# LATTICE-BASED STRONG DESIGNATE VERIFIER SIGNATURE AND ITS APPLICATIONS

***Fenghe Wang*[*], *Yupu Hu, Baocang Wang***
Key Laboratory of Computer Networks & Information Security of Ministry of Education, Xidian University,
Xi'an 710071, China
[*]Fenghe2166@163.com, Yuph@mail.xidian.edu.cn, Bcwang79@yahoo.com.cn

## *ABSTRACT*

*Motivated by the need to have secure strong designate verifier signatures (SDVS) even in the presence of quantum computers, a post-quantum lattice-based SDVS scheme is proposed based on the hardness of the short integer solution problem (SIS) and the learning with errors problem (LWE). The proposed SDVS scheme utilizes the Bonsai trees and pre-image sample-able function primitives to generate the designate verifier signature (DVS). In this construction, the un- forge-ability is based on the hardness of the SIS problem which is proven in the random oracle model and the non-transferability is based on the hardness of the LWE problem. As an application of the proposed SDVS scheme, we design a strong designate verifier ring signature scheme (SDVRS) which satisfies non-transferability. It is proven that the identity of the signer is unconditionally protected not only for any third-party but also for the designate verifier. Under the hardness of the SIS problem, the proposed SDVRS scheme is proven to be existentially un-forgeable in the random oracle model.*

**Keywords:** *Strong designate verifier signature, Strong designate verifier ring signature, Lattice-based cryptography, Pre-image sample-able function, Bonsai trees, Short integer solution problem.*

## 1.0 INTRODUCTION

The designate verifier signature (DVS) was first introduced by Jakobsson et.al. [1] which allows no one but the designated party to verify that the signer indeed generates this signature. In a DVS scheme, the designate verifier can produce the simulated signature, so he can not make anyone else believe that the DVS is generated by the signer. However, the DVS scheme can not prevent an attacker from eavesdropping on the line between the signer and the designate verifier to get the signature before the designate verifier receives it. Jakobsson et al. introduced a strong notion of DVS called the strong designate verifier signature (SDVS) against this attack [1]. The private key of the designate verifier is needed to verify the signature in an SDVS scheme. Hence, no third party but the designate verifier can verify the validity of an SDVS. For their wide applications, many DVS or SDVS have been proposed [2-5]. Furthermore, the notion of SDVS has been extended to other cryptographic fields, for example, ring signature. Combining the ring signature and the SDVS, a new cryptographic primitive which is called the designate verifier ring signature is obtained [6]. Although there have been many proposed SDVS schemes which yield very elegant constructions, these proposed schemes are not secure in the quantum era, for, according to the literature [7], both factorization and discrete logarithm problems can be solved in polynomial time on the quantum computer. Hence, it is necessary to design a secure SDVS scheme that is robust even in the quantum era. Note that lattice problems are hard even on the quantum computer and no efficient quantum algorithms are known for solving lattice problems in the worst-case. Hence, the lattice assumption provides a choice on which to design a quantum-secure SDVS scheme.

Recently, lattice-based cryptography has been a hot research topic and many cryptographic lattice-based primitives have been proposed [8-14]. Gentry et al. show that lattice problems are sufficient to construct a kind of trapdoor primitive called a pre-image sample-able function (PSF), which is a basic tool to construct lattice-based signatures and lattice-based IBE (Identity-based encryption) [8,11,14]. A new technique as a development of the PSF, called bonsai trees or basis delegation, is proposed in 2010 [11], which, at a high level, allows one to use a short basis of a given lattice to derive a short basis of related lattice in a secure way. Although many breakthroughs are achieved in the lattice-based cryptography, there are still many open problems that need to be studied. Among these problems that we may ask, can we design a lattice-based cryptosystem with some "added" function just like we have done in the classic cryptography? For example, lattice-based group signature scheme [15], lattice-based blind signature scheme [16] and lattice-based threshold ring signature scheme [17]. As far as we understand, there is still no a secure lattice-based SDVS scheme that has been proposed.

We use PSF and Bonsai trees to build a SDVS scheme in this paper. More precisely, PSF primitives and Bonsai tree techniques are used to generate the signature of the message. Subsequently, this signature is encrypted by a LWE-based trapdoor one-way function [8] to achieve the robustness in our construction. Moreover, in order to improve the efficiency of our SDVS scheme we use hybrid encryption techniques to finish the encryption. If the SIS problem is hard, the proposed scheme is un-forgeable which has been proven in the random oracle model. We also show that the non-transferability of the proposed scheme is based on the hardness of the LWE problem. Finally, the proposed SDVS scheme is extended into the ring signature and a strong designate verifier ring signature (SDVRS) scheme is proposed which satisfies the unconditional anonymity, robustness, un-forge-ability properties. Note that the extension from the proposed SDVS to the SDVRS is natural, because a DVS scheme can be seen as a ring signature scheme with only two ring members.

The rest of this paper is organized as follows. We first introduce some basic notions about lattices, and formalize the security model of the SDVS and the SDVRS in Section 2.0. In Section 3.0, we detail our lattice-based strong designate verifier signature scheme and its security proof. We present the designate verifier ring signature and analyze its security in Section 4.0. Finally, we give a summary in Section 5.0.

## 2.0 PRELIMINARIES

### 2.1 Notations

Throughout the paper, we use bold lower-case letters to denote vectors in column form, and bold upper-case letters to denote matrices. When we write a matrix as $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2)$ (or vector $\mathbf{v}^t = (\mathbf{v}_1^t, \mathbf{v}_2^t)$), it means that the matrix **A** is a concatenation of two matrices $\mathbf{A}_1$ and $\mathbf{A}_2$. When a function is written as $\omega$ ($f$ ($n$)), it means that the function $\omega$ ($f$ ($n$)) grows faster than $cf(n)$ for every constant $c$>0. Let $poly(n)$ denote an unspecified function $f(n) = O(n^c)$ for some constant $c$. For $\alpha \in R^+$, $\Phi_\alpha$ is defined to be the distribution on T of a normal variable with mean 0 and standard deviation α/2π, reduce modulo 1, where T is an additive group on interval [0,1) with modulo 1 addition. $\Phi_\alpha^m$ denotes the natural extension of the distribution $\Phi_\alpha$ to an $m$-dimension space. In this paper, for a vector, we always consider its Euclidean norm which is written as $\|\cdot\|$. By convention, we say, the norm of a matrix is the norm of its longest column. For any matrix **T**, $\tilde{\mathbf{T}}$ denotes the Gram-Schmidt orthogonalized matrix. $D_{(\Lambda,\ s,\ c)}$ denotes the Gaussian distribution with centre **c** and parameter $s$ over the lattice.

### 2.2 Lattice

For a set of $n$ linearly independent vectors, $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n\}$, a lattice Λ generated by **B** is defined as follows, $\Lambda = \{\mathbf{Bc} \mid \mathbf{Bc} = c_1\mathbf{b}_1 + \cdots + c_n\mathbf{b}_n, c_i \in Z\}$.

In this case **B** is referred to as the basis of the lattice Λ. We call a basis is a trapdoor basis if the vectors from this basis are smallest vectors of the lattice. In fact, if the norms of vectors from a basis are small enough, they can still be recognized as a trapdoor basis. In cryptographic applications, any trapdoor basis is kept secret by its holder. In this paper, we will restrict our attention to a special class of $q$-ary lattices which are more easily described by a matrix that functions like a parity check matrix from coding theory. More precisely, for some integers ($q$, $m$, $n$), given a matrix $\mathbf{A} \in Z_q^{n \times m}$, define the following $m$-dimensional lattice

$$\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{e} \in Z_q^m, \mathbf{Ae} = 0(\mathrm{mod}\, q)\}$$

i.e. the lattice that contains all vectors that are orthogonal modulo $q$ to the rows of the matrix **A**.

We recall some results on lattice-based cryptography as the following Lemmas.

**Lemma 1**. (PSF)[8]. There is a probabilistic polynomial-time (PPT) algorithm that, given a trapdoor basis **B** of an $n$-dimensional lattice Λ, a Gaussian parameter $s > \|\tilde{\mathbf{B}}\| \omega(\sqrt{\log n})$, and a center $\mathbf{c} \in R^n$, outputs a sample **e** from a distribution that is statistically close to $D_{(\Lambda,s,c)}$. Moreover, $\|\mathbf{e}\| \leq s\sqrt{m}$ holds with highly probability.

We use PSF and Bonsai trees to build a SDVS scheme in this paper. More precisely, PSF primitives and Bonsai tree techniques are used to generate the signature of the message. Subsequently, this signature is encrypted by a LWE-based trapdoor one-way function [8] to achieve the robustness in our construction. Moreover, in order to improve the efficiency of our SDVS scheme we use hybrid encryption techniques to finish the encryption. If the SIS problem is hard, the proposed scheme is un-forgeable which has been proven in the random oracle model. We also show that the non-transferability of the proposed scheme is based on the hardness of the LWE problem. Finally, the proposed SDVS scheme is extended into the ring signature and a strong designate verifier ring signature (SDVRS) scheme is proposed which satisfies the unconditional anonymity, robustness, un-forge-ability properties. Note that the extension from the proposed SDVS to the SDVRS is natural, because a DVS scheme can be seen as a ring signature scheme with only two ring members.

The rest of this paper is organized as follows. We first introduce some basic notions about lattices, and formalize the security model of the SDVS and the SDVRS in Section 2.0. In Section 3.0, we detail our lattice-based strong designate verifier signature scheme and its security proof. We present the designate verifier ring signature and analyze its security in Section 4.0. Finally, we give a summary in Section 5.0.

## 3.0 PRELIMINARIES

### 2.1 Notations

Throughout the paper, we use bold lower-case letters to denote vectors in column form, and bold upper-case letters to denote matrices. When we write a matrix as $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2)$ (or vector $\mathbf{v}^t = (\mathbf{v}_1^t, \mathbf{v}_2^t)$), it means that the matrix **A** is a concatenation of two matrices $\mathbf{A}_1$ and $\mathbf{A}_2$. When a function is written as $\omega$ ($f$ ($n$)), it means that the function $\omega$ ($f$ ($n$)) grows faster than $cf(n)$ for every constant $c>0$. Let $poly(n)$ denote an unspecified function $f(n) = O(n^c)$ for some constant $c$. For $\alpha \in R^+$, $\Phi_\alpha$ is defined to be the distribution on T of a normal variable with mean 0 and standard deviation α/2π, reduce modulo 1, where T is an additive group on interval [0,1) with modulo 1 addition. $\Phi_\alpha^m$ denotes the natural extension of the distribution $\Phi_\alpha$ to an $m$-dimension space. In this paper, for a vector, we always consider its Euclidean norm which is written as $\|\cdot\|$. By convention, we say, the norm of a matrix is the norm of its longest column. For any matrix **T**, $\tilde{\mathbf{T}}$ denotes the Gram-Schmidt orthogonalized matrix. $D_{(\Lambda,\ s,\ c)}$ denotes the Gaussian distribution with centre **c** and parameter $s$ over the lattice.

### 2.2 Lattice

For a set of $n$ linearly independent vectors, $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n\}$, a lattice Λ generated by **B** is defined as follows, $\Lambda = \{\mathbf{Bc} \mid \mathbf{Bc} = c_1\mathbf{b}_1 + \cdots + c_n\mathbf{b}_n, c_i \in Z\}$.

In this case **B** is referred to as the basis of the lattice Λ. We call a basis is a trapdoor basis if the vectors from this basis are smallest vectors of the lattice. In fact, if the norms of vectors from a basis are small enough, they can still be recognized as a trapdoor basis. In cryptographic applications, any trapdoor basis is kept secret by its holder. In this paper, we will restrict our attention to a special class of $q$-ary lattices which are more easily described by a matrix that functions like a parity check matrix from coding theory. More precisely, for some integers ($q$, $m$, $n$), given a matrix $\mathbf{A} \in Z_q^{n \times m}$, define the following $m$-dimensional lattice

$$\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{e} \in Z_q^m, \mathbf{Ae} = 0(\bmod q)\}$$

i.e. the lattice that contains all vectors that are orthogonal modulo $q$ to the rows of the matrix **A**.

We recall some results on lattice-based cryptography as the following Lemmas.

**Lemma 1**. (PSF)[8]. There is a probabilistic polynomial-time (PPT) algorithm that, given a trapdoor basis **B** of an *n*-dimensional lattice Λ, a Gaussian parameter $s > \|\tilde{\mathbf{B}}\| \omega(\sqrt{\log n})$, and a center $\mathbf{c} \in R^n$, outputs a sample **e** from a distribution that is statistically close to $D_{(\Lambda,s,c)}$. Moreover, $\|\mathbf{e}\| \leq s\sqrt{m}$ holds with highly probability.

**Lemma 2**. (The Trapdoor Sampling Algorithm)[12]. For any prime $q = poly(n)$ and $m \geq 5n\log q$, there is a PPT algorithm that, on input $1^n$ output a matrix $\mathbf{A} \in Z_q^{n\times m}$, and a full-rank set $\mathbf{S} \subset \Lambda_q^{\perp}(\mathbf{A})$, where the distribution of A is statistically close to the uniform distribution, and $\|\mathbf{S}\| \leq O(n\log q)$. In particular, the set **S** can be efficiently converted to a trapdoor basis **T** of the lattice $\Lambda_q^{\perp}(\mathbf{A})$.

**Lemma 3**. (Bonsai Trees)[11] Let $\mathbf{B}_S$ be a trapdoor basis of the lattice $\Lambda_q^{\perp}(\mathbf{A}_S)$ for $\mathbf{A}_S \in Z_q^{n\times m}$ whose columns generate the entire group $Z_q^n$. Let $\mathbf{A}' \in Z_q^{n\times m'}$ be a arbitrary matrix and $s > \|\tilde{\mathbf{B}}_S\| \omega(\sqrt{\log n})$ be a Gaussian parameter. There is a PPT algorithm with ( $\mathbf{B}_S, \mathbf{A} = (\mathbf{A}_S, \mathbf{A}'), s$ ) as input, that outputs a trapdoor basis of the lattice $\Lambda_q^{\perp}(\mathbf{A})$.

### 2.3 Lattice Problems

We introduce two lattice problems used in this paper and the first lattice problem is the short integer solution (SIS) problem which may be seen as the average-case problem related to the family of the lattices $\Lambda_q^{\perp}(\mathbf{A})$. It is shown in [8] that the SIS problems in the average-case are as hard as approximating the SIVP problem (shortest independent vectors problem) in the worst-case.

**Definition 1**. Given an uniform and random matrix $\mathbf{A} \in Z_q^{n\times m}$ and (*n*, *m*, *q*, *β*) are parameters, the goal of the SIS problem is to find a nonzero integer vector $\mathbf{v} \in Z_q^m$ such that $\|\mathbf{v}\| \leq \beta$ and **Av** = 0(mod *q*).

The other lattice problem which we will use in the present paper is the learning with errors (LWE) problem. For parameters (*n*, *m*, *q*), $\mathbf{s} \in Z_q^n$ and an error distribution χ over $Z_q^m$, $A_{(s,\chi)}$ is a distribution over $Z_q^{n\times m} \times Z_q^m$ according to $\{\mathbf{A}, \mathbf{A}^t\mathbf{s} + \mathbf{x}(\bmod q)\}$ where $\mathbf{A} \in Z_q^{n\times m}$ is chosen randomly and the errors vector **x** is a sample from the distribution χ. Then the LWE problem is defined as follows.

**Definition 2**. Given a sample from the distribution $A_{(s,\chi)}$, the goal of the search variant of the LWE problem is to outputs **s** with a noticeable probability. The decision variant of the LWE problem is to distinguish $A_{(s,\ \chi)}$ from the uniform distribution over $Z_q^{n\times m} \times Z_q^m$.

The standard setting for the LWE problem considers the error distribution $\bar{\Phi}_\alpha^m$ which is a Gaussian distribution over $Z_q^m$. We can sample the errors vector according to the distribution $\bar{\Phi}_\alpha^m$ as follows: Sample *m* numbers $(\eta_1, \cdots \eta_m)$ according to a Gaussian distribution $\Phi_\alpha$, and compute $e_i$= $\lceil q\eta_i \rfloor$ (mod *q*) as the closest integer to $q\eta_i$. Then let $\mathbf{e} = (e_1, e_2, \cdots, e_m)$ be an error vector in the LWE problem.

### 2.4 The Strong Designate Verifier Signature and Its Security

A standard SDVS scheme for two participants, the signer Alice and the designate verifier Bob is $SIG_{sdvs}$ = (Kg, Sign, Vrf, Sim) which consists of four polynomial-time algorithms:

**1**. **Kg**. A probabilistic algorithm, which takes on input a security parameter $1^n$ and outputs $(pk_A, sk_A, pk_B, sk_B)$ where $(pk_A, sk_A)$ is Alice's public and private key and $(pk_B, sk_B)$ is Bob's public key and private key.

**2**. **Sign** ($M, sk_A, pk_B$). A probabilistic algorithm that takes on input ($M, sk_A, pk_B$), and outputs a signature *S*, denoted as *S* = Sign ($M, sk_A, pk_B$).

**3**. **Vrf** ($M, S, sk_B, pk_A$). This is a deterministic algorithm, which takes on input a message *M*, signature *S*, Bob's private key and Alice's public key. It outputs a bit *b* = 1 if *S* is a valid signature of *M*, otherwise it outputs *b* = 0.

**4**. **Sim** ($M, pk_A, sk_B, pk_B$). A probabilistic algorithm generates signatures which are indistinguishable from those produced by Sign ($M, sk_A, pk_B$).

An SDVS scheme should satisfy the following security properties.
**1**. **Correctness**. A properly formed SDVS must be accepted by the Vrf algorithm.

**2**. **Un-forge-ability**. The SDVS scheme must be un-forgeable under the chosen message attack if the advantage of any polynomial time adversary in the following game is negligible.

**Setup**: The challenger runs the Kg algorithm to generate systems parameters and the public/private key for the signer and the designate verifier. The challenger sends public keys to the adversary and keeps the private key secret.

**Sign queries**: The adversary queries signatures on some messages $M_i$ for the designate verifier Bob. The challenger outputs the SDVS signature for every message $M_i$ as a response.

**Verify queries**: The adversary can request the Vrf algorithm on an SDVS for some signer and the designated verifier. As response, the challenger outputs "True" if the SDVS is correct, otherwise it outputs False.

**Output**: Finally, the adversary outputs a new signature with the signer and the designate verifier.
If the output of the adversary can be accepted by the Vrf algorithm, the adversary wins the game. The advantage of the adversary is defined by the probability of the adversary to win the game.

**3. Non-transferability**.
The non-transferability property is ensured by a transcript simulation algorithm that can be performed by the designated verifiers to produce an indistinguishable signature from the one that should be produced by the signature holder. Formally, it is defined by the following game played between a challenger and a PPT adversary.

**Setup.** The challenger generates the public keys and the private keys of the signer and the designate verifier by the Kg algorithm and sends public keys to the adversary.

**Sign and verify queries**. The adversary queries adaptively for the sign query and the verify query.

**Challenge**. The adversary sends a new message M to the challenger. The challenger randomly chooses a fair coin $b \in \{0,1\}$. If *b* = 0, showing that the signature is invalid it then sends the actual SDVS to the adversary, if *b*=1, showing that the signature is valid it then, sends the simulated SDVS which is generated by the designate verifier as the SDVS.

**Outputs**. The adversary outputs a guess bit *b*', if $b = b'$, the adversary wins the game.
An SDVS scheme is non-transferable if $|\, pr(b = b') - 1/2 \,|$ is negligible for any PPT adversary.

**4. Robustness**
The **robustness** of the SDVS is satisfied when anyone without the knowledge of the designate verifier's private key cannot verify the SDVS and whether it is generated by some signer or otherwise.

In the SDVRS scheme, the Kg algorithm should output the public keys and the signing keys for all ring users and the designate verifier. A secure SDVRS scheme should satisfy not only the non-transferability, **robust**ness and un-forge-ability properties but also should ensure the anonymity of the signer. Namely, neither third party nor the designate verifier can find the identity of the signer from the SDVRS.

## 4.0 LATTICE-BASED STRONG DESIGNATE VERIFIER SIGNATURE

### 3.1 Lattice-based Strong Designate Verifier Signature

Let *n* be a prime number, and $m \geq 2n\log q, q > \beta\omega(\log n), \beta = poly(n)$. A bound $\tilde{L} \geq O(\sqrt{n\log q})$ and a Gaussian parameter $s = \tilde{L}\omega(\sqrt{\log n})$. There are two collision-resistant secure hash functions which $H_1$ maps $(0,1)^* \times Z_q^n$ to $Z_q^n$ and $H_2$ maps $(0,1)^* \times Z_q^n$ to $Z_q^{2m}$. If the signer is Alice and the designate verifier is Bob, the lattice-based SDVS scheme is proposed as follows:

**Kg**. Both Alice and Bob run the trapdoor sampling algorithm in Lemma 2 to generate $\mathbf{A} \in Z_q^{n\times m}$ and $\mathbf{B} \in Z_q^{n\times m}$ as their public key and the trapdoor basis $\mathbf{T}_A$ and $\mathbf{T}_B$ as their private key, respectively.

**Sign**. To sign a message $M \in (0,1)^*$, Alice does as follows:
1. Randomly chooses a vector $\mathbf{t} \in Z_q^n$, computes $H_1(M,\mathbf{t})$. Furthermore, randomly chooses a new vector $\mathbf{e}_2 \in Z_q^m$, which satisfies $\|\mathbf{e}_2\| \leq s\sqrt{m}$;
2. By using PSF, computes a vector $\mathbf{e}_1 \in Z_q^m$ satisfying $\mathbf{Ae}_1 = (H_1(M,\mathbf{t}) - \mathbf{Be}_2)(\bmod q)$. Parses $\mathbf{e} = (\mathbf{e}_1, \mathbf{e}_2)$, then **e** is the DVS signature (Indeed, the computation process in this step is just the core operation of Bonsai trees primitive).
3. Randomly chooses a new vector $\mathbf{r}' \in Z_q^n$, computes $H_2$ $(M, \mathbf{r}') \in Z_q^{2m}$.
4. Chooses an error vector $\mathbf{x} \in Z_q^m$ according to the error distribution $\bar{\Phi}_\alpha^m$. Computes:

$$\sigma = \mathbf{e} + H_2(M,\mathbf{r}')(\bmod q),\ \mathbf{r} = \mathbf{B}^t\mathbf{r}' + \mathbf{x}(\bmod q).$$

Then the SDVS of the message *M* is (*σ*, **r**, **t**).

**Vrf**. Bob does as follows to verify the SDVS of message *M*:
1. Computes $\mathbf{T}_B^t\mathbf{r} = \mathbf{T}_B^t\mathbf{x}(\bmod q)$. Since $\mathbf{T}_B$ *is* a trapdoor basis whose entries are all sufficiently small and **x** is an error vector whose entries are also small enough, it is surely that $\mathbf{T}_B^t\mathbf{r} = \mathbf{T}_B^t\mathbf{x}(\bmod q) = \mathbf{T}_B^t\mathbf{x}$ (over integer) with an overwhelm probability [13]. So Bob computes $\mathbf{x} = \mathbf{T}_B^{-t}\mathbf{T}_B^t\mathbf{x}$. And then, Bob obtains **r**' from **r** and **x**.
2. Computes $H_2(M,\mathbf{r}')$ and $\mathbf{e} = \sigma + H_2(M,\mathbf{r}')$.
3. Accepts the SDVS if and only if $\|\mathbf{e}\| \leq s\sqrt{2m}$ and $(\mathbf{A},\mathbf{B})\mathbf{e} = H_1(M,\mathbf{t})(\bmod q)$, otherwise, rejects it.

**Sim**. By Lemma 3 and with the help of his private key $\mathbf{T}_B$, Bob can also generate a vector **e** which satisfies $\|\mathbf{e}\| \le s\sqrt{2m}$ and $(\mathbf{A},\mathbf{B})\mathbf{e} = H_1(M,\mathbf{t})(\bmod q)$ (mod *q*) where **t** is a random vector. Then, Bob encrypts the vector **e** to get the SDVS (*σ*, **r**, **t**) as described in the Sign algorithm. By PSF and Bonsai trees, the simulated signature **e** is also distributed close to the Gaussian distribution just like the actual signature. Since **r** is a LWE instance and $H_2$ is a secure hash function, (*σ*, **r**, **t**) in the simulated SDVS is random and uniform, just like the actual signature. Hence, the simulated signature by Bob is indistinguishable from the actual signature by Alice.

**Correctness** It is easy to find that the parameters in this scheme satisfy three lemmas in Section 2.2, and then Alice can finish the sign algorithm as we have shown. On the other hand, Bob can use his private key to obtain **r**' from the LWE instance $(\mathbf{B}, \mathbf{r} = \mathbf{B}^t\mathbf{r}' + \mathbf{x}(\bmod q))$ which have been shown in Section 2.3. So, Bob can get the vector **e** in step 2 of the Vrf algorithm. Through the sign algorithm, we know that $\|\mathbf{e}\| \le s\sqrt{2m}$ and $(\mathbf{A},\mathbf{B})\mathbf{e} = H_1(M,\mathbf{t})(\bmod q)$ hold. Hence, a valid SDVS can be accepted by the Vrf algorithm in our scheme.

### 3.2 Security

**1. Non-transferability**.

From the definition of non-transferability we know that if the SDVS can be simulated by the designate verifier then the SDVS scheme satisfies the non-transferability property. As shown by the Sim algorithm, Bob can generate a simulated DVS whose distribution is close to Gaussian distribution just like the actual DVS. By the hardness of the LWE problem, the distribution of **r** both in the actual SDVS and in the simulated SDVS is close to the uniform distribution. Since $H_2$ is a secure hash function, *σ* both in the actual SDVS and the simulated SDVS can be seen as a random vector. And **t** is chosen randomly and uniformly. As a result, the distributions of the actual SDVS or the simulated SDVS are indistinguishable from the uniform distribution. Hence, the simulated signature by Bob is indistinguishable from the actual signature by Alice.

**2. Robustness**.

By the **robustness** hypothesis of the LWE problem, the distribution of **r** is indistinguishable from the uniform distribution. Any third party without the knowledge of Bob's private key can not get **r**' from **r**, otherwise, he can solve the LWE problem. So no one except Bob can verify the SDVS. The **robustness** is preserved.

**3. Un-forge-ability**

**Theorem 1**. This SDVS scheme is un-forgeable under the hardness of the SIS problem.

**Proof**: To derive a contradiction, we assume that there exists a PPT adversary A gaining an advantage *ε* for forging an SDVS, by accessing the random oracle $H_1$ $q_1$ times, the random oracle $H_2$ $q_2$ times and the signing oracle $q_3$ times, and furthermore, querying the Vrf algorithm $q_4$ times. Then we construct a challenger C to solve the SIS problem with probability close to $\varepsilon$. Suppose that C receives an SIS instance $(\mathbf{A} \in Z_q^{n\times 2m}, q, n, m, s)$ and hopes to get a vector **v** satisfying $\|\mathbf{v}\| \le 2s\sqrt{2m}$ and **Av** = 0 (mod *q*). Let **A** = ($\mathbf{A}_1$, $\mathbf{A}_2$) for $\mathbf{A}_1, \mathbf{A}_2 \in Z_q^{n\times m}$. Let $\mathbf{A}_1$ be Alice's public key and $\mathbf{A}_2$ Bob's public key. Three lists $L_i$, *i* = (1,2,3) is used to store the answers to the random oracles $H_1$, $H_2$ and the signature oracle, respectively.

***H*₁ query**. In order to generate the $H_1$ hash value of the message $M_i$ for $i < q_1$, C checks $L_1$ to make sure that $M_i$ is fresh. If an entry in $L_1$ is found, it returns the same answer $h_{1i}$ to A. Otherwise, C randomly chooses a vector $\mathbf{v}_i$ satisfying $\|\mathbf{v}_i\| \le s\sqrt{2m}$ and computes $\mathbf{A}\mathbf{v}_i = h_{1i}(\bmod q)$. It then randomly chooses a vector $\mathbf{t} \in Z_q^n$, and returns $h_{1i}$ as an answer. C saves $(M_i, \mathbf{v}_i, h_{1i}, \mathbf{t}_i)$ to the list $L_1$.

***H*₂ query**. In order to generate the $H_2$ hash value of the message $M_i$ and a random vector $\mathbf{r}_i' \in Z_q^n$ for $i < q_2$, C checks L$_2$ to make sure that $(M_i, \mathbf{r}_i')$ are fresh. If an entry in L$_2$ is found, it returns the same answer $h_{2i}$ to A. Otherwise, C returns a random vector $h_{2i} \in Z_q^{2m}$ which is never used in this phase as the answer and stores $(M_i, \mathbf{r}_i', h_{2i})$ to the list L$_2$.

**Sign query**. To answer the signing queries of $M_i$, C finds $(\mathbf{v}_i, h_{1i}, \mathbf{t}_i)$ from the list L$_1$, gets some $(M_i, \mathbf{r}_i', h_{2i})$ from the list L$_2$. It then, computes $\sigma_i = (\mathbf{v}_i + h_{2i})(\bmod q)$. The challenger also chooses an error vector $\mathbf{x}_i$ from the error distribution $\bar{\Phi}_\alpha^m$, and computes $\mathbf{r}_i = \mathbf{A}_2^t \mathbf{r}_i' + \mathbf{x}_i (\bmod q)$. It stores $(M_i, \sigma_i, \mathbf{r}_i)$ to L$_3$, then $(M_i, \sigma_i, \mathbf{r}_i, \mathbf{t}_i)$ is sent to the adversary as the SDVS of the message $M_i$.

**Verify query.** If the adversary asks the challenger to verify an SDVS $(M_i, \sigma_i, \mathbf{r}_i, \mathbf{t}_i)$, C checks the lists L$_1$ and L$_2$ to get the hashed value $h_{1i}$ and $h_{2i}$ respectively. It computes $\mathbf{v}_i = \sigma_i + h_{2i}(\bmod q)$. Then C can verify the SDVS as shown as in the Vrf algorithm.

After all the queries have been issued, A forges an SDVS $(M_{i^*}, \sigma_{i^*}, \mathbf{r}_{i^*}, \mathbf{t}_{i^*})$ which can be accepted by the Vrf algorithm with the probability *ε*. Then the challenger can solve the SIS problem as follows:

First, C obtains $h_{1i^*}$ and $h_{2i^*}$ from lists L$_1$, L$_2$, respectively. From the equation $h_{2i^*} + \sigma_{i^*} = \mathbf{e}_{i^*}(\bmod q)$, C gets the vector $\mathbf{e}_{i^*}$ as a DVS of message satisfying $\| \mathbf{e}_{i^*} \| \le s\sqrt{2m}$ and $\mathbf{A}\mathbf{e}_{i^*} = h_{1i^*}(\bmod q)$;

Next, C checks the list L$_1$ to get $\mathbf{v}_{i^*}$ satisfying $\| \mathbf{v}_{i^*} \| \le s\sqrt{2m}$ and $\mathbf{A}\mathbf{v}_{i^*} = h_{1i^*}(\bmod q)$;

Finally, if $\mathbf{e}_{i^*} \ne \mathbf{v}_{i^*}$, we know that $\mathbf{A}\mathbf{v}_{i^*} = \mathbf{A}\mathbf{e}_{i^*}(\bmod q)$ holds, that is $\mathbf{A}(\mathbf{v}_{i^*} - \mathbf{e}_{i^*}) = 0(\bmod q)$. Since $\| \mathbf{e}_{i^*} \| \le s\sqrt{2m}$ and $\| \mathbf{v}_{i^*} \| \le s\sqrt{2m}$ hold, then $\| \mathbf{v}_{i^*} - \mathbf{e}_{i^*} \| \le 2s\sqrt{2m}$ holds. As a result, C gets a solution of the SIS problem. $\mathbf{e}_{i^*} = \mathbf{v}_{i^*}$ holds, C aborts.

Now we analyze the advantage of the challenger. Since both $\mathbf{e}_{i^*}$ and $\mathbf{v}_{i^*}$ are the pre-images of the hash value $h_{1i^*}$ under the trapdoor function $f_A$ (**s**) = **As** (mod *q*), from the literature [8], we know that the numbers of the pre-images of $h_{1i^*}$ is at least $2^{\omega(\log n)}$. Hence, we conclude that $\mathbf{e}_{i^*} \ne \mathbf{v}_{i^*}$ holds with the probability of at least $1\text{-}2^{-\omega(\log n)}$. As a result, the challenger can solve the SIS problem with a probability of at least $(1\text{-}2^{-\omega(\log n)})\varepsilon$.

## 4.0 LATTICE-BASED STRONG DESIGNATE VERIFIER RING SIGNATURE

### 4. 1 Lattice-based Strong Designate Verifier Ring Signature

Based on the bonsai tree primitive [11], we use the proposed SDVS scheme to construct a SDVRS scheme which is described as follows.

**Kg**. The parameters (*n*, *m*, *q*, $\tilde{L}$, *s*, *β*) are the same as the proposed SDVS scheme. There is a collision-resistant secure hash functions $H_1$ which maps $(0,1)^* \times Z_q^n$ to $Z_q^n$. We need a pseudorandom generator *prg*: $Z_q^n \to Z_q^{(l+1)n}$. The ring users of the ring group are denoted by $U_i$ where $i = 1, 2 \cdots, l$. The designate verifier is Bob and Bob generates his public/private keys by the trapdoor sampling algorithm which has been introduced in Lemma 2. Let $\mathbf{B} \in Z_q^{n\times m}$ be Bob's public key and $\mathbf{T}_B \in Z_q^{m\times m}$ the private key. Every ring member $U_i$ also generates his public/private keys by the trapdoor sampling algorithm. Let $\mathbf{A}_i \in Z_q^{n\times m}$ denote $U_i$'s public key and $\mathbf{T}_i \in Z_q^{m\times m}$ denote his private key. A third party or some ring user parses $\mathbf{A}$ = ($\mathbf{A}_1$, $\mathbf{A}_2$, … , $\mathbf{A}_l$). Then ($n$, $m$, $q$, $\tilde{L}$, $s$, $\beta$, $\mathbf{A}$, $H_1$, *prg*) are public parameters.

**Sign. T**o generate a SDVRS on message $M \in (0,1)^*$ , the user $U_i$ performs the following procedures.

1. Randomly chooses a vector $\mathbf{t} \in Z_q^n$ and computes $H_1$ ($M$, $\mathbf{t}$). Subsequently, randomly chooses $l$ vectors $\mathbf{e}_b$ and $\mathbf{e}_j$, $j \neq i$ , $j \le l$ satisfying $\| \mathbf{e}_b \|, \| \mathbf{e}_j \| \le s\sqrt{m}$ .

2. By PSF and Bonsai trees, and with the help of his private key $\mathbf{T}_i$, $U_i$ generates a vector $\mathbf{e}_i \in Z_q^m$ satisfying $\| \mathbf{e}_i \| \le s\sqrt{m}$ and $\mathbf{A}_i\mathbf{e}_i$= $\mathbf{y}$ (mod $q$) where $\mathbf{y}$ = $(H_1(M,\mathbf{t}) - \mathbf{B}\mathbf{e}_b - \sum_{j,j\neq i}^{l} \mathbf{A}_j\mathbf{e}_j)(\bmod q)$ ). Parses $\mathbf{e}^t = (\mathbf{e}_1^t, \mathbf{e}_2^t, \cdots, \mathbf{e}_l^t, \mathbf{e}_b^t)$ .

3. Randomly chooses a new vector vector $\mathbf{r}' \in Z_q^n$ , computes *prg* ($\mathbf{r}'$) $\in Z_q^{(l+1)m}$ .

4. Chooses an error vector $\mathbf{x}$ according to the error distribution $\bar{\Phi}_\alpha^m$ . Computes $\sigma = (\mathbf{e} + prg(\mathbf{r}')(\bmod q)$ and $\mathbf{r} = \mathbf{B}^t\mathbf{r}' + \mathbf{x}(\bmod q)$ .

Then the SDVRS of the message $M$ is ($\sigma$, $\mathbf{r}$, $\mathbf{t}$).

**Vrf.** Bob can verify an SDVRS as follows:

1. Decrypts $\mathbf{r}$ into vectors $\mathbf{r}'$ by his private key whose operations have been described in the Vrf algorithm of the proposed SDVS scheme. Then, Bob computes *prg* ($\mathbf{r}'$)$\in Z_q^{(l+1)m}$ and $H_1(M,\mathbf{t})$ .

2. Computes $\mathbf{e} = (\sigma + prg(\mathbf{r}')(\bmod q)$ .

3. Accepts the SDVRS ($\sigma$, $\mathbf{r}$, $\mathbf{t}$) if $\| \mathbf{e} \| \le s\sqrt{(l+1)m}$ and $(\mathbf{A},\mathbf{B})\mathbf{e} = H_1(M,\mathbf{t})(\bmod q)$ hold, otherwise, rejects it.

**Sim**. For a message $M$, Bob uses his private key $\mathbf{T}_B$ to generate a simulated signature as follows:

1. Randomly chooses a vector $\mathbf{t} \in Z_q^n$ and computes $H_1(M,\mathbf{t})$ . Next, randomly chooses $l$ vectors $\mathbf{e}_j \in Z_q^m$ satisfying $\| \mathbf{e}_j \| \le s\sqrt{m}$ where $j \le l$ .

2. By Lemma 1 and Lemma 3, using his private key $\mathbf{T}_B$, Bob finds a vector $\mathbf{e}_b \in Z_q^m$ such that $\| \mathbf{e}_b \| \le s\sqrt{m}$ and $\mathbf{B}\mathbf{e}_b$= $\mathbf{y}$ (mod $q$) where $\mathbf{y}$ = (( $H_1(M,\mathbf{t})$ - $\sum_{j=1}^{l} \mathbf{A}_j\mathbf{e}_j$ )(mod $q$).
Parses $\mathbf{e}^t = (\mathbf{e}_1^t, \mathbf{e}_2^t, \cdots, \mathbf{e}_l^t, \mathbf{e}_b^t)$ then ($\mathbf{A}$, $\mathbf{B}$) $\mathbf{e}$ = $H_1$ ($M$, $\mathbf{t}$)(mod $q$) and $\| \mathbf{e} \| \le s\sqrt{(l+1)m}$ hold.

3. Randomly chooses a new vector $\mathbf{r}' \in Z_q^n$ and computes *prg*($\mathbf{r}'$).

4. Chooses an error vector $\mathbf{x}$ from the error distribution $\bar{\Phi}_\alpha^m$ and computes $\sigma = (\mathbf{e} + prg(\mathbf{r}'))(\bmod q)$ and $\mathbf{r} = \mathbf{B}^t\mathbf{r}' + \mathbf{x}(\bmod q)$ .

Then ($\sigma$, $\mathbf{r}$, $\mathbf{t}$) is a simulated SDVRS.

By Lemma 1 and Lemma 3, and with the help of his private key $\mathbf{T}_B$, Bob can also generate a vector $\mathbf{e}_b$ which satisfies $\| \mathbf{e}_b \| \le s\sqrt{m}$ and $(\mathbf{A},\mathbf{B})\mathbf{e} = H_1(M,\mathbf{t})(\bmod q)$ where $\mathbf{t}$ is a random vector. Then, the simulated signature by

Bob can be accepted by the Vrf algorithm. Furthermore, since the distribution of the simulated SDVRS is also close to the uniform distribution, just like the actual SDVRS, the simulated SDVRS by Bob is indistinguishable from the actual SDVRS by Alice. The correctness of the proposed SDVRS scheme can be proven as same as the correctness of the proposed SDVS scheme in Section 3.

### 4. 2 Security

#### 1. Anonymity

**Theorem 2**. The proposed SDVRS scheme is unconditionally anonymous.

**Proof:** For any third party, since **r** is an LWE problem instance, vector **r** gives no information about the identity of the signer. On the other hand, since *pgr* (**r**') is the outputs of a pseudorandom generator, then $\sigma$ can be seen as a random vector and it can not leak any information. As a result, any third party can not find the identity of the signer from SDVRS. Of course, Bob can decrypt the vector $\sigma$ to get the vector $\mathbf{e} \in Z_q^{(l+1)m}$, while some vector $\mathbf{e}_j$ of **e** is the output of PSF whose distribution is close to the Gaussian distribution and other-vectors are chosen randomly. Hence, **e** can not leak any identity information to Bob. As a result, neither any third party nor the designate verifier can find the identity of the signer.

#### 2. Non-transferability and Robustness

As shown by the Sim algorithm, the transcripts simulated by Bob are indistinguishable from the actual signature. Thus, the non-transferability property holds. The r**obustness** of the proposed SDVRS scheme is based on the r**obustness** of the proposed SDVS scheme which has been proven in Section 3.2.

#### 3. Un-forge-ability

**Theorem 3.** If there is an adversary who is not privy to the private keys of any ring member or the designate verifier, can generate a SDVRS with the probability $\varepsilon$, there is a challenger that can solve the SIS problem with probability approaching $\varepsilon$.

**Proof.** Suppose there exists a PPT adversary A gaining an advantage $\varepsilon$ for forging a strong designate verifier ring signature, by accessing the random oracle $H_1$ $q_1$ times and the signature oracle $q_2$ times, and furthermore querying the Vrf algorithm $q_3$ times. Then we can construct a challenger C to solve the SIS problem.Suppose that the challenger C receives an SIS instance $(\mathbf{A} \in Z_q^{n\times(l+1)m}, q, n, m, s)$, and hopes to find a vector **v** with $\|\mathbf{v}\| \le 2s\sqrt{(l+1)m}$ and **Av**=0(mod $q$). Let $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2, \cdots \mathbf{A}_{l+1})$, $\mathbf{A}_i \in Z_q^{n\times m}$. C sends $\mathbf{A}_i \in Z_q^{n\times m}$ as the ring member's public key and $\mathbf{A}_{l+1}$ as the designate verifier's public keys to A. Then C begins the query-answer game with the adversary. To finish this game, C keeps two lists L$_i$, $i$= (1, 2) to store the answers to the random oracle $H_1$ and the signature oracle respectively

***H*₁ Query**. When C receives the message $M_i$ where $i \le q_1$, the challenger C checks the list L$_1$, if an entry in L$_1$ is found, then returns the same answer $h_i$ to the adversary; otherwise, the challenger randomly chooses a vector $\mathbf{v}_i$ with $\|\mathbf{v}_i\| \le s\sqrt{(l+1)m}$ and a vector $\mathbf{t}_i \in Z_q^n$, computes $\mathbf{Av}_i = h_i(\bmod q)$, returns $h_i$ as the answer. C saves $(M_i, \mathbf{v}_i, h_i, \mathbf{t}_i)$ to the list L$_1$.

**Sign Query**. When the challenger needs to generate a SDVRS for a fresh message $M_i$, the challenger gets $(M_i, \mathbf{v}_i, h_{1i}, \mathbf{t}_i)$ from the list L$_1$, randomly chooses a vector $\mathbf{r}_i'$ and computes $prg_i = prg(\mathbf{r}_i')$. And then, C computes $\sigma_i = (\mathbf{v}_i + prg_i)(\bmod q)$. The challenger chooses an error vector **x** from the error distribution $\bar{\Phi}_\alpha^m$, and computes $\mathbf{r}_i = \mathbf{A}_{l+1}^t \mathbf{r}' + \mathbf{x}(\bmod q)$. It stores $(M_i, \sigma_i, \mathbf{r}_i, prg_i)$ to the list L$_2$. Then, $(\sigma_i, \mathbf{r}_i, \mathbf{t}_i)$ is the SDVRS of the message $M_i$. In this operation, if the sign query has been issued, it returns the same answer.

**Verify Query**. If A requires the challenger to verify the SDVRS ($M_i$, $\sigma_i$, ri, $\mathbf{t}_i$), C checks the lists $(L_1, L_2)$ to get the value $(h_i, \mathbf{t}_i)$ and $(\mathbf{r}_i, prg_i)$ respectively. Hence $\mathbf{v}_i = (\sigma_i + prg_i)(\bmod q)$. The remainder works are just like the Vrf algorithm in the proposed scheme.

After all the queries have been issued, the adversary A forges a SDVRS $(M_{i^*}, \sigma_{i^*}, \mathbf{r}_{i^*}, \mathbf{t}_{i^*})$ with the probability $\varepsilon$. Then the challenger can solve the SIS problem as follows.
First, just like what he has done in the verify query, the challenger gets the value $h_{i^*}$ and $prg_{i^*}$. Then C chooses a vector $\mathbf{e}_{i^*}$ satisfying $\| \mathbf{e}_{i^*} \| \le s\sqrt{(l+1)m}$ and $\mathbf{A}\mathbf{e}_{i^*} = h_{i^*} (\bmod q)$.
Next, C gets the vector $\mathbf{v}_{i^*}$ from the list $L_1$. Note that the vector $\mathbf{v}_{i^*}$ is generated by the challenger which satisfies $\| \mathbf{v}_{i^*} \| \le s\sqrt{(l+1)m}$ and $\mathbf{A}\mathbf{v}_{i^*} = h_{i^*} (\bmod q)$ .Hence, C gets two vectors satisfying $\mathbf{A}\mathbf{v}_{i^*} = \mathbf{A}\mathbf{e}_{i^*} (\bmod q)$, namely, $\mathbf{A}(\mathbf{v}_{i^*} - \mathbf{e}_{i^*}) = 0 (\bmod q)$.

Finally, if $\mathbf{e}_{i^*} \ne \mathbf{v}_{i^*}$, since $\| \mathbf{e}_{i^*} \| \le s\sqrt{(l+1)m}$ and $\| \mathbf{v}_{i^*} \| \le s\sqrt{(l+1)m}$, then $\| \mathbf{e}_{i^*} - \mathbf{v}_{i^*} \| \le 2s\sqrt{(l+1)m}$. Hence, the challenger gets a solution of the SIS problem. On the other hand, if $\mathbf{e}_{i^*} = \mathbf{v}_{i^*}$ the challenger aborts.

Now we analyze the advantage of C for solving the SIS problem. C can solve the SIS problem if two vectors are unequal, namely, $\mathbf{e}_{i^*} \ne \mathbf{v}_{i^*}$. Because $\mathbf{e}_{i^*}$ and $\mathbf{v}_{i^*}$ are all the pre-images of the hash value $h_{i^*}$ under the trapdoor function $f_A(\mathbf{s})$= $\mathbf{As}$ (mod $q$). If we bear in mind the conclusion of the literature [8] that the numbers of the pre-images of $h_{i^*}$ is at least $2^{\omega(\log n)}$. Hence, we conclude that $\mathbf{e}_{i^*} \ne \mathbf{v}_{i^*}$ holds with probability at least 1- $2^{-\omega(\log n)}$. As a result, the challenger can solve the SIS problem with probability at least (1- $2^{-\omega(\log n)}$ )$\varepsilon$.

## 5.0 CONCLUSIONS

This paper firstly proposes a lattice-based strong designate verifier signature scheme which is non-transferable. In the random oracle model, we prove that the forge-ability of the SDVS is based on the hardness of the SIS problem. As an application of the proposed SDVS, a lattice-based strong designate verifier ring signature is proposed, which satisfies the signer anonymity and non-transferability. We prove that the forge-ability of the proposed SDVRS is based on the hardness of the SIS problem. In order to try to design a lattice-based SDVS scheme, there are many open problems that need to be studied first, for example, how to build a lattice-based multi-DVS scheme and how to improve the efficiency of the proposed scheme. We leave the discussion of this in the next phase of our research.

## ACKNOWLEDGEMENTS

The authors thank the anonymous referees for their helpful comments. This research was supported by the National Natural Science Foundation of China (Grant Nos. 60970119，61173151，61173152)

## REFERENCES

[1] M. Jakobsson et al.,. ‘‘Designated Verifier Proofs and Their Applications’’. In *Proceedings 15th Annual International Conference on the Theory and Applications of Cryptographic Techniques,* Saragossa, 12-17 May, 1996, LNCS 1070, pp. 143-154.

[2] Y. Li, W. Susilo, Y. Mu, D. Pei. ‘‘Designated Verifier Signature: Definition, Framework and New Constructions’’. In *Proceedings of the 4th International Conference on Ubiquitous Intelligence and Computing,* Hong-kong, 11-13 July, 2007, pp. 1191-1200.

[3] S. Saeednia, S. Kremer, O. Markowitch. ‘‘An Efficient Strong Designated Verifier Signature Scheme’’. In *Proceedings of Information Security and Cryptology,* Seoul, 27-28 December 2003, LNCS 2587, pp.40-54.

[4] P. K. Kancharla, S. Gummadidala, A. Saxena. ‘‘Identity Based Strong Designated Verifier Signature Scheme’’. *Informatica*, Vol 18, No. 5, 2007, pp. 239-252.

[5] T. Raylin, T. Okamoto, E. Okamoto, ‘‘Practical Strong Designated Verifier Signature Schemes Based on Double Discrete Logarithms’’. In *Proceedings of Information Security and Cryptology*, Seoul, 2-3 December, 2005, LNCS 3506, pp. 113-127.

[6] L. Wu, D. X. Li. ‘‘Strong designated Verifier ID-based Ring Signature Scheme’’. In *International Symposium on Information Science and Engineering*, Shanghai, 20-22 December, 2008, pp. 294-298.

[7] P. W. Shor. ‘‘Polynomial-time algorithm for prime factorizeation and discrete logarithm on a quantum computer’’. *SIAM Journal on Computing*, Vol 26, No. 5, 1997, pp. 1484-1509.

[8] C. Gentry, C. Peikert, V. Vaikuntanathan. ‘‘Trapdoors for hard lattices and new cryptographic constructions’’. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, Victoria, 17-20 May, 2008, pp. 197-206.

[9] O. Regev. ‘‘On Lattice, learning with Errors, Random Linear Codes, and Cryptography’’. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, Baltimore, 22-24 May, 2005, pp. 84-93.

[10] C. Peikert. ‘‘Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem’’. In *Proceedings of the 41th Annual ACM Symposium on Theory of Computing*, Bethesda, 31 May-2 June, 2009, pp. 333-342.

[11] D. Cash, D. Hofheinz, E. Kiltz, C. Peikert. ‘‘Bonsai Trees, or How to Delegate a Lattice Basis’’. In *Proceedings 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques* (*Eurocrypt 2010*), Riviera, 30 May-3 June, 2010, LNCS 6110, pp. 523-552.

[12] J. Alwen, C. Peikert. ‘‘Generating shorter bases for hard random lattices’’. In *Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science*. Freiburg, 26-28 February, 2009, pp. 75-86.

[13] C. Gentry, S. Halevi, V. Vaikuntanathan. ‘‘A Simple BGN-type Cryptosystem from LWE’’. In *Proceedings 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques* (*Eurocrypt 2010*), Riviera, 30 May-3 June, 2010, LNCS 6110, pp. 506-522.

[14] S. Agrawal, D. Boneh, X. Boyen. ‘‘Efficient Lattice (H)IBE in the Standard Model’’. In *Proceedings 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques* (*Eurocrypt 2010*), Riviera, 30 May-3 June, 2010, LNCS 6110, pp. 553-572.

[15] S. D. Gordon, J. Katz, V. Vaikuntanathan. ‘‘A Group Signature Scheme from Lattice Assumptions’’. In *Proceedings of The 16th Annual International Conference on the Theory and Application of Cryptology and Information Security*, Singapore, 5-9 December, 2010, LNCS 6477, 2010, pp. 395-412.

[16] M. Rückert. ‘‘Lattice-based Blind signatures’’. In *Proceedings of The 16th Annual International Conference on the Theory and Application of Cryptology and Information Security*, Singapore, 5-9 December, 2010, LNCS 6477, 2010, pp. 413-430.

[17] P. L. Cayrel, R. Lindner, M. Rückert, R. Silva. ‘‘A Lattice-based Threshhold Ring Signature Scheme’’. In *Proceedings of the First International Conference on Cryptology and in Information Security in Latin America*, Puebla, 8-11 August, 2010, LNCS 6212, pp. 255-272.

## BIOGRAPHY

**Feng-he Wang** received the M.S degree in cryptography in 2006 from Xidian University. Xian, China, currently, he is a Ph.D. student of Key Laboratory of Computer Networks & Information Security of Ministry of Education at Xidian University. His research interests include lattice-based public key cryptography and digital signature. (E-mail fenghe2166@163.com).

**Yu-pu Hu** is a Professor in Key Laboratory of Computer Networks and Information Security of Ministry of Education at Xi dian University in Xi’an, China. He obtained his Ph.D degree in cryptography from School of Communication Engineering, Xidian University in 1999. His main research interests include Lattice-based cryptography, stream cryptography. (E-mail Yuph@mail.xidian.edu.cn)

**Baocang Wang** received his PhD degree in 2006 in Cryptology from Xidian University. Currently, he is an associate professor in Xidian University. His research interests lie in cryptography and network security. (Email:bcwang79@yahoo.com.cn)