, each one with its 13 rotations (all possibilities are shown in table 3).

1 f¢=-1 -1 0 0 0 0 1 1 -1 0 0 1 1
sum= 96 g'=0 23 4 2 20 1 1 0 14 7 2 0 22

2 f¢=-1 -1 0 0 1 0 -1 0 1 1 0 1 0
sum= 96 g'=7 3 16 4 0 0 16 3 4 1 20 1 21

3 f¢=-1 -1 0 1 1 0 0 -1 0 0 1 0 1
sum= 96 g'=7 3 16 3 1 0 15 5 4 0 20 1 21

4 f¢=-1 -1 1 1 -1 0 0 0 0 0 1 0 1
sum= 96 g'=3 7 0 30 0 9 8 2 7 2 10 11 7

5 f¢=-1 -1 1 1 1 0 0 -1 0 0 0 0 1
sum= 96 g'=0 30 7 10 1 8 3 9 12 6 2 8 0

6 f¢=-1 0 -1 0 0 0 0 1 1 -1 1 0 1
sum= 96 g'=0 29 10 9 0 9 1 9 13 6 2 7 1

7 f¢=-1 0 -1 0 0 1 -1 0 0 0 1 1 1
sum= 96 g'=7 2 18 3 0 0 15 5 4 0 20 0 22

8 f¢=-1 0 -1 1 0 0 1 1 0 -1 0 0 1
sum= 96 g'=0 10 0 9 14 4 3 7 1 1 27 10 10

9 f¢=-1 0 0 -1 1 1 0 0 1 0 -1 1 0
sum= 96 g'=15 5 4 1 19 0 23 5 2 19 3 0 0

10 f¢=-1 0 0 1 -1 1 0 1 0 0 -1 0 1
sum= 96 g'=1 0 31 0 1 31 0 0 0 1 0 31 0

11 f¢=-1 0 0 1 1 -1 1 0 0 -1 1 0 0
sum= 96 g'=7 3 15 5 0 0 15 3 7 0 18 3 20

The numbers of possibilities that satisfied step 7 in the algorithm, number of polynomials whose corresponding **g¢** coefficients contained $\mathbf{d_g}$ of **1** and $\mathbf{d_g}$ of **-1**, were 13 as shown below. They included **1** polynomial. This polynomials with its rotations (declared in table 4).

The final result was the polynomials:

f′=-1 0 0 1 -1 1 0 1 0 0 -1 0 1

g′=1 0 31 0 1 31 0 0 0 1 0 31 0

=1 0 -1 0 1 -1 0 0 0 1 0 -1 0

Each g coefficient has a value equal to 31, q-1, will be replaced with -1.

So, the private key f is

-1 0 0 1 -1 1 0 1 0 0 -1 0 1

And the private key g is

1 0 -1 0 1 -1 0 0 0 1 0 -1 0

We saw that there are only the private keys or their rotation will satisfy the conditions in our algorithm.

6. Conclusions

From the present work, the following conclusions are found.

1. The attack showed that it is possible to make use of public information about the structure of the private keys, such as the number of **1's** and **-1's** in private key and how to find these private keys. This attack is efficient when the number of **1's** and **-1's** in the private keys **f** and **g,** $\mathbf{d_f}$ and $\mathbf{d_f -1}$ respectively, are known to the attacker. So these numbers must be selected carefully to increase the number of permutations, which make the attack need more time to find the private keys.

2. The proposed attack was able to recover unique polynomials that represent the private key **f** and corresponding to private key *g*, or their rotations.

3. Since the rotations of each solution are distributed in the permutation possibilities, the probability of finding any one of these rotations will be considerably less than the total number of possibilities, Although this method generates a unique solution, it depends on the way of starting in selecting the permutation of the candidate private key, to find the private keys or their rotations, but it is still very expensive in time space. So this attack can be speed up by using parallel processing system to reduce the time to find the first rotation.

4. No practical effective attack method has been found which significantly impacts the core ideas in NTRU technology. The best current research affirms that NTRU makes it possible to have strong public key cryptography at a fraction of the time and processing power previously necessary.

8. References

[**1**] Hoffstein J., Silverman J., "*Optimizations for NTRU"*, proceeding of Public-Key Cryptography and Computational Number Theory, Warsaw, September , 2000

[**2**] Hoffstein J., Pipher J., Silverman J., "*NTRU: A Ring Based Public Key Cryptosystem"*, Algorithmic Number Theory (ANTS III), Portland, Lecture Notes in.

[**3**] NTRU Cryptosystems, Inc. "*Peer Review and Independent Scrutiny of the NTRUEncrypt Public Key Cryptosystem* ",35 Nagog Park, Acton, MA 01720, USA, 2005.

[**4**] Whyte W., "*Choosing NTRUEncryption Parameters*", NTRU cryptosystem, 2004.

[**5**] Bailey Daniel V. , Coffin Daniel, Elbirt Adam, Silverman Joseph H., and Adam D, "*NTRU in Constrained Devices*", , Computer Science Department, Brown University, 2001.

[**6**] Howgrave-Graham N., Silverman J., Whyte W., "*A meet-in-the-middle attack on an NTRU private key"*, NTRU Technical Report 004, version 2, 2003. Available from http://www.ntru.com/cryptolab/ tech_notes.htm#004. [**7**] Silverman J., "*Wraps, Gaps, and Lattice Constants"*. NTRU Technical Report #011, 2001.

[**8**] Howgrave-Graham N., "*A hybrid lattice-reduction and meet-in-the-middle attack against NTRU*", NTRU Cryptosystems, Inc., 2007.

[**9**] Hoffstein J., Silverman J., "*Implementation Notes for NTRU PKCS Multiple Transmissions*", NTRU Technical Report 06, 1998, Available http://www.ntru.com/cryptolab/tech_ notes.htm#006. Computer Science 1423, Springer-Verlag, Berlin. 1998.

[**10]** Linder Richard , "*Current Attack On NTRU*", master thesis in university of Darmstadt, 2006.

[**11**] Meskanen T., Renvall A., "*A Wrap Error Attack Against NTRUEncrypt*", University of Turku Technical Report TUCS 507. Available

[**12**] Silverman J, Whyte W., "*Estimating Decryption Failure Probabilities for NTRUEncrypt*, 2003. Available from http://www.ntru.com/cryptolab/ articles.htm

[**13]** Yu Weichi, Zhu Dake hixiong, "*Study on NTRU Decryption Failure*", Available yuweichi2004@yahoo.com.cn

[**14**] Hoffstein J., Silverman J., Whyte W., "*Estimating Breaking Times for NTRU Lattices*"

[**15**] Howgrave-Graham N., Nguyen P., Pointcheval D., Proos J., Silverman J., Singer A., Whyte W., "*The Impact of Decryption Failures on the Security of NTRU Encryption*", 2003. Available from http://www.ntru.com/cryptolab/articles.htm#00.
from Presented at WCC 2003. http://www.tucs.fi/Research/Series/techreports/
techrep.php

[**16**] Hoffstein J and Silverman J, "*Protecting NTRU Against Chosen ciphertext and Reaction Attacks*". NTRU Technical Report #016, 2000.

[**17**] Hoffstein J. and Silverman J., "*Reaction Attacks Against the NTRU Public Key Cryptosystem*". NTRU Technical Report #015.,2000.

[**18**] Jaulmes E. and Joux A., "*A chosen-ciphertext attack against NTRU"*, Advances in Cryptology - CRYPTO 2000, Lecture Notes in Computer Science, Springer-Verlag, to appear (August, 2000).

[**19**] Meskanen Tommi , "*On The NTRU Cryptosystem"*, University of Turku, Finland, 2005.

[**20**] Coppersmith D., Shamir A., "*Lattice attacks on NTRU*", in Proc. of EUROCRYPT 97, Lecture Notes in Computer Science, Springer-Verlag,

Table (1) [Hof98b, How03c, Sil98]

Security level	N	P	q	f		r	g
				d_f	d_f-1	d_r	d_g
Moderate	167	3	128	61	60	18	20
High	263	3	128	50	49	16	24
Highest	503	3	256	216	215	55	72

Table (2) The value of sum_g at different parameters values

Security level	N	Q	P	d_g	sum_g
Moderate	107	64	3	12	768
Standard	167	128	3	20	2560
High	263	128	3	24	3072
Highest	503	256	3	72	18432

Table (3) The appearance of the first rotation

Trial	N=11		N=13		N=15		N=17	
	First Solution	Percentage	First Solution	Percentage	First Solution	Percentage	First Solution	Percentage
1	679	0.0588	2408	0.040093	70598	0.111948	23085	0.0075
2	223	0.0193	5310	0.088412	19420	0.030795	78550	0.0256
3	538	0.0465	1739	0.028954	49450	0.078414	143419	0.0468
4	181	0.0156	139	0.002314	82	0.00013	324105	0.1058
5	217	0.0188	5812	0.10342	25454	0.040363	2643	0.0008
6	345	0.0298	3971	0.066117	11864	0.018813	237585	0.0775
7	234	0.0202	6015	0.10015	14300	0.022676	7558	0.0024
8	183	0.0158	4818	0.08022	32849	0.052089	150890	0.0492
9	821	0.07108	1399	0.023293	30310	0.048063	10476	0.0034
10	467	0.04043	5421	0.09026	46715	0.074077	38409	0.0125
Total	11550		60060		630630		3063060	

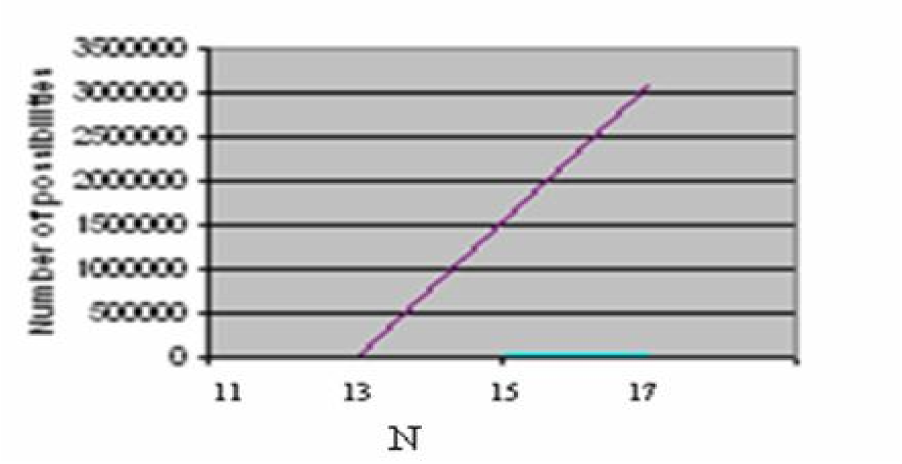

Figure (1) relation between N and number of possibilities