Comparison and intelligent analysis of NTRU and ETRU signature algorithms for public key digital signature

Zifeng Zhu*, Fei Tian

School of Mathematics and Science, Beihang University, Beijing, China, 102206

*Corresponding author: zifengzhu2019@buaa.edu.cn

Abstract. Three American mathematicians made the NTRU public-key cryptosystem in 1996, it has a fast speed, small footprint, and also it is easy to produce key advantages. The NTRU signature algorithm is based on an integer base, the performance of the signature algorithm will change when the integer base becomes other bases. Based on the definition of "high-dimensional density" of lattice signatures, this paper chooses the ETRU signature algorithm formed by replacing the integer base with the Eisenstein integer base as a representative, and analyzes and compares the performance, security of NTRU and ETRU signature algorithms, SVP and CVP and other difficult issues, the speed of signature and verification, and the consumption of resources occupied by the algorithm.

Keywords: NTRU; ETRU; signature algorithm.

1. Lattice signature algorithm

1.1. Overview of digital signatures

Digital signature, also known as public key digital signature, is a digital string that can be generated only by the sender of the information and can effectively prove the authenticity of the sent information. This digital string cannot be forged by others. Similar to ordinary physical signatures written on paper, digital signatures are a concept based on cryptography, which uses specific technical means to mark data messages with specific marking information, which serves as the sign of the sender of the data message. It also indicates that the sender approves the content of the data message.

As one of the important concepts in the development of public key cryptography, digital signature can provide security that is difficult to achieve with other encryption algorithms. In general, a digital signature must have the following characteristics:

1. Must be able to verify the date and time of the author and his signature;
2. Must be able to verify the content at the moment of signing;
3. The signature must be able to be verified by a third party to resolve disputes;

1.2. NTRU signature system

(1) NTRU signature principle

Based on the NTRU algorithm, we can construct a new digital signature, the principle is as follows:

1. In a CS lattice, finding the shortest non-zero vector is a difficult problem;
2. In a CS grid, finding the vector closest to the known vector (this vector is not necessarily in the CS grid) is one of the difficult problems:

3. If a short vector in the CS grid has been found, it is easy to find the vector closest to the given vector in the grid.

The signature process is as follows: first calculate the hash code of the signed file, and then map it into a vector \( m \). Signer using known the CS a short cell in the vector, with this signature algorithm CS found from this vector lattice \( m \) close enough to a vector of \( s \), the \( s \) as the signature of this document.

(2) Signature algorithm

First construct a non-linear function. Define the parameters \(( N,p,q )\) as \(( 251,256,3 )\), a \(( N,128 )\) extension function \( t=r(t_0,t_1,t_2,...,t_{N-1})\) is like this A type of function: It determines 128 positions for a 128-dimensional vector \( t \), and fills in the components of the 128-dimensional vector \( r \) to be expanded into the selected 128 positions at a time. The remaining positions are naturally Both are 0, and this new vector \( t \) is the result of the expansion.

1. Key generation

Three polynomials are randomly selected: \( f(X) \in L(31,30) \), \( g^{(1)}(x) \in L(30,30) \), \( g^{(2)}(x) \in L(30,30) \). Check whether \( f(X) \) is reversible in \( Z_q[X]/(X^{N-1}) \). If it is not reversible, select again. If it is reversible, store \{ \( f,g^{(1)},g^{(2)} \) \}. We define \( f_q^{-1} \) and \( f_p^{-1} \) as the inverse of \( f \) on the polynomial ring \( Z_q[X]/(X^{N-1}) \) and \( Z_p[X]/(X^{N-1}) \). Randomly select three \((N,128)\) extension functions \( t^{(1)} \in T(N,128) \), \( t^{(2)} \in T(N,128) \), \( t^{(3)} \in T(N,128) \). Calculate the following formula: \( h^{(1)}=\{ f_q^{-1} g^{(1)} \}(modq), H^{(1)}=\{ f_p^{-1} g^{(1)} \}(modp) \), \( h^{(2)}=\{ f_q^{-1} g^{(2)} \}(modq), H^{(2)}=\{ f_p^{-1} g^{(2)} \}(modp) \). Save \( \{ h^{(1)},H^{(1)},h^{(2)},H^{(2)} \} \) as the public key. And store \{ \( f,g^{(1)},g^{(2)},t^{(1)},t^{(2)},t^{(3)} \) \} as the private key.

2. Signature algorithm

The hash code of the signature file is mapped into two polynomials \( m^{(1)} \) and \( m^{(2)} \) on the polynomial ring \( Z_p[X]/(X^{N-1}) \) in a certain way. Here, with the MD5 hash function is calculated hash code 16 hexadecimal 32-character string, and then converting it to factor modulo the polynomial polynomial ring.

Randomly selecting a 128-dimensional vector \( R \), containing 1 and -1 ,which number both are D, the other component is 0, define \( t^{(1)}=t^{(1)}(r) \), \( t^{(2)}=t^{(2)}(r) \), \( t^{(3)}=t^{(3)}(r) \). Then we calculate:

\[
\begin{align*}
    s^{(1)} &= \{ f(p^{-1} m^{(1)})(modp) \} + pt^{(1)} + pq^{(2)} t^{(2)} \\
    s^{(2)} &= \{ f(p^{-1} m^{(2)})(modp) \} + pt^{(3)} - pq^{(1)} t^{(2)}
\end{align*}
\]

And check if there are:

\[
\begin{align*}
    s^{(1)}(modq) &= s^{(1)} \\
    s^{(2)}(modq) &= s^{(2)} \\
    (s^{(1)} h^{(1)} + s^{(2)} h^{(2)})(modq)(modp) &= (H^{(1)} m^{(1)} + H^{(2)} m^{(2)})(modp)
\end{align*}
\]

If the above formulas are all true, then we think that \( s^{(1)}, s^{(2)} \) are valid signatures, and send \{ \( m^{(1)},m^{(2)},s^{(1)},s^{(2)} \) \} to the file receiver Party, as a valid signature. Otherwise, we will re-select \( r \) until the verification can be passed.

3. Verification algorithm

When the receiver receives the file and signature \{ \( m^{(1)},m^{(2)},s^{(1)},s^{(2)} \) \} from the sender, the receiver only needs to verify whether the following five-step equation is true or not. Determine whether the signature is valid

\[
\begin{align*}
    s^{(1)}(modq) &= s^{(1)} \\
    s^{(1)}(modp) &= m^{(1)} \\
    s^{(2)}(modq) &= s^{(2)} \\
    s^{(2)}(modp) &= m^{(2)} \\
    (s^{(1)} h^{(1)} + s^{(2)} h^{(2)})(modq)(modp) &= (H^{(1)} m^{(1)} + H^{(2)} m^{(2)})(modp)
\end{align*}
\]
1.3. **ETRU signature system**

Ai Sunstein integer of the $Z [W] = a + bw$, where $W^3 = 1$, $W = 1/2 (-1 + i \sqrt{3})$, $a, b \in Z$. For $q = a + bw$, we have $|q|^2 = a^2 + b^2 - ab$.

The cryptosystem NTRU extend from the base to an integer Ai when Sunstein integer base, we said the new password system is ETRU. The key generation, encryption and decryption algorithm, and signature verification algorithm here are the same as NTRU. So we won't repeat it.

2. signature security analysis

2.1. Attacking the private key

(1) **NTRU**

The attacker tries to find a secret key or backup secret key that can be used for decryption. We can easily see that the cyclic shift polynomial of $f$ will do the same work as the secret key, that is, it can be used for decryption. Any pair satisfying $g' = f' \cdot h \mod q$ and the inverse of $f'$ modulo $p(f', g')$ can be used as a backup key.

(2) **ETRU**

Similar to the NTRU discussed, ETRU private key any cyclic shift of the polynomial $f'$ are as $f$ as doing the same job, mF any cyclic shift of the polynomial can also do this, wherein $\mu \in \{\pm 1, \pm w, \pm w^2\}$.

2.2. Violent search

(1) **NTRU**

The attacker tries to find the secret key for random polynomial by searching all possibilities. Therefore, the key security of a cryptographic system is

$$\min \{ |L_g|, |L_f| \} = \min \left\{ \frac{N!}{d_q!^2(N-2d_q)!}, \frac{N!}{d_f!^2(N-2d_f)!} \right\}$$

the message security is $|L_\emptyset| = \frac{N!}{d_q!^2(N-2d_q)!}$.

(2) **ETRU**

The attacker can search all possible $f \in L_f$ and check whether the coefficient array of $f\cdot h \mod q$ is a small array, or search all possible $g \in L_g$ and check whether the coefficient array of $g\cdot h^{-1} \mod q$ is a small array To launch an attack. Therefore, critical safety

$$\min \{ |L_g|, |L_f| \} = \min \left\{ \frac{N!}{d_q!^6(N-6d_q)!}, \frac{N!}{d_f!^6(N-6d_f)!} \right\}$$.

2.3. Encounter and attack halfway

(1) **NTRU**

In the form $f' = f_1 || f_2$ to $f'$ ($f'$ to $f$ cyclic shift polynomial), where $f_1$ and $f_2$ are of length $N/2$, comprising $f$ polynomial half of the non-zero entries. When $p=3$, the coefficients of $f$ are $1, -1$ and $0$ respectively. Look for $f'=f_1 || f_2$, if $f\cdot h \mod q$ is ternary, then the attack is successful. Critical safety is

$$\sqrt{\frac{N!}{d_f!^2(N-2d_f)!}}$$

(2) **ETRU**

The basic idea of this attack is the same as NTRU. Through the analysis of NTRU, it can be known that the midway encounter attack reduces the key search space approximately to the original square root size. Therefore critical safety is

$$\sqrt{\frac{N!}{d_f!^6(N-6d_f)!}}.$$
2.4. Lattice attack

(1) NTRU

This attack is based on the lattice SVP problem, and the private key of the NTRU cryptosystem can be obtained. The row of $2N \times 2N$-dimensional matrix of \[
\begin{pmatrix}
I & H \\
0 & qI
\end{pmatrix}
\] may be generated NTRU lattice, where balance parameters $\lambda \in \mathbb{R}$ is a nonzero constant, $H$ is a circulant matrix, where $h = \sum_{i=1}^{N+1} h_i x_i$ is the NTRU public key, the I is $N \times N$ matrix.

Since the coefficients of the polynomials $f$ and $g$ have norms of 0 or 1, the vector $(\lambda f, g)$ will be in the short vector in the lattice. This shows that if the attacker can find a vector of the form $(\lambda f, g)$, then the lattice attack can be carried out. Through the LLL algorithm, we can find the shortest vector $(\lambda f', g')$ of the lattice, and then find $(f', g')$ (satisfying $f' \cdot h \mod q = g'$). In order to maximize the efficiency of searching for small vectors in the lattice, here we choose the balance parameter $\lambda = \frac{||d||}{||f||}$ (when $d = d$, take $\lambda = 1$).

(2) ETRU

Similar to NTRU, our target vector is $(\lambda f, g)$, and the length of this vector is $\tau$. Let $s$ be the length of the desired shortest vector, $\lambda$ be the balance parameter, where $\tau = \sqrt{\lambda^2||f||^2 + ||g||^2}$, $s = \sqrt{\frac{2N||q||\lambda}{\pi \epsilon}}$, in order to maximize the value of $s / \tau$, we choose $\lambda = \frac{||d||}{||f||}$.

When substituting the data here for calculation, we choose the ETRU with the parameter $(N, q, p = 2, r = 2/3)$ and the parameter $(N = 2N, q', p' = 3, r' = 2/3)$ of the NTRU ( $q' = 8/3 | q |$ ).

We have noticed that under the specified circumstances of each security level, the combined security of NTRU far exceeds its target security parameters. Although ETRU has a smaller key space and lower combined security (mainly due to the smaller value of $N$), the combined security maintained by ETRU far exceeds the specified value. Therefore, we conclude that ETRU is safe and can resist brute force search and midway encounter attacks at this level. For lattice attacks, we can get that, compared with NTRU, ETRU has a tighter lattice structure, stronger lattice security, and takes up less storage space.

In general, for our selected ETRU and NTRU, the security of ETRU is the same as or greater than that of NTRU.

3. Comparison of algorithm signature and verification speed

Each signature requires the user to execute Signature algorithm and Verification algorithm. This paper do not consider the consumption of integer addition, and calculate the time complexity of signature and verification in ETRU and NTRU.

NTRU ( $N' = 2N$ ) and ETRU have the same cost in calculating the sum of polynomials. The convolution of two polynomials of degree $n-1$ nominally consumes $n$. Thus consumption NTRU $N^2 \sim 4N^3$ and ETRU consumed with $3N^3$. For the modulo $q$ operation, the cost in $Z[w]$ is $27$ times that in $Z$, so in the calculation of the modulus of the reduced polynomial coefficient, the ETRU cost is $27N$, and the NTRU cost is $N \sim 2N$. But for mold $p = 2$ operation, Ai Sunstein polynomial modulo operation reducing consumption $2N$.

Therefore, the time spent in the NTRU signature process is $10*4N^2 + 7N^2 \sim 40N^2 + 14N$ (when $N = 2N$), while ETRU only needs to consume $10*3N^2 + 3*27N + 4*2N \sim 30N^2 + 89N$; the time spent in the NTRU verification process is $4*4N^2 + 7N^2 \sim 16N^2 + 14N$ ( when $N = 2N$ ), while ETRU only needs to consume $4*3N^2 + 3*27N + 4*2N \sim 12N^2 + 89N$. This complexity is essentially independent of the size of $q$.

When $N$ is large, the speed of ETRU is significantly greater than that of NTRU.

4. Comparison of resource consumption occupied by algorithms

Observing the NTRU and ETRU signature and verification algorithms, we find that they are all composed of four basic operations: addition, multiplication, modulo operation, and inversion operation.

In order to simplify the calculation, we only analyze the resource consumption (comple Xity) of these
four operations. The following defines the complexity of integer addition and subtraction as $A$, the complexity of integer multiplication, squaring, and modular operations as $M$, the resource consumption as $C$, and the relative resource consumption, \( \eta = \frac{C[ETRU(N,q,p=2,r)]}{C[NTRU(N=r=2N,q,t,p=3,r=r)]} \).

### 4.1. Addition

It only appears in the addition of polynomials of degree $N-1 (N'-1)$. From $N'=2N$, \( \eta = \frac{C[(a+b\omega)+(c+d\omega)]}{2C(m+n)} \), \((a+b\omega)+(c+d\omega)=(a+c)+(b+d)\omega\) is the addition of two numbers on $\mathbb{Z}[\omega]$ (complexity $2A$), $m+n$ is the addition of two numbers on $\mathbb{Z}$ (complexity $A$), so \( \eta = \frac{2A}{2A} = 1 \), the addition resource consumption is the same.

### 4.2. Multiplication

Appears in the multiplication of two numbers, the multiplication of a number and a polynomial of degree $N-1 (N'-1)$, and the multiplication of a polynomial of degree $N-1 (N'-1)$ and a polynomial of degree $N-1 (N'-1)$, \( \eta > 1 \) for the multiplication of two numbers, \( \eta > 1 \) for the multiplication of numbers and polynomials, and \( \eta < 1 \) for the multiplication of polynomials and polynomials. It can be seen that the polynomials in ETRU and NTRU have different degrees of Multiplication and multiplication of numbers and polynomials consume ETRU resources, and multiplication of polynomials and polynomials NTRU consumes resources.

### 4.3. Modular operation

Since the computational complexity of a polynomial modulus polynomial is based on the polynomial modulus integer, here we only analyze the resource consumption of the polynomial modulus integer. The resource consumption of integer modulus $p'$ and $q'$ on $\mathbb{Z}$ is $M$, and the resource consumption of EISENSTEIN integer modulus $p$ and $q$ on $\mathbb{Z}[\omega]$ is $2M$. From $N'=2N$, \( \eta = \frac{C[(a+b\omega) mod(p)]}{2C(m mod p')} \), \( \eta = \frac{2M}{2M} = 1 \), we can see that the polynomial modulus Integer operations ETRU and NTRU consume the same resources.

### 4.4. Inversion operation

Since the complexity of the polynomial inversion operation is based on the inversion of an integer (EISENSTEIN integer), only the inversion operation of an integer (EISENSTEIN integer) is considered here. Since $p$ and $p'$ are small, we adopt the idea of traversal, in $\mathbb{Z}$ Given the relatively prime $m$ and $p'=3$, traverse $k$ through 1, and check whether $k\times m=1$ (mod 3) is established, so the maximum resource consumption is $2\times(2M)$ (multiplication and modular operation are combined Consume $2M$). Given relatively prime $a+b\omega$ and $p=2$ on $\mathbb{Z}[\omega]$, traverse $q+t\omega$ 1,0,1+o, and check whether \((a+b\omega) * (q+t\omega) = 1 \mod 2\) is established, so the maximum resource consumption is $3\times(5M+4A)$, where the resource consumption ETRU of the integer (EISENSTEIN integer) inversion operation is greater than the NTRU.

### 5. Difficult problems in the pattern

#### 5.1. Definition of difficult problems

- **The shortest vector problem (SVP)**: Given a set of basis for the lattice $L$, look for a non-zero vector $u$ in the lattice so that $||u||=||L||$ (Approximate shortest vector problem (APPR-SVP)): Given a set of basis shortest vector problem (APPR-SVP): Given a set of basis of lattice $L$, find a non-zero vector $u \in L$ so that $||u|| \leq f(d)||L||$, where $f(d)$ is some approximate factor related to dimension.

The nearest vector problem (CVP): Given a set of basis and vector of lattice $L$, not necessarily in $L$, look for a vector $u \in L$ so that for every $w \in L$, there is $||u-v|| \leq ||w-v||$. 

---

5
5.2. SVP problem

(1) NTRU

The NTRU signature and verification algorithms are both algorithms based on operations on the quotient ring. For each \( q \in \mathbb{Z}, h \in \mathbb{R} \), the set 
\[
M_{q,h} = \{(u, v) \in \mathbb{R}^2 | v \equiv u \otimes h \pmod{q}\}
\]
which is a lattice \( \begin{bmatrix} I_N & H \\ 0 & qI_N \end{bmatrix} \) with a dimension of 2N. This form of lattice is called an NTRU lattice. From this we can construct a lattice attack method against NTRU. The principle is: the CS matrix is reduced by lattice basis to obtain a sufficiently short vector, and then the private keys \( f \) and \( g \) can be recovered, thereby cracking the NTRU.

(2) ETRU

Both the ETRU signature and verification algorithm are based on the calculation algorithm on the quotient ring \( R = \mathbb{Z}_q[X]/[X^N - 1] \). For each \( q \in \mathbb{Z}, h \in \mathbb{R} \), the set 
\[
M_{q,h} = \{(u, v) \in \mathbb{R}^2 | v \equiv u \otimes h \pmod{q}\}
\]
is a lattice \( \begin{bmatrix} I_{2N} & <H> \\ 0 & <qI_{2N}> \end{bmatrix} \) with a dimension of 4N. From this, we can construct a lattice attack method against ETRU. The principle is: the CS matrix is reduced by lattice basis to obtain a sufficiently short vector, and then the private keys \( f \) and \( g \) can be recovered, thereby cracking the ETRU.

5.3. CVP issues

(1) NTRU

The security of NTRU relies on the intractability of the CVP problem. It can be seen from the information that, except for the simple cases of one and two dimensions, there is currently no effective algorithm to solve the CVP problem. The time complexity of solving CVP is \( O(n) \), So NTRU is considered to be the most secure public key cryptosystem at present.

(2) ETRU

Since ETRU uses an EISENSTEIN integer base, which is equivalent to a higher-dimensional and more complex base than an integer base, the CVP problem becomes more difficult. Below we give the two-dimensional optimal solution of CVP

On \( \mathbb{Z}[\omega] \): Higher-dimensional solutions have become more complicated:
6. summary
ETRU is a cost-effective and fast alternative to NTRU that provides the same or better security with a smaller key size and higher speed. Compared with other variants of NTRU, ETRU has the advantage that the ring $\mathbb{Z}[\omega]$ has a greater element density than the ring (such as $\mathbb{Z}[i]$ or $\mathbb{M}_2(\mathbb{Z})$), and its multiplication is more complex in algebra, but each pair of integers is simpler in calculation. A large number of units in $\mathbb{Z}[\omega]$ not only provide resistance to the selected ciphertext attack, but also reduce the probability of decryption failure. Using $\mathbb{Z}[\omega]$ also improves the NTRU lattice attack resistance. The introduction of Eisenstein integer by ETRU itself does not bring any new weaknesses to the NTRU cryptosystem. Therefore, extending the NTRU base to other algebraic integer rings (especially those corresponding to cyclic domains) has great prospects.

References
[1] Caring Di. The design and implementation of NTRU signature algorithm [D]. Xidian University, 2008.
[2] Wu Liqiang, Yang Xiaoyuan, Hao Bin, et al. Parallel NTRU lattice reduction algorithm [J]. Computer Engineering and Applications, 2011, 47(28): 62-64.
[3] Nie Xuyun. Security analysis of NTRUSign digital signature system [D]. Capital Normal University, 2004.
[4] Li Fangwei, Yan Shaojun, Wan Li. A new secure authentication encryption scheme that does not use redundancy and Hash [J]. Computer Engineering and Applications, 2011, 47 (028): 86-88.
[5] Zhang Junan. Research on key technologies of high-speed and high-performance direct digital frequency synthesizer (DDS) [D].