A Thesis for the Degree of Master of Science

A Study on the Security of NTRUSign digital signature scheme

SungJun Min

School of Engineering

Information and Communications University

2004

A Study on the Security of NTRUSign digital signature scheme

A Study on the Security of NTRUSign digital signature scheme

Advisor : Professor Kwangjo Kim

by

SungJun Min
School of Engineering
Information and Communications University

A thesis submitted to the faculty of Information and Communications University in partial fulfillment of the requirements for the degree of Master of Science in the School of Engineering

Daejeon, Korea
Jan. 03. 2004
Approved by

(signed)

Professor Kwangjo Kim
Major Advisor

A Study on the Security of NTRUSign digital signature scheme

SungJun Min

We certify that this work has passed the scholastic standards required by Information and Communications University as a thesis for the degree of Master of Science

Jan. 03. 2004

Approved:

Chairman of the Committee
Kwangjo Kim, Professor
School of Engineering

Committee Member
Jae Choon Cha, Assistant Professor
School of Engineering

Committee Member
Dae Sung Kwon, Ph.D
NSRI

M.S. SungJun Min
20022052

A Study on the Security of NTRUSign digital signature scheme

School of Engineering, 2004, 43p.
Major Advisor : Prof. Kwangjo Kim.
Text in English

Abstract

The lattices have been studied by cryptographers for last decades, both in the field of cryptanalysis and as a source of hard problems on which to build encryption schemes. Interestingly, though, research about building secure and efficient signature schemes using the theory of lattices is extremely sparse in the cryptographic literature. An early scheme is due to Goldreich, Goldwasser and Halevi [7], who proposed that one could sign a message by demonstrating the ability to solve the approximate closest vector problem reasonably well for a point in space (hereafter referred to as the message digest point) generated from a hash of the message, and verify by checking that the "close lattice point" returned was indeed a lattice point and that it was close enough to the message digest point to make forgeries impractical. However, this idea was not analyzed in detail by its authors.

Another public-key cryptosystem, NTRU encryption scheme, was proposed at almost the same time by Hoffstein, Pipher, and Silverman. After that they introduced a new type of authentication and digital signature scheme called NTRUSign at CT-RSA'03 [11]. NTRUSign features reasonably short, easily created keys, high speed, and low memory requirements like NTRU encryption scheme. Its security is also based on the hard problem

of solving the approximate shortest(or closest) vectors in a certain lattice, called NTRU lattice. In this scheme, the signer uses secret knowledge to find a point in the NTRU lattice close to the given point. He/She then exploits this approximate solution to the closest vector problem as his signature.

There are two reasons for seeking this alternative hard problems (like GGH or NTRU) on which cryptography may be based. First, it is prudent to hedge against the risk of potential breakthroughs in factoring and computing discrete logarithms. A second and more significant reason is efficiency. NTRU-based algorithms, for example, run hundreds of times faster while providing the same security as competing algorithms. The drawback in using alternative hard problem is that they may not be well understood. Although lattice theory has been studied for over 100 years, the algorithmic nature of hard lattice problems such the shortest vector problem(SVP) was not really studied intensively until Lenstra, Lenstra and Lovász discovered a polynomial-time lattice basis reduction algorithm in 1982. Moreover, NTRU-based schemes use specific types of lattices based on an underlying polynomial ring, and these lattices generate specific types of lattice problems that may be easier to solve than general lattice problems. Since these specific lattice problems have been studied intensively only since NTRUEncrypt's introduction in 1996, we can expect plenty of new results.

In this thesis, first we propose an attack method of NTRUSign signature scheme. Our proposed attack will allow a passive adversary who observes only a valid message-signature pair to generate another signature. Thus NTRUSign signature scheme is not secure in the sense of malleability. From this property, we can derive a second signature of the message from any message-signature pair. In this case, one cannot distinguish the second one from the original one generated by who knows the secret key, which can be in practice regarded as a forgery. Although such a weakness does not allow the attacker to change the message string, this forgery shows that the signature scheme cannot be used for all kinds of applications. For example, if one would

like to apply it to electronic cash, finding a second valid signature for a bill should be impossible. Also, an entity receiving the message-signature pairs (m, s) and (m, s') such that $s \neq s'$ at the same time, neither s nor s' will be accepted as a valid signature for the message m by him. In this scenario if a legitimate signer wants to assert s as his/her own signature for the message m, then he/she should exhibit his/her private key.

Finally, we provide a simple technique to avoid this weakness in NTRUSign scheme. Although our modification does not degenerate the security of the original NTRUSign scheme, we are not sure whether or not the repaired version of NTRUSign is non-malleable. We believe, however, that it is computationally infeasible to find another shortest vector in repaired NTRUSign because all lattice-based signature schemes use a lattice vector sufficiently close to the vector derived from a message as its signature.

Contents

List of Tables

List of Figures

List of Abbreviations

CVP Shortest Vector Problem

NSS NTRU signature scheme

NTRUEncrypt NTRU encryption scheme

NTRUSign NTRUSign signature scheme

R-NSS Revised NTRU signature scheme

SVP Shortest Vector Problem

List of Notations

a a polynomial

a^{-1} inverse polynomial of a

α annihilating polynomial

$\|\cdot\|_2$ l_2-norm

$\|\cdot\|$ centered-norm

$L(X)$ lattice generated a set X

L_h^{NT} NTRU lattice

$R(A, B)$ resultant of two polynomials A and B

$\mathcal{V}(a)$ sum of all coefficients of a modulus q

$\mathbb{R}$ set of real numbers

$\mathbb{Z}$ set of integers

$\mathbb{Z}_q$ integers modulo q

R polynomial ring $\mathbb{Z}[x]/(x^N - 1)$

R_q polynomial ring $\mathbb{Z}_q[x]/(x^N - 1)$

R_q^* multiplicative group of units in R_q

Chapter 1
Introduction

1.1 Advent of new attacks

In the last twenty years *provable security* has dramatically developed, as a means to validate the design of cryptographic schemes. Goldwasser, Micali and Rivest introduced the notion of *existential forgery against chosen-message attacks* for public key signature schemes [9]. This notion has become the *de facto* security definition for digital signature algorithms, against which all new signature algorithms are measured. The definition involves a game in which the adversary is given a target user's public key and is asked to produce a valid message-signature pair with respect to this public key. The adversary is given access to an oracle which will produce signatures on messages of his choice.

Recently, Stern *et al.* described that there may be flaws in applying provable security proof methodologies to digital signature schemes [28]. The definition of provable security model does not directly deal with the most important property of a digital signature, namely *non-repudiation*: the signer should be unable to repudiate his signature. One should not permit that an adversary against the non-repudiation property of a signature scheme would be the legitimate signer himself. Hence, such an adversary has access to the private key, and may even control the key generation process. Stern *et al.* gave further examples of flaws in security proofs, related to signature schemes. Two of them stem from a subtle point that has apparently been somehow

overlooked: in non deterministic signature schemes, several signatures may correspond to a given message. Accordingly, it should make clear whether obtaining the second signature of a given message, different from a previously obtained signature of the same message, is a forgery or not, and namely an existential forgery. Next, if a legitimate signer can produce two messages which have the same signature with respect to the same public key, then he/she could publish the signature on one message and then claim it was actually the signature on another. Such a signature we shall call a *duplicate signature*, since it is the signature on two messages. If a signature scheme allows an adversary to find an additional signature for a message of his choice, already signed by the oracle, then the range of applications for the scheme will be limited. For example, for electronic cash, finding a second valid signature for a bill should not be possible. We call such a signature scheme *malleable signature*.

1.2 Impact of malleability

If a signature scheme is malleable, we can derive the second signature of the message from any message-signature pair. In this case, one cannot distinguish the second one from the original one generated by who knows the secret key, which can be in practice regarded as a forgery. Although such a weakness does not allow the attacker to change the message string, this kind of forgery shows that the signature scheme cannot be used for all kinds of applications. For example, if one would like to apply it to electronic cash, finding a second valid signature for a bill should be impossible. Also, an entity receiving the message-signature pairs (m, s) and (m, s') such that $s \neq s'$ at the same time, neither s nor s' will be accepted as a valid signature for the message m by him. If a legitimate signer wants to assert s as his/her own signature for the message m, then he/she should exhibit his/her private key.

1.3 Motivation

Recently, Hoffstein *et al.* introduced a public-key signature scheme called NTRUSign [12] related to the NTRU encryption scheme(now called NTRUEncrypt). NTRUEncrypt and NTRUSign are not based on traditional hard problems such as factoring or computing discrete logarithms, like much of today's cryptography. Instead, NTRUEncrypt was originally conceived as a cryptosystem based on polynomial arithmetic. Based on an early attack found by Coppersmith and Shamir, however, the underlying hard problem was soon reformulated as a lattice problem. There are two reasons for seeking alternative hard problems on which cryptography may be based. First, it is prudent to hedge against the risk of potential breakthroughs in factoring and computing discrete logarithms. A second and more significant reason is efficiency. NTRU-based algorithms, for example, are touted to run hundreds of times faster while providing the same security as competing algorithms. The drawback in using alternative hard problem is that they may not be as well understood. Although lattice theory has been studied for over 100 years, the algorithmic nature of hard lattice problems such the shortest vector problem(SVP) was not so much studied intensively until Lenstra, Lenstra and Lovász discovered a polynomial-time lattice basis reduction algorithm in 1982. Moreover, NTRU-based schemes use specific lattice based on an underlying polynomial ring, and this lattice generates specific types of lattice problem that may be easier to solve than general lattice problem. Since this specific lattice problem has been studied intensively only since NTRUEncrypt's introduction in 1996, we can expect plenty of new results. In this thesis, we propose a new method to find another shortest vector from given a shortest vector in certain lattices that arise in NTRUSign, allowing us to break the scheme in terms of malleability.

1.4 Our Contributions

In this thesis, we contribute to suggest an attack method how a passive adversary who observes only a valid message-signature pair can generate another signature. The main idea of this forgery is to use specific polynomials of which norm value is zero. While this weakness might be overlooked for a few applications, NTRUSign is not secure in the malleability sense against known message attack. Finally we propose a simple technique to avoid our proposed attack.

1.5 Organization of the thesis

In Chapter 2, we provide background mathematics related to lattices in which the security of NTRUSign is based. Then, in Section 2.2 we review the basic concepts of the digital signatures and provide a set of criteria to break a signature scheme. In Chapter 3, we briefly describe the NTRUSign signature scheme. We do not give all the technical and theoretical details for the functions used in the scheme. Only the general construction is described in this thesis. We show how an attacker can forge an additional signature for a message previously signed by using some specific polynomials in Chapter 4, and then we introduce a simple method to avoid our proposed attack. Finally, we conclude this thesis in Chapter 5.

Chapter 2
Preliminaries

2.1 Lattices

In this section we describe some basic facts about lattices and hard problems related to lattice theory. A lattice is a regular arrangement of points in space. In particular, for linearly independent $b_1, b_2, \cdots, b_n \in \mathbb{R}^n$, the lattice $L = L(b_1, b_2, \cdots, b_n)$ is the set of all integer linear combinations $a_1 b_1 + \cdots + a_n b_n$, where $a_1, a_2, \cdots, a_n \in \mathbb{Z}$. The vectors $b_1, b_2, \cdots, b_n$ form a *basis* of the lattice. Obviously, there are many different bases for any given lattice. We know that some problems related to lattice theory are *NP*-complete.

Definition 2.1.1 *Let $b_1, b_2, \cdots, b_n \in \mathbb{R}^m$ be linearly independent vectors. We call the additive subgroup*

$$L(b_1, b_2, \cdots, b_n) := \sum_{i=1}^{n} b_i \mathbb{Z} = \{\sum_{i=1}^{n} t_i b_i \mid t_1, t_2, \cdots, t_m \in \mathbb{Z}\}$$

of $\mathbb{R}^m$ a lattice with basis $\{b_1, b_2, \cdots, b_n\}$. The rank or the dimension of the lattice is rank$(L) := n$.

We can consider an example: $\mathbb{Z}^m$ is a lattice of rank m, the standard unit vectors $e_1, e_2 \cdots, e_m$ form a basis.

We introduce two famous computational problems on lattices: the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). In SVP, one is given a basis $\{b_1, b_2, \cdots, b_n\}$ and must find the shortest non-zero vector in $L(b_1, b_2, \cdots, b_n)$. In CVP, one is given a basis $\{b_1, b_2, \cdots, b_n\}$ and a target

vector v (not necessarily in the lattice) and must find the lattice vector in $L(b_1, b_2, \cdots, b_n)$ closest to v.

Definition 2.1.2 ***(Shortest Vector Problem SVP)***
The problem of the shortest lattice vectors in the l_2-norm is

$$L_2 - SVP \quad := \quad \left\{ (k, m, n, b_1, \cdots, b_n) \;\middle|\; \begin{array}{c} k, m, n \in \mathbb{N}, b_1, b_2, \cdots, b_n \in \mathbb{Z}^m, \\ \exists x \in L(b_1, b_2, \cdots, b_n) \backslash \{0\} \\ : \parallel x \parallel_2^2 \leq k \end{array} \right\},$$

where l_2-norm means general Euclidean norm.

The complexity of this SVP problem is still unresolved. Although some efforts to show that L_2-SVP is *NP*-hard have failed, this problem is known to be *NP*-hard with respect to *randomized* reductions by Ajitai [1]. However, the CVP problem is known to be *NP*-complete for any norm.

Definition 2.1.3 ***(Closest Vector Problem CVP)***
The problem of the closest lattice vectors in the l_2-norm is defined as

$$L_2 - CVP \quad := \quad \left\{ (k, m, n, b_1, \cdots, b_n, z) \;\middle|\; \begin{array}{c} k, m, n \in \mathbb{N}, b_1, b_2, \cdots, b_n, z \in \mathbb{Z}^m, \\ \exists x \in L(b_1, b_2, \cdots, b_n) \\ : \parallel z - x \parallel_2^2 \leq k \end{array} \right\}.$$

Given a lattice basis $\{b_1, b_2, \cdots, b_n\} \in \mathbb{Z}^m$, the following tasks are thought to be hard lattice problems:

- Find a short non-trivial lattice vector.
- Find a basis comprised of short lattice vectors.
- Find for a given $z \in \text{span}(b_1, b_2, \cdots, b_n)$ the closest lattice vector.

In contrast, given a system of generators $b_1, b_2, \cdots, b_n \in \mathbb{Z}^m$ for a lattice L, $n \geq \text{rank}(L)$, it is possible to construct a basis for L in polynomial time.

2.2 Digital Signature

The notion of a *digital signature* may prove to be one of the most fundamental and useful inventions of modern cryptography. A signature scheme provides a way for each user to sign messages so that the signatures can later be verified by anyone else. More specifically, each user can create a matched pair of private and public keys so that only he can create a signature for a message (using his private key), but anyone can verify the signature for the message (using the signer's public key). The verifier can convince himself that the message contents have not been altered since the message was signed. Also, the signer can not later repudiate having signed the message, since no one but the signer possesses his private key.

A digital signature scheme within the public key framework, is defined as a triple of algorithms (G, σ, V) such that

- Key generation algorithm G is a probabilistic, polynomial-time algorithm which on input a security parameter 1^k, produces pairs (P, S) where P is called a public key and S a secret key. (We use the notation $(P, S) \in G(1^k)$ to indicate that the pair (P, S) is produced by the algorithm G.)

- Signing algorithm σ is a probabilistic polynomial time algorithm which is given a security parameter 1^k, a secret key S in range $G(1^k)$, and a message $m \in \{0,1\}^k$ and produces as output string s which we call the signature of m. (We use notation $s \in \sigma(1^k, S, m)$ if the signing algorithm is probabilistic, otherwise $s = \sigma(1^k, S, m)$. As a shorthand when the context is clear, the secret key may be omitted and we will write $s \in \sigma(S, m)$ to mean meaning that s is the signature of message m.)

- Verification algorithm V is a probabilistic polynomial time algorithm which given a public key P, a digital signature s, and a message m, re-

turns 1 (*i.e.*, "true") or 0 (*i.e.*, "false") to indicate whether the signature is valid. We require that $V(P, s, m) = 1$ if $s \in \sigma(m)$ and 0 otherwise. (We may omit the public key and abbreviate $V(P, s, m)$ as $V(s, m)$ to indicate verifying signature s of message m when the context is clear.)

Note that if V is probabilistic, we can relax the requirement on V to accept valid signatures and reject invalid signatures with high probability for all messages m, all sufficiently large security parameter k, and all pairs of keys $(P, S) \in G(1^k)$. The probability is taken over the coins of V and S. Note also that the signed message may be plaintext or encrypted, because the message space of the digital signature system can be any subset of $\{0, 1\}^*$.

2.2.1 Attack Types on signature schemes

The goal of an adversary is to forge a signature, that is, produce a valid signature which will be accepted by some other entity. There are two basic attacks against digital signature scheme.

1. *Key-only attack*: In this attack, an adversary knows only the signer's public key.

2. *Message attack*: An adversary is able to examine signatures corresponding either to known or chosen messages. Message attacks can be divided into three classes:

    - *Known-message attack*: An adversary has signatures for a set of messages which are known to the adversary but not chosen by him.
    - *Chosen-message attack*: An adversary obtains valid signatures from a chosen list of messages before attempting to break the signature scheme. This attack is *non-adaptive* in the sense that messages are chosen before any signature are seen.

- *Adaptive chosen-message attack*: An adversary is allowed to use the signer as an oracle. The adversary may request signatures of messages which depend on the signer's public key and he may request signatures of messages which depend on previously obtained signatures or messages.

We distinguish several levels of cryptanalyzed by an adversary.

1. *Total break*: An adversary is either able to compute the private key information of the signer, or finds an efficient signing algorithm functionally equivalent to the valid signing algorithm.

2. *Selective forgery*: An adversary is able to create a valid signature for a particular message or class of messages chosen a *priori*. Creating the signature does not directly involve the legitimate signer.

3. *Existential forgery*: An adversary is able to forge a signature for at least one message. The adversary has little or no control over the message whose signature is obtained, and the legitimate signer may be involved in the deception. This forgery can be divided by two categories:

   - *Weak-existential forgery*: An adversary can create a signature for at least one message for which no signature has been issued by the legitimate signer.
   - *Strong-existential forgery*: An adversary can create a signature-message pair that never been observed by the signer. Sometimes, s-existential forgeable signature schemes are called *malleable* schemes.

Clearly, different levels of security may be required for different applications. Sometimes, it may suffice to show that an adversary who is capable of a known-message attack can not succeed in selective forgery, while for

other applications it may be required that an adversary capable of a chosen-message attack can not succeed even at existential forgery with non-negligible probability. The security that we will focus on, in this thesis, is that with high probability a polynomial time adversary would not be able to even s-existentially forge in the presence of a chosen message attack.

We say that a digital signature is secure if an enemy who can use the real signer as "an oracle" can not forge a signature for any message whose signature was not obtained from the real signer in polynomial-time in the size of the public key. Formally, let B be a black box which maps messages m to valid signatures, *i.e.*, $V(P, B(m), m) = 1$ for all message m. Let the forging algorithm F on input the public key P have access to B, denoted as $F^B(P)$. The forging algorithm runs in two stages: it first launches a chosen message attack, and then outputs a "new forgery" which is defined to be any message-signature pair such that the message was not signed before and that signature is valid. We require that for all forging algorithms F, for all polynomials Q, for all sufficiently large k,

$$Prob(V(P, s, m) = 1 : (P, S) \leftarrow G(1^k); (m, s) \leftarrow F^B(P)) \leq \frac{1}{Q(k)}.$$

The probability is taken over the uniformly distributed choice of the keys $(P, S) \in G(1^k)$, the coin tosses of the forgery algorithm F, and the coins of B.

Chapter 3
NTRUSign Signature Scheme

3.1 History of NTRUSign scheme

The NTRU cryptosystem was first presented by Hoffstein, Pipher and Silverman at CRYPTO'96. It is a ring-based cryptosystem operating in the polynomial ring $\mathbb{Z}_q[x]/(x^N - 1)$ where N is the security parameter. NTRU has been received remarkable attention because of its encryption and decryption speed and the easiness of creating public-key/secret-key pairs, which makes it practical to change keys frequently. Its security is based on the hard mathematical problem of finding short and/or close vectors in a certain class of lattices, called NTRU lattices. Since the advent of NTRU encryption scheme, several related signature schemes such as NSS [14] and R-NSS [10] have been proposed. A fast authentication and digital signature schemes called NSS, based on the same underlying hard problem and using keys of the same form, was presented at Eurocrypt 2001 [14]. However, this scheme was broken by Mironov and Gentry *et al.*, see [5, 20]. In their Eurocrypt presentation, the authors of NSS sketched a revised version of NSS (called R-NSS) and published in the preliminary cryptographic standard document EESS [31]. Although R-NSS was significantly stronger than the previous version(NSS), Gentry and Szydlo proved that key recovery attack could be mounted [6]. The source of these weaknesses about NSS and R-NSS was an incomplete linking of the NSS method with the approximate closest vector problem in the NTRU lattice. In other words, the weaknesses of NSS and R-NSS arose

from the fact that the signer did not possess a complete basis of short vectors for the NTRU lattice L_h^{NT}. Later on, Hoffstein *et al.* proposed a new NTRU based signature scheme called NTRUSign. Unlike previous signature schemes, the link in NTRUSign between the signature and the underlying approximate closest vector problem is clear and direct: the signer must solve an "approximate CVP problem" in the lattice *i.e.,* produce a lattice point that is sufficiently close to a message digest point.

3.2 Overview of NTRUSign

In this section, we briefly describe the NTRUSign digital signature scheme. As NTRU encryption scheme, basic operations take place in the quotient ring $R = \mathbb{Z}[x]/(x^N - 1)$, where N is the security parameter. A polynomial $a(x) \in R$ can be presented by a vector $\mathbf{a}$ of its coefficients as follows:

$$\mathbf{a} = \sum_{i=0}^{N-1} a_i x^i = (a_0, a_1, \cdots, a_{N-1}).$$

For the sake of simplicity, we will use the same notation for the polynomial $a(x)$ and the vector $\mathbf{a}$. The product of two polynomials $\mathbf{a}$ and $\mathbf{b}$ in R is simply calculated by $\mathbf{a} * \mathbf{b} = \mathbf{c}$, where the k-th coefficient c_k is

$$c_k = \sum_{i=0}^{k} a_i b_{k-i} \; + \sum_{i=k+1}^{N-1} a_i b_{N+k-i} = \sum_{i+j \equiv k \pmod N} a_i b_j.$$

Hereafter, we sometimes write a polynomial $a(x)$ as simply a. In some steps, NTRUSign uses the quotient ring $R_q = \mathbb{Z}_q[x]/(x^N - 1)$, where the coefficients are reduced by modulo q, where q is typically a power of 2, for example 128. The multiplicative group of units in R_q is denoted by R_q^*. The inverse polynomial of $a \in R_q^*$ is denoted by a^{-1}. If a polynomial a has all coefficients chosen from the set $\{0, 1\}$, we call this *binary* polynomial.

The security of NTRUSign scheme is based on the approximately closest vector problem in a certain lattice, called NTRU lattice. In this scheme, the

signer can sign a message by demonstrating the ability to solve the approximately closest vector problem reasonably well for the point generated from a hashed message in a given space. The basic idea is as follows: The signer's private key is a short basis for an NTRU lattice and his public key is a much longer basis for the same lattice. The signature on a digital document is a vector in the lattice with two properties:

- The signature is attached to the digital document being signed.
- The signature demonstrates an ability to solve a general closest vector problem in the lattice.

The way in which NTRUSign achieves these two properties may be briefly summarized as follows:

Key Generation: The private key includes a short $2n$-dimensional vector denoted (f, g). The public key is the large n-dimensional vector h that specifies the NTRU lattice L_h^{NT}, that is, h is generated from f and g by $h \equiv f^{-1} * g \pmod{q}$. The private key also includes a complementary short vector (F, G) that is chosen so that (f, g) and (F, G) generate the full NTRU lattice L_h^{NT}.

Signing: The digital document to be signed is hashed to create a random vector (m_1, m_2) modulo q. The signer uses the secret short generating vectors to find a lattice vector (s, t) that is close to (m_1, m_2).

Verification: The verifier uses the public key h to verify that (s, t) is indeed the lattice L_h^{NT} and he verifies that (s, t) is appropriately close to $m_1, m_2)$.

NTRUSign algorithm uses the centered norm concept instead of Euclidean norm in verification step to measure the size of an element $a \in R$.

Definition 3.2.1 *Let $a(x)$ be a polynomial in ring $R = \mathbb{Z}[x]/(x^N - 1)$. Then the centered norm of $a(x)$ is defined by*

$$\| a(x) \|^2 = \sum_{i=0}^{N-1} (a_i - \mu_a)^2 = \sum_{i=0}^{N-1} {a_i}^2 - \frac{1}{N} (\sum_{i=0}^{N-1} a_i)^2$$

, where $\mu_a = \dfrac{1}{N} \sum_{i=0}^{N-1} a_i$ is the average of the coefficients of $a(x)$.

The centered norm of an n-tuple $(a_1, a_2, \cdots, a_n)$ with $a_1, a_2, \cdots, a_n \in R$ can be defined by this formula

$$(\| (a_1, a_2, \cdots, a_n) \|)^2 = \| a_1 \|^2 + \| a_2 \|^2 + \cdots + \| a_n \|^2 .$$

Note that the signature on a document D is a vector (s, t) in NTRU lattice L_h^{NT}, which is very close to m. To solve an approximately closest vector problem in the lattice, a signer uses a his secret "short basis" defined as below:

Definition 3.2.2 *A basis $\{(f, g), (F, G)\}$ is called a short basis in L_h^{NT} if*

$$\| f \|, \| g \| = O(\sqrt{N}), \text{ and } \| F \|, \| G \| = O(N),$$

where N is a half dimension of NTRU lattice L_h^{NT}.

3.3 NTRUSign scheme

In this section we describe NTRUSign key generation and the NTRUSign signing and verification protocols.

The system parameters of NTRUSign include

N: a (prime) dimension.

q: a power of 2.

d_f, d_g: key size parameters.

$NormBound$: a bound parameter of verification.

Key generation. A signer creates his public key h and the corresponding private key $\{(f, g), (F, G)\}$ as follows:

1. Choose binary polynomials f and g with d_f 1's and d_g 1's, respectively.
2. Compute the public key $h \equiv f^{-1} * g \pmod{q}$.
3. Compute small polynomials (F, G) satisfying $f * G - g * F = q$.

Signing. To sign a digital document D, the signer first hashes D to produce a message digest $m = (m_1, m_2)$ composed of two random mod q polynomials m_1 and m_2. We do not discuss this hash function proposed in [11]. From now, we assume that it is a randomized mapping that fulfills the necessary security requirements.

A signer generates his signature s on the digital document D as follows:

1. Obtain the polynomials (m_1, m_2) mod q for the document D by using the public hash function.
2. Write
$$\begin{aligned} G * m_1 - F * m_2 &= A + q * B \\ -g * m_1 + f * m_2 &= a + q * b, \end{aligned} \tag{3.1}$$
where A and a have coefficients between $-q/2$ and $q/2$.
3. The signature on D is a vector $(s, t) \in L_h^{NT}$, which is very close to $m = (m_1, m_2)$.
$$\begin{aligned} s &\equiv f * B + F * b \pmod{q} \\ t &\equiv g * B + G * b \pmod{q}. \end{aligned} \tag{3.2}$$
4. The polynomial s is the signature on the digital document D for the public key h.

Verification. Let s be a NTRUSign signature for the message digest $m = (m_1, m_2)$ and public key h. The signature will be valid if it demonstrates that the signer knows a lattice point in L_h^{NT} that is sufficiently close to the message digest vector m. Verification thus consists of the following steps:

1. Hash the document D to recreate (m_1, m_2).

2. With the signature s and public key h, compute the corresponding polynomial
$$t \equiv s * h \pmod{q}.$$
(Note that (s, t) is a point in the NTRU lattice L_h^{NT}.)

3. Compute the distance(*i.e.,* centered norm) from (s, t) to (m_1, m_2) and verify that it is smaller than the $NormBound$ parameter. In other words, check that
$$\| s - m_1 \|^2 + \| t - m_2 \|^2 \leq NormBound^2,$$
where the norm($\| \cdot \|$) is a centered norm.

The signing process of NTRUSign may be explained by the following matrix equation, which shows that the signer is using his short basis $\{(f, g), (F, G)\}$ to find approximate solutions to the closest vector problem:

$$\begin{aligned}
\begin{pmatrix} s & t \end{pmatrix} &= \begin{pmatrix} B & b \end{pmatrix} \begin{pmatrix} f & g \\ F & G \end{pmatrix} \\
&= \left\lfloor \begin{pmatrix} m_1 & m_2 \end{pmatrix} \begin{pmatrix} G/q & -g/q \\ -F/q & f/q \end{pmatrix} \right\rceil \begin{pmatrix} f & g \\ F & G \end{pmatrix} \\
&= \left\lfloor \begin{pmatrix} m_1 & m_2 \end{pmatrix} \begin{pmatrix} f & g \\ F & G \end{pmatrix}^{-1} \right\rceil \begin{pmatrix} f & g \\ F & G \end{pmatrix}
\end{aligned}$$

A valid signature demonstrates that the signer knows a lattice point that is within $NormBound$ of the message digest vector m. The designers argue that the suggested parameters $(N, q, d_f, d_g, NormBound) = (251, 128, 73, 71, 300)$ offer an equivalent security as 1,024 bit RSA [12].

The correctness of verification. A valid signature demonstrates that the signer knows a lattice point that is within $NormBound$ of the message digest vector m. Clearly the smaller that $NormBound$ is set, the more difficult it will be for a forger, without knowledge of the private key, to solve this problem. It is thus important to analyze how small we can set the bound $NormBound$, while still allowing valid signatures to be efficiently generated by the signer.

From *Eqs.* (3.1) and (3.2), we can calculate

$$(m_1, m_2) - (s, t) = \begin{pmatrix} A/q & a/q \end{pmatrix} \begin{pmatrix} f & g \\ F & G \end{pmatrix}. \tag{3.3}$$

We recall that the coefficients of a and A are between $-\frac{q}{2}$ and $\frac{q}{2}$, and hence

$$m_1 - s = \varepsilon_1 * f + \varepsilon_2 * F \quad \text{and} \quad m_2 - t = \varepsilon_1 * g + \varepsilon_2 * G, \tag{3.4}$$

where $\varepsilon_1 = A/q$ and $\varepsilon_2 = a/q$ are polynomials whose coefficients are between $-\frac{1}{2}$ and $\frac{1}{2}$.

As m_1 and m_2 vary across all mod q polynomials, it is easy to check that A varies uniformly across all mod q polynomials, so to all intents and purpose, the coefficients of ε_1 may be treated as independent random variables that are uniformly distributed in the interval $(-\frac{1}{2}, \frac{1}{2})$. Hence on average we have $\| \varepsilon_1 \| \approx \| \varepsilon_2 \| \approx \sqrt{N/12}$.

We can now estimate the distance from (s, t) to (m_1, m_2) using $\| \varepsilon_1 \| \approx$

$\| \varepsilon_2 \| \approx \sqrt{N/12}$ and the quasi-multiplicativity of the norm:

$$\| (m_1 - s, m_2 - t) \|^2 = \| (\varepsilon_1 f + \varepsilon_2 F, \varepsilon_1 g + \varepsilon_2 G) \|^2 \approx \frac{c^2 N^3}{72}(1 + \frac{12}{N}), \quad (3.5)$$

where c is calculated as 0.45 when $(N, q, df, dg) = (251, 128, 73, 71)$. From these parameters the right hand side of *Eq.* (3.5) yields a bound of $46601 \approx 215.87^2$. This is the average expected distance. Thus the verification of NTRUSign works well.

3.3.1 Method for Generating NTRUSign Keys

In signing step, we need to compute another short vector (F, G) such that $f * G - g * F = q$ for a given short vector (f, g). For an NTRU lattice, the first short vector (f, g) and the public parameters N and q completely determine the lattice L_h^{NT}, so the signer only has a short basis for half of the lattice. Thus he needs to use the known short vector (f, g) to find a complementary short vector (F, G) that, together with (f, g), generates L_h^{NT}. The general strategy for computing another short vector (F, G) is to project f and g down to $\mathbb{Z}$ via the resultant mapping. The definition of *resultant* is as following [2]:

Definition 3.3.1 *Let A and B be two polynomials over an integral domain $\mathcal{R}$ with quotient field K, and let $\overline{K}$ be an algebraic closure of K.*

Let $A(x) = a(x - v_1) \cdots (x - v_m)$ and $B(x) = b(x - w_1) \cdots (x - w_n)$ be the decomposition of A and B in $\overline{K}$. Then the resultant $R(A, B)$ of A and B is given by one of the equivalent formulas:

$$\begin{aligned} R(A, B) &= a^n B(v_1) \cdots B(v_m) \\ &= (-1)^{mn} b^m A(w_1) \cdots A(w_n) \\ &= a^n b^m \prod_{1 \le i \le m, 1 \le j \le n} (v_i - w_j). \end{aligned}$$

From this definition, we use the fact that

$$R_f \equiv \prod_{i=0}^{N-1} f(x^i) \bmod \Phi \in \mathbb{Z},$$

where R_f means the resultant of f with $x^N - 1$ and $\Phi(x) = \sum_{i=0}^{N-1} x^i$.

We then begin by using standard method to find polynomials $u, v \in \mathbb{Z}[x]$ satisfying

$$\begin{aligned} f * v + k_1 * (x^N - 1) &= R_f, \\ g * u + k_2 * (x^N - 1) &= R_g. \end{aligned}$$

Assuming that the resultants R_f and R_g are coprime, we apply the integer extended Euclidean algorithm to obtain integers γ and δ satisfying

$$\gamma R_f + \delta R_g = 1.$$

Combining these relations gives the formula

$$(\gamma v) * f + (\delta u) * g = 1 \pmod{x^N - 1}.$$

Thus by setting $F = -q\delta u$ and $G = q\gamma v$ we can get (F, G) satisfying

$$f * G - g * F = q.$$

Remark 1. Although the $\{(f, g), (F, G)\}$ generated as the above complete a basis for L_h^{NT}, the vector (F, G) typically has very large coefficients. However, we can dramatically reduce the size of F and G by adding or subtracting some polynomials, see [12].

Chapter 4
Cryptanalysis of NTRUSign

4.1 Proposed attack

In this section we describe that the NTRUSign is strong existential forgeable, sometimes this notion is called as malleable. Strong existential forgeability for a given signature scheme means that one can create a message-signature pair that has never been observed by the signer [28]. A different signature for a once legitimately signed message can be regarded as a forgery. In practice, this forgery shows that the NTRUSign scheme cannot be used for all kinds of applications. For example, in electronic cash system, finding a second valid signature for a bill should be impossible. Thus the application area of this scheme is limited, because a digital signature scheme is selected according to both its security level and the context of use.

Now we will describe how we can generate a valid signature different from a previous valid signature for a given message. Remind that NTRUSign signature scheme uses the centered norm in verification step. The centered norm is quasi-multiplicative, that is, $\| a(x) * b(x) \| \approx \| a(x) \| * \| b(x) \|$ for randomly chosen polynomials $a(x)$ and $b(x)$ in R, which was well discussed in [13]. The properties of the centered norm will be employed to induce a new signature from a given signature without knowing the private keys.

The following lemma describes the centered norm properties.

Lemma 4.1.1 *Let R be a quotient polynomial ring $\mathbb{Z}[x]/(x^N - 1)$. Then*

(i) In $R_q = \mathbb{Z}_q[x]/(x^N - 1)$*, there exist exactly* q *polynomials* $\alpha(x)$ *such that* $\| \alpha(x) \| = 0$.

(ii) If $\| \alpha(x) \| = 0$*, then* $\| \alpha(x) * \beta(x) \ (mod\ q) \| = 0$ *for every polynomial* $\beta(x) \in R$.

Proof:

(i) It is obvious that $\alpha_0 = \cdots = \alpha_{N-1}$ for $\alpha_i \in (-q/2, q/2]$ if and only if $\sum_{i=0}^{N-1}(a_i - \mu_a)^2 = 0$ where $\mu_a = \frac{1}{N}\sum_{i=0}^{N-1} a_i$, namely $\| a(x) \| = 0$.

(ii) From the result of (i), all coefficients of α are the same, say $\alpha = (\alpha_0, \alpha_0, \cdots, \alpha_0)$. Then, clearly the k-th coefficient of $\alpha * \beta$ is $\sum_{i=0}^{N-1}(\alpha_0\beta_{k-i}) + \sum_{i=k+1}^{N-1}(\alpha_0\beta_{N+k-i}) = \alpha_0(\beta_0 + \cdots + \beta_k + \beta_{k+1} + \beta_{N-1}) = \alpha_0 * \beta$, and so are the other coefficients of $\alpha * \beta$ the same. Again by applying to (i), we complete the proof of this lemma. □

We call these q polynomials $\alpha(x) \in R_q$ satisfying $\| \alpha(x) \| = 0$ *annihilating polynomial*. These annihilating polynomials may be used to make the NTRUSign algorithm malleable.

Hoffstein *et al.* argued that forgery of a signature in NTRUSign is equivalent to the ability to solve an approximately closest vector problem in high dimension for the class of NTRU lattices. It seems to be true if we do not consider the stronger attack model. Historically, Goldwasser, Micali and Rivest [9] introduced the notion of existential forgery against chosen-message attacks for public key signature scheme. This notion has become the *de facto* security definition of digital signature algorithm, against which all new signature algorithms are measured. In this scenario, an adversary with access to the public key of the scheme and to a signing oracle, should not be able to forge a valid signature for some new message or for a message of his choice(existential forgery and selective forgery, respectively). An even stronger requirement called the non malleability, or strong unforgeability, also forbids an adversary

to forge an additional signature for a message which might already have been signed by the oracle [28]. We can see more detail security notions for digital signature scheme and the relation between them in [9, 22].

Now we will show that one can easily generate a message-signature pair that has never been observed by the signer. To create additional valid signatures we use the following ***Remark*** and **Lemma**. Recall that all coefficients of polynomials are reduced by modulo q.

Remark 1. Let $\alpha \in R_q$ be an annihilating polynomial. Then $\| r \| \approx \| r + \alpha \pmod{q} \|$ for randomly chosen polynomial $r \in R_q$.

If both "reduced form" and "not reduced form" of polynomial $r + \alpha$ are equal, then the centered norm values of $\| r \|$ and $\| r + \alpha \pmod{q} \|$ are exactly the same. The differences between $\| r + \alpha \pmod{q} \|$ and $\| r \|$ are caused from only the gap failure. The concepts of gapping and wrapping failure are presented in [25]. We have implemented the above remark with the suggested parameters 1,000 times for each α by using Mathematica 4.2. It is clear that as the coefficients of annihilating polynomial α gets smaller, the probability of having the same norm gets higher. When the coefficient of α is ± 1 or ± 2, our experiment shows that each probability which two centered norm values are exactly the same becomes 0.15 and 0.015 approximately. Figure 4.1 describes the distribution of differences between $\| r + \alpha \pmod{q} \|$ and $\| r \|$ for random polynomial $r \in R_q$, where the α_i-axis denotes the coefficient of annihilating polynomial.

We will see some results induced from the properties of these annihilating polynomials. For any polynomial $f = (f_0, f_1, \cdots, f_{N-1}) \in R$, $\mathcal{V}(f)$ denotes the sum of all coefficients of f modulus q, that is,

$$\mathcal{V}(f) = f(1) = \sum_{i=0}^{N-1} f_i \pmod{q} \in \mathbb{Z}_q. \tag{4.1}$$

For any $f \in R$, the product $f * \alpha$ can be presented by $\mathcal{V}(f)\alpha$, where α is an annihilating polynomial(See the proof of **Lemma 4.1.1**). From *Eq.* (4.1) it is trivial that $\mathcal{V}$ has the following properties:

Lemma 4.1.2 *Let f and g be two polynomials in R.*

(i) $\mathcal{V}(f)\mathcal{V}(g) \equiv \mathcal{V}(f * g) \ (mod\ q)$.

(ii) $\mathcal{V}(f^{-1}) \equiv \mathcal{V}(f)^{-1} \ (mod\ q)$ *if f has an inverse in R_q.*

Proof: For an arbitrary annihilating polynomial α, we know that $f * \alpha = \mathcal{V}(f)\alpha \pmod{q}$. From this property, we can obtain the following equation:

$$\begin{aligned} \mathcal{V}(f * g)\alpha &\equiv (f * g) * \alpha = f * (g * \alpha) \\ &\equiv f * (\mathcal{V}(g)\alpha) = \mathcal{V}(g) f * \alpha \\ &\equiv \mathcal{V}(f)\mathcal{V}(g)\alpha \pmod{q}. \end{aligned}$$

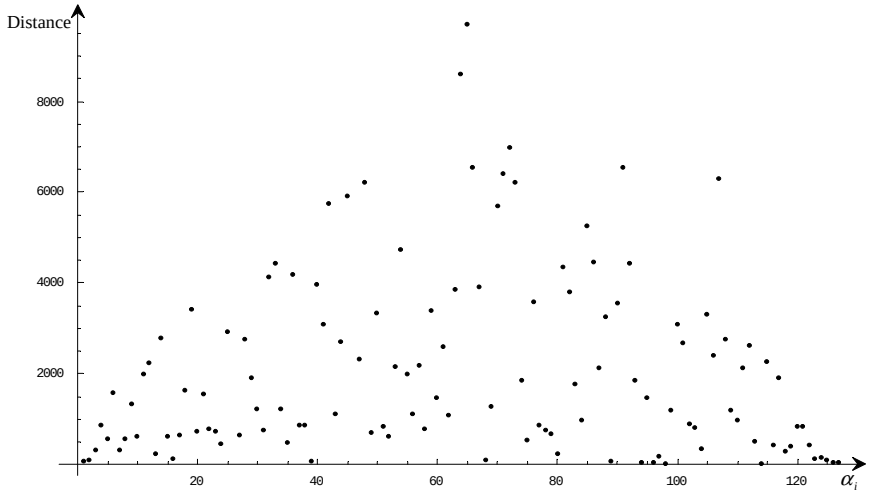

Figure 4.1: Distance between $\| r + \alpha \pmod{q} \|$ and $\| r \|$

Therefore we have $\mathcal{V}(f)\mathcal{V}(g) \equiv \mathcal{V}(f*g) \pmod{q}$. Obviously $\mathcal{V}(f^{-1})\mathcal{V}(f) \equiv \mathcal{V}(f^{-1} * f) \equiv \mathcal{V}(1) \equiv 1 \pmod{q}$, hence $\mathcal{V}(f^{-1}) \equiv \mathcal{V}(f)^{-1} \pmod{q}$. □

Assume that one chooses two polynomial pair (f, g), where f has an inverse in R_q. If there exists somewhat small integer $\alpha_0 \in (-q/2, q/2]$ satisfying $\alpha_0\mathcal{V}(f)^{-1}\mathcal{V}(g) \pmod{q}$ is also small, then we can know that both polynomial $\alpha = (\alpha_0, \alpha_0, \cdots, \alpha_0)$ and $(f^{-1}*g)*\alpha$ are annihilating polynomials with somewhat small coefficients from **Lemma 4.1.2**.

Remark 2. In the suggested parameters $(d_f, d_g) = (73, 71)$ given in [12], one has $\mathcal{V}(f) = -55$ and $\mathcal{V}(g) = -57$. In this case one can choose $\alpha = 8\sum_{i=0}^{N-1} x^i$ so that

$$\begin{aligned} h * \alpha \pmod{q} = \mathcal{V}(h)\alpha &= \mathcal{V}(f^{-1} * g) * \alpha \\ &= \mathcal{V}(f)^{-1}\mathcal{V}(g) * \alpha \\ &= -8\sum_{i=0}^{N-1} x^i. \end{aligned}$$

For a given signature $(s, t) \in L_h^{NT}$ generated under the suggested parameters, we take $s' = s + \alpha \pmod{q}$, where $\alpha = 8\sum_{i=0}^{N-1} x^i$. Then the corresponding signature pair t' is

$$\begin{aligned} t' = s' * h \pmod{q} &= s * h + \alpha * h \pmod{q} \\ &= t - 8\sum_{i=0}^{N-1} x^i \pmod{q}. \end{aligned}$$

At this time, we can expect that both $\| s - m_1 \|$ and $\| t - m_2 \|$ are small. Moreover, it is plausible that the small number of their coefficients are out of the range $(-64 + 8, 64 - 8]$. Form these reasons, the new lattice point $(s', t') = (s + 8\sum_{i=0}^{N-1} x^i, t - 8\sum_{i=0}^{N-1} x^i)$ will be another valid signature with high probability. Simply speaking, if one has $s - m_1$ without any coefficients greater than 56 and $t - m_2$ without any coefficients less than -55, then one

can have the following equation exactly:

$$\begin{aligned} \| s' - m_1 \|^2 + \| t' - m_2 \|^2 &= \| s - m_1 \|^2 + \| t - m_2 \|^2 \\ &\leq NormBound^2, \end{aligned}$$

which means (s', t') is always another valid signature.

A numerical experimental result shows that one has much more chance to succeed in the proposed attack: we examine a set P that consists of 128,000 elements from $\mathbb{Z}_{128}[x]/(x^{251}-1)$ generated in such a way that all coefficients are randomly chosen from normal distribution with uniformly chosen means $\mu \in (-64, 64]$ and a fixed standard deviation $\sigma = \sqrt{Normbound^2/N} \approx 18.9$. For two sets $P' = \{s \in P \mid \| s \|^2 < 300^2\}$ and $P'' = \{s \in P' \mid \| s + 8 \sum_{i=0}^{N-1} x^i \|^2 < 300^2\}$, we obtained the result that the set P' consists of 20,650 distinct elements and that P' and P'' coincide exactly.

We implemented the full NTRUSign signature scheme with suggested parameters using GNU MP version 4.1.2. Our experiment illustrates that the proposed forgery (s', t') almost always succeeds for given message document D and valid signature (s, t). Table 4.1 describes the number of forged valid signatures among valid signatures as the coefficients of annihilating polynomial α change. Table 4.2 depicts the approximate probability that new pair $(s', t') = (s + \alpha, t + h * \alpha)$ (mod q) would be the second signature for given valid signature (s, t). Note that α_i denotes the coefficient of annihilating polynomial α. One detail example of this forgery is given in **Appendix**.

Although the NTRUSign signature scheme is deterministic, several valid signatures are associated to the same message. This property allows an adversary to find an additional signature for a message of his choice, already signed by the oracle without solving the hard closest vector problems. This attack represents a failure of the strong unforgeability security, thus malleability.

Coefficient of α	*# of valid signatures (A)*	*# of forged signatures (B)*	*Success Prob. for forgery (B/A)*
1	944	790	0.836
2	944	608	0.644
⋮	⋮	⋮	⋮
7	944	668	0.707
8	944	840	0.889 *
9	944	805	0.852
⋮	⋮	⋮	⋮
63	944	158	0.167
64	944	156	0.165 **

Table 4.1: Experimental result using GNU MP

Coefficient α_i *of an annihilating polynomial*	*When* (s,t) *is valid the approx. probability that* (s',t') *is valid*
1	0.836
2	0.644
⋮	⋮
7	0.707
8	0.889
9	0.852
⋮	⋮
63	0.167
64	0.165

Table 4.2: Approximate forgery probability - $s' = s + \alpha$

4.2 Repairing of NTRUSign

In this section we present a simple way in order to avoid the weakness in the NTRUSign signature scheme. The strategy for repairing NTRUSign is to make the signing transformation one-to-one correspondence for a given message and secret key. It can be achieved by adding an annihilating polynomial in the signing step. Our idea is to make the top-coefficient (*i.e.,* the coefficient of x^{N-1}) of the signature s obtained from the original NTRUSign to be zero. If the distance between the new signature s' computed by this process and given point is not as close as to the expected distance, then we simply add the annihilating polynomial $\sum_{i=0}^{N-1} x^i$ to the signature s' until it becomes to a valid signature.

The repaired version of NTRUSign scheme is as follows:

Signing Signer generates his signature s' on the digital document D

INPUT: private key $\{(f, g), (F, G)\}$ and hashed message (m_1, m_2)

OUTPUT: valid signature s'

1. Obtain the signature s from the original NTRUSign.
2. Set $s' \longleftarrow s - s_{N-1} \sum_{i=0}^{N-1} x^i \pmod{q}$.
3. While $\| s' - m_1 \|^2 + \| t' - m_2 \|^2 > NormBound^2$ do the following:
   3.1. Set $s' \longleftarrow s' + \sum_{i=0}^{N-1} x^i \pmod{q}$.
4. Return(s').

Verifying Receiver verifies the signature s'

INPUT: signature s' and sender's public key h

OUTPUT: "Accept" or "Reject"

1. Compute $t' = s' * h \pmod{q}$.
2. If $\| s' - m_1 \|^2 + \| t' - m_2 \|^2 > NormBound^2$,
   then return("Reject").
3. While $s'_{N-1} \neq 0$:
   3.1. Set $s' \longleftarrow s' - \sum_{i=0}^{N-1} x^i \pmod{q}$.
   3.2. If $\| s' - m_1 \|^2 + \| t' - m_2 \|^2 \leq NormBound^2$,
   then return("Reject").
4. Return("Accept").

It is obvious that our modification does not degenerate the security of the original NTRUSign scheme. Actually two problems based on original NTRUSign and Repaired NTRUSign are computationally equivalent. The reason is that in the repaired NTRUSign, after making specific coefficient of s to be zero, we add the annihilating polynomial having coefficient 1, which is public information. Although our proposed attack cannot be applied for the repaired NTRUSign anymore, we do not know whether it is non-malleable or not yet. In worst case, there exist 2^N shortest vectors for a given target point m in NTRU lattice. We believe, however, that it is computationally infeasible to find another shortest vector because in all lattice-based signature schemes a signature is lattice vector sufficiently close to a vector derived from the message. In other words, if one could obtain two *non-trivial short lattice*

vectors for the same message, then one could also obtain a short nonzero lattice vector by subtraction, where "non-trivial short lattice vectors" mean that each lattice vector is not related to an annihilating polynomial. This sounds strange because it is supposedly very hard and would probably help lattice reduction if one could collect many such short vectors.

Chapter 5
Conclusions and Further Work

In this thesis we have described a weakness of the NTRUSign digital signature scheme that can cause significant problems in some real applications if one is unaware of it. We showed that NTRUSign signature scheme is not secure in terms of strongly existential forgery, in other words it is malleable. This notion allows an adversary to find new signatures for a message of his choice, given a signature for this message. This forgery requires a specific polynomial with small coefficient satisfying its norm value equal to zero. Even if this forgery does not admit an adversary to change the message, NTRUSign scheme cannot be used for all applications. We also proposed a simple method to repair the scheme. Although our modification does not degenerate the security of the original NTRUSign scheme, we do not know whether or not the repaired version of NTRUSign is non-malleable.

As the future work, it remains as our task to prove that the repaired NTRUSign is non-malleable signature scheme.

Appendix

An Example of Signature Forgery

Here we give an example of how to generate another signature from a given message-signature pair. Let parameters be as defined in Efficient Embedded Security Standards (EESS) [30]; $N = 251$, $q = 128$, $d_f = 73$, $d_g = 71$, and $NormBound = 300$.

The *binary* private key f, g and complementary private key F, G satisfying $f * G - g * F = q$ are as following:

---

f $(d_f = 73)$

---

0 , 1 , 0 , 1 , 0 , 1 , 1 , 1 , 0 , 0 , 1 , 0 , 1 , 1 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 1 , 1 , 0 , 1 , 1 ,
0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 1 , 0 , 0 , 0 , 0 ,
0 , 0 , 0 , 1 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 1 , 0 , 1 , 1 , 1 , 0 , 0 , 1 , 1 , 1 , 0 , 0 , 1 , 1 , 0 , 1 , 0 ,
0 , 0 , 0 , 0 , 0 , 1 , 0 , 1 , 0 , 1 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 1 , 0 , 1 , 1 , 1 ,
0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 1 , 1 , 1 , 0 , 1 , 0 , 0 , 1 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 ,
0 , 1 , 0 , 0 , 0 , 1 , 0 , 1 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 1 , 0 , 1 , 1 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 ,
0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 1 , 1 , 0 , 0 , 1 , 0 , 1 , 0 , 0 , 1 , 0 , 0 ,
0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 1 , 0 , 1 , 0 , 1 ,
0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 1 , 0 , 0 , 0

---

g $(d_g = 71)$

---

1 , 0 , 1 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 1 , 1 , 1 , 0 , 0 , 0 ,
1 , 0 , 1 , 0 , 1 , 0 , 1 , 1 , 1 , 0 , 1 , 1 , 0 , 0 , 1 , 1 , 1 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 ,
0 , 0 , 0 , 1 , 1 , 1 , 1 , 0 , 1 , 1 , 0 , 0 , 1 , 1 , 0 , 1 , 1 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 ,
0 , 1 , 0 , 0 , 0 , 0 , 0 , 1 , 1 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 1 , 1 , 1 , 0 , 0 , 0 , 1 ,
0 , 1 , 1 , 0 , 0 , 1 , 1 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 1 , 0 , 1 , 1 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 ,
0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 1 , 0 , 0 , 0 , 0 , 1 , 0 , 1 , 1 , 0 , 0 , 0 , 0 , 1 , 0 ,
0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 1 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 ,
0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 ,
0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 1 , 0 , 0 , 1 , 1 , 0 , 1 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0

F

-1 , 4 , -1 , 1 , -1 , 0 , -1 , 1 , -4 , 5 , -3 , 3 , 1 , 1 , 0 , -1 , 0 , 3 , 5 , 2 , 0 , 2 , -3 , 1 , -1 , 0 , 3 , -2 , 2 , -2 , 2 , 2 , 2 , 3 , 1 , -2 , 5 , 0 , 1 , 1 , 4 , 2 , -3 , 0 , 1 , 2 , 2 , 0 , 1 , -1 , 0 , 2 , 3 , 0 , -1 , 1 , 1 , 3 , 2 , 0 , -1 , 1 , -3 , 1 , 1 , 2 , -5 , 0 , 0 , -4 , 2 , -1 , 2 , -2 , 1 , 2 , 5 , 1 , 0 , 4 , 0 , 1 , -1 , 1 , 0 , 3 , 0 , 5 , 4 , -1 , 3 , -1 , 1 , 0 , 1 , 0 , 2 , 2 , -1 , 0 , -1 , 3 , 2 , -2 , -2 , -1 , 0 , 2 , 0 , 0 , 3 , 1 , 5 , -3 , 1 , 3 , 3 , 0 , -2 , 0 , -2 , 2 , -3 , -3 , -1 , 2 , 1 , 0 , 0 , 7 , 2 , -1 , 3 , -4 , 3 , -1 , 4 , -3 , 3 , 4 , 3 , 3 , 1 , -1 , 1 , -2 , 0 , -2 , 0 , 2 , 2 , 3 , 3 , 3 , 3 , 0 , 1 , 2 , 1 , 3 , -3 , 0 , -7 , 0 , 0 , -2 , 0 , 0 , 1 , 2 , 2 , 3 , -1 , 3 , 1 , -3 , 3 , 1 , 2 , 2 , 1 , -1 , 4 , -3 , 1 , 2 , -1 , -2 , 5 , 0 , 3 , 1 , 0 , 4 , 3 , 0 , 2 , 2 , 4 , 1 , 1 , -1 , 1 , 2 , 1 , -3 , 2 , 3 , 3 , 0 , 3 , 0 , -2 , 0 , -1 , 0 , -1 , -2 , 3 , -3 , 1 , -3 , 3 , -1 , -1 , -1 , 1 , -1 , 1 , 0 , -1 , 0 , -1 , 5 , 1 , 3 , -1 , 0 , 6 , 5 , 0 , -2 , 1 , 2 , 3 , 0 , 0 , 1 , 1 , 1 , 2 , 0

G

1 , -2 , -3 , 2 , 1 , 2 , 2 , -3 , 1 , 0 , -1 , -1 , 2 , 4 , -3 , 2 , 0 , -1 , 2 , 1 , 0 , -1 , -1 , 1 , -2 , 0 , -2 , 2 , 1 , 0 , 4 , 0 , 0 , 1 , -1 , 1 , 2 , 7 , 3 , -1 , 3 , -3 , 2 , 2 , -2 , 1 , 1 , 4 , -2 , 0 , 3 , -1 , 3 , 0 , 2 , 2 , -4 , -2 , 1 , -1 , 2 , 1 , 0 , -1 , -2 , 1 , 4 , 3 , 0 , -1 , -2 , -2 , 1 , 4 , -1 , 1 , 0 , 3 , -1 , 2 , 1 , 2 , 4 , 1 , 3 , 0 , 0 , 1 , 0 , -1 , -3 , 4 , 4 , 3 , -2 , -2 , -2 , 1 , -2 , 0 , 1 , 1 , -3 , -3 , 2 , 1 , 1 , 4 , -1 , 2 , 1 , 3 , 1 , 1 , 0 , 0 , -3 , 1 , 2 , 3 , 2 , 3 , 0 , 5 , 0 , 2 , 3 , 3 , -2 , 2 , 1 , 2 , 0 , 1 , -3 , 2 , 0 , 0 , -2 , -1 , -1 , 4 , 1 , 3 , -2 , 4 , 1 , 2 , 0 , 2 , 0 , 4 , 2 , 5 , 1 , 0 , 1 , -1 , -1 , -1 , 0 , 1 , 3 , 0 , 0 , 2 , 0 , 2 , 3 , 5 , 1 , 2 , -1 , 3 , 2 , 5 , 2 , 0 , 1 , 0 , 0 , -1 , 1 , 1 , -1 , -3 , -4 , 3 , 2 , 0 , -1 , 4 , 2 , 3 , -1 , 1 , -1 , -1 , -2 , 0 , 2 , 2 , 4 , 0 , 0 , 2 , 1 , 3 , -3 , -1 , 0 , 2 , 4 , -1 , 0 , 1 , -1 , 1 , 2 , 0 , 4 , -2 , 0 , -4 , 0 , 2 , 0 , -1 , 4 , 0 , 0 , -3 , 1 , 0 , 1 , 2 , 3 , -3 , 2 , 2 , 2 , 2 , 3 , -1 , 4 , 4 , 1 , 0 , 5 , 2 , 2 , 0

The correspondent public key $h = f^{-1} * g(\text{mod } q)$ is

$h \equiv f^{-1} * g \ (\text{mod } q)$
-23 , 36 , -50 , -28 , -4 , -17 , 14 , -16 , -40 , -4 , 40 , -39 , 1 , 14 , -55 , 8 , -62 , -42 , -21 , 6 , -49 , 64 , -63 , 9 , 35 , 18 , -44 , -14 , -2 , -17 , 5 , -4 , -7 , -30 , 49 , 27 , 62 , -28 , 46 , -15 , -16 , 41 , 42 , -53 , -22 , -42 , -29 , 15 , -24 , 37 , -52 , 39 , -23 , 56 , 43 , 53 , -22 , 50 , 37 , -51 , 60 , -31 , 52 , -16 , -34 , -5 , 37 , -61 , -5 , -50 , -3 , 61 , 40 , -42 , 25 , -57 , 20 , -45 , -1 , 36 , -6 , 62 , 17 , 54 , 32 , -55 , 52 , 16 , 12 , -49 , -30 , 2 , -30 , -62 , -34 , -27 , 15 , 25 , 22 , -37 , 31 , 64 , 49 , 56 , -10 , -15 , 1 , -43 , 18 , -63 , -16 , -29 , 6 , -4 , 11 , 34 , -61 , -47 , 22 , 15 , 47 , 14 , -18 , 6 , -36 , 43 , 26 , 34 , -39 , 19 , 25 , -60 , 28 , -16 , -12 , 39 , -35 , 38 , -43 , 2 , 8 , 24 , -18 , 12 , 20 , 26 , -16 , 3 , 15 , -7 , 32 , -38 , -28 , 41 , 45 , 8 , 0 , 57 , 29 , 1 , 6 , 23 , -18 , 24 , 48 , 38 , -36 , 17 , -33 , 60 , 30 , 43 , -38 , -56 , 38 , -33 , -24 , 3 , 58 , -10 , 56 , -37 , 4 , -17 , 62 , 23 , 57 , -52 , 5 , 19 , 64 , -41 , 34 , 45 , -23 , 21 , 55 , -29 , -7 , 49 , 19 , 9 , -41 , -14 , 10 , -46 , 57 , -49 , 17 , -22 , -31 , -25 , 36 , -12 , -9 , 10 , -31 , 58 , -20 , 13 , 55 , 25 , 47 , -36 , 44 , -61 , -25 , 11 , -21 , -6 , 8 , -61 , -45 , 48 , -52 , 12 , 52 , 30 , -12 , -2 , -59 , -22 , 48 , -58 , -26 , -52 , -22 , 1 , -49 , 19 , 29 , 0

Let the message m_1 and m_2 to be signed be

m_1
26 , 8 , 30 , -48 , 64 , -10 , 3 , 41 , -41 , 14 , 51 , -31 , 62 , 19 , 40 , -14 , 49 , -12 , -59 , -24 , 7 , -47 , -37 , 22 , -61 , -29 , -48 , 17 , 41 , 64 , 2 , 2 , 8 , -32 , 18 , 7 , 22 , -43 , -16 , 46 , 36 , -29 , -50 , 33 , 54 , 54 , -46 , 39 , -22 , -40 , -50 , 50 , -22 , -22 , 8 , -18 , 13 , 24 , 63 , -10 , 24 , 1 , 56 , -33 , 33 , 10 , 39 , -10 , 32 , -42 , -28 , 4 , -7 , -14 , -28 , -17 , -24 , -9 , -42 , 19 , 16 , -27 , 5 , 58 , 15 , -51 , -25 , -36 , 37 , -26 , 18 , -3 , 40 , 10 , 28 , 8 , -44 , 2 , 63 , 53 , 25 , -29 , -8 , -46 , 21 , 28 , 1 , 62 , -45 , 24 , 17 , 36 , 61 , -43 , 30 , 12 , -29 , -60 , 40 , -57 , -21 , -6 , 4 , -45 , -61 , -32 , 27 , -40 , 35 , 26 , -52 , -5 , 61 , 4 , 13 , 18 , -32 , -50 , 16 , -12 , 38 , -31 , -41 , 34 , -9 , 53 , -19 , 26 , 58 , -43 , 33 , -27 , 15 , -27 , -8 , 19 , 5 , -45 , 43 , -25 , 46 , 55 , 35 , 42 , -5 , -17 , -4 , 27 , -3 , -52 , -50 , -30 , -19 , -26 , -60 , 36 , -38 , -15 , -3 , -44 , 7 , -35 , -7 , -43 , 3 , 50 , 40 , -56 , -60 , 19 , -17 , 50 , 9 , -47 , 28 , -61 , 1 , -41 , 31 , 62 , -28 , 45 , -32 , 17 , -45 , -28 , -12 , -19 , 22 , 49 , 2 , -36 , -50 , 59 , -14 , 18 , 45 , -39 , 26 , 49 , 44 , -56 , 35 , -11 , -38 , -2 , -7 , 28 , 22 , -41 , 26 , 58 , -60 , 58 , 10 , -41 , -34 , 63 , 5 , 53 , 47 , -58 , -47 , 62 , -63 , 3 , 15 , 46 , 29 , -24 , 31 , 0

m_2
9 , -15 , 1 , 63 , 12 , 64 , -9 , -25 , 21 , 15 , -64 , 15 , 20 , 59 , -40 , 43 , -40 , -41 , -16 , -51 , -58 , -9 , -34 , -61 , -7 , 34 , 19 , -26 , -1 , 60 , -59 , -57 , -20 , 6 , -59 , 56 , 5 , -3 , -33 , -38 , -53 , -33 , 41 , 31 , -39 , -63 , 10 , -14 , -40 , 59 , 0 , -34 , -15 , 30 , -30 , 42 , 0 , 53 , -48 , 63 , 48 , -43 , -58 , -36 , 28 , -53 , -45 , -32 , 9 , -13 , -6 , 21 , 18 , -29 , -12 , 44 , -28 , 63 , -35 , -4 , 57 , 29 , 27 , -22 , -5 , 61 , -44 , 60 , 50 , -28 , 58 , 33 , -6 , 64 , 62 , -43 , -53 , -48 , -10 , 21 , 4 , 49 , -23 , -43 , -45 , 29 , -64 , -9 , 27 , -34 , 52 , 20 , 60 , 14 , 63 , -9 , 10 , -46 , -14 , -5 , -9 , -20 , -36 , 49 , -21 , -39 , -58 , -9 , -22 , -3 , -53 , 46 , -19 , -11 , -61 , 1 , -46 , -60 , 56 , 45 , -30 , 44 , 1 , -34 , -7 , -1 , 21 , -61 , 17 , -58 , -1 , -56 , -14 , 28 , 57 , 30 , 53 , 64 , -43 , -33 , -4 , -31 , -51 , 42 , 22 , -48 , -22 , 40 , -44 , -30 , 21 , -9 , -51 , -43 , 21 , 6 , 21 , -23 , 10 , -26 , -16 , -56 , -18 , 35 , 36 , -25 , 0 , 25 , -26 , 21 , 56 , 35 , 55 , -59 , 12 , 12 , -43 , 54 , -12 , -22 , -40 , -56 , 33 , -27 , -34 , -10 , 44 , 51 , 32 , -11 , -39 , -49 , -3 , 7 , 50 , -31 , 46 , -14 , 58 , -45 , -57 , 50 , 55 , 62 , 55 , 2 , 9 , -52 , -8 , 61 , -10 , 16 , -59 , -41 , 54 , -29 , 13 , 33 , -42 , -20 , -43 , -17 , -4 , 19 , 55 , -18 , 52 , 36 , 32 , 45 , 56 , 0

We now observe a valid signature (s, t) which is made by a legitimate signer.

s
26 , 26 , 40 , -43 , -52 , 0 , 16 , 38 , -37 , 29 , 47 , -9 , -41 , 43 , -56 , 4 , -60 , -12 , -51 , -12 , 21 , -40 , -34 , 45 , -43 , -16 , -9 , 28 , 53 , -51 , 8 , 0 , 3 , -12 , 24 , 1 , 33 , -44 , -4 , 59 , 52 , -25 , -51 , 36 , 58 , 57 , -33 , 29 , -13 , -50 , -42 , -59 , 2 , 0 , 18 , -16 , 28 , 32 , -52 , -4 , 36 , -4 , 58 , -20 , 55 , 39 , 41 , 8 , 46 , -37 , -4 , 6 , 14 , 6 , -22 , 3 , -17 , -1 , -19 , 20 , 37 , -19 , 11 , -53 , 36 , -52 , -36 , -27 , 45 , -17 , 44 , -3 , 61 , 28 , 30 , 14 , -42 , 14 , -60 , 61 , 16 , -34 , 12 , -41 , 40 , 36 , -11 , -54 , -34 , 30 , 49 , 37 , -59 , -48 , 55 , 29 , -11 , -45 , 50 , -41 , -16 , 9 , 21 , -46 , -37 , -48 , 46 , -34 , 47 , 56 , -34 , -9 , -30 , 23 , 39 , 22 , -29 , -36 , 7 , 5 , 33 , -24 , -33 , 40 , 0 , 41 , -6 , 30 , -60 , -45 , 27 , -15 , 31 , -12 , 11 , 23 , 15 , -25 , 32 , -6 , 43 , -55 , 32 , 42 , 17 , -10 , 15 , 34 , 21 , -44 , -38 , -10 , 0 , -9 , -61 , 54 , -26 , -9 , 6 , -33 , 14 , -26 , -3 , -29 , 35 , 53 , 60 , 63 , -40 , -5 , -5 , -63 , 16 , -26 , 28 , -43 , 2 , -22 , 47 , -52 , -33 , 56 , -32 , 18 , -36 , -20 , 7 , -9 , 48 , 55 , 17 , -14 , -27 , -32 , -14 , 29 , 49 , -26 , 36 , 53 , 53 , -38 , 52 , 6 , -18 , 20 , 19 , 37 , 33 , -28 , 32 , 64 , -49 , -53 , 10 , -21 , -30 , -57 , 15 , 47 , 57 , -58 , -43 , 54 , -61 , 6 , 25 , 54 , 35 , -16 , 56 , 0

$t = s * h \pmod{q}$

12 , 5 , 9 , -48 , -14 , -38 , 6 , -16 , 52 , 31 , 59 , 23 , 17 , -58 , -27 , -56 , -25 , -21 , 6 , -50 , -54 , 7 , -29 , -28 , -5 , 46 , 20 , -17 , 5 , -62 , -40 , -60 , -22 , 22 , -63 , -62 , 20 , 3 , -30 , -37 , -33 , -19 , 46 , 41 , -44 , -40 , 8 , 6 , -20 , -50 , 15 , -27 , 1 , 45 , -23 , 58 , 15 , -57 , -41 , -62 , 61 , -23 , -37 , -11 , 34 , -39 , -31 , -15 , 14 , 2 , 8 , 17 , 34 , -29 , 8 , 57 , -29 , -52 , -27 , 2 , 45 , 30 , 46 , -18 , 5 , -55 , -27 , -52 , 52 , -18 , -58 , 37 , 21 , -57 , -39 , -29 , -53 , -56 , -9 , 33 , 21 , -60 , -7 , -40 , -20 , 58 , -44 , -3 , 46 , -20 , 62 , 33 , -62 , 40 , -56 , -3 , 24 , -44 , 3 , 10 , -3 , -13 , -45 , 62 , -10 , -32 , -47 , -6 , -14 , 7 , -50 , -60 , -2 , 0 , -51 , 7 , -29 , -46 , -48 , 50 , -21 , 54 , 8 , 1 , 9 , 4 , 37 , -60 , 16 , -41 , 21 , -37 , -1 , 25 , 59 , 34 , -52 , -58 , -29 , -30 , -7 , -29 , -38 , -59 , 50 , -17 , -21 , 44 , -29 , -20 , 45 , 3 , -47 , -19 , 38 , 10 , 30 , 8 , 36 , -17 , 9 , -40 , -4 , 60 , 44 , -9 , 10 , 53 , -3 , 53 , -61 , 36 , -59 , -35 , 23 , 21 , -34 , -63 , -4 , -14 , -20 , -48 , 40 , -36 , -24 , 2 , 44 , -54 , 49 , -6 , -23 , -49 , 0 , 11 , -56 , -23 , 54 , 5 , -46 , -27 , -22 , 52 , -56 , -47 , 54 , 16 , 25 , -28 , 20 , -56 , 11 , 18 , -25 , -41 , -57 , -31 , 13 , 24 , -20 , -11 , -41 , -13 , 10 , 34 , -62 , -5 , -51 , 60 , 33 , 47 , -56 , 0

Obviously, the above signature (s, t) is valid and its norm value

$$\| s - m_1 \|^2 + \| t - m_2 \|^2 = 48203 \leq 90000$$

, where $\| s - m_1 \|^2 = 25335$ and $\| t - m_2 \|^2 = 22868$, respectively.

We can now generate the second signature (s', t') from previous signature (s, t) by adding annihilating polynomial $\alpha = 8 * \sum_{i=0}^{N-1} x^i$ to s.

$s' = s + \alpha \pmod{q}$

34 , 34 , 48 , -35 , -44 , 8 , 24 , 46 , -29 , 37 , 55 , -1 , -33 , 51 , -48 , 12 , -52 , -4 , -43 , -4 , 29 , -32 , -26 , 53 , -35 , -8 , -1 , 36 , 61 , -43 , 16 , 8 , 11 , -4 , 32 , 9 , 41 , -36 , 4 , -61 , 60 , -17 , -43 , 44 , -62 , -63 , -25 , 37 , -5 , -42 , -34 , -51 , 10 , 8 , 26 , -8 , 36 , 40 , -44 , 4 , 44 , 4 , -62 , -12 , 63 , 47 , 49 , 16 , 54 , -29 , 4 , 14 , 22 , 14 , -14 , 11 , -9 , 7 , -11 , 28 , 45 , -11 , 19 , -45 , 44 , -44 , -28 , -19 , 53 , -9 , 52 , 5 , -59 , 36 , 38 , 22 , -34 , 22 , -52 , -59 , 24 , -26 , 20 , -33 , 48 , 44 , -3 , -46 , -26 , 38 , 57 , 45 , -51 , -40 , 63 , 37 , -3 , -37 , 58 , -33 , -8 , 17 , 29 , -38 , -29 , -40 , 54 , -26 , 55 , 64 , -26 , -1 , -22 , 31 , 47 , 30 , -21 , -28 , 15 , 13 , 41 , -16 , -25 , 48 , 8 , 49 , 2 , 38 , -52 , -37 , 35 , -7 , 39 , -4 , 19 , 31 , 23 , -17 ,

40 , 2 , 51 , -47 , 40 , 50 , 25 , -2 , 23 , 42 , 29 , -36 , -30 , -2 , 8 , -1 , -53 , 62 , -18 , -1 , 14 , -25 , 22 , -18 , 5 , -21 , 43 , 61 , -60 , -57 , -32 , 3 , 3 , -55 , 24 , -18 , 36 , -35 , 10 , -14 , 55 , -44 , -25 , 64 , -24 , 26 , -28 , -12 , 15 , -1 , 56 , 63 , 25 , -6 , -19 , -24 , -6 , 37 , 57 , -18 , 44 , 61 , 61 , -30 , 60 , 14 , -10 , 28 , 27 , 45 , 41 , -20 , 40 , -56 , -41 , -45 , 18 , -13 , -22 , -49 , 23 , 55 , -63 , -50 , -35 , 62 , -53 , 14 , 33 , 62 , 43 , -8 , 64 , 8

$t' = s' * h = t + \alpha * h = t - \alpha \pmod{q}$

---

4 , -3 , 1 , -56 , -22 , -46 , -2 , -24 , 44 , 23 , 51 , 15 , 9 , 62 , -35 , 64 , -33 , -29 , -2 , -58 , -62 , -1 , -37 , -36 , -13 , 38 , 12 , -25 , -3 , 58 , -48 , 60 , -30 , 14 , 57 , 58 , 12 , -5 , -38 , -45 , -41 , -27 , 38 , 33 , -52 , -48 , 0 , -2 , -28 , -58 , 7 , -35 , -7 , 37 , -31 , 50 , 7 , 63 , -49 , 58 , 53 , -31 , -45 , -19 , 26 , -47 , -39 , -23 , 6 , -6 , 0 , 9 , 26 , -37 , 0 , 49 , -37 , -60 , -35 , -6 , 37 , 22 , 38 , -26 , -3 , -63 , -35 , -60 , 44 , -26 , 62 , 29 , 13 , 63 , -47 , -37 , -61 , 64 , -17 , 25 , 13 , 60 , -15 , -48 , -28 , 50 , -52 , -11 , 38 , -28 , 54 , 25 , 58 , 32 , 64 , -11 , 16 , -52 , -5 , 2 , -11 , -21 , -53 , 54 , -18 , -40 , -55 , -14 , -22 , -1 , -58 , 60 , -10 , -8 , -59 , -1 , -37 , -54 , -56 , 42 , -29 , 46 , 0 , -7 , 1 , -4 , 29 , 60 , 8 , -49 , 13 , -45 , -9 , 17 , 51 , 26 , -60 , 62 , -37 , -38 , -15 , -37 , -46 , 61 , 42 , -25 , -29 , 36 , -37 , -28 , 37 , -5 , -55 , -27 , 30 , 2 , 22 , 0 , 28 , -25 , 1 , -48 , -12 , 52 , 36 , -17 , 2 , 45 , -11 , 45 , 59 , 28 , 61 , -43 , 15 , 13 , -42 , 57 , -12 , -22 , -28 , -56 , 32 , -44 , -32 , -6 , 36 , -62 , 41 , -14 , -31 , -57 , -8 , 3 , 64 , -31 , 46 , -3 , -54 , -35 , -30 , 44 , 64 , -55 , 46 , 8 , 17 , -36 , 12 , 64 , 3 , 10 , -33 , -49 , 63 , -39 , 5 , 16 , -28 , -19 , -49 , -21 , 2 , 26 , 58 , -13 , -59 , 52 , 25 , 39 , 64 , 0

In this example, we can have the following equation exactly:

$$\begin{aligned} \| s' - m_1 \|^2 + \| t' - m_2 \|^2 &= \| s - m_1 \|^2 + \| t - m_2 \|^2 \\ &\leq 90000, \end{aligned}$$

, where $\| (s' - m_1) \|^2 = 25335$ and $\| (t' - m_2) \|^2 = 22868$, respectively.

NTRUSign 서명 알고리즘 안전성에 관한 연구

민성준

래티스 개념은 암호 분석 및 어려운 문제를 찾는 분야에 있어서 많은 암호학자들에 의하여 오랜 동안 연구 되어 왔다. 그럼에도 불구하고 래티스 이론에 기반 한 안전하고 효율적인 전자 서명 기법들의 제안은 매우 드물었다. 초기의 Goldreich, Goldwasser, 그리고 Halevi에 의하여 제안된 기법에서 서명자는 주어진 메시지의 해쉬값 으로부터 얻어진 임의의 점에 대하여 대략적인 가장 가까운 벡터 찾는 문제를 풀 수 있음을 증명함으로서 서명하게 된다. 또한 검증자의 경우에는 서명 값으로 받은 벡터가 실제로 래티스 상에 있는 점이며 메시지의 해쉬값과 충분히 가까움을 확인함으로서 서명 값을 증명하게 된다. 그러나 이러한 아이디어가 그 저자들에 의하여 상세하게 분석되지 않았었다.

같은 시기에 또 다른 공개키 암호시스템인 NTRU 암호 알고리즘이 Hoffstein, Pipher, 그리고 Silverman에 의하여 제안되었다. 그 후 그들은 CT-RSA'03 에서 NTRUSign이라 불리는 새로운 형태의 인증 및 전자 서명 기법을 소개하였다. NTRU 공개키 암호 알고리즘과 마찬가지로, NTRUSign 서명 기법은 짧고 쉽게 생성되는 키들, 매우 빠른 속도, 그리고 적은 메모리양을 요구하는 장점을 가지고 있다. 그것의 안전성 또한 NTRU 래티스라 불리는 임의의 래티스 상에서의 대략적 가장 짧은 벡터 찾는 문제에 기반하고 있다. 이 NTRUSign기법에서, 서명자는 임의의 점에 가까운 래티스 상의 점을 구하기 위하여 그의 비밀 정보를 사용한다. 이때, 가장 가까운 벡터문제의 대략적 해가 그의 서명 값이 된다.

이렇게 GGH 그리고 NTRU 에서 처럼 기존의 어려운 문제들(예를 들면, 인수분해 문제 그리고 이산대수 문제)과 다른 문제들을 찾고자 하는 데에는 크게 두 가지의 이유가 있다. 첫째, 인수분해 문제 및 이산대수 문제

들에 관하여 해결하고자하는 많은 연구들이 진행 되고 있다. 두 번째 더욱 중요한 이유는 효율성에 있다. 예를 들어, NTRU 기반의 알고리즘들은 다른 암호알고리즘들과 비교하여 같은 안정성을 제공하면서 그들보다 수백배 이상이 빠르다고 알려져 있다. 그러나 이러한 어려운 문제에 기반 하는 알고리즘들의 결점은 아직까지 그 안전성에 관하여 잘 연구되어지지 않았다는데 에 있다. 비록 래티스 이론이 100년 이상동안 연구 되어온 것은 사실이지만, 가장 짧은 벡터 문제와 같은 어려운 래티스 문제들의 특성들은 Lenstra, Lenstra, 과 Lovász 들이 1982년에 다항식 시간 래티스 기저 축소 알고리즘(polynomial-time lattice basis reduction algorithm)을 발견한 후에야 비로소 집중적으로 연구되었다. 더욱이, NTRU-기반의 기법들은 다항식 환에 기반 하는 특정 형태의 래티스들을 사용하고 있으며, 이러한 래티스들은 일반적인 래티스 문제들보다 해결하기에 더 쉬울지도 모르는 특정 형태의 래티스 문제들을 만들어 낸다. 이러한 특정 형태의 래티스 문제들은 단지 1996년 NTRU암호알고리즘의 소개 이후부터 활발한 연구가 되어왔기 때문에, 우리는 매우 유용한 새로운 결과들을 기대할 수 있다.

본 논문에서, 첫째 우리는 NTRUSign 서명 알고리즘에 대한 공격 방법을 제안한다. 우리가 제안하는 공격 기법을 통하여 타당한 메시지-서명 쌍만을 관찰하는 수동적 공격자로 하여금 비밀 키를 모르고도 또 다른 서명 값을 만들어 내는 것을 가능하게 한다. 그러므로 NTRUSign 서명 알고리즘은 유연성(malleability) 성질을 가지고 있다는 점에서 안전한 서명 알고리즘이 아니다. 이 유연성의 성질로부터, 우리는 임의의 메시지-서명 쌍 을 가지고 그 메시지에 대한 제 2의 서명 값을 유도해 낼 수 있다. 이 경우, 우리는 두 번째의 서명 값과 비밀 키를 알고 있는 사용자에 의하여 생성된 원래의 서명을 구분 할 수 없다. 비록 이러한 취약점이 공격자로 하여금 메시지 스트링을 바꾸는 것을 허락하지는 않지만, 이러한 종류의 위조는 그 서명 알고리즘이 모든 응용서비스에서 사용되어질 수 있는 것은 아니다. 예를 들어, 전자화폐의 경우에 있어서, 화폐에 대한 두 번째 타당한 서명 값을 찾아내는 것은 불가능 해야만 한다. 또한, 임의의 사용자가 $s \neq s'$ 을 만족하는 메시지-서명 쌍들 (m, s) 과 (m, s')을 동시에 받았을 경우, s 뿐만 아니라 s' 도 메시지 m 에 대한 타당한 서명 값으로서 결코 받아들여지지 않

을 것이다. 만약 합법적인 서명자가 메시지 m에 대한 서명 s는 그의 서명 값이라는 것을 주장하고 싶을 경우, 그는 그의 비밀 키를 드러내야만 한다.

다음으로, 우리는 NTRUSign 서명 알고리즘이 가지고 있는 이런 취약점을 피하기 위한 기법을 제공한다. 비록, 수정된 NTRUSign은 앞에서 제안한 공격으로부터 안전하지만, 그것이 유연하지 않다는 사실을 아직 증명하지 못하였다. 실제로, 제안한 수정 NTRUSign이 유연성을 회피하는 서명 알고리즘이라는 것을 증명하는 것은 앞으로 우리가 해결해야하는 하나의 과제로서 남아있다.

References

1. M. Ajitai, "The Shortest Vector Problem in L_2 is NP-Hard for Randomized Reductions", Proceedings 30th Annual ACM Symposium on Theory of Computing, 1998.

2. H. Cohen, *A course in computational algebraic number theory*, GTM 138, Springer-Verlag, 1993.

3. L. Granboulan, "How to repair ESIGN", SCN'02, LNCS, Vol.2576, Springer-Verlag, pp.234-240, 2003.

4. C. Gentry, "Key recovery and message attacks on NTRU-composite", Advances in Cryptopogy-Eurocrypt '01, LNCS, Springer-Verlag, 2001.

5. C. Gentry, J. Jonsson, J. Stern, and M. Szydlo, "Cryptanalysis of the NTRU Signature Scheme (NSS) from Eurocrypt '01" Advances in Cryptology-Asiacrypt '01, LNCS, Vol.2248, Springer-Verlag, pp.123-131, 2001.

6. C. Gentry, and M. Szydlo, "Cryptanalysis of the Revised NTRU Signature Scheme", Advances in Cryptology-Eurocrypt '02, LNCS, Vol.2332, pp.299-320, Springer-Verlag, 2002.

7. O. Goldreich, S. Goldwasser, and S. Halevi, "Public-key cryptography from lattice reduction problems", Advances in Cryptology-Crypto '97, LNCS, pp.112-131, Springer-Verlag, 1997.

8. S. Goldwasser, and M. Bellare, *Lecture Notes on Cryptography*, 2001.

9. S. Goldwasser, S. Micali, and R. Rivest, "A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks", SIAM Journal of Computing, pp.281-308, 1998.

10. J. Hoffstein, J. Pipher, and J. Silverman, "Enhanced Encoding and Verification Methods for the NTRU Signature Scheme", NTRU Technical Note #017, 2001. Available from http://www.ntru.com.

11. J. Hoffstein, N. Graham, J. Pipher, J. Silverman, and W. Whyte, "NTRUSign: Digital Signatures Using the NTRU Lattice Preliminary Draft 2", Available from http://www.ntru.com.

12. J. Hoffstein, N. Graham, J. Pipher, J. Silverman, and W. Whyte, "NTRUSign: Digital Signatures Using the NTRU Lattice", CT-RSA'03, LNCS, Vol.2612, Springer-Verlag, pp.122-140, 2003.

13. J. Hoffstein, J. Pipher, and J. Silverman, "NTRU: A Ring-Based Public Key Cryptosystem", in Algorithmic Number Theory (ANTS III), LNCS, Vol.1423, Springer-Verlag, pp.267-288, 1998.

14. J. Hoffstein, J. Pipher, and J. Silverman, "NSS: An NTRU Lattice-Based Signature Scheme", Advanced in Cryptology-Eurocrypt '01, LNCS, Vol.2045, Springer-Verlag, pp.123-137, 2001.

15. A. Joux and G. Martinet, "Some Weaknesses in Quartz Signature Scheme", NESSIE public reports, NES/DOC/ENS/WP5/026/1, 2003.

16. H. Koy, and C.-P. Schnorr, "Segment LLL-Reduction of Lattice Bases", Cryptography and Lattices Conference-Proceedings of CaLC '01, LNCS, Springer-Verlag, pp.67-80, 2001.

17. H. Koy, and C.-P. Schnorr, "Segment LLL-Reduction with Floating Point Orthogonalization", Cryptography and Lattices Conference-Proceedings of CaLC '01, LNCS, Springer-Verlag, pp.81-96, 2001.

18. A. May, and J.H. Silverman, "Dimension reduction methods for convolution modular lattices", Cryptography and Lattices Conference-Proceedings of CaLC '01, LNCS, Springer-Verlag, pp.110-125, 2001.

19. A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

20. I. Mirinov, "A note on cryptanalysis of the preliminary version of the NTRU signature scheme", IACR preprint server, Available from http://eprint.iacr.org/2001/005/.

21. T. Okamoto, E. Fujisaki, and H. Morita, "TSH-ESIGN: Efficient Digital Signature Scheme Using Trisection Size Hash (Submission to P1363a)", 1998.

22. D. Pointcheval, and J. Stern, "Security Proofs for Signature Schemes", Advances in Cryptology-Proceedings of Eurocrypt '96, LNCS, Vol.1070, Springer-Verlag, pp.387-398, 1996.

23. C.-P. Schnorr, "A more efficient algorithm for lattice basis reduction", J. Algorithms 9, pp.47-62, 1998.

24. C.-P. Schnorr, and M. Euchner, "Lattice basis reduction: improved practical algorithms and solving subset suc problems", Math. Programming 66, no. 2, pp. 181-199, 1994.

25. J. Silverman, "Wraps, Gaps and Lattice Constants" NTRU Technical Report #011, 2001, Available from http://www.ntru.com.

26. J. Stern, "A new identification scheme based on syndrome decoding", Advances in Cryptology-Crypto '93, LNCS, Springer-Verlag, pp.13-21, 1994.

27. J. Stern, "Designing identification schemes with keys of short size", Advances in Cryptology-Crypto '94, LNCS, Springer-Verlag, pp.164-173, 1994.

28. J. Stern, D. Pointcheval, J. Lee, and N. Smart, "Flaws in Applying Proof Methodologies to Signature Schemes", Advances in Cryptology-Crypto'02, LNCS, Vol.2442, Springer-Verlag, pp.93-110, 2002.

29. D. Stinson, *Cryptography: Theory and Practice*, CRC Press, 1997.

30. Consortium for Efficient Embedded Security. *Efficient Embedded Security Standard (EESS)#1: Implementation Aspects of NTRUEncrypt and NTRUSign.* Available from http://www.ceesstandards.org.

31. Consortium for Efficient Embedded Security. *Efficient Embedded Security Standard (EESS)#1: Draft 2.0.* Previously on http://www.ceesstandards.org.

Acknowledgements

First, I would like to express my sincere gratitude to Prof. Kwangjo Kim, my academic advisor, for his constant direction and support. He always has shown his consistent affection and encouragement for me to carry out my research and life in ICU. Special thanks also goes to Prof. Jae Choon Cha and Dr. Dae Sung Kwon for their generosity and agreeing to serve as committee members of my thesis.

I want to extend my thanks to Prof. JinYong Kim, my supervisor in KyungHee Univ., for always heartening me. And I am grateful to Pro. Nam Kyun Kim at Hanbat National Univ. and my senior Kisik Chang at Korea Univ. for their encouragement.

Thanks to all members of cryptology and information security laboratory: Dr. Xiaofeng Chen, Vo Duc Liem, Lv Jiqiang, Ren Kui, Yan Xie, Wang Ping, Munirathnam M. Divyan, ByeongKon Kim, SongWon Lee, HwaSun Chang, Chuljoon Choi, Sangwon Lee, JaeHyrk Park, JoongMan Kim, SuGil Choi, Zeen Kim, KyuSuk Han, SeokKyu Kang and JeongKyu Yang. They gave me lots of interests and good advices during the course of my study.

Most of all, I should mention my father and mother for their endless concerns and devotional affection. I cannot forget their trust and encouragement on me. My sister and her husband also have given me warmhearted concerns. I hope God bless my family and to be happy.

Curriculum Vitae

Name : SungJun Min

Date of Birth : Dec. 23. 1973

Sex : Male

Nationality : Korean

Education

1992.3–1999.2	Mathematics KyungHee University (B.S.)
1999.3–2001.2	Mathematics KyungHee University (M.S.)
2002.2–2004.2	Cryptology and Information Security, Engineering Information and Communications University (M.S.)

Career

2002.4–2003.2	Graduate Research Assistant Development of IT Manpower The Ministry of Information and Communications (MIC).

2002.4–2002.12 Graduate Research Assistant
Development of secure electronic trading system for online game items
Semtlo Inc.

2002.6–2003.1 Graduate Research Assistant
A Study on Public Key Cryptosystem based on Noncommutative Groups
Korea Information Security Agency(KISA).

2003.1–2003.12 Graduate Research Assistant
Research on Link Security Technology
Electronics and Telecommunications Research Institute(ETRI)

2003.2–2003.6 Undergraduate Teaching Assistnat
ICE0100 University Mathematics
School of Engineering, ICU

2003.3–2004.2 Graduate Research Assistant
Support for running the International Research Center for Information Security
The Ministry of Information and Communications (MIC).

2003.6–2003.8 Apprentice Researcher
Information Sharing Platform Laboratories(ISL), NTT, Japan

Publications

(1) 2003.09 ByeongKon Kim, SungJun Min, and Kwangjo Kim, "Fair tracing based on VSS and blind signature without Trustees", *Proc.* of CSS '03, pp.37-42, 2003.

(2) 2003.12 SungJun Min, Go Yamamoto, and Kwangjo Kim "Analysis of NTRUSign signature scheme", 2003년도 한국정보보호학회 동계학술대회, *Proc.* of CISC '03, pp.399-405, 2003.

(3) 2004.01 SungJun Min, Go Yamamoto, and Kwnagjo Kim "On the security of NTRUSign signature scheme", to appear in the Proc. of SCIS '04.