Efficient Certificateless Signature Scheme on NTRU Lattice

Jia Xie[1], Yupu Hu[1], Juntao Gao[1], Wen Gao[1], Mingming Jiang[2]
[1]School of Telecommunications Engineering, Xidian University
Xi'an, Shaanxi 710071-China
[e-mail]: xiejia199325@163.com
[2]School of Computer Science and Technology, Huaibei Normal University
Huaibei, Anhui 235000 - China
[e-mail]: jiangmm3806586@126.com
*Corresponding author: Jia Xie

Received October 22, 2015; revised May 4, 2016; revised June 22, 2016; accepted August 22, 2016; published October 31, 2016

Abstract

Because of the advantages of certificateless and no escrow feature over the regular signature and identity-based signature, certificateless signature has been widely applied in e-business, e-government and software security since it was proposed in 2003. Although a number of certificateless signature schemes have been proposed, there is only one lattice-based certificateless signature scheme which is still secure in the quantum era. But its efficiency is not very satisfactory. In this paper, the first certificateless signature scheme on NTRU lattice is proposed, which is proven to be secure in random oracle model. Moreover, the efficiency of the new scheme is higher than that of the only one lattice-based certificateless signature.

Keywords: Certificateless, Signature, NTRU, Secure, Lattice

This work was supported by the National Natural Science Foundation of China (No.61303217, 61472309, 61402353, 61502372 and 61572390), the 111 project (No. B08038), the Fundamental Research Funds for the Central Universities (No.JB140115), the Natural Science Foundation of Shaanxi province (No.2013JQ8002, 2014JQ8313), Natural Science Foundation of Anhui Higher Education Institutions(No.2016A627).

http://dx.doi.org/10.3837/tiis.2016.10.030 **ISSN : 1976-7277**

1. Introduction

Since it was proposed in 1976, the digital signature scheme has more than 40 years' history. With the further research of digital signature and the rapid development of e-commerce and e-government, the conventional digital signature can no longer meet the needs in practice. So more and more researchers pay increasing attention to the digital signatures with additional properties, such as, blind signature, identity-based signature, certificateless signature, group signature, proxy signature and so on. In blind signature scheme, the signer can complete the signature on message *m* without learning anything about *m*. So it is widely used in e-cash and e-voting. Group signature makes it possible to realize the anonymity and traceability at the same time, and it is widely applied in anonymous certificates, e-voting, e-cash, and anonymous attestation. Ring signature is an alternative to group signature, but the anonymity revocation in it is impossible. Proxy signature can realize the delegating signing capabilities in authenticated routing. As can be seen in [1, 2], all the non-conventional digital signatures can contribute to challenges in the information and communication technology.

With the advent of lightweight cryptography, the main difficulty today in digital signature is the lightweight authentication, which can be realized by decreasing the cost of infrastructures to authenticate the public/private keys. In the traditional Public Key Infrastructure (PKI), a trusted certificate authority (CA) composes certificates to ensure the authenticity of the users. It brings a vexing problem—certificate management problem. In order to deal with it, identity-based public key cryptography (IB-PKC) was first proposed in [3]. In the IB-PKC, the public key is just the product of the user's identity while the private key is generated by the trusted private key generator (PKG) and the user. It is obvious that the IB-PKC has the advantage of certificateless. However, it suffers from the key escrow problem. More specifically, the PKG knows all users' private keys. To overcome it, certificateless public key cryptography (CL-PKC) was proposed in [4], which has the significant advantages of certificateless and no escrow feature at the same time.

There have been a large number of certificateless signature (CLS) schemes so far, for example, [4-11]. And all of them are based on the hardness of the classical number theory problem, particularly the discrete logarithm assumptions. However, Shor indicated in [12] that the discrete logarithm problem and the integer factorization problem would no longer be hard when quantum computers came into reality. In view of the recent progress about quantum computers in [13], looking for a quantum-secure CLS scheme is very urgent.

Fortunately, Bernstein has conjectured in [14] that only some schemes can be reduced to computational problems on lattices, which are still hard even for quantum algorithms. What is more, lattice-based cryptographic schemes are also easy to implement because typical computations involved in them are only the integer matrix–vector multiplication and modular addition operations (refer to [15], for an overview on lattice-based

cryptography). And lattice-based cryptographic schemes are supported by the worst-case to average-case security guarantees. Considering these three advantages, lattice cryptography has entered a rapid development stage and the last ten years has met its achievements, such as cryptographic primitives [16-20], encryption schemes (public key encryption schemes [21-25], fully homomorphic encryption schemes [26-29]), signature schemes [16, 30-35]. The first lattice-based CLS scheme has been proposed in [36]. Nevertheless, its efficiency is not very satisfactory.

1.1 Our Contribution

In this paper, the first CLS scheme on NTRU lattice is proposed. We prove it is existentially unforgeable against strong adversaries in the random oracle model when the small integer solution (SIS) problem on NTRU lattice is hard. Moreover, the comparison between the two lattice-based CLS schemes indicates that the new CLS scheme is more efficient.

1.2 Paper Organization

The remainder of this paper is organized as follows. Section 2 presents some preliminaries. Section 3 describes the syntax and security model for CLS schemes. The first CLS scheme on NTRU lattice is provided in Section 4. Section 5 gives the efficiency comparison between the only two lattice-based CLS schemes. Finally, Section 6 concludes this paper.

2. Preliminaries

2.1 Notation

Throughout this paper, security parameter $n=2^t$ is a positive integer which is larger than 8. $\mathbb{R}$ and $\mathbb{Z}$ are the real space and integer space, respectively. We will work in the ring $R=\mathbb{Z}[x]/(x^n+1)$ and ring $R_q=\mathbb{Z}_q[x]/(x^n+1)$ where a prime q is bigger than 5. It is also satisfied that x^n+1 can split into k_q irreducible factors modulo prime q. $R^\times$ denotes the set of invertible elements in R. If vector $x\in\mathbb{R}^n$, then $||x||$ denotes the Euclidean norm of x. And for a matrix A, let a_i be the ith column of A and $||A||$ is defined as $\max_i(||a_i||)$.

Let $f=\sum_{i=0}^{n-1}f_ix^i$ and $g=\sum_{i=0}^{n-1}g_ix^i$ be polynomials in R.

- fg denotes polynomial multiplication in R, while $f*g=fg \bmod (x^n+1)$.

- (f) is the vector whose coordinates are respectively $f_0,...,f_{n-1}$. $(f,g)\in\mathbb{R}^{2n}=R^{1\times 2}$ is the concatenation of (f) and (g).

Definition 1(Anticirculant matrices). An n-dimensional anticirculant matrix of f is the following Toeplitz matrix:

$$C_n(f)=\begin{pmatrix} f_0 & f_1 & f_2 \ddots f_{n-1} \\ -f_{n-1} & f_0 & f_1 \ddots f_{n-2} \\ \ddots & \ddots & \ddots \ddots \ddots \\ -f_1 & -f_2 & \ddots \ddots f_0 \end{pmatrix}=\begin{pmatrix} (f) \\ (x*f) \\ \vdots \\ (x^{n-1}*f) \end{pmatrix}$$

When it is clear from context, we will drop the subscript n, and just write $C(f)$.

2.2 Lattices

An n-dimensional lattice is a full-rank discrete subgroup of $\mathbb{R}^n$. Here we focus on NTRU lattice.

Definition 2(NTRU lattice). Let q be a prime bigger than 5 and n be the power of 2. And $f, g \in R_q$ (f is invertible modulo q). Let $h=g*f^{-1} \bmod q$. The NTRU lattice associated to h and q is $\Lambda_{h,q}=\{(u,v)\in R^2 \mid u+v*h=0 \bmod q\}$. Here $\Lambda_{h,q}$ is a full-rank lattice of $\mathbb{R}^{2n}$ generated by the row of

$$A_{h,q}=\begin{pmatrix} -C_n(h) & I_n \\ qI_n & O_n \end{pmatrix}.$$

Where I_n and O_n are respectively the $n\times n$ unit matrix and $n\times n$ null matrix.

2.3 Gaussian on Lattices

Gaussian sampling was first proposed in [37] as a technique to use a short basis as a trapdoor without leaking any information about the short basis. The discrete gaussian distribution on lattice is defined as follows.

Definition 3(Discrete Gaussian distribution). For any $s>0$, $c\in\mathbb{R}^n$, define n-dimensional Gaussian function $\rho_{s,c}:\mathbb{R}^n\to(0,1]$ as

$$\rho_{s,c}(x)\triangleq \exp(-\pi\frac{\|x-c\|^2}{s^2}).$$

For any lattice $\Lambda\subset\mathbb{R}^n$, $\rho_{s,c}(\Lambda)\triangleq\sum_{x\in\Lambda}\rho_{s,c}(x)$. The probability mass function of the discrete Gaussian distribution is normalized as $D_{\Lambda,s,c}(x)=\rho_{s,c}(x)/\rho_{s,c}(\Lambda)$. For simplicity, in the rest of the paper, $D_{\Lambda,s,c}(x)$ will be abbreviated as $D_{\Lambda,s}(x)$.

In the following lemmas, we review several well-known facts about discrete Gaussian distribution.

Lemma 1[refer to [15]]: For any n-dimensional lattice Λ, center $c\in\mathbb{R}^n$, positive $\varepsilon>0$, $s>2\eta_\varepsilon(\Lambda)$. For any $x\in\Lambda$, we have

$$D_{\Lambda,s,c}(x)\le\frac{1+\varepsilon}{1-\varepsilon}2^{-n},$$

where $\eta_\varepsilon(\Lambda)$ is the smoothing parameter of the lattice Λ. For $\varepsilon<1/3$, the min-entropy of $D_{\Lambda,s,c}(x)$ is at least $n-1$.

Lemma 2: For any $\sigma>0$ and a positive integer m, we have

1. $\Pr[x \leftarrow D_{\mathbb{Z},\sigma} : |x| > 12\sigma] < 2^{-100}$.
2. $\Pr[x \leftarrow D_{\mathbb{Z}^m,\sigma} : \|x\| > 2\sigma\sqrt{m}] < 2^{-m}$.

Lemma 3[refer to [38]]: For any $v \in \mathbb{Z}^m$ and a positive real α, if $\sigma = \omega(\|v\|\sqrt{\log m})$, we have

$$\Pr[x \leftarrow D_{\mathbb{Z}^m,\sigma} : D_{\mathbb{Z}^m,\sigma}(x) / D_{\mathbb{Z}^m,\sigma,v}(x) = O(1)] = 1 - 2^{-\omega(\log m)},$$

and more specifically, if $\sigma = \alpha\|v\|$, then

$$\Pr[x \leftarrow D_{\mathbb{Z}^m,\sigma} : D_{\mathbb{Z}^m,\sigma}(x) / D_{\mathbb{Z}^m,\sigma,v}(x) < e^{12/\alpha+1/(2\alpha^2)}] = 1 - 2^{-100}.$$

The preimage sampling algorithm on NTRU lattice is defined as.

Algorithm 1:Gaussian_Sampler (B, σ, c) **:**

Input: Basis B of an n-dimensional lattice Λ, standard deviation σ>0, center $c \in \mathbb{Z}^n$.
Output: v sampled in $D_{\Lambda,\sigma,c}$

1: $v_n \leftarrow 0$
2: $c_n \leftarrow c$
3: for $i \leftarrow n,...,1$ do
4: $c_i' \leftarrow < c_i, b_i > / \|\tilde{b}_i\|^2$
5: $\sigma_i' \leftarrow \sigma / \|\tilde{b}_i\|^2$
6: $z_i \leftarrow Sample\ Z(\sigma_i, c_i')$
7: $c_{i-1} \leftarrow c_i - z_i b_i$ and $v_{i-1} \leftarrow v_i + z_i b_i$
8: return v_0

In this sampling algorithm above, the algorithm $Sample\ Z(\sigma_i, c_i')$ samples one-dimensional Gaussian $D_{\mathbb{Z},\sigma_i,c_i'}$, and $\tilde{B} = (\tilde{b}_i)_{i \in n}$ is the Gram–Schmidt orthogonalization of B.

2.4 Hardness Assumption

Definition 4 (SIS over ring R_q, namely R-SIS$_{q,m,\beta}$). The Small Integer Solution problem on ring R_q with parameters q, m, β and Φ is defined as follows: Given m polynomials a_1, a_2,…,a_m chosen uniformly and independently in $R_q = \mathbb{Z}_q[x]/(\Phi = x^n+1)$, a way to describe SIS on ring R_q is to find a solution $t \in a^{\perp} \backslash 0$ which satisfied that $\|t\| \le \beta$, where $a^{\perp} := \{(t_1, t_2, \ldots, t_m) \in R^m : \sum_i t_i a_i = 0 \bmod q\}$.

The trapdoor generation algorithm on NTRU lattice is somewhat different from that on general lattice, which is defined as shown in Algorithm 2 in the following. Here it s denoted as Trapdoor Generation.

Algorithm 2: Trapdoor Generation (n, q) **:**

Input: $n, q \in \mathbb{Z}, \sigma > 0$.

Output: $(B, h) \in \mathbb{R}^{2n \times 2n} \times R_q^{\times}$

1. Sample f and g from $D_{\mathbb{Z}^n,\sigma}$ that satisfy $(f \bmod q) \in R_q^{\times}$ and $(g \bmod q) \in R_q^{\times}$.

2. If $\|f\| > \sigma\sqrt{n}$ or $\|g\| > \sigma\sqrt{n}$, restart.

3. If $< f, g > \neq R$, restart.

4. Compute $F_1, G_1 \in R$ such that $fG_1 - gF_1 = 1$; Set $F_q = qF_1$ and $G_q = qG_1$.

5. Use Babai's nearest plane algorithm to approximate pair (F_q, G_q) by an integer linear combination of (f, g), (xf, xg), …, ($x^{n-1}f$, $x^{n-1}g$). Let (F, G) be the output, such that there exists $k \in R$ with $(F, G) = (F_q, G_q) - k(f, g)$.

6. If $\|(F, G)\| > n\sigma$, restart.

7. Return trapdoor basis $B = \begin{pmatrix} C(f) & C(g) \\ C(F) & C(G) \end{pmatrix}$ and polynomial $h = g/f \in R_q^{\times}$.

When f and g are chosen according to $D_{\mathbb{Z}^n,\sigma}$ ($\sigma > 0$ and $f, g \in R_q^{\times}$), Theorem 4.1 in [39] shows that the statistical distance between the distribution of $h=g/f$ and the uniform distribution in $R_q^{\times}$ is $2^{10n}q^{-\lfloor \varepsilon n \rfloor}$ which is negligible. So the SIS on NTRU lattice, namely R-SIS$_{q,2,\beta}$, can be defined in the following.

Definition 5 ((q,2,β)-SIS on NTRU lattice). A way to state the SIS problem on NTRU lattice is to set $R=\mathbb{Z}[x]/(x^n+1)$ and two small polynomials f, g are picked according to $D_{\mathbb{Z}^n,\sigma}$ ($\sigma > 0$ and $f, g \in R_q^{\times}$), $A=(h,1) \in R_q^{1\times 2}$ and $h=g/f$. So R-SIS$_{q,2,\beta}$ is to find the (z_1, z_2) which satisfies $A(z_1, z_2)^T=0$ and $\|(z_1, z_2)\| \leq \beta$.

Theorem 1[19]. Let $n=2^k$, $\Phi=x^n+1$ and $\varepsilon>0$. m and q are positive integers such that $q \geq \beta\sqrt{n} \cdot \omega(\log n)$ and m, $\log q \leq \mathrm{Poly}(n)$. If there exists a polynomial-time algorithm $\mathcal{A}$ solving R-SIS$_{q,.m,\beta}$ with non-negligible probability, a new algorithm $\mathcal{B}$ can be constructed to solve γ-Ideal-shortest vector problem (SVP) in polynomial-time with $\gamma \geq \beta\sqrt{n} \cdot \omega(\log n)$ by invoking algorithm $\mathcal{A}$.

So far, there is no algorithm which is known to perform non-negligibly better for γ-Ideal-SVP than for γ–SVP. According to the development of the algorithm, it is generally believed that there has not been any sub-exponential quantum algorithm that can solve the computational variants of γ–SVP or γ-Ideal-SVP in the worst case, for any γ that is polynomial in the dimension. And the smallest γ which is known to be achievable in polynomial time is exponential, up to poly-logarithmic factors in the exponent [40-42].

3. Syntax And Security Model for CLS scheme

3.1 Syntax

A CLS scheme is a set of 7 probabilistic polynomial-time (PPT) algorithms: **Setup**, **Extract-Partial-Private-Key**, **Set-Secret-Value**, **Set-Private-Key**, **Set-Public-Key**, **CL-Sign**, **CL-Verify** as follows.

Setup(n). Taking security parameter n as input, PKG outputs the master private/public key pair (msk, mpk). Note that PKG keeps msk secret.

Extract-Partial-Private-Key(msk, id). On input of the master private key msk and an identity id, PKG outputs a partial private key d_{id} and then sends it to the user via a secure channel.

Set-Secret-Value(id). Given an identity id, the user id outputs a secret value s_{id}.

Set-Private-Key(d_{id}, s_{id}). Taking the user id 's partial private key d_{id} and the secret value s_{id} as input, the user id runs this algorithm to output sk_{id} as the full private key.

Set-Public-Key(sk_{id}). On input of full private key sk_{id}, the user id outputs a public key pk_{id}.

CL-Sign(μ, id, sk_{id}). Given a message μ, the user's identity id and sk_{id}, the algorithm outputs a signature sig on μ.

CL-Verify(sig, μ, id, pk_{id}). On input of (sig, μ, id, pk_{id}), the algorithm outputs 1 if and only if sig is valid. Otherwise, outputs 0.

3.2 Security model for CLS scheme

In general, a secure CLS scheme should satisfy the following requirements:

(1) Correctness: The signature obtained from **CL-Sign** can be verified by the verifier.

(2) Unforgeability: When it comes to the unforgeability of the CLS scheme, we should consider two types of adversaries.

Type 1: The adversary models an outside attacker. So the Type 1 adversary can replace any user's public key with the value chosen by himself.

Type 2: The adversary models a malicious PKG. So the Type 2 adversary knows the master secret key msk.

However, neither Type 1 adversary nor Type 2 adversary can replace public keys and know the master secret key at the same time.

The security model consists of two games. ***Game*** 1 is played between a challenger C and a Type 1 adversary $\mathcal{A}_1$. The second game is the interaction between a challenger C and a Type 2 adversary $\mathcal{A}_2$, namely ***Game*** 2.

***Game* 1**. This game is played as follows.

Initialization: The challenger C runs the algorithms **Setup** to generate the master secret key msk. Here $\mathcal{A}_1$ is an outside attacker, so he cannot know the msk.

Queries: Adversary $\mathcal{A}_1$ can adaptively query all the oracles as follows.

(1) Create-User-Oracle. The oracle keeps the L_C-list which is a list of 5-tuples (id, d_{id}, s_{id}, sk_{id}, pk_{id}). Given an identity $id \in \{0,1\}^*$, the oracle looks up it in L_C-list. If id is found in L_C-list, pk_{id} will be returned as output. Otherwise, the oracle runs algorithms **Extract-Partial-Private-Key**, **Set-Secret-Value**, **Set-Private-Key** and **Set-Public-Key** to output d_{id}, s_{id}, sk_{id} and pk_{id}, respectively. And then the oracle stores (id, d_{id}, s_{id}, sk_{id}, pk_{id}) and returns pk_{id}.

(2) Extract-Partial-Private-Key-Oracle. Given an identity $id \in \{0,1\}^*$as input, challenger C looks up id in the L_C-list and returns the corresponding partial private key d_{id} to adversary $\mathcal{A}_1$.

(3) Extract-Secret-Value-Oracle. When $\mathcal{A}_1$ queries this oracle for identity $id \in \{0,1\}^*$, challenger C arises the L_C-list for id and the corresponding secret key s_{id} will be returned to adversary $\mathcal{A}_1$.

(4) Replace-Public-Key-Oracle. Given an identity id and a new public key pk_{id}', challenger C replaces the current public key with pk_{id}' and records this change successively.

(5) CL-Sign-Oracle. Taking an identity id, a message μ and a secret value x_{id} associated to the current public key pk_{id} as input, the challenger C first browses the L_C-list for sk_{id} and then runs **CL-Sign** to output a valid signature sig which will be verified by the public key pk_{id}. Note that if pk_{id} is derived from the Create-User-Oracle, $x_{id}=\perp$.

Forgery. Finally, adversary $\mathcal{A}_1$ outputs a forgery sig^* on (id^*, μ^*). Here pk_{id^*} is the current public key. In general, when we say $\mathcal{A}_1$ wins the game, it always means that (1) **CL-Verify** (sig^*, μ^*, id^*, pk_{id^*})=1 (2) (μ^*, id^*, x_{id^*}) has never been sent to the oracle Extract-Partial-Private-Key-Oracle for query (3) id^* has never appeared in the L_C-list.

***Game* 2**. Here a challenger C and a Type 2 adversary $\mathcal{A}_2$ interact with each other as follows.

Initialization: In order to generate the master secret key msk, challenger C runs the algorithms **Setup**. As the malicious PKG, $\mathcal{A}_2$ knows the master secret key msk.

Queries: Adversary $\mathcal{A}_2$ makes the following queries adaptively.

(1) Create-User-Oracle. The oracle keeps a list of 5-tuples (id, d_{id}, s_{id}, sk_{id}, pk_{id}), namely L_C-list. Given an identity $id \in \{0,1\}^*$, the oracle searches L_C-list. If id is found in the L_C-list, challenger C returns pk_{id} to adversary $\mathcal{A}_2$. Otherwise, the oracle successively runs the algorithms **Extract-Partial-Private-Key**, **Set-Secret-Value**, **Set-Private-Key** and **Set-Public-Key** to output (d_{id}, s_{id}, sk_{id}, pk_{id}). Finally, challenger C stores (id, d_{id}, s_{id}, sk_{id}, pk_{id}) in L_C-list and returns pk_{id} to adversary $\mathcal{A}_2$.

(2) Extract-Secret-Value-Oracle. Taking an identity $id \in \{0,1\}^*$as input, challenger C searches L_C-list for id and the secret key s_{id} will be returned to adversary $\mathcal{A}_2$.

(3) Replace-Public-Key-Oracle. Given an identity id and a new public key pk_{id}' as input, challenger C replaces the current public key with pk_{id}' and successively records this replacement.

(4) CL-Sign-Oracle. Given an identity *id*, a message μ and a secret value x_{id} associated to the current public key pk_{id}, challenger C arises L_C-list for sk_{id} and then runs the algorithm **CL-Sign** to generate a valid signature *sig* which can be verified with pk_{id}. Note that if pk_{id} is derived from the Create-User-Oracle, $x_{id}=\perp$.

Forgery. Finally, the adversary $\mathcal{A}_2$ outputs a forgery sig^* for (id^*, μ^*). Here pk_{id^*} is the current public key. When it comes into the condition that $\mathcal{A}_2$ wins the game, it always means: (1) **CL-Verify** $(sig^*, \mu^*, id^*, pk_{id^*})=1$ (2) (μ^*, id^*) has never been queried to the CL-Sign-Oracle (3) id^* has never been queried to the oracle Extract-Secret-Value-Oracle.

4. A CLS Scheme from Lattices

4.1 Construction

Let a prime $q=\tilde{\Omega}(\beta\sqrt{n})\geq 2$, n be the security parameter, κ be positive integers, $s=\Omega((q/n)\sqrt{\ln(8nq)})$, $\sigma=12s\kappa n$, H: $\{0,1\}^*\rightarrow\{v\in\mathbb{Z}_q^n\}$ and H_1: $\mathbb{Z}_q^{2n}\times\{0,1\}^*\rightarrow D_H$:$\{v\in\mathbb{Z}_q^n, 0\leq\|v\|_1\leq\kappa, \kappa<<q\}$. Our certificateless signature scheme on NTRU lattice is:

Setup(n). Taking security parameter n as input, the PKG runs the **Algorithm 2** to output a trapdoor $B=\begin{bmatrix} C(f), C(g) \\ C(F), C(G) \end{bmatrix}\in\mathbb{Z}_q^{2n\times 2n}=R_q^{2\times 2}$ as well as $h\in R_q^{\times}$ as the *msk* and *mpk*, respectively. Where B is the basis of the NTRU lattice $\Lambda_{h,q}$.

Extract-Partial-Private-Key(*msk*, *id*). Taking the master private key *msk* and an identity *id* as input, the PKG runs the preimage sampling algorithm on the NTRU lattice **Gaussian_Sampler**(B,s,($H(id)$,0)) to output (s_1,s_2). Then the PKG sends (s_1,s_2) to the user. And the user can verify whether $\|(s_1,s_2)\|\leq s\sqrt{2n}$ and $s_1+s_2*h=H(id)$. If so, the user takes (s_1,s_2) as d_{id}. Otherwise, rejects them.

Set-Secret-Value(*id*). The user *id* chooses $s_1^{'}, s_2^{'}\in D_{\mathbb{Z}^n,s}$ and outputs $s_{id}=(s_1^{'}, s_2^{'})$.

Set-Private-Key(d_{id}, s_{id}). Given the user *id* 's partial private key d_{id} and the secret value s_{id}, the user *id* outputs $sk_{id}=(d_{id}, s_{id})$ as the full private key.

Set-Public-Key(sk_{id}). Taking full private key sk_{id} as input, the user computes $pk_{id}=s_1^{'}+s_2^{'}*h$ and outputs pk_{id} as his public key.

CL-Sign(μ, *id*, sk_{id}). Given a message μ, the user's identity *id* and sk_{id}, the algorithm does as follows:

(1)Select random $y_1, y_2, y_1^{'}, y_2^{'}\in D_{\mathbb{Z}^n,\sigma}$ and define

$$y=\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, y^{'}=\begin{bmatrix} y_1^{'} \\ y_2^{'} \end{bmatrix}, \hat{y}=\begin{bmatrix} y \\ y^{'} \end{bmatrix}.$$

(2)Set $e = H_1(\begin{bmatrix} y_1 + y_2 * h \\ y_1^{'} + y_2^{'} * h \end{bmatrix}, \mu)$ and $z = \begin{bmatrix} z^{'} \\ z^{''} \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_1^{'} \\ z_2^{'} \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_1^{'} \\ s_2^{'} \end{bmatrix} * e + \begin{bmatrix} y_1 \\ y_2 \\ y_1^{'} \\ y_2^{'} \end{bmatrix}$.

(3)Output $sig=(e,z)$ with probability $\min(1, \frac{D_{\mathbb{Z}^n,\sigma}(z)}{MD_{\mathbb{Z}^n,\sigma,s_i e}(z)})$. If nothing is outputted, repeat this algorithm.

CL-Verify(sig, μ, id, pk_{id}). On input of (sig, μ, id, pk_{id}), the algorithm outputs 1 if and only if

(1)$\|z_1\| \le 2\sigma\sqrt{n}$, $\|z_2\| \le 2\sigma\sqrt{n}$, $\| z_1^{'} \| \le 2\sigma\sqrt{n}$ and $\| z_2^{'} \| \le 2\sigma\sqrt{n}$

(2)$e= H_1(\begin{bmatrix} z_1 + z_2 * h \\ z_1^{'} + z_2^{'} * h \end{bmatrix} - \begin{bmatrix} H(id) \\ pk_{id} \end{bmatrix} * e, \mu)$.

4.2 Correctness

Theorem 1. The lattice-based CLS scheme satisfies correctness.

Proof. According to the **CL-Sign** phase, we know

$$\begin{aligned} &\begin{bmatrix} z_1 + z_2 * h \\ z_1^{'} + z_2^{'} * h \end{bmatrix} - \begin{bmatrix} H(id) \\ pk_{id} \end{bmatrix} * e \\ &= \begin{bmatrix} z_1 \\ z_1^{'} \end{bmatrix} + \begin{bmatrix} z_2 \\ z_2^{'} \end{bmatrix} * h - \begin{bmatrix} H(id) \\ pk_{id} \end{bmatrix} * e \\ &= \begin{bmatrix} y_1 \\ y_1^{'} \end{bmatrix} + \begin{bmatrix} s_1 \\ s_1^{'} \end{bmatrix} * e + \left(\begin{bmatrix} y_2 \\ y_2^{'} \end{bmatrix} + \begin{bmatrix} s_2 \\ s_2^{'} \end{bmatrix} * e \right) * h - \left(\begin{bmatrix} s_1 \\ s_1^{'} \end{bmatrix} + \begin{bmatrix} s_2 \\ s_2^{'} \end{bmatrix} * h \right) * e \\ &= \begin{bmatrix} y_1 \\ y_1^{'} \end{bmatrix} + \begin{bmatrix} y_2 \\ y_2^{'} \end{bmatrix} * h \\ &= \begin{bmatrix} y_1 + y_2 * h \\ y_1^{'} + y_2^{'} * h \end{bmatrix} \end{aligned}$$

So the valid signature $sig=(e,z)$ derived from **CL-Sign** will satisfy the equality $e= H_1(\begin{bmatrix} z_1 + z_2 * h \\ z_1^{'} + z_2^{'} * h \end{bmatrix} - \begin{bmatrix} H(id) \\ pk_{id} \end{bmatrix} * e, \mu)$.

In addition, it is obvious that the distributions of z_1, z_2, $z_1^{'}$ and $z_2^{'}$ are very close to $D_{\mathbb{Z}^n,\sigma}$ by combining the rejecting technique and Theorem 3.4 in [38]. According to

Lemma 2, we have $\|z_1\|\leq 2\sigma\sqrt{n}$, $\|z_2\|\leq 2\sigma\sqrt{n}$, $\|z_1'\|\leq 2\sigma\sqrt{n}$ and $\|z_2'\|\leq 2\sigma\sqrt{n}$ with probability at least $1-2^{-n}$.

4.3 Security

Theorem 2. The CLS scheme is proven existentially unforgeable against strong adversaries in random oracle model, under the assumption the γ-Ideal-SVP against polynomial-time algorithm is hard.

Lemma 4. If the $(q, 2, (4\sigma+2\kappa s)\sqrt{n})$-SIS on NTRU Lattice $\Lambda_{h,q}$ is intractable, the new CLS scheme is existentially unforgeable against any polynomial-time strong Type 1 adversary in the random oracle model.

Proof. Assuming there is a PPT adversary $\mathcal{A}_1$ who breaks the new CLS scheme with non-negligible probability, we can construct a simulator C to solve the SIS problem on NTRU lattice as follows.

Invocation: Being invoked on a random instance of the $(q,2,\beta)$-SIS problem on NTRU lattice $\Lambda_{h,q}$, the simulator C is required to return a valid solution.

—Supplied: a polynomial $h\in R_q^{\times}$ and NTRU lattice $\Lambda_{h,q}$.

—Requested: $(s_1, s_2)\in \Lambda_{h,q}$ and $\|(s_1, s_2)\|\leq \beta$.

Queries: $\mathcal{A}_1$ can adaptively query all the oracles shown next:

(1) H-Oracle query. The simulator C keeps a list L_H-list which is a list of 3-tuples (id_i, $d_{id_i}=(s_{i1}, s_{i2})$, $s_{i1}+s_{i2}*h$). Taking an identity $id_i\in\{0,1\}^*$ as input, C looks up it in L_H-list. If id_i is found in L_H-list, the simulator C returns the $s_{i1}+s_{i2}*h$ to adversary $\mathcal{A}_1$. Otherwise, C picks two polynomials s_{i1}, s_{i2} from $D_{\mathbb{Z}^n,s}$, stores (id_i, $d_{id_i}=(s_{i1}, s_{i2})$, $s_{i1}+s_{i2}*h$) and returns $s_{i1}+s_{i2}*h$ successively to adversary $\mathcal{A}_1$.

(2) Creat-User-Oracle query. The simulator C keeps a list L_C-list which is a list of 4-tuples (id_i, $d_{id_i}=(s_{i1}, s_{i2})$, $s_{id_i}=(s_{i1}', s_{i2}')$, pk_{id_i}). On input of an identity id_i, the simulator C does as follows. If id_i is found in L_C-list, then C returns pk_{id_i}. Otherwise, the simulator C arises the H-Oracle query for d_{id_i}. Then the algorithm **Set-Secret-Value** and the algorithm **Set-Public-Key** will be performed by the simulator C to output secret value $s_{id_i}=(s_{i1}', s_{i2}')$ and $pk_{id_i}=s_{i1}'+s_{i2}'*h$, respectively. Finally, the tuples (id_i, $d_{id_i}=(s_{i1}, s_{i2})$, $s_{id_i}=(s_{i1}', s_{i2}')$, pk_{id_i}) will be stored in L_C-list and the simulator C returns pk_{id_i} to adversary $\mathcal{A}_1$.

(3) Extract-Partial-Private-Key-Oracle query. Taking an identity id_i as input, the simulator C searches the L_C-list for the partial private key s_{id_i}.

(4) Replace-Public-Key-Oracle query. On input of an identity id_i and a new public

key $pk_{id_i}^{'}$, the simulator C looks up the corresponding public key pk_{id_i} and replaced it with $pk_{id_i}^{'}$. Finally, this replacement will be recorded by the simulator C later.

(5) H_1-Oracle query. The simulator C keeps the L_{H1}-list which is $(\begin{bmatrix} y_{i1}+y_{i2}*h \\ y_{i1}^{'}+y_{i2}^{'}*h \end{bmatrix}, \mu, e_i)$. Taking $\begin{bmatrix} y_{i1}+y_{i2}*h \\ y_{i1}^{'}+y_{i2}^{'}*h \end{bmatrix}, \mu$ as input, the simulator C looks up them in L_{H1}-list. If they are found in L_{H1}-list, C returns the corresponding e_i. Otherwise, C randomly selects e_i from D_H, and stores $(\begin{bmatrix} y_{i1}+y_{i2}*h \\ y_{i1}^{'}+y_{i2}^{'}*h \end{bmatrix}, \mu, e_i)$ in L_{H1}-list. Finally, the simulator C returns e_i to adversary $\mathcal{A}_1$.

(6) CL-Sign-Oracle query. On input of a message μ, a user's identity id_i and x_{id_i}. The simulator C first searches the H-Oracle query for d_{id_i}, then the **CL-Sign** algorithm will be run to return a signature sig. Note that if pk_{id_i} is the user's current public key(that is to say pk_{id_i} has not been replaced), then $x_{id_i} = \perp$. In this case, C can run the **CL-Sign** algorithm to generate a valid signature.

Forgery: Finally, adversary $\mathcal{A}_1$ outputs a valid forgery $sig^*=(e^*, z^*)$ on (μ^*, id^*, pk_{id^*}) with non-negligible probability.

The simulator C can solve the SIS problem on NTRU lattice as follows.

After receiving the forgery $sig^*=(e^*, z^*)$, the simulator C will output a new forgery $sig^{'}=(e^{'}, z^{'})$ on the same (μ^*, id^*, pk_{id^*}) by the forking lemma[43]. So we get

$$\begin{bmatrix} z_1^*+z_2^**h \\ z_1^{'*}+z_2^{'*}*h \end{bmatrix} - \begin{bmatrix} s_1^*+s_2^**h \\ s_1^{'*}+s_2^{'*}*h \end{bmatrix} * e^* = \begin{bmatrix} z_1^{'}+z_2^{'}*h \\ z_1^{''}+z_2^{''}*h \end{bmatrix} - \begin{bmatrix} s_1^{'}+s_2^{'}*h \\ s_1^{''}+s_2^{''}*h \end{bmatrix} * e^{'}$$

So $\begin{bmatrix} (z_1^*-z_1^{'})+(z_2^*-z_2^{'})*h \\ (z_1^{'*}-z_1^{''})+(z_2^{'*}-z_2^{''})*h \end{bmatrix} = \begin{bmatrix} s_1^*+s_2^**h \\ s_1^{'*}+s_2^{'*}*h \end{bmatrix} * (e^*-e^{'})$ holds. And then the inequality $[(z_1^*-z_1^{'})-s_1^**(e^*-e^{'})]+[(z_2^*-z_2^{'})+s_2^*(e^*-e^{'})]*h=0$ also holds. Because of the inequality $\|(z_1^*-z_1^{'})-s_1^**(e^*-e^{'})\| \le \| z_1^* \| + \| z_1^{'} \| + \| s_1^* \| \cdot \| e^*-e^{'} \| \le (4\sigma+2\kappa s)\sqrt{n}$ holds and $\|(z_2^*-z_2^{'})+s_2^*(e^*-e^{'})\| \le \| z_2^* \| + \| z_2^{'} \| + \| s_2^* \| \cdot \| e^*-e^{'} \| \le (4\sigma+2\kappa s)\sqrt{n}$ also holds, so $([(z_1^*-z_1^{'})-s_1^**(e^*-e^{'})],[(z_2^*-z_2^{'})+s_2^*(e^*-e^{'})])$ is a solution to the SIS problem on NTRU lattice above, where $\beta \ge (4\sigma+2\kappa s)\sqrt{2n}$.

Lemma 5. If the $(q, 2, (4\sigma+2\kappa s)\sqrt{2n})$ SIS on NTRU $\Lambda_{h,q}$ is intractable, the new CLS scheme is existentially unforgeable against any polynomial-time strong Type 2 adversary

in the random oracle model.

Proof. Assuming there is a PPT adversary $\mathcal{A}_2$ who breaks the new CLS scheme with non-negligible probability, we can construct a simulator C to solve the SIS problem on NTRU lattice as follows.

Invocation: Simulator C is invoked on a random instance of the (q,2,β)-SIS problem on NTRU lattice $\Lambda_{h,q}$, and is asked to return an admissible solution.

—Supplied: a polynomials $h \in R_q^{\times}$ and NTRU lattice $\Lambda_{h,q}$.

—Requested: $(s_1,s_2) \in \Lambda_{h,q}$ and $\|(s_1,s_2)\| \le \beta$.

Queries: $\mathcal{A}_2$ can adaptively query all the oracles shown next:

(1) H-Oracle query. The simulator C keeps a list L_H-list which is a list of 3-tuples (id_i, $d_{id_i}=(s_{i1},s_{i2})$, $s_{i1}+ s_{i2}*h$). Taking an identity $id_i \in \{0,1\}^*$ as input, C looks up it in L_H-list. If id_i is found in L_H-list, the simulator C returns the $s_{i1}+ s_{i2}*h$ to adversary $\mathcal{A}_1$. Otherwise, C first runs **Extract-Partial-Private-Key** to obtain a partial private key $d_{id_i}=(s_{i1},s_{i2})$, stores (id_i, $d_{id_i}=(s_{i1},s_{i2})$, $s_{i1}+ s_{i2}*h$) and returns $s_{i1}+ s_{i2}*h$ successively to adversary $\mathcal{A}_2$.

(2) Creat-User-Oracle query. The simulator C keeps a list L_C-list which is a list of 4-tuples (id_i, $d_{id_i}=(s_{i1},s_{i2})$, $s_{id_i}=(s_{i1}^{'},s_{i2}^{'})$, pk_{id_i}). On input of an identity id_i, the simulator C does as follows. If id_i is found in L_C-list, then C returns pk_{id_i}. Otherwise, the simulator C arises the H-Oracle query for d_{id_i}. Then the algorithm **Set-Secret-Value** and the algorithm **Set-Public-Key** will be performed by the simulator C to output secret value $s_{id_i}=(s_{i1}^{'},s_{i2}^{'})$ and $pk_{id_i}=s_{i1}^{'}+s_{i2}^{'}*h$, respectively. Finally, the tuples (id_i, $d_{id_i}=(s_{i1},s_{i2})$, $s_{id_i}=(s_{i1}^{'},s_{i2}^{'})$, pk_{id_i}) will be stored in L_C-list and the simulator C returns pk_{id_i} to adversary $\mathcal{A}_2$.

(3) Extract-Partial-Private-Key-Oracle query. Given an identity id_i, C arises the L_C-list and returns the partial private key s_{id_i}.

(4) Replace-Public-Key-Oracle query. Taking an identity id_i and a new public key $pk_{id_i}^{'}$ as input, simulator C looks up the corresponding public key pk_{id_i} and replaced it by $pk_{id_i}^{'}$. Finally, this replacement will be recorded.

(5) H_1-Oracle query. C keeps a list L_{H1}-list which is $(\begin{bmatrix} y_{i1}+y_{i2}*h \\ y_{i1}^{'}+y_{i2}^{'}*h \end{bmatrix}, \mu, e_i)$ and is initially empty. Given $\begin{bmatrix} y_{i1}+y_{i2}*h \\ y_{i1}^{'}+y_{i2}^{'}*h \end{bmatrix}, \mu$ as input, the simulator C looks up them in L_{H1}-list. If they are found in L_{H1}-list, C returns the corresponding e_i. Otherwise, C randomly selects

e_i from D_H, and stores ($\begin{bmatrix} y_{i1}+y_{i2}*h \\ y_{i1}^{'}+y_{i2}^{'}*h \end{bmatrix}, \mu, e_i$) in L_{H1}-list and returns e_i.

(6) CL-Sign-Oracle query. Given a message μ, a user's identity id_i and s_{id_i} which is associated with the user's current public key pk_{id_i}. C firstly arises the L_C-list for d_{id_i}, then C runs **CL-Sign** algorithm to return a signature sig.

Forgery: Finally, adversary $\mathcal{A}_2$ outputs a valid forgery $sig^*=(e^*, z^*)$ on message(μ^*,id^* pk_{id^*}) with non-negligible probability. In this case, the public key pk_{id^*} is the original one created by C.

Simulator C can solve the SIS problem on NTRU lattice as follows.

After receiving the forgery $sig^*=(e^*, z^*)$, $\mathcal{A}_2$ will output a new forgery $sig^{'}=(e^{'}, z^{'})$ on the same message (μ^*,id^*, pk_{id^*}) by the forking lemma in [43]. So we get

$$\begin{bmatrix} z_1^*+z_2^**h \\ z_1^{''}+z_2^{''}*h \end{bmatrix}-\begin{bmatrix} s_1^*+s_2^**h \\ s_1^{''}+s_2^{''}*h \end{bmatrix}*e^* = \begin{bmatrix} z_1^{'}+z_2^{'}*h \\ z_1^{''}+z_2^{''}*h \end{bmatrix}-\begin{bmatrix} s_1^{'}+s_2^{'}*h \\ s_1^{''}+s_2^{''}*h \end{bmatrix}*e^{'}$$

So $[(z_1^{''}-z_1^{''})-s_1^{''}*(e^*-e^{'})]+[(z_2^{''}-z_2^{''})-s_2^{''}*(e^*-e^{'})]*h=0 \bmod q$. Because of the inequality $\|(z_1^{''}-z_1^{''})-s_1^{''}*(e^*-e^{'})\| \le \| z_1^{''}\|+\| z_1^{''}\|+\| s_1^{''}\|\cdot\| e^*-e^{'}\| \le (4\sigma+2\kappa s)\sqrt{n}$ holds and $\|(z_2^{''}-z_2^{''})-s_2^{''}*(e^*-e^{'})\| \le \| z_2^{''}\|+\| z_2^{''}\|+\| s_2^{''}\|\cdot\| e^*-e^{'}\| \le (4\sigma+2\kappa s)\sqrt{n}$ also holds. So $([(z_1^{''}-z_1^{''})-s_1^{''}*(e^*-e^{'})],[(z_2^{''}-z_2^{''})-s_2^{''}*(e^*-e^{'}))$ is the solution to the SIS problem on NTRU lattice above, where $\beta \ge (4\sigma+2\kappa s)\sqrt{2n}$.

Applying Theorem 1, Lemma 4 and Lemma 5, we obtain Theorem 2. Fortunately, the security proof for our CLS scheme falls in the class of "history-free" reductions as defined in [44], so it is shown to imply security in the quantum-accessible random oracle model.

5. The Efficiency

There has been a CLS scheme [36] from lattice which is proven security in random oracle model. Now we compare our new CLS scheme with [36] as follows.

Table 1. The efficiency comparison between two existing CLS schemes

The scheme	*msk*	d_{id}	s_{id}	*z*
[36]	$(m_1)^2\log O(\sqrt{n\log q})$	$(m_1\cdot k)\log(s_1\sqrt{m_1})$	$(m_2\cdot k)\log(2b+1)$	$m\log(2\sigma_1)$
Our scheme	$4n\log(s\sqrt{n})$	$2n\log(s\sqrt{n})$	$2n\log(s\sqrt{n})$	$4n\log(2\sigma)$

Table 1 above lists the comparison on the communication overhead of our new scheme and the existed scheme [36]. Where $m_1 \geq 2n\log q$, $m_2 \geq 64+n\log q/(2b+1)$, $m=m_1+m_2$, $s_1=\Omega(\sqrt{n\log q})$, $s=\Omega((q/n)\sqrt{\ln(8nq)})$, k, b, κ are positive integers and $\sigma_1=12s_1\kappa m$, $\sigma=12s\kappa n$. So it is obvious that the master secret key, the private key and the signature in the new scheme are considerably shorter than that in [36]. Here, we compare the concrete instances between [36] and our new scheme in **Table 2** to prove that the master secret key, the private key and the signature size of the scheme [36] are unsatisfactory and our new scheme's master secret key, private key and the signature are much shorter. So we believe the new scheme is more efficient than the existed scheme [36] in terms of communication overhead.

Table 2. Comparison of the concrete instances

	Instance 1	Instance 2	Instance 3	Instance 4	Instance 5
n	512	512	512	512	512
q	2^{27}	2^{25}	2^{33}	2^{18}	2^{26}
k	80	512	512	512	512
κ	28	14	14	14	14
b	1	1	31	1	31
msk size in [34] (bits)	125584730	106436585	191548931	53092869	115590041
msk size in the new scheme	9789	8532	13551	4123	9192
d_{id} size in [34] (bits)	2512250	14739519	20076886	10217641	15394103
d_{id} size in the new scheme	7669	7041	9550	4836	7370
s_{id} size in [34] (bits)	335359	1988244	2996866	1436158	1822863
s_{id} size in the new scheme	7669	7041	9550	4836	7370
z size in [34] (bits)	151817	134978	114186	94951	131792
z size in the new scheme	23902	22030	27049	17620	22689

6. Conclusion

With the significant advantages of certificateless and no escrow feature, the CLS scheme has absorbed the general attention since it was proposed. However, when quantum computer comes into reality, the CLS scheme based on number theory is no longer secure. So looking for quantum-immune CLS scheme is urgent. Lattice may be the best candidate. The only lattice-based CLS scheme was proposed in [36] in 2014. Nevertheless, the efficiency of the lattice-based CLS scheme in the random oracle is not very satisfactory. This paper described the first efficient CLS scheme on NTRU lattice. It is proved secure in random oracle model. And the master secret key, the private key and the signature size in the new scheme are considerably shorter than that in [36]. An efficient lattice-based CLS scheme in standard model will be our future work.

Reference

[1] D. Arroyo, J. Diaz and F. B. Rodriguez, “Non-conventional Digital Signatures and Their Implementations-A Review,” in *Proc. of International Joint Conference 2015, Advances in*

Intelligent Systems and Computing, pp.425-435, May 27, 2015. Article (CorossRefLink).
[2] P. Zhou, *Research on Special Digital Signatures*, Southwest Jiaotong University, China. Article (CrossRef Link).
[3] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. of Cryptology–CRYPTO 1984*, pp. 47-53, August 19-22, 1984. Article (CrossRef Link).
[4] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. of Cryptology—Asiacrypt 2003*, pp. 452-473, November 30 - December 4, 2003. Article (CrossRef Link).
[5] X. Huang, W. Susilo, Y. Mu and F. Zhang, "On the security of certificateless signature schemes from Asiacrypt 2003," in *Proc. of the 4th International Conference on Cryptology and Network Security (CANS'05)*, pp. 13-25, December 14-16, 2005. Article (CrossRef Link).
[6] Z. Zhang, D. S. Wong, J. Xu and D. Feng, "Certificateless public-key signature: security model and efficient construction," in *Proc. of the 4th International Conference on Applied Cryptography and Network Security (ACNS'06)*, pp. 293-308, June 6-9, 2006. Article (CrossRef Link).
[7] X. Huang, Y. Mu, W. Susilo, D. S. Wong and W. Wu, "Certificateless signature revisited," in *Proc. of the 12th Australasian Conference on Information Security and Privacy (ACISP'07)*, pp. 308-322, July 2-4, 2007. Article (CrossRef Link).
[8] J. K. Liu, M. H. Au and W. Susilo, "Self-generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model," in *Proc. of the 2nd ACM Symposium on Information, Computer and Communications Security (AsiaCCS'07)*, pp. 273-283, March 20-22, 2007. Article (CrossRef Link).
[9] B. G. Kang, J. H. Park and S. G. Hahn, "A certificate-based signature scheme," in *Proc. of Cryptology—CT-RSA 2004*, pp. 99-111, February 23-27, 2004. Article (CrossRef Link).
[10] J. Li, X. Huang, Y. Mu, W. Susilo and Q. Wu, "Certificatebased signature: security model and efficient construction," in *Proc. of the 4th European Public Key Infrastructure Workshop (EuroPKI'07)*, pp. 110-125, June 28-30, 2007. Article (CrossRef Link).
[11] J. K. Liu, J. Baek, W. Susilo and J. Zhou, "Certificate-based signature schemes without pairings or random oracles," in *Proc. of the 11th Information Security Conference (ISC'08)*, pp. 285-297, September 15-18, 2008. Article (CrossRef Link).
[12] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal of Computing*, vol. 26, no. 5, pp. 1484-1509, November, 1997. Article (CrossRef Link).
[13] M. Krenn, M. Huber, R. Fickler, R. Lapkiewicz, S. Ramelow and A. Zeilinger, "Generation and confirmation of a (100×100) dimensional entangled quantum system," in *Proc. of the national academy of the United States of America*, vol. 111, no. 17, pp. 6243-6247, April, 2014. Article (CrossRef Link).
[14] D. J. Bernstein, "Introduction to Post-Quantum Cryptography," D. J. Bernstein, J. Buchmann, E. Dahmen (Eds), *Post-Quantum Cryptography*, Springer-Verlag, Berlin, pp.1-14. Article (CrossRef Link).
[15] O. Regev, "Lattice-based cryptography," in *Proc. of the 26th Annual International Cryptology Conference*, pp.131-141, August 20-24, 2006. Article (CrossRef Link).
[16] C. Gentry, C. Peikert and V. Vaikuntanathan, "Trapdoors for Hard Lattices and New Cryptographic Constructions," in *Proc. of the 40th Annual ACM Symposium on Theory of Computing*, pp. 197-206, May 17-20, 2008. Article (CrossRef Link).
[17] J. Alwen and C. Peiker, "Generating shorter bases for hard random lattices," *Theory of*

Computing Systems, vol. 48, no. 3, pp.535-553, April, 2011. Article (CrossRef Link).
[18] D. Micciancio and C. Peikert, "Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller," in *Proc. of Cryptology–Eurocrypt 2012*, pp. 700-718, April 15-19, 2012. Article (CrossRef Link).
[19] T. Laarhoven, M. Mosca and J. van de Pol, "Finding shortest lattice vectors faster using quantum search," *Designs, Codes and Cryptography*, vol. 77, vol. 2, pp. 375-400, December, 2015. Article (CrossRef Link).
[20] V. Lyubashevsky and D. Wichs, "Simple lattice trapdoor sampling from a broad class of distributions," in *Proc. of 18th IACR International Conference on Practice and Theory in Public-Key Cryptography–PKC 2015*, pp. 716-730, March 30-April 1, 2015. Article (CrossRef Link).
[21] D. Cash, D. Hofheinz, E. Kiltz, et al, "Bonsai trees, or how to delegate a lattice basis," in *Proc. of Cryptology–Eurocrypt 2010*, pp. 523-552, May 30-June 3, 2010. Article (CrossRef Link).
[22] S. Agrawal, D. Boneh and X. Boyen, "Efficient lattice (H)IBE in the standard model," in *Proc. of Cryptology–Eurocrypt 2010*, pp. 553-572, May 30-June 3, 2010. Article (CrossRef Link).
[23] S. Agrawal, D. Boneh and X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE," in *Proc. of Cryptology–CRYPTO 2010*, pp.98-115, August 15-19, 2010. Article (CrossRef Link).
[24] D. Stehlé and R. Steinfeld, "Making NTRU as secure as worst-case problems over ideal lattices," in *Proc. of Cryptology–Eurocrypt 2011*, pp. 27-47, May 15-19, 2011. Article (CrossRef Link).
[25] L. Ducas, V. Lyubashevsky and T. Prest, "Efficient Identity-Based Encryption over NTRU Lattices," in *Proc. of Cryptology–Asiacrypt 2014*, pp. 22-41, December 7-11, 2014. Article (CrossRef Link).
[26] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. of 41st Annual ACM Symposium on Theory of Computing (STOC 2009)*, pp. 169-178, May 31-June 2, 2009. Article (CrossRef Link).
[27] C. Gentry, "Toward basing fully homomorphic encryption on worst-case hardness," in *Proc. of Cryptology–CRYPTO 2010*, pp. 116-137, August 15-19, 2010. Article (CrossRef Link).
[28] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *Proc. of Cryptology–CRYPTO 2011*, pp.505-524, August 14-18, 2011. Article (CrossRef Link).
[29] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proc. of IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011)*, pp. 97-106, October 23-25, 2011. Article (CrossRef Link).
[30] X. Boyen, "Lattice mixing and vanishing trapdoors: a framework for fully secure short signature and more," in *Proc. of 13th International Conference on Practice and Theory in Public Key Cryptography (PKC 2010)*, pp. 499-517, May 26-28, 2010. Article (CrossRef Link).
[31] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Proc. of Cryptology–Eurocrypt 2012*, pp. 738-755, April 15-19, 2012. Article (CrossRef Link).
[32] L. Ducas, A. Durmus, T. Lepoint and V. Lyubashevsky, "Lattice signatures and bimodal Gaussians," in *Proc. of Cryptology–CRYPTO 2013*, pp.40-56, August 18-22, 2013. Article (CrossRef Link).

[33] F. Laguillaumie, A. Langlois, B. Libert and D. Stehlé, "Lattice-Based Group Signatures with Logarithmic Signature Size," in *Proc. of Cryptology–Asiacrypt 2013*, pp. 41-61, December 1-5, 2013. Article (CrossRef Link).
[34] A. Langlois, S. Ling, K. Nguyen and H. X. Wang, "Lattice-based group signature scheme with verifier-local revocation," in *Proc. of PKC 2014*, pp. 345-361, March 26-28, 2014. Article (CrossRef Link).
[35] P. Q. Nguyen, J. Zhang, Z. F. Zhang, "Simpler Efficient Group Signatures from Lattices," in *Proc. of PKC 2015*, pp. 401-426, March 30-April 1, 2015. Article (CrossRef Link).
[36] M. M. Tian and L. S. Huang, "Certificateless and certificate-based signatures from lattices," *Security and Communication Networks*, vol. 2015, no. 8, pp.1575-1586, 2015. Article (CrossRef Link).
[37] C. Gentry, C. Peikert, V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. of the 40th Annual ACM Symposium on Theory of Computing*, pp.197-206, May 17-20, 2008. Article (CrossRef Link).
[38] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Proc. of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp.738–755, April 15-19, 2012. Article (CrossRef Link).
[39] D. Stehlé and R. Steinfeld, "Making NTRUEncrypt and NTRUSign as Secure as Standard Worst-Case Problems over ideal lattices," *IACR Cryptology ePrint Archive* 2013:4, 2013. Article (CrossRef Link).
[40] A. K. Lenstra, H. W. Lenstra, and L. Lovâsz, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, no.4, pp. 515-534, 1982. Article (CrossRef Link).
[41] C. P. Schnorr, "A hierarchy of polynomial time lattice basis reduction algorithms," *Theoretical Computer Science*, vol. 53, no. 2-3, pp. 201-224, 1987. Article (CrossRef Link).
[42] D. Micciancio and P. Voulgaris, "A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations," in *Proc. of STOC 2010*, pp. 351-358, June 5-8, 2010. Article (CrossRef Link).
[43] M. Bellare and G. Neven, "Multi-signatures in the plain public-key model and a general forking lemma," in *Proc. of the 13th ACM Conference on Computer and Communications Security*, pp. 390-399, October -3 November, 2006. Article (CrossRef Link).
[44] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry, "Random oracles in a quantum world," in *Proc. of Asiacrypt 2011*, pp. 41-69, December 4-8, 2011. Article (CrossRef Link).

Jia Xie is a PhD student in Xidian University. She received her BS in Communication Engineering from Henan Normal University, China in 2011 and she takes a successive postgraduate and doctoral program. Her research interests include public key cryptography and quantum computation and quantum attack. (E-mail xiejia199325@163.com).

Yupu Hu is a professor and PhD supervisor of the School of Telecommunications Engineering, Xidian University, China. He received his PhD in cryptography from Xidian University, China in 1999, and received his MS and BS in mathematics from Xidian University, China in 1987 and 1982, respectively. His main research interests include public key cryptography based on lattices and the analysis and application of fully homomorphic encryption schemes.

Juntao Gao is a professor and MS supervisor of the School of Telecommunications Engineering, Xidian University, China. He received his PhD in cryptography from Xidian University, China in 2006, and he has received his BS in mathematics from Xidian University in 2001. His main research interests include pseudorandom sequence and stream cipher.

Wen Gao is a PhD student in Xidian University. She received her BS in Electronic Information Engineering from Henan University of Technology, China in 2011 and she takes a successive postgraduate and doctoral program. Her research interests include public key cryptography and quantum computation and quantum attack.

Mingming Jiang is a lecturer in the School of Computer Science and Technology, Huaibei Normal University. He received his PhD in cryptography from Xidian University in 2014, and received his MS and BS in cryptography from Huaibei Normal University in 2010 and 2007, respectively. His research interests include public key cryptography based on lattice and provable security.