symmetry

MDPI

Article

Lattice-Based Revocable Certificateless Signature

Ying-Hao Hung, Yuh-Min Tseng * and Sen-Shan Huang

Department of Mathematics, National Changhua University of Education, Jin-De Campus, Chang-Hua 500, Taiwan; hungyinghao@gmail.com (Y.-H.H.); sshuang@cc.ncue.edu.tw (S.-S.H.)

* Correspondence: ymtseng@cc.ncue.edu.tw; Tel.: +886-4-723-2105 (ext. 3216)

Received: 27 September 2017; Accepted: 18 October 2017; Published: 20 October 2017

Abstract: Certificateless signatures (CLS) are noticeable because they may resolve the key escrow problem in ID-based signatures and break away the management problem regarding certificate in conventional signatures. However, the security of the mostly previous CLS schemes relies on the difficulty of solving discrete logarithm or large integer factorization problems. These two problems would be solved by quantum computers in the future so that the signature schemes based on them will also become insecure. For post-quantum cryptography, lattice-based cryptography is significant due to its efficiency and security. However, no study on addressing the revocation problem in the existing lattice-based CLS schemes is presented. In this paper, we focus on the revocation issue and present the first revocable CLS (RCLS) scheme over lattices. Based on the short integer solution (SIS) assumption over lattices, the proposed lattice-based RCLS scheme is shown to be existential unforgeability against adaptive chosen message attacks. By performance analysis and comparisons, the proposed lattice-based RCLS scheme is better than the previously proposed lattice-based CLS scheme, in terms of private key size, signature length and the revocation mechanism.

Keywords: cryptography; lattice; certificateless signature; short integer solution (SIS); assumption; post-quantum cryptography

1. Introduction

Identity (ID)-based public-key cryptography (ID-PKC) was introduced by Shamir [1] to break away the requirement of certificates in conventional public-key cryptography (PKC). In ID-PKC, the public key of a user is decided by his/her associated identity information, such as e-mail address, telephone number, social security number, and so on. With the public key of a user, a trusted third party (called private key generator (PKG) produces the user's associated private key and sends it to the user via a secure channel. Thus, the legitimacy of public keys can be verified publicly. Boneh and Franklin [2] employed Shamir's concept to construct a workable ID-based encryption (IBE) scheme using bilinear maps such as Ate, Tate, and Weil pairings. Since the PKG knows all the users' private keys, the PKG may impersonate all the users to forge a signature on any message and encrypt any ciphertext. In such a case, all ID-based cryptographic schemes have the key escrow problem.

In 1993, certificateless public-key cryptography (CL-PKC) was introduced by Al-Riyami and Paterson [3] to simultaneously repeal the use of certificates in conventional PKC and resolve the key escrow problem in ID-PKC. They concretely presented a certificateless signature (CLS) and a certificateless public-key encryption (CL-PKE) scheme. In CL-PKC, the private key of a user includes two parts, namely, a secret value and a partial private key. The secret value is randomly selected by the user while the partial private key is generated with her/his identity by a key generation center (KGC). Hence, the KGC does not know a user's private key so that the key escrow problem occurred in ID-PKC is avoided. In addition, the user independently generates and publishes the public key, so the need of certificates in conventional PKC is abolished. Afterwards, numerous works [4–9] have addressed the CL-PKC area.

That are several cases that request a user's public key to be invalidated before its preplanned expiration time. Thus, a public-key setting should offer a revocation method to cancel compromised or illegal users from the system. Tseng and Tsai [10] presented the revocation method using public channel. In addition, two primitives (encryption and signature) in revocable certificateless public-key cryptography (RCL-PKC) were also proposed, such as revocable certificateless public-key encryption (RCL-PKE) schemes [11,12] and revocable certificateless signature (RCLS) scheme [13,14]. Furthermore, Hung et al. [15] presented a short RCLS scheme.

Indeed, the security of these conventional PKC, ID-PKC, CL-PKC and RCL-PKC mentioned above rely on the difficulty of solving the discrete logarithm or integer factorization problems. However, when quantum computers come into reality, both hard problems would become easy to compute [16] so that those cryptographic schemes based on them would become insecure. Whereas, several new mathematical methods for cryptography have been constructed to resist quantum attacks. For post-quantum cryptography, lattice-based cryptography is significant [17] because no efficient quantum algorithm can solve the related problems that include the short integer solution (SIS) and short independent vector problem (SIVP) problems over lattices. Moreover, lattice-based cryptography is more efficient than other post-quantum cryptographies.

1.1. Related Work

For lattice-based cryptography, Goldreicha et al. [18] proposed lattice-based signature and public-key encryption schemes under the conventional PKC settings. Unfortunately, its signature scheme was completely broken in [19]. Afterward, several famous signature schemes were presented, including Gentry et al.'s scheme [20] and Lyubashevsky's schemes [21,22]. The former is provably secure. In their scheme, Gentry et al. employed the Gaussian sampling and the hash-and-sign techniques, respectively, to produce users' private keys and signatures. However, the private key is lengthy while the hash-and-sign technique is inefficient. Lyubashevsky [21] employed the Fiat-Shamir transformation technique to propose an efficient lattice-based signature scheme while its security is based on the short integer solution problem (SIS) over lattices. The Fiat-Shamir transformation turns out to be more efficient than the hash-and-sign technique when generating a signature. Moreover, to improve the efficiency further, Lyubashevsky [22] proposed another lattice-based signature scheme, which employed the rejection sampling technique to produce the signature. Lyubashevsky's second scheme is simple and needs just a few matrix-vector multiplications and rejection samplings.

To combine the advantages of ID-PKC and lattice-based cryptography, Ruckert [23] presented two ID-based signature (IBS) schemes over lattice assumptions. One was shown to be secure in the standard model and the other is secure in the random oracle model. The framework of Ruckert's scheme followed Gentry et al.'s scheme [20]. Therefore, the private key and the signature remain lengthy. Then, several lattice-based IBS schemes [24–26] were presented to enhance the security and efficiency. Recently, Xiang [27] adopted the binary tree structure used in [28] to construct a revocable IBS (RIBS) scheme over lattices. To improve the efficiency, Hung et al. [29] furthermore presented a new lattice-based RIBS. Their scheme adopted the NTRU lattice in [26] to produce the private key of a user. Therefore, the private key size and signature size are shorter than those of Xiang's scheme.

In the past, the study of the lattice-based certificateless signature (CLS) schemes received little attention. Tian and Huang [30] proposed the first lattice-based CLS scheme. Since they adopted the GPV lattice in [20] to generate the private key of a user, it is of the form $(\mathbf{S}_1, \mathbf{S}_2)$, where $\mathbf{S}_1$ is an $m_1 \times k$ matrix and $\mathbf{S}_2$ is an $m_2 \times k$ matrix, with $m_1, m_2 > 5klogq$ and q being a prime. However, the private key above turns out to be lengthy, so is the associated signature. Therefore, their scheme is inefficient and impractical. Moreover, no study on addressing the revocation problem in the existing lattice-based CLS schemes is presented.

1.2. Contribution and Organization

In this paper, we focus on the revocation issue and present the first revocable CLS (RCLS) scheme over lattices while improving the performance of Tian and Huang's CLS scheme [30] mentioned above. Our RCLS scheme provides a revocation method using public channel to cancel compromised or illegal users. The revocation method follows the revocation concept of our previous literature [10]. In our RCLS scheme, a user's private key consists of three parts that include a secret value, a time update key and a partial private key. The secret value is randomly selected by the user while the partial private key is generated with her/his identity by a key generation center (KGC). The point is that the time update key is changed along with time period and the KGC periodically sends new time update keys to non-revoked users via a public channel. If the KGC would like to cancel compromised or illegal users, the KGC just stops generating the new time update keys of these users. In our RCLS scheme, the partial private key is generated by using the key extract algorithm of Ducas et al.'s ID-based encryption over lattices [26]. In the key extract phase, Ducas et al. adopted a particular sampling algorithm to improve Gentry et al.'s key extract algorithm [20] by producing short trapdoor (private key). Meanwhile, in our signing phase, we adopt the rejection sampling technique in [22] to produce a signature. Therefore, our lattice-based RCLS has shorter private key size and signature length than others. Relied on the difficulty of solving the short integer solution (SIS) problem [31], we show that the proposed lattice-based RCLS scheme offers existential unforgeability against adaptive chosen-message attacks for three adversaries that include Type I adversary (outsider), Type II adversary (honest-but-curious KGC) and Type III adversary (revoked user). When compared with the previously proposed lattice-based CLS scheme, the proposed lattice-based RCLS scheme possesses better security and similar efficiency.

The rest of the paper is arranged as follows. In Section 2, preliminaries are presented. The framework and security model of RCLS schemes are given in Section 3. The proposed lattice-based RCLS scheme is presented in Section 4. In Section 5, the security analysis of our scheme is demonstrated. Comparisons are presented in Section 6. Conclusions are drawn in Section 7.

2. Preliminaries

2.1. Notations

Throughout this paper, we denote several parameters as follows:

- N: a power-of-two integer.
- $\mathbb{R}$: the set of real numbers.
- $\mathbb{Z}$: the set of integers.
- Z_q for a $q > 0$: the interval be the set of integers with $[-q/2,\ q/2)$.
- $R_q = Z_q[X]/(X^N+1)$: a ring of polynomials modulo X^N+1 with coefficients in Z_q.

For a vector $\mathbf{x}$ and a matrix $\mathbf{X}$, $\|\mathbf{x}\| = \sqrt{\sum \mathbf{x}_i^2}$ and $\|\mathbf{X}\|_\infty = \max[\|\mathbf{X}_i\|]$, respectively, denote the Euclidean norm of $\mathbf{x}$ and the longest norm of all columns of $\mathbf{X}$. Let $f = \sum_{i=0}^{N-1} f_i x^i$ and $g = \sum_{i=0}^{N-1} g_i x^i$ be two polynomials in R_q.

For a set S, the notation $y \leftarrow S$ denotes that y is uniformly selected at random from S. For a distribution D, $z \leftarrow D$ means that z is selected according to the distribution D.

2.2. Anticirculant Matrices

Anticirculant matrices have a special structure and useful properties. An N-dimensional anticirculant matrix $\mathbf{C}_N(f)$ is defined as follows.

Definition 1. $\mathbf{C}_N(f)$ *is a Toeplitz matrix represented by*

$$\mathbf{C}_N(f)=\begin{bmatrix}(f)\\(x\cdot f)\\\vdots\\(x^{N-1}\cdot f)\end{bmatrix}=\begin{bmatrix}f_0 & f_1 & \cdots & f_{N-2} & f_{N-1}\\-f_{N-1} & f_0 & \cdots & f_{N-3} & f_{N-2}\\\cdots & \cdots & \cdots & \cdots & \cdots\\-f_1 & -f_2 & \cdots & -f_{N-1} & f_0\end{bmatrix},$$

where $f=\sum_{i=o}^{N-1} f_i x_i \in R_q$.

For convenience, $\mathbf{C}_N(f)$ is abbreviated as $\mathbf{C}(f)$ in the sequel. Anticirculant matrices have the following nice property.

Lemma 1. *If* $f,g\in R_q$, *we have* $\mathbf{C}(f)*\mathbf{C}(g)=\mathbf{C}(f*g)$ *and* $\mathbf{C}(f)+\mathbf{C}(g)=\mathbf{C}(f+g)$ [26].

2.3. Lattice and NTRU Lattice

Here, we briefly define a lattice and an NTRU lattice. A lattice is a full-rank discrete subgroup of R^n. And an NTRU lattice comes from a particular class of convolution modular lattices. The detailed definitions are given below.

Definition 2. *Let n vectors* $\mathbf{v}_1,\mathbf{v}_2,\ \ldots,\mathbf{v}_n$ *be linearly independent and* $\mathbf{B}=\{\mathbf{v}_1,\ \ldots,\mathbf{v}_n\}$ *be the basis of the n-dimensional lattice* $\boldsymbol{\Lambda}$. *The lattice* $\boldsymbol{\Lambda}$ *produced by the basis* $\mathbf{B}$ *is presented as*

$$\boldsymbol{\Lambda}=L(\mathbf{v}_1,\ \ldots,\mathbf{v}_n)=\{\sum_{i=1}^{n} x_i\mathbf{v}_i:\ x_i\in\mathbb{R}^n\}.$$

Definition 3. *Let* $h=g*f^{-1}$, *where* $f,g\in R_q$. *The NTRU lattice* $\boldsymbol{\Lambda}_{h,q}$ *associated with h and a positive integer q is a full-rank lattice of* $\mathbb{Z}^{2N}$ *and is represented as*

$$\boldsymbol{\Lambda}_{h,q}=\{(u,\ v)\in R_q^2\mid u+v*h=0\}.$$

Indeed, $\boldsymbol{\Lambda}_{h,q}$ is produced by the rows of

$$\mathbf{A}_{h,q}=\begin{bmatrix}-\mathbf{C}(h) & \mathbf{I}_N\\ q\mathbf{I}_N & \mathbf{O}_N\end{bmatrix},$$

where $\mathbf{I}_N$ is the $N\times N$ unit matrix, $\mathbf{O}_N$ and $\mathbf{C}(h)$, respectively, denote the $N\times N$ null matrix and an N-dimensional anticirculant matrix with h. If h is uniformly distributed in R_q, the basis $\boldsymbol{\Lambda}_{h,q}$ is not suitable to solve the usual lattice problems. Hence Hoffstein et al. [32] remedied this situation by constructing another appropriate basis for $\boldsymbol{\Lambda}_{h,q}$, namely,

$$\mathbf{B}_{f,g}=\begin{bmatrix}\mathbf{C}(g) & -\mathbf{C}(f)\\ \mathbf{C}(G) & -\mathbf{C}(F)\end{bmatrix},$$

where $F,G\in R_q$ such that $f*G-g*F=q$.

Indeed, we can efficiently find F and G. By the following lemma, $\mathbf{B}_{f,g}$ is called the *short basis* for $\boldsymbol{\Lambda}_{h,q}$ due to the fact $\|\mathbf{B}_{f,g}\|\ \le\ \|\mathbf{A}_{h,q}\|$.

Lemma 2. *Let* $f, g \in R_q$ *and* $h = g * f^{-1}$, *and let* $F, G \in R_q$ *satisfy the equality* $f * G - g * F = q$ [26]. *Then,* $\mathbf{B}_{f,g}$ *generates the same NTRU lattice* $\mathbf{\Lambda}_{h,q}$ *as* $\mathbf{A}_{h,q}$ *does and* $\|\mathbf{B}_{f,g}\| \leq \|\mathbf{A}_{h,q}\|$.

Lemma 3. *Given a prime q, a power-of-two integer N and* $\sigma = 1.17\sqrt{q/(2N)}$, *there exists a probabilistic polynomial-time (PPT) algorithm* ***TrapGen****(q, N) that can produce a pair of polynomials f and g and then computes* $h = g * f^{-1}$, *and outputs a trapdoor matrix* $\mathbf{B}_{f,g}$ *as a short basis of* $\mathbf{\Lambda}_{h,q}$. *Here, h is published publicly and is statistically close to be uniform in* R_q [26].

2.4. Gaussian Distribution

Here, we present the definitions of the continuous and discrete Gaussian distributions, which are useful tools in lattice-based cryptography.

Definition 4. *The continuous Gaussian distribution over* $\mathbb{R}^N$ *with the center* $\mathbf{c} \in \mathbb{R}^N$ *and the standard deviation s > 0, is defined as*

$$\rho_{\mathbf{c},s}^N(\mathbf{x}) = \left(\frac{1}{s\sqrt{2\pi}}\right)^N e^{\frac{-\|\mathbf{x}-\mathbf{c}\|^2}{2s^2}}, \text{ where } \mathbf{x} \in \mathbb{R}^N.$$

We scale this distribution for any lattice $\Lambda \in \mathbb{R}^N$ by $\rho_{\mathbf{c},s}^N(\mathbf{\Lambda}) = \sum_{\mathbf{x} \in \mathbf{\Lambda}} \rho_{\mathbf{c},s}^N(\mathbf{x})$ so as to make the distribution fitting and acquire a probability function.

Definition 5. *The discrete Gaussian distribution over* $\mathbb{R}^N$ *with the center* $\mathbf{c} \in \mathbb{R}^N$ *and the standard deviation s > 0, is defined as* $D_{\mathbf{c},s}^N(\mathbf{x}) = \rho_{\mathbf{c},s}^N(\mathbf{x}) / \rho_{\mathbf{c},s}^N(\mathbf{\Lambda})$, *where* $\mathbf{x} \in \mathbb{R}^N$.

In this paper, ρ_s^N and D_s^N are abbreviated from $\rho_{0,s}^N$ and $D_{0,s}^N$ respectively for convenience. In the following lemma, Lyubashevsky [22] gave two properties of a discrete distribution $D_{\mathbf{c},\sigma}^N(\mathbf{x})$ in dimension *N* with standard deviation σ at center **c**.

Lemma 4. *Let* $\mathbf{c} \in \mathbb{Z}^N$.

(1) If $\sigma = \omega(\|\mathbf{c}\|\sqrt{\log N})$, *then* $Pr[\mathbf{x} \in D_\sigma^N; D_\sigma^N(\mathbf{x}) / D_{\mathbf{c},\sigma}^N(\mathbf{x}) = O(1)] = 1 - 2^{-\omega(\log N)}$.

(2) If $\sigma = \alpha\|\mathbf{c}\|$ *and* $\alpha > 0$, *then* $Pr[\mathbf{x} \in D_\sigma^N; D_\sigma^N(\mathbf{x}) / D_{\mathbf{c},\sigma}^N(\mathbf{x}) < e^{12/\alpha + 1/(2\sigma^2)}] > 1 - 2^{-100}$.

2.5. Sampling Technique

According to [31], if one takes a so-called noise vector from a Gaussian distribution and adds this vector to a lattice, then one can obtain a distribution that is statistically close to uniform one. Based on this, Gentry et al. [20] presented a sampling algorithm and a trapdoor generation algorithm by using the Gaussian sampling technique over general lattices. To reduce the private key size, Ducas et al. [26] improved Gentry et al.'s scheme to propose a particular sampling algorithm over NTRU lattices that can produce short trapdoor by using a short basis $\mathbf{B}_{f,g}$ of $\mathbf{\Lambda}_{h,q}$ which is generated by ***TrapGen*** in the previous subsection. In our scheme, we will use Ducas et al.'s technique to produce the private key of a user by the short basis $\mathbf{B}_{f,g}$ without leaking any information of $\mathbf{B}_{f,g}$. Ducas et al.'s trapdoor generation algorithm has the following properties.

Lemma 5. *Given a prime q, an N-dimensional lattice* Λ*, a short basis* $\mathbf{B}_{f,g}$*, if* $s \geq \left\|\tilde{\mathbf{B}}_{f,g}\right\| \omega(\sqrt{\log N})$ *and* $0 < \varepsilon < 1$*, where* $\tilde{\mathbf{B}}_{f,g}$ *denotes* $\mathbf{B}_{f,g}$*'s Gram-Schmidt orthogonalization, we have*

$$\Pr[\|\mathbf{x}-\mathbf{c}\| > s\sqrt{N}] \leq \frac{1+\varepsilon}{1-\varepsilon} 2^{-N} \quad \text{for any } \mathbf{c} \in \mathbb{R}^N \quad \text{and} \quad \mathbf{x} \leftarrow D_{\mathbf{c},s}^N .$$

and, there is an algorithm ***SampleGau****(* $\mathbf{B}_{f,g}$*, s,* ***c****) which produces a distribution statistically close to* $D_{\mathbf{c},s}^N$ [26].

2.6. Rejection Sampling Algorithm

Lyubashevsky [22] proposed the rejection sampling technique to sign a message in lattice-based cryptography. This technique is simple and needs just a few matrix-vector multiplications and rejection samplings. Indeed, Lyubashevsky's signing algorithm [22] is different from the one proposed by Micciancio and Peikert [33] even though both algorithms employ similar public keys. The main difference is that Lyubashevsky produces a signature by using the rejection sampling instead of the hash-and-sign technique. Moreover, the sizes of both signature and private key in Lyubashevsky's scheme are smaller than those in Micciancio and Peikert's scheme under the same security level. Here, we explain the workings of Lyubashevsky's rejection sampling technique. A signer first selects a private key **S** which is an $m \times k$ matrix of random integers of absolute value at most d. And then the signer chooses an $n \times m$ matrix **A** of random integers in Z_q and computes the other matrix $\mathbf{T} = \mathbf{AS}$. The signer's associated public key consists of **A** and **T**. In addition, a cryptographic hash function $H: \{0,1\}^* \rightarrow \{\mathbf{v} : \mathbf{v} \in \{-1,0,1\}^k, \|\mathbf{v}\|_1 \leq \lambda\}$ is selected, where λ is constant. In the sign procedure, the signer takes her/his private key **S**, public key **A** and a message μ as input, and returns a signature (**z**, **c**). Upon receiving a signature (**z**, **c**), a verifier validates the signature using the public keys **A** and **T**. The setup, sign and verify procedures of the rejection sampling technique are presented in the following Algorithm 1.

Algorithm 1: Rejection Sampling Technique

Setup(n, λ, m, k)

$H: \{0,1\}^* \rightarrow \{\mathbf{v} : \mathbf{v} \in \{-1,0,1\}^k, \|\mathbf{v}\|_1 \leq \lambda\}$, λ is constant.

Private Key: $\mathbf{S} \leftarrow \{-d, ..., 0, ..., d\}^{m \times k}$.

Verification Key: $\mathbf{A} \leftarrow Z_q^{n \times m}$, $\mathbf{T} = \mathbf{AS}$.

Sign(**A**, **S**, μ):

1. $y \leftarrow D_\sigma^m$.
2. $\mathbf{c} = H(\mathbf{A}\,\mathrm{y},\ \mu)$.
3. $\mathbf{z} = \mathbf{Sc} + y$.
4. Output the pair (**z**, **c**) with the probability min $\left[\frac{D_\sigma^m(\mathrm{z})}{M \cdot D_{\mathbf{Sc},\sigma}^m(\mathrm{z})},\ 1\right]$, where $M = O(1)$.

Verify(**A**, **T**, **z**, **c**, μ):

Accept it if both conditions $\|\mathbf{z}\| \leq 2\sigma\sqrt{m}$ and **c**=H(**Az**-**Tc**, μ) hold.

Here, we present the main concept of the signing algorithm. The rejection sampling technique is to enable the distribution of the signature (**z**, **c**) independent of the secret key **S**. Thus, we would like to obtain a target distribution **z** from D_σ^m, but **z** in the signing algorithm comes from the distribution $D_{\mathbf{Sc},\sigma}^m$. For an appropriately-chosen value M and a standard deviation σ, e.g. M = 2.72 and σ = 15,157 [22], the signing algorithm will output a valid signature satisfying both

$\|\mathbf{z}\| \le 2\sigma\sqrt{m}$ and $\mathbf{c}=H(\mathbf{Az}-\mathbf{Tc}, \mu)$ with probability approximately 1/*M*. And the distribution of $\mathbf{z}$ is statistically close to the distribution chosen from D_σ^m.

2.7. Hardness Assumptions

In this section, we present the short integer solution (SIS) problem over lattices as the security assumption. The difficulty of solving the SIS problem is equivalent to the difficulty of the worst case of solving the short independent vector problem (SIVP) with an approximation polynomial factor [34]. The SIS problem and its assumption are defined as follows.

Definition 6. *Let q and β, respectively, be a positive integer and a real number, and* $f_1, f_2, \ldots, f_m$ *be polynomials chosen uniformly and independently from* R_q. *The* $SIS_{q,m,\beta}$ *problem over lattices is to find m non-zero integers* $r_1, r_2, \ldots, r_m$ *that satisfy two conditions* $\sum_{i=1}^{m} r_i f_i = 0 \mod q$ *and* $\|(r_1, r_2, \ldots, r_m)\| \le \beta$.

Definition 7 (SIS assumption)**.** *Given a real number β, a positive integer q, and m polynomials* $f_1, f_2, \ldots, f_m$ *chosen uniformly and independently from* R_q, *there exists no probabilistic polynomial-time adversary A with non-negligible probability for solving the SIS problem. The successful probability (advantage) of the adversary A is presented as*

$$Adv_A = Pr[A(<b, q, f_1, f_2, \ldots, f_m>) = (r_1, r_2, \ldots, r_m) : \|(r_1, r_2, \ldots, r_m)\| \le \beta].$$

As stated in Lemma 3, the distribution of $h = g/f$ is statistically close to the uniform distribution of R_q [35]. Hence, the SIS problem on NTRU lattice is to find a pair $(\mathbf{z}_1, \mathbf{z}_2)$ such that $\mathbf{z}_1 + h * \mathbf{z}_2 = 0$ and $\|(\mathbf{z}_1, \mathbf{z}_2)\| \le \beta$.

3. Syntax and Security Model of RCLS

The framework of RCLS scheme is identical to that of the RCLS schemes in [14,15]. In an RCLS scheme, there are three roles, namely, a key generation center (KGC), signers and verifiers. An RCLS scheme consists of eight algorithms that are defined as follows.

Definition 8. *An RCLS scheme contains eight algorithms:*

- *Setup* (*N*): The algorithm is probabilistic and performed by an KGC. The algorithm takes as input a security parameter *N*, it returns the public parameters *Parms* and a system secret key S_{KGC}. S_{KGC} is kept secret by the KGC and *Parms* are made public.
- *Partial private key extract* (*ID*): This deterministic algorithm is performed by the KGC. Upon receiving the identity *ID* of a user, the KGC produces the user's partial private key D_{ID} and the first partial public key P_{ID} that are returned to the user.
- *Time key update* (*ID*, *t*): This deterministic algorithm is performed by the KGC. Upon receiving the identity *ID* of a user and a time period *t*, the KGC produces the time update key $T_{ID,t}$ of the user and returns it to the user.
- *Set secret value* (*ID*): This probabilistic algorithm is performed by a user with *ID*. The user randomly selects a secret value S_{ID}, with which the user computes the second partial public key R_{ID}.
- *Set private key* (D_{ID}, $T_{ID,t}$, S_{ID}): This deterministic algorithm is performed by a user with *ID*. The private key $SK_{ID} = (D_{ID}, T_{ID,t}, S_{ID})$ is set by the user.

- *Set public key* (P_{ID} , R_{ID}): This deterministic algorithm is performed by a user with *ID*. The public key PK_{ID} = (P_{ID} , R_{ID}) is set by the user, where P_{ID} and R_{ID} are the first partial and the second partial public keys respectively.
- *Sign* (*ID*, SK_{ID} , μ, *t*): This probabilistic algorithm is performed by a user with *ID*. It takes as input the private key SK_{ID} of the user, a message μ and a time period *t*, and returns a signature ζ on μ.
- *Verify* (*ID*, PK_{ID} , μ, ζ, *t*): This deterministic algorithm is performed by a verifier (or receiver). It takes as input the public key PK_{ID} of a user with *ID*, a message μ, a time period *t*, and a signature ζ and it returns "accept" if the signature ζ is validated. Otherwise, it returns "reject".

By the security model of RCLS schemes in [14,15], adversaries have three types that are presented as follows.

- Type I adversary (outsider): The adversary knows the time update key and the secret value of any entity, which are respectively obtained by listening the public channel and replacing the associated public key.
- Type II adversary (honest-but-curious KGC): The adversary may produce the partial private key and time update key of any entity, but it does not know the associated secret value.
- Type III adversary (revoked user): The adversary owns the partial private key and knows the associated secret value, but it does not get the current time update key.

Definition 9. *We say that an RCLS scheme has existential unforgeability against adaptive chosen message attacks (RCLS-UF-ACMA) if a PPT adversary A with a non-negligible advantage wins the following RCLS-UF-ACMA game, which is cooperatively performed by A and a challenger C.*

- *Setup*. The *setup* algorithm is performed by the challenger C to produce public parameters *Parms* and the system secret key S_{KGC} . S_{KGC} is kept secret for C. It is worth mentioning, that if the adversary A is Type II, S_{KGC} is sent to A. Note that for Type I and III adversaries, the KGC plays as the role of the challenger C. For Type II adversary, the honest-but-curious KGC is the adversary A.
- *Queries*: A may issue a number of different queries to C adaptively as follows. It is worth mentioning, that Type II adversary has the system secret key S_{KGC} so that it may compute the partial private key and time update key of any entity.
    - *Partial private key extract queries* (*ID*). Upon receiving the identity *ID* of a user, C performs the *partial private key extract* algorithm to produce and return the user's partial private key D_{ID} to A.
    - *Time key update queries* (*ID*, *t*). Upon receiving the identity *ID* of a user and a time period *t*, the *C* performs the *time key update* algorithm to produce and return the time update key $T_{ID,t}$ to A.
    - *Secret value queries* (*ID*). Given a user's *ID*, C performs the *set secret value* algorithm to produce and return the secret value S_{ID} to A.
    - *Public key queries* (*ID*). Upon receiving the identity *ID* of a user, C returns PK_{ID} to A.
    - *Public key replacement queries* (*ID*, PK'_{ID}). Upon receiving the identity *ID* of a user and a new public key PK'_{ID} , C records this replacement.
    - *Sign queries* (*ID*, PK_{ID} , μ, *t*). Upon receiving *ID* and PK_{ID} of a user, a message μ and a time period *t*. C plays the role of the signer and performs the *sign* algorithm to produce a valid signature ζ on μ and returns ζ to A.

- *Forgery*: Assume that the adversary A produces (ID^*, PK_{ID^*}, μ^*, ζ^*, t^*). It is worth mentioning, that ID^* is the target identity. It is said that A wins the RCLS-UF-ACMA game when the following situations hold:
  - (ID^*, μ^*, t^*) was never issued in the *sign queries*.
  - The *verify* algorithm on (ID^*, PK_{ID^*}, μ^*, ζ^*, t^*) outputs "accept".
  - If A is of Type I adversary, the *partial private key extract queries* on ID^* was never issued.
  - If A is of Type II adversary, ID^* was never issued in the *secret value* and *public key replacement queries*.
  - If A is of Type III adversary, the *time key update queries* on (ID^*, t^*) was never issued.

4. Concrete RCLS Scheme over Lattices

As defined in Definition 8 in Section 4, an RCLS scheme consists of eight algorithms. Here, we propose an efficient lattice-based RCLS scheme. Eight algorithms are presented as follows:

- *Setup*: Let $s > 0$, $\sigma > 0$, and λ be a positive integer and N be a security parameter, the KGC chooses a prime q. Then, the KGC runs ***TrapGen***(q, N) of Lemma 3 in Section 2.3 to obtain (f, g), $h = g * f^{-1}$, $\|f\| < s\sqrt{N}$, and $\|g\| < s\sqrt{N}$ with short basis $\mathbf{B} = \begin{bmatrix} \mathbf{C}(g) & -\mathbf{C}(f) \\ \mathbf{C}(G) & -\mathbf{C}(F) \end{bmatrix}$ of $\mathbf{\Lambda}_{h,q}$, where f, g, F, $G \in R_q$. Furthermore, the KGC sets the system secret key S_{KGC} as $\mathbf{B}$ and selects two system public keys $a_1, a_2 \in Z_q^N$ and three hash functions $H_0, H_1 : \{0, 1\}^* \to Z_q^N$ and H_2: $Z_q^N \times Z_q^N \times \{0, 1\}^* \to \{\mathbf{v}: \mathbf{v} \in \{-1, 0, 1\}^N, \|\mathbf{v}\|_1 \leq \lambda\}$, where $\|\mathbf{v}\|_1$ denotes the amount of nonzero elements of the vector $\mathbf{v}$. The public parameters are $Parms = <N, s, \alpha, \lambda, q, h, a_1, a_2, H_0, H_1, H_2>$.
- *Partial private key extract*: Upon receiving the identity $ID \in \{0, 1\}^*$ of a user, the KGC produces the partial private key $(\mathbf{s}_1, \mathbf{s}_2)$ such that $\mathbf{s}_1 + h * \mathbf{s}_2 = P_{ID}$ and $\|(\mathbf{s}_1, \mathbf{s}_2)\| < s\sqrt{2N}$ by running ***SampleGau***($\mathbf{B}$, s, $(P_{ID}, 0)$) of Lemma 5 in Section 2.5, where $P_{ID} = H_0(ID) \in Z_q^N$ is the first partial public key. The KGC returns the partial private key $D_{ID} = (\mathbf{s}_1, \mathbf{s}_2)$ to the user securely. Note that Lyubashevsky et al. [36] have shown that if one knows (h, P_{ID}), recovering $(\mathbf{s}_1, \mathbf{s}_2)$ is still hard.
- *Time key update*: Upon receiving the identity ID of a non-revoked user and a time period t, the KGC produces the time update key $(\mathbf{s}_3, \mathbf{s}_4)$ such that $\mathbf{s}_3 + h * \mathbf{s}_4 = T_{ID}$ and $\|(\mathbf{s}_3, \mathbf{s}_4)\| < s\sqrt{2N}$ by running ***SampleGau***($\mathbf{B}$, s, $(T_{ID}, 0)$) of Lemma 5 in Section 2.5, where $T_{ID} = H_1(ID, t) \in Z_q^N$. The KGC then sends the time update key $T_{ID,t} = (\mathbf{s}_3, \mathbf{s}_4)$ to the user by using a public channel.
- *Set secret value*: The user with ID randomly chooses a secret value $S_{ID} = (\mathbf{s}_5, \mathbf{s}_6)$ uniformly from $\{-d, \ldots, 0, \ldots, d\}$, where $1 \leq d \leq 31$. Meanwhile, the second partial public key is $R_{ID} = a_1 * \mathbf{s}_5 + a_2 * \mathbf{s}_6$.
- *Set private key*: The user with ID may set the private key $SK_{ID} = (D_{ID}, T_{ID,t}, S_{ID})$.
- *Set public key*: The user with ID may set the public key $PK_{ID} = (P_{ID}, R_{ID})$.
- *Sign*: A signer with the private key SK_{ID} takes as input a message $\mu \in \{0,1\}^*$, the signer randomly and independently selects $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4, \mathbf{y}_5, \mathbf{y}_6$ by the distribution D_σ^N, and computes the following values:

$$\mathbf{c} = H_2(\mathbf{y}_1 + h * \mathbf{y}_2, \mathbf{y}_3 + h * \mathbf{y}_4, a_1 * \mathbf{y}_5 + a_2 * \mathbf{y}_6, \mu);$$

$$\mathbf{z}_1 = \mathbf{y}_1 + \mathbf{s}_1 * \mathbf{c};\ \ \mathbf{z}_2 = \mathbf{y}_2 + \mathbf{s}_2 * \mathbf{c};\ \ \mathbf{z}_3 = \mathbf{y}_3 + \mathbf{s}_3 * \mathbf{c};$$

$$\mathbf{z}_4 = \mathbf{y}_4 + \mathbf{s}_4 * \mathbf{c};\ \ \mathbf{z}_5 = \mathbf{y}_5 + \mathbf{s}_5 * \mathbf{c};\ \ \mathbf{z}_6 = \mathbf{y}_6 + \mathbf{s}_6 * \mathbf{c},$$

where $\|(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6)\| \leq 2\sigma\sqrt{6N}$. If no such $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6)$ is produced, repeat this algorithm. The above procedure is the rejection sampling technique. Finally, there exists a constant $M = O(1)$ such that the user can produce a signature $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6, \mathbf{c})$ with probability $\min\left(D_{\sigma}^{6N}(\mathbf{z}) / MD_{\mathbf{v},\sigma}^{6N}(\mathbf{z}), 1\right)$ which is similar to the ring variants of Lyubashevsky's scheme [22], where

$$\mathbf{z} = [\mathbf{z}_1^T \| \mathbf{z}_2^T \| \mathbf{z}_3^T \| \mathbf{z}_4^T \| \mathbf{z}_5^T \| \mathbf{z}_6^T]^T$$

and

$$\mathbf{v} = [(\mathbf{s}_1 * \mathbf{c})^T \| (\mathbf{s}_2 * \mathbf{c})^T \| (\mathbf{s}_3 * \mathbf{c})^T \| (\mathbf{s}_4 * \mathbf{c})^T \| (\mathbf{s}_5 * \mathbf{c})^T \| (\mathbf{s}_6 * \mathbf{c})^T]^T.$$

- *Verify*: Given a signature $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6, \mathbf{c})$ for a user's *ID* on a message μ, a verifier needs to validate the signature by the equality

$$\mathbf{c} = H_2(\mathbf{z}_1 + h * \mathbf{z}_2 - P_{ID} * \mathbf{c}, \mathbf{z}_3 + h * \mathbf{z}_4 - T_{ID} * \mathbf{c}, a_1 * \mathbf{z}_5 + a_2 * \mathbf{z}_6 - R_{ID} * \mathbf{c}, \mu).$$

The *verify* algorithm returns "accept" if the checking equality holds. Otherwise, it returns "reject". The correctness of the checking equality follows by

$$(\mathbf{z}_1 + h * \mathbf{z}_2 - P_{ID} * \mathbf{c}, \mathbf{z}_3 + h * \mathbf{z}_4 - T_{ID} * \mathbf{c}, a_1 * \mathbf{z}_5 + a_2 * \mathbf{z}_6 - R_{ID} * \mathbf{c})$$

$$= (\mathbf{y}_1 + \mathbf{s}_1 * \mathbf{c} + h * (\mathbf{y}_2 + \mathbf{s}_2 * \mathbf{c}) - (\mathbf{s}_1 + h * \mathbf{s}_2) * \mathbf{c}, \mathbf{y}_3 + \mathbf{s}_3 * \mathbf{c} + h * (\mathbf{y}_4 + \mathbf{s}_4 * \mathbf{c}) - (\mathbf{s}_3 + h * \mathbf{s}_4) * \mathbf{c},$$

$$a_1 * (\mathbf{y}_5 + \mathbf{s}_5 * \mathbf{c}) + a_2 * (\mathbf{y}_6 + \mathbf{s}_6 * \mathbf{c}) - (a_1 * \mathbf{s}_5 + a_2 * \mathbf{s}_6) * \mathbf{c})$$

$$= (\mathbf{y}_1 + h * \mathbf{y}_2, \mathbf{y}_3 + h * \mathbf{y}_4, a_1 * \mathbf{y}_5 + a_2 * \mathbf{y}_6).$$

5. Security Analysis

In the following, we demonstrate that our lattice-based RCLS scheme is secure against both Type I adversary (outsider) and Type III adversary (revoked user) in Theorem 1 while the security against Type II adversary (honest-but-curious KGC) is proven in Theorem 2. The proof technique of both theorems employs the rejection sampling technique in [22] and the Forking lemma in [37].

Theorem 1. *Let three hash functions H_0, H_1 and H_2 be random oracles and N be the security parameter. Assume that a PPT adversary A (Types I and III) can break our lattice-based RCLS scheme with non-negligible probability ε. Thus, an algorithm C is constructed to resolve the SIS problem with non-negligible probability $(1-2^{-\omega(\log N)})\varepsilon$.*

Proof. Let q be a prime, N be a positive integer and λ, s, $\sigma > 0$. Let the algorithm C be a challenger who receives a random instance $(q, 2N, 2\lambda s\sqrt{2N} + 4\sigma\sqrt{2N})$ of the SIS problem. In the following, we will show how the challenger C can compute a non-zero vector solution $(\mathbf{u}_1, \mathbf{u}_2) \in R_q^2$ of the SIS problem by using A. Here, A (Type I or Type III adversary) interacts with C as defined in the RCLS-UF-ACMA game of Definition 9.

- *Setup*. The challenger C randomly chooses polynomials a_1, a_2, $h \in R_q$ and controls the random oracles H_0, H_1 and H_2. The public parameters $Parms = <N, s, \alpha, \lambda, q, h, a_1, a_2, H_0, H_1, H_2>$ are sent to A. Meanwhile, C maintains several initially empty lists L_0, L_1, L_2 and L_S.
- *Queries*. A can adaptively issue several queries to C as follows:
  - H_0 *queries*: Let L_0 consist of tuples of the form $< ID_i, D_{ID_i}, P_{ID_i} >$. Upon receiving a query

with ID_i from A, C produces a response to this query as follows.

1. Search ID_i in L_0. If it is found, the same answer in L_0 is returned to A because the query has been ever issued.
2. Otherwise, select $\mathbf{s}_{i1}, \mathbf{s}_{i2} \in D_s^N$ at random such that $\|(\mathbf{s}_{i1}, \mathbf{s}_{i2})\| < s\sqrt{2N}$ and compute the polynomial $P_{ID_i} = \mathbf{s}_{i1} + h * \mathbf{s}_{i2}$. Then P_{ID_i} is sent to A and $< ID_i, D_{ID_i} = (\mathbf{s}_{i1}, \mathbf{s}_{i2}), P_{ID_i} >$ is added in the list L_0.

- H_1 *queries*: Let L_1 consist of tuples of the form $< ID_i, t, \mathrm{T}_{1i}, \mathrm{T}_{ID,t} >$. Upon receiving a query with (ID_i, t) from A, C produces a response to this query as follows.

1. Search (ID_i, t) in L_1. If it is found, the same answer in L_1 is returned to A because the query has been ever issued.
2. Otherwise, select $\mathbf{s}_{i3}, \mathbf{s}_{i4} \in D_s^N$ at random such that $\|(\mathbf{s}_{i3}, \mathbf{s}_{i4})\| < s\sqrt{2N}$ and compute the polynomial $T_{1i} = \mathbf{s}_{i3} + h * \mathbf{s}_{i4}$. Then T_{1i} is sent to A and $< ID_i, t, \mathrm{T}_{1i}, \mathrm{T}_{ID,t} >$ is added in the list L_1.

- H_2 *queries*: Let L_2 consist of tuples of the form $< w_j, x_j, v_j, \mu_j, \mathbf{c}_j >$. Upon receiving a query with (w_j, v_j, x_j, μ_j) from A, C produces a response to this query as follows.

1. Search (w_j, v_j, x_j, μ_j) in L_2. If it is found, the same answer in L_2 is returned to A because the query has been ever issued.
2. Otherwise, randomly select $\mathbf{c}_j \in Z_q^N$. Then $\mathbf{c}_j$ is sent to A and $< w_j, x_j, v_j, m_j, \mathbf{c}_j >$ is added in the list L_2.

- *Partial private key queries*: A issues this query along with ID_i, C produces a response to this query as follows.

1. Search ID_i in L_0. If it is found, the same answer in L_0 is returned to A because the query has been ever issued.
2. Otherwise, issue the H_0 *query* to obtain the tuple $< ID_i, D_{ID_i}, P_{ID_i} >$. Then, return D_{ID_i} to A.

- *Time key update queries*: A issues this query along with (ID_i, t), C produces a response to this query as follows.

1. Search (ID_i, t) in L_1. If it is found, the same answer in L_1 is returned to A because the query has been ever issued.
2. Otherwise, issue the H_1 *query* to obtain the tuple $< ID_i, t, \mathrm{T}_{1i}, \mathrm{T}_{ID,t} >$. Then, return T_{1i} to A.

- *Secret value queries*: Let L_S consist of tuples of the form $< ID_i, S_{ID_i}, R_{ID_i} >$. Upon receiving a query with ID_i from A, C produces a response to this query as follows.

1. Search ID_i in L_S. If it is found, the same answer in L_S is returned to A because the query has been ever issued.
2. Otherwise, randomly select $\mathbf{s}_{i5}, \mathbf{s}_{i6} \in \{-d, \ldots, 0, \ldots, d\}$, where $1 \le d \le 31$, and compute the polynomial $R_{ID_i} = a_1 * \mathbf{s}_{i5} + a_2 * \mathbf{s}_{i6}$. Then $S_{ID_i} = (\mathbf{s}_{i5}, \mathbf{s}_{i6})$ is sent to A and $< ID_i, S_{ID_i}, R_{ID_i} >$ is added in the list L_S.

- *Public key queries*: A issues this query along with ID_i, C produces a response to this query

as follows.

1. Search ID_i in L_0 and L_s. If it is found, which means that the query has been ever issued, then C returns A with the same answer $PK_{ID_i} = (P_{ID_i}, R_{ID_i})$, where P_{ID_i} and R_{ID_i} are taken from L_0 and L_s, respectively.
2. Otherwise, issue the H_0 *query* and *Secret value query* to obtain P_{ID_i} and R_{ID_i}. Then $PK_{ID_i} = (P_{ID_i}, R_{ID_i})$ is sent to A.

- *Public key replacement queries*: A issues this query along with a new public key $PK'_{ID_i} = (P'_{ID_i}, R'_{ID_i})$ of ID_i to replace the old public key $PK_{ID_i} = (P_{ID_i}, R_{ID_i})$, C replaces the P_{ID_i} in L_0 with P'_{ID_i} and the R_{ID_i} in L_s with R'_{ID_i}.
- *Sign queries*: Upon receiving a request from A along with a message μ_j, a time period t and (ID_i, PK_{ID_i}), where $PK_{ID_i} = (P_{ID_i}, R_{ID_i})$, the challenger C makes the following steps to produce a valid signature.
  1. Search ID_i in L_0, L_1 and L_s, respectively, to obtain $< ID_i, D_{ID_i}, P_{ID_i} >$, $< ID_i, t, T_{1i}, T_{ID,t} >$ and $< ID_i, S_{ID_i}, R_{ID_i} >$.
  2. Randomly choose $\mathbf{c}_j \in \{\mathbf{v}: \mathbf{v} \in \{-1, 0, 1\}^N, \|\mathbf{v}\|_1 \leq \lambda\}$ and $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6 \in D_\sigma^N$ with $\|(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6)\| \leq 2\sigma\sqrt{6N}$. Then, compute $w_j = \mathbf{z}_1 + h * \mathbf{z}_2 - P_{ID_i} * \mathbf{c}_j$, $v_j = \mathbf{z}_3 + h * \mathbf{z}_4 - T_{1i} * \mathbf{c}_j$ and $x_j = a_1 * \mathbf{z}_5 + a_2 * \mathbf{z}_6 - R_{ID_i} * \mathbf{c}_j$.
  3. Add $< w_j, v_j, x_j, \mu_j, \mathbf{c}_j >$ in the list L_2 and send the signature $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6, \mathbf{c}_j)$ on μ_j to A.

     Note that the signature $\zeta = (\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6, \mathbf{c}_j)$ is valid because it may satisfy the following equality:

$$\mathbf{c}_j = H_2(\mathbf{z}_1 + h * \mathbf{z}_2 - P_{ID_i} * \mathbf{c}_j, \mathbf{z}_3 + h * \mathbf{z}_4 - T_{1i} * \mathbf{c}_j, a_1 * \mathbf{z}_5 + a_2 * \mathbf{z}_6 - R_{ID_i} * \mathbf{c}_j, \mu_j) = H_2(w_j, v_j, x_j, m_j).$$

     Therefore, when the adversary A issues the *Sign query*, the challenger C can output a valid signature even though C does not possess the valid secret key or time update key.

- *Forgery*: After making all the queries needed, the adversary A forges a signature tuple $(\mathbf{z}_1^*, \mathbf{z}_2^*, \mathbf{z}_3^*, \mathbf{z}_4^*, \mathbf{z}_5^*, \mathbf{z}_6^*, \mathbf{c}^*)$ on message μ^* for ID^* at time period t^*.

When A successfully forges a valid signature $(\mathbf{z}_1^*, \mathbf{z}_2^*, \mathbf{z}_3^*, \mathbf{z}_4^*, \mathbf{z}_5^*, \mathbf{z}_6^*, \mathbf{c}^*)$, the challenger C uses the Forking lemma [37] and replays A with different hash value of H_2 *queries* to produce another valid signature $(\mathbf{z}_1', \mathbf{z}_2', \mathbf{z}_3', \mathbf{z}_4', \mathbf{z}_5', \mathbf{z}_6', \mathbf{c}')$ such that $\mathbf{c}^* \neq \mathbf{c}'$ by the same random tape. Because $(\mathbf{z}_1^*, \mathbf{z}_2^*, \mathbf{z}_3^*, \mathbf{z}_4^*, \mathbf{z}_5^*, \mathbf{z}_6^*, \mathbf{c}^*)$ and $(\mathbf{z}_1', \mathbf{z}_2', \mathbf{z}_3', \mathbf{z}_4', \mathbf{z}_5', \mathbf{z}_6', \mathbf{c}')$ are two valid signatures on the message μ^* for (ID^*, PK_{ID^*}, t^*), we can obtain the equality

$$H_2(\mathbf{z}_1^* + h * \mathbf{z}_2^* - P_{ID^*} * \mathbf{c}^*, \mathbf{z}_3^* + h * \mathbf{z}_4^* - T_{1i^*} * \mathbf{c}^*, a_1 * \mathbf{z}_5^* + a_2 * \mathbf{z}_6^* - R_{ID^*} * \mathbf{c}^*, \mu^*)$$

$$= H_2(\mathbf{z}_1' + h * \mathbf{z}_2' - P_{ID^*} * \mathbf{c}', \mathbf{z}_3' + h * \mathbf{z}_4' - T_{1i^*} * \mathbf{c}', a_1 * \mathbf{z}_5^* + a_2 * \mathbf{z}_6^* - R_{ID^*} * \mathbf{c}', \mu^*),$$

which reduces to

$$\mathbf{z}_1^* + h * \mathbf{z}_2^* - P_{ID^*} * \mathbf{c}^* = \mathbf{z}_1' + h * \mathbf{z}_2' - P_{ID^*} * \mathbf{c}'.$$

Since $P_{ID^*} = \mathbf{s}_1 + h * \mathbf{s}_2$, we arrive at

$$\mathbf{z}_1^* + h * \mathbf{z}_2^* - (\mathbf{s}_1 + h * \mathbf{s}_2) * \mathbf{c}^* = \mathbf{z}_1' + h * \mathbf{z}_2' - (\mathbf{s}_1 + h * \mathbf{s}_2) * \mathbf{c}'$$

$$\mathbf{z}_1^* - \mathbf{z}_1' - \mathbf{s}_1(\mathbf{c}^* - \mathbf{c}') + h * (\mathbf{z}_2^* - \mathbf{z}_2' - \mathbf{s}_2(\mathbf{c}^* - \mathbf{c}')) = 0$$

$$(1,\ h) * (\mathbf{z}_1^* - \mathbf{z}_1' - \mathbf{s}_1(\mathbf{c}^* - \mathbf{c}'),\ \mathbf{z}_2^* - \mathbf{z}_2' - \mathbf{s}_2(\mathbf{c}^* - \mathbf{c}')) = 0.$$

Then, the challenger C sets $(\mathbf{u}_1,\ \mathbf{u}_2) = (\mathbf{z}_1^* - \mathbf{z}_1' - \mathbf{s}_1(\mathbf{c}^* - \mathbf{c}'),\ \mathbf{z}_2^* - \mathbf{z}_2' - \mathbf{s}_2(\mathbf{c}^* - \mathbf{c}'))$.

If $\left\|(\mathbf{z}_1^* - \mathbf{z}_1',\ \mathbf{z}_2^* - \mathbf{z}_2')\right\| \le 4\sigma\sqrt{2N}$ and $\left\|(\mathbf{s}_1,\ \mathbf{s}_2)\right\| \le s\sqrt{2N}$ with overwhelming probability, we can obtain $\left\|(\mathbf{u}_1,\ \mathbf{u}_2)\right\| \le 2\lambda s\sqrt{2N} + 4\sigma\sqrt{2N}$. As stated in Lemma 3, the distribution of $h = g/f$ is statistically close to the uniform distribution of R_q [35]. The SIS problem on NTRU lattice is to find a pair $(\mathbf{u}_1, \mathbf{u}_2) \in R_q^2$ such that $\mathbf{u}_1 + h * \mathbf{u}_2 = 0$ and $\left\|(\mathbf{u}_1, \mathbf{u}_2)\right\| \le \beta$, where β is $2\lambda s\sqrt{2N} + 4\sigma\sqrt{2N}$. Since the adversary A does not know the system secret key **B** generated by $g, f \in R_q$ and has generated such a pair $(\mathbf{u}_1, \mathbf{u}_2)$, we say that the adversary A solves the SIS problem. According to the same probability analysis in [22], if the adversary A can break our lattice-based RCLS scheme with non-negligible probability ε. Then, we can construct an algorithm C to solve the SIS problem with non-negligible probability $(1 - 2^{-\omega(logN)})\varepsilon$. □

Theorem 2. *Let three hash functions H_0, H_1 and H_2 be random oracles and N be the security parameter. Assume that an PPT adversary A (Types II) can break our lattice-based RCLS scheme with non-negligible probability ε. Thus, an algorithm C is constructed to resolve the SIS problem with non-negligible probability* $(1 - 2^{-\omega(logN)})\varepsilon$.

Proof. Let q be a prime, N be a positive integer and $\lambda, s, \sigma > 0$. Let the algorithm C be a challenger who receives a random instance $(q,\ 2N,\ 2\lambda d\sqrt{2N} + 4\sigma\sqrt{2N})$ of the SIS problem. In the following, we will show how C can compute a non-zero vector solution $(\mathbf{u}_1, \mathbf{u}_2)$ of the SIS problem by using A. Here, A (Type II adversary) interacts with the challenger C as defined in the RCLS-UF-ACMA game of Definition 9.

- *Setup*. The challenger C performs the *Setup* algorithm of our lattice-based RCLS scheme to set S_{KGC} = **B** and $Parms = <N, s, \alpha, \lambda, q, h, a_1, a_2, H_0, H_1, H_2>$, where three hash functions H_0, H_1 and H_2 are random oracles. The system secret key and *Parms* are then sent to A. Having the system secret key S_{KGC}, C can compute the partial private key D_{ID}, time update key $T_{ID,t}$, and partial public key P_{ID} of any user with ID_i without issuing the other queries. Meanwhile, C maintains several initially empty lists L_0, L_1, L_2 and L_s.
- *Queries*. A can adaptively issue several queries to C as follows:
  - H_0 *queries*: Let L_0 consist of tuples of the form $< ID_i,\ D_{ID_i},\ P_{ID_i} >$. Upon receiving a query with ID_i from A, C produces a response to this query as follows.
    1. Search ID_i in L_0. If it is found, the same answer in L_0 is returned to A because the query has been ever issued.
    2. Otherwise, randomly select a $P_{ID_i} \in Z_q^N$ and run the algorithm ***SampleGau***(**B**, s, (P_{ID_i}, 0)) to obtain $\mathbf{s}_{i1},\ \mathbf{s}_{i2} \in D_s^N$ such that $\left\|(\mathbf{s}_{i1},\ \mathbf{s}_{i2})\right\| < s\sqrt{2N}$. Then P_{ID_i} is sent to A and $< ID_i,\ D_{ID_i} = (\mathbf{s}_{i1},\ \mathbf{s}_{i2}),\ P_{ID_i} >$ is added in the list L_0.
  - H_1 *queries*: Let L_1 consist of tuples of the form $< ID_i,\ t,\ T_{1i},\ T_{ID,t} >$. Upon receiving a query

with (ID_i, t) from A, C produces a response to this query as follows.

1. Search (ID_i, t) in L_1. If it is found, the same answer in L_1 is returned to A because the query has been ever issued.
2. Otherwise, select $\mathbf{s}_{i3}, \mathbf{s}_{i4} \in D_s^N$ at random such that $\|(\mathbf{s}_{i3}, \mathbf{s}_{i4})\| < s\sqrt{2N}$ and compute the polynomial $T_{1i} = \mathbf{s}_{i3} + h * \mathbf{s}_{i4}$. Then T_{1i} is sent to A and $< ID_i, t, T_{1i}, T_{ID,t} >$ is added in the list L_1.

- H_2 *queries*: Let L_2 consist of tuples of the form $< w_j, x_j, v_j, \mu_j, \mathbf{c}_j >$. Upon receiving a query with (w_j, v_j, x_j, μ_j) from A, C produces a response to this query as follows.

1. Search (w_j, v_j, x_j, μ_j) in L_2. If it is found, the same answer in L_2 is returned to A because the query has been ever issued.
2. Otherwise, randomly select $\mathbf{c}_j \in Z_q^N$. Then $\mathbf{c}_j$ is sent to A and $< w_j, v_j, x_j, \mu_j, \mathbf{c}_j >$ is added in the list L_2.

- *Secret value queries*: Let L_S consist of tuples of the form $< ID_i, S_{ID_i}, R_{ID_i} >$. Upon receiving a query with ID_i from A, C produces a response to this query as follows.

1. Search ID_i in L_S. If it is found, the same answer in L_S is returned to A because the query has been ever issued.
2. Otherwise, randomly select $\mathbf{s}_{i5}, \mathbf{s}_{i6} \in \{-d, \ldots, 0, \ldots, d\}$, where $1 \le d \le 31$, and compute the polynomial $R_{ID_i} = a_1 * \mathbf{s}_{i5} + a_2 * \mathbf{s}_{i6}$. Then $S_{ID_i} = (\mathbf{s}_{i5}, \mathbf{s}_{i6})$ is sent to A and $< ID_i, S_{ID_i}, R_{ID_i} >$ is added in L_S.

- *Public key queries*: A issues this query along with ID_i, C produces a response to this query as follows.

1. Search ID_i in L_0 and L_S. If it is found, which means that the query has been ever issued, then C returns A with the same answer $PK_{ID_i} = (P_{ID_i}, R_{ID_i})$, where P_{ID_i} and R_{ID_i} are taken from L_0 and L_S, respectively.
2. Otherwise, issue the H_0 queries and *Secret value queries* to obtain P_{ID_i} and R_{ID_i}. Then $PK_{ID_i} = (P_{ID_i}, R_{ID_i})$ is sent to A.

- *Public key replacement queries*: A issues this query along with a new public key $PK'_{ID_i} = (P'_{ID_i}, R'_{ID_i})$ of ID_i to replace the old public key $PK_{ID_i} = (P_{ID_i}, R_{ID_i})$, C replaces the P_{ID_i} in L_0 with P'_{ID_i} and the R_{ID_i} in L_S with R'_{ID_i}.

- *Sign queries*: Upon receiving a query from A along with (μ_j, ID_i, PK_{ID_i}) at time period t, where $PK_{ID_i} = (P_{ID_i}, R_{ID_i})$, the challenger C makes the following steps to produce a valid signature.

1. Search ID_i in L_0, L_1 and L_S, respectively, to obtain $< ID_i, D_{ID_i}, P_{ID_i} >$, $< ID_i, t, T_{1i}, T_{ID,t} >$ and $< ID_i, S_{ID_i}, R_{ID_i} >$.
2. Randomly choose $\mathbf{c}_j \in \{\mathbf{v}: \mathbf{v} \in \{-1, 0, 1\}^N, \|\mathbf{v}\|_1 \le \lambda\}$ and $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6 \in D_\sigma^N$ with $\|(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6)\| \le 2\sigma\sqrt{6N}$. Then, compute $w_j = \mathbf{z}_1 + h * \mathbf{z}_2 - P_{ID_i} * \mathbf{c}_j$, $v_j = \mathbf{z}_3 + h * \mathbf{z}_4 - T_{1i} * \mathbf{c}_j$ and $x_j = a_1 * \mathbf{z}_5 + a_2 * \mathbf{z}_6 - R_{ID_i} * \mathbf{c}_j$.
3. Add $< w_j, v_j, x_j, \mu_j, \mathbf{c}_j >$ in the list L_1 and send the signature $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6, \mathbf{c}_j)$ on

μ_j to A.

Finally, as in the proof of Theorem 1, the signature $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6, \mathbf{c}_j)$ is valid and can pass the verification.

- *Forgery*: After making all the queries needed, the adversary A forges a valid signature tuple $(\mathbf{z}_1^*, \mathbf{z}_2^*, \mathbf{z}_3^*, \mathbf{z}_4^*, \mathbf{z}_5^*, \mathbf{z}_6^*, \mathbf{c}^*)$ on message μ^* for ID^* at time period t^*.

When A successfully forges a valid signature $(\mathbf{z}_1^*, \mathbf{z}_2^*, \mathbf{z}_3^*, \mathbf{z}_4^*, \mathbf{z}_5^*, \mathbf{z}_6^*, \mathbf{c}^*)$, the challenger C uses the Forking lemma [37] and replays A with different hash value of H_2 *queries* to produce another valid signature $(\mathbf{z}_1', \mathbf{z}_2', \mathbf{z}_3', \mathbf{z}_4', \mathbf{z}_5', \mathbf{z}_6', \mathbf{c}')$ such that $\mathbf{c}^* \neq \mathbf{c}'$ by the same random type. Because $(\mathbf{z}_1^*, \mathbf{z}_2^*, \mathbf{z}_3^*, \mathbf{z}_4^*, \mathbf{z}_5^*, \mathbf{z}_6^*, \mathbf{c}^*)$ and $(\mathbf{z}_1', \mathbf{z}_2', \mathbf{z}_3', \mathbf{z}_4', \mathbf{z}_5', \mathbf{z}_6', \mathbf{c}')$ are two valid signatures for (μ^*, ID^*, PK_{ID^*}), we can obtain the equation

$$\begin{aligned} &H_2(\mathbf{z}_1^* + h * \mathbf{z}_2^* - P_{ID^*} * \mathbf{c}^*,\ \mathbf{z}_3^* + h * \mathbf{z}_4^* - T_{1i^*} * \mathbf{c}^*,\ a_1 * \mathbf{z}_5^* + a_2 * \mathbf{z}_6^* - R_{ID^*} * \mathbf{c}^*,\ \mu^*) \\ &= H_2(\mathbf{z}_1' + h * \mathbf{z}_2' - P_{ID^*} * \mathbf{c}',\ \mathbf{z}_3' + h * \mathbf{z}_4' - T_{1i^*} * \mathbf{c}',\ a_1 * \mathbf{z}_5^* + a_2 * \mathbf{z}_6^* - R_{ID^*} * \mathbf{c}',\ \mu^*), \end{aligned}$$

which reduces to

$$a_1 * \mathbf{z}_5^* + a_2 * \mathbf{z}_6^* - R_{ID^*} * \mathbf{c}^* = a_1 * \mathbf{z}_5' + a_2 * \mathbf{z}_6' - R_{ID^*} * \mathbf{c}'.$$

Since $R_{ID^*} = a_1 * \mathbf{s}_5 + a_2 * \mathbf{s}_6$, we arrive at

$$\begin{aligned} &a_1 * \mathbf{z}_5^* + a_2 * \mathbf{z}_6^* - (a_1 * \mathbf{s}_5 + a_2 * \mathbf{s}_6) * \mathbf{c}^* = a_1 * \mathbf{z}_5' + a_2 * \mathbf{z}_6' - (a_1 * \mathbf{s}_5 + a_2 * \mathbf{s}_6) * \mathbf{c}' \\ &a_1 * (\mathbf{z}_5^* - \mathbf{z}_5') + a_2 * (\mathbf{z}_6^* - \mathbf{z}_6') - a_1 * \mathbf{s}_5(\mathbf{c}^* - \mathbf{c}') - a_2 * \mathbf{s}_6(\mathbf{c}^* - \mathbf{c}') = 0 \\ &a_1 * (\mathbf{z}_5^* - \mathbf{z}_5' - \mathbf{s}_5(\mathbf{c}^* - \mathbf{c}')) + a_2 * (\mathbf{z}_6^* - \mathbf{z}_6' - \mathbf{s}_6(\mathbf{c}^* - \mathbf{c}')) = 0 \\ &(a_1,\ a_2) * (\mathbf{z}_5^* - \mathbf{z}_5' - \mathbf{s}_5(\mathbf{c}^* - \mathbf{c}'), \mathbf{z}_6^* - \mathbf{z}_6' - \mathbf{s}_6(\mathbf{c}^* - \mathbf{c}')) = 0 \end{aligned}$$

Let $(\mathbf{u}_1,\ \mathbf{u}_2) = (\mathbf{z}_5^* - \mathbf{z}_5' - \mathbf{s}_5(\mathbf{c}^* - \mathbf{c}'),\ \mathbf{z}_6^* - \mathbf{z}_6' - \mathbf{s}_6(\mathbf{c}^* - \mathbf{c}'))$.

If $\left\|(\mathbf{z}_5^* - \mathbf{z}_5',\ \mathbf{z}_6^* - \mathbf{z}_6')\right\| \leq 4\sigma\sqrt{2N}$ and $\left\|(\mathbf{s}_5,\ \mathbf{s}_6)\right\| \leq 2d\lambda\sqrt{2N}$ with overwhelming probability, we can obtain $\left\|(\mathbf{u}_1,\ \mathbf{u}_2)\right\| \leq 2d\lambda\sqrt{2N} + 4\sigma\sqrt{2N}$. As stated in Lemma 3, the distribution of $h = g/f$ is statistically close to the uniform distribution of R_q [35]. The SIS problem on NTRU lattice is to find a pair $(\mathbf{u}_1, \mathbf{u}_2) \in R_q^2$ such that $\mathbf{u}_1 + h * \mathbf{u}_2 = 0$ and $\left\|(\mathbf{u}_1, \mathbf{u}_2)\right\| \leq \beta$, where β is $2\lambda s\sqrt{2N} + 4\sigma\sqrt{2N}$. Since the adversary A does not know the system secret key **B** generated by $g, f \in R_q$ and has generated such a pair $(\mathbf{u}_1, \mathbf{u}_2)$, we say that the adversary A solves the SIS problem. According to the same probability analysis in [22], if the adversary A can break our lattice-based RCLS scheme with non-negligible probability ε. Then, we can construct an algorithm C to solve the SIS problem with non-negligible probability $(1 - 2^{-\omega(\log N)})\varepsilon$. □

6. Comparisons

To show the advantages of the proposed RCLS scheme, we make the comparisons between the previous schemes and ours. For convenience, we define the following notations to count the computational costs.

- T_s: The required time of performing a sampling operation D_σ.
- T_m: The required time of performing a multiplication operation.
- T_a: The required time of performing an addition/subtraction operation.

Table 1 demonstrates comparisons between Tian and Huang's CLS scheme ([30]) and the proposed RCLS scheme in terms of lattice type, public key setting, averting key escrow problem, private key size, signature length, the computation costs of signing and verifying processes. Tian and Huang's CLS scheme is constructed under the certificateless public key settings to solve the key escrow problem, but it does not address the revocation problem. Our RCLS scheme resolves both the revocation and key escrow problems. For the usage of Lattices, Tian and Huang's CLS scheme uses the GPV lattice in [20] to generate the private key of a user. It is worth mentioning, that the related parameters in Table 1 have the following relationships: $m_1 > 2Nlogq$, $m_2 > 64 + Nlogq$, $\hat{s}_1 = \sqrt{m_1}\omega(\sqrt{\log N})$, $\hat{s}_2 = \sqrt{m_2}\omega(\sqrt{\log N})$, $s = N^{5/2}\sqrt{2q}\omega(\sqrt{\log N})$, $\hat{\sigma} = 12\hat{s}\lambda m_1$, $\sigma = 12\lambda sN$. In Table 2, we choose concrete parameters: N = 512, $q = 2^{26}$, k = 512, d = 31, λ = 14, m_1 = 38,400, m_2 = 25,600 and make the comparisons of instances in bit-length. According to Tables 1 and 2, for both the private key size, signature length, the computation costs of signing and verifying processes, our RCLS scheme is better than Tian and Huang's CLS scheme. Our scheme adopts public channels to send the periodic time update keys.

Indeed, the signing processes of all three schemes mentioned above employ the same the rejection sampling technique in Lyubashevsky's scheme [22] to produce signatures. Here, lets discuss the rejection probability in the signing process. If the rejection probability is too large, the performance of generating signatures may be inefficient. In our scheme, the signer can produce a useful signature $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5, \mathbf{z}_6, \mathbf{c})$ with probability min $\left(D_\sigma^{6N}(\mathbf{z}) / MD_{\mathrm{v},\sigma}^{6N}(\mathbf{z}), 1\right)$. That is, the signer with probability $(1 - 2^{-100})/M$ may output a useful signature by Lemma 4. According to the specific parameters N = 512, $q \approx 2^{26}$, k = 512 and d = 31 in [22], the M value is about 7.4. Hence, the performance of signing process still remains efficiency.

Table 1. Comparisons among previously proposed RIBS, CLS schemes and ours.

Properties	Tian and Huang's CLS Scheme	Our RCLS Scheme
Lattice type	GPV lattice	NTRU lattice
Public-key setting	CLS	RCLS
Revocable functionality	No	Public channel
Averting key escrow problem	Yes	Yes
Private key size	$2m_1k\log(\hat{s}_1\sqrt{m_1}) + 2m_2k\log(\hat{s}_2\sqrt{m_2})$	$6N\log(s\sqrt{N})$
Signature length	$(m_1 + m_2)\log(12\hat{\sigma}) + \lambda(\log k+1)$	$6N\log(12\sigma) + \lambda(\log N+1)$
Computational cost of signing	$(m_1 + m_2)(T_s + 2NT_m + T_a)$	$6NT_s + 9N(T_m + T_a)$
Computational cost of verifying	$2N(m_1 + m_2)T_m + 2NT_a$	$7NT_m + 6NT_a$

Table2. Comparisons of concrete instances in bit-length.

Bit-length	Tian and Huang's CLS Scheme	Our RCLS Scheme
Private key size	595,222,811	127,749
Signature length	2,026,680	175,312

7. Conclusions

In this paper, we proposed the first provably secure RCLS scheme with a public channel over lattices, which possesses existential unforgeability against adaptive chosen-message attacks. Under the SIS assumption and in the random oracle model, we formally established the security of our lattice-based RCLS scheme for three types of adversaries, namely, outside adversary, honest-but-curious KGC and revoked user. By performance analysis and comparisons, we have demonstrated that the proposed lattice-based RCLS scheme is better than the previously proposed lattice-based CLS scheme, in terms of private key size, signature length, the security property and the revocation mechanism.

Acknowledgments: The authors would like to appreciate anonymous referees for their valuable comments and constructive suggestions. This research was partially supported by Ministry of Science and Technology, Taiwan, under contract no. MOST106-2221-E-018-007-MY2.

Author Contributions: For the research paper, Ying-Hao Hung and Yuh-Min Tseng proposed and designed the RCLS scheme with a public channel over lattices. Sen-Shan Huang presented the background about lattices. Three authors cooperatively proved the security of the proposed scheme. Ying-Hao Hung and Yuh-Min Tseng made performance comparisons.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shamir, A. Identity-Based cryptosystems and signature schemes. In Proceedings of the Cryptology 1984 (Crypto'84), Santa Barbara, CA, USA, 19–22 August 1984; Springer: New York, NY, USA, 1985; LNCS Volume 196, pp. 47–53.
2. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. In Proceedings of the Cryptology 2001 (Crypto'01), Santa Barbara, CA, USA, 19–23 August, 2001; Springer: New York, NY, USA, 2001; LNCS Volume 2139, pp. 213–229.
3. Al-Riyami, S.S.; Paterson, K.G. Certificateless public key cryptography. In Proceedings of the Advances in Cryptology (ASIACRYPT'03), Taipei, Taiwan, 30 November–4 December, 2003; Springer: New York, NY, USA, 2003; LNCS Volume 2894, pp. 452–473.
4. Al-Riyami, S.S.; Paterson, K.G. CBE from CL-PKE: A generic construction and efficient schemes. In Proceedings of the Public Key Cryptography (PKC'05), Les Diablerets, Switzerland, 23–26 January, 2005; Springer: New York, NY, USA, 2005; LNCS Volume 3386, pp. 398–415.
5. Libert, B.; Quisquater, J.J. On constructing certificateless cryptosystems from identity based encryption. In Proceedings of the Public Key Cryptography (PKC'06), New York, NY, USA, 24–26 April, 2006; Springer: New York, NY, USA, 2006; LNCS Volume 3958, pp. 474–490.
6. Huang, X.; Mu, Y.; Susilo, W.; Wong, D.; Wu, W. Certificateless signature revisited. In Proceedings of the Australasian Conference on Information Security and Privacy (ACISP'06), Melbourne, Australia, 3–5 July, 2006; Springer: New York, NY, USA, 2007; LNCS Volume 4586, pp. 308–322.
7. Hwang, Y.H.; Liu, J.K.; Chow. S.S.M. Certificateless public key encryption secure against malicious KGC attacks in the standard model. *J. Universal Comput. Sci.* **2008**, *14*, 463–480.
8. Chen, Y.C.; Tso, R.; Susilo, W.; Huang, X.; Horng, G. Certificateless signatures: Structural extensions of security models and new provably secure schemes. In *Cryptology ePrint Archiv*: *Report 2013/193*; IACR: Santa Barbara, CA, USA, 2013.
9. Hung, Y.H.; Huang, S.S.; Tseng, Y.M.; Tsai, T.T. Certificateless signature with strong unforgeability in the standard model. *Informatica* **2015**, *26*, 663–684.
10. Tseng, Y.M.; Tsai, T.T. Efficient revocable ID-based encryption with a public channel. *Comput. J.* **2012**, *55*, 475–486.
11. Tsai, T.T.; Tseng, Y.M.; Huang, S.S. Efficient revocable certificateless public key encryption with a delegated revocation authority. *Secur. Commun. Netw.* **2015**, *8*, 3713–3725.
12. Shen, L.; Zhang, F.; Sun, Y. Efficient revocable certificateless encryption secure in the standard model. *Comput. J.* **2014**, *57*, 592–601.
13. Sun, Y.; Zhang, F.; Shen, L.; A revocable certificateless signature scheme. *J. Comput.* **2014**, *9*, 1843–1850.
14. Tsai, T.T.; Huang, S.S.; Tseng, Y.M. Secure certificateless signature with revocation in the standard model. *Math. Probl. Eng.* **2014**, *2014*, 728591, doi:10.1155/2014/728591.
15. Hung, Y.H.; Tseng, Y.M.; Huang, S.S. A revocable certificateless short signature scheme and its authentication application. *Informatica* **2016**, *27*, 549–572.
16. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **1997**, 26, 1484–1509.
17. Bernstein, D.J. *Introduction to Post-Quantum Cryptography. Post-Quantum Cryptography*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–14.
18. Goldreich, O.; Goldwasser, S.; Halevi, S. Public-key cryptosystems from lattice reduction problems. In Proceedings of the Advances in Cryptology (CRYPTO'97), Santa Barbara, California, USA, 17–21 August, 1997; Springer: New York, NY, USA, 1997; LNCS Volume 1294, pp. 112–131.

19. Nguyen, P.; Regev, O. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. *J. Cryptol.* **2009**, *22*, 139–160.
20. Gentry, C.; Peikert, C.; Vaikuntanathan, V. How to use a short basis: Trapdoors for hard lattices and new cryptographic constructions. In Proceedings of the Annual Symposium on the Theory of Computing (STOC'08), Victoria, Canada, 17–20 May, 2008; ACM Press: New York, NY, USA, 2008; pp. 197–206.
21. Lyubashevsky, V. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Proceedings of the Advances in Cryptology (ASIACRYPT'09), Tokyo, Japan, 6–10 December 2009; Springer: New York, NY, USA, 2009; LNCS Volume 5912, pp. 598–616.
22. Lyubashevsky, V. Lattice signatures without trapdoors. In Proceedings of the Advances in Cryptology (EUROCRYPT'12), Cambridge, UK, 15–19 April, 2012; Springer: New York, NY, USA, 2012; LNCS Volume 7237, pp. 738–755.
23. Ruckert, M. Strongly unforgeable signatures and hierarchical identity-based signatures over lattices without random oracles. In Proceedings of the Post-Quantum Cryptography (PQC'10), Darmstadt, Germany, 25–28 May 2010; Springer: New York, NY, USA, 2010; LNCS Volume 6061, pp. 182–200.
24. Liu, Z.H.; Hu, Y.P.; Zhang, X.S.; Li, F. Efficient and strongly unforgeable identity-based signature scheme over lattices in the standard model. *Secur. Commun. Netw.* **2013**, *6*, 69–77.
25. Tian, M.; Huang, L. Efficient identity-based signature from lattices. In Proceedings of the IFIP International Information Security Conference (SEC'14), Marrakech, Morocco, 2–4 June 2014; IFIPAICT Volume 428, pp. 321–329.
26. Ducas, L.; Lyubashevsky, V.; Prest, T. Efficient identity-based encryption over NTRU lattices. In Proceedings of the Advances in Cryptology (ASIACRYPT'14), Kaohsiung, Taiwan, 7–11 December 2014; Springer: New York, NY, USA, 2014; LNCS Volume 8874, pp. 22–41.
27. Xiang, X. Adaptive secure revocable identity-based signature scheme over lattices. *Comput. Eng.* **2015**, *41*, 126–129.
28. Boldyreva, A.; Goyal, V.; Kumar, V. Identity-based encryption with efficient revocation. In Proceedings of the ACM Conference on Computer and Communications Security (ACM CCS'08), Alexandria, VA, USA , 27–31 October 2008; ACM Press: New York, NY, USA, 2008; pp. 417–426.
29. Hung, Y.H.; Tseng, Y.M.; Huang, S.S. Revocable ID-based signature with short size over lattices. *Secur. Commun. Netw.* **2017**, *2017*, 7571201, doi:10.1155/2017/7571201.
30. Tian, M.; Huang, L. Certificateless and certificate-based signatures from lattices. *Secur. Commun. Netw.* **2105**, *8*, 1575–1586.
31. Micciancio, D.; Regev, O. Worst-case to average-case reductions based on Gaussian measure. *SIAM J. Comput.* **2007**, *37*, 267–302.
32. Hoffstein, J.; Howgrave-Graham, N.; Pipher, J.; Silverman, J.; Whyte, W. Ntrusign: Digital signatures using the ntru lattice. In Proceedings of the Cryptographers' Track at the RSA Conference (CT-RSA'03), San Francisco, CA, USA, 13–17April, 2003; Springer: New York, NY, USA, 2003; LNCS Volume 2612, pp. 122–140.
33. Micciancio, D.; Peikert, C. Trapdoors for lattices: Simpler, tighter, faster, smaller. In Proceedings of the Advances in Cryptology (EUROCRYPT'12), Cambridge, UK, 15–19 April 2012; Springer: New York, NY, USA, 2012; LNCS Volume 7237, pp. 700–718.
34. Ajtai, M. Generating hard instances of lattice problems. In Proceedings of the ACM Symposium on Theory of Computing (STOC'96), Philadelphia, PA, USA, 22–24 May 1996; ACM Press: New York, NY, USA, 1996; pp. 99–108.
35. Stehle, D.; Steinfeld, R. Making NTRUEnrypt and NTRUSign as secure as standard worst-case problems over ideal lattices. In *Cryptology ePrint Archive*: *Report 2013/4*; IACR: Santa Barbara, CA, USA, 2013.
36. Lyubashevsky, V.; Peikert, C.; Regev, O. On ideal lattices and learning with errors over rings. In Proceedings of the Advances in Cryptology (EUROCRYPT'10), French Riviera, 30 May–3 June 2010; Springer: New York, NY, USA, 2010; LNCS Volume 6110, pp. 1–23.
37. Pointcheval, D.; Stern, J. Security arguments for digital signatures and blind signatures. *J. Cryptol.* **2000**, *13*, 361–396.