Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 536527, 9 pages
http://dx.doi.org/10.1155/2014/536527

Research Article

An Efficient Homomorphic Aggregate Signature Scheme Based on Lattice

Zhengjun Jing[1,2]

[1] *Department of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China*
[2] *Department of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*

Correspondence should be addressed to Zhengjun Jing; zhengjun.jing@outlook.com

Received 19 June 2014; Accepted 15 September 2014; Published 6 November 2014

Academic Editor: Guoqiang Zhang

Homomorphic aggregate signature (HAS) is a linearly homomorphic signature (LHS) for multiple users, which can be applied for a variety of purposes, such as multi-source network coding and sensor data aggregation. In order to design an efficient postquantum secure HAS scheme, we borrow the idea of the lattice-based LHS scheme over binary field in the single-user case, and develop it into a new lattice-based HAS scheme in this paper. The security of the proposed scheme is proved by showing a reduction to the single-user case and the signature length remains invariant. Compared with the existing lattice-based homomorphic aggregate signature scheme, our new scheme enjoys shorter signature length and high efficiency.

1. Introduction

The homomorphic signature, proposed originally by Johnson et al. [1], is an important cryptographic primitive commonly used to secure computation. In a linear homomorphic signature scheme, a user generates a set of signatures on the corresponding messages in an information subspace. When the collection of messages is operated by a linear function which generates a new message belonging to the same information subspace, any other user, who does not know the signing private key, can produce a valid signature on this result of the linear function.

The linear homomorphic signature has been the subject of many researches in terms of its definitions, security model, and privacy property. The homomorphic property of signature scheme, proposed by Boneh et al. in [2], was viewed as signing a subspace and instantiated based on the bilinear maps over large prime fields in the random model. Later on, Gennaro et al. [3] showed the efficient homomorphic signature based on RSA over integers in the random model too. In the standard model, Freeman [4] defined a generic framework of linearly homomorphic signatures, in which three ordinary signature schemes having certain properties could be converted into linearly homomorphic signature schemes. More importantly, the framework provides enhanced security in the standard model under the computational Diffie-Hellman assumption, the q-strong Diffie-Hellman assumption, and the RSA assumption, respectively. Recently, the breakthrough has been achieved by Bohen and Freeman [5, 6]. Their works give an example of linearly homomorphic signature built using the lattice assumption over binary field [5], while they also show that a homomorphic signature supporting authenticated polynomial functions on signed data can be constructed by using "ideal lattice" in the random model [6]. Follow-up work by Wang et al. [7] implements an efficient lattice-based linearly homomorphic signature scheme using an additive homomorphic hash function over F_2, in which both the public key size and signature length are shorter. In addition, for the privacy of homomorphic signature, a notion of the so-called "weakly context hiding" is defined in [5], which requires the derived signature not to leak any information about the original messages provided that the original signatures are kept private. For attaining the stronger privacy notion, Ahn et al. [8] defines the concept of "strong context hiding," which requires the infeasibility of linking the derived signature to the signatures it was derived from even in the condition that the original signatures are public. After that, Attrapadung et al. [9, 10] proposed a new definition of privacy, called "adaptively context hiding," which requires context hiding on adversarially chosen signature with private key exposure. In other words, if a linearly homomorphic signature scheme guarantees unlinkability even when the original signatures are produced by illegitimate signing algorithm, this scheme holds the privacy of adaptively context hiding.

The linearly homomorphic signature can be used for many purposes, such as authenticating packets in network coding protocols and computing statistics on authenticated data. In particular for the secure network coding, it is the most effective cryptographic tool to prevent "pollution attack." Most of the above-mentioned linearly homomorphic signature schemes can be applied to prevent the pollution attack by the malicious node.

Here we want to point out that all of the above-mentioned authentication schemes could only be applicable to the case of single user or single source network coding system. Usually, in real world some applications involve many signatures on messages produced by many different users or sources. For example, in the multi-source network coding system [11, 12], packets from multiple different sources are needed to be linearly combined so as to exploit the benefits provided by network coding. For such multi-source network coding in an adversary situation, Agrawal et al. [13] constructed a complex scheme against pollution attack in the general case, in which a merged algorithm is used to generate several public keys and signatures in the mediate nodes. In order to find more efficient and practical solutions, the follow-up works [14–16] all considered the specific case where only the packets (or messages) that have the same identifier are combined together. Czap and Vajda's work [14] is obtained from the pairing-based homomorphic signature scheme proposed in [2], while Yan et al. [15] proposed an elegant homomorphic signature scheme based on the bilinear pairings and obtained a shorter homomorphic signature by using a novel homomorphic hash function. Recently, Zhang et al. [16] introduced aggregation property into homomorphic signature for multiple users case and formed a homomorphic aggregation signature scheme (HAS) by using preimage sampling function and Bonsai tree technique over random lattice.

However, these authentication schemes designed for multiuser case (or multi-source case) all have their own flaws. As is shown from the above, the unforgeability of both [14, 15] is based on CDH (computational Diffie-Hellman problem) in the bilinear group. As a result, these schemes involve a large number of point multiplication on elliptic curve. If a homomorphic signature scheme for multi-source can be used in network coding, it is necessary that this scheme should support the linearly homomorphic operations over binary field, just like that in [5, 7] for the single source. In addition, we need to emphasize that their security based on classical number theoretic problem is threatened by the power of quantum computers. As for the HAS proposed in [16], although its security is based on the hard assumption over lattice which is considered infeasible even under the quantum computer, the length of the aggregate signature is two times that of each original signature. We know the larger the length of the signature, the higher overhead of verification. Hence, it is significant to construct an efficient postquantum linearly homomorphic signature scheme for the multiple users case.

In this paper, we propose a short latticed-based linearly homomorphic aggregate signature scheme over binary field after optimizing our initial scheme in the multiple users case. Our scheme is an extension of the lattice-based linearly homomorphic signature scheme over F_2 in [7]. Each user, in our scheme, signs the original messages using their own private key, and the aggregate signature on aggregate message which is the combination of original messages from different users can be generated just by using the combination of original signatures without knowing any user's private key. In this way, these valid aggregate messages can be authenticated using a common public key formed by all the users' public keys. We point out that the common public key is independent of the signed message space, which means our signature scheme still supports signature on multiple message subspaces (or files) without updating the public keys. Compared to the HAS in [16], the length of aggregate signature in our scheme is as short as that of original signatures, which is only half the length of aggregate signature proposed in [16]. More importantly, this length of aggregate signature is independent of the number of the signing users. In addition, we also prove that the security of our solution can be reduced to that of the latticed-based LHS scheme in the single user case in [7].

The rest of this paper is organized as follows. In Section 2, we introduce the background about lattice and briefly overview the model of linearly homomorphic signature based on lattice over binary field in [7]. Our HAS scheme is described in detail in Section 3, including the general model definition, the initial scheme, and optimization. Section 4 proves the security of the presented scheme, and Section 5 is the analysis of the efficiency. Finally, in Section 6, we summarize this paper.

2. Preliminaries

2.1. Notation. We use $\mathbb{Z}$ and $\mathbb{R}$ to denote the set of integers and the set of real numbers, respectively. For any integer q, let $\mathbb{Z}_q$ denote the ring of integer mod q. By convention, we use bold lower case letters for vectors (e.g., $\mathbf{a}$) and bold upper case letters for matrix (e.g., $\mathbf{A}$). The member of vector is denoted by lowercase (e.g., a_i), while the ith column of a matrix is denoted by $\mathbf{a}_i$. For a positive integer k, $[k]$ denotes $\{1, \ldots, k\}$. In this paper, let $\widetilde{\mathbf{A}}$ denote the Gram-Schmidt orthogonalization of matrix $\mathbf{A}$. The Euclidean norm of a vector is considered as its length (e.g., $\|\mathbf{a}\|$), and the length of a matrix is the norm of its longest column vector (e.g., $\|\mathbf{A}\| = \max \|\mathbf{a}_i\|$). In addition, the function negl(n) is negligible in n if it is smaller than all polynomial fractions for larger n.

2.2. Random Lattice and Hard Assumption. Let $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^m$ be a set of n linearly independent vectors; then, the lattice Λ generated by the basis $\mathbf{B}$ is $\Lambda = \{\mathbf{B} \cdot \mathbf{x} : \mathbf{x} \in \mathbb{Z}^n\} \subset \mathbb{R}^m$. In the cryptography based on lattice, we always focus on the integer lattice where the lattice points are contained in $\mathbb{Z}^m$. For some integer $q \geq 2$, m, n, let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ be a random matrix. Then, the two kinds of full rank random lattice defined by $\mathbf{A}$ are used in this paper. Their specific definitions are as follows:

$$\Lambda_q^{\perp}(\mathbf{A}) = \left\{\mathbf{e} \in \mathbb{Z}_q^m : \mathbf{A}\cdot\mathbf{e} = \mathbf{0} \bmod q\right\},$$
$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \left\{\mathbf{e} \in \mathbb{Z}_q^m : \mathbf{A}\cdot\mathbf{e} = \mathbf{u} \bmod q\right\}. \quad (1)$$

In fact, the lattice $\Lambda_q^{\mathbf{u}}(\mathbf{A})$ is a coset of $\Lambda_q^{\perp}(\mathbf{A})$. Namely, $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \Lambda_q^{\perp}(\mathbf{A}) + \mathbf{t}$, where $\mathbf{A}\cdot\mathbf{t} = \mathbf{u} \bmod q$. In addition, it is noted that the variable n is the security parameter as the prior works defined, and the other variables are the functions of n. Typically, m is $O(n\log n)$ and the modulus q is some small polynomial, for example, $O(n^3)$.

Hard Assumption. The security of lattice-based LHS schemes [5, 7] is all based on the hardness assumption of the short integer solution (SIS) problem over the lattice $\Lambda_q^{\perp}(\mathbf{A})$. The definition of SIS problem is as follows.

Definition 1. Given positive integers q, m, n and a real δ, for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the goal of SIS (q, m, δ) is to find a nonzero vector $\mathbf{v} \in \Lambda_q^{\perp}(\mathbf{A})$ such that $\|\mathbf{v}\| \leq \delta$.

In [17], it has been proven that solving SIS problem on the average is as hard as approximating certain lattice problems in the worst case, such as SIVP problem (shortest independent vectors problem).

2.3. Gaussian Distribution on Lattices. Gaussian distribution technique is widely used in the analysis of the results in the area of lattice-based cryptography. Here, we briefly review some important conclusion from previous works [5, 15, 16], which will be used to analyze our scheme.

Discrete Gaussian Distribution. For the parameter $\sigma > 0$ and any vector $\mathbf{c} \in \mathbb{R}^m$, the probability density function of Gaussian distribution $D_{\sigma,\mathbf{c}}$ on $\mathbb{R}^m$ centered at $\mathbf{c}$ is defined as $\rho_{\sigma,\mathbf{c}} = \exp(-\pi\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2)$. For the m-dimensional lattice Λ, the discrete Gaussian distribution over Λ is a conditional probability distribution with center $\mathbf{c}$ and parameter σ, which is defined as $D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) = \rho_{\sigma,\mathbf{c}}(\mathbf{x})/\sum_{\mathbf{v}\in\Lambda}\rho_{\sigma,\mathbf{c}}(\mathbf{v})$, where $\mathbf{x} \in \Lambda$. Micciancio and Regev [17] introduced the notion of "smoothing parameter" of lattice and showed that if the parameter σ is greater than the smoothing parameter, then the discrete distribution $D_{\Lambda,\sigma,\mathbf{c}}$ is statically close to the continuous distribution $D_{\sigma,\mathbf{c}}$. In particular, $\mathrm{P}_{\mathbf{x}\sim D_{\Lambda,\sigma,\mathbf{c}}}\{\|\mathbf{x}-\mathbf{c}\| > \sigma\sqrt{m}\} \leq \mathrm{negl}(n)$.

Sampling from Discrete Gaussian. Gentry et al. [18] gave a new bound on the smoothing parameter relative to a certain lattice quality and showed algorithm for sampling from discrete Gaussian distribution which was commonly used in signature scheme [5, 7, 16, 19]. In addition, Boneh and Freeman in [5] showed that the sum of independent discrete Gaussian variables still remains discrete Gaussian distribution. Some relevant facts are listed as follows.

Lemma 2 (see [18, Theorem 4.1]). *Given a basis $\mathbf{T}$ of any dimension lattice Λ, a parameter $\sigma > \|\overline{\mathbf{T}}\|\cdot\omega(\sqrt{\log n})$, and a center $\mathbf{c} \in \mathbb{R}^m$, there is a PPT (probabilistic polynomial-time) algorithm that outputs a sample from a distribution that is statically close to $D_{\Lambda,\sigma,\mathbf{c}}$.*

Lemma 3 (see [18, Theorem 5.6]). *Let n be a positive integer, $q \geq 2$, and $m > 2n\log(q)$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, let $\mathbf{T}$ be a basis of $\Lambda_q^{\perp}(\mathbf{A})$ and $\sigma > \|\overline{\mathbf{T}}\|\cdot\omega(\sqrt{\log n})$; then, one has the following.*

(1) *For any $\mathbf{t} \in \mathbb{Z}^n$, there is a probability polynomial-time algorithm SamplePre $(\mathbf{A}, \mathbf{T}, \sigma, \mathbf{t})$ that outputs a sample $\mathbf{t}'$ from a distribution that is statically close to $D_{\Lambda_q^{\mathbf{t}}(\mathbf{A}),\sigma}$. In particular, the vector $\mathbf{t}'$ satisfies $\|\mathbf{t}'\| \leq \sigma\sqrt{m}$ with overwhelming probability.*

(2) *For any $\mathbf{t} \sim D_{\mathbb{Z}^m,\sigma,\mathbf{0}}$, the distribution of syndrome $\mathbf{u} = \mathbf{A}\cdot\mathbf{t} \bmod q$ is statically close to uniform over $\mathbb{Z}_q^n$.*

Lemma 4 (see [5, Theorem 9]). *For a lattice Λ, the parameter $\sigma \in \mathbb{R}$, and $\mathbf{t}_{i\in[k]} \in \mathbb{Z}^m$, let $\mathbf{x}_{i\in[k]}$ be mutually independent random variables sampled from a discrete Gaussian distribution $D_{\mathbf{t}_i+\Lambda,\sigma,\mathbf{0}}$. Let $\mathbf{c} = (c_1,\ldots,c_k) \in \mathbb{Z}^k$, $g = \gcd(c_1,\ldots,c_k)$, and $\mathbf{t} = \sum_{i=1}^k c_i\cdot\mathbf{t}_i$. Suppose that $\sigma > \|\mathbf{c}\|\cdot\eta_\varepsilon(\Lambda)$ where $\eta_\varepsilon(\Lambda)$ is the smooth parameter of lattice Λ for some negligible number ε; then, $\mathbf{z} = \sum_{i=1}^k c_i\cdot\mathbf{x}_i$ is statically close to $D_{\mathbf{t}+g\Lambda,\|\mathbf{c}\|\cdot\sigma,\mathbf{0}}$.*

2.4. Short Basis of Lattice. In cryptography based on lattice, a short basis of a lattice can be considered a trapdoor basis which was used as private key in cryptographic application. For the lattice $\Lambda_q^{\perp}(\mathbf{A})$, its short basis $\mathbf{T}$ can be generated using the *TrapGen* algorithm proposed by Alwen and Peikert in [20]. In addition, the common public key used to sign in our initial HAS scheme consists of the public keys of multiple users in the form of $\mathbf{A} = \mathbf{A}_1\|\cdots\|\mathbf{A}_l$, where l is the number of the signing users. To derive a new short basis of the high-dimension lattice $\Lambda^{\perp}(\mathbf{A}) \subset \mathbb{Z}_q^{lm}$, some lemmas about the basis delegation mechanism proposed by Cash et al. in [21] will be employed. All of them are listed below.

Lemma 5 (see [19, Theorem 3.2]). *For $q > 2$ and $m > 5n\log q$, there is a probabilistic polynomial-time algorithm $TrapGen(1^n)$ that outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ statically close to a uniform matrix in $\mathbb{Z}_q^{n\times m}$ and a basis $\mathbf{T} \in \mathbb{Z}_q^{n\times m}$ of the lattice $\Lambda_q^{\perp}(\mathbf{A})$ such that $\|\overline{\mathbf{T}}\| \leq \mathrm{O}(\sqrt{\mathrm{n}\log\mathrm{q}})$ with overwhelming probability.*

Lemma 6. *For an arbitrary basis $\mathbf{T} \in \mathbb{Z}_q^{m\times m}$ of the lattice $\Lambda^{\perp}(\mathbf{A})$ about a random matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and the parameter $\sigma > \|\overline{\mathbf{T}}\|\cdot\omega(\sqrt{\log n})$; then,*

(1) *(see [20, Lemma 3.2]) for any matrix $\mathbf{A}' \in \mathbb{Z}_q^{n\times m'}$, there is a deterministic polynomial-time algorithm ExtBasis$(\mathbf{T}, \mathbf{B} = \mathbf{A}\|\mathbf{A}')$ that outputs a basis $\mathbf{T}'$ of the lattice $\Lambda_q^{\perp}(\mathbf{B}) \subset \mathbb{Z}^{(m+m')\times(m+m')}$ such that $\|\overline{\mathbf{T}'}\| = \|\overline{\mathbf{T}}\|$;*

(2) *(see [20, Lemma 3.3]) there is a probabilistic polynomial-time algorithm RandBasis*$(\mathbf{T}, \sigma)$ *that outputs another basis* $\mathbf{T}'$ *of the lattice* $\Lambda_q^{\perp}(\mathbf{A})$*, which is statically independent of the original basis* $\mathbf{T}$ *and is still short.*

2.5. Linearly Homomorphic Signature Scheme (LHS) Based on Lattice over F_2. Our homomorphic aggregate signature scheme is an extension of an efficient linearly homomorphic signature scheme proposed by Wang et al. in the single user case [7], which makes improvement on the scheme of Boneh and Freeman [5]. At present, lattice-based LHS over F_2 is called L-limited, which means we can only guarantee successful verification for combination of a finite number of valid signatures where L is the maximal number. Here, we briefly describe the *LHS1* as follows, and the details about lattice-based LHS over binary field can be referred to [5, 7].

Homomorphic Hash $h_{\boldsymbol{\alpha}}$ *Based-on Lattice*. Lyubashevsky and Micciancio in [22] defined a secure hash function based on the approximate SVP (short vector problem) of lattice, which was used in [5]. This hash function family maps $\mathbb{Z}_q^m$ to $\mathbb{Z}_q$ in the way of inner product and holds homomorphic property. Specifically, given that vectors $\boldsymbol{\alpha}$, $\mathbf{v}_1$, $\mathbf{v}_2$ belonged to $\mathbb{Z}_q^m$ with vector $\boldsymbol{\alpha}$ fixed, the hash function $h_{\boldsymbol{\alpha}} = \langle \boldsymbol{\alpha}, \mathbf{v}_i \rangle$ (i is 1 or 2) satisfies linearly homomorphic conditions. Namely, it holds $h_{\boldsymbol{\alpha}}(\mathbf{v}_1 + \mathbf{v}_2) = h_{\boldsymbol{\alpha}}(\mathbf{v}_1) + h_{\boldsymbol{\alpha}}(\mathbf{v}_2)$ and $h_{\boldsymbol{\alpha}}\langle c \cdot \mathbf{v}_i \rangle = c \cdot h_{\boldsymbol{\alpha}}(\mathbf{v}_i)$ where $c \in \mathbb{Z}_q$.

Wang's Signature Scheme. The Wang's lattice-based LHS scheme [7], which will be called *LHS1*, consists of four polynomial-time algorithms proposed as follows.

WSetup. Let the parameters (q, m, n, σ) be the same as those in Lemma 5. Given that H be a collision-resistant hash function which maps $(0, 1)^*$ to $\mathbb{Z}_q^m$ and letting the coefficients a of the linearly function belong to F_2, the signer runs $TrapGen(1^n)$ algorithm to produce the pair of public key and private key $\{\mathbf{T} \in \mathbb{Z}_q^{m\times m}, \mathbf{A} \in \mathbb{Z}_q^{n\times m}\}$.

WSign. To sign a subspace V_{id} of the message space $\{0, 1\}^m$, where $\mathrm{id} \in \{0, 1\}^n$ is a identifier of V, given the fact that the set of vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ is the basis of V, the signer does as follows to sign a basis vector $\mathbf{v}_{j\in[k]}$.

(1) Compute n vectors $\boldsymbol{\alpha}_{i\in[n]} = H(\mathrm{id}\|i) \subset \mathbb{Z}_q^m$.

(2) Compute the hash value of $\mathbf{v}_j$ through homomorphic hash function $h_{\boldsymbol{\alpha}}$, and denote it as a column vector $\mathbf{h}_j = (v_{i\in[n]})^T \in \mathbb{Z}_q^n$, where the element is $v_i = h_{\boldsymbol{\alpha}}(\mathbf{v}_j) \bmod q$.

(3) Use the $SamplePre(\mathbf{A}, \mathbf{T}, \sigma, \mathbf{h}_j)$ algorithm in Lemma 3 to attain a signature $\mathbf{s}_j$ of the hash value $\mathbf{h}_j$, and the linearly homomorphic signature on $\mathbf{v}_j$ can be denoted by $(\mathrm{id}, \mathbf{s}_j)$.

WVerify. Let L be the maximal number of signatures that can be combined. To verify the linearly homomorphic signature of the message $\mathbf{v}_j$, the verifier firstly computes the hash value $\mathbf{h}_j$ of $\mathbf{v}_j$ just as steps (1) and (2) in *Sign* do and then outputs 1 (accept) if and only if the conditions hold, such as $\mathbf{A} \cdot \mathbf{s}_j = \mathbf{h}_j \bmod q$ and $\|\mathbf{e}_j\| \le L\sigma\sqrt{m}$, or outputs 0 (reject).

WCombine. Given l pairs of $(a_i, \mathbf{s}_i, \mathbf{v}_i)$, where $l \le L$, it outputs a vector $\sum_{i=1}^{l} a_i \cdot \mathbf{s}_i (\mathrm{mod}\, q)$ as the signature of the message $\sum_{i=1}^{l} a_i \cdot \mathbf{v}_i (\mathrm{mod}\, q)$.

For the linearly homomorphic signature in adversary situation, there is two types of forgeability. For example, an output $(\mathrm{id}^*, \mathbf{v}^*, \mathbf{s}^*)$ produced by an adversary can be accepted by *Verify* algorithm where either (1) $\mathrm{id}^* \ne \mathrm{id}$ or (2) $\mathrm{id}^* = \mathrm{id}$, but $\mathbf{v}^* \notin V_{\mathrm{id}}$. Theorem 2 in [7] shows that *LHS1* is unforgeable against these types of adversaries in oracle model.

Lemma 7 (see [7, Theorem 2]). *If an stateful adversary without knowing the signing private can output any kind of the above forgery with the probability* ε*, the SIS problem can be solved by a challenger with a probability* 2ε*.*

3. Homomorphic Aggregate Signature Based on Lattice

For the most general setting of multiuser, Agrawal et al. in [13] have shown that it is difficult to find efficient solutions to homomorphic signature. In this paper, we deal with the specific case, the same one considered in [11–13], where only the messages tagged the same identifier are combined together. In addition, our signature scheme requires that a trusted private key generator (PKG) is available, which makes it possible that all users have their own public-private key pairs.

In the HAS, assuming $\mathrm{id} \in \{0, 1\}^n$ is a unique identifier of messages subspace, for a message $\mathbf{v}_i$ from the subspace V_{id}, the signed message is a tuple of $(\mathrm{id}, \mathbf{v}_i, \mathbf{s}_i)$, where $\mathbf{s}_i$ is the signature of $\mathbf{v}_i$ from the jth user using his (or her) own private key. So, the aggregate message is gained through linearly combination of different messages tagged the same id from distinct users. Now, we present the system model of HAS and give a detailed structure of our signature scheme.

3.1. Definition of HAS. The presented system definition of lattice-based HAS is a variant of that of linearly homomorphic signature in [5]. Compared with the model of single-user homomorphic signature [5, 7], the *Setup* and *Verify* parts of HAS system need to define some new properties and additional operators, while the *Sign* part does not change. Formally, the definition of HAS is a tuple of polynomial-time algorithms $HAS(Setup, Sign_{sk_i}, Combine, Verify_{cpk})$, which is as follows.

$Setup(1^n, L)$. This probabilistic algorithm takes as input the security parameter 1^n and the maximum number of users L and outputs the public-private pair (pk_i, sk_i) for each user $(0 < i \le L)$ and the common public key pck shared with everyone.

$Sign(id, \mathbf{m}_i, sk_i)$. For the jth user, this probabilistic algorithm takes as input a message $\mathbf{m}_i$ of subspace V_{id} and the private key sk_j, and outputs the signature $\mathbf{s}_i$ on message $\mathbf{m}_i$.

$Combine(id, \mathbf{a}, \{(\mathbf{m}_i, \mathbf{s}_i)\}_{i=1}^{l})$. Given the combination coefficient vector $\mathbf{a}$ and l $(l \le L)$ pairs of message sharing the same id and the corresponding signature, output the aggregate signature $\mathbf{s}$ on the aggregate message $\sum_{i=1}^{l} a_i\mathbf{m}_i$.

Verify$_{cpk}(id, \sum_{i=1}^{l} a_i\mathbf{m}_i, \mathbf{s})$. This is a deterministic algorithm. Given an identifier id, the common public key cpk, the aggregate message $\sum_{i=1}^{l} a_i\mathbf{m}_i$, and the corresponding signature $\mathbf{s}$, output either 1 (accept) or 0 (reject).

In terms of the correctness and security for the homomorphic aggregate signature scheme, it should have not only the characteristics of general linearly homomorphic signature in case of one user, but also some features of its own in multiuser case. On the one hand, assume that each user is honest, the verification should be able to accept the valid signed message from each user, while a forged signed message must be rejected, which is the same as in the case of the single user. On the other hand, given a series of valid signed message (id, $\mathbf{m}_i, sk_i$), where $i \in [l]$, a new valid signed message denoted by (id, $\sum_{i=1}^{l} a_i\mathbf{m}_i, \mathbf{s}$) can be produced without having access to any private key.

3.2. Our Scheme. According to the definition of HAS, we show how to extend the homomorphic signature scheme *LHS1* described in Section 2.5 to handle multiuser case. Our initial HAS scheme is as follows.

Setup$(1^n, L)$. Given a security parameter n and the maximum number of users L, the PKG initializes the scheme from four aspects.

(1) Choose parameters params = $\{m, q\}$. For $\beta = \text{poly}(n)$ and let $q \geq \beta\omega(\sqrt{\log n})$ and $m > cn\log(q)$, where $c > 0$ is a constant.

(2) For L users, *TrapGen*(1^n) algorithm is repeatedly run L times to generate matrix $\mathbf{A}_i \in \mathbb{Z}_q^{n\times m}$ and the corresponding trapdoor basis $\mathbf{T}_i$ of $\Lambda_q^{\perp}(\mathbf{A}_i)$, where $0 < i \leq L$.

(3) Let $H : \{0,1\}^* \rightarrow \mathbb{Z}_q^{Lm}$ be a collision-resistant hash function which is viewed as a random oracle, and let h be a lattice-based homomorphic hash function described in Section 2.5.

(4) The pair of $\{\mathbf{A}_i, \mathbf{T}_i\}$ is assigned to the corresponding user u_i as the user's private key and public key, respectively. Let $\mathbf{A}_0 = \mathbf{A}_1\|\cdots\|\mathbf{A}_L \in \mathbb{Z}_q^{n\times Lm}$ be the common public key and publish it to all the users. Of course, it is required that delivering the private key should be done secretly.

Sign$(id, \mathbf{m}_i, \mathbf{T}_i)$. For the ith user, given the common signing key $\mathbf{A}_0$, private key $\mathbf{T}_i$, and one of basis vectors of message subspace V_{id}, for example, $\mathbf{m}_i \in V_{\text{id}} \subset \mathbb{Z}_2^m$ where id $\in \{0,1\}^n$, the signature on message $\mathbf{m}_i$ is produced as follows.

(1) To obtain the short basis of $\Lambda^{\perp}(\mathbf{A}_0) \subset \mathbb{Z}^{lm}$, the algorithm *ExtendBasis*$(\mathbf{T}_i, \mathbf{A}_0)$ in Lemma 6 is run to get $\mathbf{S}_i$ such that $\|\widetilde{\mathbf{S}}_i\| = \|\widetilde{\mathbf{T}}_i\|$.

(2) Use the homomorphic hash function to produce the hash value $\mathbf{h}_i \in \mathbb{Z}_q^n$ of the message $\mathbf{m}_i$, as is done in the *WSign* of *LHS1*.

(3) Output the signature $\mathbf{e}_i \in \mathbb{Z}_q^{lm}$ on the hash value by using algorithm *SamplePre*$(\mathbf{A}_0, \mathbf{S}_i, \sigma, \mathbf{h}_i)$.

Combine$(id, \mathbf{A}_0, \{(c_i, \mathbf{m}_i, \mathbf{e}_i)\}_{i=1}^{l})$. Given a common public key $\mathbf{A}_0$ and the $l \leq L$ messages tagged the same id from the corresponding users, output an aggregate signature $\sum_{i=1}^{l} c_i \cdot \mathbf{e}_i$ on the combined message $\sum_{i=1}^{l} c_i \cdot \mathbf{m}_i$, where $c_i \in \{0,1\}$.

Verify$(id, \mathbf{A}_0, \mathbf{m}_{agg} = \sum_{i=1}^{l} c_i\mathbf{m}_i, \mathbf{e}_{agg} = \sum_{i=1}^{l} c_i\mathbf{e}_i)$. Given a common public key $\mathbf{A}_0$, an identifier id, an aggregate message $\mathbf{m}_{\text{agg}}$, and the corresponding signature $\mathbf{e}_{\text{agg}}$, do the following.

(1) Compute the hash value $\mathbf{h}_{\text{agg}}$ of $\mathbf{m}_{\text{agg}}$ using the homomorphic hash function just like the step (2) of *Sign* does.

(2) Verify two conditions such as $\mathbf{A}_0\mathbf{e}_{\text{agg}} = \mathbf{h}_{\text{agg}} \pmod q$ and $\|\mathbf{e}_{\text{agg}}\| \leq L\sigma\sqrt{lm}$.

(3) Output 1 (accept) if and only if the above two conditions hold. Otherwise, output 0 (reject).

3.3. Correctness. First of all, we show the correctness of the proposed HAS scheme provided that the related functions are all computed successfully, such as homomorphic hash function $h_{\boldsymbol{\alpha}}$, preimage sampling function, and collision-resistant hash function H.

We know that $\mathbf{m}_{\text{agg}} = \sum_{i=1}^{l} c_i\mathbf{m}_i$, $\mathbf{e}_{\text{agg}} = \sum_{i=1}^{l} c_i\mathbf{e}_i$, and $h_{\boldsymbol{\alpha}_{i,j}}(\mathbf{m}_i) = (h_{\boldsymbol{\alpha}_{i,1}}(\mathbf{m}_i)\cdots h_{\boldsymbol{\alpha}_{i,n}}(\mathbf{m}_i))^T$, where $c_i \in \{0,1\}$, $0 < i \leq l \leq L$, $0 < j \leq n$ and the vector $\boldsymbol{\alpha}_{i,j} = H(\text{id}\|j)$. Since the messages combined are tagged the same identifier, the vectors $\boldsymbol{\alpha}_{i\in[l],j\in[n]}$ originated from $H(\text{id}\|j)$ are the same for each user. Thus, we can directly use $h_{\boldsymbol{\alpha}}(\mathbf{m}_i)$ to represent the hash value of $\mathbf{m}_i$ in order to simplify the notations. Then

$$\begin{aligned}
\mathbf{A}_0 \cdot \mathbf{e}_{\text{agg}} &= \mathbf{A}_0 \cdot \sum_{i=1}^{l} c_i\mathbf{e}_i \\
&= \sum_{i=1}^{l} c_i\mathbf{A}_0\mathbf{e}_i \\
&= \sum_{i=1}^{l} c_i h_{\boldsymbol{\alpha}}(\mathbf{m}_i) \\
&= \sum_{i=1}^{l} h_{\boldsymbol{\alpha}}(c_i\mathbf{m}_i) \\
&= h_{\boldsymbol{\alpha}}\left(\sum_{i=1}^{l} c_i\mathbf{m}_i\right) \\
&= h_{\boldsymbol{\alpha}}(\mathbf{m}_{\text{agg}}) \bmod q.
\end{aligned} \tag{2}$$

Hence, the condition one in the progress of *Verify* holds, while (2) holds because of the homomorphic property of function $h_{\boldsymbol{\alpha}}$. Furthermore, since each signature $\mathbf{e}_i$ on massage is obtained by preimage sampling algorithm *SamplePre*$(\mathbf{A}_0, \mathbf{S}_i, \sigma, \mathbf{h}_i)$, the length of $\mathbf{e}_i$ denoted by the Euclidean norm is not larger than $\sigma\sqrt{lm}$. Thus, the upper length bound of aggregate signature is as follow:

$$\left\|\mathbf{e}_{\text{agg}}\right\| = \left\|\sum_{i=1}^{l} c_i\mathbf{e}_i\right\| \le \sum_{i=1}^{l}\left\|c_i\mathbf{e}_i\right\| \le L\sigma\sqrt{lm}, \quad (3)$$

where $c_i \in \{0,1\}$ and the first inequality over (3) holds by the triangle inequality theorem. It is inferred from the above analysis that the signature $\mathbf{e}_{\text{agg}}$ on message $\sum_{i=1}^{l} c_i\mathbf{m}_i$ can be verified and can satisfy the requirement of correctness of aggregate signature model defined in Section 2.1.

3.4. Optimization. The aggregated signature $(\text{id}, \mathbf{m}_{\text{agg}}, \mathbf{e}_{\text{agg}})$ of the proposed HAS scheme is accepted by *Verify* algorithm and confirms to the definition of HAS. However, compared with the signature of single user scheme in [7], it is easy to observe that, in our scheme, the length of each signature on message grows linearly with the number of users. The cause for this problem is that the dimension of random lattice used to sign in our multiuser scheme increases with the number of users, which can be clearly seen from the common public key $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times lm}$. Obviously, the larger the common public key size, the longer the length of aggregate signature and the larger communication cost. Through the observation of the entire HAS scheme, it is clear that the common public key $\mathbf{A}_0$ is shared by each user. In order to reduce the common public key size, the algorithm *RandomBasis* in Lemma 6 is introduced to our signature scheme, which could generate several different short bases through an arbitrary basis of a lattice. So, we can optimize the proposed HAS scheme from the following aspects.

In the $Setup(1^n, L)$ phase, while the parameters are consistent with those of the initial scheme, we firstly use $TrapGen(1^n)$ algorithm only once to get a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a short basis $\mathbf{T}_1$ of lattice $\Lambda^{\perp}(\mathbf{A}) \in \mathbb{Z}_q^{m}$, and then, call $RandomBasis(\mathbf{T}_1, \sigma)$ algorithm repeatedly to generate $l-1$ independent short basis $\mathbf{T}_i$ $(2 \le i < l)$ of lattice $\Lambda^{\perp}(\mathbf{A})$. Therefore, let the vectors $\mathbf{T}_i$ $(1 \le i < l)$ be user's private key, respectively, and let $\mathbf{A}$ be the common public key shared by everyone, just like the way in [16]. In $Sign(\text{id}, \mathbf{m}_i, \mathbf{T}_i)$, we can directly take preimage sampling algorithm $SamplePre(\mathbf{A}, \mathbf{T}_i, \sigma, h_{\boldsymbol{\alpha}}(\mathbf{m}_i))$ to get the signature $\mathbf{e}_i \in \mathbb{Z}_q^m$ on the hash value of $\mathbf{m}_i$ denoted by $h_{\boldsymbol{\alpha}}(\mathbf{m}_i)$, where $h_{\boldsymbol{\alpha}}(\mathbf{m}_i)$ is obtained as done in the initial scheme. It should be noted that each user should use his private key in the signing progress. As for the *Combine* and *Verify*, the operation of both sections is almost unchanged except that one of verification conditions becomes $\|\mathbf{e}_{\text{agg}}\| \le L\sigma\sqrt{m}$.

It can be proved that the optimized solution meets the correctness of homomorphic aggregate signature model, just like the way of proving in the initial HAS scheme. More importantly, the optimization reduces the size of the common public key and signature to as short as the sizes of those in the single user scheme *LHS1.*

4. Security Analysis

For the security of linearly homomorphic signature (LHS), two aspects, including unforgeability and privacy, are generally considered in solutions [4]. Clearly, this consideration also can apply to the security of linearly homomorphic aggregate signature scheme (HAS). In this section, we focus on the security of LHS scheme.

To prove the unforgeability and privacy of the proposed signature scheme, a reduction to the case of the single user is shown [7]. Obviously, if the number of users equals one $(L = 1)$, our (optimized) scheme is almost identical to LHS1 scheme. Next, we discuss unforgeability and privacy of the proposed HAS scheme in the multiuser case.

4.1. Unforgeability. Assuming that no polynomial-time algorithm can solve SIS problem in the average case, Lemma 7 proves that in LHS1 the advantage in winning the unforgeability game is negligible. Based on this result, it is able to prove computational security from a reduction of our HAS signature scheme to LHS1 and get the following theorem.

Theorem 8. *The presented HAS scheme is unforgeable, if the lattice-based linearly homomorphic signature scheme LHS1 in [7] is unforgeable.*

Proof. With the usual method of reduction, assuming there is a polynomial-time algorithm A^* to generate a forged signed aggregate message in HAS, an efficient algorithm C is able to be constructed to produce a forged signed message for *LHS1* in polynomial time.

The algorithm A^* takes as input the tuple of public parameters, the common signing key shared by all users, and the set of corresponding signed messages in the subspace V_{id} from $l \le L$ users. The signed message from the ith user is denoted by $(\text{id}, \mathbf{m}_i, \mathbf{s}_i)$, where id is the valid identifier for message subspace. The output of algorithm A^*, denoted by $(\text{id}^*, \boldsymbol{\beta}^*, \mathbf{y}^*, \mathbf{s}^*)$, can be accepted by the *Verify* algorithm of the presented homomorphic aggregated signature, where $\boldsymbol{\beta}^* \in F_2^l$ is the aggregate coefficient vector. However, this is a forged signed aggregate message, in which either $\text{id}^* \neq \text{id}$ or $\text{id}^* = \text{id}$ and $\mathbf{y}^* \neq \sum_{i=1}^{l} \beta_i^*\mathbf{m}_i$ for $\mathbf{y}^* \neq \mathbf{0}$.

We assume that the challenger takes the system parameters (q, m, n, σ, l, H) and the key-pair $(\mathbf{A}, \mathbf{T})$ to employ A^* algorithm, where $\mathbf{T}$ is a short vector of lattice $\Lambda^{\perp}(\mathbf{A}) \in \mathbb{Z}_q^m$ and l is the dimension of the subspace V_{id}. Thus, the construction of algorithm C by challenger is as follows.

(1) Construct a homomorphic aggregated signature scheme (HAS) with l users. First of all, the challenger extends the *WSetup* of *LHS1* to generate the l key-pairs $(\mathbf{A}, \mathbf{T}_{j\in[l]})$ for users. Specifically, let matrix $\mathbf{A}$ be the common public key shared by users, and the corresponding private key $\mathbf{T}_j$ of the jth user is the output of the algorithm $RandomBasis(\mathbf{T}, \sigma)$, which run repeatedly l times. Then, for message subspace V_{id}, assume that challenger keeps several answers from the random oracle H and is stored in a list (e.g., *List*1), which is just like what the *H-Query* does in *LHS1.* As a result, each element a_{id} of *List*1 is a tuple $(\text{id}, \{(\mathbf{e}_j, \mathbf{u}_j)\}_{i=1}^{l})$ where $\mathbf{e}_j \sim D_{\mathbb{Z}^m,\sigma,\mathbf{0}}$ is directly sampled from the discrete Gaussian distribution $D_{\mathbb{Z}^m}$ for Gaussian parameter σ, and $\mathbf{u}_j = \mathbf{A}\mathbf{e}_j \bmod q$ is

statically close to uniform distribution according to Lemma 3, which could be considered as the basis vectors of V_{id}. Thereby, a signature $\mathbf{s}_j$ on message $\mathbf{u}_j$ from the jth user can be produced by calling the algorithm *SamplePre*$(\mathbf{A}, \mathbf{T}_i, \mathbf{u}_j)$.

(2) Call algorithm A^*, which takes on inputs the system parameters and l signed messages $(\mathrm{id}, \mathbf{v}_j, \mathbf{s}_j)$. The output of A^* is a forged signed message $(\mathrm{id}^*, \boldsymbol{\beta}^*, \mathbf{y}^*, \mathbf{s}^*)$ in the HAS scheme, where $\boldsymbol{\beta}^* \in F_2^l$ is the aggregate coefficient vector and $\mathbf{s}^*$ is the aggregate signature on the aggregate message $\mathbf{y}^* = \sum_{j=1}^{l} \mathbf{u}_j^*$. It is worth noting that the vectors $\mathbf{u}_j^* \in \mathbb{Z}_q^n$ eventually exist, though they are not known and may not be unique.

(3) The algorithm C outputs the signed message $(\mathrm{id}^*, \boldsymbol{\beta}^*, \mathbf{y}^*, \mathbf{s}^{**})$ where $\mathbf{s}^{**} = \mathbf{s}^*$.

Now, we analyze the reduction and show that the output of C is a forged signed message in *LHS1*. Firstly, it should be determined whether the output of C can be accepted by *LHS1*. Since the output of $\mathbf{A}^*$ passes the verification of the homomorphic aggregate signature (HAS) scheme, we can know $\mathbf{A}\mathbf{s}^* = \mathbf{y}^* (\mathrm{mod}\, q)$ and $\|\mathbf{s}^*\| \le L\sigma\sqrt{m}$. According to this result, it is easy to obtain that it can pass the verification in *LSH1* because the parameters used to verify in both schemes are the same, for example, the matrix $\mathbf{A}$ and σ. On the other hand, the output of C is a forged signed message in *LHS1* if and only if either $\mathrm{id}^* \neq \mathrm{id}$ or $\mathrm{id}^* = \mathrm{id}$ and $\mathbf{y}^* \neq \sum_{i=1}^{l} \beta_i^* \mathbf{m}_i$ for $\mathbf{y}^* \neq \mathbf{0}$. Since the signed message $(\mathrm{id}^*, \boldsymbol{\beta}^*, \mathbf{y}^*, \mathbf{s}^*)$ is a forged signed aggregate message, this means that the condition above is satisfied. Consequently, this signed message $(\mathrm{id}^*, \mathbf{y}^*, \mathbf{s}^{**})$ is a forged signed message in *LHS1*, and Theorem 8 is proved. □

4.2. Privacy. In order to prove the privacy of this presented signature scheme, we introduce the "weakly context hiding" of the linearly homomorphic aggregate signature, which is adapted from [5] in the case of single user. The "weakly context hiding" property of the linearly homomorphic signature means the signature on the derived message in some message subspace spanned by $\{\mathbf{v}_1, \dots, \mathbf{v}_l\}$ does not disclose any information about the original messages $\{\mathbf{v}_1, \dots, \mathbf{v}_l\}$. However, the linear function to combine the messages $\mathbf{v}_i$ is not hidden while the original signatures on these messages are kept private, which is why it is called "weakly context hiding." According to the definition of the linearly homomorphic aggregate signature, we know that this property also applies to our scheme in the case of multiuser, which is shown through proving Theorem 9.

Theorem 9. *The proposed linearly homomorphic aggregate signature scheme has the "weakly context hiding" property.*

Proof. Suppose that there are $l \le L$ users who are assigned the corresponding private key $\mathbf{T}_{i \in [l]}$ by running the *Setup* of the proposed signature scheme. In the proposed signature scheme, the ith user employs the algorithm *SamplePre* to sign original message $\mathbf{m}_i$, and the aggregate signature $\mathbf{s}_{\mathrm{agg}}$ on message $\sum_{i=1}^{l} c_i \mathbf{m}_i$ is generated through combining the original signatures $\mathbf{s}_i$ of the corresponding messages $\mathbf{m}_i$, where each message combined should be tagged the same id and $c_i \in \{0, 1\}$. According to Lemma 3 about the distribution of original signatures, despite coming from different users, we know they are all statically close to the Gaussian distribution, for example, $\mathbf{s}_i \sim D_{\mathbf{t}_i + \Lambda^{\perp}(\mathbf{A}), \sigma, \mathbf{0}}$, where $\mathbf{t}_i \in \mathbb{Z}^m$ is an arbitrary solution to $\mathbf{A}\mathbf{t}_i = h_{\boldsymbol{\alpha}}(\mathbf{m}_i) \bmod q$ and the definitions of remaining variables are the same as those in the *Sign* part of proposed scheme. Hence, by knowledge of Lemma 4, the distribution of the aggregate signature $\mathbf{s}_{\mathrm{agg}} = \sum_{i}^{l} c_i \cdot \mathbf{s}_i (\mathrm{mod}\ q)$ on the aggregate message $\sum_{i=1}^{l} c_i \mathbf{m}_i$ is statically close to Gaussian distribution, for example, $\mathbf{s}_{\mathrm{agg}} \sim D_{\mathbf{t} + g\Lambda, \|\mathbf{c}\|\sigma, \mathbf{0}}$, where $\mathbf{t} = \sum_{i=1}^{l} c_i \cdot \mathbf{t}_i$ and $c_i \in \{0, 1\}$. Formally, the distribution of signature on aggregate message only depends on the linear function that was used to compute $\mathbf{m}_{\mathrm{agg}}$ rather than on each original message $\mathbf{m}_i$. Thus, it follows that the aggregate signature does not leak any of the original message except for the aggregate message itself, and Theorem 9 above is true. □

5. Efficiency

In order to analyze the performance, we compare the proposed signatures scheme with previous lattice-based linearly homomorphic signature schemes [5, 7, 16] in terms of public key size, signature length, signing cost, verifying overhead, and multiuser supporting, respectively.

Let T_{ps} denote the time cost to run once *SamplePre* algorithm and let T_{bs} denote the time cost to run once *ExtBasis* algorithm. Since these two algorithms commonly used in lattice-based signature scheme take up most of the time cost throughout the whole signature process, T_{ps} and T_{bs} could be the main indicators in comparison of signing cost. However, the verifying part of schemes mainly involves simple addition and multiplication operations over the modulo, so we use space overhead as the indicator in comparison of the verifying cost. In addition, in the comparison of signature length, for example, $(\mathrm{id}, \mathbf{s})$, the length of id can be ignored because this length is the same for each scheme.

Table 1 shows that Wang's scheme [7] is more efficient than Boneh's scheme [5] in the case of single user. In the case of multiple users, our scheme displays the same efficiency as Wang's in the single user case. Compared with our scheme, the length of signature in Zhang's scheme [16] is twice that of our scheme, and its verifying cost is four times.

6. Conclusions

In this paper, we propose a novel lattice-based HAS scheme with short signature, which is an extension of LHS scheme based on lattice over binary field in the single user case [7]. Our scheme holds both homomorphic property and aggregate property, in which a signed aggregate message can be verified by using the combination of signatures of the original messages and the common public key derived from the public keys of the corresponding users. We prove its security through decreasing to the single user case. At the same time, the "weakly context hiding" property holds in the proposed scheme. Furthermore, it is more practical than the

Table 1: Comparison of several lattice-based linearly homomorphic signature schemes.

Scheme	Multiuser supporting	Public key length	Signature length	Signing cost	Verifying cost (Space overhead)
Boneh's [5]	No	$nm + nm\lg q$	$2m + 2m\lg q$	$T_{\text{ps}} + T_{\text{bs}}$	$nm^2 + 2nm^2\lg q + nm^2(\lg q)^2$
Wang's [7]	No	$nm\lg q$	$m\lg q$	T_{ps}	$nm^2(\lg q)^2$
Zhang's [16]	Yes	$nm\lg q$	$2m\lg q$	$T_{\text{ps}} + T_{\text{bs}}$	$4nm^2(\lg q)^2$
Our scheme	Yes	$nm\lg q$	$m\lg q$	T_{ps}	$nm^2(\lg q)^2$

Zhang's HAS scheme [16]. However, there is still much work to be done in order to improve the capability of the scheme, such as how to design a variant of the scheme with "strong context hiding" property and how to take advantage of "ideal lattice" to decrease the public key size.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research has been supported by the National Natural Science Foundation of China (no. 61374180, 61373136), the Research Foundation for Humanities and Social Sciences of Ministry of Education, China (no. 12YJAZH120, 14YJAZH023), the Research Fund for the Graduate Innovation Program of Jiangsu Province (no. CXZZ13_0493), and the Natural Science Foundation of Universities of Jiangsu province (no. 13KJB520005).

References

[1] R. Johnson, D. Molnar, D. Song, and D. Wagner, "Homomorphic signature schemes," in *Proceedings of the Cryptology (CT-RSA '02)*, pp. 244–262, Springer, Berlin, Germany, 2002.

[2] D. Boneh, D. Freeman, J. Katz, and B. Waters, "Signing a linear subspace: signature schemes for network coding," in *Public Key Cryptography—PKC 2009*, vol. 5443 of *Lecture Notes in Computer Science*, pp. 68–87, Springer, Berlin, Germany, 2009.

[3] R. Gennaro, J. Katz, H. Krawczyk, and T. Rabin, "Secure network coding over the integers," in *Public Key Cryptography—PKC 2010*, vol. 6056 of *Lecture Notes in Computer Science*, pp. 142–160, Springer, Berlin, Germany, 2010.

[4] D. Freeman, "Improved security for linearly homomorphic signatures: a generic framework," in *Public Key Cryptography—PKC 2012*, Lecture Notes in Computer Science, pp. 697–714, Springer, Berlin, Germany, 2012.

[5] D. Boneh and D. M. Freeman, "Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures," in *Public Key Cryptography—PKC*, Lecture Notes in Computer Science, pp. 1–16, Springer, Heidelberg, Germany, 2011.

[6] D. Boneh and D. M. Freeman, "Homomorphic signatures for polynomial functions," in *Advances in cryptology—EUROCRYPT 2011*, vol. 6632 of *Lecture Notes in Computer Science*, pp. 149–168, Springer, Berlin, Germany, 2011.

[7] F. Wang, Y. Hu, and B. Wang, "Lattice-based linearly homomorphic signature scheme over binary field," *Science China Information Sciences*, vol. 56, no. 11, pp. 1–9, 2013.

[8] J. H. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, A. Shelat, and B. Waters, "Computing on authenticated data," in *Theory of Cryptography*, vol. 7194 of *Lecture Notes in Computer Science*, pp. 1–20, Springer, Heidelberg, Germany, 2012.

[9] N. Attrapadung, B. Libert, and T. Peters, "Computing on authenticated data: new privacy definitions and constructions," in *Advances in Cryptology—ASIACRYPT 2012*, vol. 7658 of *Lecture Notes in Computer Science*, pp. 367–385, Springer, Berlin, Germany, 2012.

[10] N. Attrapadung, B. Libert, and T. Peters, "Efficient completely context-hiding quotable and linearly homomorphic signatures," in *Public-Key Cryptography—PKC 2013*, vol. 7778 of *Lecture Notes in Computer Science*, pp. 386–404, Springer, Berlin, Germany, 2013.

[11] R. W. Yeung, "Multi-source network coding," in *Information Theory and Network Coding*, pp. 505–540, Springer, 2008.

[12] Y. Wu, "On constructive multi-source network coding," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '06)*, pp. 1349–1353, Seattle, Wash, USA, July 2006.

[13] S. Agrawal, D. Boneh, X. Boyen, and D. M. Freeman, "Preventing pollution attacks in multi-source network coding," in *Public Key Cryptography—PKC 2010*, vol. 6056 of *Lecture Notes in Computer Science*, pp. 161–176, Springer, Berlin, Germany, 2010.

[14] L. Czap and I. Vajda, "Signatures for Multi-source Network Coding," Cryptology ePrint Archive: Report 2010/328, http://eprint.iacr.org/2010/328.

[15] W. Yan, M. Yang, L. Li, and H. Fang, "Short signature scheme for multi-source network coding," *Computer Communications*, vol. 35, no. 3, pp. 344–351, 2012.

[16] P. Zhang, J. Yu, and T. Wang, "A homomorphic aggregate signature scheme based on lattice," *Chinese Journal of Electronics*, vol. 21, no. 4, pp. 701–704, 2012.

[17] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," *SIAM Journal on Computing*, vol. 37, no. 1, pp. 267–302, 2007.

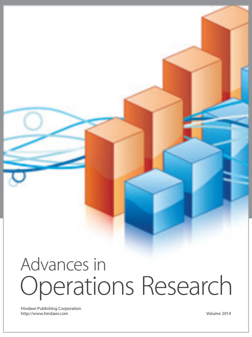
[18] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC '08)*, pp. 197–206, 2008.

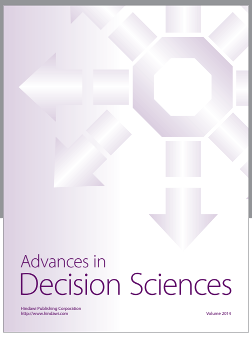
[19] X. Zhao and X. Wang, "An efficient identity-based signcryption from lattice," *International Journal of Security and Its Applications*, vol. 8, no. 2, pp. 363–369, 2014.

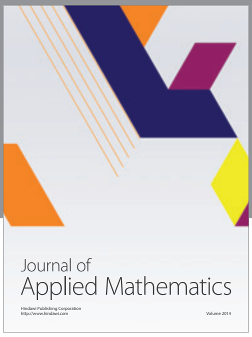
[20] J. Alwen and C. Peikert, "Generating shorter bases for hard random lattices," in *Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science (STACS '09)*, vol. 3, pp. 75–86, 2009.

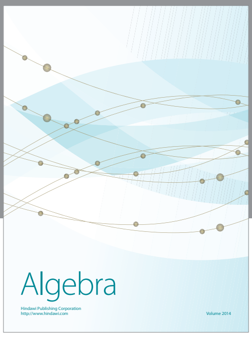
[21] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," in *Advances in Cryptology—EUROCRYPT 2010*, vol. 6110 of *Lecture Notes in Computer Science*, pp. 523–552, Springer, Berlin, Germany, 2010.

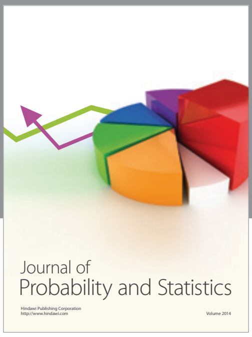
[22] V. Lyubashevsky and D. Micciancio, "Asymptotically efficient lattice-based digital signatures," in *Theory of Cryptography*, vol. 4948 of *Lecture Notes in Computer Science*, pp. 37–54, Springer, Berlin, Germany, 2008.


Advances in
Operations Research

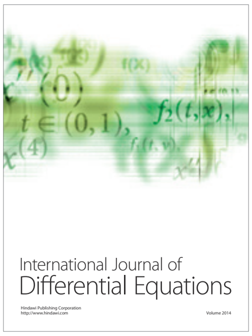

Advances in
Decision Sciences


Journal of
Applied Mathematics

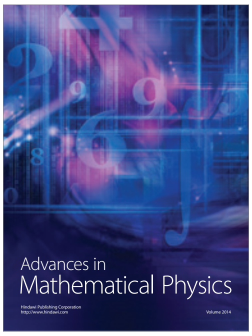

Algebra

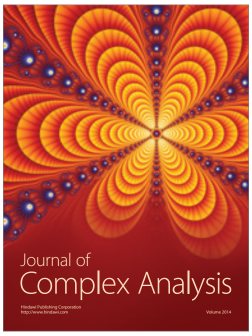

Journal of
Probability and Statistics

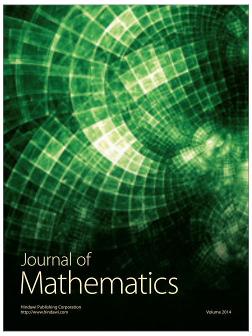
The Scientific
World Journal

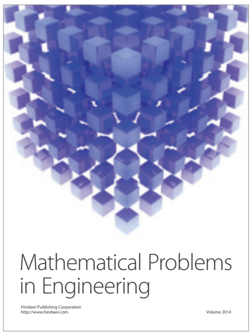

International Journal of
Differential Equations

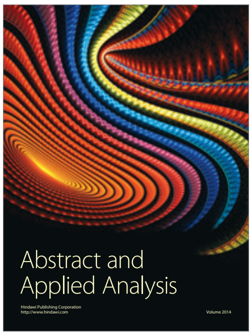
International Journal of
Combinatorics

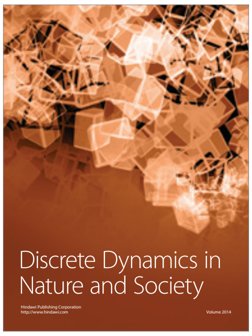

Advances in
Mathematical Physics

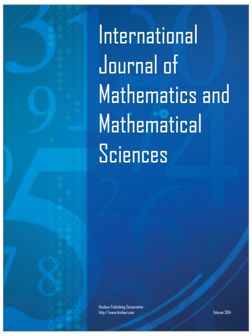

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

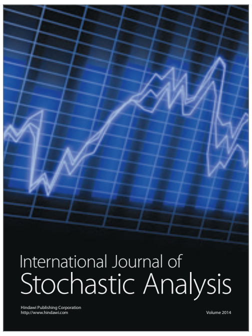

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization