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 والعلومات المعلنة)

الخلاصة
ينتناول البحث طريقة مفترضة جديدة لـِمهاجمة نظام التشفير المعلن NTRU. من خلال استخدام المعلومات المعلنة عن معلمات النظام، مع استخدام المفتاح العام المعلن في إيجاد وتعريف المفتاحين الخاصين المستخدمين في توليد ذلك المفتاح العام دون الخوض في تفاصيل التشفير والحل. تتمحور الدراسة على معرفة عدد الوالدات والأصفر في قيم معاملات المفتاحين الخاصين مع f و g. تستفيد الدراسة المفتوحة من إجراء عمليات التبديل التي تحتوي عند (Permutations) على مذكرة حدود مختارة من الوالدات والأصفر. كمرشحة لـِمفتاح الخاص f والتي تحتوي عند (f') من الوالدات والأصفر. تمكنت الدراسة من أخذ f أو g أو أخذ f' أو g' من قيم مذكرة حدود مختارة. اعتماداً على طريقة البدء، في اختيار التبديل للمفتاحين الخاصين. على الرغم من نجاح هذه الدراسة، إلا أنها تستغرق وقت كبير وتمكنت على كيفية تحديد طريقة اختيار بداية أجزاء التبادي.

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

https://doi.org/10.30684/etj.28.6.2
2412-0758/University of Technology-Iraq, Baghdad, Iraq
This is an open access article under the CC BY 4.0 license http://creativecommons.org/licenses/by/4.0
Introduction

NTRU is a public key cryptosystem proposed by J. Hoffstein, J. Pipher and J. Silverman \cite{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 $\mathbb{Z}[x]/(x^N-1)$ 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.\cite{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: $N$, $p$, $q$ \cite{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 $d_t$, $d_p$, $d_m$ and $d_r$. 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 secret key, two "small" polynomials \( f \) and \( g \) in the ring of truncated polynomials \( R \); the polynomial \( f \) should be invertible in \( R_p \) and \( R_q \). The polynomial \( f \) has \( d_f \) 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 \( +1 \), \( d_g - 1 \) 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 \( f*f_q = 1 \) (modulo \( q \)) and \( f*f_p = 1 \) (modulo \( p \)).

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 \( f_p \) to compute \( c = f_p*b \) (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 $e = r + h \ast m$ such that the width of $p \ast r \ast g + f \ast m$ is greater than $q$. 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 $h$, $f \ast h = g \mod q$, 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, or -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's equal to d_f, a number of -1's in private key g, number of -1's equal to d_f and the rest is 0.

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 $3^N$, 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 $d_f$ and $d_f - 1$ 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:
$$f_0 + f_1 x + \ldots + f_{N-1} x^{N-1} \in \mathbb{R}.$$ If we have $f' * h \equiv p g \mod q$; Then $f' * h \equiv p g * x^i \mod q$; where the polynomial $f' * h$ 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 $\mathbb{Z}[x]/(x^N-1)$ that satisfies the following:
   1.1. $p^{-1} f' * h \equiv g' \mod q = \{-1,0,1\}$. Where $p^{-1}$ is the inverse of the integer $p$ modulo q.
   1.2. The polynomial f' coefficients contains a number of 1's equal to $d_f$, a number of 1's in private key f, number of -1's equal to $d_f - 1$ 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 \(d_g\) coefficients equal to 1, \(d_g\) 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 \(d_g\)'s and \(d_g\)'s coefficients, the summation of \(g^{'s}\) coefficients can be found by the formula

\[
\text{sum}_g = d_g \cdot 1 + d_g \cdot (q-1)
\]  

Where \(0 \leq g_i < q, i = 0 \ldots (N-1)\).

There are \(N\) rotations for each case of \(f'\) that yielded \(g^{'\prime}\) rotation, which satisfy formula (2) and take the same \(\text{sum}_g\). It may be given the same \(\text{sum}_g\) 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 \(d_0, d_g-1, d_g\), 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, \(\text{sum}_g\), using formula (2).

**step 2.** Compute \(h = p^{-1} \mod q\).

**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 \(g^{'\prime}\) by performing the convolution multiplication between \(f'\) and \(h \mod q\).

**step 6.** Find the summation of the coefficients of \(g^{'\prime}\), \(\text{sum}_g^{'\prime}\), using formula (2). If \(\text{sum}_g\) not equal to \(\text{sum}_g^{'\prime}\), goto step 3.

**step 7.** Count the frequency of 1's and -1's in \(g^{'\prime}\) coefficients. If they are not equal to \(d_g\), goto step 3.

**step 8.** \(f=f^{'\prime}, g = g^{'\prime}\), after replace every coefficients equal to \((q-1)\) in \(g^{'\prime}\) to -1. And exit.

4. The Probabilities and Performance Analysis of the Proposed Attack

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

\[
n_r = \frac{N!}{n_1! \cdot n_2! \cdot n_3!}
\]

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:

\[
\text{Sum}_g = d_g p + d_g (q-p) \quad \ldots\ldots(3)
\]

Where \( 0 \leq g < q \).

The values of the \( \text{sum}_g \) 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:

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

\[
g = 1 - x^2 + x^4 - x^5 + x^9 - x^{11}
\]

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

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

It is written as

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

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

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

The public key

\[
h = pf_q^*g
\]

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

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, \( \text{sum}_g \), using formula (2). The value of \( \text{sum}_g = 96 \). 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,
A New Attack on NTRU Public Key Cryptosystem Depend on Using Public Key and Public Information

each one with its 13 rotations (all possibilities are shown in table 3).

1. $f' = -1 \cdot 1 \cdot 0 \cdot 0 \cdot 0 \cdot 1 \cdot 1 \cdot 1 \cdot 0 \cdot 0 \cdot 1 \cdot 0$
   - sum = 96
   - $g' = 0 \cdot 23 \cdot 4 \cdot 2 \cdot 0 \cdot 1 \cdot 1 \cdot 0$
   - 1472022

2. $f' = -1 \cdot 1 \cdot 0 \cdot 0 \cdot 1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 0$
   - sum = 96
   - $g' = 7 \cdot 3 \cdot 1 \cdot 6 \cdot 4 \cdot 0 \cdot 0 \cdot 1 \cdot 6 \cdot 3$
   - 4120121

3. $f' = -1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 0 \cdot 1 \cdot 0 \cdot 1$
   - sum = 96
   - $g' = 7 \cdot 3 \cdot 1 \cdot 6 \cdot 3 \cdot 1 \cdot 0 \cdot 15 \cdot 5$
   - 4020121

4. $f' = -1 \cdot 1 \cdot 1 \cdot -1 \cdot 0 \cdot 0 \cdot 0 \cdot 0 \cdot 1 \cdot 0$
   - sum = 96
   - $g' = 3 \cdot 7 \cdot 0 \cdot 3 \cdot 0 \cdot 9 \cdot 8 \cdot 2 \cdot 7$
   - 210117

5. $f' = -1 \cdot 1 \cdot 1 \cdot 1 \cdot 0 \cdot 0 \cdot 1 \cdot 0 \cdot 0 \cdot 0 \cdot 1$
   - sum = 96
   - $g' = 0 \cdot 3 \cdot 0 \cdot 7 \cdot 1 \cdot 0 \cdot 1 \cdot 8 \cdot 3 \cdot 9$
   - 126280

6. $f' = -1 \cdot 0 \cdot 1 \cdot 0 \cdot 0 \cdot 0 \cdot 1 \cdot 1 \cdot -1 \cdot 1 \cdot 0$
   - sum = 96
   - $g' = 0 \cdot 29 \cdot 1 \cdot 0 \cdot 9 \cdot 0 \cdot 9 \cdot 1 \cdot 9$
   - 136271

7. $f' = -1 \cdot 0 \cdot 1 \cdot 0 \cdot 0 \cdot 1 \cdot -1 \cdot 0 \cdot 0 \cdot 0 \cdot 1$
   - sum = 96
   - $g' = 7 \cdot 2 \cdot 1 \cdot 8 \cdot 3 \cdot 0 \cdot 0 \cdot 1 \cdot 5 \cdot 5 \cdot 4$
   - 020022

8. $f' = -1 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 0 \cdot 1 \cdot 1 \cdot 0 \cdot 1 \cdot 0$
   - sum = 96
   - $g' = 0 \cdot 1 \cdot 0 \cdot 0 \cdot 9 \cdot 1 \cdot 4 \cdot 4 \cdot 3 \cdot 7 \cdot 1$
   - 1271010

9. $f' = -1 \cdot 0 \cdot 0 \cdot -1 \cdot 1 \cdot 1 \cdot 0 \cdot 0 \cdot 1 \cdot -1 \cdot 0$
   - sum = 96
   - $g' = 1 \cdot 5 \cdot 5 \cdot 4 \cdot 1 \cdot 9 \cdot 0 \cdot 2 \cdot 3 \cdot 5$
   - 219300

10. $f' = -1 \cdot 0 \cdot 0 \cdot 1 \cdot -1 \cdot 1 \cdot 0 \cdot 0 \cdot 1 \cdot 0 \cdot 1$
    - sum = 96
    - $g' = 1 \cdot 0 \cdot 3 \cdot 1 \cdot 0 \cdot 1 \cdot 3 \cdot 1 \cdot 0 \cdot 0 \cdot 0$
    - 10310

11. $f' = -1 \cdot 0 \cdot 0 \cdot 1 \cdot -1 \cdot 1 \cdot 0 \cdot 0 \cdot -1 \cdot 1 \cdot 0$
    - sum = 96
    - $g' = 7 \cdot 3 \cdot 1 \cdot 5 \cdot 5 \cdot 0 \cdot 0 \cdot 1 \cdot 5 \cdot 3 \cdot 7$
    - 018320

The numbers of possibilities that satisfied step 7 in the algorithm, number of polynomials whose corresponding $g'$ coefficients contained $d_f$ of 1 and $d_g$ 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:

- $f' = 1 \cdot 0 \cdot 0 \cdot 1 \cdot -1 \cdot 1 \cdot 0 \cdot 1 \cdot 0 \cdot 1$
- $g' = 1 \cdot 0 \cdot 3 \cdot 1 \cdot 0 \cdot 1 \cdot 3 \cdot 1 \cdot 0 \cdot 0 \cdot 1 \cdot 0 \cdot 3 \cdot 1$
- $= 1 \cdot 0 \cdot 1 \cdot 0 \cdot 1 \cdot -1 \cdot 0 \cdot 0 \cdot 1 \cdot 0$
- $-1 \cdot 0$

Each g coefficient has a value equal to 31, $q-1$, will be replaced with -1. So, the private key $f$ is

- $-1 \cdot 0 \cdot 1 \cdot -1 \cdot 1 \cdot 0 \cdot 1 \cdot 0 \cdot 0 \cdot 1 \cdot 0$
- 1

And the private key $g$ is

- $1 \cdot 0 \cdot 1 \cdot -1 \cdot 1 \cdot 0 \cdot 0 \cdot 1 \cdot 0 \cdot -1$
- 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 $g$, $d_f$ and $d_g$, are respectively, 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


[20] Coppersmith D., Shamir A., "Lattice attacks on NTRU", in Proc. of EUROCRIPT 97, Lecture Notes in Computer Science, Springer-Verlag,
Table (1) \[Hof98b, How03c, Sil98\]

<table>
<thead>
<tr>
<th>Security level</th>
<th>N</th>
<th>P</th>
<th>q</th>
<th>f_{d_l}</th>
<th>r_{d_r-1}</th>
<th>g_{d_r}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moderate</td>
<td>167</td>
<td>3</td>
<td>128</td>
<td>61</td>
<td>60</td>
<td>18</td>
</tr>
<tr>
<td>High</td>
<td>263</td>
<td>3</td>
<td>128</td>
<td>50</td>
<td>49</td>
<td>16</td>
</tr>
<tr>
<td>Highest</td>
<td>503</td>
<td>3</td>
<td>256</td>
<td>216</td>
<td>215</td>
<td>55</td>
</tr>
</tbody>
</table>

Table (2) The value of \(\text{sum}_g\) at different parameters values

<table>
<thead>
<tr>
<th>Security level</th>
<th>N</th>
<th>P</th>
<th>q</th>
<th>f_{d_g}</th>
<th>\text{sum}_g</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moderate</td>
<td>107</td>
<td>64</td>
<td>3</td>
<td>12</td>
<td>768</td>
</tr>
<tr>
<td>Standard</td>
<td>167</td>
<td>128</td>
<td>3</td>
<td>20</td>
<td>2560</td>
</tr>
<tr>
<td>High</td>
<td>263</td>
<td>128</td>
<td>3</td>
<td>24</td>
<td>3072</td>
</tr>
<tr>
<td>Highest</td>
<td>503</td>
<td>256</td>
<td>3</td>
<td>72</td>
<td>18432</td>
</tr>
</tbody>
</table>

Table (3) The appearance of the first rotation

<table>
<thead>
<tr>
<th>Trial</th>
<th>N=11</th>
<th>N=13</th>
<th>N=15</th>
<th>N=17</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>First Solution</td>
<td>Percentag e</td>
<td>First Solution</td>
<td>Percentag e</td>
</tr>
<tr>
<td>1</td>
<td>679</td>
<td>0.0588</td>
<td>2408</td>
<td>0.040093</td>
</tr>
<tr>
<td>2</td>
<td>223</td>
<td>0.0193</td>
<td>5310</td>
<td>0.088412</td>
</tr>
<tr>
<td>3</td>
<td>538</td>
<td>0.0465</td>
<td>1739</td>
<td>0.028954</td>
</tr>
<tr>
<td>4</td>
<td>181</td>
<td>0.0156</td>
<td>139</td>
<td>0.002314</td>
</tr>
<tr>
<td>5</td>
<td>217</td>
<td>0.0188</td>
<td>5812</td>
<td>0.10342</td>
</tr>
<tr>
<td>6</td>
<td>345</td>
<td>0.0298</td>
<td>3971</td>
<td>0.066117</td>
</tr>
<tr>
<td>7</td>
<td>234</td>
<td>0.0202</td>
<td>6015</td>
<td>0.10015</td>
</tr>
<tr>
<td>8</td>
<td>183</td>
<td>0.0158</td>
<td>4818</td>
<td>0.08022</td>
</tr>
<tr>
<td>9</td>
<td>821</td>
<td>0.07108</td>
<td>1399</td>
<td>0.023293</td>
</tr>
<tr>
<td>10</td>
<td>467</td>
<td>0.04043</td>
<td>5421</td>
<td>0.09026</td>
</tr>
<tr>
<td>Total</td>
<td>11550</td>
<td>0.06060</td>
<td>630630</td>
<td>0.106306</td>
</tr>
</tbody>
</table>
Figure (1) relation between N and number of possibilities