Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 702539, 18 pages
http://dx.doi.org/10.1155/2013/702539

Research Article

Efficient Lattice-Based Signcryption in Standard Model

Jianhua Yan,[1,2] Licheng Wang,[1] Lihua Wang,[3] Yixian Yang,[1] and Wenbin Yao[1]

[1] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *School of Information and Electric Engineering, Ludong University, Yantai 264025, China*
[3] *Network Security Research Institute, National Institute of Information and Communications Technology, 4-2-1 Nukui-Kitamachi, Koganei-shi, Tokyo 184-8795, Japan*

Correspondence should be addressed to Licheng Wang; wanglc2012@126.com

Received 25 June 2013; Revised 26 August 2013; Accepted 27 August 2013

Academic Editor: Wang Xing-yuan

Signcryption is a cryptographic primitive that can perform digital signature and public encryption simultaneously at a significantly reduced cost. This advantage makes it highly useful in many applications. However, most existing signcryption schemes are seriously challenged by the booming of quantum computations. As an interesting stepping stone in the post-quantum cryptographic community, two lattice-based signcryption schemes were proposed recently. But both of them were merely proved to be secure in the random oracle models. Therefore, the main contribution of this paper is to propose a new lattice-based signcryption scheme that can be proved to be secure in the standard model.

1. Introduction

In many situations, we need to simultaneously realize confidentiality, integrity, authentication, and non-repudiation. There are generally two approaches to accomplish this task: the signature-then-encryption approach and signcryption proposed by Zheng [1]. Compared with the former, signcryption can perform both signature and encryption simultaneously at a lower cost. Hence, the signcryption scheme is more appropriate in many environments such as smart cards, mobile communications, and electronic commerce. Up to date, many efficient signcryption schemes [2–6] have been designed based on various assumptions in number theory. However, the cryptography based on number theory has been seriously challenged due to the booming of quantum computation. Under this situation, many researchers make efforts to probe new cryptosystems based on new security fundamentals, such as quantum cryptography [7–9], chaos cryptography [10, 11], DNA cryptography [12], and so forth. However, as far as we know, there is no efficient signcryption schemes based on these new fundamentals. Therefore, we have to pay our attention to another new upsurging branch of modern cryptography—post-quantum cryptography, including lattice-based cryptography, code-based cryptography, hash-based cryptography, and multivariate cryptography [13].

Recently, Li et al. [14] (LMK12) and Wang et al. [15] (WHW12) have succeeded in designing signcryption schemes based on lattice. Lattice-based cryptography has been regarded as the most attractive option for resisting quantum attacks. Meanwhile, it has many important advantages. Firstly, the security of lattice-based cryptography is based on worst-case hardness of lattice problems, while the previous cryptography constructed on number theory is based on average-case hardness. Secondly, the main operations in a lattice-based cryptographic scheme are addition and multiplications over a moderate modulus (say not larger than 1024). Thus, taking a long-term look, lattice-based cryptosystems can be performed extremely rapid, compared to the currently used cryptosystems (such as RSA) in which the exponentiations over a huge modulus (say not less than 2^{1024}) are always involved.

However, both of Li et al.'s scheme and Wang et al.'s scheme are merely proved to be secure in the random oracle model. After the publication of Canetti's critical statement on provable security reduction based on random

Table 1: Performance comparison.

Versus schemes	Ciphertext length	Computational cost	Public/private keys size	Public parameter size	security
WHW12 [15]	↘	↘	↘	↘	↗
LMK12 [14]	≈	↘	↘	↗	↗

oracles (ROMs) [16], it is always an interesting practice to design/prove cryptographic schemes that are not based on ROMs. In this paper, we construct a lattice-based signcryption scheme and present its security reductions without using ROMs. Our original ideas can be formulated as follows. The lattice generated by [17] has advantages in small trapdoor and small public key, but its public key encryption scheme can only achieve CCA1 security. The challenger cannot reply the decryption queries for the ciphertext with the first tuple $\boldsymbol{\mu}$ identical to the first tuple $\boldsymbol{\mu}_0$ in the challenge ciphertext in phase two. Moreover, the ciphertext of [17] is malleable. One of the typical methods for transforming an encryption scheme from CCA1 to CCA2 is to make use of a one time strongly unforgeable signature to ensure the nonmalleability of ciphertexts. However, this method will increase the ciphertext length and encryption/decryption time. We set $\boldsymbol{\mu}$ to be the hash value $H(r, \boldsymbol{\sigma})$, where r is a random number but $\boldsymbol{\sigma}$ is the signature generated in the signcryption process. The domain of $\boldsymbol{\mu}$ is big enough such that the probability that the first tuple in the ciphertext generated normally is equal to $\boldsymbol{\mu}_0$ is negligible. Hence, the challenger can reply the decryption queries in phase two. Further, we use CCA security of the symmetric encryption and collision resistance of hash function H to prevent the malleability of ciphertext. In the proving process, the hash function H can be replaced with a chameleon hash function H_c, so the challenger can generate $\boldsymbol{\mu}_0$ to form challenge ciphertext. If there exists an adversary who can forge a valid ciphertext, he/she can find a collision of H_c. The probability for the above event is negligible according to [18], so our signcryption scheme can achieve CCA2 security. The strong unforgeability of the signcryption can be obtained by the strong unforgeability of the original signature. In summary, the proposed scheme is

(i) indistinguishable against *inner* adaptively chosen ciphertext attacks (IND-CCA2) under the learning with errors (LWE) assumption in the standard model,

(ii) strongly unforgeable against *inner* adaptively chosen message attacks (SUF-CMA) under small integer solution (SIS) assumption in the standard model.

Here, the term "inner" means that in the IND-CCA2 (resp., SUF-CMA) game, the sender (resp., receiver) who possesses the signing (resp., decryption) key is allowed to launch the corresponding attacks. Apparently, an "inner" attacker is much stronger than outer ones. Thus, the inner security of our proposal also implies its outer security. In addition, our scheme has the advantages both in computational cost and in public/private keys size. That is, our main contribution can be summarized in Table 1. In order to make the trapdoors to be consistent, we construct a chameleon hash function by using the new trapdoor technique of [17], that may be of independent interest. In fact, our chameleon hash function is similar with the one in [19]. Although the chameleon hash function in [19] can be used in our scheme, it will lead to use two different kinds trapdoors technique and reduce efficiency.

The rest of this paper is organized as follows. In Section 2, the necessary preliminaries on lattice-based cryptographic assumptions and algorithms are introduced. In Section 3, the security models of signcryption, including the IND-CCA2 game and SUF-CMA game, are reviewed. In Section 4, the main contribution, that is, the proposed lattice-based signcryption scheme is presented in detail, followed by the proof on its consistency. The security proofs are given in Section 5 and the performance comparisons are given in Section 6. Finally, the concluding remarks are given in Section 7.

2. Preliminaries

Throughout this paper, we denote the set of integers by $\mathbb{Z}$, residue class mod$q$ by $\mathbb{Z}_q$, the real numbers by $\mathbb{R}$, and real interval $[0, 1)$ by $\mathbb{T}$. The expression $\mathbb{Z}^n$ (resp. $\mathbb{Z}_q^n$, $\mathbb{R}^n$) denotes vectors space on $\mathbb{Z}$ (resp. $\mathbb{Z}_q$, $\mathbb{R}$) in which every vector has n elements. Similarly, the expression $\mathbb{Z}^{n\times m}$ (resp. $\mathbb{Z}_q^{n\times m}$, $\mathbb{R}^{n\times m}$) denotes matrice space on $\mathbb{Z}$ (resp. $\mathbb{Z}_q$, $\mathbb{R}$) in which every matrix has n rows and m columns. We denote the set $\{1, 2, \ldots, k\}$ by $[k]$, for an integer $k > 0$. The symbol "|" denotes strings concatenation operators and "‖" denotes matrice concatenation operators. The vectors are denoted by lower-case and bold letters (e.g., $\mathbf{x}$), matrices by upper-case and bold letters (e.g., $\mathbf{A}$), and the Gram-Schmidt orthogonalization of $\mathbf{A}$ by $\widetilde{\mathbf{A}}$. The order for a matrix's column vectors can be interchangeable. The function $s_1(\cdot)$ denotes the largest singular value of a matrix. For a given distribution $\boldsymbol{\chi}$ over space $\mathscr{P}$, we use $\mathbf{s}\leftarrow_{\$}\boldsymbol{\chi}(\mathscr{P})$ to denote that s is picked at random from the space $\mathscr{P}$ according to the distribution $\boldsymbol{\chi}$. If the sampling space $\mathscr{P}$ is specified from the context, we also simply use $\mathbf{s} \in \boldsymbol{\chi}$ or $\mathbf{s}\leftarrow_{\$}\boldsymbol{\chi}$ to denote the same meaning. Also, we use $\mathbf{s}\leftarrow_{\$}U(P)$ to denote that s is picked at random from the space $\mathscr{P}$ according to the uniform distribution.

2.1. Lattice and Gaussian Distribution

Definition 1 (Lattice). An n-dimensional lattice $\boldsymbol{\Lambda}$ is a discrete additive subgroup of $\mathbb{R}^m$ ($m \geq n$). Formally, let $\mathbf{B} = \{\mathbf{b_1}, \mathbf{b_2}, \ldots, \mathbf{b_n}\}$ be n linearly independent vectors. The lattice generated by $\mathbf{B}$ is

$$\boldsymbol{\Lambda} = \mathscr{L}(\mathbf{B}) = \left\{\mathbf{Bx} = \Sigma_{i=1}^{n} x_i \mathbf{b_i} \mid \mathbf{x} \in \mathbb{Z}^n\right\}, \tag{1}$$

where $\mathbf{B}$ is called a basis for $\boldsymbol{\Lambda}$. In many cryptographic applications, a particular family which is called q-ary integer

lattices is frequently used. For positive integers n, m ($\geq n$) and q and matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the q-ary lattices are defined by

$$\begin{aligned} \Lambda^{\perp}(\mathbf{A}) &= \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A}\mathbf{z} = \mathbf{0} \bmod q\}, \\ \Lambda\left(\mathbf{A}^{\mathbf{t}}\right) &= \left\{\mathbf{z} \in \mathbb{Z}^m : \exists \mathbf{s} \in \mathbb{Z}_q^n \text{ s.t. } \mathbf{z} = \mathbf{A}^{\mathbf{t}}\mathbf{s} \bmod q\right\}. \end{aligned} \tag{2}$$

For integers $n > 0$ and $q > 2$, some probability distribution $\boldsymbol{\chi}$ over $\mathbb{Z}_q$ and a vector $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{A}_{\mathbf{s},\boldsymbol{\chi}}$ is defined as the distribution of $(\mathbf{a}, \mathbf{a}^{\mathbf{t}}\mathbf{s} + \mathbf{x})$ on $\mathbb{Z}_q^n \times \mathbb{Z}_q$, where $\mathbf{a}$ and $\mathbf{x}$ are chosen uniformly from $\mathbb{Z}_q^n$ and $\boldsymbol{\chi}$, respectively.

Definition 2 (Learning with Errors (LWE) [23]). For an integer $q = q(n)$ and a distribution $\boldsymbol{\chi}$ on $\mathbb{Z}_q$, the target of learning with errors $\mathrm{LWE}_{q,\boldsymbol{\chi}}$ is to distinguish with nonnegligible probability between the distribution $\mathbf{A}_{\mathbf{s},\boldsymbol{\chi}}$ and the uniform distribution on $\mathbb{Z}_q^n \times \mathbb{Z}_q$ by accessing the oracle for the given distribution, where $\mathbf{s} \leftarrow_{\$} U(\mathbb{Z}_q^n)$.

For $\alpha \in \mathbb{R}^+$, Ψ_α is defined as the distribution on $\mathbb{T}$ of a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$, reduced modulo 1. When normal variable x obeys distribution Ψ_α, $\overline{\Psi}_\alpha$ is the discretized normal distribution on $\mathbb{Z}_q$ of random variable $\lfloor q \cdot x \rceil \bmod q$, where $\lfloor\rceil$ denotes rounding.

Proposition 3 (hardness of LWE [23]). *Let $\alpha = \alpha(n) \in (0,1)$ and $q = q(n)$ be a prime to satisfy $\alpha q > 2\sqrt{n}$. If there is an efficient (possibly quantumn) algorithm that can solve $\mathrm{LWE}_{q,\overline{\Psi}_\alpha}$, then there is an efficient quantum algorithm for approximating SIVP_γ within $\widetilde{O}(n/\alpha)$ factors (referring to [24] for its hardness) in the worst case.*

Definition 4 (Small Integer Solution (SIS) [18]). Given an integer q, a real $\beta > 0$ and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the goal of $\mathrm{SIS}_{q,\beta}$ is to find a nonzero integer vector $\mathbf{z} \in \mathbb{Z}^m$ to satisfy $\mathbf{A}\mathbf{z} = \mathbf{0} \bmod q$ and $\|\mathbf{z}\| \leq \beta$.

Proposition 5 (hardness of SIS Theorem 5.16 [18]). *For any polybounded m, $\beta = poly(n)$, and for any prime $q \geq \beta \cdot \omega(\sqrt{n \log n})$, the average-case problem $\mathrm{SIS}_{q,\beta}$ is as hard as approximating the SIVP problem (among others) in the worst case to within certain $\gamma = \beta \cdot \widetilde{O}(\sqrt{n})$ factors.*

Definition 6 (Gaussian measure [18]). Given any vector $\mathbf{c}, \mathbf{x}$, and real $s > 0$, let

$$\rho_{s,\mathbf{c}}(\mathbf{x}) = e^{-\pi\|(\mathbf{x}-\mathbf{c})/s\|^2} \tag{3}$$

be a Gaussian function around $\mathbf{c}$ with parameter s. Its total measure is $\int_{\mathbf{x}\in\mathbb{R}^n} \rho_{s,\mathbf{c}}(\mathbf{x})d\mathbf{x} = s^n$. The probability density function of the corresponding continuous Gaussian distribution is defined as

$$\forall \mathbf{x} \in \mathbb{R}^n, \quad D_{s,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{s^n}. \tag{4}$$

When $\mathbf{c} = \mathbf{0}$, it is always omitted.

Definition 7 (discrete Gaussian distribution [18]). For any vector $\mathbf{c}$, real $s > 0$, and lattice $\boldsymbol{\Lambda}$, the distribution

$$\forall \mathbf{x} \in \boldsymbol{\Lambda}, \quad D_{\boldsymbol{\Lambda},s,\mathbf{c}}(\mathbf{x}) = \frac{D_{s,\mathbf{c}}(\mathbf{x})}{D_{s,\mathbf{c}}(\boldsymbol{\Lambda})} = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\boldsymbol{\Lambda})} \tag{5}$$

is called discrete Gaussian distribution over $\boldsymbol{\Lambda}$.

Proposition 8 (Claim 5.3 [23]). *Let $K > 1$ be a constant, and let $m > Kn \log q$ be an integer. The columns of a uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ generate all of $\mathbb{Z}_q^n$, except with $2^{-\Omega(n)}$ probability.*

Proposition 9. *Let $\mathbf{B}$ be a basis of $\Lambda^{\perp}(\mathbf{A})$, where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and the columns of $\mathbf{A}$ generate $\mathbb{Z}_q^n$. Let $s \geq \|\widetilde{\mathbf{B}}\|\omega(\sqrt{\log n})$.*

(1) *(Theorem 3.1 [20]) Let $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,s}$; the distribution of $\mathbf{y} = \mathbf{A}\mathbf{x} \in \mathbb{Z}_q^n$ is $\mathrm{negl}(n)$-far from the uniform distribution over $\mathbb{Z}_q^n$, and the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ is $D_{\Lambda_{\mathbf{y}}^{\perp}(A),s}$.*

(2) *(Lemma 4.4 [18]) $\Pr_{\mathbf{x}\sim D_{\Lambda,s,\mathbf{v}}}\{\|\mathbf{x} - \mathbf{v}\| > s\sqrt{n}\} \leq ((1+\epsilon)/(1-\epsilon)) \cdot 2^{-n}$.*

According to nonasymptotic theory of random matrices [25], we have the following lemma.

Proposition 10 (Lemma 2.9 [17]). *For any δ-sub-Gaussian with parameter s random matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ and any real $t > 0$, there is a constant $k > 0$, such that $s_1(\mathbf{A}) \leq ks \cdot (\sqrt{m} + \sqrt{n} + t)$ with at least probability $1 - 2^{\delta}e^{-\pi t^2}$.*

2.2. Universal Hashes and Chameleon Hashes. In general, we hope a hash function used in cryptographic schemes to be collision resistant. But in our construction, we need further assumptions on involved hashes. One is universal property and another is the so-called Chameleon property.

Definition 11 (universal hash functions [26]). We say that a family of hash functions $\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y}\}$ is universal if for every distinct pair $x, x' \in \mathcal{X}$, $\Pr_{h\leftarrow\mathcal{H}}[h(x) = h(x')] = 1/|\mathcal{Y}|$ holds.

In addition, a kind of specifical hash named chameleon hash introduced by Krawczyk and Rabin [27] is used in our work. The chameleon hash functions have the following four properties: (1) efficient forward computation, (2) standard collision-resistance property, (3) uniformity property, and (4) chameleon property. We will construct a chameleon hash family based on the lattice-trapdoor technique given in [17] and prove it has the above properties in Section 4.1; hence we do not describe these properties here in detail.

2.3. Related Algorithms for Inverting and Sampling. Micciancio and Peikert [17] proposed new, simpler, easy-to-implement and more efficient methods to generate and utilize "strong trapdoors" in cryptographic lattices. These methods include specialized algorithms for inverting LWE, which are important for encryption and signature.

Firstly, we introduce the related matrices. Let $\mathbf{g^t} = [1, 2, 4, \ldots, 2^{k-1}] \in \mathbb{Z}_q^{1\times k}$, $k \in \mathbb{N}$. Define matrix $\mathbf{S_k}$ as

$$\mathbf{S_k} = \begin{bmatrix} 2 & & & & q_0 \\ -1 & 2 & & & q_1 \\ & -1 & \ddots & & \vdots \\ & & \ddots & 2 & q_{k-2} \\ & & & -1 & q_{k-1} \end{bmatrix} \in \mathbb{Z}^{k\times k}. \quad (6)$$

The matrix $\mathbf{S_k}$ can be easily constructed in the following two cases: (1) when q is a power of 2, let $k = \lceil \log q \rceil$, $q_i = 0$ for $0 \le i \le k-2$, and $q_{k-1} = 2$; (2) when q is not a power of 2, q_i is the ith bit of q. In the former, $\|\widetilde{\mathbf{S_k}}\| = 2$ and in the latter $\|\widetilde{\mathbf{S_k}}\| < \sqrt{3}$ by Lemma 4.3 of [17]. It can be verified that $\mathbf{S_k}$ is a basis for $\mathbf{\Lambda}^{\perp}(\mathbf{g^t})$.

Let $g_{\mathbf{G}}(\mathbf{s}, \mathbf{e}) = s \cdot \mathbf{g^t} + \mathbf{e^t} \bmod q$. Given $\mathbf{b} \in \mathbb{Z}^{k-1} \bmod q$ there exist efficient algorithms to find $s \in \mathbb{Z}_q$ and $\mathbf{e} \in \mathbb{Z}^k$ such that $\mathbf{b^t} = g_{\mathbf{G}}(\mathbf{s}, \mathbf{e})$, when $\mathbf{e} \in [-q/4, q/4)^k = \mathcal{P}_{1/2}(q \cdot \widetilde{\mathbf{S_k}}^{-\mathbf{t}})$. There are two cases for q: q is a power of 2 or not. In the former case, Algorithm 1 can finish this task.

In the latter case, the above algorithm can work, but the interval for error vector $\mathbf{e}$ needs to be changed into $[-q/2\sqrt{3}, q/2\sqrt{3})^k = \mathcal{P}_{1/2}(q \cdot \widetilde{\mathbf{S_k}}^{-\mathbf{t}})$. For convenience, the algorithm for the latter case is also called **InvertG**.

The primitive vector $\mathbf{g^t}$ and the corresponding lattice $\mathbf{\Lambda}(\mathbf{g^t})$ basis $\mathbf{S_k}$ can be used to construct parity-check matrix $\mathbf{G}$ and matrix $\mathbf{S}$ as follows. It follows that $\mathbf{G}$ is a primitive matrix, and $\mathbf{S}$ is a basis for lattice $\mathbf{\Lambda}^{\perp}(\mathbf{G})$:

$$\mathbf{G} = \begin{bmatrix} \cdots \mathbf{g^t} \cdots & & & \\ & \cdots \mathbf{g^t} \cdots & & \\ & & \ddots & \\ & & & \cdots \mathbf{g^t} \cdots \end{bmatrix} \in \mathbb{Z}_q^{n\times nk},$$
$$\mathbf{S} = \begin{bmatrix} \mathbf{S_k} & & & \\ & \mathbf{S_k} & & \\ & & \ddots & \\ & & & \mathbf{S_k} \end{bmatrix} \in \mathbb{Z}_q^{nk\times nk}. \quad (7)$$

Definition 12. Given matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{T} \in \mathbb{Z}_q^{m_0\times kn}$ and invertible matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times n}$ for positive integers n, m, m_0, k, if $\mathbf{A}\begin{bmatrix} \mathbf{T} \\ \mathbf{I} \end{bmatrix} = \mathbf{MG}$ and $\|\mathbf{T}\|$ is small enough, $\mathbf{T}$ is called a $\mathbf{G}$-trapdoor of $\mathbf{A}$ corresponding to $\mathbf{M}$.

Given a function $g_{\mathbf{G}}(\widehat{\mathbf{s}}, \widehat{\mathbf{e}}) = \widehat{\mathbf{s}}^{\mathbf{t}}\mathbf{G} + \widehat{\mathbf{e}}^{\mathbf{t}} \bmod q$ with suitably small $\widehat{\mathbf{e}} \in \mathbb{Z}^{nk}$, an efficient oracle $\mathcal{O}(\mathbf{b} \in \mathbb{Z}_q^{nk})$ for inverting $g_{\mathbf{G}}(\widehat{\mathbf{s}}, \widehat{\mathbf{e}})$ can be achieved by calling Algorithm 1 for n times.

Given an LWE instance $\mathbf{b^t} = g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{s^t A} + \mathbf{e^t} \bmod q$ with suitably small $\mathbf{e} \in \mathbb{Z}^m$ and the $\mathbf{G}$-trapdoor $\mathbf{T}$ with corresponding matrix $\mathbf{M}$, Algorithm 2 can recover $\mathbf{s}$ and $\mathbf{e}$.

Finally, we recall the algorithms, denoted by SampleD and due to Peikert [28], for sampling from Gaussian distribution with short basis.

The mechanism of [17] for generating a trapdoor is different from that of [29]. As a result, it uses a new algorithm but also named SampleD to sample from a discrete Gaussian over $\Lambda_u^{\perp}(\mathbf{A})$ in [17], in which Algorithm 3 is called. It is used in signature and delegation. For distinction, let us call it SampleDG. The reader can refer to Theorem 5.5 of [17] for the correctness (Algorithm 4).

3. Signcryption: Primitive and Security Models

Signcryption was invented in 1996 but was first disclosed to the public at CRYPTO 1997 [1]. Signcryption is a public key cryptographic method that achieves unforgeability and confidentiality simultaneously with significantly smaller overhead than that required by "digital signature followed by public key encryption." It does this by signing and encrypting a message in a single step, fulfilling a cryptographer's dream to "kill two birds with one stone" [1, 3]. Signcryption techniques are now a global standard for data protection [30].

The primitive of signcryption provides confidentiality of the message against all entities except the intended receiver and meanwhile it provides the authenticity of the sender (i.e., the signer) for the intended receiver. It is clear that the authenticity embedded in the signcryption primitive is unidirectional, instead of bidirectional. In particular, if an intended receiver can forge a signature on behalf of some signer, he/she can plant some false evidence against the signer and then encrypt the signature for himself/herself. By doing so, the singer is incriminated. Therefore, in considering the security of signcryption, we should take into account the orthogonal combination of two kinds of attackers (i.e., inner attackers and outer attackers) and two protection goals (i.e., unforgeability and confidentiality). In 2005, Dent [31, 32] gave comprehensive elaborations on the inner security and outer security of signcryption. With the purpose of providing a handy consult for the security reduction given latter, we give a review on the security models of signcryption from LMK12 [14], in which we merely formulated the security models against inner attacker because in general an inner attacker is much stronger than an outer ones.

Definition 13 (signcryption). A signcryption scheme consists of the following four algorithms.

(i) **Setup**(1^n): this is an initialization algorithm that should be executed only once by any honest user in the system. It takes as input the security parameter 1^n and outputs the public parameters $\mathcal{P}_p$ that are shared by all users in the system.

(ii) **KeyGen**$(1^n, \mathcal{P}_p)$: this is a key generation algorithm that should be executed by each user only once. It takes as inputs the security parameter 1^n as well as the public parameters $\mathcal{P}_p$ and outputs the public/private key pair (PK, SK) where PK will be published publicly while SK will be kept known only to the user himself/herself. (In sequel, let us assume that the sender's public/private key pair is (PK_s, SK_s), while the receiver's is (PK_r, SK_r).)

(iii) **Signcrypt**$(\mathbf{u}, PK_r, PK_s, SK_s)$: this is a signcryption algorithm that should be executed by a senders whenever he/she wants to send a message to someone. It takes as inputs a message $\mathbf{u}$, the intended receiver's public key PK_r, and the sender's public/private key

Input:
Vector $\mathbf{b} \in \mathbb{Z}^k$, where $\mathbf{b}^{\mathbf{t}} = [b_0, \ldots, b_{k-1}]$.
Output:
Scalar $s \in \mathbb{Z}_q$ and a vector $\mathbf{e} \in \mathbb{Z}^k$.
(1) $s \leftarrow 0$;
(2) **for** $j = k - 1; k \geq 0;$ step -1 **do**
(3) **if** $(b_i - 2^i \cdot s) \in [-q/4, q/4) \bmod q$ **then**
(4) $t = 0$;
(5) **else**
(6) $t = 1$;
(7) **end if**
(8) $s = s + t \cdot 2^{k-1-j}$;
(9) $e_i = b_i - 2^i \cdot s \in [-q/4, q/4)$;
(10) **end for**
(11) Output $s \in \mathbb{Z}_q$ and $\mathbf{e} = (e_0, \ldots, e_{k-1}) \in [-q/4, q/4)^k \subset \mathbb{Z}^k$.

Algorithm 1: InvertG($\mathbf{b}$).

Input:
Parity-check matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$;
$\mathbf{G}$-trapdoor $\mathbf{T} \in \mathbb{Z}^{m_0 \times kn}$ of $\mathbf{A}$ and corresponding invertible tag $\mathbf{M_I} \in \mathbb{Z}_q^{n \times n}$;
Vector $\mathbf{b}^{\mathbf{t}} = g_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$ for suitably small $\mathbf{e} \in \mathbb{Z}^m$.
Output:
Vectors $\mathbf{s}$ and $\mathbf{e}$.
(1) Compute $\widehat{\mathbf{b}}^{\mathbf{t}} = \mathbf{b}^{\mathbf{t}} \begin{bmatrix} \mathbf{T} \\ \mathbf{I} \end{bmatrix}$;
(2) Obtain $(\widehat{\mathbf{s}}, \widehat{\mathbf{e}})$ by calling $\mathcal{O}(\widehat{\mathbf{b}})$;
(3) Returns $\mathbf{s} = \mathbf{M_I}^{-\mathbf{t}} \widehat{\mathbf{s}}$ and $\mathbf{e} = \mathbf{b} - \mathbf{A}^{\mathbf{t}}\mathbf{s}$.

Algorithm 2: Invert$^{\mathcal{O}}$ ($\mathbf{T}$, $\mathbf{A}$, $\mathbf{b}$, $\mathbf{M_I}$).

pair (PK_s, SK_s) and outputs a signcryption ciphertext $\mathbf{C}$.

(iv) **Unsigncrypt**($\mathbf{C}, PK_r, SK_r, PK_s$): this is a unsigncryption algorithm that should be executed by a receiver. It takes as inputs a signcryption ciphertext $\mathbf{c}$ and the receiver's public/private key pair (PK_r, SK_r), as well as the sender's public key PK_s, and outputs a plaintext $\mathbf{u}$ or $\perp$.

Definition 14 (consistency of signcryption). We say that a signcryption scheme defined above is consistent if the following probability

$$\Pr\left[\begin{array}{c} \mathscr{P}_p \longleftarrow \mathbf{Setup}(1^n); \\ (PK_r, SK_r) \longleftarrow \mathbf{KeyGen}(1^n, \mathscr{P}_p); \\ (PK_s, SK_s) \longleftarrow \mathbf{KeyGen}(1^n, \mathscr{P}_p); \\ \mathbf{C} \longleftarrow \mathbf{Signcrypt}(\mathbf{u}, PK_r, PK_s, SK_s); \\ \mathbf{u}' \longleftarrow \mathbf{Unsigncrypt}(\mathbf{C}, SK_r, PK_s) : \mathbf{u}' = \mathbf{u} \end{array}\right] = 1 - \epsilon(n) \tag{8}$$

is exponentially close to 1; that is, $\epsilon(n)$ is negligible with respect to n.

To capture the confidentiality of a signcryption defined above, we need to introduce a game, denoted by **Game IND-CCA2**, between a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$ as follows.

Game IND-CCA2

(i) Initial: $\mathscr{C}$ runs **Setup**(1^k) algorithm to produce public parameter $\mathscr{P}_p$ and then generates his/her own public/private keys (PK_r^*, SK_r^*) by running **KeyGen**($1^n, \mathscr{P}_p$) algorithm. Finally, $\mathscr{C}$ gives PK_r^* and $\mathscr{P}_p$ to $\mathscr{A}$.

(ii) Phase 1: $\mathscr{A}$ can perform polynomially bounded unsigncryption queries in an adaptive manner and $\mathscr{C}$ responds accordingly. More precisely, $\mathscr{A}$'s query is specified by a triple $(\mathbf{C}, PK_s)$ and $\mathscr{C}$'s responds with the corresponding plaintext $\mathbf{u}$ if $\mathbf{C}$ is a valid signcryption ciphertext with respect to the receiver's public key PK_r^* and sender's public key PK_s or $\perp$ otherwise.

(iii) Challenge: $\mathscr{A}$ chooses two equal length plaintexts $\mathbf{u}_0, \mathbf{u}_1$ and sends $(\mathbf{u}_0, \mathbf{u}_1, PK_s^*, SK_s^*)$ to $\mathscr{C}$, and $\mathscr{C}$ tosses a fair coin $b \in \{0, 1\}$ and sets $\mathbf{C}^* \leftarrow$ **Signcrypt**($\mathbf{u}_b, PK_r^*, PK_s^*, SK_s^*$). Finally, $\mathscr{C}$ sends $\mathscr{A}$ the challenged signcryption ciphertext $\mathbf{C}^*$.

Input:
Offline phase: Basis $\mathbf{B}$ of a q-ary integer $\Lambda = L(\mathbf{B})$; rounding parameter $r = \omega(\sqrt{\log n})$; positive definite covariance matrix $\Sigma \geq r^2 \cdot (4\mathbf{B}\mathbf{B}^{\mathbf{t}} + \mathbf{I})$;
Online phase: a vector $c \in \mathbb{Z}^n$;
Output:
Vector $\mathbf{v} \in \boldsymbol{\Lambda} + \mathbf{c}$ drawn from a distribution within $negl(n)$ statistical distance of $D_{\Lambda+c,\sqrt{\Sigma}}$;
Offline phase:
(1) Compute $\mathbf{Z} = q \cdot \mathbf{B}^{-\mathbf{1}} \in \mathbb{Z}^{n\times n}$;
(2) Let $\boldsymbol{\Sigma}_\mathbf{1} = 2r^2 \cdot \mathbf{B}\mathbf{B}^{\mathbf{t}}$, let $\boldsymbol{\Sigma}_\mathbf{2} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}_\mathbf{1} \geq r^2 \cdot (2\mathbf{B}\mathbf{B}^{\mathbf{t}}) + \mathbf{I}$, and compute some $\mathrm{M}\mathbf{B}_\mathbf{1} = \sqrt{\boldsymbol{\Sigma}_\mathbf{2} - r^2}$;
(3) Choose $\mathbf{y}$ from $D_{\mathbb{Z}^n,\sqrt{\Sigma_2}}$ by letting $\mathbf{y} \leftarrow \lfloor \mathbf{B}_\mathbf{1} \cdot D_1 \rceil_r$;
Online phase:
(4) Return $x \leftarrow \lfloor \mathbf{Z}(\mathbf{c} - \mathbf{y})/q \rceil_r$.

Algorithm 3: SampleD($\mathbf{B}$, c, r, $\boldsymbol{\Sigma}$).

Input:
Offline phase:
(i) Trapdoor matrix $\mathbf{T} \in Z_q^{m_0\times w}$;
(ii) Partial parity-check matrix $\mathbf{A}_\mathbf{0} \in \mathbb{Z}_q^{n\times m_0}$;
(iii) Positive definite $\boldsymbol{\Sigma} \geq \begin{bmatrix}\mathbf{T}\\ \mathbf{I}\end{bmatrix}(2 + \boldsymbol{\Sigma}_\mathbf{G})[\mathbf{T}^{\mathbf{t}} \parallel \mathbf{I}]$, for example, any $\boldsymbol{\Sigma} = s^2 \geq (s_1(\mathbf{T})^2 + 1)(s_1(\boldsymbol{\Sigma}_\mathbf{G}) + 2)$.
Online phase:
(i) Invertible tag $\mathbf{M}_\mathbf{I} \in \mathbb{Z}_q^{n\times m}$ defining $\mathbf{A} = [\mathbf{A}_\mathbf{0} \parallel \mathbf{M}_\mathbf{I}\mathbf{G} - \mathbf{A}_\mathbf{0}\mathbf{T}]$;
(ii) Syndrome $\mathbf{y} \in \mathbb{Z}_q^n$.
Output:
A vector $\mathbf{x}$, which is statistically close to $D_{\boldsymbol{\Lambda}(\mathbf{1}/\mathbf{y})(\mathbf{A}),r\sqrt{\boldsymbol{\Sigma}}}$.
Offline phase:
(1) Compute $\boldsymbol{\Sigma}_\mathbf{p} = \boldsymbol{\Sigma} - \begin{bmatrix}\mathbf{T}\\ \mathbf{I}\end{bmatrix}\boldsymbol{\Sigma}_\mathbf{G}[\mathbf{T}^{\mathbf{t}} \parallel \mathbf{I}] \geq \mathbf{2}\begin{bmatrix}\mathbf{T}\\ \mathbf{I}\end{bmatrix}[\mathbf{T}^{\mathbf{t}} \parallel \mathbf{I}]$, choose vector $\mathbf{p} \leftarrow D_{\mathbb{Z}^m,r\sqrt{\boldsymbol{\Sigma}_\mathbf{p}}}$;
(2) Parse $\mathbf{P} = \begin{bmatrix}\mathbf{P}_\mathbf{1}\\ \mathbf{P}_\mathbf{2}\end{bmatrix}$ where $\mathbf{p}_\mathbf{1} \in \mathbb{Z}^{m_0}$, $\mathbf{p}_\mathbf{2} \in \mathbb{Z}^w$. Compute $\mathbf{w}_\mathbf{0} = \mathbf{A}_\mathbf{0}(\mathbf{p}_\mathbf{1} - \mathbf{T}\mathbf{p}_\mathbf{2}) \in \mathbb{Z}_q^n$ and $\mathbf{w} = \mathbf{G}\mathbf{P}_\mathbf{2} \in \mathbb{Z}_q^n$;
Online phase:
(3) Compute $\mathbf{v} \leftarrow \mathbf{M}_\mathbf{I}^{-\mathbf{1}}(\mathbf{y} - \mathbf{w}_\mathbf{0}) - \mathbf{w} = \mathbf{M}_\mathbf{I}^{-\mathbf{1}}(\mathbf{y} - \mathbf{A}\mathbf{p}) \in \mathbb{Z}_q^n$;
(4) Choose $\mathbf{z} \leftarrow D_{\Lambda(\mathbf{1}/\mathbf{V})(\mathbf{G}),r\sqrt{\boldsymbol{\Sigma}_\mathbf{G}}}$ by calling SampleD($\mathbf{S}_\mathbf{k}$, $\mathbf{v}$, r, $\sqrt{\boldsymbol{\Sigma}_\mathbf{G}}$) see Algorithm 3;
(5) Return $\mathbf{x} = \mathbf{p} + \begin{bmatrix}\mathbf{T}\\ \mathbf{I}\end{bmatrix}\mathbf{z}$.

Algorithm 4: SampleDG($\mathbf{T}$, $\mathbf{A}_\mathbf{0}$, $\mathbf{M}_\mathbf{I}$, $\mathbf{y}$, s).

(iv) Phase 2: phase 1 is repeated with the restriction that $\mathcal{A}$ is not allowed to ask unsigncryption query on triple $(\mathbf{C}^*, PK_s^*)$.

(v) Guess: $\mathcal{A}$ outputs a bit b' as his/her guessing on b.

Then, the advantage of $\mathcal{A}$ to win **Game IND-CCA2** is defined as $\mathrm{Adv}(\mathcal{A}) = |\Pr[b = b'] - (1/2)|$.

Definition 15 (confidentiality of signcryption). A signcryption scheme is said to be indistinguishable against inner chosen ciphertext attacks (IND-CCA2), if there is no probabilistic polynomial time adversary that can win **Game IND-CCA2** with nonnegligible advantage.

To capture the (strong) unforgeability of a signcryption defined above, we need to introduce another game, denoted by **Game SUF-CMA**, between a challenger $\mathscr{C}$ and a forgery $\mathscr{F}$ as follows.

Game SUF-CMA

(i) Initial: $\mathscr{C}$ runs **Setup**(1^k) algorithm to produce public parameter $\mathscr{P}_p$ and then generates his/her own public/private keys (PK_s^*, SK_s^*) by running **KeyGen**(1^n, $\mathscr{P}_p$) algorithm. Finally, $\mathscr{C}$ gives PK_s^* and $\mathscr{P}_p$ to $\mathscr{F}$.

(ii) Singcrypt query: $\mathscr{F}$ can perform polynomially bounded signcryption queries in an adaptive manner. More precisely, $\mathscr{F}$'s query is specified by a pair $(\mathbf{u}, PK_r)$ and $\mathscr{C}$'s responds with $\mathbf{C} \leftarrow$ **Signcrypt**($\mathbf{u}$, PK_r, PK_s^*, SK_s^*). (Here, PK_r is the intended receiver's public key and the corresponding private key SK_r is known to $\mathscr{F}$. Furthermore, $\mathscr{F}$ is

allowed to either obtain (PK_r, SK_r) by calling the algorithm $\mathbf{KeyGen}(1^n, \mathscr{P}_p)$ or pick them randomly.)

(iii) Forgery: $\mathscr{F}$ outputs a tuple $(\mathbf{u}^*, \mathbf{C}^*, PK_r^*, SK_r^*)$ with the restriction that $\mathscr{C}$ never responds to $\mathscr{F}$ with $\mathbf{C}^*$ for answering $\mathscr{F}$'s signcryption query on $(\mathbf{u}^*, PK_r^*)$.

Then, the advantage of $\mathscr{F}$ to win **Game SUF-CMA** is defined as

$$\mathrm{Adv}(\mathscr{F}) = \Pr\left[\mathbf{u}^* = \mathbf{Unsigncrypt}\left(\mathbf{C}^*, PK_r^*, SK_r^*, PK_s^*\right)\right]. \tag{9}$$

Definition 16 (strong unforgeability of signcryption). A signcryption scheme is said to be strongly unforgeable against inner adaptively chosen message attacks (SUF-CMA), if no probabilistic polynomial time adversary can win **Game SUF-CMA** with nonnegligible advantage.

4. Proposed Lattice-Based Signcryption Scheme

In this section, we firstly present a chameleon hash function based on the lattice-trapdoor technique given in [17]. Next, based on the signature scheme and the encryption scheme given in [17], we propose a signcryption scheme. Finally, we prove the consistency of the proposed scheme. Note that, the matrices $\mathbf{G}, \mathbf{S}, \mathbf{S_k}$ used in this section are as in Section 2.3.

4.1. Building Block: Lattice-Based Chameleon Hash Functions. According to [33, 34], we know that by using a chameleon hash function, one can transfer an SUF-SMA secure signature scheme to an SUF-CMA secure one. To guarantee the consistency of the proposed scheme, we need to construct a chameleon hash function based on lattice-based trapdoors of given in [17]. In fact, it is a analogue to the scheme based on the trapdoors given in [29].

Let $n \geq 1$, $q \geq 2$, $m_0 = O(n \log q)$, and $k = O(\log q)$ be integers. Let integer $l > 0$ be message length. For matrices $\mathbf{N^{(0)}} \in \mathbb{Z}_q^{n\times l}$, $\mathbf{N^{(z)}} \in \mathbb{Z}_q^{n\times m_0}$, and $\mathbf{T} \sim D^{m_0\times nk}_{\mathbb{Z},\omega(\sqrt{\log n})}$ and invertible matrix $\mathbf{M} \in \mathbb{Z}_q^{n\times n}$ construct $\mathbf{N^{(1)}} = [\mathbf{N^{(z)}} \parallel \mathbf{N^{(y)}}] = [\mathbf{N^{(z)}} \parallel -\mathbf{N^{(z)}T} + \mathbf{MG}] \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{N} = [\mathbf{N^{(0)}} \parallel \mathbf{N^{(1)}}]$. Let s be a Gaussian parameter satisfying $s \approx \sqrt{5}s_1(\mathbf{T})\omega(\sqrt{\log n})$ suggested by [17]. Define a message space $\mathscr{M} = \{0,1\}^l$ and random space $\mathscr{R} = D_{\mathbb{Z}^m,s}$ (then, for $\mathbf{r} \in \mathscr{R}$, $\|\mathbf{r}\| \leq s\sqrt{m}$ holds with overwhelming probability according to Proposition 9 items (2)) and $\mathscr{Y} \in z_q^n$. For $\mathbf{u} \in \mathscr{M}$ and $\mathbf{r} \in \mathscr{R}$, using matrix $\mathbf{N}$ define hash function $H_{\mathbf{N}} : \mathscr{M} \times \mathscr{R} \to \mathscr{Y}$ as

$$H_{\mathbf{N}}(\mathbf{u}, \mathbf{r}) = \mathbf{N^{(0)}u} + \mathbf{N^{(1)}r}. \tag{10}$$

Lemma 17. *The family $\mathscr{H} = \{H_{\mathbf{N}}\}$ (under the uniform distribution over $\mathscr{H}$) is a family of chameleon hash functions, assuming the hardness of $SIS_{q,\beta}$ for $\beta^2 = l + 4s^2m$.*

Proof. It is enough to prove the hash family $\mathscr{H} = \{H_{\mathbf{N}}\}$ has the four properties described in Section 2.2.

For efficient forward computation. Clearly, given a message $\mathbf{u} \in \mathscr{M}$ and $\mathbf{r} \in \mathscr{R}$, each $H_{\mathbf{N}}(\mathbf{u}, \mathbf{r})$ is efficiently computable.

For collision-resistance property. Assuming that it is easy to find a collision $[\mathbf{u} \parallel \mathbf{r}] \neq [\mathbf{u}' \parallel \mathbf{r}'] \in \mathscr{M} \times \mathscr{R}$ for $H_{\mathbf{N}}$, then $\mathbf{x} = [\mathbf{u} - \mathbf{u}' \parallel \mathbf{r} - \mathbf{r}'] \neq \mathbf{0}$ is a solution for $\mathbf{Nx} = \mathbf{0}$, and according to the triangle inequality, we have that $\|\mathbf{x}\|^2 \leq l + 4s^2m$. It implies that $\mathbf{x}$ is also a solution for the instance $\mathbf{N}$ of $\mathrm{SIS}_{q,\beta}$. This contradicts the hardness of $\mathrm{SIS}_{q,\beta}$ for $\beta = \sqrt{l + 4s^2m}$. Therefore, the hash family is collision-resistant.

For uniformity property, we first show the matrix $\mathbf{N}$ is uniform. The matrix $\mathbf{N^{(z)}}$ is uniform, so $\mathbf{N^{(z)}T}$ is also uniform when $\mathbf{T} \sim D^{m_0\times nk}_{\mathbb{Z},\omega(\sqrt{\log n})}$ is $\mathrm{negl}(n)$-far from uniform (cf. Section 6.2 of [17]). On the other hand, the matrix $\mathbf{MG}$ is fixed when $\mathbf{M}$ is fixed. Consequently, the matrix $\mathbf{N^{(1)}}$ is $\mathrm{negl}(n)$-far from uniform. On the other hand, $\mathbf{N^{(0)}}$ is uniform; hence $\mathbf{N}$ is $\mathrm{negl}(n)$-far from uniform. It is clear that given any $m \in \mathscr{M}$ and $\mathbf{r} \leftarrow D_{\mathbb{Z}^m,s}$ and each matrix $\mathbf{N}$ generated as above, the distribution of $H_{\mathbf{N}}(m, r)$ is negligible far from the uniform distribution over $\mathscr{H} \times \mathscr{Y}$ by Proposition 9 items (1).

For chameleon property. Given $\mathbf{u}, \mathbf{u}' \in \mathscr{M}$ and $\mathbf{r} \in \mathscr{R}$, one with $\mathbf{G}$-trapdoor $\mathbf{T}$ can easily find $\mathbf{r}' \in \mathscr{R}$ satisfying $H_{\mathbf{N}}(\mathbf{u}, \mathbf{r}) = H_{\mathbf{N}}(\mathbf{u}', \mathbf{r}')$ as follows: compute $\mathbf{y} = H_{\mathbf{N}}(\mathbf{u}, \mathbf{r}) - \mathbf{N^{(0)}u}'$, and then sample preimage $\mathbf{r}' = \mathrm{SampleDG}(\mathbf{T}, \mathbf{N^{(z)}}, \mathbf{M}, \mathbf{y}, s)$. □

Lemma 18. *The above chameleon hash family is universal; for every distinct $\mathbf{u}, \mathbf{u}' \in \mathscr{M}$ and distinct $\mathbf{r}, \mathbf{r}' \in \mathscr{R}$, $Pr_{H_{\mathbf{N}}\leftarrow\mathscr{H}}[H_{\mathbf{N}}(\mathbf{u}, \mathbf{r}) = H_{\mathbf{N}}(\mathbf{u}', \mathbf{r}')] = q^{-n}$.*

Proof. Assuming that $H_{\mathbf{N}}(\mathbf{u}, \mathbf{r}) = H_{\mathbf{N}}(\mathbf{u}', \mathbf{r}')$, it follows that $\mathbf{N^{(1)}r}' = H_{\mathbf{N}}(\mathbf{u}, \mathbf{r}) - \mathbf{N^{(0)}u}'$. When $\mathbf{u}, \mathbf{r}, \mathbf{u}'$ is fixed, the vector $\mathbf{z} = H_{\mathbf{N}}(\mathbf{u}, \mathbf{r}) - \mathbf{N^{(0)}u}'$ is a fixed element in $\mathbb{Z}_q^n$.

The matrix $\mathbf{N^{(1)}}$ is uniform as described above. For $m_0 = O(n \log q)$, $m = m_0 + nk \geq 2n \log q$, the columns of $\mathbf{N^{(1)}} \in \mathbb{Z}_q^{n\times m}$ generate $\mathbb{Z}_q^n$ with overwhelming probability by Proposition 8. In addition, $s \geq \|\widetilde{\mathbf{S}}\|\omega(\sqrt{\log n})$ and $\mathbf{r}' \sim D_{\mathbb{Z}^m,s}$, where $\mathbf{S}$ is as in Section 2.3.

It follows that $\mathbf{N^{(1)}r}'$ is uniform over $\mathbb{Z}_q^n$ (up to negligible statistical distance) by Proposition 9 items (1). Consequently, $\Pr[\mathbf{N^{(1)}r}' = \mathbf{z}] = q^{-n}$. In other words, $\Pr_{H_{\mathbf{N}}\leftarrow\mathscr{H}}[H_{\mathbf{N}}(\mathbf{u}, \mathbf{r}) = H_{\mathbf{N}}(\mathbf{u}', \mathbf{r}')] = q^{-n}$. □

4.2. Signcryption Scheme. In [17], Micciancio and Peikert gave a special collector (MP collector) that maps elements from a certain ring $\mathfrak{R}$ into matrices $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ as required by their trapdoor construction. Let us call it MP Collector and denote it by $\hbar : \mathfrak{R} \to \mathbb{Z}_q^{n\times n}$. Given a monic degree-$n$ irreducible polynomial $f(x) = x^n + f_{n-1}x^{n-1} + \cdots + f_0 \in \mathbb{Z}[x]$, a ring can be defined as $\mathfrak{R} = \mathbb{Z}[x]/(f(x))$ and the elements of $\mathfrak{R}$ can be represented as vectors in $\mathbb{Z}^n$ relative to the standard basis of monomials $1, x, \ldots, x^{n-1}$. Now, given a ring element $a(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1} \in \mathfrak{R}$, $\hbar(\mathbf{a})$ can be constructed as follows:

$$\begin{aligned} \mathbf{h^{(0)}} &= (a_0, a_1, \ldots, a_{n-1}), \\ \mathbf{h^{(i)}} &= \left(0, h_0^{(i-1)}, \ldots, h_{n-2}^{(i-1)}\right) - h_{n-2}^{(i-1)}\left(f_0, f_1, \ldots, f_{n-1}\right), \end{aligned} \tag{11}$$

for $0 \le i < n$, where $\mathbf{h}^{(\mathbf{i})}$ is the ith column of $\hbar(\mathbf{a})$. Clearly, $\hbar$ has the following properties. Firstly, $\hbar$ is a ring homomorphism, namely, $\hbar(\mathbf{x_1}) - \hbar(\mathbf{x_2}) = \hbar(\mathbf{x_1} - \mathbf{x_2})$ for $\mathbf{x_1}, \mathbf{x_2} \in \mathfrak{R}$. Secondly, multiplication by a ring element $\mathbf{a} \in \mathfrak{R}$ can be represented by the matrix $\hbar(\mathbf{a})$; furthermore, the product coefficients vector equals $\sum_{i=0}^{n-1} b_i \mathbf{h}_{\mathbf{a}}^{(\mathbf{i})}$, where $\mathbf{h}_{\mathbf{a}}^{(\mathbf{i})}$ is the ith column of $\hbar(\mathbf{a})$ and b_i is the ith coefficient of ring element. Thirdly, $\hbar(\mathbf{x}) \in GL_n(q)$ if and only if $\mathbf{x}$ is a unit of $\mathfrak{R}$, where $GL_n(q)$ is a group composed of the invertible elements in $\mathbb{Z}_q^{n\times n}$. Finally, the ring $\mathfrak{R}$ has "units difference" property, namely, for any $\mathbf{x_i}, \mathbf{x_j} \in \mathfrak{R}^*$ ($\mathfrak{R}^*$ denotes the units set in $\mathfrak{R}$), $\mathbf{x_i} - \mathbf{x_j} \in \mathfrak{R}^*$.

Our signcryption scheme consists of the following four algorithms. Note that we also adopt a symmetrical encryption scheme $\Pi = (\mathscr{K}, \mathscr{E}, \mathscr{D})$ (with keyspace $\mathscr{K}$, encryption algorithm $\mathscr{E}$, and decryption algorithm $\mathscr{D}$) in our construction.

(i) **Setup**(1^n): Suppose the security parameter is 1^n. Then, the the public parameters $\mathscr{P}_p$ for the system can be specified as follows.

(1) $\mathbf{G} \in \mathbb{Z}_q^{n\times nk}$ is the matrix as defined in Section 2.3, where $q = p^e = \text{poly}(n)$ is a prime power and is large enough (cf. [17]).
(2) $k = O(\log q) = O(\log n)$.
(3) $m_0 = O(nk)$, $m = m_0 + nk$, $m_1 = m_0 + 2nk$, and suitable $l > 0$.
(4) $\alpha \in (0, 1)$ is an LWE error rate, such that $1/\alpha = O(nk) \cdot \omega(\sqrt{\log n})$.
(5) $s' \ge \sqrt{5} \cdot \omega(\sqrt{\log n})$.
(6) A monic degree-n irreducible polynomial $f(x) = x^n + f_{n-1}x^{n-1} + \cdots + f_0 \in \mathbb{Z}_p[x]$ and a ring defined as $\mathfrak{R} = \mathbb{Z}_q[x]/(f(x))$.
(7) 4 hash functions:
   (a) $H_0 : \{0,1\}^* \times \mathbb{Z}_q^{n\times m} \to \{0,1\}^\lambda$ is a collision-resistant hash function, where $\lambda < n(p-1)$;
   (b) $H_1 : \mathbb{Z}_q^{m_1} \to \{0,1\}^l$ is a universal hash function;
   (c) $H_{\mathbf{F}} : \{0,1\}^l \times \mathbb{Z}_q^m \to \mathfrak{R}^*$ is chosen from a universal family, where $\mathbf{F}$ is a matrix in $\mathbb{Z}_q^{n\times(l+m)}$. More precisely, $\mathbf{F} = [\mathbf{F}^{(0)} \parallel \mathbf{F}^{(1)}] \in \mathbb{Z}_q^{n\times l} \times \mathbb{Z}_q^{n\times m}$, and $\mathbf{F}^{(0)}$ (resp., $\mathbf{F}^{(1)}$) has the same distribution with $\mathbf{N}^{(0)}$ (resp., $\mathbf{N}^{(1)}$) described in Section 4.1 (note that the elements in $\mathfrak{R}$ can be represented by vectors in $\mathbb{Z}_q^n$);
   (d) $H_2 : \{0,1\}^{nk} \times \mathbb{Z}_{2q}^m \to \{0,1\}^{n\cdot k'}$ is a universal hash function with suitably specified k'.
(8) An ordered matrix set $\mathfrak{B} = \{\mathbf{B}^{(0)}, \ldots, \mathbf{B}^{(\lambda)}\}$ where $\mathbf{B}^{(i)} \leftarrow_\$ U(\mathbb{Z}_q^{n\times nk})$ for $0 \le i \le \lambda$.
(9) $s_s = O(\sqrt{\lambda nk})\omega(\sqrt{\log n})^2$ is the Gaussian parameter for signature.
(10) An arbitary basis $\mathbf{Q} \in \mathbb{Z}^{nk\times nk}$ for $\mathbf{\Lambda} = \mathbf{\Lambda}(\mathbf{G^t})$.
(11) MP Collector $\hbar : \mathfrak{R} \to \mathbb{Z}_q^{n\times n}$.

(ii) **KeyGen**$(1^n, \mathscr{P}_p)$: any user can generate his/her public key PK and private key SK as follows.

(a) Sample $\mathbf{A}^{(0)} \leftarrow_\$ \mathbb{Z}_q^{n\times m_0}$ and $\mathbf{T} \leftarrow_\$ D_{\mathbb{Z},\omega(\sqrt{\log n})}^{m_0\times nk}$;
(b) Evaluate $\mathbf{A}^{(1)} = -\mathbf{A}^{(0)}\mathbf{T}$;
(c) Let $PK \triangleq \mathbf{A} = [\mathbf{A}^{(0)} \parallel \mathbf{A}^{(1)}] \in \mathbb{Z}_q^{n\times m}$ and $SK \triangleq \mathbf{T}$.

(iii) **Signcrypt**$(\mathbf{u}, A_r, A_s, T_s)$: a sender with public/private key pair $(\mathbf{A_s}, \mathbf{T_s})$ can send a signcryption ciphertext $\mathbf{c}$ on some message $\mathbf{u}$ to a receiver with public key $\mathbf{A_r}$ as follows.

(1) Sign message $\mathbf{u}$ to obtain $(\boldsymbol{\sigma}, \mathbf{r_1})$, as follows.
   (a) Compute $\mathbf{h} = H_0(\mathbf{u}, \mathbf{A_r})$, $\mathbf{A}' = \mathbf{B}^{(\mathbf{0})} + \sum_{i\in[\lambda]} \mathbf{h}_i \mathbf{B}^{(\mathbf{i})}$ and then build
   $$\mathbf{A}_{\mathbf{s}}^{(\mathbf{h})} = \left[\mathbf{A_s} \parallel \mathbf{A}'\right] \in \mathbb{Z}^{n\times m_1}, \tag{12}$$
   where $\mathbf{h} = (h_1, \ldots, h_\lambda) \in \{0,1\}^\lambda$ is the binary representation of $\mathbf{h}$.
   (b) Construct chameleon hash function according to the method in Section 4.1. Concretely, replace $\mathbf{N}^{(1)}, \mathbf{N}^{(0)}, \mathbf{M}$ with $\mathbf{A_s}$, arbitrary l columns of $\mathbf{A}'$, $\mathbf{I} \in \mathbb{Z}_q^{n\times n}$, respectively. The others are invariant. The hash function is denoted as $H_{\mathbf{A}_{\mathbf{s}}^{(\mathbf{h})}}$.
   (c) Sample $\mathbf{r_1} \leftarrow_\$ D_{\mathbb{Z}^{nk},s_s}$.
   (d) Compute $\mathbf{h_r} = H_{\mathbf{A}_{\mathbf{s}}^{(\mathbf{h})}}(\mathbf{h}, \mathbf{r_1})$.
   (e) Sample $\mathbf{y} \leftarrow_\$ D_{\mathbb{Z}^{nk},s_s}$ and compute $\mathbf{y}' = \mathbf{h_r} - \mathbf{A}'\mathbf{y}$.
   (f) Sample $\boldsymbol{\sigma}_{\mathbf{or}} \leftarrow \text{SampleDG}(\mathbf{T_s}, \mathbf{A}_{\mathbf{s}}^{(\mathbf{0})}, \mathbf{I}, \mathbf{y}', s_s)$, where $\mathbf{I} \in \mathbb{Z}^{n\times n}$ is identity matrix.
   (g) Let $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_{\mathbf{or}}, \mathbf{y})$.
(2) Parse $\boldsymbol{\sigma} = (\boldsymbol{\sigma_1}, \boldsymbol{\sigma_2})$ such that $\boldsymbol{\sigma_1} \in \{0,1\}^{nk}$ and denote the remainder as $\boldsymbol{\sigma_2}$.
(3) Compute $\mathbf{w} = H_1(\boldsymbol{\sigma})$.
(4) Encrypt $\boldsymbol{\sigma_1}$ as follows.
   (a) Sample $\mathbf{e_0} \leftarrow_\$ D_{\mathbb{Z},\alpha q}^{m_0}$ and evaluate $s_r = \sqrt{\|\mathbf{e_0}\|^2 + m_0(\alpha q)^2} \cdot \omega(\sqrt{\log n})$.
   (b) Sample $\mathbf{e_1} \leftarrow_\$ D_{\mathbb{Z},s_r}^{nk}$, and let $\mathbf{e} = (\mathbf{e_0}, \mathbf{e_1})$.
   (c) Sample $\mathbf{r_2} \leftarrow_\$ D_{\mathbb{Z}^{nk},s'}$, then compute $\boldsymbol{\mu} = H_{\mathbf{F}}(\mathbf{w}, \mathbf{r_2})$, and construct $\mathbf{A}_{\mathbf{r}}^{(\boldsymbol{\mu})} = [\mathbf{A}_{\mathbf{r}}^{(\mathbf{0})} \parallel \mathbf{A}_{\mathbf{r}}^{(\mathbf{1})} + \hbar(\boldsymbol{\mu})\mathbf{G}]$.
   (d) Encode $\boldsymbol{\sigma_1}$ as $\boldsymbol{\sigma}_1' = \text{encode}(\boldsymbol{\sigma_1}) = \mathbf{Q}\boldsymbol{\sigma_1} \in \mathbb{Z}^{nk}$.
   (e) Choose a vector $\mathbf{s} \leftarrow_\$ \mathbb{Z}_q^n$ uniformly and let
   $$\mathbf{b^t} = 2\left(\mathbf{s^t}\mathbf{A}_{\mathbf{r}}^{(\boldsymbol{\mu})} \bmod q\right) + \mathbf{e^t} + \left(\mathbf{0}, \boldsymbol{\sigma}_1'\right)^t \bmod 2q. \tag{13}$$
(5) Encrypt $\mathbf{u}$ as follows.
   (a) Let $k' = H_2(\boldsymbol{\sigma_1}, \mathbf{b}) \in \mathscr{K}$.
   (b) Let $\mathbf{c} = \mathscr{E}_{k'}(\mathbf{u} \mid \boldsymbol{\sigma_2} \mid \mathbf{r_1} \mid \mathbf{r_2})$.
(6) Output the signcryption ciphertext $(\boldsymbol{\mu}, \mathbf{b}, \mathbf{c})$.

(iv) **Unsigncrypt**$(\mathbf{C}, A_r, T_r, A_s)$: upon receiving a signcryption ciphertext $(\boldsymbol{\mu}, \mathbf{b}, \mathbf{c})$ from a sender with the public key $\mathbf{A_s}$, the receiver with the private key $\mathbf{T_r}$ performs the following steps.

(1) Decrypt $(\boldsymbol{\mu}, \mathbf{b})$ to achieve $\boldsymbol{\sigma}_1$ as follows.

(a) If $\boldsymbol{\mu} = \mathbf{0}$, output $\perp$ and then abort; otherwise continue.

(b) Call $\mathbf{Invert}^{\mathcal{O}}(\mathbf{T_r}, \mathbf{A_r^{(\mu)}}, \mathbf{b}, \hbar(\boldsymbol{\mu}))$ to obtain $(\widetilde{\mathbf{z}}, \widetilde{\mathbf{e}})$, where $\widetilde{\mathbf{z}} \in \mathbb{Z}_q^n$ and $\widetilde{\mathbf{e}} = (\widetilde{\mathbf{e}}_0, \widetilde{\mathbf{e}}_1) \in \mathbb{Z}^{m_0} \times \mathbb{Z}^{nk}$.

(c) If $\|\widetilde{\mathbf{e}}_0\| \geq \alpha q\sqrt{m_0}$ or $\|\widetilde{\mathbf{e}}_1\| \geq \alpha q\sqrt{2m_0nk} \cdot \omega(\sqrt{\log n})$ output $\perp$ and abort; otherwise continue.

(d) Let $\mathbf{v} = \mathbf{b} - \widetilde{\mathbf{e}} \bmod 2q$ and then parse $\mathbf{v}$ as $(\mathbf{v_1}, \mathbf{v_2}) \in \mathbb{Z}_{2q}^{m_0} \times \mathbb{Z}_{2q}^{nk}$.

(e) If $\mathbf{v_1} \notin 2\boldsymbol{\Lambda}((\mathbf{A_r^0})^{\mathbf{t}})$, output $\perp$ and abort; otherwise, continue.

(f) Let $\widetilde{\boldsymbol{\sigma}}_\mathbf{1} = \mathbf{Q}^{-1}(\mathbf{v^t} \left[\begin{smallmatrix} \mathbf{T_r} \\ \mathbf{I} \end{smallmatrix}\right] \bmod 2q)$.

(2) Decrypt $\mathbf{c}$ as follows.

(a) Compute $k' = H_2(\widetilde{\boldsymbol{\sigma}}_\mathbf{1}, \mathbf{b}) \in \mathscr{K}$.

(b) Compute $\widetilde{\mathbf{u}}|\widetilde{\boldsymbol{\sigma}}_\mathbf{2}|\widetilde{\mathbf{r}}_\mathbf{1}|\widetilde{\mathbf{r}}_\mathbf{2} = \mathscr{D}_{k'}(\mathbf{c})$.

(3) Check the integrity of ciphertext as follows.

(a) Obtain $\widetilde{\boldsymbol{\sigma}}$ by composing $\widetilde{\boldsymbol{\sigma}}_\mathbf{1}, \widetilde{\boldsymbol{\sigma}}_\mathbf{2}$, and then compute $\widetilde{\mathbf{w}} = H_1(\widetilde{\boldsymbol{\sigma}})$.

(b) If $\boldsymbol{\mu} \neq H_\mathbf{F}(\widetilde{\mathbf{w}}, \widetilde{\mathbf{r}}_\mathbf{2})$ output $\perp$ and abort; otherwise, continue.

(c) If $\|\widetilde{\mathbf{r}}_\mathbf{2}\| \geq s'\sqrt{m}$ output $\perp$ and abort; otherwise, continue.

(4) Verifying the sender's authenticity as follows.

(a) Compute $\widetilde{\mathbf{h}} = H_0(\widetilde{\mathbf{u}}, \mathbf{A_r})$ and then build $\mathbf{A_s}^{(\widetilde{\mathbf{h}})} = [\mathbf{A_s} \parallel \mathbf{B^{(0)}} + \sum_{i\in[\lambda]} \widetilde{\mathbf{h}}_\mathbf{i}\mathbf{B^{(i)}}]$, where $\widetilde{\mathbf{h}} = (\widetilde{\mathbf{h}}_1, \ldots, \widetilde{\mathbf{h}}_\lambda) \in \{0,1\}^\lambda$ is the binary representation of $\widetilde{\mathbf{h}}$.

(b) If $\widetilde{\boldsymbol{\sigma}} > s_s\sqrt{m_1}$ output $\perp$ and abort; otherwise, continue.

(c) Compute $\widetilde{\mathbf{h}}_{\widetilde{\mathbf{r}}} = H_{\mathbf{A_s}^{(\widetilde{\mathbf{h}})}}(\widetilde{\mathbf{h}}, \widetilde{\mathbf{r}}_\mathbf{1})$.

(d) If $\mathbf{A_s}^{(\widetilde{\mathbf{h}})}\widetilde{\boldsymbol{\sigma}} = \widetilde{\mathbf{h}}_{\widetilde{\mathbf{r}}}$ then output $\widetilde{\mathbf{u}}$; otherwise, output $\perp$.

4.3. Consistency and Unsigncryption Error

Theorem 19 (consistency). *The above signcryption scheme can unsigncrypt correctly with* $1 - 2^{-\Omega(n)}$ *probability.*

Proof. We analyze the procedure along the unsigncryption algorithm, when a valid ciphertext $\mathbf{c_p} = (\boldsymbol{\mu}, \mathbf{b}, \mathbf{c})$ is input to the unsigncryption.

Firstly, we demonstrate that the correct $\boldsymbol{\sigma}_\mathbf{1}$ can be obtained with overwhelming probability in step (1) of unsigncryption.

(i) Firstly, let us prove that after calling $(\widetilde{\mathbf{z}}, \widetilde{\mathbf{e}}) \leftarrow \mathrm{Invert}^{\mathcal{O}}(\mathbf{T_r}, \mathbf{A_r^{(\mu)}}, \mathbf{b}, \hbar(\boldsymbol{\mu}))$, the probability of $\widetilde{\mathbf{e}} = \mathbf{e}$ is overwhelming, where $\mathbf{e}$ is the vector used in **Signcrypt** algorithm. At first, we need to show that this calling can work; that is, $\mathbf{T_r}$ is a $\mathbf{G}$-trapdoor for $\mathbf{A_r^{(\mu)}}$. For convenience, let $\mathbf{R_I} = \left[\begin{smallmatrix} \mathbf{T_r} \\ \mathbf{I} \end{smallmatrix}\right]$. Because $\mathbf{Q}$ is a basis of $\boldsymbol{\Lambda}(\mathbf{G^t})$, there must exist a matrix $\mathbf{Q}' \in \mathbb{Z}_q^{n\times nk}$ such that $\mathbf{Q} = \mathbf{G}^t\mathbf{Q}'$. As a result,

$$\begin{aligned}
&\widehat{\mathbf{b}}^{\mathbf{t}} \bmod q \\
&= \mathbf{b^t R_I} \\
&= \left(2\left(\mathbf{s^t A_r^{(\mu)}} \bmod q\right) + \mathbf{e^t} + \left(\mathbf{0}, \boldsymbol{\sigma}'\right)\right)\mathbf{R_I} \\
&= \left(2\left(\mathbf{s^t}\left[\mathbf{A_r^{(0)}} \parallel -\mathbf{A_r^{(0)}T_s} + \hbar(\boldsymbol{\mu})\mathbf{G}\right] \bmod q\right)\right. \\
&\quad \left. + (\mathbf{e_0}, \mathbf{e_1})^t + \left(\mathbf{0}, (\mathbf{Q}\boldsymbol{\sigma}_\mathbf{1})^t\right)\right)\mathbf{R_I} \\
&= 2\left(\mathbf{s^t}\hbar(\boldsymbol{\mu})\mathbf{G} \bmod q\right) + \left(\mathbf{e_0^t T_s} + \mathbf{e_1^t}\right) + \left(\mathbf{G}^t\mathbf{Q}'\boldsymbol{\sigma}_\mathbf{1}\right)^t \\
&= \left(2\mathbf{s^t} + \boldsymbol{\sigma}_\mathbf{1}^\mathbf{t}\mathbf{Q}'\hbar(\boldsymbol{\mu})^{-1}\right)\hbar(\boldsymbol{\mu})\mathbf{G} + \left(\mathbf{e_0^t T_r} + \mathbf{e_1^t}\right).
\end{aligned} \tag{14}$$

Clearly, the $\widehat{\mathbf{b}}^{\mathbf{t}}$ is the form of $g_\mathbf{G}(\widehat{\mathbf{s}}, \widehat{\mathbf{e}})$ for some $(\widehat{\mathbf{s}}, \widehat{\mathbf{e}})$; namely, $\mathbf{T_r}$ is a $\mathbf{G}$-trapdoor for $\mathbf{A_r^{(\mu)}}$. As a result, a vector $\widetilde{\mathbf{e}} \in \mathbb{Z}_m$ can be returned. We next demonstrate the probability that the probability for $\widetilde{\mathbf{e}} = \mathbf{e}$ is overwhelming by calling Invert$^{\mathcal{O}}$ (Algorithm 2). Clearly, if the $\mathcal{O}(\widehat{\mathbf{b}})$ in $\mathbf{Invert}^{\mathcal{O}}$ can return desired value, $\mathbf{Invert}^{\mathcal{O}}$ can obtain desired $\mathbf{e}$. The oracle $\mathcal{O}(\widehat{\mathbf{b}})$ is realized by calling **InvertG** (Algorithm 1); consequently, it only needs to prove the constraint condition is satisfied, that is, the error vector $\mathbf{e_0^t T_r} + \mathbf{e_1^t} \in \mathscr{P}_{1/2}(q \cdot \widetilde{\mathbf{S_k}}^{-\mathbf{t}})$, referring to Section 2.3 for the definition of $\mathbf{S_k}$. Because $\mathbf{e_0} \sim D_{\mathbb{Z},\alpha q}^{m_0}$, $\mathbf{e_1} \sim D_{\mathbb{Z},s_r}^{nk}$ for $s_r = \sqrt{\|\mathbf{e_0}\|^2 + m_0(\alpha q)^2} \cdot \omega(\sqrt{\log n})$, it follows that $\|\mathbf{e_0}\| < \alpha q\sqrt{m_0}$ and $\|\mathbf{e_1}\| < \alpha q\sqrt{2m_0nk}\omega(\log n)$ except with probability $2^{-\Omega(n)}$ by Proposition 9 items (2). When the parameters are set as in **Setup** and the sender's public/private keys are produced as in **KeyGen**, it follows that the maximum singular value for $\mathbf{T_r}$ satisfies $s_1(\mathbf{T_r}) \leq O(\sqrt{nk}) \cdot \omega(\sqrt{\log n})$ except probability $2^{-\Omega(n)}$ according to Proposition 10. Let $\mathbf{e} = (\mathbf{e_0}, \mathbf{e_1})$; it follows that $\|\mathbf{e^t R_I}\| \leq \|\mathbf{e_0^t T_r}\| + \|\mathbf{e_1^t}\| < \alpha q \cdot O(nk) \cdot \omega(\sqrt{\log n}) < O(q) \in \mathscr{P}_{1/2}(q \cdot \widetilde{\mathbf{S_k}}^{-\mathbf{t}})$ except probability $2^{-\Omega(n)}$, because $1/\alpha = O(nk) \cdot \omega(\sqrt{\log n})$ is large enough.

(ii) Secondly, when the correct $\mathbf{e}$ is obtained, the test in step (c) can be passed and the analysis is included in the above proof.

(iii) Thirdly, in step (e), for $\mathbf{v} = (\mathbf{v_0}, \mathbf{v_1}) = \mathbf{b} - \mathbf{e} \bmod 2q$, it follows that $\mathbf{v_0} \in 2\boldsymbol{\Lambda}(\mathbf{A_0^t})$ as desired.

(iv) Finally, in step (f), $\mathbf{v^t R_I} = 2(\mathbf{s^t}\hbar(\boldsymbol{\mu})\mathbf{G} \bmod q) + (\mathbf{Q}\boldsymbol{\sigma}_\mathbf{1})^t \bmod 2q$; as a result, $\mathbf{v^t R_I}$ and $(\mathbf{Q}\boldsymbol{\sigma}_\mathbf{1})^t$ are in the identical coset, so the decryption can obtain $\boldsymbol{\sigma}_\mathbf{1}$ exactly.

Next, after obtaining correct $\boldsymbol{\sigma}_1$ and $\mathbf{b}$ via step (1), we get the correct key used for symmetrical encryption, so we can obtain correct $(\boldsymbol{\mu}, \boldsymbol{\sigma}_2, \mathbf{r}_1, \mathbf{r}_2)$ in step (2), and the verification for hash values in step (3) can be passed.

Finally, let us analyze step (4). Specifically, we prove that the signature verification can be passed with overwhelming probability. By now, we have got correct $(\boldsymbol{\sigma}, \mathbf{r}_1)$ that is a signature for $\mathbf{u}$, and we only need to prove that it is valid. First, we evaluate the probability for $\boldsymbol{\sigma} \le s_s\sqrt{m_1}$. $\boldsymbol{\sigma} = \left[\begin{smallmatrix}\boldsymbol{\sigma}_{\mathbf{or}}\\ \mathbf{y}\end{smallmatrix}\right]$, where $\boldsymbol{\sigma}_{\mathbf{or}}$ is obtained by calling the algorithm SampleDG. It is known by SampleDG that $\mathbf{A}_{\mathbf{s}}^{(\mathbf{0})}\boldsymbol{\sigma}_{\mathbf{or}} = \mathbf{y}'$ and $\|\boldsymbol{\sigma}_{\mathbf{or}}\| \le s_s\sqrt{m}$ with probability $1 - 2^{-\Omega(n)}$ by Proposition 9 items (2). On the other hand, $\mathbf{y} \le s_s\sqrt{nk}$ with probability $1 - 2^{-\Omega(n)}$ by the same lemma. Therefore, $\boldsymbol{\sigma} \le s_s\sqrt{m_1}$ with probability $1 - 2^{-\Omega(n)}$. Second,

$$\begin{aligned}\mathbf{A}_{\mathbf{s}}^{(\mathbf{h})}\boldsymbol{\sigma} &= \left[A_s \parallel A'\right]\begin{bmatrix}\boldsymbol{\sigma}_{\mathbf{or}}\\ \mathbf{y}\end{bmatrix} = \mathbf{y}' + \mathbf{A}'\mathbf{y}\\ &= \mathbf{h}_{\mathbf{r}} = H_{\mathbf{A}_{\mathbf{s}}^{(\mathbf{h})}}\left(H_0\left(\mathbf{u}, \mathbf{A}_{\mathbf{r}}\right), \mathbf{r}\right).\end{aligned} \tag{15}$$

Consequently, the signature is valid with probability $1-2^{-\Omega(n)}$. □

5. Security Proofs

Before giving the proofs on the confidentiality and unforgeability of the proposed scheme, we need at first to prove the following lemma.

Lemma 20. *For a given unit $\boldsymbol{\mu}^* \in \mathfrak{R} = \mathbb{Z}_q[x]/(f(x))$, if $\mathbf{u} \leftarrow_{\$} \{0,1\}^*$, $\mathbf{r}_2 \leftarrow_{\$} D_{\mathbb{Z}^m, s'}$, and $(\boldsymbol{\sigma}, \mathbf{r}_1)$ is a signature obtained in step (1) of the **Signcrypt** algorithm, then the probability for $H_{\mathbf{F}}(H_1(\boldsymbol{\sigma}), \mathbf{r}_2) = \boldsymbol{\mu}^*$ is negligible. More precisely, $Pr[H_{\mathbf{F}}(H_1(\boldsymbol{\sigma}), \mathbf{r}_2) = \boldsymbol{\mu}^* \mid \mathbf{u} \leftarrow_{\$} \{0,1\}^*] = \Omega(q^{-n})$.*

Proof. We first evaluate the number of units in the above ring $\mathfrak{R}$. As defined in [17], the monic degree-n polynomial $f(x) = x^n + f_{n-1}x^{n-1} + \cdots + f_0 \in \mathbb{Z}[x]$ is irreducible modulo every prime p dividing q. Because $f(x)$ is irreducible, $(f(x))$ is maximum ideal and $\mathbb{Z}_p[x]/(f(x))$ is a field according to Chinese remainder theorem. An element $a \in \mathfrak{R}$ is a unit if and only if it is nonzero modulo any prime p dividing q. Assume that q has prime factors $p_1, p_2, \ldots, p_t$. The amount of elements which are zero modulo prime factor p_i is $q/p_i + 1$. By inclusion-exclusion principle, the amount of units in $\mathfrak{R}$ is

$$\begin{aligned}&q^n - \left[\left(\frac{q}{p_1}+1\right)^n + \left(\frac{q}{p_2}+1\right)^n + \cdots + \left(\frac{q}{p_t}+1\right)^n\right]\\ &\quad + \left[\left(\frac{q}{(p_1p_2)}+1\right)^n + \left(\frac{q}{(p_1p_3)}+1\right)^n\right.\\ &\qquad \left. + \cdots + \left(\frac{q}{(p_{t-1}p_t)}+1\right)^n\right]\\ &\quad + \cdots + (-1)^t\left(\frac{q}{(p_1p_2\cdots p_t)}+1\right)^n\end{aligned} \tag{16}$$

$$\begin{aligned}&\approx q^n - \left[\left(\frac{q}{p_1}\right)^n + \left(\frac{q}{p_2}\right)^n + \cdots + \left(\frac{q}{p_t}\right)^n\right]\\ &\quad + \left[\left(\frac{q}{(p_1p_2)}\right)^n + \left(\frac{q}{(p_1p_3)}\right)^n + \cdots + \left(\frac{q}{(p_{t-1}p_t)}\right)^n\right]\\ &\quad + \cdots + (-1)^t\left(\frac{q}{(p_1p_2\cdots p_t)}\right)^n\end{aligned} \tag{17}$$

$$\begin{aligned}&= q^n - \left[\frac{q^n}{p_1^n} + \frac{q^n}{p_2^n} + \cdots + \frac{q^n}{p_t^n}\right]\\ &\quad + \left[\frac{q^n}{((p_1^np_2^n))} + \frac{q^n}{((p_1^np_3^n))} + \cdots + \frac{q^n}{((p_{t-1}^np_t^n))}\right]\\ &\quad + \cdots + \frac{(-1)^tq^n}{(p_1^np_2^n\cdots p_t^n)}\end{aligned} \tag{18}$$

$$\begin{aligned}&= q^n\left(1 - \frac{1}{p_1^n}\right)\left(1 - \frac{1}{p_2^n}\right)\cdots\left(1 - \frac{1}{p_t^n}\right)\\ &= O\left(q^n\right),\end{aligned}$$

where the approximating from (17) to (18) is implied by that q is large enough. In the proposed scheme, $q = p^e$ and $q^n - (q/p+1)^n \approx q^n - (q/p)^n = q^n(1 - 1/p^n)$. On the other hand, the hash functions H_1 and $H_{\mathbf{F}}$ are both universal. Based on the above two reasons, this lemma holds. □

Theorem 21 (confidentiality). *The proposed signcryption is indistinguishable against inner adaptively chosen ciphertext attack (IND-CCA2) assuming the decision-LWE$_{q,\alpha'}$ problem (for $\alpha' = \alpha/3 \ge 2\sqrt{n}/q$) is intractable.*

Proof. At first, let us define the following game sequence between a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$.

(i) The game G_0 is exactly the IND-CCA2 attack with the system as described in Section 3.

(ii) In game G_1, the challenger change the way to construct the receiver's public key $\mathbf{A}_{\mathbf{r}}$ and the way to answer unsigncryption queries. The receiver's public key $\mathbf{A}_{\mathbf{r}}$ is produced as follows. At the start of the game, choose $\mathbf{A}_{\mathbf{r}}^{(\mathbf{0})}, \mathbf{T}_{\mathbf{r}}$ as in game G_0 and let $\mathbf{T} = \mathbf{T}_{\mathbf{r}}$, next choose $\boldsymbol{\mu}_{\mathbf{0}} \leftarrow \mathfrak{R}$, and then construct $\mathbf{A}_{\mathbf{r}} = [\mathbf{A}_{\mathbf{r}}^{(\mathbf{0})} \parallel -\mathbf{A}_{\mathbf{r}}^{(\mathbf{0})}\mathbf{T} - \hbar(\boldsymbol{\mu}_{\mathbf{0}})\mathbf{G}]$. The challenger gives the adversary $\mathbf{A}_{\mathbf{r}}$ as the sender's public key. Whenever $\mathscr{A}$ invokes a unsigncryption query on $(\mathbf{c}', \mathbf{A}_{\mathbf{s}}, \mathbf{T}_{\mathbf{s}}) = ((\boldsymbol{\mu}, \mathbf{b}, \mathbf{c}), \mathbf{A}_{\mathbf{s}}, \mathbf{T}_{\mathbf{s}})$, $\mathscr{C}$ responds as normal except that in step (1) of **Unsigncrypt** algorithm, the decryption for $(\boldsymbol{\mu}, \mathbf{b})$ is changed as follows.

(1) Decrypt $(\boldsymbol{\mu}, \mathbf{b})$ to achieve $\boldsymbol{\sigma}_1$ as follows.

(a′) If $\boldsymbol{\mu} = \mathbf{0}$ or $\boldsymbol{\mu} = \boldsymbol{\mu}_{\mathbf{0}}$, output $\perp$ and then abort; otherwise continue.

(b′) Call $\mathbf{Invert}^{\mathscr{O}}(\mathbf{T}_{\mathbf{r}}, \mathbf{A}_{\mathbf{r}}^{(\boldsymbol{\mu})}, \mathbf{b}, \hbar(\boldsymbol{\mu} - \boldsymbol{\mu}_0))$ to obtain $(\bar{\mathbf{z}}, \bar{\mathbf{e}})$, where $\bar{\mathbf{z}} \in \mathbb{Z}_q^n$ and $\bar{\mathbf{e}} = (\bar{\mathbf{e}}_{\mathbf{0}}, \bar{\mathbf{e}}_{\mathbf{1}}) \in \mathbb{Z}^{m_0} \times \mathbb{Z}^{nk}$.

(c) If $\|\widetilde{\mathbf{e}}_0\| \geq \alpha q\sqrt{m_0}$ or $\|\widetilde{\mathbf{e}}_1\| \geq \alpha q\sqrt{2m_0nk} \cdot \omega(\sqrt{\log n})$ output $\perp$ and abort; otherwise continue.

(d) Let $\mathbf{v} = \mathbf{b} - \widetilde{\mathbf{e}} \bmod 2q$ and then parse $\mathbf{v}$ as $(\mathbf{v}_1, \mathbf{v}_2) \in \mathbb{Z}_{2q}^{m_0} \times \mathbb{Z}_{2q}^{nk}$.

(e) If $\mathbf{v}_1 \notin 2\boldsymbol{\Lambda}((\mathbf{A}_\mathbf{r}^0)^\mathbf{t})$, output $\perp$ and abort; otherwise, continue.

(f′) Let $\widetilde{\boldsymbol{\sigma}}_1 = \mathbf{Q}^{-1}(\mathbf{v}^\mathbf{t}\left[\begin{smallmatrix}\widehat{\mathbf{T}}\\ \mathbf{I}\end{smallmatrix}\right] \bmod 2q)$ where $\widehat{\mathbf{T}}$ is an arbitrary solution of $\mathbf{A}_\mathbf{r}^{(0)}\widehat{\mathbf{T}} = \mathbf{A}_0\mathbf{T} - \hbar(\boldsymbol{\mu}_0)\mathbf{G}$.

(ii) In game G_2, the challenger only changes hash function $H_\mathbf{F}$ and the method to produce challenge ciphertext $(\boldsymbol{\mu}^*, \mathbf{b}^*, \mathbf{c}^*)$ as follows. The change for hash function $H_\mathbf{F}$ is as follows. The challenger replaces the hash function $H_\mathbf{F}$ with a chameleon hash function $H_\mathbf{E}$ without revealing the trapdoor, where the matrix $\mathbf{E} = [\mathbf{E}^{(0)} \parallel \mathbf{E}^{(1)}] \in \mathbb{Z}_q^{n\times l} \times \mathbb{Z}_q^{n\times m}$ and $\mathbf{E}^{(0)}$ (resp., $\mathbf{E}^{(1)}$) has the same distribution with $\mathbf{F}^{(0)}$ (resp., $\mathbf{F}^{(1)}$). The challenge ciphertext is produced as follows. The adversary provides two equal length messages $\mathbf{u}_0, \mathbf{u}_1$ and the sender's public/private keys $\mathbf{A}_\mathbf{s}^*, \mathbf{T}_\mathbf{s}^*$. The challenger tosses a fair coin $r_c \in \{0, 1\}$, and then signcrypts $\mathbf{u}_{\mathbf{r}_c}$ with a slightly change. The challenger signs $\mathbf{u}_{\mathbf{r}_c}$ normally to obtain $(\boldsymbol{\sigma}, \mathbf{r}_1)$, next chooses $\boldsymbol{\mu}^* = \boldsymbol{\mu}_0$, and then chooses $\mathbf{r}_2$ such that $\boldsymbol{\mu}^* = H_\mathbf{E}(H_1(\boldsymbol{\sigma}), \mathbf{r}_2)$ ($\mathscr{C}$ can do this since he/she knows the trapdoor of the chameleon hash $H_\mathbf{E}$). The subsequent signcryption operation is the same as G_1.

(iii) In game G_3, the challenger continues to change the how the challenge ciphertext $(\boldsymbol{\mu}^*, \mathbf{b}^*, \mathbf{c}^*)$ is created. Concretely, only the way to produce $\mathbf{b}^*$ is changed as follows. The challenger normally chooses $\mathbf{s} \leftarrow_\$ \mathbb{Z}_q^n$, $\mathbf{e}_0 \leftarrow D_{\mathbb{Z}^{m_0}, \alpha q}$ and let $\mathbf{b}_0^\mathbf{t} = 2(\mathbf{s}^\mathbf{t}\mathbf{A}_\mathbf{r}^{(0)} \bmod q) + \mathbf{e}_0^\mathbf{t} \bmod 2q$. Next, choose $\widehat{\mathbf{e}} \leftarrow_\$ D_{\mathbb{Z}^{nk}, \alpha q\sqrt{m}\omega(\sqrt{\log n})}$ and let $\mathbf{b}_1^\mathbf{t} = -\mathbf{b}_0^\mathbf{t}\mathbf{T}_\mathbf{r} + \widehat{\mathbf{e}}^\mathbf{t} + (\mathbf{Q}\boldsymbol{\sigma}_1)^t \bmod 2q$. Let $(\mathbf{b}^*)^\mathbf{t} = (\mathbf{b}_0^\mathbf{t}, \mathbf{b}_1^\mathbf{t})$. All the others are same as in game G_2.

(iv) In game G_4, the challenger continues to change the challenge ciphertext. The challenger chooses $\mathbf{b}_0 \leftarrow_\$ \mathbb{Z}_{2q}^{m_0}$ uniformly. All the others are identical to G_3.

Then, this theorem is implied by the indistinguishability between two successive games G_i and G_{i+1} $(i = 0, 1, 2, 3)$ that are presented in Lemmas 22, 23, 24, and 25, respectively. □

Lemma 22. *The adversary's views in game G_0 and game G_1 are statistically indistinguishable. Meanwhile, G_1 can unsigncrypt correctly (with overwhelming probability).*

Proof. We first prove the indistinguishability for public key. Given $\boldsymbol{\mu}_0 \in \mathfrak{R}^*$, $F(\boldsymbol{\mu}_0) \in GL_n(q)$ is a fixed matrix. On the other hand, $\mathbf{A}_\mathbf{r}^{(0)}\mathbf{T}_\mathbf{r}$ is negl(n)-uniform by leftover hash lemma. Therefore, $\mathbf{A}_\mathbf{r}^{(0)}\mathbf{T}_\mathbf{r} - \hbar(\boldsymbol{\mu}_0)\mathbf{G}$ is negl(n)-uniform. Consequently, the value of $\boldsymbol{\mu}_0$ is statistically hidden from the adversary and the distribution of public key in G_0 and G_1 is statistically indistinguishable.

Next, we illustrate the challenger $\mathscr{C}$ in the game G_1 can unsigncrypt correctly and $\mathscr{C}$'s unsigncryption behavior in G_0 and G_1 is indistinguishable from the view of the adversary $\mathscr{A}$. When the ciphertext queried is not valid, both games will abort. Therefore, we only need to analyze the case for a valid ciphertext. In the unsigncryption process of game G_1, only the decryption for $\boldsymbol{\mu}, \mathbf{b}$ (i.e., public key decryption process) is changed. Therefore, it is enough to prove the correctness of public key decryption. At first, if $\boldsymbol{\mu} = \mathbf{0}$, both games will output $\perp$. Otherwise, there are two cases for $\boldsymbol{\mu}$: $\boldsymbol{\mu} \neq \boldsymbol{\mu}_0$ or not.

We firstly analyze the former. In this case, both games call Invert$^\mathscr{O}$ to obtain $(\mathbf{z}, \mathbf{e})$ such that $\mathbf{b}^\mathbf{t} = \mathbf{z}^\mathbf{t}\mathbf{A}_\mathbf{r}^{(\boldsymbol{\mu})} + \mathbf{e}^\mathbf{t} \bmod q$ (refer to Section 4.3). In game G_1,

$$\begin{aligned}\mathbf{A}_\mathbf{r}^{(\boldsymbol{\mu})} &= \left[\mathbf{A}_\mathbf{r}^{(0)} \parallel -\mathbf{A}_\mathbf{r}^{(0)}\mathbf{T} - \hbar\left(\boldsymbol{\mu}_0\right)\mathbf{G} + \hbar\left(\boldsymbol{\mu}\right)\mathbf{G}\right]\\ &= \left[\mathbf{A}_\mathbf{r}^{(0)} \parallel -\mathbf{A}_\mathbf{r}^{(0)}\mathbf{T} + \hbar\left(\boldsymbol{\mu} - \boldsymbol{\mu}_0\right)\mathbf{G}\right].\end{aligned} \tag{19}$$

Clearly, conditioned on $\boldsymbol{\mu} \neq \boldsymbol{\mu}_0$, $\hbar(\boldsymbol{\mu} - \boldsymbol{\mu}_0) \in \mathbb{Z}_q^{n\times n}$ is invertible according to the "unit differences" on $\mathfrak{R}$, which is necessary for calling Invert$^\mathscr{O}$. It also follows that $\mathbf{T}$ is the $\mathbf{G}$-trapdoor for $\mathbf{A}_\mathbf{r}^{(\boldsymbol{\mu})}$ corresponding to invertible tag $\hbar(\boldsymbol{\mu} - \boldsymbol{\mu}_0)$. Therefore, the challenger needs to replace $\mathbf{M}_\mathbf{I} = \hbar(\boldsymbol{\mu})$ with $\mathbf{M}_\mathbf{I} = \hbar(\boldsymbol{\mu} - \boldsymbol{\mu}_0)$ when calling Invert$^\mathscr{O}$. In step (c), if there is $\mathbf{e}$ obtained from step (b′) that satisfies the constraint condition, it follows that $\mathbf{e}^\mathbf{t}\left[\begin{smallmatrix}\mathbf{T}\\ \mathbf{I}\end{smallmatrix}\right] \in \mathscr{P}_{1/2}(q \cdot \mathbf{S}_\mathbf{k}^{-\mathbf{t}})$ in both games, where $\mathbf{S}_\mathbf{k}$ has been defined in Section 2.3. Therefore, this $\mathbf{e}$ can be obtained by calling Invert$^\mathscr{O}$ in both games; otherwise, if there is no such an $\mathbf{e}$, both games will output $\perp$. In step (e), if $\mathbf{v}_1 \notin 2\boldsymbol{\Lambda}((\mathbf{A}_\mathbf{r}^{(0)})^\mathbf{t})$, both games output $\perp$; otherwise, there exist $\mathbf{s} \in \mathbf{Z}_q^n$ and $\boldsymbol{\sigma}_1' \in \mathbb{Z}_{2q}^{nk}$ such that

$$\mathbf{v}^\mathbf{t} = \mathbf{b}^\mathbf{t} - \mathbf{e}^\mathbf{t} = 2\left(\mathbf{s}^\mathbf{t}\mathbf{A}_\mathbf{r}^{(\boldsymbol{\mu})} \bmod q\right) + \left(\mathbf{0}, \boldsymbol{\sigma}_1'\right)^t \bmod 2q. \tag{20}$$

In step (f) of game G_0, $\mathscr{C}$ computes

$$\begin{aligned}&\mathbf{Q}^{-1}\left(\mathbf{v}^\mathbf{t}\begin{bmatrix}\mathbf{T}\\ \mathbf{I}\end{bmatrix} \bmod 2q\right)\\ &\quad= \mathbf{Q}^{-1}\left(2\left(\mathbf{s}^\mathbf{t}\hbar\left(\boldsymbol{\mu}\right)\mathbf{G} \bmod q\right) + \boldsymbol{\sigma}_1'^\mathbf{t} \bmod 2q\right),\end{aligned} \tag{21}$$

while in step (f′) of game G_1, $\mathscr{C}$ does as follows: first, find any $\widehat{\mathbf{T}}$ such that $\mathbf{A}_\mathbf{r}^{(0)}\widehat{\mathbf{T}} = -\mathbf{A}_\mathbf{r}^{(0)}\mathbf{T} - \hbar(\boldsymbol{\mu}_0)\mathbf{G}$, and then, compute

$$\begin{aligned}&\mathbf{Q}^{-1}\left(\mathbf{v}^\mathbf{t}\begin{bmatrix}\widehat{\mathbf{T}}\\ \mathbf{I}\end{bmatrix} \bmod 2q\right)\\ &\quad= \mathbf{Q}^{-1}\left(2\left(\mathbf{s}^\mathbf{t}\hbar\left(\boldsymbol{\mu}\right)\mathbf{G} \bmod q\right) + \boldsymbol{\sigma}_1'^\mathbf{t} \bmod 2q\right).\end{aligned} \tag{22}$$

Clearly, $\mathbf{v}^\mathbf{t}\left[\begin{smallmatrix}\mathbf{T}_\mathbf{r}\\ \mathbf{I}\end{smallmatrix}\right]$ in G_0, $\mathbf{v}^\mathbf{t}\left[\begin{smallmatrix}\widehat{\mathbf{T}}\\ \mathbf{I}\end{smallmatrix}\right]$ in G_1, and $\boldsymbol{\sigma}_1'^t$ are in the same coset $\boldsymbol{\Lambda}(\mathbf{G}^\mathbf{t})/2\boldsymbol{\Lambda}(\mathbf{G}^\mathbf{t})$; therefore G_0 and G_1 can both decrypt the desired value.

We next discuss the latter case; that is, $\boldsymbol{\mu} = \boldsymbol{\mu}_0$. In this case, game G_1 cannot unsigncrypt because $\hbar(\boldsymbol{\mu} - \boldsymbol{\mu}_0)$ is not invertible, but since $\boldsymbol{\mu}_0$ is unknown to the adversary in G_1, the probability for $\boldsymbol{\mu} = \boldsymbol{\mu}_0$ is negligible according to Lemma 20. Based on the above analysis, the games G_0 and G_1 are indistinguishable. □

Lemma 23. *The adversary's views in game G_2 and game G_1 are statistically indistinguishable.*

Proof. At first, because the matrices used for constructing hash functions $H_{\mathbf{E}}$ and $H_{\mathbf{F}}$ have identical distribution, the games G_2 and G_1 are statistically indistinguishable when the hash function is replaced. Although the way for producing the challenge ciphertext $(\boldsymbol{\mu}^*, b^*, c^*)$ in G_2 is changed, the adversary cannot distinguish $\boldsymbol{\mu}^* = \boldsymbol{\mu}_0$ from $\boldsymbol{\mu}^* = H_{\mathbf{E}}(\mathbf{w}, \mathbf{r_2})$ without knowing $\boldsymbol{\mu}_0, w, r_2$ in advance, considering that $H_{\mathbf{E}}$ is universal and $\boldsymbol{\mu}_0$ is random selected. □

Lemma 24. *The adversary's views in game G_3 and game G_2 are statistically indistinguishable.*

Proof. The key idea of this lemma's proof is similar to a section in Theorem 6.3 of [17]. The change of challenge ciphertext in G_3 is only at the public encryption section, more precisely, only the component $\mathbf{b} = (\mathbf{b_0}, \mathbf{b_1}) \in \mathbb{Z}_{2q}^{m_0} \times \mathbb{Z}_{2q}^{nk}$ as described above. The distribution of $\mathbf{b_0}$ in both games is identical. With respect to $\mathbf{b_1}$, in game G_2,

$$\mathbf{b_1^t} = 2\left(-\mathbf{s^t A_r^{(0)} T_r} \bmod q\right) + \mathbf{e_1^t} + \left(\mathbf{Q}\boldsymbol{\sigma_1}\right)^t \bmod 2q, \quad (23)$$

where $\mathbf{e_1} \sim D_{\mathbb{Z}^{m_0}, s}$, for $s = \sqrt{\|\mathbf{e_0}\|^2 + m_0(\alpha q)^2}\,\omega(\sqrt{\log n})$. In game G_3,

$$\begin{aligned}\mathbf{b_1^t} &= -\mathbf{b_0^t T_r} + \hat{\mathbf{e}}^{\mathbf{t}} + \left(\mathbf{Q}\boldsymbol{\sigma_1}\right)^t \bmod 2q \\ &= 2\left(-\mathbf{s^t A_r^{(0)} T_r} \bmod q\right) + \left(\mathbf{e_0^t T_r} + \hat{\mathbf{e}}^{\mathbf{t}}\right) + \left(\mathbf{Q}\boldsymbol{\sigma_1}\right)^t \bmod 2q.\end{aligned} \quad (24)$$

It only needs to prove that the statistical distance between $\mathbf{e_0^t T_r} + \hat{\mathbf{e}}^{\mathbf{t}}$ and $\mathbf{e_1^t}$ is negligible. Express $\mathbf{T_r}$ as $(\mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_{nk}}) \in \mathbb{Z}^{m_0 \times nk}$, where $\mathbf{t_i} \sim D_{\mathbb{Z}^{nk}, \omega(\sqrt{\log n})}$. On the other hand $\hat{\mathbf{e}} \sim D_{\mathbb{Z}, \alpha q\sqrt{m}\omega(\sqrt{\log n})}$. It follows that for fixed $\mathbf{e_0}$, $\langle \mathbf{e_0}, \mathbf{t_i} \rangle + \hat{\mathbf{e}}_{\mathbf{i}}$ is negl (n)-far from $D_{\mathbb{Z}, s_1}$ for $s_1^2 = (\|\mathbf{e_0}\|^2 + m_0(\alpha q)^2)\omega(\sqrt{\log n})^2$ according to Corollary 3.10 of [23] and Theorem 3.1 of [28]. In other words, $\mathbf{b_1}$ in G_3 has distribution negl(n)-far from $\mathbf{b_1}$ in G_2. Consequently, the challenge ciphertext $(\boldsymbol{\mu}^*, \mathbf{b}^*, \mathbf{c}^*)$ in both games G_3 and G_2 is statistically indistinguishable. □

Lemma 25. *The adversary's views in game G_4 and game G_3 are computationally indistinguishable and the adversary's advantage in G_4 is negligible, assuming that the decision-LWE$_{q,\alpha'}$ problem (for $\alpha' = \alpha/3 \geq 2\sqrt{n}/q$) is intractable.*

Proof. The idea of this lemma's proof is similar to a section in Theorem 6.3 of [17]. In order to show the indistinguishability, a method to discretize LWE is needed. Concretely, $\mathbf{A}_{\mathbf{s},\boldsymbol{\alpha}'}$ is a LWE instance over $\mathbb{Z}_q^n \times \mathbb{T}$. The O_s samples (for $\mathbf{s} \in \mathbb{Z}_q^n$) $(\boldsymbol{\alpha}, b = \langle \mathbf{s}, \boldsymbol{\alpha} \rangle/q + e \bmod 1)$ can be transformed to $(\boldsymbol{\alpha}, 2\langle \mathbf{s}, \boldsymbol{\alpha} \rangle \bmod q + \mathbf{e}' \bmod 2q) \in \mathbb{Z}_q^n \times \mathbb{Z}_{2q}$ by mapping $b \mapsto 2qb + D_{\mathbb{Z}-2qb,s}$, for $e' \leftarrow D_{\mathbb{Z},\alpha q}$ and $s^2 = (\alpha q)^2 - (2\alpha' q)^2 \geq 4n \geq \eta_\epsilon(\mathbb{Z})^2$ according to Theorem 3.1 of [28]. Clearly, by the above mapping, the uniform instance $O_\$$ over $\mathbb{Z}_q^n \times \mathbb{T}$ is mapped to the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_{2q}$.

In game G_3, $(\mathbf{A_r^{(0)}}, \mathbf{b_0})$ is in fact an instance of O_s. In game G_4, $(\mathbf{A_r^{(0)}}, \mathbf{b_0})$ is an uniform random instance $O_\$$ over $\mathbb{Z}_q^{n \times m_0} \times \mathbb{Z}_{2q}^n$. Because LWE is pseudorandom, the above discretized distribution is also pseudorandom under the constraint condition for $\alpha' = \alpha/3 \geq 2\sqrt{n}/q$. Therefore, under discretized LWE assumption, the games G_4 and G_3 are computationally indistinguishable.

Next, we analyze the adversary's advantage in the game G_4. According to leftover hash lemma, $(\mathbf{A_r^{(0)}}, \mathbf{b_0}, \mathbf{A_0 T_r}, \mathbf{b_0 T_r})$ is negl(n)-uniform, when choosing $\mathbf{T_r}$ as in G_4. Therefore, the challenge ciphertext has at most negl(n)-far distribution when encrypting any different messages. Consequently, the adversary's advantage in G_4 is negligible. □

Theorem 26 (unforgeability). *In standard model, the proposed signcryption is strongly unforgeable against inner adaptively chosen message attacks (SUF-CMA) assuming that $SIS_{q,\beta}$ for large enough $\beta = O(\lambda(nk)^{3/2}) \cdot \omega(\sqrt{\log n})^3$ is hard.*

Proof. We prove it by contradiction. If an adversary $\mathscr{F}$ can forge a signcryption in the proposed scheme, then the simulator can forge a signature of the above SUF-CMA signature scheme used in the proposed scheme.

Initial: $\mathscr{C}$ gets public parameter $\mathscr{P}_p$ and his/her public/private keys (A_s^*, T_s^*) by running successively the algorithms Setup and KeyGen and then $\mathscr{C}$ gives A_s^* and $\mathscr{P}_p$ to $\mathscr{F}$.

Singcrypt query: in this phase, the adversary $\mathscr{F}$ can perform polynomially bounded signcryption queries as follows. When $\mathscr{F}$ submits a message ($\mathbf{u}$ and a intended receiver's public key $\mathbf{A_r}$) for querying. (For convenience, we denote the intended receiver's private key by $\mathbf{T_r}$). $\mathscr{C}$ gets a signcryption value by $\mathbf{c}' \leftarrow \mathbf{Signcrypt}(\mathbf{u}, \mathbf{A_r}, \mathbf{A_s^*}, \mathbf{SK_s^*})$ and gives $\mathbf{c}'$ to $\mathscr{F}$.

Forgery: $\mathscr{F}$ outputs a receivers public/private keys $(\mathbf{A_r^*}, \mathbf{T_r^*})$ and a fresh ciphertext $c^* = (\mathbf{u}, \mathbf{b}, \mathbf{c})$ under the sender's public key $\mathbf{A_s^*}$ and the receiver's private key $\mathbf{T_r^*}$. Because c^* is a valid ciphertext, $\mathscr{C}$ does what follows.

(1) Decrypt $(\mathbf{u}, \mathbf{b})$ with $\mathbf{T_r^*}$ to obtain $\boldsymbol{\sigma_1}$.

(2) Decrypt $\mathbf{c}$ with $H_2(\boldsymbol{\sigma_1}, b)$ to obtain $(\mathbf{u} \mid \boldsymbol{\sigma_2} \mid \mathbf{r_1} \mid \mathbf{r_2})$.

(3) Combine $\boldsymbol{\sigma_1}$ and $\boldsymbol{\sigma_2}$ to obtain $\boldsymbol{\sigma}$.

In the proposed scheme, we use the signature scheme of [17]. However the syndrome $\mathbf{y}$ in the signature scheme is replaced by a chameleon hash value of the message $\mathbf{u}$ and some random $\mathbf{r}$. For convenience, the signature scheme involving a chameleon hash function is called MP$_c$ signature scheme. The signature scheme of [17] is SUF-SMA in the standard model assuming the hardness of SIS$_{q,\beta}$ for large enough $\beta = O(\lambda(nk)^{3/2}) \cdot \omega(\sqrt{\log n})^3$; therefore the MP$_c$ signature scheme is SUF-CMA in the standard model according to [33, Lemma 2.3] or [34, Lemma 2.1].

Because c^* is a valid ciphertext, $(\boldsymbol{\sigma}, \mathbf{r_1})$ is a valid MP$_c$ signature on message $\mathbf{u}$. Now, we have got a contradiction. Consequently, the proposed signcryption scheme is also SUF-CMA assuming the hardness of SIS$_{q,\beta}$ for $\beta = O(\lambda(nk)^{3/2}) \cdot \omega(\sqrt{\log n})^3$. □

6. Performance Analysis and Simulations

This section compares the ciphertext length, computational cost, key size, and so forth in the proposed scheme with that in the normal signature-then-encryption diagram and the existing signcryption schemes based on lattice.

The dimension of public keys $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ need to be declared firstly. Assuming that the security parameter n is the same in LMK12 [14], WHW12 [15], and ours. In LMK12 [14] and WHW12 [15], the public/private keys for signature and encryption are all generated by the approach in [29], and the dimension m meets $m' = 6n\log q$. In our scheme, the trapdoor generation algorithm is the approach proposed in [17]. In order to meet the conditions: statistically close to uniform and computationally pseudorandom, let dimension $m = 2n\log q$ for public key in $\mathbb{Z}_q^{n\times m}$. For convenience, the modulo q is assumed to be same in the three schemes, although the q in our scheme might be smaller than that in LMK12 [14] and WHW12 [15]. It needs to be illustrated that the signature scheme used in ours is that proposed in [17]. The matrix used in it is in $\mathbb{Z}^{n\times m_1}$, where $m_1 = m+nk$ for $k = \log q$.

6.1. Comparison with Signature-Then-Encryption Diagram. First, we compare on ciphertext length. When we send an l'-bit length message, the normal signature-then-encryption diagram uses [17] to sign, and the signature length is approximately $m_1 \log q$ bits. It uses [17] to encrypt the plaintext, and the ciphertext length is

$$\left\lceil \frac{l'}{(nk)} \right\rceil [n\log q + m\log 2q] \approx 2l'\log q, \tag{25}$$

but this can only achieve CCA1 security. Aiming to achieve CCA2 security, a good candidate is to produce one time signature (OTS) for the ciphertext. For efficiency, it can achieve this target to sign for the hash value of ciphertext. The signature length of OTS is also $m_1 \log q$. In this way, the total sum of bits is about

$$\begin{aligned} &m_1\log q + \left\lceil \frac{l'}{(nk)} \right\rceil (n\log q + m\log q) + m_1\log q \\ &\quad\approx 6n\log^2 q + 2l'\log q. \end{aligned} \tag{26}$$

In our scheme, the form of ciphertext is $(\boldsymbol{\mu}, \mathbf{b}, \mathbf{c})$. In a ciphertext, $\mathbf{c}$ is the ciphertext for $(\mathbf{u} \mid \boldsymbol{\sigma}_2 \mid \mathbf{r}_1 \mid \mathbf{r}_2)$ with a symmetric encryption scheme whose plaintext and ciphertext are of equal length. The length for $\boldsymbol{\mu}, \mathbf{b}, \boldsymbol{\sigma}_2$ is (about) $n\log q$, $m\log 2q$, and $m_1\log q$, respectively. In the following discussion, we assume that the bits need for representing a variable nearly equals to its min-entropy. Because $\mathbf{r}_1 \leftarrow_{\$} D_{\mathbb{Z}^{nk}, s_s}$ (see Section 4.2), its min-entropy is about nk by Proposition 9 items (2). As a result, the length of $\mathbf{r}_1$ is about nk bits. In a similar way, the length of $\mathbf{r}_2$ is about nk. Consequently, the ciphertext length of our scheme is about

$$\begin{aligned} &l' + n\log q + m\log 2q \\ &\quad + (n\log q + m_1\log q - nk + nk + nk) \approx 5n\log^2 q + l'. \end{aligned} \tag{27}$$

The ciphertext length of ours is shorter than that of the signature-then-encryption diagram. Furthermore, the longer the length of the plaintext is, the larger our advantage is.

Second, we compare on the computational cost. We first compare on signcryption. The computational cost of the signature-then-encryption diagram mainly consists of the cost of signature and that of public key encryption. The cost of signature is two pre-image sampling by using Algorithm 4. The public key encryption needs roughly $\lceil l'/(nk)\rceil$ times. The main computational cost of our scheme consists of the following: the cost of signature, the cost of public key encryption, and the cost of symmetrical encryption. Because the cost of symmetrical encryption (resp., decryption) is much smaller than the public key encryption (resp., decryption) and signature (resp., verification for signature), it can be ignored. The cost of signature is also a preimage sampling by using Algorithm 4. The public key encryption is one time. Clearly, the computational cost of **Signcrypt** is far less than that of the signature-then-encryption diagram. With the growth of plaintext length, the advantage in the total computational cost of our **Signcrypt** becomes larger.

Next, we compare on **Unsigncrypt**. Our public decryption and signature verification are both one time, while the signature-then-encryption diagram needs $\lceil l'/(nk)\rceil$ times and two times, respectively. From the above analysis, the computational cost of **Signcrypt** and the ciphertext length are much lower those of than the signature-then-encryption diagram, in particular for long plaintext.

6.2. Comparison with the Schemes of LMK12 [14] and WHW12 [15]. Due to employing simpler, tighter, and more efficient trapdoors, our scheme inherits some advantages from the technique suggested by [17]. Now, let us compare the ciphertext length, computational cost, public key size, private key size, security model and security, and so forth among our scheme and the existing lattice-based signcryption schemes such as LMK12 [14] and WHW12 [15].

Firstly, we compare on ciphertext length. The ciphertext of WHW12 [15] is the form $(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{l_0})$ where l_0 is big enough (say, $l_0 \geq 80$) such that 2^{-l_0} is negligible. The length of $\mathbf{b}_i$ is $m'\log q = 6n\log^2 q$ for $i \in [l_0]$. Consequently, the total length of ciphertext is $l' + 6l_0 n\log^2 q$. The ciphertext of LMK12 [14] is the form of $(c, r, \mathbf{u})$. The length of r can be omitted since its length is much smaller than the length of c or $\mathbf{u}$. The length of $\mathbf{u}$ is $m'\log q = 6n\log^2 q$ and the length of c is equal to the length of plaintext. As a result, the ciphertext length of LMK12 [14] is about $l' + 6n\log^2 q$. Our ciphertext length is also $l' + 5n\log^2 q$ (see Section 6.1).

Secondly, we compare the public parameter size. Since in WHW12 [15] scheme and our proposal, the public parameters include several matrices, while in LMK12 [14] scheme the public parameters just include some scales such as n, m', and q, it is convenient to merely count the representation size for the involved matrices. The role of the parameter λ in the proposed scheme is the same as l_0 in WHW12 [15], and we replace it with l_0 for convenience. Then, the public parameter sizes of WHW12 [15] and ours are $2(l_0-1)nm'\log q = 12(l_0-1)n^2\log^2 q$ and $l_0 nm\log q = 2l_0 n^2\log^2 q$, respectively, while

the public parameter size of LMK12 [14] can be neglected. Thirdly, we compare the public/private keys size, that is, the bits number needed for representing the keys. The public key size of WHW12 [15] is $2nm' \log q = 12n^2 \log^2 q$, and the private key size is $2m'^2 \log q = 72n^2 \log^3 q$. The public key size of LMK12 [14] is $nm \log q = 6n^2 \log^2 q$ and the private key size is $m'^2 \log q = 36n^2 \log^3 q$. The public key size of ours is $nm \log q = 2n^2 \log^2 q$, and the private key size is $m_0 nk \log q = m_0 n \log^2 q = n^2 \log^3 q$, where $m_0 = n \log q$ (see Section 4.2).

Finally, let us compare on computational cost. Without loss of generality, the cost of all hash calculation in WHW12 [15] and LMK12 [14] and general hashes H_0, H_1, and H_2 in our scheme are not considered. However, the hashes in our scheme with special structures, such as H_F and $H_{\mathbf{A}_s^{(h)}}$, are taken into account. In detail, our analysis is given below.

(i) First, we compare on the computational cost of **Signcrypt**. In WHW12 [15], the signature cost is one time preimage sampling (PIS) with complexity $\Omega(n^2 \log^2 q)$ [17]; in addition, the cost of its public key encryption is $l_0 nm' = 6l_0 n^2 \log q$ multiplication operations over $\mathbb{Z}_q$, $6l_0 n \log q$ addition operations over $\mathbb{Z}_q$ plus $(l_0 - 1)m' = 6(l_0 - 1)n \log q$ discrete Gaussian samples (DGS); moreover, the cost of symmetric encryption is $t_{\mathcal{E}}$. In LMK12 [14], the cost of its signature is also one time preimage sampling. However, his scheme used preimage obtained from signature as Gaussian error of public key encryption by utilizing the technique in [35] and need not discrete Gaussian sample. As a result, the cost of its public key encryption is $nm' = 6n^2 \log q$ multiplication operations over $\mathbb{Z}_q$ and $6n \log q$ addition operations over $\mathbb{Z}_q$. In addition, his symmetric encryption realized by XOR operation and the corresponding cost can be neglected. In our scheme, the cost of the signature is one time preimage sampling by using Algorithm 4 with complexity $O(n \log q)$ [17], and the cost of symmetric encryption is $t_{\mathcal{E}}$. The cost of public key encryption is about $8n^2 \log q + n^2 \log^2 q$ multiplications over $\mathbb{Z}_q$ plus $l_0/2 \cdot n^2 \log^2 q$ additions over $\mathbb{Z}_q$ and $7n \log q$ DGS.

(ii) Next, we compare on the computational cost of **Unsigncrypt**. In the three schemes, all algorithms **Unsigncrypt** involve the multiplications over $\mathbb{Z}_q$, additions over $\mathbb{Z}_q$, and symmetric decryption operations. The unsigncryption cost of WHW12 [15] is $72n^2 \log^2 q + (l_0 + 6)n^2 \log q$ multiplications over $\mathbb{Z}_q$ plus $6l_0 n \log q$ additions over $\mathbb{Z}_q$. The unsigncryption cost of LMK12 [14] is $72n^2 \log^2 q + 6n^2 \log q$ multiplications over $\mathbb{Z}_q$ plus $6n \log q$ additions over $\mathbb{Z}_q$. The cost of our unsigncryption is about $9n^2 \log q + 3n^2 \log^2 q$ multiplications over $\mathbb{Z}_q$ and $l_0/2 \cdot n^2 \log^2 q$ additions over $\mathbb{Z}_q$. The symmetric decryption in WHW12 [15] and ours is unspecified and in LMK12 [14] is XOR operation. Although the involved symmetric decryption is apparently more expensive than the involved XOR operation in LMK12 [14] scheme, the experiments show that this cost is very small.

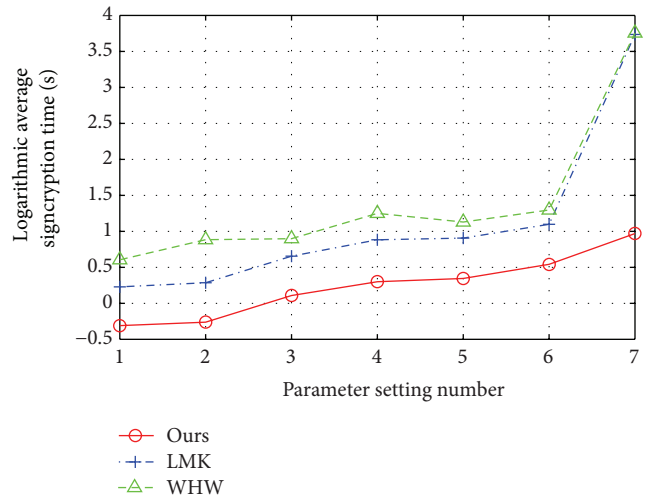

Figure 1: Average signcryption speed.

To summarize, the above comparisons can be collected in Table 2. An overview of this table can also be abstracted in Table 1.

6.3. Simulations. Our simulation environments are given below:

(i) CPU: Intel(R) Core(TM) i7 860 @CPU 2.79 GHz;

(ii) RAM: 8 G;

(iii) OS: Win 7, 64 bit;

(iv) programming platform: Visual Studio 2008.

We conduct 7 simulations with different parameter settings. These settings, given in Table 3, are suggested from [21], [22], and [17], respectively. In particular, under the suggestion of [17], to break a related lattice-based cryptosystem, one needs about 2^{46} core-year computation time by using the state-of-the-art in lattice basis reduction [36, 37] on a 64-bit 1.86 GHz Xeon platform. Note that with the purpose to achieve the same security level, the lattice dimensions ns are different by using different lattice generation techniques.

Then, for each setting, we perform random signcryption and unsigncryption 100 times and then collect the average time cost for signcryption and unsigncryption. The results are given in Table 4 and illustrated in Figures 1 and 2. Note that in these figures, we adopt logarithmic coordinates with the purpose to give visible changes on those data that are different hugely.

From Tables 2, 4, 5, and 6 and Figures 1, 2, 3, 4, 5, and 6, we can see that our improvements are observably as follows.

(i) Under the settings 1~6, the average signcryption time of our scheme increases slowly from 0.489 s to 3.476 s, about 3 times and 5~13 times faster than Li's scheme and Wang's scheme, respectively. Under these settings, the average unsigncryption time of our scheme increases slowly from 0.454 s to 3.48 s, about 3 times and 260 times faster than Li's scheme and Wang's scheme, respectively.

Table 2: Performance comparison with WHW12 [15] and LMK12 [14].

	WHW12 [15]	LMK12 [14]	Ours
Cipher length in bits	$l' + 6l_0 n\log^2 q$	$l' + 6n\log^2 q$	$l' + 5n\log^2 q$
public para. size in bits	$12(l_0 - 1)n^2\log^2 q$	0	$l_0 n^2\log^2 q$
public key size in bits	$12n^2\log^2 q$	$6n^2\log^2 q$	$2n^2\log^2 q$
private key size in bits	$72n^2\log^3 q$	$36n^2\log^3 q$	$n^2\log^3 q$
Sigcry cost			
DGS*	$6(l_0 - 1)n\log q$	0	$7n\log q$
PIS†	1 PIS of [20]	1 PIS of [20]	1 PIS of [17]
$\times_{\mathbb{Z}_q}$‡	$6l_0 n^2\log q$	$6n^2\log q$	$8n^2 log\, q + n^2\log^2 q$
$+_{\mathbb{Z}_q}$**	$6l_0 n\log q$	$6n\log q$	$l_0/2 \cdot n^2\log^2 q$
Unsc cost			
$\times_{\mathbb{Z}_q}$‡	$72n^2\log^2 q + (l_0 + 6)n^2\log q$	$72n^2\log^2 q + 6n^2\log q$	$9n^2\log q + 3n^2\log^2 q$
$+_{\mathbb{Z}_q}$**	$6(l_0 - 1)n\log q$	$6n\log q$	$l_0/2 \cdot n^2\log^2 q$
Security			
Model	RO	RO	ST
Conf.***	IND-CCA2	IND-CCA2	IND-CCA2
UF.#	SUF-CMA	SUF-CMA	SUF-CMA

*Discrete Gaussian sampling, that is, SampleZ: $z_i \leftarrow D_{\mathbb{Z},s_i',c_i'}$ in [20], †preimage sampling, ‡multiplication over $\mathbb{Z}_q$, **addition over $\mathbb{Z}_q$, ***confidentiality, #unforgeability.

Table 3: Parameter Settings.

	Settings no.	1	2	3	4	5	6	7
LMK/WHW	n	128	136	192	214	256	320	436
	q	2053	2003	4093	16381	4093	4093	2^{32}
	m	8448	8976	13824	17976	18432	23040	466644
Ours	n	128	136	192	214	256	320	284
	q	2048	2048	4096	16384	4096	4096	2^{24}
	m	2816	2992	4608	5992	6144	7680	13812
	Referred	[21]	[22]	[21]	[22]	[21]	[21]	[17]

Table 4: Average signcryption/unsigncryption time (s).

	Settings no.	1	2	3	4	5	6	7
LMK	Signcrypt	1.69	1.929	4.514	7.626	8.048	12.53	5434.248
	Unsigncrypt	1.624	1.715	4.062	6.922	7.251	11.72	4515.09
WHW	Signcrypt	4.025	4.535	9.173	14.068	15.3	23.47	5688.381
	Unsigncrypt	2.5635	2.813	6.756	10.431	12.0712	18.415	4695.269
Ours	Signcrypt	0.489	0.547	1.283	1.991	2.212	3.476	9.309
	Unsigncrypt	0.454	0.522	1.257	2.175	2.229	3.48	10.954

Table 5: Average matrix computation cost in signcryption/unsigncryption (s).

	Settings no.	1	2	3	4	5	6	7
Li	Signcrypt	0.012	0.014	0.03	0.044	0.053	0.09	2.248
	Unsigncrypt	1.624	1.715	4.062	6.922	7.251	11.72	4515.09
Wang	Signcrypt	0.975	1.138	2.428	3.553	4.294	7.271	179.971
	Unsigncrypt	2.5635	2.813	6.756	10.431	12.0712	18.415	4695.269
Ours	Signcrypt	0.316	0.356	0.857	1.307	1.504	2.384	7.506
	Unsigncrypt	0.454	0.522	1.257	2.175	2.229	3.48	10.954

Table 6: Average sampling cost in signcryption.

	Settings no.	1	2	3	4	5	6	7
Li	PIS	1.678	1.915	4.484	7.582	7.995	12.44	5432
	DGS	0	0	0	0	0	0	0
Wang	PIS	1.678	1.915	4.484	7.582	7.995	12.44	5432
	DGS	1.372	1.482	2.261	2.933	3.011	3.759	76.41
Ours	PIS	0.155	0.171	0.399	0.659	0.67	1.045	1.748
	DGS	0.018	0.02	0.027	0.025	0.038	0.047	0.055

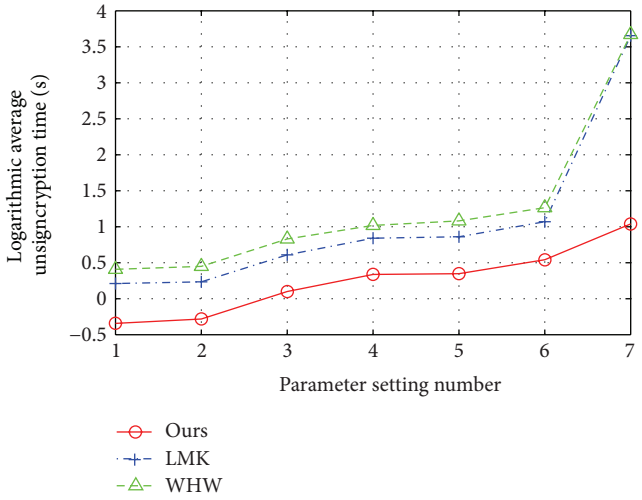

Figure 2: Average unsigncryption speed.

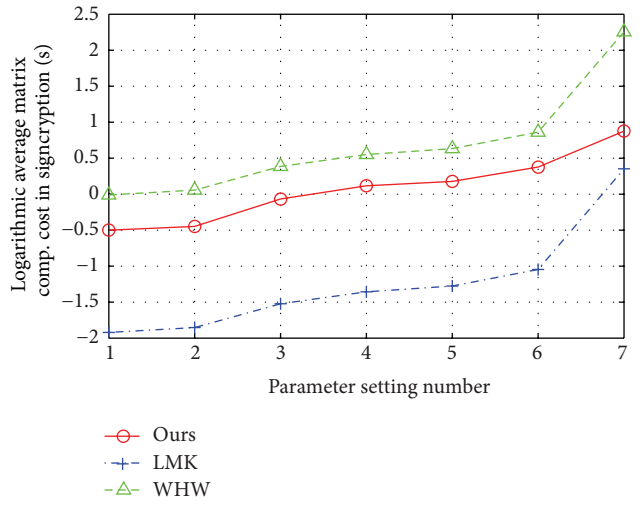

Figure 3: Average matrix computation cost in signcryption.

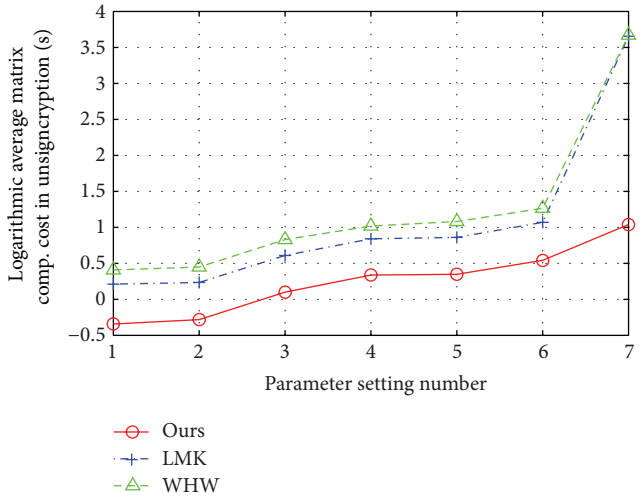

Figure 4: Average matrix computation cost in unsigncryption.

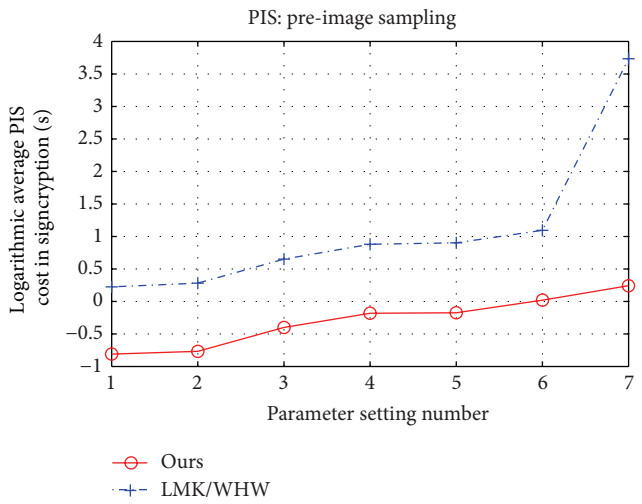

Figure 5: Average preimage sampling cost in signcryption.

(ii) Under the setting 7, the average signcryption time of our scheme is 9.309 s, about 580 times faster than Li's scheme and Wang's scheme. Under this setting, the average unsigncryption time of our scheme is 10.954 s, about 400 times and 33000 times faster than Li' scheme and Wang's scheme, respectively.

To understand the above huge difference on the performance of our scheme, Li's scheme, and Wang's scheme, we would like to give the following further explanations. The time cost of signcryption is mainly occupied by three categories of computation:

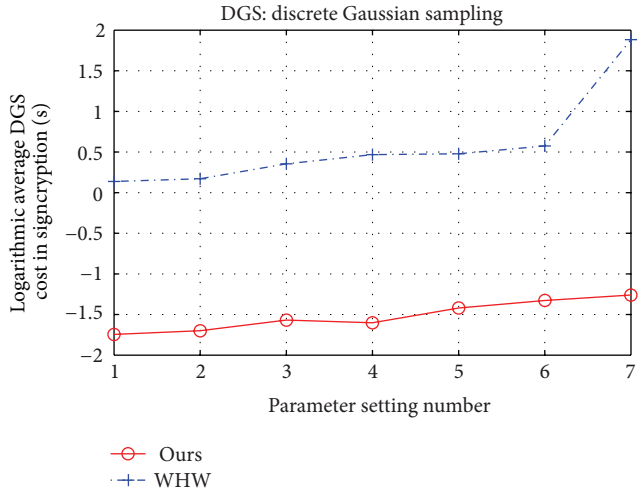

Figure 6: Average discrete gaussian sampling cost in signcryption (s).

(1) matrix operations, including modular addition between matrices and modular multiplication among matrices and vectors,

(2) preimage sampling,

(3) discrete Gaussian sampling.

First, in our signcryption process, the time cost of matrix operations is mainly occupied in step 4(d) and 1(a) (see Section 4.2). Since step 1(a) is directly related to messages that are to be signcrypted, this is not an easy method to optimize this step; say by using precomputation. However, step 4(d) can be optimized since the matrix $\mathbf{Q}$ has nonzero entries merely in the main and the second diagonals. By utilizing this feature, we reduce the computation cost of step 4(d) from $n^3 \log^3 q$ multiplications to $2n \log q$ multiplications. The performance comparisons on average matrix computation cost in signcryption and unsigncryption are given in Table 5 and Figures 3 and 4, respectively. Second, both Li and Wang use the preimage sampling technique given in [20] and its complexity is $\Omega(n^2 \log^2 q)$, while we use the preimage sampling technique given in [17], where the sample oracle is instantiated by the technique given in [28], and its complexity is reduced to $\mathcal{O}(n \log q)$. Third, our signcryption process needs to perform $5n \log q$ times Gaussian sampling, while Wang's signcryption needs to perform in total $6l_0 n \log q$ times Gaussian sampling, where l_0 should enable 2^{-l_0} to be negligible; say $l_0 \geq 80$. Note that Li's signcryption reuses preimages as Gaussian error vectors and thus does not need further Gaussian sampling. The performance comparisons on average sampling cost, including the cost for preimage sampling and the cost for discrete Gaussian sampling, are collected in Table 6 and depicted in Figures 5 and 6. Note that in unsigncryption process, there is no sampling cost.

7. Conclusions

In this paper, we proposed a signcryption scheme in the standard model based on lattice hard problems. The scheme is proven to be indistinguishable against inner adaptively chosen ciphertext attacks under LWE assumption and strongly unforgeable against inner adaptively chosen message attacks under SIS assumption. Moreover, by using simpler, tighter, and more efficient trapdoors suggested by Micciancico and Perkeit, the cost of our scheme is much lower than existing lattice-based signcryption schemes. Another attractive problem is designing an efficient identity-based signcryption scheme in the standard model.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (NSFC) (nos. 61370194, 61103198, 61121061, and 61070251), the NSFC A3 Foresight Program (no. 61161140320), and the JSPS A3 Foresight Program. The third author was also partially supported by JSPS KAKENHI Grant (no. 23500031).

References

[1] Y. Zheng, "Digital signcryption or how to achieve cost(signature & encryption) ≪ cost(signature) + cost(encryption)," in *Proceedings of the Advances in Cryptology: 17th Annual International Cryptology Conference (CRYPTO '97), Santa Barbara, California, USA, 1997*, vol. 1294 of *Lecture Notes in Computer Science*, pp. 17165–21179, Springer, 1997.

[2] R. Steinfeld and Y. A. Zheng, "signcryption scheme based on integer factorization," in *Information Security*, G. Goos, J. Hartmanis, J. Leeuwen, J. Pieprzyk, J. Seberry, and E. Okamoto, Eds., vol. 1975 of *Lecture Notes in Computer Science*, pp. 308–322, Springer, Berlin, Germany, 2000.

[3] J. Malone-Lee and W. Mao, "Two birds one stone: signcryption using RSA," in *Topics in cryptology ł CT-RSA 2003*, M. Joye, Ed., vol. 2612 of *Lecture Notes in Computer Science*, pp. 211–225, Springer, Berlin, Germany, 2003.

[4] X. Boyen, "Multipurpose identity-based signcryption: a Swiss Army knife for identity-based cryptography," in *Proceedings of the Annual International Cryptology Conference (CRYPTO '03)*, vol. 2729, pp. 383–399, Springer, Berlin, Germany, 2003.

[5] L. Chen and J. Malone-Lee, "Improved identity-based signcryption," in *Public Key Cryptography (PKC '05)*, vol. 3386 of *Lecture Notes in Computer Science*, pp. 362–379, Springer, Berlin, Germany, 2005.

[6] P. S. L. M. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," in *Advances in Cryptology (ASIACRYPT '05)*, vol. 3788 of *Lecture Notes in Computer Science*, pp. 515–532, Springer, Berlin, Germany, 2005.

[7] C. H. Bennett and G. Brassard, "Quantum cryptography: public key distribution and coin tossing," in *Proceedings of the IEEE International Conference on ComputersSystems and Signal Processing*, pp. 175–179, IEEE Press, New York, NY, USA, 1984.

[8] A. K. Ekert, "Quantum cryptography based on Bell's theorem," *Physical Review Letters*, vol. 67, no. 6, pp. 661–663, 1991.

[9] A. K. Ekert, J. G. Rarity, P. R. Tapster, and G. Massimo Palma, "Practical quantum cryptography based on two-photon interferometry," *Physical Review Letters*, vol. 69, no. 9, pp. 1293–1295, 1992.

[10] X. Wang and D. Luan, "A novel image encryption algorithm using chaos and reversible cellular automata," *Communications*

in Nonlinear Science and Numerical Simulation, vol. 18, no. 11, pp. 3075–3085, 2013.

[11] X. Y. Wang and X. M. Bao, "A novel block cryptosystem based on the coupled chaotic map lattice," *Nonlinear Dynamics*, vol. 72, no. 4, pp. 707–715, 2013.

[12] H. Liu, X. Wang, and A. Kadir, "Image encryption using DNA complementary rule and chaotic maps," *Applied Soft Computing Journal*, vol. 12, no. 5, pp. 1457–1466, 2012.

[13] D. J. Bernstein, J. Buchmann, and E. Dahmen, *Post Quantum Cryptography*, Springer, New York, NY, USA, 1st edition, 2008.

[14] F. Li, F. T. B. Muhaya, M. K. Khan, and T. Takagi, *Lattice-Based Signcryption*, John Wiley & Sons, New York, NY, USA, 2012.

[15] F. Wang, Y. Hu, and C. Wang, "Post-Quantum secure hybrid signcryption from lattice assumption," *Applied Mathematics and Information Sciences*, vol. 6, no. 1, pp. 23–28, 2012.

[16] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Advances in Cryptology (EUROCRYPT '04)*, vol. 3027, pp. 207–222, Springer, Berlin, Germany, 2004.

[17] D. Micciancio and C. Peikert, "Trapdoors for lattices: simpler, tighter, faster, smaller," in *Advances in Cryptology (EUROCRYPT '12)*, D. Pointcheval and T. Johansson, Eds., vol. 7237 of *Lecture Notes in Computer Science*, pp. 700–718, Springer, Berlin, Germany, 2012.

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

[19] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," *Journal of Cryptology*, vol. 25, no. 4, pp. 601–639, 2012.

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

[21] R. Lindner and C. Peikert, "Better key sizes (and attacks) for LWE-based encryption," in *Proceedings of the 11th International Conference on Topics in Cryptology (CT-RSA '11)*, pp. 319–339, Springer, Berlin, Germany, 2011.

[22] D. Micciancio and O. Regev, "Lattice-based cryptography," in *Post-Quantum Cryptography*, D. Bernstein, J. Buchmann, and E. Dahmen, Eds., pp. 147–191, Springer, Berlin, Germany, 2009.

[23] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC '05)*, pp. 84–93, ACM, New York, NY, USA, 2005.

[24] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems: A Cryptographic Perspective*, vol. 671, Springer, New York, NY, USA, 2002.

[25] Y. Eldar and G. Kutyniok, Eds., *Compressed Sensing, Theory and Applications*, chapter 5, Cambridge University Press, New York, NY, USA, 2012.

[26] M. N. Wegman and J. L. Carter, "New hash functions and their use in authentication and set equality," *Journal of Computer and System Sciences*, vol. 22, no. 3, pp. 265–279, 1981.

[27] H. Krawczyk and T. Rabin, "Chameleon hashing and signatures," IACR Eprint archive, March 1998, http://eprint.iacr.org/1998/010.

[28] C. Peikert, "An efficient and parallel Gaussian sampler for lattices," in *Advances in Cryptology (CRYPTO '10)*, T. Rabin, Ed., vol. 6223 of *Lecture Notes in Computer Science*, pp. 80–97, Springer, Berlin, Germany, 2010.

[29] J. Alwen and C. Peikert, "Generating shorter bases for hard random lattices," *Theory of Computing Systems*, vol. 48, no. 3, pp. 535–553, 2011.

[30] Z. Yuliang, "Signcryption," http://www.signcryption.org/.

[31] A. W. Dent, "Hybrid signcryption schemes with insider security," in *Information Security and Privacy*, C. Boyd and J. Gonzlez Nieto, Eds., pp. 253–266, Springer, Berlin, Germany, 2005.

[32] A. W. Dent, "Hybrid signcryption schemes with outsider security," in *Information Security*, J. Zhou, J. Lopez, R. Deng, and F. Bao, Eds., vol. 3650 of *Lecture Notes in Computer Science*, pp. 203–217, Springer, Berlin, Germany, 2005.

[33] S. Hohenberger and B. Waters, "Short and stateless signatures from the RSA assumption," in *Advances in Cryptology (CRYPTO '09)*, S. Halevi, Ed., vol. 5677 of *Lecture Notes in Computer Science*, pp. 654–670, Springer, Berlin, Germany, 2009.

[34] M. Rückert, "Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles," in *Post-Quantum Cryptography*, N. Sendrier, Ed., vol. 6061 of *Lecture Notes in Computer Science*, pp. 182–200, Springer, Berlin, Germany, 2010.

[35] S. D. Gordon, J. Katz, and V. Vaikuntanathan, "A group signature scheme from lattice assumptions," in *Advances in Cryptology (ASIACRYPT '10)*, M. Abe, Ed., vol. 6477 of *Lecture Notes in Computer Science*, pp. 395–412, Springer, Berlin, Germany, 2010.

[36] N. Gama and P. Q. Nguyen, "Predicting lattice reduction," in *Advances in Cryptology (EUROCRYPT '08)*, N. Smart, Ed., vol. 4965 of *Lecture Notes in Computer Science*, pp. 31–51, Springer, Berlin, Germany, 2008.

[37] Y. Chen and P. Q. Nguyen, "BKZ 2.0: better lattice security estimates," in *Advances in Cryptology C (ASIACRYPT '11)*, D. Lee and X. Wang, Eds., vol. 7073 of *Lecture Notes in Computer Science*, pp. 1–20, Springer, Berlin, Germany, 2011.