 CMC, vol.56, no.1, pp.19-34, 2018

An Advanced Quantum-Resistant Signature Scheme for Cloud Based on Eisenstein Ring

Faguo Wu[1, 2], Xiao Zhang[1, 2], Wang Yao[1, 2], Zhiming Zheng[1, 2], Lipeng Xiang[3] and Wanpeng Li[4]

Abstract: Signature, widely used in cloud environment, describes the work as readily identifying its creator. The existing signature schemes in the literature mostly rely on the Hardness assumption which can be easily solved by quantum algorithm. In this paper, we proposed an advanced quantum-resistant signature scheme for Cloud based on Eisenstein Ring (ETRUS) which ensures our signature scheme proceed in a lattice with higher density. We proved that ETRUS highly improve the performance of traditional lattice signature schemes. Moreover, the Norm of polynomials decreases significantly in ETRUS which can effectively reduce the amount of polynomials convolution calculation. Furthermore, storage complexity of ETRUS is smaller than classical ones. Finally, according to all convolution of ETRUS enjoy lower degree polynomials, our scheme appropriately accelerate 56.37% speed without reducing its security level.

Keywords: Signature, quantum-resistant, Eisenstein Ring, ETRUS.

1 Introduction

In recent years, there is growing interest in cryptography based on hard lattice problems, classical signature schemes, such as discrete algorithm [ElGamal (1985)], security sensitive applications and encrypted searching, have been proved unsafe based on the quantum computing capacity [Gerjuoy (2005)], it is meaningful to research unbreakable signature schemes under quantum computer's model. Lattice-based signature schemes' construction hold a great promise for post-quantum cryptography, as they enjoy very strong security proofs based on worst-case hardness [Bi and Cheng (2014)]. Besides, lattice signature schemes' calculation mostly relate to the polynomials convolution, so compared with some classical algorithm (like RSA-1024 ECDSA-163), Latticed based signature schemes need a smaller amount of calculations. In this way, lattice-based digital signature algorithm technologies are initially developed for resource-constrained devices

[1] Key Laboratory of Mathematics, Informatics and Behavioral Semantics, Ministry of Education, School of Mathematics and Systems Science, Beihang University, Beijing, 100191, China.

[2] Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, 100191, China.

[3] The Fifth Electronics Research Institute of Ministry of Industry and Information Technology, Guangzhou, 510610, China.

[4] Department of Electrical and Electronic Engineering, City University of London, UK.

* Corresponding Author: Wang Yao. Email: yaowang@buaa.edu.cn.

[Oder, Pöppelmann and Güneysu (2014)], for example, embedded devices and IC card.

In 1997, Goldreich et al. [Goldreich, Goldwasser and Halevi (1997)] proposed the first lattice-based (GGH cryptography system) signature scheme which has no strict security proof. In 2001, Hoffstein et al. [Hoffstein, Pipher and Silverman (2001)] proposed NSS which security based on the closest vector problem (CVP), however, it was broken by [Mironov (2001)]. In 2002, a modified signature scheme R-NSS is proposed based on NSS which was proved unsafe by Stern [Stern (2001)] in the same year. In 2003, Hoffstein et al. [Hoffstein, Howgrave-Graham, Pipher et al. (2003)] proposed NTRUSIGN signature schemes which security are based on the approximate the closest vector problem (APPR-CVP) [Goldreich, Micciancio, Safra et al. (1999)]. Compared with the former signature schemes, NTRUSIGN enjoy higher security, and in recent years, many new signature schemes are being proposed based on NTRU-lattice.

As a family of classical quantum-resist signature schemes, NTRUSIGN are worth being improved. In 2004, Min et al. [Min, Yamamoto and Kim (2004)] make the signing transformation one-to-one correspondent on a given secret key to improve security of NTRUSIGN. In 2005, Hoffstein et al. [Hoffstein, Howgrave-Graham and Pipheretal (2005)] provided a specific parameter generation algorithm to improve their performance. In 2009, Zhang et al. [Zhang and Ji (2009)] improved NTRUSign-based by anonymous multi-proxy signature scheme. In 2013, Stehle et al. [Stehlé and Steinfeld (2011)] improved their security over ideal lattice by extending it is provably category. In 2014, Melchor et al. [Melchor, Boyen, Deneuville et al. (2014)] gave a set of concrete parameters to gauge the efficiency of the signature scheme by sealing the leak on Classical NTRU Signatures. However, due to a large number of polynomials convolution calculation in each part of NTRUSIGN, the speed of them can still be improved.

In this paper, we improve the performance of NTRUSIGN by replacing the integer ring Z with the ring of Eisenstein Z[ω] at the first time. In Section 2, we introduce some necessary properties of Eisenstein integer and ring. In Section 3, we introduce our advanced signature scheme ETRUS, re-choose parameters. In rest sections, we analyze the security, storage complexity, implement performance of ETRUS, and compare it with NTRUSIGN.

2 Preliminaries

In this paper, we proposed an advanced signature scheme based on Eisenstein ring in rest section, so in this section, we discuss some necessary properties of Eisenstein integer and Eisenstein ring to be used as lattice signature base.

Eisenstein integer is an integer of complex, its basis are 1 and ω, ω is the non-real root of $x^3-1=0$, all the element of it can be represented as $a+b\omega(a,b\in\mathbb{Z})$. Eisenstein ring is denoted as $\mathbb{Z}[\omega]$, and some properties of Eisenstein integer and Eisenstein ring are presented as follow.

Let $a+b\omega$ and $c+d\omega\in\mathbb{Z}[\omega]$, it is easy to get some properties as follow.

(1) $\text{Norm}^2(a+b\omega)=(a+b\omega)(a+b\omega^2)=a^2-ab+b^2$.

(2) $\mathbb{Z}[\omega]$ has greater density (has more points) than $\mathbb{Z}$ in same dimension of space.

Proof

$a+b\omega=(a-b/2)+\sqrt{3}b/2$, so the distance from the origin to the point $a+b\omega$ is $\sqrt{a^2-ab+b^2}$, and $\sqrt{a^2-ab+b^2}<a^2+b^2$, the inequality shows are more Eisenstein integer than integer within the same radius of a circle.

It is obviously that Eisenstein integer have greater density than integer in 2-Dimension, and it is easy to calculate that when r=20, Eisenstein integer=36295, integer=31417, Eisenstein space is "tighter" than integer space.

(3) The amount of multiplication and addition between two Eisenstein integers.

$(a+b\omega)*(c+d\omega)=(ac-bd)+(b-a)(c-d)\omega+ac\omega$

Wherein (3) shows that $(a+b\omega)(c+d\omega)$ cost three multiplication and four addition, (4) is very important for reducing the amount of calculation in ETRUS, we will discuss it in Section 6.

(4) Eisenstein ring is an Euclidean domain.

Proof

Eisenstein ring $\Rightarrow \forall a+b\omega, a, b\in\mathbb{Z}, \exists c+d\omega\in\mathbb{Z}[\omega]$

$\Rightarrow \text{Norm}(c+d\omega-a-b\omega)<1$

$\Rightarrow c'=\lfloor a \rfloor, d'=\lfloor d \rfloor$ it certainly established

According to (4), we can easily have following property.

(5) For any $a+b\omega$ and $c+d\omega\in\mathbb{Z}$, there exist t, $r\in\mathbb{Z}$ such that $a+b\omega=t(c+d\omega)+r$ where either r=0 or Norm (r) <Norm(c+ dω).

(6) 2N dimensional vectors in $\mathbb{Z}[\omega]$ can form a lattice.

Proof

According to the following signature scheme's construction, 2N dimensional vector $V_i=(a_{i_1},b_{i_1},a_{i_2},b_{i_2},...,a_{i_N},b_{i_N})$ in $\mathbb{Z}[\omega]$ is consist of N Eisenstein integers as

$$a_{i_1}+b_{i_1}\omega, a_{i_2}+b_{i_2}\omega,...,a_{i_N}+b_{i_N}\omega \quad (1)$$

In order to form a 2N-dimensional lattice by these vectors, we choose 2N linearly independent vectors as

$V_1,V_2,...,V_i...,V_{2N}$, $V_i=(0,0,...0,1,0,...,0)$ (*i*-th is 1, else are 0).

We let $V_i=(a_{i_1}=0,b_{i_1}=0,a_{i_2}=0,b_{i_2=0},...,a_{i_i}=1,b_{i_i+1=0},...,a_{i_N},b_{i_N})$. Therefore lattice L in $\mathbb{Z}[\omega]$ with 2N dimension can be expressed as

$L=x_1V_1+...+x_iV_i+...+x_{2N}V_{2N}$. x_i is integer.

Indeed, $\mathbb{Z}[\omega]$ is isomorphism to $\mathbb{Z}[x]$, it can easily form a 2N-dimensional lattice.

3 The proposed signature scheme on Eisenstein ring

In this section, we introduce our advanced quantum-resistant signature scheme for Cloud Based on Eisenstein Ring, we named it for ETRUS. Compared with NTRUSIGN, we choose suitable parameters for our signature scheme.

The steps to construct ETRUS are as follow.

3.1 Public parameters selection

(1) Select suitable integer N, $q = \theta + \eta\omega \in \mathbb{Z}[\omega]$, **NormBound**.

In ETRUS, we let $N \approx N'/2$, $\| q \| \approx q'/2$, we store all polynomials as 2N dimensional vector by following construction, it means all the calculation in ETRUS is in \$2N\$ dimensional lattice, when reference with Kouzmenko [Kouzmenko (2006)], we let N'=2N. By abstract algebra, it is not difficult to obtain that $\mathbb{Z}[\omega]/q$ (ETRUS) has $Norm^2(q)$ elements, $\mathbb{Z}/q'$ has q' elements, so each coefficient of polynomial in ETRUS mod q' has $Norm^2(q)$ kinds of choice, it has q' kinds of choice in NTRUSIGN. In order to resist lattice reduction attack [Joux and Stern (1998)] and exhausting attack, we let $Norm^2(q) > q'$. Besides ensure security on the choice of parameters, we let $\| q \| \approx q'/2$ to simplify the calculation in the remaining sections, and we let **NormBound**(< **NormBound**′) (compute in verification step).

3.2 Public key generation

(1) Choose two polynomial f and g which is from ring $\mathbb{Z}[\omega][X]/X^N - 1$. Compute $h = f^{-1} * g \, mod(q)$, (f^{-1} is calculated like in Kouzmenko [Kouzmenko (2006)]).

(2) Compute two small polynomials (F, G) satisfying $f * G - g * F = q$. Due to special structure of $\mathbb{Z}[\omega]$, we do not need to use Extended Euclidean algorithm [Brent (1976)] in this step which is explained as follow.

In ETRUS, we let $\| f \| = \| g \| \approx C\sqrt{N}$, $\| F \| = \| G \| \approx CN/\sqrt{6}$ (compute in verification step). In Step 1, we let $f = f_0 + f_1 * x + ... + f_{N-1} X^{N-1}$ where $f_i x^i = (a_i + b_i\omega)x^i$, and store f and g coefficients as 2N dimensional vector, according to above proof, these vectors are points in 2N dimensional lattice.

In ETRUS, once we get $R_f = x_1 + x_2\omega$ and $R_g = y_1 + y_2\omega$, we let two Eisenstein integer $m = \alpha_1 + \alpha_2\omega$, $n = \gamma_1 + \gamma_2\omega$. So when (α_1, γ_2) or (α_2, γ_1) are determined in advance, in order to get F, G which satisfy $mR_f - nR_g = q$, we only need to solve two variable linear equation to get two Eisenstein integer (m, n), so we need not to use the Extended Euclidean Algorithm in our advance signature scheme to get F, G. However, in

NTRUSIGN, in order to get (F', G'), we need to use the Extended Euclidean Algorithm to get two important integers (m', n') [Hoffstein, Howgrave-Graham, Pipher et al. (2003)]. (We will have a detailed calculation in Speed comparison) in Section 5 of Public Key Generation.

3.3 Signing

(1) Hash the document to get two 2N dimensional vector (m_1, m_2), Denote $m_{1_{2i}} = c_i$, and $m_{1_{2i+1}} = d\omega$, so $m_{1_{2i}} + m_{1_{2i+1}} \in \mathbb{Z}[\omega]$. Then m can be presented as follow

$$m = (m_0 + m_1) + ... + (m_{1_{2i}} + m_{1_{2i+1}})x^i + ... + (m_{N-2} + m_{N-1}\omega)X^{N-1}.$$

(2) Compute B, b

$$G * m_1 - F * m_2 = A + q * B,\ -g * m_1 + f * m_2 = a + q * b$$

A and a have coefficients $e + d\omega$ in $\mathbb{Z}[\omega]$, $e \in (-1/2, 1/2) * \theta, d \in (-1/2, 1/2) * \eta$.

(3) Signature is $S = f * B + F * b (\mathrm{mod}\, q)$

We store document (after hash) as Eisenstein integer to accelerate signing and verification speed in Section 6.

3.4 Verification

(1) Compute $T = S * h \ (\mathrm{mod}\, q)$.

(2) Verify if $\| S - m_1 \|^2 + \| T - m_2 \|^2 \le \mathbf{NormBound}^2$, if so, accept this signature, otherwise, reject it.

Verification is $\| S - m_1 \|^2 + \| T - m_2 \|^2 \le \mathbf{NormBound}^2$. According to (1) and (2)

$$(S\ T) = (B\ b)\begin{pmatrix} f & g \\ F & G \end{pmatrix} = \left\lfloor (m_1, m_2) \begin{pmatrix} G/q & -g/q \\ -F/q & f/q \end{pmatrix} \right\rfloor \begin{pmatrix} f & g \\ F & G \end{pmatrix} \quad (2)$$

Combine with (1), so we can easily obtain following expressions

$$(m_1, m_2) - (S, T) = (A/q \quad a/q)\begin{pmatrix} f & g \\ F & G \end{pmatrix}$$

From (5), as our construction, A and a have coefficients $e + f\omega$ in $\mathbb{Z}[\omega]$, and coefficients of A/q and a/q between $(-1/2, 1/2)$. We can easily have following expression

$$m_1 - S = \varepsilon f + \beta F, m_2 - T = \varepsilon f + \beta F \quad (3)$$

Regard ε as uniformly distributed variable, and then we let $\varepsilon_0, \varepsilon_1, ..., \varepsilon_{N-1}$, let $\varepsilon_i = a_i + b_i\omega$. a_i, b_i belong to (-1/2, 1/2). According to the above mentioned description,

We can compute Norm$^2(\varepsilon_i)$= $a_i^2 + b_i^2 - a_i b_i$, in order to compute Norm$^2(\varepsilon)$, we should compute an estimated value of Norm$^2(\varepsilon_i)$, which requires Norm$^2(\varepsilon_i)$ expectations.

We regard a_i and b_i as independent random variable uniformly distributions in the interval (-1/2, 1/2). Therefore we have following expressions

$$\begin{cases} p(a_i, b_i) = 1, a_i, b_i \in (-1/2, 1/2) \\ p(a_i, b_i) = 0, else \end{cases} \tag{4}$$

Therefore Norm$^2(\varepsilon)$ is calculated as follow

$$\int_{-1/2}^{+1/2}\int_{-1/2}^{+1/2}\ldots\int_{-1/2}^{+1/2} (Norm^2(\varepsilon_0) + \ldots Norm^2(\varepsilon_{N-1}))d\varepsilon_0 \ldots d\varepsilon_{N-1}$$
$$= N\int_{-1/2}^{+1/2}\int_{-1/2}^{+1/2} Norm^2(\varepsilon_i)d\varepsilon_i = N/6$$

So $\| \varepsilon \|^2 \approx N/6$. In the same way, we can obtain $\| \beta \|^2 \approx N/6$.

We now can estimate norm of ($S - m_1, T - m_2$).

$$\| S - m_1 \|^2 + \| T - m_2 \|^2 = \| (\varepsilon f + \beta F, \varepsilon f + \beta F) \|$$

In ETRUS, according to the above mentioned calculation, we would better let $\| f \| = \| g \| \approx C\sqrt{N}$, and (F, G) satisfy $\| F \| = \| G \| \approx CN/\sqrt{6}$. While in NTRUSIGN, $\| F' \| = \| G' \| \approx CN'/\sqrt{12} = CN/\sqrt{3}$, $\| f \| = \| g \| \approx C\sqrt{N'} = C\sqrt{2N}$.

In ETRUS, through above calculation. $\| (\varepsilon f + \beta F, \varepsilon f + \beta F) \| \approx \frac{N^3}{18}(1 + \frac{6}{N})$. However, in NTRUSIGN. $\| S - m_1 \|^2 + \| T - m_2 \|^2 \approx \frac{N'^3}{72}(1 + \frac{12}{N'}) \approx \frac{N^3}{8}(1 + \frac{6}{N})$. So in ETRUS, signer should choose one suitable Appr-CVP **NormBound**^{2} $\geq$ $\frac{N^3}{18}(1 + \frac{6}{N})$, the verifier calculate $\| S - m_1 \|^2 + \| T - m_2 \|^2$, if the result is smaller than **NormBound**^{2}, then the verification is succeed, otherwise failed.

We can also use the new perturbation [Hu, Wang and He (2008)] in 2008 to avoid the flaw [Nguyen and Regev (2006)] found in 2006.

According to the above construction, we proposed an advanced signature scheme ETRUS by replacing the ring Z in NTRUSIGN with the ring Z[ω]. Compared to NTRUSIGN, we can realize a simpler process for ETRUS by suitable parameters.

4 Security analysis of ETRUS

$\mathbb{Z}[\omega]$ is isomorphism to $\mathbb{Z}[x]$. Our ETRUS signature scheme is secure under four typical attacks, named Lattice Reduction Attack, Exhaustive Search Attack, GCD Lattice

Attack [Gentry and Szydlo (2002)], and Averaging Attack [Hoffstein, Kaliski Jr, Lieman et al. (2000)].

4.1 Lattice reduction attack

Lattice reduction attack is to trying to find a very short non-zero vector in L_h, since (f, g) and rotations are probably the shortest such vectors. According to the above description in Section 3, lattice dimension is N'=2N, according to Gaussian heuristic [Gama, Nguyen and Regev (2010)], a general convolution modular lattice L_h has dimension \$2N\$ and determinant q^N, it is probable shortest vector and closest vectors have approximate size.

$\lambda_{Gauss(L_h)} = \sqrt{Nq/\pi e}$ In ETRUS, we take (f, g) as the probably shortest vectors, they have shortest vector approximately as $\sqrt{Nr} = \sqrt{2N'r}$, according to Hoffstein et al. [Hoffstein, Howgrave-Graham, Pipher et al. (2010)], the ratio of $\lambda_{Gauss(L_h)} / \sqrt{Nr}$ is proved small enough to resist Lattice reduction attack to find probably the shortest vector (f, g).

Forger can also use lattice reduction to directly locate signature (S, T), in signature scheme, $\| S - m_1 \|^2 + \| T - m_2 \|^2 \le \mathbf{NormBound}^2$, it indicates (S, T) is close to (m_1, m_2). From the Gaussian heuristic, we can find that potential forger select a random point in 2N dimensional lattice which distance to (m_1, m_2) must no more than $\mathbf{NormBound}$ / $\lambda_{Gauss(L_h)}$ times the expected distance to the actual closest point in lattice. In ETRUS, when we choose appropriate parameters satisfy $\mathbf{NormBound} / \sqrt{Nr} > \mathbf{NormBound}' / \lambda_{Gauss(L_h)}$.

In particular, we can choose $N = N'/2 = 251/2 \approx 126$, when r=2/3, the Gaussian heuristic of ETRUS is $\sqrt{2N'r}$ approximately to 123. Hence setting $\mathbf{NormBound}$ =300 means that forger needs to find a point is no more than 2.43 times the expected the shortest distance, when we choose $\mathbf{NormBound}$ =250, this ratio goes down to 2.03. When we choose satisfy small $\mathbf{NormBound}$ in ETRUS close to 1. This appropriate closest vector problem (App-CVP) proved to be NP-hard [Dinur (2002)].

Therefore, in ETRUS, it is more difficult to get (f, g) than NTRUSIGN due to preliminaries. When we choose suitable $\mathbf{NormBound}$ which discussed in verification. ETRUS can avoid this type adversary. So ETRUS can effectively resist Lattice reduction attack.

4.2 Exhaustive search attack

Exhaustive search attack is trying to find the other half $(m_1 - S, m_2 - T)$. In Section 3,

we have discussed that $\| q \| \approx \frac{q'}{2}$, in this situation, forger choose an integer point in L_h is $\| q \|^{-N}$ probability, so attack can easily create a lattice point (s, t). Therefore, half of $(m_1 - S, m_2 - T)$ is being leaked, but the remaining work is too hard to finish, attack should find other half of vector y from $\lfloor \approx -\| q \|, \approx \| q \| \rfloor$. Furthermore, they should satisfy verification inequality. We can calculate its probability as follow

$$P(\| Y \|^2 < \mathbf{NormBound}^2) < \frac{\pi^{N/2}}{\Gamma(1+N/2)} (\frac{\mathbf{NormBound}}{\| q \|})^N$$

In particular, compared with classical NTRUSIGN, we choose $\| q \| \approx \frac{q'}{2}$ =67, $\mathbf{NormBound}$ = 300. $N = N'/2 \approx 126$.

Therefore, we have $P(\| Y \|^2 < \mathbf{NormBound}^2) \approx 2^{-121.44}$. When we choose (N, q), $P(\| Y \|^2)$ which is small enough to prevent exhaustive search attack.

4.3 GCD lattice attack

GCD lattice attack is an effective way to break lattice signature scheme, like NSS. In ETRUS, attacker want use GCD lattice attack to get some $f * x_i$ without mod q in $\mathbb{R}$, and $f * x_i$ probably generate the closest vector in lattice. However, due to ETRUS signature scheme $S = f * B + F * b \pmod{q}$, it is difficult for attacker to get independent $f * x_i$. Furthermore, even attacker can get $(x_i), (x_j)$, $(| x_i |, | x_j |) = 1$, he cannot break ETRUS down, as in ETRUS, $| x_i |, | x_j | \in R[\omega]$, $(| x_i |, | x_j |) = 1$ cannot get $a * x_i + b * x_j = 1$, so attacker finally can't get $\mathrm{GCD}(f * x_i, f * x_j) = (f)$.

Due to special structure of $R[\omega]$ and Eisenstein integer. ETRUS can effectively resist GCD lattice attack

4.4 Averaging attack

Averaging attack is trying to get $f * \overline{f}$ through thousands of signatures, in ETRUS, adversary uses following average equation to get $f * \overline{f}$

$$A = \lim_{r \to \infty} (1/r) \sum_{i=1}^{r} (f * B + F * b) * (\overline{f * B + F * b})$$

when r tends to ∞, $B * \overline{B}$ and $b * \overline{b}$ tends to constant, so attacker can only get $f * \overline{f} + g * \overline{g}$ through Eq. (14), he doesn't have effective way to get one of the $f * \overline{f}$ and $g * \overline{g}$. Therefore, ETRUS is safe under averaging attack. Through the above mentioned

analysis, we know ETRUS resist averaging attack if and only if polynomials B and b are non-zero.

According to the above mentioned analysis, ETRUS can effectively resist four typical attack with suitable parameters.

5 Storage complexity analysis

In this section, we analyze the storage complexity of ETRUS and NTRUSIGN under the same security level. In order to achieve this goal, we have presented the Public key size, Document size and Signature size of ETRUS and NTRUSIGN, and choose the parameters as discussed in previous sections $2N = N', \| q \| \approx q^{'} / 2$.

In the actual process of signature, computer store Eisenstein integer $a + b\omega$ as a pair of integer (a, b), so in ETRUS, we store every Eisenstein integer $a + b\omega (mod\ q)$ size as $2\lceil log_2(4 \| q \| /3) \rceil$ bits from Jarvis et al. [Jarvis and Nevins (2015)], in NTRUSIGN, we store every integer c $(mod\ q')$ size as $\lceil log_2(q') \rceil$ bits.

(1) Public Key Size

Public key is $h = f^{-1} * g (\text{mod q})$.

In ETRUS, according to the above mentioned discussion, it is easy to calculate the size as $Size_{h_E} = 2N \lceil log_2(4 \| q \| /3) \rceil = 2N \lceil log_2(2q' / 3) \rceil$ bits.

In NTRUSIGN, $Size_{h'_N} = N \lceil log_2(q') \rceil = 2N \lceil log_2(q') \rceil$ bits.

$Size_{h_E} < Size_{h'_N}$. Therefore, ETRUS have smaller public key size than NTRUSIGN.

(2) Document Size

In this comparison, as we described in Section 3, document size is the size stored in computer after Hash.

In ETRUS, we have a transform of document $H_m = (m_1, m_2)$, so document size is $Size_{document_E} = Size_{m_1} + Size_{m_2} = 4N \lceil log_2(2q' / 3) \rceil$bits.

In NTRUSIGN, $Size_{document_N} = 4N \lceil log_2(q') \rceil$bits.

$Size_{document_E} < Size_{document_N}$, therefore, ETRUS have smaller document size than NTRUSIGN.

(3) Signature Size

The signature is $S = f * B + F * b \ (\text{mod q})$.

In ETRUS, we can easily obtain $Size(siganature_E) = 4N \lceil log_2(2q' / 3) \rceil$bits.

In NTRUSIGN, $Size(siganature_N) = 4N \lceil log_2(q') \rceil$ bits.

$Size(siganature_E) < Size(siganature_N)$. Therefore, ETRUS have smaller signature size than NTRUSIGN.

When we combine lattice dimension with document size, it is surprising means that when lattice dimension have a linear extension, the number of signature points in lattice also increase at linear level.

In particular, when we compared to classical NTRUSIGN with $N' = 251, q' = 128$, and we choose ETRUS almost at same security level with $N = 127$, $q = 67 + 0$, $\omega = 67$ (in order to simplify calculation Process, we let q=67), then we have appropriate comparison Tab. 1 as follow.

Table 1: Size of Classical NTRUSIGN and ETRUS

Signature Scheme	NTRUSIGN	ETRUS
Public Key Size(bits)	1764	1615
Document Size(bits)	3528	3230
Signature Size(bits)	3528	3230

According to the above Tab. 1. Document and Signature size almost double times than Public Key size, and it is bigger than current signature schemes (RSA, DSA) to resist quantum computer's attack.

Through above analysis, ETRUS need smaller computer storage space than NTRUSIGN, and size of each part reduce $\lceil log_2(3/2) \rceil / \lceil log_2(2q'/3) \rceil$%.

6 Performance analysis

In this section, we presented the performance analysis of ETRUS and NTRUSIGN. Without affecting the safety of the two signature schemes, we compare ETRUS, for parameters (N , q , **NormBound**) with NTRUSIGN, for parameters(N '=2 N , q ' , **NormBound** ').

There are many different ways to get the complexity of implement performance of NTRUSIGN and ETRUS. Obviously, it closely relies on the hardware platform and the implementation details, so if we only implement this algorithm on a computer, our results do not have the universality and persuasiveness, hence the main purpose of this section is to give a universality and persuasiveness implement performance comparison between NTRUSIGN and ETRUS.

We split the entire implement performance into three part: Key Generation, Signing and Verification, convert the implement performance comparison to speed comparison of Key Generation，Signing, Verification. We simplify the algorithmic process into elementary operations like addition, subtraction, multiplication, or division integers.

The more advanced the CPU use internal microinstruction fast multiplication algorithm, for example, in reg32, addition(A) consume 1 to 3 clock cycles, multiplying(M)

consumption 13 to 26 clock cycles, and according to Jarvis et al. [Jarvis and Nevins (2015)], module(D) in $\mathbb{Z}[\omega]$ consumed almost the 27 times than multiplying, in $\mathbb{Z}$, module(D)=multiplying(M). In order to obtain a uniform result, we unify all the operation time as approximately multiplying time, so M=5A, M=D' in NTRUSIGN and 27M=D in ETRUS, and computation in the array to store large Number is also ignored.

(1) Key Generation Speed

Firstly, Key Generation need signer to compute public key $h = f^{-1} * g(\text{mod q})$. In ETRUS scheme, the convolution of two polynomial with degree $N-1$ cost $3N^2 * M$ multiplication, and each coefficient of polynomial h cost $4(N-1) * A$ addition, so totally cost $(4N^2 - 4N) * A$ addition, and $N * D$ modular. In NTRUSIGN scheme, the convolution of two polynomial of degree $N'-1$ cost $N'^2 * M$ =4 $N^2 * M$ multiplication, $N'(N'-1) * A = (4N^2 - 2N) * A$ addition, and $N' * D'$ modular.

Secondly, in ETRUS, signer should calculate two small polynomials as previously mentioned $(F, G) \in \mathbb{Z}[\omega][X] / X^N - 1$ satisfying $f * G - g * F = q$, the process of its implementation in the need for hundreds of large numbers of operation, because of this, secret key generation rate is greatly reduced. In order to find suitable (F, G), we should find F_1 and $G_1 \in \mathbb{Z}[\omega][X] / X^N - 1$ satisfy the following equation

$$q = \begin{pmatrix} f & g \\ F_1 & G_1 \end{pmatrix} \tag{5}$$

In order to find F_1 and G_1, we should find two polynomial u and v satisfy

$$\begin{cases} f * v + k_1 * (x^N - 1) = R_f \\ g * u + k_2 * (x^N - 1) = R_g \end{cases} \tag{6}$$

Where R_f and R_g are the (integer) resultants of ($f, x^N - 1$) and ($g, x^N - 1$), and we know that ($f, x^N - 1$) equal to $\prod_{i=1}^{N-1} f(x^i) \mod (1 + x + x^2 + ... + x^{N-1})$. In order to get R_f and R_g. In ETRUS, we need $2n * N * (N-1)$ times convolution (where n is non-zero coefficient number of f) to compute R_f and R_g , so it costs $6n * N * (N-1) * M$ multiplication, and $8n * N * (N-1) * A$ addition, same in NTRUSIGN, it cost $4n'N(2N-1) * M$ multiplication (where n' is non-zero coefficient number of f') and $2n' * N * (N-1) * A$ addition. We use R_f and R_g to solve Eqs. (4) and (5). In order to get polynomial u and v. We need to solve the following linear equation

$$\begin{pmatrix} f_0 & f_{N-1} & \cdots & f_1 \\ f_1 & f_0 & \cdots & f_2 \\ \vdots & \vdots & & \vdots \\ f_{N-1} & f_{N-2} & \cdots & f_0 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{pmatrix} = \begin{pmatrix} R_f \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{7}$$

Where $f_0, f_1, ..., f_{N-1}$ are coefficients of f and $u_0, u_1, ..., u_{N-1}$ are coefficient of v.

In ETRUS and NTRUSIGN, we solve this equation with Gauss-Jordan algorithm [Dekker and Hoffmann (1989)] which need a small amount of multiplication. In ETRUS, we let $(a+b\omega)/(c+d\omega) = (a+b\omega)(\overline{c+d\omega})/R$, then we can know that it costs $6n^3 * M$ multiplication and $(12n^3 - 8n^2 + 4n) * A$ addition to get u and v. In NTRUSIGN, it costs $6n'^3 * M$ multiplication, and $2*(n'*((n'-1)^2)*A =$ addition to get u', v'.

Then in NTRUSIGN, we should use Extended Euclidean algorithm to get m', n' satisfy $m'R_{f'} - n'R_{g'} = q'$, and according to Stark [Stark (2005)], algorithm complexity of Extended Euclidean algorithm is $O(log_2(R_{f'}) * log_2(R_{g'}))$. According to Section 3 verification step, in ETRUS, the time of this step can be ignored.

In order to have a more intuitive expression, we let n=N/4, n'=N'/4=N/2, then we have appropriate Key Generation Speed of NTRUSIGN and ETRUS as following Tab. 2 (Unify all operations as multiplication in verification step).

Table 2: Key Generation Speed of NTRUSIGN and ETRUS

Signature Scheme	Key Generation Speed
NTRUSIGN	$(800N^3 264N^2 288N)/160 + C(log_2(R_{f'}) * log_2(R_{g'}))$
ETRUS	3528

C is a constant in NTRUSIGN.

According to the Tab. 2, we can easily find that ETRUS costs much less time than NTRUSIGN in Public Key Generation.

$$Accelerate\,rate_{Key}\% \approx \lim_{N\to\infty} \frac{M_{NTRUSIGN} - M_{ETRUS}}{M_{NTRUSIGN}} \approx 56.37\%$$

Compared with the NTRUSIGN, Key Generation Speed approximately accelerate 56.37\% in ETRUS when N trends to ∞, according to algorithm of ETRUS, Key Generation needs much less polynomial convolution at each step than NTRUSIGN, and due to special properties of Eisenstein integer, it also eliminate a number of time-consuming steps (like Extended Euclidean algorithm), so ETRUS's speed has been improved a lot.

(2) Signing and Verification Speed

According the same analysis method as the above Key Generation Speed, we can easily have the comparison of Signing and Verification in following Tab. 3.

Table 3: Signing Speed of NTRUSIGN and ETRUS

Signature Scheme	Signing Speed	Verification Speed
NTRUSIGN	$(144N^2+28N)/5$	$(24N^2+52N)/5$
ETRUS	$(114N^2+159N)/5$	$(19N^2+165N)/5$

Compared with NTRUSIGN, Signing and Verification Speed approximately accelerate 20.83\% and 22.73\% when N trends to ∞, respectively.

(3) Total Comparison

According to the above analysis and calculation, it is not difficult to have a total speed comparison between NTRUSIGN and ETRUS by combining Key Generation speed, signing speed, and Verification speed.

Table 4: Speed Comparision of NTRUSIGN and ETRUS

Scheme	Speed Comparison
NTRUSIGN	$(800N^3+5112N^2+2048N)/160+C(log_2(R_{f'})*log_2(R_{g'}))$
ETRUS	$(349N^3+264N^2+164N)/160$

$$Accelerate\,rate_{Total}\% \approx \lim_{N\to\infty}\frac{M_{NTRUSIGN}-M_{ETRUS}}{M_{NTRUSIGN}} \approx 56.37\%$$

It is not surprising that whole signature scheme and Public Key Generation speed accelerate almost the same percentage at 56.37\% when N trends to ∞, because in ETRUS and NTRUSIGN, 99.51\% of the calculation is occupied by Public Key Generation when N'=251, and this ratio will increase when N becomes bigger.

When we implemented the ETRUS $(N=127, q=67)$, NTRUSIGN $(N=251, q=128)$ (appropriate parameters) in practice for average time, we have following Tab. 5.

Table 5: Comparision with concrete parameters

Signature Scheme	NTRUSIGN-251	ETRUS-127
Key Generation Speed (ms)	183.7	113.46
Signing Speed	1.41	1.24
Verification Speed	1.32	1.146

Tab. 5 shows that Key Generation Speed, Signing Speed, and Verification Speed accelerate significantly in practice. We can easily calculate that Key Generation Speed appropriately accelerate 38.18\%, Signing Speed appropriately accelerate 12.06\%, Verification Speed appropriately accelerate 12.42\%, growth rate of Key Generation, Signing, and Verification speed consistent with the theoretical result. However, due to $N \ll \infty$ in practice, accelerate rate is smaller than the theoretical value.

6 Conclusion

With the surprising development of quantum computer, lattice-based signature schemes, which are constructed to resist quantum attack, become more and more attractive. In this paper, we introduce an advanced signature scheme, namely ETRUS. By discussing the essential properties of $\mathbb{Z}[\omega]$ to be used as signature base, selecting appropriate parameters and complex polynomials convolution, we have reduced. Norm of (f, g) from $C\sqrt{2N}$ to $C\sqrt{N}$, Norm of (F, G) from $CN/\sqrt{3}$ to $CN/\sqrt{6}$. Furthermore, we have proved that ETRUS is secure under four typical attacks: Lattice Reduction attack, Exhausting attack, GCD attack, and averaging attack. When compared with NTRUSIGN at same security level, ETRUS has smaller storage complexity, whole size reduces $\lceil 10N * log_2(3/2) \rceil$. Besides, by theoretical analysis and performance comparison, compared with NTRUSIGN, ETRUS has 56.37\% speed improvement. (Public key Generation 56.37%, signing and verification 20.83%). Therefore, the proposed scheme on Eisenstein lattice is proved to be a secure signature scheme based on NTRU-lattice, with less storage complexity and higher speed than classical lattice-based signature scheme.

Acknowledgement: The authors wish to express their appreciation to the reviewers for their helpful suggestions which greatly improved the presentation of this paper. This work was supported by the Major Program of National Natural Science Foundation of China (11290141).

References

Bi, J.; Cheng, Q. (2014): Lower bounds of shortest vector lengths in random NTRU lattices. *Theoretical Computer Science*, vol. 560, pp. 121-130.

Brent, R. P. (1976): *Analysis of the Binary Euclidean Algorithm*, pp. 6-7. Oxford University.

Dekker, T. J.; Hoffmann, W. (1989): Rehabilitation of the gauss-jordan algorithm. *Numerische Mathematik*, vol. 54, no. 5, pp. 591-599.

Dinur, I. (2002): Approximating SVP to within almost polynomial factors is np-hard. *Theoretical Computer Science*, vol. 285, no. 1, pp. 55-71.

ElGamal, T. (1985): A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469-472.

Gama, N.; Nguyen, P. Q.; Regev, O. (2010): Lattice enumeration using extreme pruning. *Advances in Cryptology EUROCRYPT 2010*, pp. 257-278.

Gentry, C.; Szydlo, M. (2002): Cryptanalysis of the revised NTRU signature scheme. *Advances in Cryptology EUROCRYPT 2002*, pp. 299-320.

Gerjuoy, E. (2005): Shor's factoring algorithm and modern cryptography illustration of the capabilities inherent in quantum computers. *American Journal of Physics*, vol. 73, no. 6, pp. 521-540.

Goldreich, O.; Goldwasser, S.; Halevi, S. (1997): Public-key cryptosystems from lattice reduction problems. *Advances in Cryptology CRYPTO*, pp. 112-131.

Goldreich, O.; Micciancio, D.; Safra, S.; Seifert, J. P. (1999): Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, vol. 71, no. 2, pp. 55-61.

Hoffstein, J.; Howgrave-Graham, N.; Pipher, J.; Silverman, H.; Whyte, W. (2003): Ntrusign: Digital signatures using the ntru lattice. *Topics in Cryptology CT-RSA 2003*, pp. 122-140.

Hoffstein, J.; Howgrave-Graham, N.; Pipher, J.; Silverman, H.; Whyte, W. (2005): Performance improvements and a baseline parameter generation algorithm for ntrusign. *IACR Cryptology EPrint Archive*, vol. 2005, pp. 274.

Hoffstein, J.; Howgrave-Graham, N.; Pipher, J.; Whyte, W. (2010): Practical lattice-based cryptography: Ntruencrypt and ntrusign. *LLL Algorithm*, pp. 349-390.

Hoffstein, J.; Kaliski Jr, B. S.; Lieman, D. B.; Robshaw, M.; Yin, Y. (2000): Secure user identification based on constrained polynomials. *US Patent*, pp. 76-63.

Hoffstein, J.; Pipher, J.; Silverman, J. H. (2001): Nss: An NTRU lattice-based signature scheme. *Advances in Cryptology Eurocrypt 2001*, pp. 211-228.

Hu, Y.; Wang, B.; He, W. (2008): Ntrusign with a new perturbation. *IEEE Transactions on Information Theory*, vol. 54, no. 7, pp. 3216-3221.

Jarvis, K.; Nevins, M. (2015): ETRU: NTRU over the eisenstein integers. *Designs, Codes and Cryptography*, vol. 74, no. 1, pp. 219-242.

Joux, A.; Stern, J. (1998): Lattice reduction: A toolbox for the cryptanalyst. *Journal of Cryptology*, vol. 11, no. 3, pp. 161-185.

Kouzmenko, R. (2006): Generalizations of the NTRU cryptosystem. *Diploma Project, École Polytechnique Fédéralede Lausanne*, vol. 9, pp. 15-32.

Melchor, C. A.; Boyen, X.; Deneuville, J. C.; Gaborit, P. (2014): Sealing the leak on classical NTRU signatures. *Post-Quantum Cryptography*, pp. 1-21.

Min, S.; Yamamoto, G.; Kim, K. (2004): On the security of ntrusign signature scheme. *2004 Symposium on Cryptography and Information Security*, vol. 4, no. 1, pp. 625-630.

Mironov, I. (2001): A note on cryptanalysis of the preliminary version of the NTRU signature scheme. *IACR Cryptology EPrint Archive*, vol. 2001, pp. 5.

Nguyen, P. Q.; Regev, O. (2006): Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. *Advances in Cryptology EUROCRYPT 2006*, pp. 271-288.

Oder, T.; Pöppelmann, T.; Güneysu, T. (2014): Beyond ECDSA and RSA: Lattice-based digital signatures on constrained devices. *Proceedings of the 51st Annual Design Automation Conference*, pp. 1-6.

Stark, H. M. (2005): An introduction to number theory. *MIT Preview*, vol. 418, no. 3, pp. 537-540.

Stehlé, D.; Steinfeld, R. (2011): Making NTRU as secure as worst-case problems over ideal lattices. *Advances in Cryptology EUROCRYPT 2011*, pp. 27-47.

Stern, J. (2001): Cryptanalysis of the NTRU signature scheme (NSS) from Eurocrypt 2001. *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 1-20.

Zhang, J.; Ji, C. (2009): An id-based and repairing ntrusign-based anoapproximatingy-mous multi-proxy signature scheme. *Computational Intelligence and Software Engineering*, pp. 1-4.