Hindawi
Security and Communication Networks
Volume 2021, Article ID 2459628, 9 pages
https://doi.org/10.1155/2021/2459628

Research Article

Provably Secure Lattice-Based Self-Certified Signature Scheme

Qiang Yang [1,2] **and Daofeng Li** [1,2]

[1]*School of Computer, Electrical and Information, Guangxi University, Nanning 530004, China*
[2]*Guangxi Colleges and Universities Key Laboratory of Multimedia Communications and Information Processing, Guangxi University, Nanning 530004, China*

Correspondence should be addressed to Daofeng Li; ldf_0123@163.com

Received 10 September 2021; Revised 6 December 2021; Accepted 8 December 2021; Published 31 December 2021

Academic Editor: De Rosal Ignatius Moses Setiadi

Digital signatures are crucial network security technologies. However, in traditional public key signature schemes, the certificate management is complicated and the schemes are vulnerable to public key replacement attacks. In order to solve the problems, in this paper, we propose a self-certified signature scheme over lattice. Using the self-certified public key, our scheme allows a user to certify the public key without an extra certificate. It can reduce the communication overhead and computational cost of the signature scheme. Moreover, the lattice helps prevent quantum computing attacks. Then, based on the small integer solution problem, our scheme is provable secure in the random oracle model. Furthermore, compared with the previous self-certified signature schemes, our scheme is more secure.

1. Introduction

Due to the development of the Internet, various networks require to verify the authenticity of the message or the user, such as data verification for Internet of Vehicles [1] and user authentication for industry and mobility networks [2, 3]. In general, these authentication systems can be divided into traditional public key infrastructure- (PKI-) based cryptosystems with certificates [4] and certificateless cryptosystems [5, 6]. In a PKI-based cryptosystem, a certificate issued by the certificate authority (CA) is often required to certify the authenticity of the relationship between the public key and the user. However, it increases the computation overheads and communication costs because certificate management is complicated. Furthermore, it is vulnerable to public key replacement attacks if certificates are not used. To avoid certificate management, in 1984, Shamir [5] introduced the idea of an identity-based (IB) cryptosystem. The public key of a user is the user's identity information. Although it does not require any certificates, it suffers from the key escrow problem.

To deal with the above problems, in 1991, Girault [7] introduced the notion of self-certified public keys, which can enable the public key to implicitly authorize the user's identity without an extra certificate. Concretely, the public key is computed by the user and the authority. Then, the verification of the authenticity of the signature and public key is placed in a logically single step. Thus the self-certified signature scheme can mitigate the burdens for the certificate management and storage and prevent the key escrow problem. Therefore, self-certified signature schemes have a promising future for environments with limited memory and computational capacity, such as smart mobile devices [8], wireless sensor networks [9], the cloud [10], and vehicular ad hoc networks [11].

However, most of them are based on discrete logarithm problems or large integer factorization problems. Unfortunately, Shor [12] pointed out that the two problems are easily solved by quantum computers, so the signature schemes based on the two problems are no longer secure in the quantum era. Lattice cryptography is one of the post-quantum candidate schemes proposed by the National Institute of Standards and Technology [13]. Meanwhile, in recent years, lattice cryptography on the refinements of the security assessment and the fast implementation [14, 15] has achieved rapid developments. Thus, to prevent quantum

attacks, in this paper, based on lattice cryptography, we propose a provably secure self-certified signature scheme in the random oracle model (ROM).

Our contributions are mainly as follows: firstly, our scheme adopts the advantages of self-certified public keys and lattice signature schemes, which simplifies the public key authentication process of the scheme and avoids the key escrow problems and public key replacement attacks. Moreover, it can resist quantum attacks. Secondly, based on the hardness of the small integer solution (SIS) problem, our scheme is existential unforgeability against two types of adversaries in the ROM.

Related work: in PKI-based signature schemes [16], CA issues a certificate for the user, which increases the burden of certificate management and storage.

Many schemes are proposed to reduce the cost of certification management and storage. For instance, numerous IB signature schemes are proposed. In these schemes, the public key is the user's identity. Therefore, it does not need an extra certificate, but it is vulnerable to the key escrow problem because the key generation center (KGC) generates the private keys of all users [17]. Thus, a malicious KGC can impersonate any user. The certificateless public key cryptography [6] also does not require a separate certificate. Recently, Gowri et al. [18] proposed a certificateless signatures scheme from ECC, but later Xu et al. [19] found that it is vulnerable to signature forgery attacks.

To solve the above problems, Girault [7] proposed the self-certified cryptosystem, in which there are no certificates, and neither the user nor the authority can independently obtain the full private key of the user.

Since the self-certificated public key was introduced, many self-certified signature schemes have been proposed. Shao [20] proposed a novel self-certified signature scheme from pairings, but it was later proved to be insecure. In addition, some self-certified signature schemes based on discrete logarithm problems were also proposed, Xie [21] and Wu and Xu [22]. However, Sadeghpour [23] pointed out that Wu and Xu's scheme [22] is vulnerable to internal attacks. Moreover, there exist also several self-certified signature and authentication schemes applied to specific scenarios [9, 11, 24, 25]. Nevertheless, these schemes are not secure against quantum attacks because the hard assumptions are not difficult for quantum computers.

Using the lattice to implement the postquantum self-certified signature scheme is a considerable method. Li et al. [8] first proposed a lattice-based self-certified signatures scheme. However, this scheme is based on NTRUSign, so it lacks rigorous security proof. Moreover, there are no other self-certified signature schemes over lattice. Tian and Huang [26] and others propose several lattice-based certificateless signatures schemes, which do not need the certificate too. However, they cannot prevent insider attack, and there exist other flaws [27, 28]. Hence, in this paper, we aim to propose a provably secure self-certified signature scheme over lattice.

The rest of the paper is organized as follows: in Section 2, we introduce some basic concepts of lattice signature schemes. In Section 3, we introduce the syntax of the self-certified signature scheme and the security model. In Section 4, we introduce our scheme. In Section 5, we analyze the correctness, security, and comparisons. Finally, we give our conclusion and further work.

2. Preliminaries

2.1. Notations. The notations used in this paper are listed as follows:

(1) Let $\mathbb{R}$ be the set of real numbers, $\mathbb{Z}$ the set of integers, and $\mathbb{N}$ the set of nonnegative integer numbers. For a positive integer p, $\mathbb{Z}_p$ is the set of integers modulo p.

(2) Column vectors are written as bold lowercase letters, for example, $\mathbf{v}$. The i th component of $\mathbf{v}$ is represented by $\mathbf{v}_i$, the l_p norm of $\mathbf{v}$ is denoted by $\|\mathbf{v}\|_p$, and define $\|\mathbf{v}\|_p := (\sum \mathbf{v}_i{}^p)^{1/p}$. The l_2 norm of $\mathbf{v}$ can be denoted by $\|\mathbf{v}\|$, and the l_∞ norm of $\mathbf{v}$ is denoted by $\|\mathbf{v}\|_\infty$, and define $\|\mathbf{v}\|_\infty := \max_i\{|\mathbf{v}_i|\}$. Matrices are represented by bold uppercase letters, for example, $\mathbf{B}$. Let the i th column of $\mathbf{B}$ be represented by $\mathbf{b}_i$ and define the norm $\|\mathbf{B}\| = \max_i\|\mathbf{b}_i\|$. If $\mathbf{B}$ is a full rank square matrix, its Gram–Schmidt orthogonalization is denoted by $\widetilde{\mathbf{B}}$, and $\|\widetilde{\mathbf{B}}\| = \max_i\|\mathbf{b}_i\|$. Let $\mathbf{v}^t$ and $\mathbf{B}^t$ denote the transpositions of the vector $\mathbf{v}$ and the matrix $\mathbf{B}$, respectively.

(3) For a real number σ, D_σ denotes the Gaussian distribution with the standard variance σ. Moreover, $\mathbf{a}\leftarrow\mathbf{D}_\sigma^\mathbf{m}$ denotes that each component of the vector a is sampled from D_σ^m. For a random value $\mathbf{v}$ and a set S, $\mathbf{v}\leftarrow S$ denotes that $\mathbf{v}$ is sampled uniform from the set S.

2.2. Lattice

Definition 1 (lattice). A lattice is a discrete subgroup of $\mathbb{R}^\mathbf{n}$. Let $\mathbf{B} = \{\mathbf{b_1}, \mathbf{b_2}, \ldots, \mathbf{b_n}\} \in \mathbb{R}^{\mathbf{n}\times\mathbf{n}}$, where $\mathbf{b_1}, \mathbf{b_2}, \ldots, \mathbf{b_n}$ are $\mathbf{n}$ linear independent vectors. The lattice $\mathscr{L}(\mathbf{B})$ in the $\mathbf{n}$-dimensional Euclidean space generated by the basis $\mathbf{B}$ is defined as $\mathscr{L}(\mathbf{B}) = \mathbf{\Lambda}(\mathbf{B}) = \{\sum_{i=1}^\mathbf{n} \mathbf{c}_i\mathbf{b}_i\colon \mathbf{c}_i \in \mathbb{Z}\}$.

For $q, \mathbf{n}, m \in \mathbb{N}$, let $q \geq 2$ and $\mathbf{A} \in \mathbb{Z}^{\mathbf{n}\times m}$ and define the q-ary lattice as follows:

$$\begin{aligned}\mathbf{\Lambda}^\perp(\mathbf{A}) \text{ or } \mathbf{\Lambda}_q^\perp(\mathbf{A})\colon \mathbf{x} \in \mathbb{Z}^m\colon \mathbf{Ax} = 0 \bmod \mathbf{q},\\ w \in \mathbb{Z}_q^\mathbf{n}, \mathbf{\Lambda}_w^\perp(\mathbf{A})\colon \mathbf{x} \in \mathbb{Z}^m\colon \mathbf{Ax} = w \bmod \mathbf{q}.\end{aligned} \tag{1}$$

The dual lattice of $\mathbf{\Lambda}$ is denoted by $\mathbf{\Lambda}^*$, defined as $\mathbf{\Lambda}^* = \{\mathbf{x} \in \mathbb{R}^\mathbf{n}\colon \forall \mathbf{v} \in \mathbf{\Lambda}, \langle \mathbf{x}, \mathbf{v}\rangle = \mathbb{Z}\}$.

Definition 2 (small integer solution (SIS) problem [29]). For any $\mathbf{n} \in \mathbb{Z}$, given positive integers $q, m \in \mathbb{Z}$, a matrix $\mathbf{A} \in \mathbb{Z}_q^{\mathbf{n}\times\mathbf{m}}$ and $\beta \in \mathbb{R}$, $\mathbf{SIS}_{\mathbf{n},m,q,\beta}$ problem is finding a nonzero integer vector $\mathbf{x} \in \mathbb{Z}^m$ satisfying $\mathbf{Ax} = 0 \bmod \mathbf{q}$ and $\|\mathbf{x}\| \leq \beta$.

Definition 3 (inhomogeneous SIS(ISIS) problem). *For any* $\mathbf{n} \in \mathbb{Z}$, *positive integers* $q, m \in \mathbb{Z}$, $\mathbf{A} \in \mathbb{Z}_q^{\mathbf{n}\times m}$, *and* $\beta \in \mathbb{R}$, $\mathbf{ISIS}_{\mathbf{n},m,q,\beta}$ *problem is defined as follows: given* $\mathbf{y} \in \mathbb{Z}_q^\mathbf{n}$, *find a vector* $\mathbf{x} \in \mathbb{Z}^m$ *such that* $\mathbf{Ax} = \mathbf{y} \bmod \mathbf{q}$ *and* $\|\mathbf{x}\| \leq \boldsymbol{\beta}$.

The hardness of (I)SIS is based on the lattice problems in the worst case [29, 30].

Lemma 1 (hardness). *For any polynomial bounds m, β and a prime number $q \geq \beta\omega(\sqrt{\mathbf{n}\log\mathbf{n}})$, solving (I)SIS problems is as hard as solving GapSVP and SIVP on an arbitrary $\mathbf{n}$-dimensional lattice.*

2.3. Gaussian on Lattices

Definition 4 (Gaussian function). For any real $s > 0$, center $\mathbf{c} \in \mathbb{R}^{\mathbf{n}}$ and define the Gaussian function on $\mathbb{R}^{\mathbf{n}}$ as

$$\forall \mathbf{x} \in \mathbb{R}^{\mathbf{n}}, \boldsymbol{\rho}_{\mathbf{c},s}(\mathbf{x}) = \exp\left(-\pi\|\mathbf{x}-\mathbf{c}\|^2/\mathbf{s}^2\right). \tag{2}$$

For simplicity, when s and $\mathbf{c}$ are taken to be 1 and 0, respectively, they can be omitted.

Discrete Gaussian distribution: the Gaussian distribution over $\mathbb{Z}$ is defined as $\mathbf{D}_{\mathbf{s}}(\mathbf{x}) = \boldsymbol{\rho}_s(\mathbf{x})/\boldsymbol{\rho}_s(\mathbb{Z})$.

For any $s > 0$, $\mathbf{c} \in \mathbb{R}^{\mathbf{n}}$, and $\mathbf{n}$-dimensional lattice $\boldsymbol{\Lambda}$, define the discrete Gaussian distribution over lattice $D_{\boldsymbol{\Lambda}+\mathbf{c},s}$ as

$$\forall \mathbf{x} \in \boldsymbol{\Lambda}+\mathbf{c}, \mathbf{D}_{\boldsymbol{\Lambda}+\mathbf{c},s}(\mathbf{x}) = \boldsymbol{\rho}_s(\mathbf{x})/\boldsymbol{\rho}_{\mathbf{s}}(\boldsymbol{\Lambda}+\mathbf{c}). \tag{3}$$

Definition 5 (smoothing parameter [31]). For a lattice $\boldsymbol{\Lambda}$ and real $\varepsilon > 0$, the smoothing parameter $\eta_\varepsilon(\boldsymbol{\Lambda})$ of the lattice is the smallest real $s > 0$ such that $\boldsymbol{\rho}_{1/s}(\boldsymbol{\Lambda}^* \setminus \{0\}) < \varepsilon$.

For $s \geq 2\eta_\varepsilon(\boldsymbol{\Lambda})$, the Gaussian distribution $D_{\boldsymbol{\Lambda},s}$ is close to a uniform distribution.

Lemma 2 (see[30]). *For the lattice $\boldsymbol{\Lambda}$ with a basis $\mathbf{B}$, let Gaussian parameter $\mathbf{s} \geq \|\widetilde{\mathbf{B}}\| \cdot \boldsymbol{\omega}(\sqrt{\log\mathbf{m}})$, $\mathbf{Pr}[\mathbf{x} \leftarrow \mathbf{D}_{\boldsymbol{\Lambda},\mathbf{s}}: \|\mathbf{x}\| \geq \mathbf{s}\sqrt{\mathbf{m}}] \leq \mathbf{negl}(\mathbf{n})$.*

Lemma 3 (see[30]). *For any $\mathbf{n}$-dimensional lattice $\boldsymbol{\Lambda}$ with basis $\mathbf{B}$ and real $\varepsilon > 0$, there is $\boldsymbol{\eta}_\varepsilon(\boldsymbol{\Lambda}) \leq \|\widetilde{\mathbf{B}}\| \cdot \sqrt{\ln(2\mathbf{n}(1+1/\varepsilon))/\pi}$, where $\|\widetilde{\mathbf{B}}\| = \sqrt{\mathbf{b}^2+1}$ with a base $\mathbf{b}$ and $\eta_\varepsilon(\mathbb{Z}^{\mathbf{n}}) \leq \sqrt{\ln(2\mathbf{n}(1+1/\varepsilon))/\pi}$.*

2.4. Short Bases of Lattice

Definition 6 (gadget-lattice [32]). The gadget matrix $\mathbf{G}$ is defined as $G = I_{\mathbf{n}} \otimes g^t \in \mathbb{Z}^{\mathbf{n}\times\mathbf{nk}}$ with $\mathbf{k}$-dimensional gadget vector $\mathbf{g}^{\mathbf{t}} = (1, \mathbf{b}, \ldots, \mathbf{b}^{\mathbf{k}-1})$, where $\mathbf{k} = \lceil \log_{\mathbf{b}} q \rceil$, and the default value of $\mathbf{b}$ is 2. The q-ary lattice $\boldsymbol{\Lambda}_q^{\perp}(\mathbf{G})$ is the sum of $\mathbf{n}$ copies of the lattice $\boldsymbol{\Lambda}_{\mathbf{q}}^{\perp}(\mathbf{g}^{\mathbf{t}})$. And $\boldsymbol{\Lambda}_u^{\perp}(\mathbf{G}) = \boldsymbol{\Lambda}_{\mathbf{u}_1}^{\perp}(\mathbf{g}^{\mathbf{t}}) \oplus \cdots \oplus \boldsymbol{\Lambda}_{\mathbf{u}_{\mathbf{n}_1}}^{\perp}(\mathbf{g}^{\mathbf{t}})$, where $\boldsymbol{\Lambda}_{\mathbf{u}}^{\perp}(\mathbf{g}^{\mathbf{t}}) = \{\mathbf{x} \in \mathbb{Z}^{\mathbf{k}}: g^t\mathbf{x} = u \bmod q\} = \boldsymbol{\Lambda}_{\mathbf{q}}^{\perp}(\mathbf{g}^t) + \mathbf{u}$ and $\mathbf{u}_i \in \mathbb{Z}_{\mathbf{q}}$.

The (I)SIS problems are easily solved on the gadget matrix G.

Definition 7 (G-Trapdoor [32]). For the matrices $\mathbf{A} \in \mathbb{Z}_q^{\mathbf{n}\times m}$ and $\mathbf{G} \in \mathbb{Z}_q^{\mathbf{n}\times w}$, where $m = O(\mathbf{n}\log q)$, $w = \mathbf{nk}$, $\mathbf{k} = \lceil \log_2 q \rceil$, and $m \geq w \geq \mathbf{n}$. The $\mathbf{G}$-Trapdoor of $\mathbf{A}$ is a matrix $\mathbf{R} \in \mathbb{Z}^{(m-w)\times w}$ such that $\mathbf{A}\begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix} = \mathbf{HG}$ for $\mathbf{H} \in \mathbb{Z}_q^{\mathbf{n}\times\mathbf{n}}$.

Here, $\mathbf{H}$ is an invertible matrix. The quality of the trapdoor is determined by its maximum singular value $\mathbf{s_1}(\mathbf{R})$.

The most efficient trapdoor generation algorithm now is the G-TrapGen [32].

Lemma 4 (G-TrapGen [32]). *Let $\overline{m} = m - w$ and $\overline{\mathbf{A}} \in \mathbb{Z}_q^{\mathbf{n}\times\overline{m}}$, and set $\mathbf{H} = I$. There is a probabilistic polynomial time (PPT) algorithm that outputs a parity-check matrix $\mathbf{A} = [\overline{\mathbf{A}}|\mathbf{G} - \overline{\mathbf{A}}\mathbf{R}] \in \mathbb{Z}_{\mathbf{q}}^{\mathbf{n}\times\mathbf{m}}$ and the trapdoor $\mathbf{R} \in \mathbb{Z}^{\overline{m}\times w}$.*

The matrix $\mathbf{A}$ is statistically close to uniformly random. The quality of the trapdoor is $s_1(\mathbf{R}) \leq \sqrt{m} \cdot \omega(\sqrt{\log\mathbf{n}})$.

Lemma 5 (G-Trapdoor SamlePre [32]). *For the matrix-trapdoor pair $(\mathbf{A}, \mathbf{R}) \in \mathbb{Z}_q^{\mathbf{n}\times m} \times \mathbb{Z}^{\overline{m}\times w}$, a positive definite matrix is defined as $\boldsymbol{\Sigma}_{\mathbf{p}} := \mathbf{s}^2\mathbf{I} - \sigma^2\begin{bmatrix}\mathbf{RR^t} & \mathbf{R^t}\\ \mathbf{R} & \mathbf{I}\end{bmatrix}$, where $\sigma \geq \eta_\varepsilon(\boldsymbol{\Lambda}_q^{\perp}(\mathbf{G}))$ and $\mathbf{s} \approx \mathbf{s}_1(\mathbf{R}) \cdot \sigma^2$. Given a vector $\mathbf{u} \in \mathbb{Z}^m$, there is a PPT algorithm that outputs the preimage $\mathbf{y} = \mathbf{p} + \begin{bmatrix}\mathbf{R}\\ \mathbf{I}\end{bmatrix}\mathbf{z} \in \mathbb{Z}^{\mathbf{m}}$ where $\mathbf{p} \leftarrow D_{\mathbb{Z}^m, \sqrt{\Sigma_p}}$ is a perturbation and $\mathbf{z} \leftarrow \mathbf{D}_{\boldsymbol{\Lambda}_{\mathbf{v}}^{\perp}(G),\sigma}$, such that $\mathbf{v} = \mathbf{u} - \mathbf{Ap} \in \mathbb{Z}_{\mathbf{q}}^{\mathbf{n}}$.*

Lemma 6 (SampleMat [26]). *Let the integers $\mathbf{n} \geq 1$, $q \geq 2$, $m = O(\mathbf{n}\log q)$, given a matrix $\mathbf{A} \in \mathbb{Z}^{\mathbf{n}\times m}$ and the trapdoor $\mathbf{R}$, for $\mathbf{U} \in \mathbb{Z}^{\mathbf{n}\times\mathbf{k}}$ and the Gaussian parameter $s = O(\sqrt{\mathbf{n}\log q})$. There is a PPT algorithm SampleMAT $(\mathbf{A}, \mathbf{R}, \mathbf{s}, \mathbf{U})$ that outputs the preimage $\mathbf{X} \in \mathbb{Z}^{\mathbf{m}\times\mathbf{k}}$ such that $\mathbf{AX} = \mathbf{U}$ and $\|\mathbf{X}\| \leq \mathbf{s}\sqrt{\mathbf{m}}$.*

2.5. Rejection Sampling Technique

Lemma 7 (see [31]). *For any Gaussian parameter $\sigma > 0$ and positive integer m,*

$$\begin{aligned}\Pr\left[\mathbf{x} \leftarrow D_\sigma^1: |\mathbf{x}| > 12\sigma\right] &< 2^{-100},\\ \Pr\left[\mathbf{x} \leftarrow \mathbf{D}_\sigma^m: \|\mathbf{x}\| > 2\sigma\sqrt{m}\right] &< 2^{-m}.\end{aligned} \tag{4}$$

Lemma 8 (see [33]). *For any $\mathbf{c} \in \mathbb{Z}^m$, positive real α and $\sigma = \omega(\|\mathbf{c}\|\sqrt{\log m})$, $\mathbf{x} \leftarrow D_\sigma^m$, we have*

$$\mathbf{Pr}\left[\mathbf{D}_{\boldsymbol{\sigma}}^{\mathbf{m}}(\mathbf{x})/\mathbf{D}_{\mathbf{c},\boldsymbol{\sigma}}^{\mathbf{m}}(\mathbf{x}) = \mathbf{O}(1)\right] < 1 - 2^{\boldsymbol{\omega}(\sqrt{\log\mathbf{m}})}, \tag{5}$$

and more specifically, if $\sigma = \alpha\|\mathbf{c}\|$, then

$$\mathbf{Pr}\left[D_\sigma^m(\mathbf{x})/\mathbf{D}_{\mathbf{c},\sigma}^m(\mathbf{x}) < e^{12/\boldsymbol{\alpha}+1/2\boldsymbol{\alpha}^2}\right] > 1 - 2^{-100}. \tag{6}$$

The rejection sampling technique ensures that the distribution of the outputted signature is independent of the

signing key so that a valid signature is generated without leaking any useful information about the key.

Concretely, given the distribution D_σ^m, the signing key $\mathbf{S}$, and a message μ, first sample $\mathbf{y} \leftarrow \mathbf{D}_\sigma^m$, then compute $\mathbf{c} = \mathbf{H}(y, \mu)$ and $\mathbf{z} = \mathbf{Sc} + \mathbf{y}$, and the signature is $(\mathbf{z}, \mathbf{c})$. Here, we want to obtain $\mathbf{z}$ sampled from D_σ^m instead of $D_{\mathbf{Sc},\sigma}^m$, where $D_{\mathbf{Sc},\sigma}^m$ is the distribution from the shift of the distribution D_σ^m by an offset vector $\mathbf{Sc}$. Therefore, we select an appropriate value M and output $\mathbf{z}$ with the probability $\min(1, \mathbf{D}_\sigma^{\mathbf{m}}(\mathbf{z})/M\mathbf{D}_{\mathbf{Sc},\sigma}^{\mathbf{m}}(\mathbf{z}))$, so that the distribution of $\mathbf{z}$ is statistically indistinguishable from the distribution D_σ^m.

3. Self-Certified Signature Scheme

3.1. The Syntax of the SCS. A self-certified signature (SCS) scheme consists of the following algorithms: Setup, KeyGen, Extract, Sign, and Verify:

(1) **Setup**: it takes a security parameter $\mathbf{n}$ as input and returns the system parameter pp.

(2) **KeyGen**: CA selects the master private key s and generates its public key $\mathbf{P}_{\mathrm{CA}}$.

(3) **Extract**: each user $\mathbf{U}_i$ first selects his private key and the partial public key Y_{id} and then sends Y_{id} to CA. After receiving the request, CA extracts the partial private key s_{id} of the user. Therefore, the full public key is $(\mathbf{P}_{\mathrm{CA}}, \mathrm{ID}, Y_{\mathrm{id}})$, and the full private key is $(\mathbf{x}_{\mathrm{id}}, s_{\mathrm{id}})$.

(4) **Sign**: the user generates a signature sig of the message m with the private key $(\mathbf{x}_{\mathrm{id}}, s_{\mathrm{id}})$.

(5) **Verify**: a verifier checks the signature sig.

3.2. Security Model. In this section, we give the security model of the SCS scheme.

SCS schemes are secure against two types of adversaries, which are classified as external and internal adversaries as follows.

Type 1: Adversary (Outsider). A type 1 adversary knows the secret value of any user by listening to the public channel or replacing the public key.

Type 2: Adversary (Honest-But-Curious CA). A type 2 adversary can compute the partial private key of any user, but it does not know the user's secret value.

Definition 8 (type 1 attack). A SCS scheme is existentially unforgeable against adaptive chosen message type 1 attacks if no polynomial bounded type 1 adversary $\mathscr{A}_1$ with a nonnegligible advantage wins the following game.

Setup: the challenger $\mathscr{C}$ takes a security parameter as input and runs the setup and the KeyGen algorithms. It gives the system parameters and CA's master public key to the adversary $\mathscr{A}_1$ and keeps the master secrete key secret.

Queries: $\mathscr{A}_1$ makes following adaptive queries:

(i) Hash queries: given any $M \in \{0,1\}^*$, $\mathscr{C}$ returns the hash value $H(M)$ to $\mathscr{A}_1$.

(ii) Secret key queries: given a user's identity ID $\in \{0,1\}^*$, $\mathscr{C}$ returns the user's secret key to $\mathscr{A}_1$.

(iii) Partial private key queries: given a user's identity ID, $\mathscr{C}$ returns the user's partial private key to $\mathscr{A}_1$.

(iv) Public key queries: given a user's identity ID, $\mathscr{C}$ returns the user's public key to $\mathscr{A}_1$.

(v) Public key replacement queries: given a user's identity ID and a public key pk^*, $\mathscr{C}$ replaces the user's public key with pk^*.

(vi) Sign queries: given a message m, $\mathscr{C}$ returns a signature σ of m to $\mathscr{A}_1$.

(vii) Verify queries: given a signature σ, $\mathscr{C}$ responds the verification result to $\mathscr{A}_1$.

Forgery: $\mathscr{A}_1$ outputs a new signature σ^* for a message m^*, and $\mathscr{A}_1$ wins the game if the outputted signature σ^* is valid and without making a partial private key query or a sign query for the message m^*.

Definition 9. (type 2 attack). A SCS scheme is existentially unforgeable under adaptive chosen message type 2 attacks if no polynomial bounded type 2 adversary $\mathscr{A}_2$ with a nonnegligible advantage wins the following game.

Setup: the challenger $\mathscr{C}$ takes a security parameter as input and runs the setup and the KeyGen algorithms. It gives the system parameters and CA's master public key to the adversary $\mathscr{A}_2$ and keeps the master secrete key secret.

Queries: $\mathscr{A}_2$ makes following adaptive queries:

(i) Hash queries: given any $M \in \{0,1\}^*$, $\mathscr{C}$ returns the hash value $H(M)$ to $\mathscr{A}_2$.

(ii) Secret key queries: given a user's identity ID, $\mathscr{C}$ returns the user's secret key to $\mathscr{A}_2$.

(iii) Public key queries: given a user's identity ID, $\mathscr{C}$ returns the user's public key to $\mathscr{A}_2$.

(vi) Sign queries: given a message m, $\mathscr{C}$ returns a signature σ of m to $\mathscr{A}_2$.

(v) Verify queries: given a signature σ, $\mathscr{C}$ returns the verification result to $\mathscr{A}_2$.

Forgery: $\mathscr{A}_2$ outputs a new signature σ^* for a message m^*, and $\mathscr{A}_2$ wins the game if the outputted signature σ^* is valid and without making a secret key query or a sign query for the message m^*.

Definition 10 (unforgeability). A SCS scheme is secure if it is existentially unforgeable under adaptive chosen attacks; namely, the advantages that the adversaries $\mathscr{A}_1$ and $\mathscr{A}_2$ successfully forge a valid signature are negligible.

4. Our Signature Scheme

Our scheme consists of five algorithms: Setup, KeyGen, Extract, Sign, and Verify.

(1) Setup: given security parameters $\lambda, \mathbf{n}$, for positive integers $q, m, d, \mathbf{k}, \kappa$, and $\sigma, s \in \mathbb{R}$. Let $q \geq 2$, $m = O$

$(\mathbf{n}\log q) > 64 + \mathbf{n}\log q/\log(2d+1)$, $d \ll q^{\mathbf{n}/m}$, $2^{\kappa}\binom{\mathbf{n}}{\kappa} \geq 2^{\lambda}$, $s = O(\sqrt{\mathbf{n}\log q})$, $\sigma = 12s\kappa\sqrt{2m}$, and select two secure hash functions: $H_1\colon \{0,1\}^* \longrightarrow \mathbb{Z}_q^{\mathbf{n}\times\mathbf{k}}$, $H_2\colon \{0,1\}^* \longrightarrow \{\mathbf{v}\colon \mathbf{v} \in \{-1,0,1\}^{\mathbf{k}}, \|\mathbf{v}\|_1 \leq \kappa\}$. Finally, publish system parameters $\text{Para} = \{\lambda, \mathbf{n}, q, m, d, \mathbf{k}, \kappa, s, \sigma, H_1, H_2\}$.

(2) KeyGen: the KeyGen algorithm is described in Algorithm 1. CA runs the algorithm G-TrapGen $(\mathbf{n})$ to output the public-private key pair. The master private key is the trapdoor $\mathbf{R}$, and the public key is the public matrix $\mathbf{A}$.

(3) Extract: the extract algorithm is described in Algorithm 2. The user's public-private key pair is generated by the user and CA. User_{ID} first selects his secret key $\mathbf{B}_{\text{ID}}$ and computes the partial public key $\mathbf{P}_{\text{ID}}$. CA generates the partial private key $\mathbf{X}_{\text{ID}}$ and sends it to the user through a secure channel. Therefore, the full private key is $(\mathbf{B_{ID}}, \mathbf{X_{ID}})$ and the full public key is the $(\mathbf{A}, \mathbf{P_{ID}}, \text{ID})$.

(4) Sign: the Sign algorithm is described in Algorithm 3. User_{ID} generates a signature of the message μ using the rejection sampling technique.

Remark 1. According to the rejection sampling technique described in Section 2, at most M attempts, we will output a signature such that the distribution of $\mathbf{z}$ is statistically close to D_σ^{2m}, and we have $\Pr[\|\mathbf{z}\| \leq 2\sigma\sqrt{2m}] \geq 1 - 2^{-2m}$.

(5) Verify: the KeyGen algorithm is described in Algorithm 4. The verifier verifies the signature $(\mathbf{z}, \mathbf{c})$ of the message μ on ID.

5. Analysis

5.1. Correctness. The correctness of the scheme is as follows:

First, we have $\mathbf{AX_{ID}} = \mathbf{U}$, $\mathbf{BB_{ID}} = \mathbf{P_{ID}}$, so

$$\begin{aligned} & H_2(\mathbf{Az}_1 + \mathbf{Bz}_2 - \mathbf{P_{ID}c} - \mathbf{Uc}, \mu), \\ &= H_2(\mathbf{A}(\mathbf{X_{ID}c} + \mathbf{y}_1) + \mathbf{B}(\mathbf{B_{ID}c} + \mathbf{y}_2) - \mathbf{P_{ID}c} - \mathbf{Uc}, \mu), \\ &= H_2(\mathbf{AX_{ID}c} + \mathbf{A}y_1 + \mathbf{BB_{ID}c} + \mathbf{B}y_2 - \mathbf{P_{ID}c} - \mathbf{Uc}, \mu), \\ &= H_2((\mathbf{AX_{ID}} - \mathbf{U} + \mathbf{BB_{ID}} - \mathbf{P_{ID}})\mathbf{c} + \mathbf{A}y_1 + \mathbf{B}y_2, \mu), \\ &= H_2(\mathbf{A}y_1 + \mathbf{B}y_2, \mu), \\ &= \mathbf{c}. \end{aligned} \tag{7}$$

At this moment, we complete the proof.

5.2. Security Analysis. Our scheme is existentially unforgeable under the adaptive chosen message attacks in the random oracle model.

Theorem 1. *Our SCS scheme is existentially unforgeable in ROM for a polynomial-time type 1 adversary. If the adversary can successfully forge a valid signature, then it can solve the* $SIS_{\mathbf{n},m,q,\beta}$ *problem, where* $\beta = 2(2\sigma + s\kappa)\sqrt{m}$.

Proof. Assume that there is a type 1 adversary $\mathcal{A}$ who can break the scheme with nonnegligible probability. Then, we can construct a polynomial-time challenger $\mathcal{C}$, who runs $\mathcal{A}$ as a subroutine to solve the $\text{SIS}_{\mathbf{n},m,q,\beta}$ problem with nonnegligible probability; that is, $\mathcal{C}$ wins Game 1:

Game 1 setup: input the security parameter $\mathbf{n}$. $\mathcal{C}$ runs the setup and the KeyGen algorithm to obtain $\text{Para} = \{\lambda, \mathbf{n}, q, m, d, \mathbf{k}, \kappa, s, \sigma, H_1, H_2\}$ and $(\mathbf{A}, \mathbf{R})$. Then, $\mathcal{C}$ publishes $\mathbf{P}ara$ and $\mathbf{A}$ and keeps $\mathbf{R}$ secret. $\mathcal{C}$ maintains several initially empty lists: List 0, List 1, List 2, List 3, and List 4. List 0 contains $(\text{ID}_i, \mathbf{P_{ID}}, \mathbf{U} = H_1(\mathbf{A}, \mathbf{P}_{\text{ID}}, \text{ID}_i))$, where ID_i is the user's identity and $\mathbf{P}_{\text{ID}}$ is the partial public key. List 1 contains $(\text{ID}_i, \mathbf{B_{ID}}, \mathbf{P_{ID}}, \mathbf{U})$, where $\mathbf{B}_{\text{ID}}$ is the secret key. List 2 contains $(\text{ID}_i, \mathbf{B}_{\text{ID}}, \mathbf{P}_{\text{ID}}, \mathbf{U}, \mathbf{X})$, where $\mathbf{X}$ is the partial private key. List 3 contains $(\mu, \mathbf{y}_1, \mathbf{y}_2, \mathbf{c} = H_2(\cdot))$, where $\mathbf{y}_1, \mathbf{y}_2$ are two random vectors. List 4 contains $(\text{ID}_i, \mu, (\mathbf{z}, \mathbf{c}))$, where $(\mathbf{z}, \mathbf{c})$ is the signature.

Queries: $\mathcal{A}$ adaptively issues several queries to $\mathcal{C}$:

(i) H_1 *queries*: $\mathcal{A}$ sends a user's identity ID_i to $\mathcal{C}$; then, $\mathcal{C}$ performs as follows:

(1) First looks up ID_i in List 0. If found, $\mathcal{C}$ directly returns the hash value $\mathbf{U}$ of the public key.
(2) Otherwise, $\mathcal{C}$ randomly selects a matrix $\mathbf{U} \in \mathbb{Z}_q^{m\times\mathbf{k}}$ and returns it; then, it selects a matrix $\mathbf{P}_{\text{ID}} \in \mathbb{Z}_q^{m\times\mathbf{k}}$, and adds $(\text{ID}_i, \mathbf{P_{ID}}, \mathbf{U})$ to List 0.

(ii) H_2 queries: $\mathcal{A}$ sends a message μ to $\mathcal{C}$; then, $\mathcal{C}$ performs as follows:

(1) First, it looks up μ in List 3. If found, $\mathcal{C}$ directly returns the hash value $\mathbf{c}$ of message μ.
(2) Otherwise, $\mathcal{C}$ randomly selects a vector $\mathbf{c}$ from $\{\mathbf{c}\colon \mathbf{c} \in \{-1,0,1\}^{\mathbf{k}}, \|\mathbf{c}\|_1 \leq \kappa\}$ and returns it. Then, it randomly selects two vectors $\mathbf{y}_1, \mathbf{y}_2 \leftarrow D_\sigma^m$ and adds $(\mu, \mathbf{y}_1, \mathbf{y}_2, \mathbf{c})$ to List 3.

(iii) Secret matrix queries: $\mathcal{A}$ sends an identity ID_i to $\mathcal{C}$; then, $\mathcal{C}$ performs as follows:

(1) First, it looks up ID_i in List 1. If found, $\mathcal{C}$ directly returns the user's secret matrix $\mathbf{B}_{\text{ID}}$.
(2) Otherwise, $\mathcal{C}$ selects a matrix $\mathbf{B}_{\text{ID}} \in \mathbb{Z}^{m\times\mathbf{k}}$ from $\{-d, \ldots, 0, \ldots, d\}^{m\times\mathbf{k}}$ and returns it. Then, it computes $\mathbf{P}_{\text{ID}} = \mathbf{BB}_{\text{ID}}$, $\mathbf{U} = H_1(\mathbf{A}, \mathbf{P}_{\text{ID}}, \text{ID}_i)$ and adds $(\text{ID}_i, \mathbf{B}_{\text{ID}}, \mathbf{P}_{\text{ID}}, \mathbf{U})$ to List 1.

(vi) Partial private key queries: $\mathcal{A}$ sends ID_i to $\mathcal{C}$; then, $\mathcal{C}$ performs as follows:

(1) First, it looks up ID_i in List 2. If found, $\mathcal{C}$ directly returns the partial private key $\mathbf{X}$.
(2) Otherwise, $\mathcal{C}$ issues a secret matrix query to obtain $(\mathbf{B}_{\text{ID}}, \mathbf{P}_{\text{ID}}, \mathbf{U})$ and runs the algorithm SampleMAT $(\mathbf{A}, \mathbf{R}, \mathbf{s}, \mathbf{U})$ to output a matrix $\mathbf{X} \in \mathbb{Z}^{m\times\mathbf{k}}$ as the partial private key.
(3) Finally, $\mathcal{C}$ returns $\mathbf{X}$ and adds $(\text{ID}_i, \mathbf{B}_{\text{ID}}, \mathbf{P}_{\text{ID}}, \mathbf{U}, \mathbf{X})$ to List 2.

(v) Public key replacement queries: $\mathcal{A}$ sends ID_i and a public key $\mathbf{P}_{\text{ID}}^*$ to $\mathcal{C}$ and then wants to replace the user's public key. After receiving the identity ID_i, $\mathcal{C}$ replaces the public key with $\mathbf{P}_{\text{ID}}^*$ and records this replacement.

Input: security parameter n.
Output: $(\mathbf{A}, \mathbf{R})$.
(1) Output $(\mathbf{A}, \mathbf{R}) \leftarrow$ G-TrapGen (n), $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ is the master public key and $R \in \mathbb{Z}^{2n\times n \log q}$ is the master private key.
(2) Choose a random uniform matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$.
(3) Publish $\mathbf{A}$ and $\mathbf{B}$, and keep $\mathbf{R}$ secret.

Algorithm 1: KeyGen.

Input: User_{ID} with an identity $\text{ID} \in \{0,1\}^*$, $\mathbf{B} \in \mathbb{Z}_{\mathbf{q}}^{\mathbf{n}\times\mathbf{m}}$.
Output: the user's key $(\mathbf{P_{ID}}, \mathbf{B_{ID}}, \mathbf{X_{ID}})$.
(1) $\text{User}_{I\,D}$ selects a random uniform matrix $B_{\mathbf{ID}} \in \mathbb{Z}^{m\times k}$ from $\{-d, \ldots, 0 \cdots, d\}^{m\times k}$ as the secret key, where $\|B_{\mathbf{ID}}\| \le d\sqrt{m}$.
(2) Computes $\mathbf{P_{ID}} = \mathbf{BB_{ID}} \bmod q$, and sends $(\mathbf{P_{ID}}, \text{ID})$ to CA.
(3) Upon receiving the $(\mathbf{P}_{\text{ID}}, \text{ID})$, CA performs as follows:
(i) computes $\mathbf{U} = \mathbf{H}_1(\mathbf{A}, \mathbf{P_{ID}}, \mathbf{ID}) \in \mathbb{Z}_q^{n\times k}$.
(ii) uses the algorithm SampleMAT $(\mathbf{A}, \mathbf{R}, \mathbf{s}, \mathbf{U})$ to output $\mathbf{X_{ID}} \in \mathbb{Z}^{m\times k}$.
(iii) sends $\mathbf{X_{ID}}$ to User_{ID} through a secure channel.
(4) $\text{User}_{I\,D}$ checks the authenticity of $\mathbf{X_{ID}}$ by verifying $\mathbf{AX_{ID}} = \mathbf{U}$ and $\|\mathbf{X_{ID}}\| \le \mathbf{s}\sqrt{\mathbf{m}}$. If so, output X_{ID} as their partial private key. Otherwise, reject it.

Algorithm 2: Extract.

Input: a message $\mu \in \{0,1\}^*$, ID, the signing key $(\mathbf{B_{ID}}, \mathbf{X_{ID}})$ and other public parameters.
Output: the signature $(\mathbf{z}, \mathbf{c})$.
(1) Randomly select $\mathbf{y}_1, \mathbf{y}_2 \leftarrow \mathbf{D}_\sigma^m$, and compute $\mathbf{c} = \mathbf{H}_2(\mathbf{Ay}_1 + \mathbf{By}_2, \mu)$.
(2) Compute $\mathbf{z}_1 = \mathbf{X}_{\text{ID}}\mathbf{c} + \mathbf{y}_1$, $\mathbf{z}_2 = \mathbf{B_{ID}c} + \mathbf{y}_2$, and let $\mathbf{z} = [\mathbf{z}_1^t, \mathbf{z}_2^{\mathbf{t}}]^{\mathbf{t}}$.
(3) Output the signature $(\mathbf{z}, \mathbf{c})$ with probability $\min(1, D_\sigma^{2m}(\mathbf{z})/\text{MD}_{v,\sigma}^{2m}(\mathbf{z}))$, where $v = [(\mathbf{X_{ID}c})^t, (\mathbf{B_{ID}c})^t]^t$.
(4) If $\|\mathbf{z}\| > 2\sigma\sqrt{2m}$, repeat this algorithm.

Algorithm 3: Sign.

Input: $(\mathbf{A}, \mathbf{B}, \mathbf{P_{ID}})$ and ID, the message μ, the signature (z, c).
Output: the verification result.
(1) Compute $\mathbf{U} = \mathbf{H}_1(\mathbf{A}, \mathbf{P_{ID}}, \text{ID})$.
(2) Check whether $\|\mathbf{z}\| \le 2\sigma\sqrt{2m}$ and $\mathbf{c} = \mathbf{H}_2(\mathbf{Az}_1 + \mathbf{Bz}_2 - \mathbf{P_{ID}c} - \mathbf{Uc}, \boldsymbol{\mu})$.
(3) If so, return accept. Otherwise, return reject.

Algorithm 4:Verify.

(vi) Sign queries: $\mathscr{A}$ sends a message μ, ID_i, and a secret matrix $\mathbf{B}_{\text{ID}}$ to $\mathscr{C}$. Then, $\mathscr{C}$ performs as follows:
(1) First, it looks up the parameters in List 4. If found, $\mathscr{C}$ directly returns the signature.
(2) Otherwise, $\mathscr{C}$ issues a partial private key query to obtain the signing key $(\mathbf{X}, \mathbf{B_{ID}})$, issues a H_2 query to obtain $\mathbf{c}$, and computes $\mathbf{z}_1 = \mathbf{Xc} + \mathbf{y}_1$, $\mathbf{z}_2 = \mathbf{B_{ID}c} + \mathbf{y}_2$.
(3) Finally, $\mathscr{C}$ returns the signature $(\mathbf{z}, \mathbf{c})$, where $\mathbf{z} = [\mathbf{z}_1^t, \mathbf{z}_2^{\mathbf{t}}]^{\mathbf{t}}$, and adds $(\text{ID}_i, \mu, (\mathbf{z}, \mathbf{c}))$ to List 4.

Forgery: after polynomial-time queries finish, $\mathscr{A}$ outputs a forgery $(\mathbf{z}^*, \mathbf{c}^*)$ on message μ^* for ID_i with nonnegligible probability. If the signature can pass the verification and partial private key queries and the sign queries for the message μ^* are never involved in this game, then $\mathscr{A}$ wins the game.

Using the forking lemma [34], $\mathscr{C}$ replays $\mathscr{A}$ with different hash values of H_2 queries to get another valid signature $(\mathbf{z}', \mathbf{c}')$ such that $\mathbf{c}^* \ne \mathbf{c}'$ and $\mathbf{H}_2(\mathbf{Az}_1^* + \mathbf{Bz}_2^* - \mathbf{P}_{\text{ID}}\mathbf{c}^* - \mathbf{Uc}^*, \mu^*) = H_2(\mathbf{Az}_1' + \mathbf{Bz}_2' - \mathbf{P_{ID}c}' - \mathbf{Uc}', \mu^*)$.

Because $\mathscr{A}$ is a type 1 adversary that can make public key replacement attacks, it is easy to obtain $\mathbf{P}_{\mathrm{ID}}$ and $\mathbf{B}_{\mathrm{ID}}$. Then, only the partial private key $\mathbf{X}$ needs to be considered. Since $\mathbf{U} = \mathbf{AX}$, we have the equality:

$$\begin{aligned}\mathbf{Az}_1^* - \mathbf{Uc}^* &= \mathbf{Az}'_1 - \mathbf{Uc}',\\ \mathbf{Az}_1^* - \mathbf{AXc}^* &= \mathbf{Az}'_1 - \mathbf{AXc}',\\ &\mathbf{A}\left(\mathbf{z}_1^* - \mathbf{z}_1' + \mathbf{Xc}' - \mathbf{Xc}^*\right).\end{aligned} \tag{8}$$

As $\|\mathbf{z}_1^*\| \le 2\sigma\sqrt{m}$, $\|\mathbf{z}_1'\| \le 2\sigma\sqrt{m}$, $\|\mathbf{Xc}^*\| \le s\kappa\sqrt{m}$, $\|\mathbf{Xc}'\| \le s\kappa\sqrt{m}$ with overwhelming probability, we can see that

$$\left\|\mathbf{z}_1^* - \mathbf{z}_1' + \mathbf{Xc}' - \mathbf{Xc}^*\right\| \le 2(2\sigma + s\kappa)\sqrt{m}. \tag{9}$$

Now, we prove $\mathbf{z}_1^* - \mathbf{z}_1' + \mathbf{Xc}' - \mathbf{Xc}^* \neq 0$ with non-negligible probability.

Since $m > 64 + \mathbf{n}\log q/\log(12s)$, then, by Lemma 4.2 from [33], there is another $\mathbf{X}'$ with probability larger than $1 - 2^{-100}$ such that all the columns except for the i th column are the same as $\mathbf{X}$. And $\mathbf{AX} = \mathbf{AX}' = \mathbf{U}$. So, if $\mathbf{z}_1^* - \mathbf{z}_1' + \mathbf{Xc}' - \mathbf{Xc}^* = 0$, then we have $\mathbf{z}_1^* - \mathbf{z}_1' + \mathbf{X}'\mathbf{c}' - \mathbf{X}'\mathbf{c}^* \neq 0$. Because $\mathbf{X}$ and $\mathbf{X}'$ have the same role in our scheme, $\mathscr{C}$ does not know which one is used in the simulation. Hence, $\mathscr{C}$ can make $\mathbf{z}_1^* - \mathbf{z}_1' + \mathbf{Xc}' - \mathbf{Xc}^* \neq 0$ with the probability not less than 0.5. As a result, we can construct an algorithm to solve the $\mathbf{SIS}_{\mathbf{n},m,q,\beta}$ problem with overwhelming probability, where $\beta = 2(2\sigma + s\kappa)\sqrt{m}$. □

Theorem 2. *Our SCS scheme is existentially unforgeable in ROM for a polynomial-time type 2 adversary. If the adversary can successfully forge a valid signature, then it can solve the* $SIS_{\mathbf{n},m,q,\beta}$ *problem, where* $\beta = 2(2\sigma + d\kappa)\sqrt{m}$.

Proof. Assume that there is a type 2 adversary $\mathscr{A}$ that can break out the scheme with nonnegligible probability. Then, we can construct a polynomial-time challenger $\mathscr{C}$ that runs $\mathscr{A}$ as a subroutine to solve the $\mathrm{SIS}_{\mathbf{n},m,q,\beta}$ problem with nonnegligible probability, that is, $\mathscr{C}$ wins Game 2:

Game 2 Setup: input the security parameter $\mathbf{n}$. $\mathscr{C}$ runs the setup and KeyGen algorithm to obtain Para $= \{\lambda, \mathbf{n}, q, m, d, \mathbf{k}, \kappa, s, \sigma, H_1, H_2\}$ and $(\mathbf{A}, \mathbf{R})$. Then, $\mathscr{C}$ publishes Para and $\mathbf{A}$ and keeps $\mathbf{R}$ secret. $\mathscr{C}$ maintains several initially empty lists: List 0, List 1, List 3, and List 4.

Queries: $\mathscr{A}$ adaptively issues several queries to $\mathscr{C}$:

(i) H_1 queries: it is the same as the above proof.

(ii) H_2 queries: it is the same as the above proof.

(iii) Secret matrix queries: $\mathscr{A}$ sends a user's identity ID_i to $\mathscr{C}$; then, $\mathscr{C}$ performs as follows:

(1) First, it looks up ID_i in List 1. If found, $\mathscr{C}$ directly returns the user's secret matrix $\mathbf{B}_{\mathrm{ID}}$.

(2) Otherwise, $\mathscr{C}$ randomly selects a matrix $\mathbf{B}_{\mathrm{ID}} \in \mathbb{Z}^{m\times\mathbf{k}}$ from $\{-d, \ldots, 0, \ldots, d\}^{m\times\mathbf{k}}$, returns it, then computes $\mathbf{P}_{\mathbf{ID}} = \mathbf{BB}_{\mathbf{ID}}$, $\mathbf{U} = H_1(\mathbf{A}, \mathbf{P}_{\mathbf{ID}}, \mathrm{ID}_i)$, and adds $(\mathrm{ID}_i, \mathbf{B}_{\mathbf{ID}}, \mathbf{P}_{\mathbf{ID}}, \mathbf{U})$ to List 1.

(vi) Sign queries: $\mathscr{A}$ sends ID_i and a message μ to $\mathscr{C}$; then, $\mathscr{C}$ performs as follows:

(1) First, it looks up the parameters in List 4. If found, $\mathscr{C}$ directly returns the signature.

(2) Otherwise, $\mathscr{C}$ issues a secret matrix query to obtain $\mathbf{B}_{\mathrm{ID}}$, issues a H_1 query to obtain the public key, and then computes $\mathbf{X}$. $\mathscr{C}$ also issues a H_2 query to obtain $\mathbf{c}$ and computes $\mathbf{z}_1 = \mathbf{Xc} + y_1, \mathbf{z}_2 = \mathbf{B}_{\mathbf{ID}}\mathbf{c} + \mathbf{y}_2$.

(3) Finally, $\mathscr{C}$ returns the signature $(\mathbf{z}, \mathbf{c})$, where $\mathbf{z} = [\mathbf{z}_1^t, \mathbf{z}_2^t]^t$, and adds $(\mathrm{ID}_i, \mu, (\mathbf{z}, \mathbf{c}))$ to List 4.

Forgery: after polynomial-time queries finish, $\mathscr{A}$ outputs a forgery $(\mathbf{z}^*, \mathbf{c}^*)$ on message μ^* for ID_i with nonnegligible probability. If the signature can pass the verification, and the secret matrix queries and the sign queries for the message μ^* are never involved in this game, then $\mathscr{A}$ wins the game.

Using the forking lemma, $\mathscr{C}$ replays $\mathscr{A}$ with different hash values of H_2 queries to get another valid signature $(\mathbf{z}', \mathbf{c}')$ such that $\mathbf{c}^* \neq \mathbf{c}'$ and $H_2(\mathrm{Az}_1^* + \mathrm{Bz}_2^* - \mathbf{P}_{\mathrm{ID}}\mathbf{c}^* - \mathbf{Uc}^*, \mu^*) = H_2(\mathbf{Az}'_1 + \mathbf{Bz}'_2 - \mathbf{P}_{\mathbf{ID}}\mathbf{c}' - \mathbf{Uc}', \mu^*)$.

Because $\mathscr{A}$ is a type 2 adversary that can obtain the partial private key $\mathbf{X}_{\mathrm{ID}}$, only the secret key $\mathbf{B}_{\mathrm{ID}}$ needs to be considered. Moreover, since $\mathbf{P}_{\mathrm{ID}} = \mathrm{BB}_{\mathrm{ID}}$, we have the equality

$$\begin{aligned}\mathbf{Bz}_2^* - \mathbf{P}_{\mathbf{ID}}\mathbf{c}^* &= \mathbf{Bz}'_2 - \mathbf{P}_{\mathbf{ID}}\mathbf{c}',\\ \mathbf{Bz}_2^* - \mathrm{BB}_{\mathbf{ID}}\mathbf{c}^* &= \mathbf{Bz}'_2 - \mathbf{BB}_{\mathbf{ID}}\mathbf{c}',\\ \mathbf{B}\left(\mathbf{z}_2^* - \mathbf{z}_2' + \mathbf{B}_{\mathbf{ID}}\mathbf{c}' - \mathbf{B}_{\mathbf{ID}}\mathbf{c}^*\right) &= 0.\end{aligned} \tag{10}$$

As $\|\mathbf{z}_2^*\| \le 2\sigma\sqrt{m}$, $\|\mathbf{z}_2'\| \le 2\sigma\sqrt{m}$, $\|\mathbf{B}_{\mathbf{ID}}\mathbf{c}^*\| \le d\kappa\sqrt{m}$, $\|\mathbf{B}_{\mathbf{ID}}\mathbf{c}'\| \le d\kappa\sqrt{m}$ with overwhelming probability, we can see that

$$\left\|\mathbf{z}_2^* - \mathbf{z}_2' + \mathbf{B}_{\mathrm{ID}}\mathbf{c}' - \mathbf{B}_{\mathrm{ID}}\mathbf{c}^*\right\| \le 2(2\sigma + d\kappa)\sqrt{m}. \tag{11}$$

Now, we prove $\mathbf{z}_2^* - \mathbf{z}_2' + \mathbf{B}_{\mathbf{ID}}\mathbf{c}' - \mathbf{B}_{\mathrm{ID}}\mathbf{c}^* \neq 0$ with non-negligible probability.

Since $m > 64 + \mathbf{n}\log q/\log(2\ d + 1)$ and according to Lemma 4.2 from [33], there is another secret key $\mathbf{B}'_{\mathrm{ID}}$ with the probability larger than $1 - 2^{-100}$ such that all columns except for the i th column are the same as $\mathbf{B}_{\mathrm{ID}}$. And $\mathbf{P}_{\mathbf{ID}} = \mathrm{BB}'_{\mathbf{ID}} = \mathrm{BB}_{\mathbf{ID}}$. So, if $\mathbf{z}_2^* - \mathbf{z}_2' + \mathbf{B}_{\mathbf{ID}}\mathbf{c}' - \mathbf{B}_{\mathbf{ID}}\mathbf{c}^* = 0$, then we know $\mathbf{z}_2^* - \mathbf{z}_2' + \mathbf{B}'_{\mathbf{ID}}\mathbf{c}' - \mathbf{B}'_{\mathbf{ID}}\mathbf{c}^* \neq 0$. Because $\mathbf{B}_{\mathrm{ID}}$ and $\mathbf{B}'_{\mathrm{ID}}$ have the same role in our scheme, the challenger $\mathscr{C}$ does not know which one is used in the simulation. Therefore, $\mathscr{C}$ can make $\mathbf{z}_2^* - \mathbf{z}_2' + \mathbf{B}_{\mathbf{ID}}\mathbf{c}' - \mathbf{B}_{\mathbf{ID}}\mathbf{c}^* \neq 0$ with probability not less than 0.5. Finally, we can construct an algorithm to solve the $\mathrm{SIS}_{\mathbf{n},m,q,\beta}$ problem with overwhelming probability, where $\beta = 2(2\sigma + d\kappa)\sqrt{m}$. □

5.3. Comparisons. In this section, we compare our scheme with several typical self-certified authentication or signature schemes. We mainly focus on the computational costs, the storage overheads, and several security properties.

As in Table 1, we compare the computational costs. Let t_{bp}, t_{bpm}, t_{eccm}, t_{mul}, and t_h denote the execution times of a bilinear pairing operation, a scale multiplication for the bilinear pairing, a scale multiplication for the ECC, a polynomial

Table 1: The comparisons of computational costs.

Scheme	Signing cost	Verify cost
Li et al.'s scheme [10]	$t_{\text{bpm}} + t_h$	$3t_{\text{bp}} + t_h$
Tahat et al.'s scheme [35]	$4t_{\text{eccm}} + 3t_h$	$2t_{\text{eccm}} + 3t_h$
Li et al.'s scheme [8]	$7t_{\text{mul}} + t_h$	$t_{\text{mul}} + t_h$
Our scheme	$4nt_{\text{mul}} + t_h$	$4nt_{\text{mul}} + 2t_h$

Table 2: The comparisons of storage overheads.

Scheme	Public key size	Private key size	Signature length
Li et al. [10]	$\|G_1\|$	$\|G_1\|$	$2\|G_1\|$
Tahat et al. [35]	$2\|G\|$	$\|G\|$	$3\|G\|$
Li et al. [8]	$n \log q$	$4n \log q$	$n \log q$
Our	$nk \log q$	$mk \log(2s) + mk \log(2\ d + 1)$	$2m \log(12\sigma)$

Table 3: The comparisons of security properties.

Scheme	Provable security	Assumption	Postquantum
Li et al. [10]	Yes	Bilinear pairings	No
Tahat et al. [35]	No	ECDLP	No
Li et al. [8]	No	NTRU CVP	Yes
Our	Yes	Lattice SIS	Yes

multiplication, and a general hash function operation. Although our scheme has higher computational costs, based on the lattice, our scheme provides more robust security.

As depicted in Table 2, we compare the storage overheads. For the key size, we use an improved trapdoor generation algorithm [32] based on GPV [30] to reduce the dimensionality of the trapdoor. For the signature length, we only use Gaussian sampling in the key extract phase but use the rejection sampling technique in the signature generation phase, which helps to reduce the signature length. Moreover, according to Lemma 2 in [14], if the key is distributed as a discrete Gaussian distribution with the parameter σ such that $\|\mathbf{x}\| \le s\sqrt{m}$, the bit size of $\mathbf{x}$ is bounded by m log 2 s bits.

In Table 3, we compare the security properties. According to the above tables, Li et al. [10] and Tahat et al.'s [35] schemes are more efficient than the SCS schemes over lattice, where $|G| = 320$ bits and $|G_1| = 1024$ bits. However, their schemes are not secure against quantum attacks because the security is based on the pairing or elliptic curve discrete logarithm problems (ECDLP). Li et al.'s [8] scheme is the first SCS scheme over lattice. It has a shorter key size and signature length because of using NTRU lattice, but it lacks provable security. Our scheme is based on the standard lattice, so the key size and the signature length are less efficient than Li et al.'s scheme. However, our scheme is provably secure in the ROM under the SIS assumption. Therefore, our scheme is more secure against quantum computers.

6. Conclusion and Further Work

In this paper, we propose a self-certified signature scheme over the standard lattice, which authenticates the integrity of the message and the user's public key and identity without the need for additional certificates, thus not only avoiding the key escrow problems and public key replacement attacks but also preventing quantum attacks. Based on the hardness of the SIS assumption, our scheme is provably secure in the random oracle model. Our scheme is more feasible than previous schemes.

Future work: we consider the standard model. The standard model is more secure and practical than the random oracle model. Hence, our further work is to transfer our scheme into a SCS scheme in the standard model. Furthermore, the efficiency of the scheme can be further improved.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (no. 61662004).

References

[1] Qi Jiang, X. Zhang, N. Zhang, Y. Tiane, X. Maa, and J. Ma, "Three-factor authentication protocol using physical unclonable function for iov," *Computer Communications*, vol. 173, pp. 45–55, 2021.

[2] C. Wang, D. Wang, G. Xu, and D. He, "Efficient privacy-preserving user authentication scheme with forward secrecy for industry 4.0," *Science China Information Sciences*, vol. 65, no. 1, p. 112301, 2021.

[3] S. Qiu and D. Wang, "Revisiting three anonymous two-factor authentication schemes for roaming service in global mobility networks," *Journal of Surveillance, Security and Safety*, vol. 2, no. 2, pp. 66–82, 2021.

[4] A. Boldyreva, M. Fischlin, A. Palacio, and B. Warinschi, "A closer look at PKI: security and efficiency," in *Proceedings of the 10th International Conference on Practice and Theory in Public-Key Cryptography*, T. Okamoto and X. Wang, Eds., pp. 458–475, Springer, Beijing, China, April 2007.

[5] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of the Advances in Cryptology, Proceedings of CRYPTO '84*, pp. 47–53, Springer, Santa Barbara, California, USA, August 1984.

[6] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proceedings of the 9th International Conference on the Theory and Application of Cryptology and Information Security*, C.-S. Laih, Ed., pp. 452–473, Springer, Taipei, Taiwan, November-December 2003.

[7] M. Girault, "Self-certified public keys," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*, pp. 490–497, Springer, Brighton, UK, April 1991.

[8] D. Li, H. Chen, C. Zhong, T. Li, and F. Wang, "A new self-certified signature scheme based on ntrusing for smart mobile communications," *Wireless Personal Communications*, vol. 96, no. 3, pp. 4263–4278, 2017.

[9] Y. Chen and J. Chen, "Anonymous and provably secure authentication protocol using self-certified cryptography for wireless sensor networks," *Multimedia Tools and Applications*, vol. 80, no. 10, pp. 15291–15313, 2021.

[10] H. Li, F. Guo, L. Wang, J. Wang, B. Wang, and C. Wu, "A blockchain-based public auditing protocol with self-certified public keys for cloud data," *Security and Communication Networks*, vol. 2021, no. 1, pp. 6623639–10, 2021.

[11] X. Li, Y. Han, J. Gao, and J. Niu, "Secure hierarchical authentication protocol in VANET," *IET Information Security*, vol. 14, no. 1, pp. 99–110, 2020.

[12] P. W. Shor, *Polynomial-time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, https://arxiv.org/abs/quant-ph/9508027, 1997.

[13] L. Ducas, E. Kiltz, T. Lepoint et al., "Crystals-dilithium: a lattice-based digital signature scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 1, pp. 238–268, 2018.

[14] R. El Bansarkhani and J. Buchmann, "Improvement and efficient implementation of a lattice-based signature scheme," in *Proceedings of the Selected Areas in Cryptography - Sac 2013*, vol. 8282, pp. 48–67, Burnaby, BC, Canada, August 2013.

[15] N. Genise and D. Micciancio, "Faster Gaussian sampling for trapdoor lattices with arbitrary modulus," in *Proceedings of the 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, N. Jesper Buus and R. Vincent, Eds., pp. 174–203, Springer, Tel Aviv, Israel, April-May 2018.

[16] M. N. S. Perera and T. Koshiba, "Fully dynamic group signature scheme with member registration and verifier-local revocation," in *Proceedings of the ICMC: International Conference on Mathematics and Computing*, D. Ghosh, D. Giri, R. N. Mohapatra, K. Sakurai, E. Savas, and T. Som, Eds., pp. 399–415, Springer Singapore, Varanasi India, January 2018.

[17] W. Shen, J. Qin, J. Yu, R. Hao, and J. Hu, "Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 331–346, 2019.

[18] T. Gowri, G. Srinivasa Rao, P. Vasudeva Reddy, N. B. Gayathri, and D. V. Rama Koti Reddy, "Efficient pairing-free certificateless signature scheme for secure communication in resource-constrained devices," *IEEE Communications Letters*, vol. 24, no. 8, pp. 1641–1645, 2020.

[19] Z. Xu, M. Luo, M. K. Khan, K.-K. R. Choo, and D. He, "Analysis and improvement of a certificateless signature scheme for resource-constrained scenarios," *IEEE Communications Letters*, vol. 25, no. 4, pp. 1074–1078, 2021.

[20] Z. Shao, "Self-certified signature scheme from pairings," *Journal of Systems and Software*, vol. 80, no. 3, pp. 388–395, 2007.

[21] X. Qi, "Provably secure self-certified multi-proxy signature with message recovery," *Journal of Networks*, vol. 7, no. 10, pp. 1616–1623, 2012.

[22] F. Wu and L. Xu, "An improved and provable self-certified digital signature scheme with message recovery," *International Journal of Communication Systems*, vol. 28, no. 2, pp. 344–357, 2015.

[23] M. Sadeghpour, "Cryptanalysis and modification of an improved self- certified digital signature scheme with message recovery," *AMERICAN SCIENTIFIC RESEARCH JOURNAL FOR ENGINEERING, TECHNOLOGY, AND SCIENCES (ASRJETS)*, vol. 28, pp. 257–265, 2017.

[24] B. Ying and A. Nayak, "Lightweight remote user authentication protocol for multi-server 5g networks using self-certified public key cryptography," *Journal of Network and Computer Applications*, vol. 131, pp. 66–74, 2019.

[25] B. A. Alzahrani, S. Ashraf Chaudhry, B. Ahmed, A. Al-Barakati, and T. Shon, "An anonymous device to device authentication protocol using ecc and self certified public keys usable in internet of things based autonomous devices," *Electronics*, vol. 9, no. 3, 2020.

[26] M. Tian and L. Huang, "Certificateless and certificate-based signatures from lattices," *Security and Communication Networks*, vol. 8, no. 8, pp. 1575–1586, 2015.

[27] S. Chang, H. Sook Lee, J. Lee, and S. Lim, "Security analysis of a certificateless signature from lattices," *Security and Communication Networks*, vol. 2017, no. 1, pp. 3413567–7, 2017.

[28] K.-A. Shim, "Security vulnerabilities of four signature schemes from NTRU lattices and pairings," *IEEE Access*, vol. 8, pp. 85019–85026, 2020.

[29] M. Ajtai, "Generating hard instances of lattice problems (extended abstract)," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC '96*, pp. 99–108, Association for Computing Machinery, New York, NY, USA, July 1996.

[30] G. Craig, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 197–206, Association for Computing Machinery, New York, NY, USA, May 2008.

[31] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 372–381, Rome, Italy, October 2004.

[32] D. Micciancio and C. Peikert, "Trapdoors for lattices: simpler, tighter, faster, smaller," in *Proceedings of the 31st Annual International Conference on Theory and Applications of Cryptographic Techniques, EUROCRYPT'12*, pp. 700–718, Springer-Verlag, Cambridge, UK, April 2012.

[33] V. Lyubashevsky, "Lattice signatures without trapdoors," in *Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 738–755, Cambridge, UK, April 2012.

[34] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology*, vol. 13, no. 3, pp. 361–396, 2000.

[35] N. Tahat, A. K. Alomari, O. M. Al-Hazaimeh, and M. F. Al-Jamal, "An efficient self-certified multi-proxy signature scheme based on elliptic curve discrete logarithm problem," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 23, no. 4, pp. 935–948, 2020.