SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2015)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.1297

RESEARCH ARTICLE

Identity-based signcryption from lattices

Jianhua Yan[1,2], Licheng Wang[1]*, Mianxiong Dong[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] National Institute of Information and Communications Technology, 3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan

ABSTRACT

Signcryption as a cryptographic primitive can carry out signature and encryption simultaneously at a remarkably reduced cost. Identity-based cryptography is more convenient than public key infrastructure-based cryptography in certificate management. As a result, identity-based signcryption has been studied extensively, and many efficient and provably secure constructions have been proposed. However, most of these schemes are based on intractability assumptions from number theory, and these assumptions have been threatened by the booming quantum computation. Therefore, a recent trend in cryptography is to construct cryptosystems that are based on lattice-based intractability assumptions because of their plausible features of quantum attack resistance. In this paper, several identity-based signcryption schemes from lattice hardness assumptions are proposed. In the standard model, these schemes are indistinguishable against *inner* adaptively chosen ciphertext attacks (IND-CCA2) and strongly unforgeable against *inner* chosen message attacks. In our construction, it does not matter whether the original encryption scheme used to construct signcryption is deterministic or probabilistic; the resulted signcryption schemes can reach IND-CCA2 security. To achieve this, we carefully combine three techniques—the identity-based encryption from lattice due to Agrawal–Boneh–Boyen (EUROCRYPT 2010), the framework of lattice-based short signature due to Boyen (Public Key Cryptography 2010), and the Canetti–Halevi–Katz (abbr. CHK) technique, with necessary and tailored optimization—for transforming an $(\ell+1)$-level indistinguishable under chosen plaintext attack secure hierarchical identity-based encryption (HIBE) into an ℓ level IND-CCA2 secure HIBE scheme. In addition, our security proof also contains a more efficient simulation tool that might have separate interest in cryptographic applications.

KEYWORDS
signcryption; lattice; standard model; learning with errors; learning with rounding; short integer solution

***Correspondence**
Licheng Wang, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China.
E-mail: wanglc@bupt.edu.cn

1. INTRODUCTION

In the public key infrastructure-based cryptography, each user generates his or her own secret key and submits his or her public key to the certificate authority. The certificate authority generates a corresponding certificate for the user and keeps it in the directory of certificates. When user A wants to encrypt a message to user B, A needs to query the certificate authority to fetch the public key of user B, followed by encrypting the message. For efficiency and convenience, Shamir [1] proposed the identity-based framework. In the identity-based framework, each user uses his or her identity information such as email address and staff number as his or her public key. The user's private key is generated by private key generator (PKG). As a result, when two users communicate, they do not need the online service of the certificate authority or to keep the corresponding public keys in their lists. Hence, it is meaningful to design identity-based schemes.

In many cases, it is necessary to simultaneously realize confidentiality, integrity, authentication, and non-repudiation. The confidentiality can be achieved by encryption, while the integrity, authentication, and non-repudiation can be fulfilled by digital signature. As a result, a natural approach to accomplish the aforementioned task is signature-then-encryption. However, the signcryption proposed by Zheng [2] is a better choice. The signcryption can perform both signature and encryption in a logical

step, at an obviously lower cost than the signature-then-encryption mechanism. Because of this fact, the signcryption is suitable in many environments such as electronic commerce, smart cards, mobile communications, and key management. The signcryption has been widely studied, and many efficient schemes have been proposed [3–6].

Because of the lower overhead of signcryption and the convenience of identity-based framework, it is natural to find an efficient method to implement the signcryption primitive under the identity-based framework. In fact, many efficient identity-based signcryption (IBSC) schemes have been proposed in the last few years [5,7–9]. However, all these schemes are based on the hardness assumptions from number theory. The boom of the quantum computation seriously threatens the security of cryptosystems based on these number-theoretical assumptions. Cryptographers have carried out numerous research to find new cryptographic tools that can resist (known) quantum attacks.

Lattice becomes one of the most attractive cryptographic tools to resist the known quantum attacks [10]. In fact, lattice-based cryptography has two other important advantages. On one hand, its security is based on the worst-case hardness of lattice problems. In other words, if an adversary can break an average case, then it can succeed in solving any instance of a certain lattice problem. On the other hand, lattice-based cryptographic primitive only needs modular additions and multiplications. As a result, lattice-based cryptography has developed rapidly and obtained a series of fruits such as encryption schemes [11–13], signature schemes [11,14,15], fully homomorphic encryption [16–19], and functional encryption [20,21]. Therefore, it is of both theoretical and practical interest to design an IBSC from lattice-based assumptions.

Our main contribution is to design an IBSC scheme from lattices. Our first construction is based on two recently developed techniques: the identity-based encryption from lattices due to Agrawal–Boneh–Boyen [22] and the framework of lattice-based short signature due to Boyen [23]. As far as we know, this is the first lattice-based IBSC scheme that is proven to be selective-ID secure in the standard model. Moreover, it can also be extended to be fully secure against adaptively chosen identity attacks by using the artificial abort technique [22,24]. The proposed signcryption has the following three merits: (i) The involved framework for constructing signcryption is flexible in the sense that whether the underlying encryption scheme is deterministic or probabilistic, the signcryption schemes can be proven indistinguishable against *inner* adaptively chosen ciphertext attacks (IND-CCA2). (ii) The involved framework is more efficient than the universal transformation from an $(\ell + 1)$-level indistinguishable under chosen plaintext attack secure hierarchical identity-based encryption (HIBE) scheme to an ℓ-level IND-CCA2 secure HIBE scheme [25,26]. More precisely, our construction saves a strongly unforgeable signature. In fact, the cost of such a signature scheme is more expensive than an encryption scheme, so the proposed construction is more efficient. (iii) The involved signature is compact. More specifically, we combine the form of identity [22] and the technique of mixing public key matrices [23] to obtain an identity-based signature with short length. In addition, other highlights of our proposal involve a more efficient pre-image sampling algorithm, named **SampleRightv**, and an identity-based deterministic encryption. The algorithm **SampleRightv** is designed to sample a pre-image $\mathbf{x}$ to satisfy $\mathbf{Ax} = \mathbf{y}$ by using the trapdoor of $\mathbf{\Lambda}^{\perp}(\mathbf{B})$, where $\mathbf{A} = [\mathbf{A_1} \| \mathbf{A_1 R} + \mathbf{HB} \| \mathbf{C}]$. The general method is to run **SampleBasisRight** and **SamplePre** in a successive manner. The cost of **SampleRightv** in an answer for a single signcryption/unsigncryption query is about 1/8 of the general method if the secret key extraction query for the same identity has been asked. Otherwise, the cost of **SampleRightv** is merely $1/O(m \log m)$ of the general method. The involved encryption scheme is the first identity-based deterministic encryption scheme based on learning with rounding (LWR) assumption.

The rest of this paper is organized as follows. In Section 2, the necessary preliminaries on lattice-based cryptographic assumptions are introduced. In Section 3, the security models of selectively secure IBSC, including the indistinguishable against inner selective identity chosen ciphertext attacks (IND-sID-CCA2) game and strongly unforgeable against inner selective identity adaptively chosen message attacks (SUF-sID-CMA) game, are reviewed. In Section 4, the main contribution, that is, a selective IBSC scheme from LWR assumption, is presented in detail, followed by the proofs on its consistency and security. The selectively secure scheme from learning with errors (LWE) assumption is given in Section 5. The adaptively secure schemes from LWR and LWE assumptions are proposed in Section 6. The performance of the proposed schemes is given in Section 7. Finally, the concluding remarks are given in Section 8.

2. PRELIMINARIES

We list the notations and their representation in Table I.

2.1. Lattice and hardness assumptions

Definition 1 (Lattice). *Let a basis* $\mathbf{B} = \{\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_n}\} \in \mathbb{R}^{m \times n}$ *be composed by n linearly independent vectors. The lattice generated by* $\mathbf{B}$ *is*

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

In many cryptographic applications, for a given matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*, q-ary integer lattice is defined as*

$$\mathbf{\Lambda}^{\perp}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = \mathbf{0} \bmod q\} \tag{2}$$

For a vector $\mathbf{y} \in \mathbb{Z}_q^n$*, a coset of q-ary integer lattice is defined as*

DOI: 10.1002/sec

Table I. Notations.

$\mathbb{Z}/\mathbb{R}$: integers / real numbers	$\mathbb{Z}^n/\mathbb{Z}_q^n, \mathbb{R}^n$: vectors space on $\mathbb{Z}/\mathbb{Z}_q, \mathbb{R}$
$\mathbb{T}$: real interval $[0, 1)$	$\mathbb{Z}^{n\times m}/\mathbb{Z}_q^{n\times m}$: matrices space on $\mathbb{Z}/\mathbb{Z}_q$
$\mathbb{Z}_q$: residue class mod q	lower-case and bold letters : vectors
$[k]$: $\{1, 2, \ldots, k\}$	upper-case and bold letters : matrices
$\lfloor\rfloor/\lceil\rfloor$: round down/nearly	$\tilde{\mathbf{A}}$: Gram–Schmidt orthogonalization of $\mathbf{A}$
$s_1(\cdot)$: the largest singular value of a matrix	$U(\mathcal{P})$: the uniform distribution over $\mathcal{P}$
$\mathbf{s} \xleftarrow{\$} \mathcal{P}$: choose from $\mathcal{P}$ uniformly and randomly	$\|$: matrices concatenation operators
$\mathbf{s} \leftarrow \chi(\mathcal{P})$: choose from P according to χ	$\|\cdot\|$ the maximum singular value
abbreviate as $\mathbf{s} \leftarrow \chi$ for explicit P	of a matrix or the 2-norm of a vector

$$\mathbf{\Lambda}_{\mathbf{y}}^{\perp}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = \mathbf{y} \bmod q\} \tag{3}$$

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

Definition 2 (Learning with errors [10])**.** *For an integer $q = q(n)$ and a distribution χ on $\mathbb{Z}_q$, the target of $LWE_{q,\chi}$ is to distinguish between the distribution $\mathbf{A}_{\mathbf{s},\chi}$ and $U\left(\mathbb{Z}_q^n \times \mathbb{Z}_q\right)$, where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$.*

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

Proposition 1 (Hardness of LWE [10])**.** *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 quantum) algorithm that can solve $LWE_{q,\bar{\Psi}_{\alpha}}$, then there is an efficient quantum algorithm for approximating the shortest independent vector problem ($SIVP_{\gamma}$) within $\tilde{O}(n/\alpha)$ factors (referring to* [27] *for its hardness) in the worst case.*

Proposition 2 (Learning with rounding and hardness [28])**.** *For security parameter λ, integer $n = n(\lambda), m = m(\lambda), q = q(\lambda), p = p(\lambda)$, the $LWR_{n,m,q,p}$ problem is to distinguish the distribution of $(\mathbf{A}, \lfloor \mathbf{As} \rfloor_p)$ and $(\mathbf{A}, \lfloor \mathbf{u} \rfloor)$, where $A \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n, \mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$. Let β denote the bound of χ. If $LWE_{l,m,q,\chi}$ assumption holds and $q \geq 2\beta\gamma nmp$, $n \geq (l + \lambda + 1)\log q / \log 2\gamma + 2\lambda$, then the two distributions are computationally indistinguishable.*

Definition 3 (Small integer solution (SIS) [29])**.** *Given an integer q, a real $\beta > 0$ and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the goal of $SIS_{q,\beta}$ is to find a $\mathbf{z} \in \mathbb{Z}^m$ to satisfy $\mathbf{z} \neq \mathbf{0}, \mathbf{Az} = \mathbf{0} \bmod q$ and $\|\mathbf{z}\| \leq \beta$.*

Proposition 3 (Hardness of SIS Theorem 5.16 [29])**.** *Given a security parameter n, any poly-bounded m, $\beta = poly(n)$ and any prime $q \geq \beta \cdot \omega(\sqrt{n \log n})$, the average-case problem $SIS_{q,\beta}$ is as hard as approximating the SIVP problem in the worst case to within certain $\gamma = \beta \cdot \tilde{O}(\sqrt{n})$ factors.*

Proposition 4 (Theorem 3.2 [30])**.** *Let $\delta > 0$ be a fixed value. The* **TrapGen**(n, m, q) *algorithm takes inputs $n, q > 2$ and $m \geq (5 + 3\delta) \cdot n \log q$ and outputs $\mathbf{A} \in \mathbb{Z}_q^{n\times m}, \mathbf{S} \in \mathbb{Z}_q^{m\times m}$, such that $\mathbf{A}$ is with negligible distance from $U\left(\mathbb{Z}_q^{n\times m}\right)$ and $\|\mathbf{S}\| \leq O(n \log q)$, $\|\tilde{\mathbf{S}}\| \leq O(\sqrt{n \log q})$ with overwhelming probability (w.o.p.).*

2.2. Gaussian distribution and sampling algorithms

Definition 4 (Discrete Gaussian distribution [29])**.** *For any vector $\mathbf{c}$, real $s > 0$, and lattice $\mathbf{\Lambda}$, the discrete Gaussian distribution over $\mathbf{\Lambda}$ is defined as*

$$\forall \mathbf{x} \in \mathbf{\Lambda}, D_{\mathbf{\Lambda},s,\mathbf{c}}(\mathbf{x}) = \frac{D_{s,\mathbf{c}}(\mathbf{x})}{D_{s,\mathbf{c}}(\mathbf{\Lambda})} = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\mathbf{\Lambda})} \tag{4}$$

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

Proposition 5. *Let $\mathbf{B}$ be a basis of $\mathbf{\Lambda}^{\perp}(\mathbf{A})$, where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and the columns of $\mathbf{A}$ generate $\mathbb{Z}_q^n$. Let $s \geq \|\tilde{\mathbf{B}}\|\omega(\sqrt{\log n})$,*

(1) (Theorem 3.1 [11]*) Let $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,s}$, the distribution of $\mathbf{y} = \mathbf{Ax} \in \mathbb{Z}_q^n$ is negl(n)-far from $U\left(\mathbb{Z}_q^n\right)$, and the conditional distribution of $\mathbf{x}$ given $\mathbf{y}$ is $D_{\mathbf{\Lambda}_{\mathbf{y}}^{\perp}(A),s}$.*

(2) (Lemma 4.4 [29]*) $\Pr_{\mathbf{x}\sim D_{\mathbf{\Lambda},s,\mathbf{v}}}\{\|\mathbf{x} - \mathbf{v}\| > s\sqrt{n}\} \leq \frac{1+\epsilon}{1-\epsilon} \cdot 2^{-n}$.*

(3) ([11]*) For arbitrary $\mathbf{y} \in \mathbb{Z}_q^n$,* **SamplePre**$(\mathbf{B}, \mathbf{A}, \mathbf{y}, s)$ *outputs a pre-image $\mathbf{x} \in \mathbb{Z}^m$ with distribution statistically close to $D_{\mathbf{\Lambda}_{\mathbf{y}}^{\perp}(\mathbf{A}),s}$, and the min-entropy of $\mathbf{x}$ is at least $\omega(\log n)$.*

Proposition 6 (Theorem 17 [22])**.** *Let $q > 2, m > n$.* **SampleLeft**$(\mathbf{A}, \mathbf{B}, \mathbf{T_A}, \mathbf{y}, s)$ *takes inputs $\mathbf{B} \in \mathbb{Z}_q^{n\times m'}$, $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the trapdoor for $\mathbf{\Lambda}^{\perp}(\mathbf{A}), s > \|\widetilde{\mathbf{T_A}}\| \cdot \omega(\sqrt{\log(m + m')})$ and $\mathbf{y} \in \mathbb{Z}^m$, and outputs $\mathbf{x} \in \mathbb{Z}^{m+m'}$ with negligible distance with $D_{\mathbf{\Lambda}_{\mathbf{y}}^{\perp}(\mathbf{A}\|\mathbf{B}),s}$.*

Proposition 7 (Theorem 19 [22]). *Let $q > 2, m > n$. $\mathbf{SampleRight}(\mathbf{A}, \mathbf{B}, \mathbf{H}, \mathbf{R}, \mathbf{T_B}, \mathbf{y}, s)$ takes inputs $\mathbf{A} \in \mathbb{Z}_q^{n\times m'}, \mathbf{R} \in \mathbb{Z}_q^{m'\times m}, \mathbf{B} \in \mathbb{Z}_q^{n\times m}$, invertible matrix $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$, the trapdoor for $\mathbf{\Lambda}^{\perp}(\mathbf{B}), s > \|\widetilde{\mathbf{T_B}}\| \cdot s_R\omega(\log m)$ and $\mathbf{y} \in \mathbb{Z}^m$, and outputs $\mathbf{x} \in \mathbb{Z}^{m+m'}$ with negligible distance with $D_{\mathbf{\Lambda}_{\mathbf{y}}^{\perp}(\mathbf{F}),s}$, for $\mathbf{F} = [\mathbf{A}\|\mathbf{AR} + \mathbf{HB}]$.*

Proposition 8 (Lemma 29, Corollary 30, Corollary 31 [22]). $\mathbf{SampleBasisRight}(\mathbf{A}, \mathbf{B}, \mathbf{H}, \mathbf{R}, \mathbf{T_B}, s)$ *on inputs as* **SampleRight** *except not required for* $\mathbf{y}$*, runs* **SampleRight** *less than $O(m \log m)$ w.o.p. $2m$ times with $\mathbf{y} = \mathbf{0}$, and outputs a basis $\mathbf{T}$ for $\mathbf{\Lambda}^{\perp}(\mathbf{F})$, where $\mathbf{F} = [\mathbf{A}\|\mathbf{AR}+\mathbf{HB}]$, $\|\mathbf{T}\| \le s\sqrt{m}$. Similarly,* $\mathbf{SampleBasisLeft}(\mathbf{A}, \mathbf{AR} + \mathbf{HB}, \mathbf{T_A}, s)$ *on inputs as* **SampleRight** *except not required for* $\mathbf{y}$*, runs* **SampleLeft** *less than $O(m \log m)$ w.o.p. $2m$ times with $\mathbf{y} = \mathbf{0}$, and outputs a basis $\mathbf{T}'$ for $\mathbf{\Lambda}^{\perp}(\mathbf{F}')$ where $\mathbf{F}' = [\mathbf{A}\|\mathbf{C}]$ and $\|\mathbf{T}'\| \le s\sqrt{m}$. For identical $\mathbf{A}, s$ and $\mathbf{C} = \mathbf{AR} + \mathbf{HB}$, the two bases $\mathbf{T}$ and $\mathbf{T}'$ are statistically close.*

2.3. Chameleon hash and others

Chameleon hash was firstly proposed in [31], and all the chameleon hash functions have four essential properties [6,31,32]: (i) efficient forward computation, (ii) collision-resistance property, (iii) uniformity property, and (iv) chameleon property.

Proposition 9 (Lemma 4.1 [32]). *Let $n \ge 1, q \ge 2, m = O(n \log q), k \ge 1$ be integers and $s = O(\sqrt{n \log q})$ be real number. With respect to $\mathbf{A_0} \in \mathbb{Z}_q^{n\times k}, \mathbf{A_1} \in \mathbb{Z}_q^{n\times m}, \mathbf{A} = \mathbf{A_0}\|\mathbf{A_1}$, define hash function $h_{\mathbf{A}} : \mathcal{M} \times \mathcal{R} \to \mathcal{Y}$ as $h_{\mathbf{A}}(\mathbf{m}, \mathbf{r}) = \mathbf{A}(\mathbf{m}\|\mathbf{r}) = \mathbf{A_0}\mathbf{m} + \mathbf{A_1}\mathbf{r}$, where $\mathcal{M} \in \{0,1\}^k$ is message space, randomness space $\mathcal{R} = \{\mathbf{r} \in \mathbb{Z}^m | \|\mathbf{r}\| \le s\sqrt{m}\}$ has distribution $D_{\mathbb{Z}^m,s}$, and $\mathcal{Y} = \mathbb{Z}_q^n$ is range. The hash family $\mathcal{H} = \{h_{\mathbf{A}}\}$ is a chameleon hash functions family, supposing the hardness of $SIS_{q,\beta}$ for $\beta = \sqrt{k + 4s^2m}$.*

Proposition 10 (Lemma 15 [22]). *When $\mathbf{R} \xleftarrow{\$} \{-1, 1\}^{k\times m}$, there is a universal C such that $\Pr[\|\mathbf{R}\| > C\sqrt{k+m}] < e^{-(k+m)}$ and $C = 12$ is sufficient.*

Proposition 11 (Leftover hash lemma, Lemma 13 [22]). *Let prime $q > 2$, $m > (n+1)\log_2 q + \omega(\log n)$, $k = k(n)$ with the polynomial size of n, $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{B} \xleftarrow{\$} \mathbb{Z}_q^{n\times k}$ and $\mathbf{R} \xleftarrow{\$} \{1,-1\}^{m\times k}$, respectively. Then, the distribution of $(\mathbf{A}, \mathbf{AR}, \mathbf{R}^T\mathbf{z})$ is with negligible distance from the distribution $(\mathbf{A}, \mathbf{B}, \mathbf{R}^T\mathbf{z})$, where $\mathbf{z} \in \mathbb{Z}_q^m$ is arbitrary.*

3. SIGNCRYPTION: PRIMITIVE AND SECURITY MODELS

Definition 5 (Identity-based signcryption). *An IBSC scheme consists of the following four algorithms:*

- $\mathbf{Setup}(1^n)$*: This algorithm is executed by a PKG. It takes a security parameter 1^n as input and outputs public parameters $\mathcal{P}_p$, master public key mpk, and master private key msk.*
- $\mathbf{Extract}(msk, ID)$*: PKG executes this algorithm to generate private key for an identity ID. This algorithm takes as inputs an identity ID and the master private key msk and outputs the corresponding private key SK_{ID} for the identity ID.*
- $\mathbf{Signcrypt}(u, SK_{ID_s}, ID_r)$*: The sender executes this algorithm to generate a signcryption ciphertext for a given message u. This algorithm takes as inputs a message u, the sender's private key SK_{ID_s}, and the receiver's identity ID_r, then outputs a signcryption ciphertext c.*
- $\mathbf{Unsigncrypt}(c, SK_{ID_r}, ID_s)$*: The receiver runs this algorithm to carry out the unsigncryption operation. This algorithm takes as inputs a signcryption ciphertext c, the receiver's identity ID_r, private key SK_{ID_r}, and the sender's identity ID_s, then outputs the corresponding plaintext u or $\perp$. Here, $\perp$ represents an invalid ciphertext.*

Definition 6 (Consistency of signcryption). *The successful probability of the unsigncryption for a signcryption scheme is defined as follows:*

$$p = \Pr\left[\begin{array}{l} \mathcal{P}_p \leftarrow \mathbf{Setup}(1^n); \\ SK_{ID_r} \leftarrow \mathbf{Extract}(msk, ID_r); \\ SK_{ID_s} \leftarrow \mathbf{Extract}(msk, ID_s); \\ c \leftarrow \mathbf{Signcrypt}(u, SK_s, ID_r); \\ u' \leftarrow \mathbf{Unsigncrypt}(c, SK_{ID_r}, ID_s) : u' = u \end{array}\right] \tag{5}$$

If the probability $1 - p$ is negligible with respect to the security parameter n, then we say that the signcryption scheme is consistent.

Although the essence of confidentiality of an IBSC scheme is similar to that of the general IBE, there are some differences, especially for inner security. We refer to [9,33,34] to introduce a game, named Game IND-sID-CCA2, between adversary $\mathcal{A}$ and challenger $\mathcal{C}$ as follows.

Game IND-sID-CCA2

- Initial: $\mathcal{A}$ chooses an identity ID_r^* that it wants to attack and sends ID_r^* to $\mathcal{C}$. $\mathcal{C}$ runs $\mathbf{Setup}(1^k)$ algorithm to generate public parameters $\mathcal{P}_p$ and a master public key then sends them to $\mathcal{A}$, but $\mathcal{C}$ keeps the master private key to itself.
- Phase 1: $\mathcal{A}$ can perform polynomially bounded queries as follows:
  - Key extraction queries (ID_i): $\mathcal{A}$ selects a receiver's identity ID_i that it wants to query

and sends ID_i to $\mathcal{C}$. If $ID_i \neq ID_r^*$, $\mathcal{C}$ computes $SK_{ID_i} = \textbf{Extract}(msk, ID_i)$, and gives it to $\mathcal{A}$; otherwise, $\mathcal{C}$ gives $\perp$.
– Signcrypt queries (u, ID_s, ID_r): $\mathcal{A}$ selects a message u, a sender's identity ID_s, and a receiver's identity ID_r, then submits them to $\mathcal{C}$. $\mathcal{C}$ runs $\textbf{Extract}(msk, ID_s)$ to obtain SK_{ID_s}, then computes $c = \textbf{Signcrypt}\ (u, SK_{ID_s}, ID_r)$, and uses c as the reply.
– Unsigncrypt queri es (c, ID_r, ID_s): $\mathcal{A}$ submits (c, ID_r, ID_s) as a query. If $ID_r \neq ID_r^*$, $\mathcal{C}$ performs $SK_{ID_r} = \textbf{Extract}(msk, ID_r)$, then execute $u = \textbf{Unsigncrypt}(c, SK_{ID_r}, ID_s)$, followed by replying u to $\mathcal{A}$.

- Challenge: $\mathcal{A}$ chooses two equal length messages u_0, u_1 and a sender's identity ID_s and sends them to $\mathcal{C}$. $\mathcal{C}$ flips $b \in \{0,1\}$ fairly, then computes $c^* = \textbf{Signcrypt}\left(u_b, SK_{ID_s}, ID_r^*\right)$, followed by sending c^* as the challenge ciphertext.
- Phase 2: $\mathcal{A}$ continues the queries as in Phase 1, except that he or she cannot ask unsigncryption query on $\left(c^*, ID_r^*, ID_s\right)$, where ID_s is the same sender's identity as in the challenge.
- Guess: $\mathcal{A}$ outputs a bit b' as its guess on b.

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

Definition 7 (Confidentiality of signcryption)**.** *An IBSC scheme is said to be IND-sID-CCA2, if there exists no probabilistic polynomial time adversary that can win* **GameIND – sID – CCA2** *with non-negligible advantage.*

To capture the (strong) unforgeability of an IBSC defined earlier, let us introduce another game, denoted by Game SUF-sID-CMA, played between a challenger $\mathcal{C}$ and a forger $\mathcal{F}$ as follows.

Game SUF-sID-CMA

- Initial: $\mathcal{F}$ submits an identity ID_s^* to $\mathcal{C}$ as the target identity. $\mathcal{C}$ executes $\textbf{Setup}(1^k)$ algorithm to generate public parameters $\mathcal{P}_p$, master public key mpk and master secret key msk.
- Query: $\mathcal{F}$ makes polynomially bounded queries in an adaptive manner. Concretely, $\mathcal{A}$ is allowed the same queries as in Game IND-sID-CCA2.
- Forgery: $\mathcal{F}$ outputs a tuple $\left(u^*, c^*, ID_r, ID_s^*\right)$ such that c^* is not an answer for signcryption query on (u^*, ID_r).

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

$$Adv(\mathcal{F}) = \Pr\left[u^* = \textbf{Unsigncrypt}\left(c^*, ID_s^*, ID_r\right)\right]$$

Definition 8 (Strong unforgeability of signcryption)**.** *An IBSC scheme is said to be SUF-sID-CMA, if there is no probabilistic polynomial time forger who can win* **Game SUF – sID – CMA** *with non-negligible advantage.*

4. SELECTIVELY SECURE IDENTITY-BASED SIGNCRYPTION FROM LWR ASSUMPTION

In the first construction, denoted by SR_SC, the involved identity-based encryption scheme is based on the LWR problem [28,35]. However, it is different from the deterministic scheme in [28]. We adopt the form of the identity matrix [22] and introduce appropriate compensation vector in the ciphertext to design an identity-based deterministic encryption scheme. On the other hand, we adopt the mixing technique of Boyen [23] and the identity matrix form to obtain an IBS with short signatures Finally, we borrowed the translating framework [25] but omit the strong unforgeable signature used to guarantee the non-malleable property of ciphertext by proper designing to obtain the first IBSC scheme.

4.1. Construction

The IBSC involves the following four algorithms:

- **Setup**(1^n): PKG takes a security parameter 1^n as input and produces public parameters $\mathcal{P}_p$, a master public key mpk and a master secret key msk as follows:

  (1) Choose $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$, where H is an encoding with full-rank differences [22].
  (2) Choose appropriate positive integers $\gamma, \iota, \kappa, \rho, \varrho$.
  (3) Choose the following hash functions:

  – $H_1 : \{0,1\}^* \times \{0,1\}^\gamma \times \mathbb{Z}_q^n \to \{0,1\}^\kappa$.
  – $H_2 : \{0,1\}^\rho \times \mathbb{Z}_p^n \times \mathbb{Z}_p^n \to \{0,1\}^\varrho$.
  – $H_N : \{0,1\}^\iota \times \left\{\mathbf{r} \in \mathbb{Z}^{m'} : \|\mathbf{r}\| \leq \tilde{s} \cdot \sqrt{m'}\right\} \to \mathbb{Z}_q^n$, where $\tilde{s} = O(\sqrt{n \log q}) \cdot \omega(\sqrt{\log n})$. H_N has identical form with the chameleon hash functions in [32], namely H_N is specified by a matrix $\mathbf{N} = [\mathbf{N_0} \| \mathbf{N_1}]$, where $\mathbf{N_0} \in \mathbb{Z}_q^{n \times \iota}$ and $\mathbf{N_1} \in \mathbb{Z}_q^{n \times m'}$.

  (4) Generate master public key and secret key:

  – Generate uniformly random matrix $\mathbf{A_0} \in \mathbb{Z}_q^{n \times m}$ with the corresponding

trapdoor $\mathbf{T_{A_0}} \in \mathbb{Z}_q^{m\times m}$ for $\mathbf{\Lambda}^{\perp}(\mathbf{A_0})$ by running algorithm $\mathbf{TrapGen}(n, m, q)$.
– Select uniformly random matrices $\mathbf{A_1}, \mathbf{A_2}, \mathbf{B_1}, \mathbf{B_2} \in \mathbb{Z}_q^{n\times m}, \mathbf{G} \in \mathbb{Z}_q^{n\times \ell}, \mathfrak{F} = \left\{\mathbf{F_i} \in \mathbb{Z}_q^{n\times m}\right\}_1^{\kappa}$ and a vector $\mathbf{y} \in \mathbb{Z}_q^n\backslash\{\mathbf{0}\}$; $mpk \triangleq \{\mathbf{A_0}, \mathbf{A_1}, \mathbf{A_2}, \mathbf{B_1}, \mathbf{B_2}, \mathbf{G}, \mathbf{y}, \mathfrak{F}, H, H_1, H_2, H_N\}$, $msk \triangleq \{\mathbf{T_{A_0}}\}$.
– Select an appropriate Gaussian parameter s. (Its value will be determined later.)

- **Extract**(msk, ID): Suppose that the identity ID belongs to $\mathbb{Z}_q^n$. If this is not true, anyone can use a universal hash function to map the identity ID to $\mathbb{Z}_q^n$. PKG generates secret key SK_{ID} for ID as follows:

(1) Run $\mathbf{T'_{ID}} \leftarrow \mathbf{SampleBasisLeft}(\mathbf{A_0}, [\mathbf{A_1} + H(ID)\mathbf{B_1}], \mathbf{T_{A_0}}, s) \in \mathbb{Z}^{m\times m}$.
(2) Let $SK_{ID} = \mathbf{T'_{ID}}$. $\mathbf{T'_{ID}}$ is a trapdoor of $\mathbf{\Lambda}^{\perp}\left(\mathbf{A'_{ID}}\right)$ according to Proposition 8, where $\mathbf{A'_{ID}} = [\mathbf{A_0}\|\mathbf{A_1} + H(ID)\mathbf{B_1}]$.
(3) Send SK_{ID} to ID.

- **Signcrypt** $\left(u, \mathbf{T'_{ID_s}}, ID_r\right)$: The sender ID_s signs a message $u \in \{0,1\}^*$ with his or her own private key $\mathbf{T'_{ID_s}}$, then encrypts the relevant information under the receiver's identity ID_r as follows:

(1) Sign message u to obtain (σ, r_1) as follows:

(a) Choose $r_1 \xleftarrow{\$} \{0,1\}^{\gamma}$, compute $\upsilon = H_1(u, r_1, ID_r)$;
(b) Compute $\mathbf{F_{sr}} = \sum_{i=1}^{\kappa}(-1)^{\upsilon[i]}\mathbf{F_i}$, where $\upsilon[i]$ is the ith bit of υ;
(c) Sample $\mathbf{w_2} \leftarrow D_{\mathbb{Z}^m, s_1, 0}$, where s_1 is the Gaussian parameter;
(d) Compute $\mathbf{z} = \mathbf{F_{sr}}\mathbf{w_2}$;
(e) Sample $\mathbf{w_1} = \mathbf{SamplePre}\left(\mathbf{T'_{ID_s}}, \mathbf{y} - \mathbf{z}, s_1\right)$, set $\mathbf{w} = (\mathbf{w_1}, \mathbf{w_2})$, where $\mathbf{w_1} \in \mathbb{Z}^{2m}$, $\mathbf{w_2} \in \mathbb{Z}^m$.

(2) Transform $\mathbf{w}$ into a bit string (σ_1, σ_2), where $\sigma_1 \in \{0,1\}^{\iota}$ and σ_2 is the remainder string.
(3) Encrypt the σ_1 using the receiver's identity:

(a) Choose $r_2 \xleftarrow{\$} D_{\mathbb{Z}^{m'}, \tilde{s}}$ and compute $\mathbf{t} = H_N(\sigma_1, r_2)$, if $\mathbf{t} = \mathbf{0}$, repeat this step;
(b) Construct the matrix used for encryption $\mathbf{A_{ID_r}} = \left[\mathbf{A'_{ID_r}}\|\mathbf{A_2} + H(\mathbf{t})\mathbf{B_2}\right]$, where $\mathbf{A'_{ID_r}} = [\mathbf{A_0}\|\mathbf{A_1} + H(ID_r)\mathbf{B_1}]$ is same as in the **Extract** step;
(c) Choose a vector $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n\backslash\{\mathbf{0}\}$;
(d) Choose two matrices $\mathbf{R} \xleftarrow{\$} \{-1,1\}^{m\times m}$ and $\mathbf{Q} \xleftarrow{\$} \{-1,1\}^{m\times m}$;
(e) Compute $\mathbf{c_0} = \lfloor\mathbf{G}^T\mathbf{s}\rfloor_p + \sigma_1\lfloor p/2\rfloor$;
(f) Compute $\mathbf{c_1} = \lfloor\mathbf{A}_{\mathbf{ID_r}}^T\mathbf{s}\rfloor_p - \begin{bmatrix}\mathbf{0}\\ \lfloor\mathbf{R}^T\mathbf{e_0}\rfloor\\ \lfloor\mathbf{Q}^T\mathbf{e_0}\rfloor\end{bmatrix}$, where $\mathbf{e_0} = (p/q)\mathbf{A_0}^T\mathbf{s} - \lfloor\mathbf{A_0}^T\mathbf{s}\rfloor_p$

(4) Encrypt u, σ_2 as follows: $c_2 = H_2(\sigma_1, \mathbf{c_0}, \mathbf{c_1}) \oplus (\sigma_2\|r_1\|r_2\|u)$. Output ciphertext $\mathbf{c} = (\mathbf{t}, \mathbf{c_0}, \mathbf{c_1}, c_2)$. Note that when the total length of $(\sigma_2\|r_1\|r_2\|u)$ is longer than ϱ, namely, the length of the hash value of H_2, we can replace the H_2 with a pseudo-random number generator (PRG), meanwhile the hash value of H_2 is used as the seed for this PRG.

- **Unsigncrypt**$(\mathbf{c}, \mathbf{T_{ID_r}}, ID_s)$: The receiver decrypts $\mathbf{c}$ with his or her own secret key $\mathbf{T_{ID_r}}$ and verifies authenticity with ID_r, ID_s as follows:

(1) Parse $\mathbf{c}$ as $\left(\mathbf{t'}, \mathbf{c'_0}, \mathbf{c'_1}, c'_2\right)$. If $\mathbf{t'} = 0$, output $\perp$; otherwise, decrypt $\left(\mathbf{c'_0}, \mathbf{c'_1}\right)$ to achieve σ'_1 as follows:

(a) Sample $\mathbf{x_i} \leftarrow D_{\mathbb{Z}^m, s}$, then compute $\mathbf{x'_i} = [\mathbf{A_2} + H(\mathbf{t})\mathbf{B_2}]\mathbf{x_i}$ for $i \in [\ell]$.
(b) Let $\mathbf{E_{ID_r}} = \begin{bmatrix}\mathbf{d_1} & \cdots & \mathbf{d}_\ell\\ \mathbf{x_1} & \cdots & \mathbf{x}_\ell\end{bmatrix}$, where $\mathbf{d_i} = \mathbf{SamplePre}\left(\mathbf{T'_{ID_r}}, \mathbf{A'_{ID_r}}, \mathbf{g_i} - \mathbf{x'_i}, s\right)$ for $i \in [\ell]$, $\mathbf{A'_{ID_r}} = [\mathbf{A_0}\|\mathbf{A_1} + H(ID)\mathbf{B_1}]$ and $\mathbf{g_i}$ denotes the ith column of $\mathbf{G}$.
(c) Compute $\mathbf{b} = \mathbf{c_0} - \mathbf{E}_{\mathbf{ID_r}}^T\mathbf{c_1}$.
(d) For $j \in [\ell]$, if $\|\mathbf{b}[j] - \lfloor p/2\rfloor\| < \lfloor p/4\rfloor\|$, then $\sigma'_1[j] = 1$; otherwise, $\sigma'_1[j] = 0$. Here, $\mathbf{b}[j]$ represents the jth element of string $\mathbf{b}$.

(2) Let $\sigma'_2\|r'_1\|r'_2\|u' = H_2\left(\sigma'_1, \mathbf{c'_0}, \mathbf{c'_1}\right) \oplus c_2$.
(3) Verify the sender's authenticity as follows:

(a) Compose a bit string $\sigma' = \left(\sigma'_1, \sigma'_2\right)$ and transform it into $\mathbf{w} \in \mathbb{Z}^{3m}$;
(b) If $\|\mathbf{w}\| > s_1\sqrt{3m}$, output $\perp$;
(c) Compute $\upsilon' = H_1\left(u', r'_1, ID_r\right)$;
(d) Compute $\tilde{\mathbf{F}}_{\mathbf{sr}} = \sum_{i=1}^{\kappa}(-1)^{\upsilon'[i]}\mathbf{F_i}$;
(e) If $[\mathbf{A_0}\|\mathbf{A_1} + H(ID_s)\mathbf{B_1}\|\tilde{\mathbf{F}}_{\mathbf{sr}}]\mathbf{w} \neq \mathbf{y}$ output $\perp$; else output u'.

4.2. Consistency and parameter settings

Lemma 1. *The size of every entry of the error vector in the decryption is lower than $O(m^2)$ with high probability*

Proof. In the decryption step,

$$\begin{aligned} w &= \mathbf{c_0} - \mathbf{E}_{\mathbf{ID_r}}^T \mathbf{c_1} \\ &= (\lfloor \mathbf{G}^T \mathbf{s} \rceil_p + \sigma_1 \lfloor p/2 \rfloor) - \mathbf{E}_{\mathbf{ID_r}}^T \mathbf{c_1} \\ &= (\lfloor \mathbf{G}^T \mathbf{s} \rceil_p + \sigma_1 \lfloor p/2 \rfloor) - \mathbf{E}_{\mathbf{ID_r}}^T \left((p/q)(\mathbf{A}_{\mathbf{ID_r}}^T \mathbf{s}) - \mathbf{e_1} - \begin{bmatrix} \mathbf{0} \\ \lfloor \mathbf{R}^T \mathbf{e_0} \rfloor \\ \lfloor \mathbf{Q}^T \mathbf{e_0} \rfloor \end{bmatrix} \right) \\ &= \left(\lfloor \mathbf{G}^T \mathbf{s} \rceil_p + \sigma_1 \lfloor p/2 \rfloor \right) - \left(\lfloor \mathbf{G}^T \mathbf{s} \rceil_p + \mathbf{e_2} - \mathbf{E}_{\mathbf{ID_r}}^T \mathbf{e_1} - \mathbf{E}_{\mathbf{ID_r}}^T \begin{bmatrix} \mathbf{0} \\ \lfloor \mathbf{R}^T \mathbf{e_0} \rfloor \\ \lfloor \mathbf{Q}^T \mathbf{e_0} \rfloor \end{bmatrix} \right) \\ &= \sigma_1 \lfloor p/2 \rfloor + \left(\mathbf{E}_{\mathbf{ID_r}}^T \mathbf{e_1} + \mathbf{E}_{\mathbf{ID_r}}^T \begin{bmatrix} \mathbf{0} \\ \lfloor \mathbf{R}^T \mathbf{e_0} \rfloor \\ \lfloor \mathbf{Q}^T \mathbf{e_0} \rfloor \end{bmatrix} - \mathbf{e_2} \right) \end{aligned} \tag{6}$$

where

$$\mathbf{e_0} = (p/q)\mathbf{A}_{\mathbf{0}}^T \mathbf{s} - \lfloor \mathbf{A}_{\mathbf{0}}^T \mathbf{s} \rceil_p \in [0,1)^m \tag{7}$$

$$\mathbf{e_1} = (p/q)\left(\mathbf{A}_{\mathbf{ID_r}}^T \mathbf{s}\right) - \lfloor \mathbf{A}_{\mathbf{ID_r}}^T \mathbf{s} \rceil_p \in [0,1)^{3m} \tag{8}$$

$$\mathbf{e_2} = (p/q)\left(\mathbf{E}_{\mathbf{ID_r}}^T \mathbf{A}_{\mathbf{ID_r}}^T \mathbf{s}\right) - \lfloor \mathbf{E}_{\mathbf{ID_r}}^T \mathbf{A}_{\mathbf{ID_r}}^T \mathbf{s} \rfloor_p \in [0,1)^{3m} \tag{9}$$

The error vector is $\mathbf{E}_{\mathbf{ID_r}}^T \mathbf{e_1} + \mathbf{E}_{\mathbf{ID_r}}^T \begin{bmatrix} \mathbf{0} \\ \lfloor \mathbf{R}^T \mathbf{e_0} \rfloor \\ \lfloor \mathbf{Q}^T \mathbf{e_0} \rfloor \end{bmatrix} - \mathbf{e_2}$. Every entry of the error vector is $e = \mathbf{e}_{\mathbf{id}}^{\mathbf{T}} \mathbf{e_1} + \mathbf{e}_{\mathbf{id}}^{\mathbf{T}} \begin{bmatrix} \mathbf{0} \\ \lfloor \mathbf{R}^T \mathbf{e_0} \rfloor \\ \lfloor \mathbf{Q}^T \mathbf{e_0} \rfloor \end{bmatrix} + e_2'$, where $\mathbf{e_{id}}$ denotes a column of $\mathbf{E}_{\mathbf{ID_r}}$ and e_2' is the corresponding element of $\mathbf{e_2}$.

$$\begin{aligned} \|e\| &\le \left\| \mathbf{e}_{\mathbf{id}}^{\mathbf{T}} \mathbf{e_1} + \mathbf{e}_{\mathbf{id}}^{\mathbf{T}} \begin{bmatrix} \mathbf{0} \\ \lfloor \mathbf{R}^T \mathbf{e_0} \rfloor \\ \lfloor \mathbf{Q}^T \mathbf{e_0} \rfloor \end{bmatrix} \right\| \\ &\le \|\mathbf{e}_{\mathbf{id}}^{\mathbf{T}}\| \sqrt{3m} + \|\mathbf{e}_{\mathbf{id|1}}^{\mathbf{T}}\| \|\mathbf{R}^T\| \sqrt{m} + \|\mathbf{e}_{\mathbf{id|2}}^{\mathbf{T}}\| \|\mathbf{Q}^T\| \sqrt{m} \\ &\le s_1 \sqrt{3m} \sqrt{3m} + s_1 \sqrt{m} C \sqrt{m+m} \sqrt{m} \\ &\quad + s_1 \sqrt{m} C \sqrt{m+m} \sqrt{m} \le C' m^2 \end{aligned} \tag{10}$$

where C is a constant and $\mathbf{e_{id|i}}$ represents a section of $\mathbf{e_{id}}$ with the subscript from $i \cdot m$ to $(i+1) \cdot m$. This completes the proof. □

In order to guarantee the system to work well, the following requirements should be satisfied:

- The LWR problem must be hard, that is, $q \ge 2\beta\gamma nmp$ [28] and $\alpha q > 2\sqrt{n}$ [36] (or see Proposition 2).
- **TrapGen** algorithm should work well, that is, $m \ge 6n \log q$ [30].
- **SampleLeft** and **SampleRight** algorithms should work well. It needs s_1 to be big enough, that is, $s_1 > O(m)\sqrt{\log m}$, according to Propositions 5 and 7.
- The error should be small enough, that is, $C' m^2 \le \lfloor p/4 \rfloor$.

Let n be the security parameter. In addition, for convenience, we set δ is a real number such that $n^\delta > \lceil \log q \rceil$. To satisfy all these constraints, the parameters can be set as follows:

$$\begin{aligned} &m = 6n^{1+\delta}, \quad s_1 = m \cdot \omega(\sqrt{\log n}), \quad p = O(m^2), \\ &\alpha = (m^2 \omega(\log n))^{-1}, \\ &q \ge 2\beta\gamma nmp \ge 4\alpha q \omega(\sqrt{\log n}) nmp \approx \sqrt{n}\omega\left(\sqrt{\log n}\right) \\ &\qquad nmO(m^2) \approx m^{3+1.5(1+\delta)^{-1}} \omega(\sqrt{\log n}) \end{aligned} \tag{11}$$

4.3. Security

In order to answer the signcryption/unsigncryption queries with lower cost and shorter running time, an algorithm SampleRightv used to sample pre-image is presented before the security proof. Given a matrix $\mathbf{A} = [\mathbf{A_1} \| \mathbf{A_1 R} + \mathbf{HB} \| \mathbf{C}] \in \mathbb{Z}_q^{n \times (2m+\kappa)}$ and a vector $\mathbf{y} \in \mathbb{Z}_q^n$, where only the trapdoor $\mathbf{T_B}$ of $\mathbf{\Lambda}^\perp(\mathbf{B})$ is known and $\mathbf{H}$ is invertible, it is used frequently in the security proof to sample a pre-image of $\mathbf{y}$. In order to efficiently simulate the sampling, we give the algorithm **SampleRightv** in Algorithm 1.

Algorithm 1 SampleRightv($\mathbf{A_1}, \mathbf{B}, \mathbf{H}, \mathbf{R}, \mathbf{C}, \mathbf{T_B}, \mathbf{y}, s$)

Require:

- Matrices $\mathbf{A_1} \in \mathbb{Z}_q^{n \times m}, \mathbf{B} \in \mathbb{Z}_q^{n \times m}, \mathbf{C} \in \mathbb{Z}_q^{n \times k}$, $\mathbf{R} \in \{-1, 1\}^{m \times m}$, invertible $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$;
- A vector $\mathbf{y} \in \mathbb{Z}_q^n$;
- Gaussian parameter $s > \|\widetilde{\mathbf{T_B}}\| \cdot \omega(\sqrt{\log m})$.

Ensure:

A vector $\mathbf{v}$ such that $\|\mathbf{v}\|$ is small enough and $[\mathbf{A_1} \| \mathbf{A_1 R} + \mathbf{HB} \| \mathbf{C}]\mathbf{v} = \mathbf{y}$.

1: sample $\mathbf{x} \leftarrow D_{\mathbb{Z}^m, s_1}$;
2: compute $\mathbf{x}' = \mathbf{Cx} \bmod q$;
3: compute $\mathbf{y}' = \mathbf{y} - \mathbf{x}' \bmod q$;
4: compute $\mathbf{y}'' = \mathbf{H}^{-1}\mathbf{y}' \bmod q$;
5: $\mathbf{z} = \mathbf{SamplePre}(\mathbf{T_B}, \mathbf{B}, \mathbf{y}'', s_2)$;
6: Output $\mathbf{v} = [-\mathbf{Rz}, \mathbf{z}, \mathbf{x}]^T$.

Firstly, it is obvious that **SampleRightv** needs to run **SamplePre** only one time.

Secondly, the vector $\mathbf{v}$ is indeed a pre-image of $\mathbf{y}$:

$$\begin{aligned} &\mathbf{A}[-\mathbf{Rz}, \mathbf{z}, \mathbf{x}]^T = [\mathbf{A_1} \| \mathbf{A_1 R} + \mathbf{HB} \| \mathbf{C}][-\mathbf{Rz}, \mathbf{z}, \mathbf{x}]^T \\ &= \mathbf{A_1}(-\mathbf{Rz}) + (\mathbf{A_1 R} + \mathbf{HB})\mathbf{z} + \mathbf{Cx} = \mathbf{HBz} + \mathbf{Cx} \\ &= \mathbf{HH}^{-1}\mathbf{y}' + \mathbf{x}' = \mathbf{y}' + \mathbf{x}' = \mathbf{y} \end{aligned} \tag{12}$$

Thirdly, the vector $\mathbf{v}$ is short. The Gaussian parameters used to sample $\mathbf{x}$ and $\mathbf{z}$ in **SampleRightv** are $s_1 = \eta_\epsilon(\mathbb{Z})$ and $s_2 = \|\widetilde{\mathbf{T_B}}\| \cdot \omega(\sqrt{\log m})$, respectively. Therefore,

according to item 2 of Proposition 5,

$$\|\mathbf{v}\| \leq \sqrt{[\|\widetilde{\mathbf{T}_\mathbf{B}}\|^2 \cdot \omega(\log m)\left(s_R^2+1\right)+\eta_\epsilon(\mathbb{Z})^2]m} \quad (13)$$

Theorem 1 (Confidentiality)**.** *In the standard model, the proposed signcryption scheme under the parameters in (11) is indistinguishable against inner selective identity and chosen ciphertext attacks (IND-sID-CCA2) assuming the intractability of the decision-LWR$_{n,m,q,p}$ problem.*

Proof

Let us define a series of games between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$ firstly.

- Game G_0: This is the original IND-sID-CCA2 game defined in Section 3. $\mathcal{C}$ knows the trapdoor $\mathbf{T}_{\mathbf{A_0}}$ for $\mathbf{\Lambda}^{\perp}(\mathbf{A_0})$, so it can reply all legal queries.
- Game G_1: $\mathcal{C}$ slightly changes the way in which $\mathbf{A_1}$ and $\mathbf{A_2}$ are generated. More precisely, $\mathcal{C}$ randomly chooses $\mathbf{R}^* \in \{-1,1\}^{m\times m}$ in initial phase, then computes $\mathbf{A_1} = \mathbf{A_0}\mathbf{R}^* - H\left(ID_r^*\right)\mathbf{B_1}$, where ID_r^* is the challenge identity. $\mathcal{C}$ randomly chooses $\mathbf{Q}^* \in \{-1,1\}^{m\times m}$ and $\mathbf{t}^* \in \mathbb{Z}_q^n$ in initial phase, followed by computing $\mathbf{A_2} = \mathbf{A_0}\mathbf{Q}^* - H(\mathbf{t}^*)\mathbf{B_2}$. As in Game G_0, $\mathcal{C}$ can reply all legal queries (Lemma 2).
- Game G_2: $\mathcal{C}$ replaces the hash function H_N with a chameleon hash function H_C with the same form but keeps the trapdoor of H_C to itself. As in Game G_1, $\mathcal{C}$ can reply all legal queries.
- Game G_3: $\mathcal{C}$ changes the way how to produce $\mathbf{A_0}, \mathbf{B_1}$ and $\mathbf{B_2}$. $\mathcal{C}$ chooses $\mathbf{A_0} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ and generates $\mathbf{B_1}$ and $\mathbf{B_2}$ by running **TrapGen**(n,m,q) algorithm. The construction of $\mathbf{A_1}$ and $\mathbf{A_2}$ is the same as in Game $\mathbf{G_2}$. $\mathcal{C}$ can reply all legal queries except with a negligible probability, that is, in the case that the receiver is ID_r^* and the tag is $\mathbf{t}^*$, simultaneously (Lemma 4).
- Game G_4: $\mathcal{C}$ changes the way to generate the challenge ciphertext $\mathbf{c}^*$. Recall in Game G_3, the vector $\mathbf{r_2}$ is chosen from $D_{\mathbb{Z}^{m'},s_1}$ and satisfies $\|\mathbf{r_2}\| \leq s_1\sqrt{m'}$. Now, $\mathcal{C}$ can choose $\mathbf{r_2}$ to satisfy $H_C(\sigma_1,\mathbf{r_2}) = \mathbf{t}^*$ and $\|\mathbf{r_2}\| \leq s_1\sqrt{m'}$ by utilizing the trapdoor of the chameleon hash function H_C, where σ_1 is a part of the signature value of the message u (see Signcrypt algorithm). As in Game G_3, $\mathcal{C}$ can reply all the legal queries that $\mathcal{C}$ of Game G_4 can.
- Game G_5: $\mathcal{C}$ continues changing the way to generate the challenge ciphertext $\mathbf{c}^*$. $\mathcal{C}$ chooses $\left(\mathbf{c}_\mathbf{0}^*, \mathbf{c}_\mathbf{1}^*\right) \xleftarrow{\$} \mathbb{Z}_p^n \times \mathbb{Z}_p^n$. $\mathcal{C}$ can reply all queries that challenger $\mathcal{C}$ of Game G_4 can reply.
- Game G_6: $\mathcal{C}$ replaces c_2 with an isometric random string c_2'. Because the challenge ciphertext is always fresh random, $\mathcal{A}$'s advantage in winning the IND-sID-CCA2 game is negligible.

The indistinguishability between two sequential games G_i and G_{i+1} for i from 0 to 5 will be proved in the following lemmas. In the last game (Game G_6), the adversary's advantage is negligible. This completes the proof for the theorem. □

Lemma 2. *The adversary's views in Games G_1 and G_0 are statistically indistinguishable and $\mathcal{C}$ can reply all valid queries.*

Proof. In both Games G_1 and G_0,

$$\mathbf{A_{ID_r}} = [\mathbf{A_0}\|\mathbf{A_1} + H(ID)\mathbf{B_1}\|\mathbf{A_2} + H(\mathbf{t})\mathbf{B_2}]$$

In Game $G_0, \mathbf{A_0} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ and $\mathbf{A_1} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, respectively. However, in Game G_1,

$$\mathbf{A_1} = \mathbf{A_0}\mathbf{R}^* - H\left(ID_r^*\right)\mathbf{B_1}, \quad \mathbf{A_2} = \mathbf{A_0}\mathbf{Q}^* - H(\mathbf{t}^*)\mathbf{B_2}$$

where $\mathbf{R}^* \xleftarrow{\$} \{-1,1\}^{m\times m}$ and $\mathbf{Q}^* \xleftarrow{\$} \{-1,1\}^{m\times m}$, respectively. In the challenge ciphertext, $\lfloor \mathbf{R}^{*T}\mathbf{e_0} \rfloor$ (respectively, $\lfloor \mathbf{Q}^{*T}\mathbf{e_0} \rfloor$) leaks some information of $\mathbf{R}^*$ (respectively, $\mathbf{Q}^*$), where $\mathbf{e_0} \in [0,1)^m$. The information about $\mathbf{R}^*$(respectively, $\mathbf{Q}^*$) revealed by $\lfloor \mathbf{R}^{*T}\mathbf{e_0} \rfloor$ (respectively, $\lfloor \mathbf{Q}^{*T}\mathbf{e_0} \rfloor$) is no more than that by $\mathbf{R}^{*T}\lfloor q\mathbf{e_0} \rceil$(respectively, $\mathbf{Q}^{*T}\lfloor q\mathbf{e_0} \rceil$). According to Proposition 11, $(\mathbf{A_0}, \mathbf{A_0}\mathbf{R}^*, \mathbf{R}^{*T}\lfloor q\mathbf{e_0} \rceil)$ is within negligible distance from $(\mathbf{A_0}, \mathbf{A}_\mathbf{r}', \mathbf{R}^{*T}\lfloor q\mathbf{e_0} \rceil)$, where $\mathbf{A}_\mathbf{r}' \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$. Then, after ID^* and $\mathbf{B_1}$ is fixed, $(\mathbf{A_0}, \mathbf{A_0}\mathbf{R}^* - H(ID^*)\mathbf{B_1}, \mathbf{R}^{*T}\lfloor q\mathbf{e_0} \rceil)$ is statistically close from $(\mathbf{A_0}, \mathbf{A}'', \mathbf{R}^{*T}\lfloor q\mathbf{e_0} \rceil)$, where the distribution of $\mathbf{A}_\mathbf{r}'' \in \mathbb{Z}_q^{n\times m}$ is uniform. Similarly, $(\mathbf{A_0}, \mathbf{A_0}\mathbf{Q}^* - H(\mathbf{t}^*)\mathbf{B_2}, \mathbf{Q}^{*T}\lfloor q\mathbf{e_0} \rceil)$ is statistically close from $(\mathbf{A_0}, \mathbf{A}'', \mathbf{Q}^{*T}\lfloor q\mathbf{e_0} \rceil)$, where $\mathbf{A}'' \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$. Hence, in $\mathcal{A}$'s view, Game G_0 and Game G_1 are indistinguishable.

Let us analyze why $\lfloor \mathbf{R}^{*T}\mathbf{e_0} \rfloor$ reveals information about $\mathbf{R}^*$ no more than $\mathbf{R}^{*T}\lfloor q\mathbf{e_0} \rceil \bmod q$ does, for $\mathbf{e_0} \in [0,1)^m$. Let $\mathbf{a} \triangleq \mathbf{R}^{*T}\lfloor q\mathbf{e_0} \rceil \bmod q$ for some $\mathbf{R}^*$. When $\mathbf{R}^*$ goes through its domain $\{-1,1\}^{m\times m}$, the probability that $\mathbf{R}^{*T}\lfloor q\mathbf{e_0} \rceil \bmod q = \mathbf{a}$ is roughly q^{-m}. That is to say, for a fixed $\mathbf{a}$, the number of the $\mathbf{R}^*$ satisfying $\mathbf{R}^{*T}\lfloor q\mathbf{e_0} \rceil \bmod q = \mathbf{a}$ is roughly $q^{-m}2^{m^2}$. Let $\mathbf{a}' = \lfloor \mathbf{R}^{*T}\mathbf{e_0} \rfloor$ for some $\mathbf{R}^*$. Its entropy is $\vartheta = -\log_2(q^{-m}2^{m^2})^{-1} = m^2 - m\log_2 q$. When $\mathbf{R}^*$ goes through its domain, $\lfloor \mathbf{R}^{*T}\mathbf{e_0} \rfloor$ runs at most $\{-m, -m+1, \ldots, m\}^m$, whose size is $(2m)^m$. The probability that $\lfloor \mathbf{R}^{*T}\mathbf{e_0} \rfloor = \mathbf{a}'$ is roughly $(2m)^{-m}$. Namely, for a fixed $\mathbf{a}'$, the number of the corresponding $\mathbf{R}^*$ is roughly $(2m)^{-m}2^{m^2}$. Its entropy is $\vartheta' = -\log_2((2m)^{-m}2^{m^2})^{-1} = m^2 - m\log_2 m$. And because $q > m, \vartheta' > \vartheta$, $\lfloor \mathbf{R}^{*}\mathbf{e_0} \rfloor$ reveals information no more than $\mathbf{R}^{*T}\lfloor q\mathbf{e_0} \rceil \bmod q$ does.

In Game G_1, $\mathbf{A'_{ID_r}} = [\mathbf{A_0}\|\mathbf{A_1} + H(ID)\mathbf{B_1}] = \left[\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^* + \left(H(ID) - H\left(ID_r^*\right)\right)\mathbf{B_1}\right]$, and $\mathcal{C}$ still keeps the master key $\mathbf{T_{A_0}}$. Hence, $\mathcal{C}$ can use **SampleBasisLeft** to extract a private key for the identity ID. Therefore, $\mathcal{C}$ can answer all queries. □

Lemma 3. *The adversary's views in Games G_2 and G_1 are statistically indistinguishable.*

Proof. On one hand, the distributions of the inputs of the hash functions H_N are identical to that of H_C. On the other hand, according to Proposition 4, the distributions of matrices used for constructing H_N and H_C are also the same. As a result, the ranges of the hash functions H_N and H_C are identical. The only difference is that $\mathcal{C}$ of Game G_2 knows the trapdoor of the hash function (but does not use it); however, $\mathcal{C}$ of Game G_1 does not know the trapdoor. Consequently, the adversary's views in the two games are indistinguishable. □

Lemma 4. *The adversary's views in Games G_3 and G_2 are statistically indistinguishable. Moreover, $\mathcal{C}$ can reply all legal queries from $\mathcal{A}$ w.o.p. except $ID_r = ID_r^*$ and $\mathbf{t} = \mathbf{t}^*$, simultaneously. In addition, the probability that $\mathbf{c} = (\mathbf{t}^*, \cdot, \cdot, \cdot) \neq \left(\mathbf{t}^*, \mathbf{c_0^*}, \mathbf{c_1^*}, c_2\right)$ and $\mathbf{c}$ is a valid ciphertext is negligible.*

Proof. Firstly, let us prove that Games G_3 and G_2 are statistically indistinguishable in the adversary's views. In both Games G_2 and G_3, the public key matrices are both

$$\begin{aligned}\mathbf{A_{ID_r}} &= [\mathbf{A_0}\|\mathbf{A_1} + H(ID_r)\mathbf{B_1}\|\mathbf{A_2} + H(\mathbf{t})\mathbf{B_2}] \\ &= \left[\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^* + \left(H(ID_r) - H\left(ID_r^*\right)\right)\mathbf{B_1}\|\mathbf{A_0}\mathbf{Q}^*\right. \\ &\quad \left. +(H(\mathbf{t}) - H(\mathbf{t}^*))\mathbf{B_2}\right] \qquad (14)\end{aligned}$$

In Game G_2, the matrix $\mathbf{A_0} \leftarrow \mathbf{TrapGen}(n, m, q)$; however, in Game G_3, $\mathbf{A_0} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$. According to Proposition 4, the two matrices are statistically indistinguishable. Similarly, the matrices $\mathbf{B_1}$ (respectively, $\mathbf{B_2}$) in Game G_2 and $\mathbf{B_1}$ (respectively, $\mathbf{B_2}$) in G_3 are statistically indistinguishable. Hence, the public key matrices in the two games are indistinguishable, after addition and concatenation.

Secondly, let us prove the correctness of the second part of this lemma.

- $\mathcal{A}$ issues private key extraction queries with identity $ID_i \neq ID_r^*$. Because the matrix corresponding to the identity ID_i is $\mathbf{A'_{ID_i}} = [\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^* + (H(ID_i) - H(ID_r^*))\mathbf{B_1}]$, $\mathcal{C}$ runs **SampleBasisRight** $(\mathbf{A_0}, \mathbf{B_1}, H(ID_i) - H(ID_r^*), \mathbf{R}^*, \mathbf{T_{B_1}}, s)$ to obtain a basis $\mathbf{T'_{ID_i}}$ as the secret key of ID_i.
- When $\mathcal{A}$ issues unsigncrypt queries, $\mathcal{C}$ can answer as follows:
  - Case 1: $\mathcal{A}$ asks unsigncrypt queries with $(\mathbf{c}, ID_r, ID_s)$ and $ID_r \neq ID_r^*$. Note that the matrix corresponding to ID_r is

    $$\mathbf{A'_{ID_r}} = \left[\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^* + (H(ID_r) - H(ID_r^*))\mathbf{B_1}\right]$$

    and the matrix used to encrypt for identity ID_r is

    $$\mathbf{A_{ID_r}} = \left[\mathbf{A'_{ID_r}}\|\mathbf{A_0}\mathbf{Q}^* + (H(\mathbf{t}) - H(\mathbf{t}^*))\mathbf{B_2}\right]$$

    Because $\mathcal{C}$ does not know the trapdoor of $\mathbf{\Lambda}^{\perp}(\mathbf{A_0})$, step 1(a) in Unsigncrypt algorithm should be changed slightly. Because $\mathcal{C}$ knows the trapdoor $\mathbf{T_{B_1}}$ of $\Lambda^{\perp}(\mathbf{B_1})$, $\mathcal{C}$ runs

    $$\mathbf{s_i} = \mathbf{SampleRightv}(\mathbf{A_0}, \mathbf{B_1}, H(ID_r) - H(ID_r^*), \mathbf{R}^*, (H(\mathbf{t}) - H(\mathbf{t}^*))\mathbf{B_2}, \mathbf{T_{B_1}}, \mathbf{g_i}, s)$$

    for $i \in [\ell]$ where $\mathbf{g_i}$ is the ith column of the matrix $\mathbf{G}$. $\mathcal{C}$ sets the ith column of $\mathbf{E_{ID_r}}$ to be $\mathbf{s_i}$. Now, we have that $\mathbf{A_{ID_r}}\mathbf{E_{ID_r}} = \mathbf{G}$. Each column vector of $\mathbf{E_{ID_r}}$ is shorter than the vector obtained by ABB-mid-SamplePre method in the statistical sense (Remark 3). On the other hand, the matrix obtained by ABB-mid-SamplePre method can be used for decryption. Therefore, the matrix $\mathbf{E_{ID_r}}$ can be used to decrypt. Next, the other steps in Unsigncrypt algorithm are executed normally. Finally, if the signature can pass the verification, then $\mathcal{C}$ returns the message obtained in the procedure; otherwise, $\mathcal{C}$ returns $\perp$.
  - Case 2: $\mathcal{A}$ asks unsigncrypt queries with $\left(\mathbf{c}, ID_r^*, ID_s\right)$. There are two cases according to whether $\mathbf{t} = \mathbf{t}^*$ or not.

    If $\mathbf{t} = \mathbf{t}^*$, $\mathcal{C}$ aborts. The probability in a valid ciphertext that the receiver is ID_r^* and the tag simultaneously is $\mathbf{t}^*$ is negligible. Firstly, in Phase 1, $\mathbf{t}^*$ is hidden from $\mathcal{A}$. Secondly, in Phase 2, because H_C is a chameleon hash function, if $\mathcal{A}$ can find (σ', r') satisfying $H_C(\sigma', r') = \mathbf{t}^* = H_C(\sigma, r_2)$, we can construct an algorithm $\mathcal{B}$ to solve $SIS_{q,\beta}$ problem according to Proposition 9. According to Proposition 3, the probability for this event is negligible.

    If $\mathbf{t} \neq \mathbf{t}^*$, $\mathcal{C}$ can answer the unsigncrypt queries with the similar method as the earlier ones when $ID_r \neq ID_r^*$ but can use the trapdoor of $\mathbf{\Lambda}^{\perp}(\mathbf{B_2})$ to sample.
- When $\mathcal{A}$ asks the signcrypt queries, $\mathcal{C}$ does as follows:
  - Case 1: $\mathcal{A}$ asks signcrypt queries with $(\mathbf{u}, ID_s, ID_r)$, where the sender's identity

$ID_s \neq ID_r^*$. Note that the matrix used for signature for the identity ID_s is

$$\begin{aligned}\mathbf{A_{ID_s}} &= \left[\mathbf{A'_{ID_s}} \| \mathbf{F_{sr}}\right] \\ &= \left[\mathbf{A_0} \| \mathbf{A_0 R}^* + (H(ID_s) \right. \\ &\quad \left. - H(ID_r^*))\mathbf{B_1} \| \mathbf{F_{sr}}\right]\end{aligned}$$

Note that $\mathcal{C}$ does not directly know the trapdoor $\mathbf{T'_{ID_s}}$ for $\mathbf{\Lambda}^{\perp}\left(\mathbf{A'_{ID_s}}\right)$. Before step 1(e) in Signcrypt, $\mathcal{C}$ runs

$$\mathbf{SampleBasisRight}(\mathbf{A_0}, \mathbf{B_1}, H(ID_s) - H(ID_r^*), \mathbf{R}^*, \mathbf{T_{B_1}}, s)$$

to obtain a basis $\mathbf{T'_{ID_s}}$ as the secret key of ID_s. Then, the other steps in **Signcrypt** algorithm are executed normally to generate a ciphertext as the answer.

– Case 2: $\mathcal{A}$ asks signcrypt queries with the sender's identity $ID_s = ID_r^*$. In order to reply the signcrypt queries, a sub-game G'_5 is needed here. In sub-game G'_5, $\mathcal{C}$ chooses $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,\sigma,0}$ at random and computes $\mathbf{y}' = \mathbf{A_0 x}$. $\mathcal{C}$ replaces the random vector $\mathbf{y}$ with $\mathbf{y}'$ and retains the other matrices in mpk unchanged, namely, $mpk \triangleq \{\mathbf{A_0}, \mathbf{A_1}, \mathbf{A_2}, \mathbf{B_1}, \mathbf{B_2}, \mathbf{G}, \mathbf{y}', \mathfrak{F}, H, H_1, H_2, H_N\}$. Now $\mathcal{C}$ can answer the signcrypt queries with the similar method as the signcrypt queries in the case of $ID_s = ID_s^*$ in Theorem 2. □

Lemma 5. *The adversary's views in Games G_4 and G_3 are statistically indistinguishable.*

Proof. Let us analyze the difference between Game G_4 and Game G_3, namely the challenge ciphertext. In Game G_3, $H_N(\sigma_1, r_2)$ is uniformly random according to the uniformity property of chameleon hashing(Lemma 4.1. in [32]). In Game G_4, $H_N(\sigma_1, r_2) = \mathbf{t}^*$, where $\mathbf{t}^* \overset{\$}{\leftarrow} \mathbb{Z}_q^n$ at the initial of the game, and $\mathbf{t}^*$ is hidden from $\mathcal{A}$ before producing challenge to the ciphertext. The adversary cannot distinguish $\mathbf{t}^*$ from $\mathbf{t} = H_N(\sigma_1, r_2)$ according to the uniformity property of H_N. This completes the proof. □

Lemma 6. *The adversary's views in Games G_5 and G_4 are computationally indistinguishable.*

Proof. This lemma can be proved by contradiction. Assuming that $\mathcal{A}$ has non-negligible advantage in distinguishing Games G_4 and G_5, we can construct an algorithm $\mathcal{C}$ to solve LWR with non-negligible probability.

Note that H_2 is a universal hash function and $c_2 = H_2(\sigma_1, \mathbf{c_0}, \mathbf{c_1}) \oplus (\sigma_2 \| r_1 \| r_2 \| u)$, then c_2 is always uniformly random, when the inputs $\mathbf{c_0}, \mathbf{c_1}$ of H_2 are uniformly random. As a result, $\mathcal{A}$ cannot distinguish c_2 from a uniformly random bit string with the same length. Therefore, the fact that $\mathcal{A}$ can distinguish Games G_4 and G_5 implies that $\mathcal{A}$ can distinguish $\left(\mathbf{c_0^*}, \mathbf{c_1^*}\right)$ is a valid LWR encryption for some value or not. Recall the definition of LWR in Proposition 2, the LWR instance is provided an oracle either pseudo-random $\mathcal{O}_s$ or truly random $\mathcal{O}_\$$. The algorithm $\mathcal{C}$ can use $\mathcal{A}$ to distinguish the two cases as follows:

- **Init.** $\mathcal{C}$ requests the oracle $\mathcal{O}$ and obtains some instances $(\mathbf{u_i}, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_p$ for $i \in [m+\ell]$. $\mathcal{C}$ publishes some relevant public parameters $\mathcal{P}_p$.
- **KeyGen.** $\mathcal{A}$ declares the target identity ID_r^* that he or she intends to attack. $\mathcal{C}$ forms the master public key as follows:
  - $\mathcal{C}$ composes the matrix $\mathbf{A_0}$ by setting the ith column $(\mathbf{A_0})_i = \mathbf{u_i}$ for $i \in [m]$. Similarly, $\mathcal{C}$ composes the matrix $\mathbf{G}$ by setting the ith column $\mathbf{G}_i = \mathbf{u_{i+m}}$ for $i \in [\ell]$.
  - $\mathcal{C}$ generates $\mathbf{B_1}$ and $\mathbf{B_2}$ by running $\mathbf{TrapGen}(n, m, q)$ algorithm as in Games G_4 and G_5 but does not reveal their trapdoors.
  - $\mathcal{C}$ chooses the other public matrix as in Games G_4 and G_5. Then, $\mathcal{C}$ publishes public key:

$$mpk \triangleq \{\mathbf{A_0}, \mathbf{A_1}, \mathbf{A_2}, \mathbf{B_1}, \mathbf{B_2}, \mathbf{G}, \mathbf{y}, \mathfrak{F}, H, H_1, H_2, H_N\}$$

- Phase 1. $\mathcal{C}$ replies the queries as in Games G_4 and G_5.
- Challenge. When $\mathcal{A}$ submits a message u, the sender's identity ID_s, and the receiver's identity $ID_r^* \neq ID_r \neq ID_s$, $\mathcal{C}$ produces a challenge ciphertext as follows:
  (1) Let $(\mathbf{u_i}, v_i)$ be the LWR instance in **Initial** for $i \in [m+\ell]$. Note that if $\mathcal{O} = \mathcal{O}_s$, $v_i = \lfloor \langle \mathbf{u_i}, s \rangle \rfloor_p$.
  (2) Set $\mathbf{v_2^*} = \begin{bmatrix} v_{m+1} \\ \vdots \\ v_{m+\ell} \end{bmatrix}$, $\mathbf{c_0^*} = \mathbf{v_2^*} + \sigma_1^* \lfloor q/2 \rfloor$.
  (3) Set $\mathbf{v_1^*} = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix}$, $(\mathbf{v_1^*})' = \mathbf{R}^{*T}\mathbf{v}^*$, $(\mathbf{v_1^*})'' = \mathbf{Q}^{*T}\mathbf{v}^*$, then let $\mathbf{c_1^*} = \begin{bmatrix} \mathbf{v_1^*} \\ \left(\mathbf{v_1^*}\right)' \\ \left(\mathbf{v_1^*}\right)'' \end{bmatrix}$.
  (4) Give $\left(\mathbf{c_0^*}, \mathbf{c_1^*}\right)$ to $\mathcal{A}$ as the challenge ciphertext.

It can be proved that the preceding challenge ciphertext is exactly the part $\left(\mathbf{c_0^*}, \mathbf{c_1^*}\right)$ of the challenge ciphertext $\left(\mathbf{t}^*, \mathbf{c_0^*}, \mathbf{c_1^*}, \mathbf{c_2^*}\right)$ of Game G_4, when the oracle $\mathcal{O} = \mathcal{O}_s$.

Firstly, let us observe the matrix used for encryption. At this moment, the matrix corresponding to the

receiver is

$$\mathbf{A}_{\mathbf{ID}_\mathbf{r}^{*\prime}} = \left[\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^* + (H(ID_r^*) - H(ID_r^*))\mathbf{B_1}\right] = [\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^*]$$

$\mathcal{C}$ can sample a random vector $\mathbf{r_2}$ to satisfy $H_C(\sigma_1, \mathbf{r_2}) = \mathbf{t}^*$ by making use of the trapdoor of H_C, where $\sigma_1 \in \{0,1\}^\kappa$ is a part of signature for some message (Signcrypt 4.1). As a result, the matrix used for encryption is

$$\mathbf{A}_{\mathbf{ID}_\mathbf{r}^*} = \left[\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^* + (H(ID_r^*) - H(ID_r^*))\mathbf{B_1}\|\mathbf{A_0}\mathbf{Q}^* + (H(\mathbf{t}^*) - H(\mathbf{t}^*))\mathbf{B_2}\right] = \left[\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^*\|\mathbf{A_0}\mathbf{Q}^*\right]$$

Secondly, let us check the challenge ciphertext.

– The first case is $\mathcal{O} = \mathcal{O}_s$. At first, v_i satisfies $v_i = \lfloor\langle\mathbf{u_i}, \mathbf{s}\rangle\rfloor_p$, for $i \in \{m+1, m+2, \cdots, m+\ell\}$. Because $\mathbf{G} = [\mathbf{u_{m+1}}, \mathbf{u_{m+2}}, \cdots \mathbf{u_{m+\ell}}]$,

$$\lfloor\mathbf{G}^T\mathbf{s}\rfloor_p + \sigma_1\lfloor p/2\rfloor = \begin{bmatrix} v_{m+1} \\ \vdots \\ v_{m+\ell} \end{bmatrix} + \sigma_1\lfloor p/2\rfloor = \mathbf{v}_2^* + \sigma_1\lfloor p/2\rfloor = \mathbf{c}_0^*$$

Consequently, $\mathbf{c}_0^*$ defined above is consistent with $\mathbf{c}_0^*$ in the challenge ciphertext of Game G_4. Secondly, v_i also satisfies $v_i = \lfloor\langle\mathbf{u_i}, s\rangle\rfloor_p$ for $i \in \{1, 2, \ldots, m\}$. Because

$$\mathbf{A_0} = [\mathbf{u_1}, \mathbf{u_2}, \cdots \mathbf{u_m}], \quad \mathbf{v}_1^* = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} = \lfloor\mathbf{A}_0^T\mathbf{s}\rfloor_p$$

the $\mathbf{c}_1^*$ in the challenge ciphertext is as follows:

$$\begin{aligned}
\mathbf{c}_1^* &= \begin{bmatrix} \mathbf{v}_1^* \\ (\mathbf{v}_1^*)' \\ (\mathbf{v}_1^*)'' \end{bmatrix} = \begin{bmatrix} \lfloor\mathbf{A}_0^T\mathbf{s}\rfloor_p \\ \mathbf{R}^{*T}\lfloor\mathbf{A}_0^T\mathbf{s}\rfloor_p \\ \mathbf{Q}^{*T}\lfloor\mathbf{A}_0^T\mathbf{s}\rfloor_p \end{bmatrix} \\
&= \begin{bmatrix} \lfloor\mathbf{A}_0^T\mathbf{s}\rfloor_p \\ \mathbf{R}^{*T}((p/q)\mathbf{A}_0^T\mathbf{s} - \mathbf{e_0}) \\ \mathbf{Q}^{*T}((p/q)\mathbf{A}_0^T\mathbf{s} - \mathbf{e_0}) \end{bmatrix} \\
&= \begin{bmatrix} \lfloor\mathbf{A}_0^T\mathbf{s}\rfloor_p \\ \lfloor(\mathbf{A_0}\mathbf{R}^*)^T\mathbf{s}\rfloor_p + \mathbf{e_1} - (\lfloor\mathbf{R}^{*T}\mathbf{e_0}\rfloor + \mathbf{e_2}) \\ \lfloor(\mathbf{A_0}\mathbf{Q}^{*T})^T\mathbf{s}\rfloor_p + \mathbf{e_3} - (\lfloor\mathbf{Q}^{*T}\mathbf{e_0}\rfloor + \mathbf{e_4}) \end{bmatrix} \\
&= \begin{bmatrix} \lfloor\mathbf{A}_0^T\mathbf{s}\rfloor_p \\ \lfloor(\mathbf{A_0}\mathbf{R}^*)^T\mathbf{s}\rfloor_p - \lfloor\mathbf{R}^{*T}\mathbf{e_0}\rfloor \\ \lfloor(\mathbf{A_0}\mathbf{Q}^{*T})^T\mathbf{s}\rfloor_p - \lfloor\mathbf{Q}^{*T}\mathbf{e_0}\rfloor \end{bmatrix} \\
&= \begin{bmatrix} \lfloor\mathbf{A}_0^T\mathbf{s}\rfloor_p \\ \lfloor(\mathbf{A_0}\mathbf{R}^*)^T\mathbf{s}\rfloor_p \\ \lfloor(\mathbf{A_0}\mathbf{Q}^{*T})^T\mathbf{s}\rfloor_p \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \lfloor\mathbf{R}^{*T}\mathbf{e_0}\rfloor \\ \lfloor\mathbf{Q}^{*T}\mathbf{e_0}\rfloor \end{bmatrix}
\end{aligned} \tag{15}$$

where $\mathbf{e_1} = (p/q)(\mathbf{A_0}\mathbf{R}^*)^T\mathbf{s} - \lfloor(\mathbf{A_0}\mathbf{R}^*)^T\mathbf{s}\rfloor_p$, $\mathbf{e_2} = \mathbf{R}^{*T}\mathbf{e_0} - \lfloor\mathbf{R}^{*T}\mathbf{e_0}\rfloor, \mathbf{e_3} = (p/q)((\mathbf{A_0}\mathbf{Q}^*)^T\mathbf{s} - \lfloor(\mathbf{A_0}\mathbf{Q}^*)^T\mathbf{s}\rfloor_p, \mathbf{e_4} = \mathbf{Q}^{*T}\mathbf{e_0} - \lfloor\mathbf{Q}^{*T}\mathbf{e_0}\rfloor$. It concludes that this is exactly the $\mathbf{c}_1^*$ of the challenge ciphertext in Game G_4.

– The second case is $\mathcal{O} = \mathcal{O}_\$$. In this case, $\mathbf{v}_2^*$ is uniform in $\mathbb{Z}_p^m$ hence $\mathbf{c}_0^*$ is also. In addition, $\mathbf{v}^*$ is uniformly random in $\mathbb{Z}_p^m$; $\mathbf{R}^*$ and $\mathbf{Q}^*$ are both fixed matrices. Hence, $\mathbf{c}_1^*$ is uniform in $\mathbb{Z}_p^{3m}$ according to leftover hash lemma (Theorem 8.38 in [37]).

- Phase 2. $\mathcal{A}$ makes queries and $\mathcal{C}$ replies accordingly as in Phase 1, but $\mathcal{A}$ cannot ask the unsigncrypt queries with the challenge ciphertext whose receiver's identity is ID_r^*.
- Guess. When $\mathcal{A}$ is satisfied, he or she stops querying and guesses that he or she is interacting with challenger $\mathcal{C}$ of Game G_4 or $\mathcal{C}$ of Game G_5. Challenger $\mathcal{C}$ outputs $\mathcal{A}$'s answer as the guess to the LWR oracle.

When $\mathcal{O} = \mathcal{O}_s$, the adversary $\mathcal{A}$'s view is the same as that in Game G_4. When $\mathcal{O} = \mathcal{O}_\$$, the adversary's view is the same as that in Game G_5. As a result, the simulator's advantage in solving the LWR problem is equal to the adversary $\mathcal{A}$'s advantage in distinguishing between Game G_4 and Game G_5. This completes the proof. □

Lemma 7. *The adversary's views in Games G_5 and G_6 are statistically indistinguishable.*

Proof. Recall that in Game $G_5, c_2 = H_2(\sigma_1, \mathbf{c_0}, \mathbf{c_1}) \oplus (\sigma_2\|r_1\|r_2\|u)$. On one hand, $\mathbf{c_0} \xleftarrow{\$} \mathbb{Z}_p^m$ and $\mathbf{c_1} \xleftarrow{\$} \mathbb{Z}_p^{3m}$, respectively. On the other hand, the hash function H_2 is universal. As a result, $H_2(\sigma_1, \mathbf{c_0}, \mathbf{c_1})$ is uniformly random. And because $\sigma_2, \mathbf{r_1}, \mathbf{r_2}, u$ are all fixed, c_2 is uniformly random. In Game G_6, c_2 is also uniformly random. Hence, Games G_5 and G_6 are statistically indistinguishable. Because in Game G_6 the challenge ciphertext is totally random and has no information about the message, $\mathcal{A}$'s advantage in G_6 is negligible. □

Theorem 2 (Unforgeability)**.** *In the standard model, the proposed signcryption scheme is SUF-sID-CMA assuming that $SIS_{q,\beta}$ is hard for large enough $\beta = sm\cdot(2C + \sqrt{2\kappa}s_1)$.*

Proof. What we consider is an inner adversary, which means that the adversary $\mathcal{F}$ has the private key for decryption and he or she can obtain the corresponding signature from a signcryption ciphertext. Then $\mathcal{F}$ forges a valid signcryption ciphertext, which means that $\mathcal{F}$ can forge a valid signature. Consequently, it only remains to prove the signature scheme is SUF-sID-CMA. This can be proved by contradiction. Assuming that there exists a probabilistic polynomial time adversary $\mathcal{F}$ who can forge a valid

signature, we can construct an algorithm $\mathcal{C}$ to solve $SIS_{q,m,\beta}$ with non-negligible probability.

- Init: $\mathcal{F}$ submits an identity ID_s^* that he or she intends to attack. $\mathcal{C}$ generates matrices $\mathbf{B_1}$,$\mathbf{B_2}$ $\in \mathbb{Z}_q^{n\times m}$ with the corresponding trapdoors $\mathbf{T_{B_1}}$,$\mathbf{T_{B_2}}$ $\in \mathbb{Z}_q^{m\times m}$ for $\Lambda^{\perp}(\mathbf{B_1})$,$\Lambda^{\perp}(\mathbf{B_2})$ by running $\mathbf{TrapGen}(n,m,q)$ algorithm. $\mathcal{C}$ chooses $\mathbf{A_0} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{A_2} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, and $\mathbf{G} \xleftarrow{\$} \mathbb{Z}_q^{n\times \ell}$. Then $\mathcal{C}$ chooses matrices $\mathbf{F_i'} \leftarrow D_{\mathbb{Z}^m,s}^{m\times m}$ for $i \in [k]$, followed by constructing $\mathbf{F_i} = \mathbf{A_0}\mathbf{F_i'}$ for $i \in [k]$. $\mathcal{C}$ chooses $\mathbf{R}^* \xleftarrow{\$} \{-1,1\}^{m\times m}$, followed by setting $\mathbf{A_1} = \mathbf{A_0}\mathbf{R}^* - H(ID_s^*)\mathbf{B_1}$. $\mathcal{C}$ chooses $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,s,0}$ at random and computes $\mathbf{y} = \mathbf{A_0}\mathbf{x}$. If $\mathbf{y} = \mathbf{0}$, this operation is repeated until $\mathbf{y} \neq \mathbf{0}$. Finally, $\mathcal{C}$ publishes $mpk \triangleq \{\mathbf{A_0}, \mathbf{A_1}, \mathbf{A_2}, \mathbf{B_1}, \mathbf{B_2}, \mathbf{G}, \mathbf{y}, \mathfrak{F}, H, H_1, H_2, H_N\}$ as public keys, where H, H_1, H_2, H_N are hash functions (same as in Setup 4.1).
- Queries: $\mathcal{A}$ can ask the key extraction, signcryption, and unsigncryption queries, then $\mathcal{C}$ answers as follows:
  - Key extraction queries (ID_i): $\mathcal{A}$ submits an identity $ID_i \neq ID_s^*$. $\mathcal{C}$ runs $\mathbf{T'_{ID_i}} \leftarrow \mathbf{SampleBasisRight}(\mathbf{A_0}, \mathbf{B_1}, H_{ID_i} - H_{ID_s^*}, \mathbf{R}^*, \mathbf{T_{B_1}}, s)$, then returns $\mathbf{T'_{ID_i}}$ as the answer.
  - Signcrypt queries (u, ID_s, ID_r): One case is that the sender's identity $ID_s \neq ID_s^*$. $\mathcal{C}$ can use the private key obtained by calling **SampleBasisRight** algorithm to sign, then encrypts the related information as aforementioned to obtain a ciphertext $\mathbf{c}$, and finally, returns $\mathbf{c}$ to $\mathcal{F}$.

    The other case is that the sender's identity $ID_s = ID_s^*$. $\mathcal{C}$ chooses $r \xleftarrow{\$} \{0,1\}^{\gamma}$, followed by evaluating hash value $\upsilon = H_1(u, r, ID) \in \{0,1\}^{\kappa}$. Next, $\mathcal{C}$ computes $\mathbf{F'_{sr}} = \sum_{i=1}^{\kappa}(-1)^{\upsilon[i]}\mathbf{F'_i}$, followed by choosing $\mathbf{w_2} \xleftarrow{\$} D_{\mathbb{Z}^{2m},s,0}$. Then $\mathcal{C}$ computes $\mathbf{y_1} = [\mathbf{R}^* \| \mathbf{F'_{sr}}]\mathbf{w_2}$ and $\mathbf{w_1} = \mathbf{x} - \mathbf{y_1}$. In fact, $(r, [\mathbf{w_1}, \mathbf{w_2}]^T)$ is a valid signature for the message u under the identity ID_s^*. This claim will be proven in Lemma 8. Finally, $\mathcal{C}$ encrypts the related information as aforementioned to obtain a ciphertext $\mathbf{c}$ and sends it to $\mathcal{F}$ as a reply.
  - Unsigncrypt queries $(\mathbf{c}, ID_s, ID_r)$: One case is that the receiver's identity $ID_r \neq ID_s^*$. $\mathcal{C}$ decrypts the ciphertext $\mathbf{c}$ to obtain a signature $(r, \mathbf{w})$ and a message u by using the trapdoor of ID_r. If $(r, \mathbf{w})$ is a valid signature for u under identity ID_s, $\mathcal{C}$ replies u as the answer; otherwise, $\mathcal{C}$ replies $\perp$.

    The other case is that the receiver's identity $ID_r = ID_s^*$. Note that in this case, the corresponding matrix used for encryption is

$$\begin{aligned}\mathbf{A_{ID_r}} &= \Big[\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^* + (H(ID_r) - H(ID_s^*))\\ &\qquad \mathbf{B_1}\|\mathbf{A_0}\mathbf{Q}^* + (H(\mathbf{t}) - H(\mathbf{t}^*))\mathbf{B_2}\Big]\\ &= [\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^*\|\mathbf{A_0}\mathbf{Q}^* + (H(\mathbf{t}) - H(\mathbf{t}^*))\mathbf{B_2}]\end{aligned} \tag{16}$$

    $\mathcal{C}$ does not know the trapdoor $\mathbf{T'_{ID}}$, but it w.o.p. can use the trapdoor of $\Lambda^{\perp}(\mathbf{B_2})$ to decrypt. The probability that $\mathcal{F}$ generates a valid ciphertext with $\mathbf{t} = \mathbf{t}^*$ is negligible, because the vector $\mathbf{t}^*$ is hidden from $\mathcal{F}$, and $\mathbf{t}$ is the hash value of H_N (Setup 4.1). As a result, $\mathcal{C}$ can change steps 1(a) and (b) of Unsigncrypt as follows:

    1(a′) Sample

$$\mathbf{s_i} = \mathbf{SampleRightv}(\mathbf{A_0}, \mathbf{B_2}, H(\mathbf{t}) - H(\mathbf{t}^*), \mathbf{Q}^*, \mathbf{0}, \mathbf{T_{B_2}}, \mathbf{g_i}, s)$$

    for $i \in [\ell]$, where $\mathbf{g_i}$ is the ith column of matrix $\mathbf{G}$.

    1(b′) Let $\mathbf{E_{ID_r}} = [\mathbf{s_1}, \cdots, \mathbf{s_\ell}] \in \mathbb{Z}^{2m\times\ell}$.

    Now, we have $\mathbf{A_{ID_r}}\mathbf{E_{ID_r}} = \mathbf{G}$, and the vector $\mathbf{s_i}$ is not longer than the vector obtained in the original steps 1(a) and (b), by a similar discussion in Remark 3. As a result, $\mathcal{C}$ can obtain the correct vector $\mathbf{b}$ by using $\mathbf{E_{ID_r}}$ in step 1(c). Then, $\mathcal{C}$ executes normally the subsequent steps in Unsigncrypt to answer the queries.
- Forgery: Finally, $\mathcal{F}$ outputs a valid signcryption $\mathbf{c}$ under the challenge identity ID_s^* and some receiver ID_r. $\mathcal{C}$ decrypts $\mathbf{c}$ as aforementioned to obtain a valid signature $\left(r', \left[\mathbf{z'_0}, \mathbf{z'_1}, \mathbf{z'_2}\right]^T\right)$ for a message u under ID_s^*. According to whether the signcryption of u under $ID_s = ID_s^*$ has been queried, there exist two cases:
  - Case 1: The signcryption of u under $ID_s = ID_s^*$ has not been queried. This means that $\left(r', \left[\mathbf{z'_0}, \mathbf{z'_1}, \mathbf{z'_2}\right]^T\right)$ is an existential forgery, where $\mathbf{z'_0} \in \mathbb{Z}_q^m, \mathbf{z'_1} \in \mathbb{Z}_q^m, \mathbf{z'_2} \in \mathbb{Z}_q^m$. By this forgery signature, we obtain $\left[\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^*\|\mathbf{A_0}\mathbf{F'_{sr}}\right]\begin{bmatrix}\mathbf{z'_0}\\ \mathbf{z'_1}\\ \mathbf{z'_2}\end{bmatrix} = \mathbf{y}$. On the other hand, we have $\mathbf{A_0}\mathbf{x} = \mathbf{y}$. Let $\mathbf{x_1} = \mathbf{z'_0} + \mathbf{R}^*\mathbf{z'_1} + \mathbf{F'_{sr}}\mathbf{z'_2} - \mathbf{x}$. It follows that $\mathbf{A_0}\mathbf{x_1} = \mathbf{0}$. According

to Proposition 5 term 5, it follows that $\Pr[\mathbf{x_1} = \mathbf{0}]$ is negligible. Let us compute $\|\mathbf{x_1}\|$.

$$\begin{aligned}\|\mathbf{x_1}\| &= \|\mathbf{z_0'} + \mathbf{R}^*\mathbf{z_1'} + \mathbf{F_{sr}'}\mathbf{z_2'} - \mathbf{x}\| \\ &\le \|\mathbf{z_0'}\| + \|\mathbf{R}^*\| \cdot \|\mathbf{z_1'}\| + \|\mathbf{F_{sr}'}\| \cdot \|\mathbf{z_2'}\| + \|\mathbf{x}\| \\ &\le s\sqrt{m} + \|\mathbf{R}^*\| \cdot s\sqrt{m} + \|\mathbf{F_{sr}'}\| \cdot s\sqrt{m} \\ &\quad + s\sqrt{3m} \\ &\le s\sqrt{m} + C\sqrt{m+m} \cdot s\sqrt{m} + \kappa s_1\sqrt{m} \cdot s\sqrt{m} \\ &\quad + s\sqrt{3m} \\ &\le sm \cdot (\sqrt{2}C + \kappa s_1) \\ &< \beta \end{aligned} \tag{17}$$

The last inequality holds, because $C = 12$ according to Proposition 10, and κ needs to be big enough, that is, $k > 80$. Consequently, if $\mathcal{A}$ makes an existential forgery, $\mathcal{C}$ can obtain a solution for $SIS_{q,m,\beta}$ w.o.p.

– Case 2: The signcryption of $\mathbf{u}$ under $ID_s = ID_s^*$ has been queried, which means that $(r', [\mathbf{z_0'}, \mathbf{z_1'}, \mathbf{z_2'}]^T)$ is a strong forgery. It concludes that $[\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^*\|\mathbf{A_0}\mathbf{F_{sr}''}] \begin{bmatrix} \mathbf{z_0'} \\ \mathbf{z_1'} \\ \mathbf{z_2'} \end{bmatrix} = \mathbf{y}$ where $\upsilon' = H_1(u, r', ID_r)$ and $\mathbf{F_{sr}''} = \sum_{i=1}^{\kappa}(-1)^{\upsilon'[i]}\mathbf{F_i}$ as in the scheme. Let $(r, [\mathbf{w_1}, \mathbf{w_2}]^T)$ be a signature included in a replied signcryption, where $\mathbf{w_1} \in \mathbb{Z}^m, \mathbf{w_2} \in \mathbb{Z}^{2m}$. For simplicity, we write $[\mathbf{w_1}, \mathbf{w_2}]^T$ as $[\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}]^T$, where $\mathbf{z_0} \in \mathbb{Z}^m, \mathbf{z_1} \in \mathbb{Z}^m, \mathbf{z_2} \in \mathbb{Z}^m$. Then, it concludes that $[\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^*\|\mathbf{A_0}\mathbf{F_{sr}'}] \begin{bmatrix} \mathbf{z_0} \\ \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix} = \mathbf{y}$, where $\upsilon = H_1(u, r, ID_r)$, $\mathbf{F_{sr}'} = \sum_{i=1}^{\kappa}(-1)^{\upsilon[i]}\mathbf{F_i}$. Let $\mathbf{x_1} = (\mathbf{z_0'} - \mathbf{z_0}) + \mathbf{R}^*(\mathbf{z_1'} - \mathbf{z_1}) + \mathbf{F_{sr}'}(\mathbf{z_1'} - \mathbf{z_1})$. Then it follows that $\mathbf{A_0}\mathbf{x_1} = \mathbf{0}$. It concludes that $\Pr[\mathbf{x_1} = \mathbf{0}] \le 1/3$ according to Lemma 26 in [23]. Let us evaluate the size of $\|\mathbf{x_1}\|$:

$$\begin{aligned}\|\mathbf{x_1}\| &= \|(\mathbf{z_0'} - \mathbf{z_0}) + \mathbf{R}^*(\mathbf{z_1'} - \mathbf{z_1}) + \mathbf{F_{sr}'}(\mathbf{z_1'} - \mathbf{z_1})\| \\ &\le \|\mathbf{z_0'} - \mathbf{z_0}\| + \|\mathbf{R}^*\| \cdot \|\mathbf{z_1'} - \mathbf{z_1}\| + \|\mathbf{F_{sr}'}\| \\ &\quad \cdot \|\mathbf{z_2'} - \mathbf{z_2}\| \\ &\le s\sqrt{2m} + \|\mathbf{R}^*\| \cdot s\sqrt{2m} + \|\mathbf{F_{sr}'}\| \cdot s\sqrt{2m} \\ &\le s\sqrt{2m} + C\sqrt{m+m} \cdot s\sqrt{2m} + \kappa s_1\sqrt{m} \\ &\quad \cdot s\sqrt{2m} \\ &\le sm \cdot (2C + \sqrt{2}\kappa s_1) \\ &\le \beta \end{aligned} \tag{18}$$

As a result, $\mathcal{C}$ can obtain a solution for $SIS_{q,m,\beta}$ with at least 2/3 probability. □

Lemma 8. *The above* $(r, [\mathbf{w_1}, \mathbf{w_2}]^T)$ *is a valid signature.*

Proof. This lemma can be guaranteed by the following two aspects.

First, $\upsilon = H_1(u, r, ID)$, $\mathbf{F_{sr}'} = \sum_{i=1}^{\kappa}(-1)^{\upsilon[i]}\mathbf{F^{(i)\prime}}$,

$$\begin{aligned}&[\mathbf{A_0}\|\mathbf{A_1} + H_1(ID_s^*)\mathbf{B_1}\|\mathbf{F_{sr}}] \begin{bmatrix} \mathbf{w_1} \\ \mathbf{w_2} \end{bmatrix} \\ &= [\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^* + (H_1(ID_s^*) - H_1(ID_s^*))\mathbf{B_1}\|\mathbf{A_0}\mathbf{F_{sr}'}] \begin{bmatrix} \mathbf{w_1} \\ \mathbf{w_2} \end{bmatrix} \\ &= [\mathbf{A_0}\|\mathbf{A_0}\mathbf{R}^*\|\mathbf{A_0}\mathbf{F_{sr}'}] \begin{bmatrix} \mathbf{w_1} \\ \mathbf{w_2} \end{bmatrix} \\ &= [\mathbf{A_0}\|\mathbf{A_0}(\mathbf{R}^*\|\mathbf{F_{sr}'})] \begin{bmatrix} \mathbf{w_1} \\ \mathbf{w_2} \end{bmatrix} \\ &= \mathbf{A_0}\mathbf{w_1} + \mathbf{A_0}(\mathbf{R}^*\|\mathbf{F_{sr}'})\mathbf{w_2} \\ &= \mathbf{A_0}(\mathbf{x} - \mathbf{y_1}) + \mathbf{A_0}\mathbf{y_1} \\ &= \mathbf{A_0}\mathbf{x} \\ &= \mathbf{y} \end{aligned} \tag{19}$$

Second,

$$\begin{aligned}&\left\| \begin{bmatrix} \mathbf{w_1} \\ \mathbf{w_2} \end{bmatrix} \right\| = \left\| \begin{bmatrix} \mathbf{x} - \mathbf{y_1} \\ \mathbf{w_2} \end{bmatrix} \right\| \le \|\mathbf{x} - \mathbf{y_1}\| + \|\mathbf{w_2}\| \le \|\mathbf{x}\| \\ &+ \|\mathbf{w_2}\| + \|\mathbf{w_2}\| \\ &\le \|\mathbf{x}\| + \|[\mathbf{R}^*\|\mathbf{F_{sr}'}]\mathbf{z_1}\| + \|\mathbf{w_2}\| \le s\sqrt{m} \\ &\quad + \max(\|\mathbf{R}^*\|, \|\mathbf{F_{sr}'}\|)s\sqrt{2m} + s\sqrt{2m} \\ &\le s\sqrt{m} + \max(C\sqrt{m+m}, \kappa s_1\sqrt{m})s\sqrt{2m} + s\sqrt{2m} \\ &\le sm \cdot (2C + \sqrt{2}\kappa s_1) = \beta \end{aligned} \tag{20}$$

□

Remark 1. There is a kind of trivial attacks against the strong unforgeability of the proposed signcryption schemes. Note that in inner attacks, the forger $\mathcal{F}$ is the receiver ID_r. The attack procedure is as follows. First, $\mathcal{F}$ queries the signcryption about a message u^* under the sender ID_s^* and the receiver ID_r to obtain an answer $\mathbf{c}^*$. Next, $\mathcal{F}$ decrypts $\mathbf{c}^*$ with his or her own private key to obtain a signature $(r, \mathbf{w})$, where $\mathbf{w} \in \mathbb{Z}^{3m}$. Then, $\mathcal{F}$ encrypts the signature $(r, \mathbf{w})$ and the message u^* under a new tag to obtain a new ciphertext $\mathbf{c}'$. Obviously, it is a valid ciphertext. That is to say, it is a valid forgery. However, because the binding relationship between the message u^* and the signature $(r, \mathbf{w})$ is not changed, the receiver cannot make any false accusation against the sender by using this kind of forgery.

5. EXTENSION I: SELECTIVELY SECURE IBSC FROM LWE ASSUMPTION

5.1. Construction

In fact, the proposed method to construct IBSC can be easily transformed to a construction based on LWE. For

convenience, let us name this scheme as SE_SC. The construction frame is similar. Hence, we only list the different steps. In the Signcrypt algorithm, steps 3(e) and (f) can be replaced with steps 3(e′) and 3(f′) given as follows:

3(e′) Choose $\mathbf{x} \leftarrow \bar{\Psi}_{\alpha}^{m}(\in \mathbb{Z}_q^m), \mathbf{z_0} \leftarrow \bar{\Psi}_{\alpha}^{m}(\in \mathbb{Z}_q^m)$, then compute $\mathbf{z_1} = \mathbf{R}\mathbf{z_0} \in \mathbb{Z}_q^m, \mathbf{z_2} = \mathbf{Q}\mathbf{z_0} \in \mathbb{Z}_q^m$.

3(f′) Compute $\mathbf{c_0} = \mathbf{G}^T\mathbf{s} + \mathbf{x} + \sigma_1\lfloor q/2 \rfloor \in \mathbb{Z}_q^m$, and compute $\mathbf{c_1} = \mathbf{A}_{\mathbf{ID_r}}^T\mathbf{s} + \begin{bmatrix} \mathbf{z_0} \\ \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix} \in \mathbb{Z}_q^{3m}$.

5.2. Consistency and parameter settings

Lemma 9. *The size of every entry of the error vector in the decryption is lower than* $O(sm^{1.5}) + sm\alpha q\omega(\sqrt{\log m})$ *with high probability*

Proof. In the decryption step,

$$\begin{aligned}\mathbf{w} = \mathbf{c_0} - \mathbf{E}_{\mathbf{ID_r}}^T\mathbf{c_1} &= (\mathbf{G}^T\mathbf{s} + \mathbf{x} + \sigma_1\lfloor q/2 \rfloor) \\ &\quad - \mathbf{E}_{\mathbf{ID_r}}^T\left(\mathbf{A}_{\mathbf{ID_r}}^T\mathbf{s} + \begin{bmatrix} \mathbf{z_0} \\ \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix}\right) \\ &= \mathbf{x} - \mathbf{E}_{\mathbf{ID_r}}^T\begin{bmatrix} \mathbf{z_0} \\ \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix} + \sigma_1\lfloor q/2 \rfloor\end{aligned} \tag{21}$$

The error vector is $\mathbf{w}' = \mathbf{x} - \mathbf{E}_{\mathbf{ID_r}}^T\begin{bmatrix} \mathbf{z_0} \\ \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix}$. Let the ith column of the matrix $\mathbf{E}_{\mathbf{ID_r}}$ be $\mathbf{e_i} = \begin{bmatrix} \mathbf{e_{i_0}} \\ \mathbf{e_{i_1}} \\ \mathbf{e_{i_2}} \end{bmatrix} \in \mathbb{Z}_q^{3m}$ for $0 \le i < m$, where $\mathbf{e_{i_0}}, \mathbf{e_{i_1}}, \mathbf{e_{i_2}} \in \mathbb{Z}_q^m$. The ith element of the error vector is

$$\mathbf{w}'[i] = \mathbf{x} - \left\langle \begin{bmatrix} \mathbf{e_{i_0}} \\ \mathbf{e_{i_1}} \\ \mathbf{e_{i_2}} \end{bmatrix}, \begin{bmatrix} \mathbf{z_0} \\ \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix} \right\rangle$$

Its magnitude is

$$\begin{aligned}|\mathbf{w}'[i]| &= \left|\mathbf{x} - \left\langle \begin{bmatrix} \mathbf{e_{i_0}} \\ \mathbf{e_{i_1}} \\ \mathbf{e_{i_2}} \end{bmatrix}, \begin{bmatrix} \mathbf{z_0} \\ \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix} \right\rangle\right| \\ &\le |\mathbf{x}| + |\langle \mathbf{e_{i_0}}, \mathbf{z_0}\rangle| + |\langle \mathbf{e_{i_1}}, \mathbf{z_1}\rangle| + |\langle \mathbf{e_{i_2}}, \mathbf{z_2}\rangle| \\ &= |\mathbf{x}| + |\langle \mathbf{e_{i_0}}, \mathbf{z_0}\rangle| + |\langle \mathbf{e_{i_1}}, \mathbf{R}\mathbf{z_0}\rangle| + |\langle \mathbf{e_{i_2}}, \mathbf{Q}\mathbf{z_0}\rangle| \\ &= |\mathbf{x}| + |\langle \mathbf{e_{i_0}}, \mathbf{z_0}\rangle| + |\langle \mathbf{R}^T\mathbf{e_{i_1}}, \mathbf{z_0}\rangle| + |\langle \mathbf{Q}^T\mathbf{e_{i_2}}, \mathbf{z_0}\rangle| \\ &= |\mathbf{x}| + |\langle \mathbf{e_{i_0}} + \mathbf{R}^T\mathbf{e_{i_1}} + \mathbf{Q}^T\mathbf{e_{i_2}}, \mathbf{z_0}\rangle| \\ &\le |\mathbf{x}| + (s\sqrt{m} + s\sqrt{m}C\sqrt{m+m} + s\sqrt{m}C\sqrt{m+m})\|\mathbf{z_0}\| \\ &\le O(sm)(\sqrt{m} + \alpha q\omega(\sqrt{\log m})) \\ &= O(sm^{1.5}) + sm\alpha q\omega(\sqrt{\log m})\end{aligned} \tag{22}$$

where the last inequality holds due to Proposition 10. This completes the proof. □

In order to guarantee the system to work well, the following requirements should be satisfied:

- **TrapGen** algorithm should work well, that is, $m \ge 6n\log q$ [30].
- **SampleLeft** and **SampleRight** algorithm should work well. It needs s_1 to be big enough, that is, $s_1 > m\sqrt{\log m}$ [22].
- The error should be small enough, that is, $C'm^2 \le \lfloor q/4 \rfloor$.
- LWE must be hard, that is, $\alpha q > 2\sqrt{n}$.

Similarly, set δ to be real such that $n^{\delta} > \lceil \log q \rceil$ where n is the security parameter. To satisfy the earlier constraints, the parameters could be set as follows:

$$\begin{aligned} &m = 6n^{1+\delta}, \quad s_1 = m \cdot \omega(\sqrt{\log n}), \\ &q = m^{2.5} \cdot \omega(\sqrt{\log n}), \quad \alpha = (m^2\omega(\log n))^{-1}\end{aligned}$$

The security proof, including the methods to insert difficulty problems and the skill to answer the queries from the adversary, for SE_SC is similar to the proof given in Section 4.3.

6. EXTENSION II: ADAPTIVELY SECURE CONSTRUCTION

6.1. Adaptively secure IBSC from LWR assumption

By using the technique [22,23], we can construct a fully secure scheme based on LWR, denoted by AR_SC.

- **Setup**(1^n):
  (1) This step is identical to the corresponding step of SR_SC.
  (2) Choose appropriate positive integers $\gamma, \iota, \kappa, \rho, \varrho, \tau$.
  (3) Identical.
  (4) Generate master public key and a secret key:
    – Identical.
    – Select uniformly random matrices $\mathbf{A_i} \in \mathbb{Z}_q^{n\times m}$ for $i \in [\tau + 1]$, $\mathbf{B_1} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{B_2} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{G} \in \mathbb{Z}_q^{n\times \ell}$, $\mathfrak{F} = \{\mathbf{F_i} \in \mathbb{Z}_q^{n\times m}\}_1^{\kappa}$ and a vector $\mathbf{y} \in \mathbb{Z}_q^n \backslash \{\mathbf{0}\}$; $mpk \triangleq \{\mathbf{A_0}, \mathbf{A_1}, \mathbf{A_2}, \ldots, \mathbf{A_\tau}, \mathbf{A_{\tau+1}}, \mathbf{B_1}, \mathbf{B_2}, \mathbf{G}, \mathbf{y}, \mathfrak{F}, H, H_1, H_2, H_N\}$, $msk \triangleq \mathbf{T_{A_0}}$.
    – Identical.

- **Extract**(msk, ID):

  (1) Compute $\widehat{\mathbf{A_{ID}}} = \mathbf{B_1} + \sum_{i=1}^{\tau} f_i \mathbf{A_i}$, where $ID = (f_1, f_2, \ldots, f_\tau) \in \{0,1\}^\tau$. Run $\mathbf{T'_{ID}} \leftarrow \mathbf{SampleBasisLeft}(\mathbf{A_0}, \widehat{\mathbf{A_{ID}}}, \mathbf{T_{A_0}}, s) \in \mathbb{Z}^{m\times m}$.
  (2) Let $SK_{ID} = \mathbf{T'_{ID}}$. $\mathbf{T'_{ID}}$ is a trapdoor of $\mathbf{\Lambda}^{\perp}(\mathbf{A'_{ID}})$ according to Proposition 8, where $\mathbf{A'_{ID}} = [\mathbf{A_0} \| \widehat{\mathbf{A_{ID}}}]$.
  (3) Identical.

- **Signcrypt**$(u, \mathbf{T'_{ID_s}}, ID_r)$:

  (1–2) The steps are identical to the corresponding steps 1–2 of Signcrypt algorithm of SR_SC;
  (3) Encrypt σ_1:

    (a) Identical.
    (b) Compute $\widehat{\mathbf{A_{ID_r}}} = \mathbf{B_1} + \sum_{i=1}^{\tau} f_i \mathbf{A_i}$, where $ID_r = (f_1, f_2, \cdots, f_\tau) \in \{0,1\}^\tau$. Construct $\mathbf{A'_{ID_r}} = [\mathbf{A_0} \| \widehat{\mathbf{A_{ID_r}}}]$. Construct the matrix used for encryption $\mathbf{A_{ID_r}} = [\mathbf{A'_{ID_r}} \| \mathbf{A_{\tau+1}} + H(\mathbf{t})\mathbf{B_2}]$.
    (c) Identical.
    (d) Choose matrices $\mathbf{R_i} \xleftarrow{\$} \{-1,1\}^{m\times m}$ for $i \in [\tau]$ and $\mathbf{Q} \xleftarrow{\$} \{-1,1\}^{m\times m}$. Compute $\mathbf{R} = \sum_{i=1}^{\tau} f_i \mathbf{R_i}$, where f_i is the ith bit of $ID_r \in \{0,1\}^\tau$.
    (e)–(f) The steps are identical to the corresponding steps in SR_SC.

  (4) Identical.

- **Unsigncrypt**$(\mathbf{c}, \mathbf{T_{ID_r}}, ID_s)$:

  (1) Public decrypt

    (a) Let $\mathbf{g_i}$ denote the ith column of $\mathbf{G}$. Sample $\mathbf{x_i} \leftarrow D_{\mathbb{Z}^m, s}$, then compute $\mathbf{x'_i} = [\mathbf{A_{\tau+1}} + H(\mathbf{t})\mathbf{B_2}]\mathbf{x_i}$ for $i \in [m]$.
    (b) Let $\mathbf{E_{ID_r}} = \begin{bmatrix} \mathbf{d_1} \ldots \mathbf{d_m} \\ \mathbf{x_1}, \ldots, \mathbf{x_m} \end{bmatrix}$, where $\mathbf{d_i} = \mathbf{SamplePre}(\mathbf{T'_{ID_r}}, \mathbf{A'_{ID_r}}, \mathbf{g_i} - \mathbf{x'_i}, s)$ for $i \in [m]$, $\widehat{\mathbf{A_{ID_r}}} = \mathbf{B_1} + \sum_{i=1}^{\tau} f_i \mathbf{A_i}$ and $\mathbf{A'_{ID_r}} = [\mathbf{A_0} \| \widehat{A}_{\mathbf{ID_r}}]$.
    (c)–(d) The steps are identical to the corresponding steps in SR_SC.

  (2) Identical.
  (3) Verify the sender's authenticity:

    (a)–(d) The steps are identical to the corresponding steps in SR_SC.
    (e) If $[\mathbf{A_0} \| \mathbf{B_1} + \sum_{i=1}^{\tau} f'_i \mathbf{A_i} \| \mathbf{F'_{sr}}]\sigma' \neq \mathbf{y}$ output $\perp$; else, output u'. Here, f'_i is the ith bit of the receiver's identity.

The analysis for error is basically identical to that in SR_SC. The only difference step, namely inequality 10, should be changed as follows:

$$\begin{aligned}\|e\| &\leq \left\| \lfloor \mathbf{e_{id}^T e_1} \rfloor + \mathbf{e_{id}^T} \begin{bmatrix} \mathbf{0} \\ \lfloor \mathbf{R}^T \mathbf{e_0} \rfloor \\ \lfloor \mathbf{Q}^T \mathbf{e_0} \rfloor \end{bmatrix} \right\| \leq \|\mathbf{e_{id}^T}\| \sqrt{3m} + \|\mathbf{e_{id|1}^T}\| \\ &\quad \|\mathbf{R}^T\| \sqrt{m} + \|\mathbf{e_{id|2}^T}\| \|\mathbf{Q}^T\| \sqrt{m} \\ &\leq s_1 \sqrt{3m} \sqrt{3m} + s_1 \sqrt{m} \tau C \sqrt{m+m} \sqrt{m} \\ &\quad + s_1 \sqrt{m} C \sqrt{m+m} \sqrt{m} \leq C'' m^2 \end{aligned} \tag{23}$$

Let Q denote the number of private key queries issued by $\mathcal{A}$. For security proof , $q > 2Q$ is required [22]. The parameters can be set as follows:

$$m = 6n^{1+\delta}, \quad s_1 = m\tau \cdot \omega(\sqrt{\log n}), \quad p = O(m^2)$$

$$\alpha = (m^2 \omega(\log n))^{-1}, \quad q = \max(2Q, m^{3+1.5(1+\delta)^{-1}} \omega(\sqrt{\log n}))$$

The security of SE_SC can be proved by composing the proof technique in SR_SC and the "artificial abort" technique of Agrawal *et al.* [22].

6.2. Adaptively secure identity-based signcryption from LWE assumption

An adaptively secure IBSC scheme from LWE assumption, denoted by AE_SC, can also be constructed. Only steps 3(e) and (f) need to be changed as follows.

3(e′) Choose $\mathbf{x} \leftarrow \bar{\Psi}_\alpha^m (\in \mathbb{Z}_q^m), \mathbf{z_0} \leftarrow \bar{\Psi}_\alpha^m (\in \mathbb{Z}_q^m)$, then compute $\mathbf{z_1} = \mathbf{R}\mathbf{z_0} \in \mathbb{Z}_q^m, \mathbf{z_2} = \mathbf{Q}\mathbf{z_0} \in \mathbb{Z}_q^m$;
3(f′) Compute $\mathbf{c_0} = \mathbf{G}^T\mathbf{s} + \mathbf{x} + \sigma_1 \lfloor q/2 \rfloor \in \mathbb{Z}_q^m$, and compute

$$\mathbf{c_1} = \mathbf{A}_{\mathbf{ID_r}}^T \mathbf{s} + \begin{bmatrix} \mathbf{z_0} \\ \mathbf{z_1} \\ \mathbf{z_2} \end{bmatrix} \in \mathbb{Z}_q^{3m}.$$

The analysis for error is similar to that of SE_SC (Lemma 9). Because in AE_SC, $\mathbf{R} \leq \tau C \sqrt{m+m}$, the error $\mathbf{w'}[i] \leq O(sm^{1.5}) + sm\tau\alpha q\omega(\sqrt{\log m})$. Therefore, the parameters can be set as follows:

$$\begin{aligned} m &= 6n^{1+\delta}, \quad q = \max(2Q, m^{2.5} \cdot \omega(\sqrt{\log n})), \\ s_1 &= m\tau \cdot \omega(\sqrt{\log n}), \quad \alpha = (\tau^2 m^2 \omega(\log n))^{-1} \end{aligned}$$

The procedure for security reduction is basically similar to that of AR_SC.

7. PERFORMANCE

7.1. Performance of the proposed schemes

Firstly, let us analyze the sizes of the public parameters (pp), the master public key (mpk), the master private key

(msk) and the private keys of users (usk), and the ciphertext (cp) length. In SR_SC, SE_SC, AR_SC, and AE_SC, the master public keys are all $nm \log q$-bit length, the master private keys are all $m^2 \log q$-bit length, and the private keys are all $4m^2 \log q$-bit length. In SR_SC and SE_SC, the sizes of the public parameters are both $((5+\kappa)m + \ell + 1) \cdot n \log q$ bits. In AR_SC and AE_SC, the sizes of the public parameters are both $((4+\kappa+\tau)m + \ell + 1) \cdot n \log q$ bits. In SR_SC and AR_SC, the sizes of the ciphertext are both $(3m + \ell) \log p + n \log q + \varrho$ bits. In SE_SC and AE_SC, the sizes of the ciphertext are both $(3m + \ell + n) \cdot \log q + \varrho$ bits.

Secondly, let us evaluate the cost of the signcryption schemes. In SR_SC, SE_SC, AR_SC, and AE_SC, the numbers of discrete Gaussian sample (DGS) are all 1 with dimension m, and the pre-image samples (PIS) are all 1 with dimension $2m$. The multiplications over $\mathbb{Z}_q$ distribute at steps 1(d), 3(b), (e), and (f). In SR_SC and AR_SC, the numbers of multiplications over $\mathbb{Z}_q$ are both about $n^2m + 4nm + n\ell$. In SE_SC and AE_SC, the numbers of multiplications over $\mathbb{Z}_q$ are both about $n^2m + 4nm + n\ell + 2m^2$. Here, we ignore the costs of additions over $\mathbb{Z}_q$ and $\mathbb{Z}_p$, because they are far less than that of multiplications.

Thirdly, let us evaluate the cost of the unsigncryption schemes. In SR_SC, SE_SC, AR_SC, and AE_SC, the numbers of DGS are all ℓ with dimension m, and the numbers of PIS are all ℓ with dimension $2m$. The multiplications over $\mathbb{Z}_q$ distribute at steps 1(a), 1(c), and 3(e). In SR_SC and AR_SC (respectively, SE_SC and AE_SC), the numbers of multiplications over $\mathbb{Z}_q$ are both about $n^2m + 4nm$ (respectively, $n^2m + 4nm + 3\ell m$). In addition, in SR_SC and AR_SC, the numbers of the multiplications over $\mathbb{Z}_p$ are all $3\ell m$. For clarity, the results are listed in Table II.

7.2. Performance comparisons with sign-then-encrypt approach

The proposed schemes should be compared with the sign-then-encrypt approach (**StE**) to indicate their efficiency, because the function of signcryption can be implemented by another mechanism, namely **StE** approach. For the sake of fairness, the comparisons should be executed under the same (or similar) requirements, namely, using the same trapdoor technique [30] and having the same security level. Suppose that the hybrid encryption is also used in **StE** to improve efficiency and the signature procedure in **StE** is the same as in step 1 of Signcrypt algorithm. There is no IBE based on the trapdoor of Alwen and Peikert [30] to achieve IND-CCA2 security. A general method to construct an IND-CCA2 secure IBE is the transform technique from an $(\ell+1)$-level indistinguishable under chosen plaintext attack secure HIBE † scheme to an ℓ-level IND-CCA2 secure HIBE scheme with method [25,26](CHK transformation). To save space cost, the sender can use his or her own public key and private key to finish the extra signature. As a result, the public/private key sizes of the sender (respectively, the receiver) in the proposed schemes are identical to that of the sender (respectively, the receiver) of **StE**. In the following, we compare the ciphertext size, the computational cost of the sender, and the computational cost of the receiver between the proposed schemes and the approach of **StE**, respectively. Note that, for simplicity, we only compare the scheme SR_SC with the approach of **StE** because the comparisons between the other proposed schemes and **StE** are similar.

Firstly, let us compare the ciphertext size. The ciphertext of SR_SC is $\mathbf{c_{SR}} = (\mathbf{t}, \mathbf{c_0}, \mathbf{c_1}, c_2)$, where $|\mathbf{c_2}| = |\sigma_2| + |r_1| + |r_2| + |u|$. As a result, the total length is $|\mathbf{c_{SR}}| = |\mathbf{t}| + |\mathbf{c_0}| + |\mathbf{c_1}| + |\sigma_2| + |r_1| + |r_2| + |u|$. In **StE**, the ciphertext is $\mathbf{c_{StE}} = (\mathbf{t}, \mathbf{c_0}, \mathbf{c_1}, \mathbf{c_2}, r_1, \sigma, r_2, \sigma')$ where $|c_2| = |u|$, $|r_2| = |r_1|, |\sigma| = |\sigma'| = |\sigma_1| + |\sigma_2|$. Therefore, the total length of ciphertext in **StE** is $|\mathbf{c_{StE}}| = |\mathbf{t}| + |\mathbf{c_0}| + |\mathbf{c_1}| + |r_1| + |r_2| + |u| + |\sigma| + |\sigma'|$. As a result, our ciphertext saves $|\sigma_1| + |\sigma|$, namely, $\ell + 3m \log q$ bits. The percentage between the ciphertext expansion of scheme SR_SC and that in **StE** is as follows:

$$
\begin{aligned}
\wp &= \frac{|\mathbf{c_{SR}}| - |u|}{|\mathbf{c_{StE}}| - |u|} \\
&= \frac{|\mathbf{t}| + |\mathbf{c_0}| + |\mathbf{c_1}| + |\sigma_2| + |r_1| + |r_2|}{|\mathbf{t}| + |\mathbf{c_0}| + |\mathbf{c_1}| + |r_1| + |r_2| + |\sigma| + |\sigma'|} \\
&= \frac{n \log q + \ell \log p + 3m \log p + 3m \log q - \iota + m' + \gamma + \iota}{n \log q + \ell \log p + 3m \log p + \gamma + 3m \log q + m' + \iota + \gamma + 3m \log q} \\
&\approx \frac{n \log q + (\ell + 3m) \log p + 3m \log q}{n \log q + (\ell + 3m) \log p + 6m \log q} \\
&\approx \frac{3m \log p + 3m^{*}2.5 \log p}{3m \log p + 6m^{*}2.5 \log p} \\
&= \frac{7}{12}
\end{aligned}
\tag{24}
$$

Here, the first "$\approx$" holds because it is enough to set γ and ι to be 80 for the collision-resistance property of the hash function H_1, H_N, and this value is far less than the other values. The second "$\approx$" holds because of $m' < m$, $n \ll m$, $\ell \ll m$ and $p < q$. According to the parameter setting in Section 4.2, we have $\log q \approx 2.5 \log p$. As a result, the last "$\approx$" holds. The analysis shows that the ciphertext expansion of SR_SC is 7/12 of that of **StE**.

Secondly, let us compare the computational cost of the sender. The signcryption cost of the proposed scheme is $(n\ell + 3nm + n^2m)\mathtt{M} + m\mathtt{DGS} + 2m\mathtt{PIS}$. As noted earlier, **StE** needs an extra signature to ensure the encryption to be IND-CCA2, whose cost is $m\mathtt{DGS} + 2m\mathtt{PIS}$. As a result, the ratio between the computational cost of SR_SC and that of

† Hierarchical identity-based encryption (HIBE) is a generalization for identity-based encryption (IBE), in which the entities are arranged according to tree structure. Each entity can only delegate the secret keys for its descendants.

Table II. Performance of schemes.

		SR_SC	SE_SC	AR_SC	AE_SC
mpk. size		$nm\log q$	$nm\log q$	$nm\log q$	$nm\log q$
msk. size		$m^2\log q$	$m^2\log q$	$m^2\log q$	$m^2\log q$
usk size		$4m^2\log q$	$4m^2\log q$	$4m^2\log q$	$4m^2\log q$
pp. size		$((5+\kappa)m+\ell+1)$ $\cdot n\log q$	$((5+\kappa)m+\ell+1)$ $\cdot n\log q$	$((4+\kappa+\tau)m+$ $\ell+1)\cdot n\log q$	$((4+\kappa+\tau)m+$ $\ell+1)\cdot n\log q$
cp. size		$(3m+\ell)\log p$ $+n\log q+\varrho$	$(3m+\ell+n)$ $\cdot\log q+\varrho$	$(3m+\ell)\log p$ $+n\log q+\varrho$	$(3m+\ell+n)$ $\cdot\log q+\varrho$
sg. cost	DGS*	$1\times m^{\P}$	$1\times m$	$1\times m$	$1\times m$
	PIS†	$1\times 2m^{\|\|}$	$1\times 2m$	$1\times 2m$	$1\times 2m$
	$\times\mathbb{Z}_q$‡	$n^2m+4nm+n\ell$	n^2m+4nm $+n\ell+2m^2$	$n^2m+4nm+nl$	n^2m+4nm $+n\ell+2m^2$
us. cost	DGS	$\ell\times m$	$\ell\times m$	$\ell\times m$	$\ell\times m$
	PIS	$\ell\times 2m$	$\ell\times 2m$	$\ell\times 2m$	$\ell\times 2m$
	$\times\mathbb{Z}_q$	n^2m+4nm	n^2m+4nm $+3\ell m$	n^2m+4nm	n^2m+4nm $+3\ell m$
	$\times\mathbb{Z}_p$§	$3\ell m$	0	$3\ell m$	0

*Discrete Gaussian sampling (DGS), that is, SampleZ: $z_i \leftarrow D_{\mathbb{Z},s_i',c_i'}$ in [11].

†Pre-image sampling (PIS).

‡Multiplication over $\mathbb{Z}_q$.

¶The number of DGS is 1, and the dimension of each DGS is m.

§Multiplication over $\mathbb{Z}_p$.

‖The number of PIS is 1, and the dimension of each PIS is $2m$.

StE is given as follows:

$$
\begin{aligned}
\varsigma &= \frac{c(\mathrm{SR\_SC})}{c(\mathbf{StE})}\\
&= \frac{(n\ell+4nm+n^2m)\mathtt{M}+m\mathtt{DGS}+2m\mathtt{PIS}}{(n\ell+4nm+n^2m)\mathtt{M}+m\mathtt{DGS}+2m\mathtt{PIS}+m\mathtt{DGS}+2m\mathtt{PIS}}\\
&= \frac{(n\ell+4nm+n^2m)\mathtt{M}+m\mathtt{DGS}+\{[n2m-0.5(n-1)n+4(2m)^2]\mathtt{M}+2m\,\mathtt{DGS}\}}{(n\ell+4nm+n^2m)\mathtt{M}+2m\mathtt{DGS}+2\{[n2m-0.5(n-1)n+4(2m)^2]\mathtt{M}+2m\,\mathtt{DGS}\}} \quad (\mathbf{Lemma}10)\\
&= \frac{3m\,\mathtt{DGS}+[n(\ell+6m-0.5(n-1))+n^2m+16m^2]\mathtt{M}}{6m\,\mathtt{DGS}+[n(\ell+8m-(n-1))+n^2m+32m^2]\mathtt{M}}\\
&\approx \frac{3m\,\mathtt{DGS}+[n(\ell+6m-0.5n)+n^2m+16m^2]\mathtt{M}}{6m\,\mathtt{DGS}+[n(\ell+8m-n)+n^2m+32m^2]\mathtt{M}}\\
&\approx \frac{3m\,\mathtt{DGS}+(n^2m+16m^2)\,\mathtt{M}}{6m\,\mathtt{DGS}+(n^2m+32m^2)\,\mathtt{M}}\\
&\approx \frac{3m\,\mathtt{DGS}+16m^2\,\mathtt{M}}{6m\,\mathtt{DGS}+32m^2\,\mathtt{M}}\\
&= \frac{1}{2}
\end{aligned}
\tag{25}
$$

Here, the second "≈" holds because $m \approx 6n\log q$, $\ell \ll m$. According to the parameter settings of lattice [38], we have $n \ll 96\log q$, then $n^2m \ll 16m^2$. The signcryption cost of SR_SC is only 1/2 of that of **StE**.

Finally, let us compare the computational cost of the receiver. The proposed scheme saves a verification for the extra signature used in **StE**. This cost is $3nm$ multiplications over $\mathbb{Z}_q$. Because the cost of $3nm$ multiplications over $\mathbb{Z}_q$ only accounts a small percentage of the total cost, the cost of the sender of SR_SC roughly equals that of **StE**.

Above all, the computational overhead of senders and ciphertext extension factor can be reduced approximately to half. In the construction, the translating technique of Canetti *et al.* and Boneh *et al.* [25,26] is borrowed from. However, we guarantee the non-malleable property of the ciphertext by the signature for the message rather than an additional signature for the ciphertext. That is, the ability of the signature for the message is reused once again such that the additional signature is saved. The cost of such a signature scheme is more expensive than an encryption scheme. As a result, the proposed schemes become more efficient.

Remark 2. Note that the **StE** method and the proposed schemes can both use the technique of Boneh and Katz [39] to further improve efficiency. The proposed schemes will save the computation cost brought by a message authentication or an encapsulation [39]. In the proposed schemes, the ciphertext will save the length of a message authentication code and a de-commitment string [39] that should be long enough to be mapped onto an element of $\mathbb{Z}_q^n$.

There have been several IBSC schemes from bilinear pairings [9,40]. Perhaps, some of them are more efficient than ours. However, most of them suffer from the known quantum attacks. Thus, compared with them, the most important advantage of the proposed schemes lies in that our lattice-based constructions have the potential to resist the known quantum attacks.

Remark 3. In this section, the soundness and efficiency for the algorithm SampleRightv are shown by comparison. Recall that SampleRightv is used to sample a pre-image with the matrix form as $[\mathbf{A_1}\|\mathbf{A_1R}+\mathbf{HB}\|\mathbf{C}]$. If this algorithm is not designed, the general method [22] must be used to complete this task. The general method [22] (Section 8), for convenience named ABB-mid-SamplePre, can be formalized as follows.

- $\mathbf{T}' = \mathbf{SampleRightBasis}(\mathbf{A_1}, \mathbf{B}, \mathbf{H}, \mathbf{R}, \mathbf{T_B}, s)$, namely $\mathbf{T}'$ is a basis for $\mathbf{\Lambda}^{\perp}(\mathbf{A}')$, where $\mathbf{A}' = [\mathbf{A_1}\|\mathbf{A_1R}+\mathbf{HB}]$;

- $\mathbf{T}'' = \mathbf{ExtendBasis}(\mathbf{A}', \mathbf{C}, \mathbf{T}')$;
- $\mathbf{v}' = \mathbf{SamplePre}(\mathbf{T}'', \mathbf{A}, \mathbf{y}, s'') \in \mathbb{Z}^{2m+\kappa}$;

Firstly, let us prove that the vector obtained by **SampleRightv** is shorter than that obtained by the ABB-mid-SamplePre method such that it can be used to reply queries in the security proof. According to the consistency of the scheme, in order to make **SampleRight** work well, we set $\|\widetilde{\mathbf{T_B}}\| s_R \omega(\log n) \leq s = m\omega(\log n)$. Then

$$\begin{aligned} \|\mathbf{v}\| &\leq \sqrt{\left[\|\widetilde{\mathbf{T_B}}\|^2 \cdot \omega(\log m)\left(s_R^2 + 1\right) + \eta_\epsilon(\mathbb{Z})^2\right] m} \\ &\leq \sqrt{\left[s^2 + (s/s_R)^2 + \eta_\epsilon(\mathbb{Z})^2\right] m} \leq \sqrt{3s^2 m} \end{aligned} \tag{26}$$

In ABB-mid-SamplePre method, the Gaussian parameter s'' is determined by $\|\widetilde{\mathbf{T}''}\|$ and $\|\widetilde{T}''\| = \|\widetilde{T}'\|$ according to Lemma 3.2 in [32]. The procedure of sampling makes basis become worse, namely, $s'' > s$. Therefore, $\|\mathbf{v}\|$ is less than $\|\mathbf{v}'\| \leq \sqrt{3s''m}$.

Secondly, let us analyze the influence to the reduction time caused by SampleRightv. The total reduction time is the sum of initialization time, queries time, answer time, and time to translate the attacks from the adversary to a solution for the difficulty problem. Because the Gaussian elimination and the Gram–Schmit orthogonalization must be executed in queries under different identities, their cost needs to be considered. By a routine computation, we have a conclusion that the computation overhead of m-dimension `PIS` in [11] is

$$\left(3m^3 - 3m^2 + 3n^2 m - n^3\right)/6 = O(m^3)$$

multiplications, and m DGS, where `PIS` is sampling a pre-image $\mathbf{x}$ to satisfy $\mathbf{Ax} = \mathbf{y}$ for some syndrome $\mathbf{y}, \mathbf{A} \in \mathbb{Z}_q^{n \times m}$. Hence, the initialization time, queries time, and translation time can be negligible compared with the answer time. The answer can be divided into three types, that is, answer for secret key queries, answer for signcryption queries, and answer for unsigncryption queries. The algorithm SampleRightv mainly influences the efficiency of the answer for signcryption queries and unsigncryption queries.The influence can be roughly divided into two categories. (1) The first case is that the adversary asks signcryption or unsigncryption after it asked secret key query for the same identity. From the above expression, for the cost of `PIS`, we conclude that its cost is roughly proportional to the third power of the sample dimension (namely, m^3). Because the dimension of PIS in ABB-mid-SamplePre and SampleRightv is $2m$ and m, respectively, the answer time for a single signcryption query and unsigncryption query by using SampleRightv is 1/8 of that of the ABB-mid-SamplePre method. (2) The second case is that the adversary issues direct signcyption or unsigncryption queries without having asked the secret key query for this identity. The times of SamplePre in ABB-mid-SamplePre method to obtain a basis is $O(m \log m)$ according to Proposition 8. As a result, the answer time for a single query under SampleRightv is only $1/O(m \log m)$ of that of the ABB-mid-SamplePre method.

8. CONCLUSIONS

In this paper, we proposed some IBSC schemes based on lattice hard problems. In the standard model, the schemes are proved indistinguishable against inner chosen ciphertext attacks under LWR/LWE assumptions and strongly unforgeable against inner adaptively chosen message attacks under SIS assumption. The proposed schemes are efficient because of the following two reasons. Firstly, new identity-based signature schemes with shorter signature length are proposed to construct the signcryption. Secondly, in our construction frame, the strongly unforgeable signature used in the black box transformation [25,26] can be removed. Furthermore, no matter the underlying encryption scheme is deterministic or probabilistic, the proposed signcryption schemes can be IND-CCA2. In addition, an efficient simulation algorithm for security proof is designed.

ACKNOWLEDGEMENTS

This work is partially supported by the National Natural Science Foundation of China (NSFC) (no. 61370194) and the NSFC A3 Foresight Program (no. 61411146001). The third author is partially supported by JSPS KAKENHI Grant Number 26730056, JSPS A3 Foresight Program.

REFERENCES

1. Shamir A. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19–22, 1984, Proceedings*, vol. 196, Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 1984; 47–53.
2. Zheng Y. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In *Advances in Cryptology—CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17–21, 1997, Proceedings*, vol. 1294, Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 1997; 165–179.
3. Malone-Lee J, Mao W. Two birds one stone: signcryption using RSA. In *Topics in Cryptology—CT-RSA 2003*, vol. 2612, Joye M (ed), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2003; 211–226.

4. Boyen X. Multipurpose identity-based signcryption—a Swiss army knife for identity-based cryptography. In *Proceedings of CRYPTO 2003*. Springer-Verlag: Berlin Heidelberg, 2003; 383–399.
5. Barreto P, Libert B, McCullagh N, Quisquater JJ. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In *Advances in Cryptology—ASIACRYPT 2005*, vol. 3788, Roy B (ed), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2005; 515–532.
6. Yan J, Wang L, Wang L, Yang Y, Yao W. Efficient lattice-based signcryption in standard model. *Mathematical Problems in Engineering* 2013; **2013**: 18, Article ID 702539, DOI: 10.1155/2013/702539.
7. Malone-Lee J. Identity-based signcryption. *Cryptology ePrint Archive*, 2002. http://eprint.iacr.org/2002/098.pdf.
8. Libert B, Quisquater JJ. A new identity-based signcryption scheme from pairings, *2003 IEEE Information Theory Workshop, 2003. Paris, France, Proceedings*, 2003; 155–158. DOI: 10.1109/ITW.2003.1216718.
9. Boyen. Multipurpose identity-based signcryption: a Swiss army knife for identity-based cryptography, *CRYPTO: Proceedings of Crypto*, 2003.
10. Regev O. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing, STOC '05*. ACM: New York, NY, USA, 2005; 84–93.
11. Gentry C, Peikert C, Vaikuntanathan V. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC '08*. ACM: New York, NY, USA, 2008; 197–206.
12. Peikert C. Public-key cryptosystems from the worst-case shortest vector problem. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*. ACM: New York, 2009; 333–342.
13. Peikert C, Waters B. Lossy trapdoor functions and their applications. *SIAM Journal on Computing* 2011; **40** (6): 1803–1844.
14. Lyubashevsky V. Lattice signatures without trapdoors. In *Advances in Cryptology C EUROCRYPT 2012*, vol. 7237, Pointcheval D, Johansson T (eds), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2012; 738–755.
15. Ducas L, Durmus A, Lepoint T, Lyubashevsky V. Lattice signatures and bimodal Gaussians. In *Advances in Cryptology C CRYPTO 2013*, vol. 8042, Canetti R, Garay J (eds), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2013; 40–56.
16. Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE. In *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*. FOCS '11, IEEE Computer Society: Washington, DC, USA, 2011; 97–106.
17. Brakerski Z, Gentry C, Vaikuntanathan V. Fully homomorphic encryption without bootstrapping. *Electronic Colloquium on Computational Complexity (ECCC)* 2011; **18**: 111–111.
18. Brakerski Z. Fully homomorphic encryption without modulus switching from classical GapSVP. In *Advances in Cryptology C CRYPTO 2012*, vol. 7417, Safavi-Naini R, Canetti R (eds), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2012; 868–886.
19. Gentry C, Sahai A, Waters B. Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology C CRYPTO 2013*, vol. 8042, Canetti R, Garay J (eds), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2013; 75–92.
20. Agrawal S, Freeman D, Vaikuntanathan V. Functional encryption for inner product predicates from learning with errors. In *Advances in Cryptology C ASIACRYPT 2011*, vol. 7073, Lee D, Wang X (eds), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2011; 21–40.
21. Gorbunov S, Vaikuntanathan V, Wee H. Attribute-based encryption for circuits. In *Proceedings of the 45th Annual ACM Symposium on Symposium on Theory of Computing, STOC '13*. ACM: New York, NY, USA, 2013; 545–554.
22. Agrawal S, Boneh D, Boyen X. Efficient lattice (H)IBE in the standard model. In *Advances in Cryptology C EUROCRYPT 2010*, vol. 6110, Gilbert H (ed), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2010; 553–572.
23. Boyen X. Lattice mixing and vanishing trapdoors: a framework for fully secure short signatures and more. In *Public Key Cryptography C PKC 2010*, vol. 6056, Nguyen P, Pointcheval D (eds), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2010; 499–517.
24. Waters B. Efficient identity-based encryption without random oracles. In *Advances in Cryptology C EUROCRYPT 2005*, vol. 3494, Cramer R (ed), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2005; 114–127.
25. Canetti R, Halevi S, Katz J. Chosen-ciphertext security from identity-based encryption. In *Advances in Cryptology—EUROCRYPT'2004*, vol. 3027,

Cachin C, Camenisch J (eds), Lecture Notes in Computer Science. Springer: Berlin, 2004; 207–222.
26. Boneh D, Canetti R, Halevi S, Katz J. Chosen-ciphertext security from identity-based encryption. *SIAM Journal of Computing* 2006; **36**(5): 1301–1328.
27. Micciancio D, Goldwasser S. *Complexity of Lattice Problems: A Cryptographic Perspective*, Vol. 671. Springer: Berlin Heidelberg, 2002.
28. Alwen J, Krenn S, Pietrzak K, Wichs D. Learning with rounding, revisited. In *Advances in Cryptology C CRYPTO 2013*, vol. 8042, Canetti R, Garay J (eds), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2013; 57–74.
29. Micciancio D, Regev O. Worst-case to average-case reductions based on Gaussian measure. *SIAM Journal on Computing* 2007; **37** (1): 267–302, DOI: 10.1137/S0097539705447360. Preliminary version in FOCS 2004.
30. Alwen J, Peikert C. Generating shorter bases for hard random lattices. *Theory of Computing Systems* 2011; **48**: 535–553.
31. Krawczyk H, Rabin T. Chameleon hashing and signatures. *IACR Eprint archive*, 1998. http://eprint.iacr.org/1998/010, appeared in the THEORY OF CRYPTOGRAPHY LIBRARY and has been included in the ePrint Archive. talr@watson.ibm.com 10500 received March 17th, 1998.
32. Cash D, Hofheinz D, Kiltz E, Peikert C. Bonsai trees, or how to delegate a lattice basis. *Journal of Cryptology* 2012; **25**: 601–639.
33. Dent A. Hybrid signcryption schemes with insider security. In *Information Security and Privacy*, vol. 3574, Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2005; 253–266.
34. Chen L, Malone-Lee J. Improved identity-based signcryption. In *Public Key Cryptography—PKC 2005*, vol. 3386, Vaudenay S (ed), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2005; 362–379.
35. Banerjee A, Peikert C, Rosen A. Pseudorandom functions and lattices. In *Advances in Cryptology C EUROCRYPT 2012*, vol. 7237, Pointcheval D, Johansson T (eds), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2012; 719–737.
36. Aharonov D, Regev O. Lattice problems in NP insert CONP. *Journal of the ACM* 2005; **52**(5): 749–765.
37. Shoup V. *A Computational Introduction to Number Theory and Algebra*, Vol. 2nd edn. Cambridge University Press: Cambridge, 2008.
38. Lindner R, Peikert C. Better key sizes (and attacks) for LWE-based encryption, *In CT-RSA*, San Francisco, CA, USA, 2011; 319–339.
39. Boneh D, Katz J. Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In *Topics in Cryptology C CT-RSA 2005*, vol. 3376, Menezes A (ed), Lecture Notes in Computer Science. Springer: Berlin Heidelberg, 2005; 87–103.
40. Malone-Lee J. Identity-based signcryption. *IACR Cryptology ePrint Archive* 2002; **2002**: 98.

APPENDIX

Lemma 10. *The computational cost of m-dimension* `PIS` *is about* $[nm-\frac{1}{2}(n-1)n+4m^2]$ *multiplications over* $\mathbb{Z}_q$(`M`) *and m* `DGS` *when omitting the cost of the Gram–Schmidt orthogonalization and the Gaussian elimination.*

Proof. The aim of `PIS` is to sample a short vector **x** satisfying **Ax** = **y** for "parity check" matrix **A** and some fixed vector **y** (see Section 5.3 in [11]). Let matrix **B** be a basis of $\Lambda^{\perp}(\mathbf{A})$. The primary operations involved in the procedure of `PIS` and the corresponding cost are as follows:

- Gram-Schmidt orthogonalization over **B**: the result matrix can be stored to avoid repeated calculation.
- Solving an equation **At** = **y**: the Gaussian elimination will be executed over **A**, and the result matrix can also be stored to avoid repeated calculation. The number of multiplications over $\mathbb{Z}_q$ in the back substitution is

$$c' = [m-(n-1)]+[m-(n-2)]+\cdots+[m-(n-n)] = nm-\frac{1}{2}(n-1)n$$

- $4m^2$ multiplications over $\mathbb{Z}$ less than q: for simplicity, let us regard this cost as $4m^2$ multiplications over $\mathbb{Z}_q$.
- m `DGS`.

As a result, the total computational cost is $\left[nm-\frac{1}{2}(n-1)n+4m^2\right]$ `M` $+ m$ `DGS`. □